

CENTRO UNIVERSITÁRIO SENAC

**André Dias Marques
Gustavo Borges Martins
Gustavo Henrique Monteiro**

**Big Data auxiliando na redução do tempo de deslocamento de
veículos de emergência**

São Paulo, 30 de novembro de 2018.

**ANDRÉ DIAS MARQUES
GUSTAVO BORGES MARTINS
GUSTAVO HENRIQUE MONTEIRO**

**Big Data auxiliando na redução do tempo de deslocamento de veículos
de emergência**

Trabalho de conclusão de curso
apresentado ao Centro Universitário
SENAC – Campus Santo Amaro, como
exigência parcial para obtenção do grau
de Especialista em Big Data.

Orientador:
Prof. Msc. Alexandre Machado

São Paulo, 30 de novembro de 2018.

Elaborada pelo sistema de geração automática de ficha catalográfica do Centro Universitário Senac São Paulo com dados fornecidos pelo autor(a).

Borges, Gustavo

Big Data auxiliando na redução do tempo de deslocamento de veículos de emergência. / Gustavo Borges, André Marques, Gustavo Monteiro - São Paulo (SP), 2018.
88 f.: il. color.

Orientador(a): Alexandre Machado

Trabalho de Conclusão de Curso (Pós-Graduação em Especialização em Big Data) - Centro Universitário Senac, São Paulo, 2018.

1. Gestão de tráfego, 2. Big Data, 3. Emergência, 4. IoT I. Marques, André II. Monteiro, Gustavo III. Machado, Alexandre (Orient.) IV. Título

Aos nossos companheiros e parceiros que estiveram presentes durante toda jornada.

AGRADECIMENTOS

Agradecemos a todas as pessoas que de alguma forma contribuíram para a realização deste trabalho. Em especial agradecemos ao professor Alexandre Machado por sua paciência e sabedoria para nos orientar e para estruturar nosso TCC. Por último, e o mais importante, agradecemos a Deus por estar sempre ao nosso lado nas horas mais difíceis, que não foram poucas. Obrigado.

Aos professores do SENAC Alexandre Machado e João Carlos Néto. *“A tecnologia tornou possível a existência de grandes populações. Grandes populações agora tornam a tecnologia indispensáveis”.*
Autor Joseph Krutch

RESUMO

O trânsito congestionado é um problema recorrente em grandes centros metropolitanos, tornando difícil a locomoção de ambulâncias em meio a tantos veículos parados. Com base nas tecnologias recentes é possível aumentar a eficácia e temporizar os semáforos para que estes possam dar maior fluência ao trânsito engarrafado. Entretanto, mesmo com essa evolução nos sistemas semaforicos, os veículos de emergência como, por exemplo, as ambulâncias ainda ficam presas ao trânsito sem poder trafegar e chegar ao seu destino o mais rápido possível para que haja o atendimento médico em tempo hábil. O objetivo deste trabalho é reduzir o deslocamento de ambulâncias em vias com grande fluxo de tráfego de automóveis, por meio da implementação de um protótipo utilizando tecnologias de Big Data que emula as condições reais de trânsito de vias de grandes cidades, otimizando o tempo dos semáforos, para que a ambulância possa fluir com maior rapidez.

Palavras-chave: Gestão de tráfego, Big Data, Emergência, IoT.

ABSTRACT

Congested traffic is a recurring problem in large metropolitan centers, making ambulance locomotion difficult amid so many stationary vehicles. Based on recent technologies, it is possible to increase the efficiency and timing of traffic lights so that they can give greater influence to bottled traffic. However, even with this evolution in light systems, emergency vehicles such as ambulances are still stuck in traffic without being able to travel and reach their destination as fast as possible so that medical attention is available in a timely manner. The aim of this work is to reduce the displacement of ambulances in a high traffic flow of cars, through the implementation of a prototype based on Big Data technologies that emulate the real traffic conditions of major city roads, optimizing the traffic lights time, so the ambulance can flow faster.

Keywords: Traffic management, Big Data, Emergency, IoT.

SUMÁRIO

| | |
|---|-----------|
| 1. INTRODUÇÃO | 12 |
| 1.1. Objetivo | 13 |
| 1.1.1. Objetivos Gerais..... | 13 |
| 1.1.2. Objetivos Específicos | 13 |
| 1.2. Abrangência..... | 14 |
| 1.3. Trabalhos Relacionados | 14 |
| 1.4. Organização do Trabalho | 15 |
| 2. FUNDAMENTAÇÃO TEORICA DE PESQUISA..... | 16 |
| 2.1. O Problema do Caixeiro Viajante | 16 |
| 2.2. O Algoritmo de Dijkstra | 16 |
| 2.3. Internet das Coisas (IoT)..... | 17 |
| 2.4. MongoDB..... | 17 |
| 2.5. NEO4j..... | 18 |
| 2.6. Apache Spark..... | 20 |
| 2.7. Python | 20 |
| 2.8. Urgência e Emergência..... | 21 |
| 2.8.1. Principais órgãos que prestam serviços de emergência | 22 |
| 2.9. Circulação de Veículos de Emergência..... | 23 |
| 3. METODOLOGIA | 27 |
| 3.1. Critérios de seleção da metodologia de pesquisa | 27 |
| 3.2. A Definição do Projeto de Pesquisa de Estudo de Caso | 27 |
| 3.2.1. As Questões de Estudo | 30 |
| 3.2.2. As Proposições | 30 |
| 3.2.3. A Unidade de Análise..... | 31 |
| 3.2.4. A Lógica que Une os Dados às Proposições | 32 |
| 3.2.5. Os Critérios para Interpretar as Constatações | 32 |
| 3.2.6. A Coleta de Evidências..... | 33 |
| 3.3. A Análise Qualitativa das Evidências | 33 |
| 3.4. Framework | 34 |
| 3.5. Definição do Melhor Caminho | 34 |
| 3.6. Otimização do Trajeto | 36 |
| 3.7. Arquitetura Física | 37 |
| 3.7.1. Camada de Emulação do Trânsito..... | 37 |
| 3.7.2. Camada de Análise dos Dados dos Sensores..... | 40 |
| 3.7.3. Camada de Definição e Otimização de Trajeto | 42 |
| 3.7.4. Topologia | 44 |
| 3.7.5. Métricas de Qualidade | 45 |
| 4. ESTUDO DE CASO | 46 |
| 4.1. Comparação com o Waze | 50 |
| 4.2. Comparação com o Google Maps / Routes..... | 51 |
| 4.3. Comparação com outros Algoritmos | 54 |

| | | |
|---------------|---|-----------|
| 5. | AVALIAÇÃO DOS RESULTADOS | 55 |
| 5.1. | Comparação com o Waze | 55 |
| 5.2. | Comparação com o Google Maps / Routes..... | 57 |
| 5.2.1. | Cenário 1 – Baixo Fluxo..... | 58 |
| 5.2.2. | Cenário 2 – Médio Fluxo | 59 |
| 5.2.3. | Cenário 3 – Fluxo Caótico | 61 |
| 6. | CONCLUSÃO | 64 |
| 6.1. | Limitações..... | 65 |
| 6.2. | Trabalhos Futuros | 66 |
| | REFERÊNCIAS BIBLIOGRÁFICAS..... | 67 |
| | APÊNDICE A – MANUAL DO USUÁRIO..... | 71 |
| A1. | Importando a Máquina virtual | 71 |
| A2. | Introdução a arquitetura | 75 |
| A3. | Iniciando o MongoDB | 76 |
| A4. | Iniciando o Neo4j..... | 78 |
| A5. | Iniciando o Spark..... | 80 |
| A6. | Executando o algoritmo da Camada de Emulação do Trânsito..... | 81 |
| A7. | Executando o algoritmo da Camada de Análise dos Dados dos Sensores | 83 |
| A8. | Executando o algoritmo Camada de Definição e otimização do Trajeto | 87 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 - Pesquisa realizada pela StackOverflow sobre bancos de dados mais utilizados. Extraído de: (Stackoverflow, 2018) | 18 |
| Figura 2 - Ranking de banco de dados gráfico mais populares. Extraído de: (DB-Engines, 2018) | 19 |
| Figura 3 - Ranking de linguagens de programação mais populares em 2018. Extraído de: (IEEE Spectrum, 2018) | 21 |
| Figura 4 - Fluxo de veículos da cidade de São Paulo em 2018. Adaptado de (DETRAN SP, 2018) | 31 |
| Figura 5 - Intersecções do Trajeto. Adaptado de (Google Maps, 2018) | 35 |
| Figura 6 - Funcionamento do Algoritmo. Extraído de (Dijkstra, 2018) | 36 |
| Figura 7 - Ilustração o Funcionamento do Algoritmo | 39 |
| Figura 8 - Processo de Captura dos Arquivos Gerados para obter a Velocidade Média | 40 |
| Figura 9 - Definição de node e aresta. Adaptado de (Google Maps, 2018) | 42 |
| Figura 10 - Visualização do Processo para Definição da Melhor Rota..... | 44 |
| Figura 11 - Projeto Sinal Verde da CET: Semáforos em operação no trajeto. Extraído de: (Sinal Verde, 2018) | 47 |
| Figura 12 - Mapeamento de Interseções das vias e de seus trajetos. Adaptado de: (Google Maps, 2018) | 48 |
| Figura 13 - Visualização do grafo referente a região estudada. | 49 |
| Figura 14 - Tabela de Comparação do Trajeto..... | 50 |
| Figura 15 - Trajetos sugeridos. Adaptado de: (Waze, 2018) | 51 |
| Figura 16 - Trajeto analisado no cenário 1. Extraído de: (Google Maps, 2018) | 52 |
| Figura 17 - Trajeto analisado no cenário 2. Extraído de (Google Maps, 2018) | 53 |
| Figura 18 - Trajeto analisado no cenário 3. Extraído de (Google Maps, 2018) | 54 |
| Figura 19 - Trajetos sugeridos. Adaptado de: (Waze, 2018) | 57 |
| Figura 20 - Concentração de veículos por região análise do cenário 2. Adaptado de (Heatmapper, 2018) | 61 |
| Figura 21 - Concentração de veículos por região análise do cenário 3. Adaptado de (Heatmapper, 2018) | 63 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Dados do percurso obtidos pelo algoritmo. | 56 |
| Tabela 2 – Velocidade média do trajeto selecionado pelo algoritmo..... | 58 |
| Tabela 3 – Informações detalhadas do trajeto sugerido, tempo e velocidade média para o Cenário 1 – Baixo Fluxo | 59 |
| Tabela 4 - Informações detalhadas do trajeto sugerido, tempo e velocidade média para o Cenário 2 – Médio Fluxo | 60 |
| Tabela 5 - Informações detalhadas do trajeto sugerido, tempo e velocidade média para o Cenário 3 – Fluxo Caótico | 62 |

1. INTRODUÇÃO

Uma das áreas críticas no setor da saúde é a que congrega as organizações que disponibilizam serviços de urgência médica. Considerando que o rápido e adequado atendimento pode salvar vidas, especialmente quando a pessoa atendida precisa ser removida para um centro hospitalar, ou necessita ser atendida de imediato no próprio local onde se encontra, percebe-se então a enorme relevância deste tipo de atividade.

Os problemas relacionados ao deslocamento de veículos em vias de grandes centros urbanos são uma realidade conhecida de muitos brasileiros. Congestionamentos que são causados pela falta de inteligência em como as autoridades gerenciam o trânsito é um problema comum em grandes metrópoles. Com soluções de Big Data pode ser possível resolver parte destes problemas, como o caso de lentidão no deslocamento de veículos de emergência por conta de grandes congestionamentos causados principalmente por semáforos que atuam de forma independente uns dos outros.

O perfil da mortalidade se alterou ao longo das últimas décadas, tanto no Brasil, quanto no mundo. Se por um lado, a melhoria das condições sanitárias e os progressos da medicina reduziram as mortes por vários tipos de doenças, a massificação do automóvel, o sedentarismo e a violência urbana, dentre outros fatores, criaram ou acentuaram urgências médicas: traumas (acidentes de trânsito) e clínicas (acidentes cardiovasculares, por exemplo), que por sua vez levam ao óbito das vítimas (Tani, 2003).

De acordo com o 2º Sargento do Corpo de Bombeiros de São Bernardo, José Rubens Zaparoli, nos horários de maior pico de trânsito, as 7H às 8H30 e das 16H às 21H, o número de acidentes envolvendo motos também é elevado, piora a circulação de veículos e aumenta o tempo de viagem até a emergência (Dos Anjos et al., 2010). Como é possível notar neste depoimento o trânsito é o maior problema enfrentado por veículos de emergência em deslocamento por vias urbanas.

Porém, muitas dessas mortes poderiam ser evitadas se o atendimento à vítima ocorresse nos primeiros instantes após a ocorrência da causa da urgência médica, pois esse tempo é determinante para a sua sobrevivência (Elliot, 2000).

1.1. Objetivo

Os objetivos deste trabalho estão subdivididos nas seções abaixo:

1.1.1. Objetivos Gerais

O objetivo geral deste artigo é demonstrar que aplicando as tecnologias atuais, principalmente tecnologias de Big Data como o Internet of Things (IoT), é possível diminuir o tempo de deslocamento de veículos de emergência, como ambulâncias, através da mudança em tempo real dos tempos de semáforos no trajeto deste veículo, fazendo o trânsito fluir mais rapidamente e consequentemente fazendo o veículo de emergência diminuir seu tempo de deslocamento até o ponto final de atendimento.

1.1.2. Objetivos Específicos

Para objetivos específicos os seguintes itens foram levantados:

- ✓ Definir arquitetura e implementação da mesma no formato de protótipo;
- ✓ Efetuar a seleção do melhor ferramental aplicado a arquitetura;
- ✓ Elaborar modelo de simulação de trânsito;
- ✓ Implementar o Apache Spark para recebimento e processamento das informações das etapas anteriores;
- ✓ Implementar banco de dados de grafos para servir de ajuda ao modelo para escolher a rota com o menor tempo de deslocamento;
- ✓ Gerar dados de logs para que seja possível avaliar a eficácia do modelo;
- ✓ Demonstrar no protótipo a diferença de tempo entre o deslocamento comum e o deslocamento baseado no modelo;

1.2. Abrangência

Os objetos de entrega desse estudo contemplam a busca pela melhoria do desempenho das atividades operacionais de serviços que atuam com atendimento de emergências médicas e transporte de pacientes, para isso será elaborado um protótipo de aplicação que forneça as funcionalidades necessárias para controle de semáforos inteligentes. Para isso todas as etapas mencionadas nos objetivos específicos serão implementadas. Por ser um protótipo exclui-se assim a necessidade de criação de hardware específico e troca de informações com sistemas de semáforos reais. Será necessário a utilização de simuladores que consigam simular o comportamento de vias terrestres, veículos e semáforos para troca de informações com o sistema.

1.3. Trabalhos Relacionados

O trabalho para a melhora da eficácia da programação de semáforos vem sendo tema de diversos artigos e aplicações no decorrer do tempo. Uma empresa que opera no Brasil possui um sistema que melhora a resposta dos semáforos de acordo com o número de veículos na via (Seebot, 2017). Esta empresa demonstra que é possível aplicar inteligência ao trânsito de cidades brasileiras com o seu sistema de gerenciamento de semáforos que utilizam tecnologias que envolvem Big Data.

O artigo “Análise de interação do Waze nas condições do trânsito na cidade de São Paulo” também utiliza tecnologia para diminuição dos grandes engarrafamentos na cidade de São Paulo (Silveira et al., 2015).

“Streaming Computing que trabalha com dados em tempo real e grande fluxo de dados, como, em sistemas de trânsito, que monitoram o tráfego de veículos em determinadas cidades, e que transmitem ao usuário qual a melhor rota a ser tomada para chegar ao seu destino, através de seus algoritmos, traz soluções práticas e rápidas aos seus usuários.” Retirado do artigo: “Big Data: Ferramentas e aplicabilidade” (Galdino, 2016).

Neste contexto de soluções prévias que já auxiliam hoje em dia a programação mais eficiente de semáforos nas grandes cidades, este trabalho tem como objetivo mostrar como estas soluções já existentes podem servir de base para uma solução mais abrangente que sirva também para temporização dos semáforos com o intuito de diminuir o tempo de deslocamento dos veículos de emergência até os pontos de atendimento.

1.4. Organização do Trabalho

O artigo está organizado da seguinte forma para que o objetivo seja atendido:

- ✓ O capítulo 1 – Introdução, aborda os objetivos gerais, específicos, abrangências e trabalhos relacionados;
- ✓ O capítulo 2 – Fundamentação Teórica, aborda as técnicas de pesquisa utilizadas no desenvolvimento do trabalho;
- ✓ O capítulo 3 – Metodologias, aborda as metodologias utilizadas em diferentes níveis no processo de desenvolvimento do trabalho;
- ✓ O capítulo 4 – Estudo de Caso, aborda todos os cenários analisados e aplicações utilizadas sobre eles.
- ✓ O capítulo 5 – Avaliação de Resultados, são apresentados os resultados obtidos com aplicação do algoritmo em comparação ao cenário momentâneo.
- ✓ O capítulo 6 – Considerações Finais, onde é apresentada nossas conclusões finais e definições sobre o protótipo/algoritmo criado.

2. FUNDAMENTAÇÃO TEORICA DE PESQUISA

Para atingir os objetivos propostos para este trabalho foram aplicadas as técnicas de pesquisa abaixo, tão bem como as respectivas ferramentas que serão utilizadas:

2.1. O Problema do Caixeiro Viajante

O **problema do caixeiro viajante** (PCV) é um problema hipotético que visa diminuir o tempo de trajeto entre dois ou mais pontos, inicialmente ele foi concebido com o intuito de otimizar o trajeto de vendedores para que os mesmos pudessem efetuar o maior percurso com o menor tempo de atuação. Embora o algoritmo tenha sido concebido com esse objetivo o mesmo possui uma aplicabilidade para os problemas logísticos de uma forma geral.

Segundo Conte (Conte, 2002):

“Formalmente, o PCV pode ser definido considerando um grafo completo. Dado $G = (V, A)$ um grafo onde V é o conjunto de N vértices e A é o conjunto de arcos ou arestas que conectam cada par de cidades $I, J \in V$. A cada arco/aresta está associado um custo C_{ij} . O PCV consiste em encontrar a rota de menor custo, passando por cada vértice uma única vez. No caso simétrico, $C_{ij} = C_{ji}$ para toda cidade $i, j \in V$, enquanto o caso assimétrico possui pelo menos um caso em que $C_{ij} \neq C_{ji}$.”

2.2. O Algoritmo de Dijkstra

O algoritmo de Dijkstra, elaborado pelo cientista da computação holandês Edsger Dijkstra em 1956 e publicado no ano de 1959, foi desenvolvido com o intuito de solucionar o problema do caminho mais curto em um grafo. A sua abordagem permite a adição de pesos ao trajeto permitindo assim a customização de suas regras de acordo com a necessidade, inicialmente essa metodologia foi implementada nos chamados GPS (Global Positioning System).

Segundo Carvalho (Carvalho, 2008):

“O algoritmo de Dijkstra (E.W. Dijkstra) é um dos algoritmos que calcula o caminho de custo mínimo entre vértices de um grafo [...]. Este algoritmo parte de uma estimativa inicial para a distância mínima, que é considerada infinita ($^{\infty}$), e vai sucessivamente ajustando esta distância. Ele considera que uma cidade estará “fechada” quando já estiver sido obtido um caminho de distância mínima da cidade tomada como origem da busca até ela.”

2.3. Internet das Coisas (IoT)

A **Internet das Coisas (IOT)** é um conceito onde diversos objetos são conectados a uma rede comum e possuem a capacidade de coletar e transmitir dados.

Segundo Santos (Santos et al., 2016):

“Internet das Coisas, em poucas palavras, nada mais é que uma extensão da internet atual, que proporciona aos objetivos do dia-a-dia (quaisquer que sejam), mas com capacidade computacional e de comunicação, se conectarem a internet. A conexão com a rede mundial de computadores viabilizará, primeiro, controlar remotamente os objetos e, segundo, permitir que os próprios objetos sejam acessados como provedores de serviços.”

O objetivo principal do uso desse tipo de tecnologia é o trabalho com sensores, que atuam em rede e com baixo consumo de energia, sendo ideal para trabalhar no monitoramento dos semáforos.

2.4. MongoDB

O armazenamento de informações geradas pelo protótipo do projeto é necessário para que seja possível efetuar os cálculos necessários. Para esta finalidade foi escolhido o banco de dados NoSql MongoDB. Dentre alguns motivos que levaram a escolha deste banco de dados para o armazenamento das informações estão:

- ✓ O MongoDB é um banco de dados distribuído em seu núcleo, de modo que a alta disponibilidade, o dimensionamento horizontal e a distribuição geográfica são integrados e fáceis de usar (MongoDB, 2018).
- ✓ MongoDB é grátis e open-source. (MongoDB, 2018).

Como as informações são armazenadas em BSON, objetos com estrutura semelhante a JSON, a sua facilidade de uso para troca de informações que utilizam JSON fica facilitada. Além de todas as vantagens mencionadas o MongoDB é atualmente um dos bancos de dados mais utilizados no mundo, dentre os bancos de dados NoSql é o mais utilizado, como é demonstrado na Figura 1 onde se tem os bancos de dados mais utilizados no mundo em 2017 (entre relacionais e não relacionais). Esses números foram extraídos da pesquisa Developer Survey Results do site Stackoverflow:

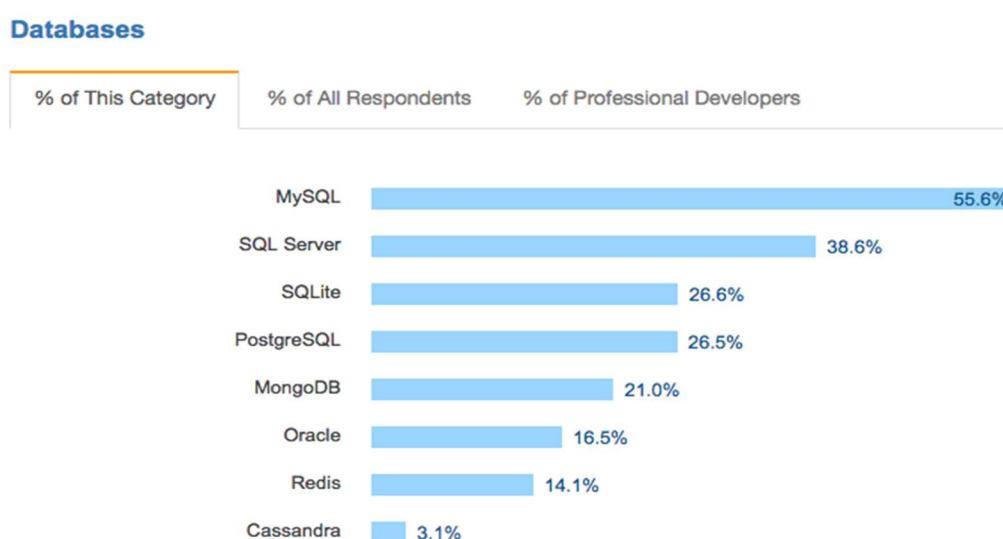


Figura 1 - Pesquisa realizada pela StackOverflow sobre bancos de dados mais utilizados. Extraído de: (Stackoverflow, 2018)

2.5. NEO4j

O Neo4j é um banco de dados do tipo gráfico. Este tipo de banco de dados foi utilizado no projeto por prover um sistema mais eficaz e veloz para os tipos de dados e ligações que são utilizadas no protótipo do projeto. Como trabalha com nós e arestas para o armazenamento dos dados foi considerado o ideal para o

armazenamento das informações que dizem respeito aos semáforos, suas distancias e vias que os ligam. A Figura 2 mostra o rank dos bancos de dados que suportam grafos, e sua evolução no rank em um período de um ano. Neste rank o Neo4J aparece como o mais utilizado desde o ano de 2017 até a extração do rank realizada em setembro de 2018.

31 sistemas no ranking, setembro de 2018

| Classificação | | | DBMS | Modelo de banco de dados | Ponto | | |
|------------------|----------------|------------------|---------------------------|--------------------------|------------------|----------------|------------------|
| Setembro de 2018 | Agosto de 2018 | Setembro de 2017 | | | Setembro de 2018 | Agosto de 2018 | Setembro de 2017 |
| 1 | 1 | 1 | Neo4j | SGBD gráfico | 40,10 | -0,83 | +1,67 |
| 2 | 2 | 2 | Microsoft Azure Cosmos DB | Modelo múltiplo | 19,18 | -0,35 | +7,95 |
| 3 | 3 | | Datastax Enterprise | Modelo múltiplo | 7,76 | +0,46 | |
| 4 | 4 | ↓ 3 | OrientDB | Modelo múltiplo | 5,48 | +0,57 | -0,42 |
| 5 | 5 | 5 | ArangoDB | Modelo múltiplo | 4,05 | +0,71 | +1,05 |
| 6 | 6 | 6 | Virtuoso | Modelo múltiplo | 2,06 | +0,01 | +0,17 |
| 7 | ↑ 8 | | Amazon Netuno | Modelo múltiplo | 1,12 | +0,31 | |
| 8 | ↓ 7 | ↓ 7 | Giraph | SGBD gráfico | 1,02 | +0,03 | -0,05 |
| 9 | ↑ 11 | ↑ 16 | JanusGraph | SGBD gráfico | 0,90 | +0,36 | +0,68 |
| 10 | 10 | ↓ 9 | GraphDB | Modelo múltiplo | 0,63 | +0,06 | +0,02 |
| 11 | ↓ 9 | ↓ 8 | AllegroGraph | Modelo múltiplo | 0,60 | +0,02 | -0,04 |
| 12 | 12 | ↓ 10 | Stardog | Modelo múltiplo | 0,54 | +0,01 | -0,04 |
| 13 | ↑ 17 | 13 | Dgraph | SGBD gráfico | 0,41 | +0,17 | +0,14 |
| 14 | ↑ 15 | ↑ 15 | Blazegraph | Modelo múltiplo | 0,36 | +0,08 | +0,12 |
| 15 | ↓ 13 | ↓ 11 | Sqrrl | Modelo múltiplo | 0,34 | -0,00 | -0,17 |

Figura 2 - Ranking de banco de dados gráfico mais populares. Extraído de: (DB-Engines, 2018)

Este tipo de banco de dados foi utilizado no projeto por prover um sistema mais eficaz e veloz para os tipos de dados e ligações que são utilizadas no protótipo do projeto. Como trabalha com nós e arestas para o armazenamento dos dados foi considerado o ideal para o armazenamento das informações que dizem respeito aos semáforos, suas distancias e vias que os ligam. Além das razões de projeto, abaixo são citadas outras razões pela qual o Neo4j foi escolhido:

- ✓ Escalabilidade de leitura e gravação de alto desempenho (Neo4j, 2018).
- ✓ Alto desempenho graças ao armazenamento e processamento de grafos nativos (Neo4j, 2018).
- ✓ Maior e mais ativa comunidade gráfica do planeta (Neo4j, 2018).

2.6. Apache Spark

O **Apache Spark** é um mecanismo de análise unificada para processamento de dados em larga escala (Apache Spark, 2018). Seu objetivo principal é processar de forma paralela e distribuída grandes conjuntos de dados. Com o Apache Spark é possível distribuir o processamento de um conjunto de dados por um cluster com vários nós e processá-los paralelamente sendo possível assim diminuir o tempo de processamento, podendo chegar a um processamento até cem vezes mais rápido que o processamento com o Apache Hadoop (Apache Spark, 2018). Além das operações de MapReduce, suporta consultas SQL, streaming de dados, aprendizagem de máquina e processamento de grafos, esses recursos podem ser usados no modo stand-alone ou combinados em um único pipeline (Penchikala, 2015). Pipeline é uma técnica de hardware que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada, em resumo é processo pelo qual uma instrução de processamento é subdividida em etapas, uma vez que cada uma dessas etapas é executada por uma porção especializada da CPU, podendo colocar mais uma instrução em execução simultaneamente, isso traz um uso racional da capacidade computacional com ganho de velocidade (Shen, 2013). Para o processamento de dados com fluxo constante, onde uma das necessidades é processamento em tempo real, é utilizada uma das bibliotecas do Apache Spark, o Apache Spark Streaming. Através do uso de microbatches e janelas móveis o Apache Spark consegue entregar processamento perto do tempo real de um grande conjunto de dados, fazendo dele a ferramenta ideal para o processamento dos conjuntos de dados do apresentado no artigo.

2.7. Python

O Python é uma linguagem de programação interpretada, interativa e orientada a objetos (Python, 2018). É open source inclusive para projetos comerciais e possui um grande suporte da comunidade de desenvolvedores que desenvolve frequentemente módulos e pacotes para uma grande variedade de aplicações. Pode ser usada para uma grande variedade de atividades como aplicações web,

desktop, e móvel e tudo isso sendo multiplataforma. É atualmente uma das linguagens de programação mais utilizadas por quem trabalha com análise de dados e tratamento de grandes bases de dados por utilizar uma sintaxe mais clara e de fácil uso se comparada a outras linguagens de programação do mercado. Python despontou como uma das linguagens mais utilizadas para Data Science ao lado de outras linguagens como Java e R. Mas Python obteve grande vantagem competitiva devido à grande quantidade de pacotes e módulos prontos que podem ser utilizados por cientistas de dados. A Figura 3 mostra o rank das linguagens de programação mais utilizadas em 2018 segundo a IEEE Spectrum, bem como para quais propósitos essas linguagens são mais utilizadas atualmente.

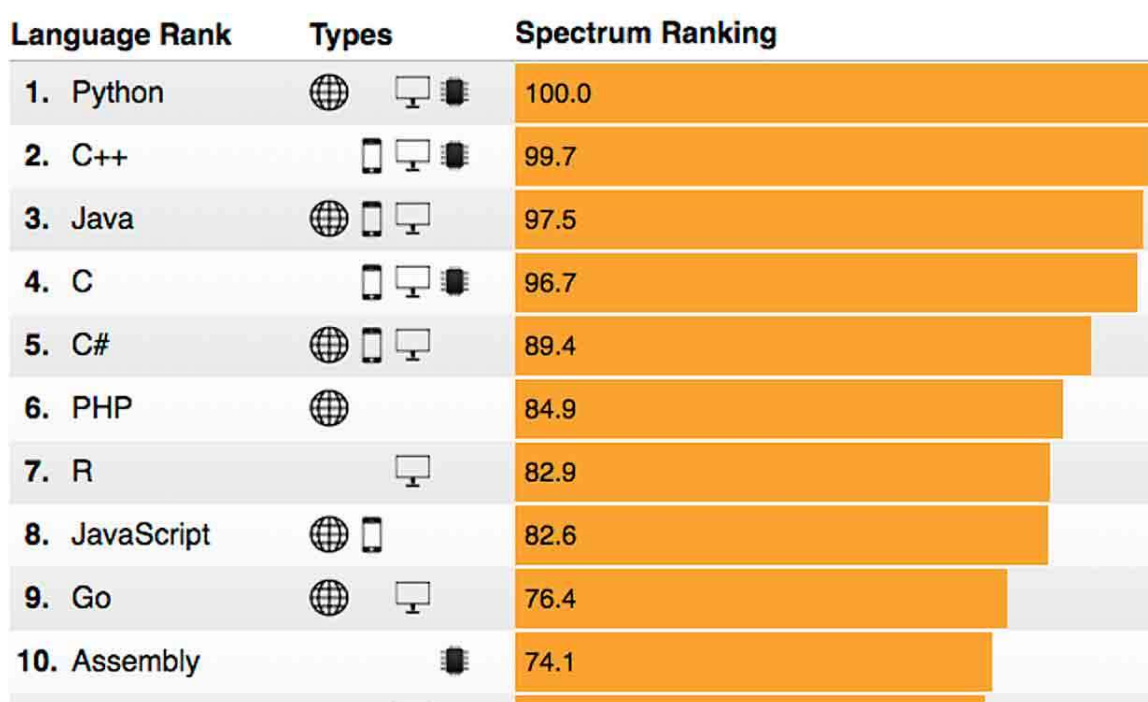


Figura 3 - Ranking de linguagens de programação mais populares em 2018. Extraído de: (IEEE Spectrum, 2018)

2.8. Urgência e Emergência

Para o propósito definido nesse artigo define-se por emergência a constatação médica de condições de agravo à saúde que implique em risco iminente de morte ou sofrimento intenso, exigindo, portanto, tratamento médico imediato. (Silva, 2018).

Define-se por urgência a ocorrência imprevista de agravo a saúde com ou sem risco potencial de vida, cujo portador necessita de assistência médica imediata. (Silva, 2018).

2.8.1. Principais órgãos que prestam serviços de emergência

✓ Serviço de Atendimento Móvel de Urgência - Samu (192)

O SAMU é um serviço pré-hospitalar que tem como principal finalidade o atendimento ágil a vítimas em situações de emergência ou urgência. O atendimento funciona 24hrs por dia e 7 dias por semana, podem ser realizados em qualquer lugar, desde residências, empresas, parques, ruas, dentre outros. O acionamento é feito via contato telefônico, gratuita, passando por uma triagem onde é identificado o nível de urgência, e após isso direcionada a um médico regulador, que é responsável por iniciar o atendimento, diagnosticar e definir a melhor orientação ao paciente, ou a pessoa que fez a chamada.

Após análise do médico regulador, o mesmo deve designar o procedimento adequado para o paciente/contato, seja ela a ida a um hospital, posto de saúde, ou até mesmo o acionamento de uma ambulância com socorristas, médicos e auxiliares para deslocamento até o local do fato. Esse médico regulador tem autoridade sanitária, podendo assim comunicar a urgência ou emergência a hospitais e postos de saúde, assim como disponibilidade de serviços necessários para o atendimento. O SAMU deve atender todos os tipos de chamadas, de quaisquer naturezas, sejam clínicas ou traumáticas.

✓ Corpo de Bombeiros (193)

É uma corporação cujo processo de acionamento segue os padrões informado no exemplo anterior SAMU, a atividade de atendimento pré-hospitalar é atualmente uma das principais atividades da corporação, tem como foco o atendimento a vítimas de acidentes de trânsito, soterramentos, incêndios, ferimentos por armas de fogo ou brancas, problemas de risco iminente a vida, dentre outros.

Realizando diariamente o socorro adequado em condições diversas, assim como o transporte aos hospitais, a fim de evitar agravamento das lesões e melhor condição de sobrevivência dos acidentados, suas principais atividades consistem na Prevenção e Combate, Incêndios, Buscas, Defesa Civil, Salvamentos e Socorros Públicos, acidentes de trabalho, dentre outros.

2.9. Circulação de Veículos de Emergência

Devido à alta importância dada a circulação dos veículos destinados a emergência, foram estabelecidas regras especiais para o deslocamento destes veículos. Essas regras foram estabelecidas no artigo 29, inciso VIII, do Código de Trânsito Brasileiro, e Resolução do Conselho Nacional de Trânsito n. 268/08, abrangendo os seguintes veículos:

- I) destinados a socorro de incêndio e salvamento (Corpo de Bombeiros) e os de salvamento difuso destinados a serviço de emergência decorrentes de acidentes ambientais - os veículos da Defesa Civil (incluído pelo artigo 1º, § 3º, da Resolução n. 268/08);
 - II) os de Polícia (em sentido **estrito**, são as viaturas de órgãos de Segurança pública, estabelecidos nos incisos do artigo 144 da Constituição Federal: Polícia Federal, Polícia Ferroviária Federal, Polícia Rodoviária Federal, Polícias Civis e Polícias Militares; em sentido **amplo**, são os veículos operacionais destinados à **proteção das cidades**, ou seja, serviço de **polícia**, o que engloba também as viaturas das Guardas Municipais, destinadas à proteção dos bens, serviços e instalações dos municípios, conforme § 8º do artigo 144 da CF);
 - III) os de fiscalização e operação de trânsito;
 - IV) as ambulâncias (independentemente de pertencerem à Administração pública ou à iniciativa privada).
- (Julyver, 2015)

Isso se deve ao fato desses veículos estarem constantemente expostos a situações onde a velocidade do deslocamento possui um alto impacto na

prestação do serviço de alta importância para a sociedade de uma forma geral, o código prevê determinadas regras e exceções que lhes são aplicáveis:

Suas prerrogativas são, basicamente, duas: a **1ª é a prioridade de trânsito** (devendo os outros condutores deterem a marcha e cederem a passagem, para sua rápida locomoção) e a **2ª é a liberdade de deslocamento e imobilização** (ou seja, podem circular, parar e estacionar na esquina ou sobre o passeio e etc). Obviamente, que tais liberdades legais não podem ser exercidas a qualquer momento e de qualquer forma, sem cuidados quanto à segurança viária, existindo duas condições essenciais, para que tais veículos se enquadrem nesta situação excepcional.

A primeira condição é que se verifique a situação de urgência, que se difere da nomenclatura “emergência” isto é, veículos de emergências são estes dos quais ora tratamos, que podem não se encontrar em serviço de urgência, definição que se encontra no artigo 1º § 2º da Resolução n. 268/08: “Entende-se por prestação de serviço de urgência os deslocamentos realizados pelos veículos de emergência, em circunstâncias que necessitem de brevidade para o atendimento, sem a qual haverá grande prejuízo à incolumidade pública”.

A segunda exigência é que, além de estar diante de uma situação de urgência, os veículos estejam devidamente identificados, para que os demais usuários da via possam reconhecer a condição especial em que se encontra; tal identificação é composta pelo sistema luminoso (iluminação vermelha intermitente) e alarme sonoro (sirene característica de tais veículos).

(Julyver, 2015).

Além disso é previsto no Código de Trânsito Brasileiro que as demais regras deverão ser seguidas pelos demais veículos ao depararem com um veículo em serviço de urgência e devidamente identificado:

- I) os outros condutores devem deixar livre a passagem pela faixa do lado esquerdo, deslocando-se para a direita e imobilizando, se necessário. Nota-se que a exigência legal é de liberação do lado esquerdo da via, não havendo previsão para que os demais condutores cometam infrações de trânsito para dar passagem ao

veículo de emergência, como avançar o sinal vermelho do semáforo ou transitar sobre calçadas ou canteiro central, entre outras condutas, embora tal ocorrência seja extremamente comum;

- II) os pedestres devem aguardar no passeio a passagem do veículo de emergência, para, somente após efetuar a travessia da via.

(Julyver, 2015).

Caso o condutor venha a desobedecer a regra descrita acima, ou seja, deixando de dar passagem aos veículos de emergência obstruindo o caminho do mesmo, estará cometendo uma infração contra o artigo 189 do CTB. O mesmo se vale para caso o mesmo tente se aproveitar do espaço deixado pelo veículo de emergência, para seguir atrás dele, nesse caso o mesmo terá infringido o artigo 190.

O código impõe regras referentes ao uso desse benefício, segundo Julyver (Julyver, 2015), “Com objetivo de se fixar a associação entre a utilização dos dispositivos de veículos de emergência, efetiva necessidade, prevê a alínea ‘c’ do inciso VII do artigo 29, que “o uso de dispositivos de alarme sonoro e de iluminação vermelha intermitente só poderá ocorrer quando da efetiva prestação de serviço de urgência”, não sendo correto, portanto, utilizar os mesmo em deslocamento cotidianos, em que não se verifique a necessidade de brevidade no atendimento (tal regra ocasionou a adoção, por alguns órgãos, de cores diferentes no sistema de iluminação de veículos de emergência, como a cor azul, a ser utilizada na circulação rotineira, o que, entretanto, não tem base legal para implantação).”.

Além disso o código considera como uma infração de trânsito, cometida exclusivamente pelos veículos de emergência por “Deixar de manter ligado, nas situações de atendimento de emergência, o sistema de iluminação vermelha intermitente dos veículos de polícia, de socorro de incêndio e salvamento, de fiscalização de trânsito e das ambulâncias, ainda que parados” (artigo 222).

Temos também a alínea ‘d’ do inciso VII do artigo 29, ainda prescreve que “a *prioridade de passagem na via e no cruzamento devere se dar com velocidade reduzida e com os devidos cuidados de segurança, obedecidas as demais normas desse código*”; isto significa que embora tenham as prerrogativas legais que lhe

garantem a prioridade de trânsito e a livre circulação, estacionamento e parada, os condutores são responsáveis por garantir a segurança por onde passam, não lhes sendo lícito colocar outros em risco, ou, pior, se envolver em ocorrências de trânsito. (Julyver, 2015).

3. METODOLOGIA

Para chegar à conclusão deste trabalho foram utilizadas metodologias para os diferentes níveis no processo de desenvolvimento do trabalho.

3.1. Critérios de seleção da metodologia de pesquisa

Para a resolução de tal problema foi escolhido o método hipotético dedutivo concebido por Karl Popper, essa metodologia foi selecionada devido ao fato de a mesma trabalhar sobre a premissa de dedução de hipótese sobre um determinado problema e a validação da eficácia do mesmo.

Segundo Silveira (Silveira, 1989):

“Ela transmite a verdade das premissas para a conclusão, ou seja, sendo verdadeiro as premissas de um raciocínio dedutivo, será necessariamente verdadeira a conclusão. Ela retransmite a falsidade da conclusão para as premissas, ou seja, se a conclusão de um raciocínio dedutivo for falsa, então uma ou mais premissas são falsa. Ela não retransmite a verdade da conclusão para as premissas, ou seja, sendo a conclusão de um raciocínio dedutivo verdadeira, poderão ser falsas uma ou mais premissas.”

A pesquisa será dividida em três etapas, sendo elas:

- ✓ Definição do produto, etapa onde o problema é definido, compreendido e mapeado.
- ✓ Conjuntura, etapa onde é definida uma teoria que possa resolver ou minimizar o impacto do problema.
- ✓ Falseamento, etapa onde é definido.

3.2. A Definição do Projeto de Pesquisa de Estudo de Caso

A pesquisa aplicada foi definida por necessidade de melhoria nos problemas relacionados ao deslocamento de veículos de emergências em vias de grandes

centros urbanos. Problemas esses que podem impactar diretamente pessoas com problemas de saúde ou que necessitam de um atendimento de emergência.

O SAMU (serviço de atendimento móvel de urgência) é um serviço de extrema importância para a saúde pública nacional, segundo Silva (Silva et al., 2012) “no período de janeiro de 2005 a dezembro de 2011 [...] o SAMU realizou 445.238 atendimentos, dos quais 295.726 geraram deslocamento das viaturas”.

As ambulâncias, por sua vez, desempenham um papel fundamental, pois são elas que transportam o pessoal e o material para o atendimento, bem como a vítima ao hospital, se necessário. No Brasil, são classificadas conforme em 6 categorias, de acordo com sua capacitação e modal de transporte (BRASIL, 2004).

Nesse processo, a maioria dos indicadores de desempenho é baseada em medições de tempo. No atendimento médico de urgência isto não podia ser diferente. Porém, segundo Moeller (Moeller, 2004), a determinação dos tempos decorridos numa ação de atendimento é motivo de dúvidas e controvérsias, dificultando a comparação entre entidades diferentes e, que as "imprecisões", em muitos casos, são executadas de forma consciente, de forma a melhorar os indicadores.

Embora esse serviço possua uma abrangência nacional alguns estudos como o de Mortari apontam algumas brechas no sistema de saúde. Segundo Mortari (Mortari, 2015) “o tempo de resposta é maior entre as 19 e 20 horas. Exceto pelos horários de 5 e 6 horas da manhã os maiores tempos estão no turno da noite”.

Essa demora no tempo de resposta pode ser correlacionada ao chamado horário de pico de congestionamento, alguns autores como SCARINGELLA evidenciam o crescimento do número de veículos na cidade como a diminuição da velocidade média ao longo dos anos.

De acordo com SCARINGELLA (SCARINGELLA, 2001):

A maior cidade brasileira tem 25% da frota nacional, o que hoje representa perto de cinco milhões de veículos. Praticamente temos um carro para cada dois habitantes. A pesquisa Origem-Destino, realizada a cada dez anos desde 1967, abrangendo a área mais fortemente urbanizada da Região Metropolitana de São Paulo que registra perto de seis milhões de veículos

identificou em sua última versão 30 milhões de deslocamentos diários, sendo 10 milhões em transporte coletivo, 10 milhões em transporte individual e os restantes 10 milhões a pé.

Alguns estudos apontam que houve um considerável crescimento na média de velocidade na metrópole paulistana nos últimos anos, segundo SCARINGELLA (SCARINGELLA, 2001) “Num período de cinco anos (entre 1992 e 1997), a média de quilômetros de congestionamento medidos pela CET no sistema viário principal da cidade passou de 40 km, na hora de pico da tarde, para 120 km.”.

Esse cenário tem se intensificado ao longo dos anos, segundo Rolnik (Rolnik, 2011). “nos horários de pico. Entre os anos 2000 e 2008, a velocidade média do trânsito nos horários de pico da manhã e da tarde na cidade de São Paulo foi de 19,30 km/h. O tempo médio gasto no trânsito pelos paulistanos para realizar todos os deslocamentos diários é de 2h42min. A cada mês, o paulistano passa dois dias e seis horas no carro ou no transporte público para se locomover. Os paulistanos perdem, em média, 27 dias por ano presos no congestionamento. [...] Em 2009, o recorde de índice de congestionamento em São Paulo - 294 km de lentidão - foi quebrado duas vezes no mesmo dia. Ao longo da última década, 118 km de vias congestionadas têm sido a média diária da cidade”

Segundo o relatório de Mobilidade no Sistema Viário Principal da CET de 2016 (CET, 2016) existe uma distribuição irregular de tráfego nas diversas vias da cidade, a tabela completa com os dados de velocidade média por via pode ser observada na seção de Tabelas.

O conceito básico inerente à internet das coisas está no uso de troca de dados em rede para facilitar processos. A identificação de gatilhos a esses processos é, nesse cenário, viável por meio de sensores diversos, que monitoram o meio de interesse e, mediante a um evento, encaminham a ocorrência para algum outro nó da rede (Atzoria et al., 2010), responsável em realizar alguma ação. Todos os dias, milhares de operações como essas ocorrem, gerando a necessidade de estruturas de processamento de ações baseadas em dados, bastante consistente em termos de regras de negócios, volume de dados e escaláveis para suportar a tendência cada vez maior de envio de informações, derivada do crescimento da

rede pelo acesso à tecnologia por novos agentes. Esses nós processadores, que acabam por converter dados em informações úteis, podem assumir a característica de um big data muitas vezes, quando centraliza um grande volume de dados para processamento e os organiza para viabilizar análises (Schadt et al., 2011).

Com base nesses estudos foi possível detectar que existe uma deficiência no Serviço de Atendimento Móvel de Urgência (SAMU), que é ocasionada pela sobrecarga das rodovias, ruas e pela má distribuição veículos na cidade de São Paulo.

3.2.1. As Questões de Estudo

As questões de pesquisa estão abaixo:

- ✓ Os autores assumem a responsabilidade técnica e a garantia do funcionamento do protótipo?
- ✓ Será possível mostrar um trajeto mais rápido, e qual o processo feito para obter o mesmo?
- ✓ Seria possível um monitoramento dos trajetos?

3.2.2. As Proposições

Com o crescimento da frota de veículos é possível notar na Figura 4 o aumento do fluxo de veículos na cidade de São Paulo, mesmo com medidas paliativas para tentar minimizar esse impacto como, por exemplo, o rodízio municipal, é possível notar que o problema não é sanado, tendo em vista que a cada dia temos mais veículos entrando em circulação, e por outro lado, poucas medidas governamentais estão sendo implantadas para tratar esse tipo de problema.

Legenda:

Coluna 1: ciclomotor, motoneta, motocicleta, triciclo e quadriciclo
Coluna 2: micro-ônibus, camioneta, caminhonete e utilitário
Coluna 3: automóvel
Coluna 4: ônibus
Coluna 5: caminhão
Coluna 6: reboque e semirreboque
Coluna 7: outros (caminhão-trator, trator de rodas, trator de esteiras, trator misto, chassi/plataforma, sidecar, motor-casa).

2018

| Abril 2018 | | | | | | | |
|----------------|-----------|-----------|------------|---------|---------|---------|-------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Capital | 1.174.392 | 1.094.222 | 6.111.071 | 47.588 | 169.004 | 90.615 | 7.574 |
| Estado | 5.658.433 | 3.477.810 | 18.581.801 | 163.962 | 873.781 | 584.957 | 124.254 |
| Total | | | | | | | |
| | | | | | | | 8.694.466 |
| | | | | | | | 29.464.998 |
| Março 2018 | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Capital | 1.169.474 | 1.088.229 | 6.094.130 | 47.502 | 168.856 | 90.580 | 7.566 |
| Estado | 5.639.561 | 3.460.472 | 18.530.954 | 163.703 | 872.574 | 582.057 | 124.218 |
| Total | | | | | | | |
| | | | | | | | 8.666.337 |
| | | | | | | | 29.373.539 |
| Fevereiro 2018 | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Capital | 1.165.774 | 1.083.208 | 6.079.593 | 47.381 | 168.422 | 90.338 | 7.559 |
| Estado | 5.624.451 | 3.447.016 | 18.488.808 | 163.502 | 871.312 | 579.470 | 124.193 |
| Total | | | | | | | |
| | | | | | | | 8.642.375 |
| | | | | | | | 29.298.752 |
| Janeiro 2018 | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Capital | 1.162.422 | 1.079.544 | 6.067.166 | 47.347 | 168.159 | 90.232 | 7.557 |
| Estado | 5.610.984 | 3.435.731 | 18.450.834 | 163.334 | 870.678 | 577.203 | 124.181 |
| Total | | | | | | | |
| | | | | | | | 8.622.427 |
| | | | | | | | 29.232.945 |

Figura 4 - Fluxo de veículos da cidade de São Paulo em 2018. Adaptado de (DETRAN SP, 2018)

A Figura 04 apresenta o número de veículos registrados legalmente na capital e estado de São Paulo. Podemos notar o número elevado de veículos, em torno de oito milhões e quatrocentos mil, e crescendo em uma média de vinte mil veículos por mês.

Com o processamento e análise de dados pretende-se melhorar o deslocamento entre vias com o intuito de diminuir consideravelmente o tempo de deslocamento em dois pontos para veículos de emergência.

3.2.3. A Unidade de Análise

O estudo concentra-se na necessidade da criação de um protótipo para identificação dos padrões que acarretam o número excessivo de veículos em determinados locais e horários.

São unidades de análise o tempo de retardamento, que são aqueles em que os veículos se encontram parados durante o percurso, pode-se classificar em dois tipos:

- ✓ Retardamento semaforico – quando o veículo fica parado em virtude de um semáforo fechado (ou vermelho).
- ✓ Congestionamento – quando o veículo se encontra parado em virtude do excesso de veículos na via, causado pela demanda dessa via, obras, colisão acidentes, etc.

- ✓ Tempo de deslocamento – Tempo do deslocamento de veículos de emergência ou não entre pontos no mapa.

3.2.4. A Lógica que Une os Dados às Proposições

Os problemas encontrados em grandes centros urbanos no deslocamento de veículos de emergência são os mais vários possíveis. Este artigo busca oferecer uma solução para um dos problemas encontrados que é a dificuldade no deslocamento destes veículos em situação de emergência pelo trânsito engarrafado de grandes centros urbanos. A proposta é auxiliar o serviço de emergência para que seja possível maximizar as chances de sobrevivência das pessoas que necessitam desse atendimento, desde o acontecimento da situação de risco até o momento do atendimento/entrada da pessoa em uma unidade de saúde e hospital especializado. Todo o trabalho é realizado por veículos de transporte e suporte à vida, com a diminuição do tempo de resposta no deslocamento do veículo de emergência, as chances de sobrevivência de um indivíduo aumentam consideravelmente.

Devido a isso a busca incessante por novas tecnologias com soluções inovadoras que auxiliem no deslocamento/posicionamento de veículos de emergência tem atingido boa parte das empresas que prestam tal serviço.

Nesse estudo, é aplicado a utilização de um algoritmo criado com finalidade de tentar minimizar ao máximo esse tempo de atendimento.

3.2.5. Os Critérios para Interpretar as Constatações

O critério fundamental para análise dos dados coletados é a pesquisa bibliográfica, incluindo o estudo de trabalhos recentemente publicados em livros, teses, artigos de revista, anais de congressos, em bibliotecas e na Internet. Assim, é ampliado o número de fontes relativas à pesquisa, possibilitando validar os dados coletados através do cruzamento das diversas fontes de evidências.

3.2.6. A Coleta de Evidências

A pesquisa bibliográfica é de suma importância para a realização da pesquisa deste artigo além disto são válidas as seguintes fontes para a coleta de evidências, dados coletados referente ao tempo de deslocamento de veículos de emergência em várias faixas de horários, para identificar em qual faixa esses veículos enfrentam mais lentidão. Esses dados serão cruzados com o percentual de lentidão para o horário da CET.

Os dados obtidos durante a coleta serão analisados da seguinte forma:

- ✓ Etapa 1 – Velocidade média de um ponto A até um ponto B.
- ✓ Etapa 2 – Velocidade média de um ponto A até um ponto B em três dias diferentes.

A validação dos resultados obtidos no protótipo será feita através da análise dos dados gerados pelos diversos cenários possíveis. O principal ponto a ser observado é que o deslocamento do ponto A até o ponto B deve ser feito em um menor tempo em relação ao valor inserido no parâmetro inicial, assim como a velocidade média de deslocamento deve ser maior que o valor inserido no parâmetro inicial do protótipo.

3.3. A Análise Qualitativa das Evidências

A análise qualitativa é válida na elaboração das deduções específicas sobre um acontecimento fundamentado na presença dos objetos em análise e não na frequência da sua ocorrência, em cada comunicação individual. (BARDIN, 1977) É a melhor opção quando você não tem uma hipótese definida e o objetivo é explorar o comportamento para definir um problema e desenvolver uma abordagem apropriada.

Neste trabalho é possível destacar as seguintes:

- ✓ Compreender e interpretar comportamentos e tendências;
- ✓ Identificar hipóteses para um problema;

3.4. Framework

Este capítulo tem como foco descrever o funcionamento do mesmo. Como citado na Seção 1.1.1 do Capítulo 1 esse trabalho tem como intuito definir um mecanismo que atue na otimização do trajeto de veículos de emergência, para tal foi elaborado uma topologia de framework. O framework foi elaborado para funcionar sobre o conceito de duas etapas lógicas sendo elas a **definição do melhor caminho** e a **otimização do trajeto** que serão descritas nas seções seguintes.

3.5. Definição do Melhor Caminho

A etapa de definição do melhor caminho tem como objetivo traçar a melhor rota de trajeto para a ambulância, na nossa topologia a chamada melhor rota é definida como o trajeto entre a origem e o destino com o menor tempo de duração. Por exemplo, caso haja N trajetos o algoritmo terá como função analisar os mesmos e selecionar aquele com o menor tempo de deslocamento. Para que isso ocorra, o funcionamento o algoritmo deve se basear nas informações captados pelos sensores instalados nas chamadas interseções, que são definidos na nossa topologia como o cruzamento entre uma ou mais vias de trajetos. Esse comportamento pode ser observado na Figura 5 onde a Av. Silvio Sciumbata possui N cruzamentos com outras vias cada cruzamento será tratado como uma intersecção de trajeto, essa interseção irá receber um sensor de movimento que será responsável por indicar quantos veículos passaram pelo trecho e a velocidade média.

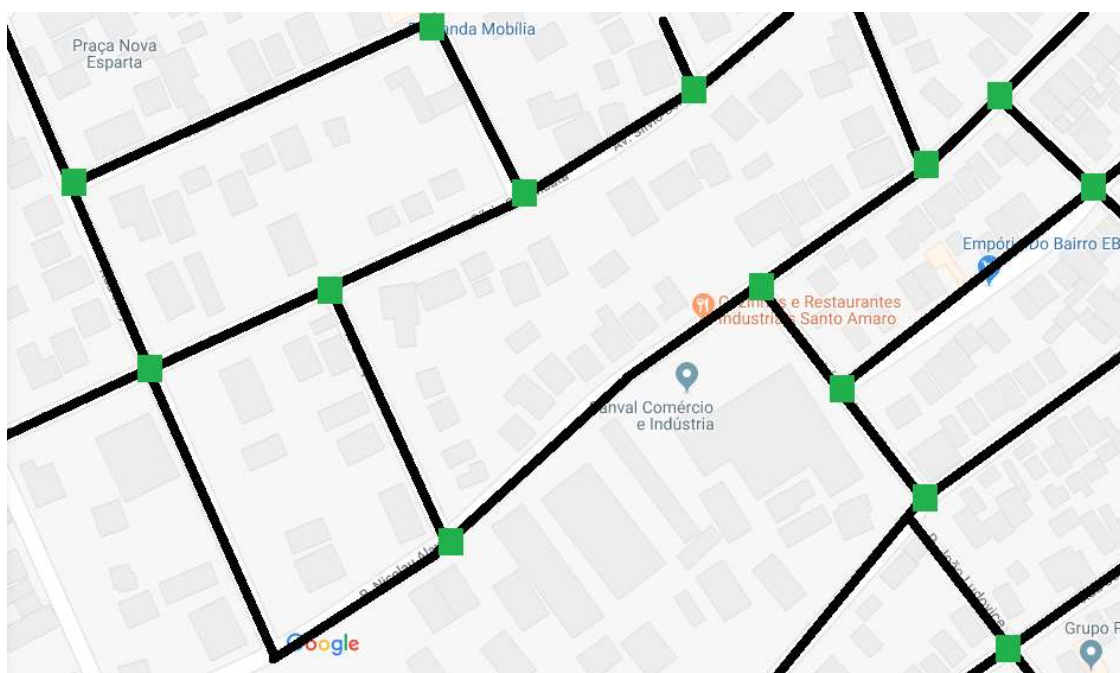


Figura 5 - Intersecções do Trajeto. Adaptado de (Google Maps, 2018)

As informações de tráfego captadas pelos sensores serão responsáveis pelo mapeamento da situação do trajeto de uma forma granular permitindo assim uma tomada de decisão mais assertiva e com uma maior possibilidade de tomada de decisões, por exemplo, caso dez veículos passem pelo cruzamento entre a Av. Silvio Scumbata e a Rua Aurélia uma velocidade de 20 km o algoritmo irá concluir que a velocidade média do trecho será de 20 km por hora.

Esse processo é demonstrado na fórmula abaixo onde VM é a velocidade média do trajeto enquanto VT é a soma das velocidades e NV o número total de veículos no trajeto.

$$VM = \frac{VT}{NV}$$

Essa abordagem embora simples se mostrou muito eficiente nos cenários analisados, porém a mesma possui um ponto negativo, quanto maior o número de veículos trafegados em determinado trajeto maior a sua assertividade logo em trechos sem nenhuma movimentação ou com uma movimentação mínima como, por exemplo, um ou dois veículos por hora a mesma poderá apresentar resultados opostos a realidade.

Com base nessas informações será gerado uma abstração do tráfego, sobre essa abstração é rodado o algoritmo responsável pela definição do melhor caminho, para exemplificar o funcionamento do algoritmo foi adicionado a Figura 6:

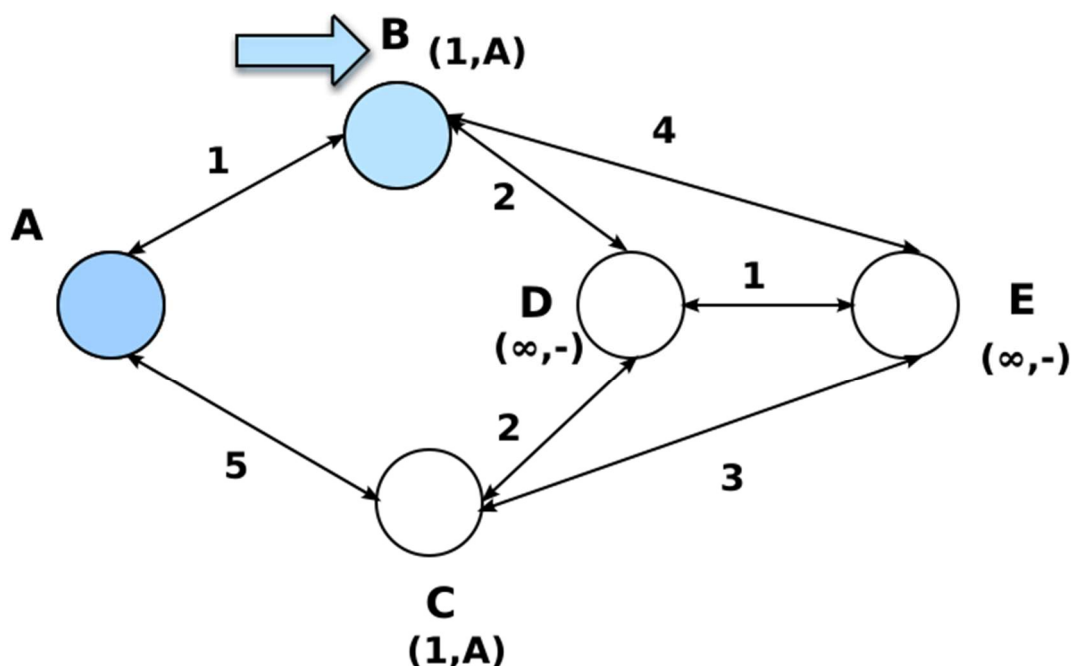


Figura 6 - Funcionamento do Algoritmo. Extraído de (Dijkstra, 2018)

Partindo do princípio de que o ponto A e o ponto E são respectivamente os pontos de partida e destino, o algoritmo irá calcular as somas do peso de todos trajetos os trajetos calculados, nesse caso o algoritmo iria concluir que o trajeto ABDE é o melhor caminho disponível pois teria um peso total de 4, seguido pelo trajeto ABD com um peso 5 e os trajetos ACE e ACDE com o peso 8.

3.6. Otimização do Trajeto

A segunda etapa é o gerenciamento do fluxo de trânsito com o intuito de aumentar a velocidade média das vias indicadas pela primeira etapa do framework, para tal o modelo irá gerenciar o comportamento dos semáforos assim represando ou liberando o tráfego de acordo com a melhor estratégia definida.

Esta etapa ocorrerá na sequência do término da primeira etapa e tem como objetivo aumentar a velocidade média do trajeto selecionado através do

direcionamento do fluxo, para isto, o mesmo trata os semáforos como pontos de controle atuando tanto no represamento dos veículos como no escoamento do tráfego. Por exemplo, caso haja um grande volume de veículos sendo direcionado para o mesmo trajeto que a ambulância o modelo poderá estender o período do “sinal vermelho” deste ponto para que haja uma redução do fluxo de veículos sendo direcionados para o trajeto da ambulância, ou atuando na liberação de trajetos através da configuração do semáforo para a liberação do fluxo em pró da otimização do trajeto percorrido pela ambulância.

Essa abordagem tem demonstrado um resultado satisfatório nos testes práticos aplicados porém devido ao fato da mesma operar de uma forma invasiva no trânsito o mesmo pode vir a se manifestar de uma forma negativa para os veículos trafegando nas vias adjacentes implicando em um trade-off de carga, logo em um cenário real haveria a necessidade de estudos direcionados para cada região e horário com o intuito de minimizar os impactos dessa abordagem.

3.7. Arquitetura Física

A arquitetura física foi desenvolvida e dividida em três camadas, sendo elas a camada de emulação do trânsito, emulação de sensores e a de tomada de decisões. Devido a necessidade de uma arquitetura robusta e escalável, o conjunto ferramental Spark foi selecionado como o motor principal deste framework em ambas as etapas devido a implementação nativa do seu módulo de processamento streaming além da sua alta performance devido a sua arquitetura que prioriza utilização da memória nos processos de manipulações de seus objetos. O mesmo atuará no processo de enriquecimento dos dados coletados pelos sensores em ambas as etapas. Ambas as camadas foram descritas a seguir.

3.7.1. Camada de Emulação do Trânsito

A camada de emulação do trânsito tem como função emular o comportamento dos veículos no cenário selecionado, para tal a mesma se baseia em três conjuntos de datasets, onde:

- O primeiro concentra as informações dos semáforos e as suas configurações (dentre essas configurações encontra-se o tempo para a mudança de status, e o status inicial de cada entre outros), esse dataset foi armazenado em uma collection no MongoDB.
- O segundo dataset concentra uma massa de veículos com a sua origem e destino, e alguns fatores que definem um comportamento randômico ao veículo como a velocidade máxima, dimensões entre outros, devido à grande quantidade de dados essa informação foi armazenada em um grande número de collections distribuídas em um banco de dados MongoDB com o intuito de otimizar as consultas do mongo.
- O terceiro dataset armazena um grafo que é a abstração da via onde cada nó do mesmo é a representação de uma intersecção entre duas ou mais ruas, enquanto o relacionamento entre esses nós é a representação do trajeto entre elas, devido a topologia deste dataset foi selecionado o banco de dados Neo4J para armazená-lo.

Com base nesses três datasets, o algoritmo irá emular a movimentação dos veículos no cenário selecionado em ciclos de execução, cada ciclo terá como resultado final um arquivo .txt com uma massa de informação emulando a captura dos dados por parte dos sensores.

Ao longo de cada ciclo o algoritmo poderá ser impactado pelo funcionamento dos demais algoritmos, por exemplo, caso determinado semáforo receba uma instrução de redução no tempo de abertura o mesmo irá apresentar uma redução no número de veículos que poderão trafegar durante o ciclo de execução nesse trecho.

A Figura 7 abaixo tem como objetivo ilustrar o funcionamento do algoritmo, nela podemos observar que o algoritmo se baseia em um input de dados de origem e destinos de veículos, sobre esses dados é executado o algoritmo responsável pela emulação do trânsito, após o término da sua execução é gerado um lote de arquivos físicos que são persistidos em uma pasta do sistema.



Figura 7 - Ilustração o Funcionamento do Algoritmo

O algoritmo segue as seguintes etapas de funcionamento:

Inicialmente todos os veículos se encontram no ponto de origem, para cada via são computados o tempo em que o semáforo se encontra fechado e aberto, através de um processo iterativo ele calcula quantos veículos poderão passar pela janela de tempo e com base nesse resultado é calculado até que ponto o veículo poderá se movimentar durante o ciclo de execução do algoritmo.

Esse comportamento pode ser descrito através da seguinte fórmula:

$$TS = TS - (TT / VV)$$

Onde a variável TS representa o tempo restante para o fechamento do semáforo, a variável TS receberá a cada veículo que conseguir passar pelo trecho será subtraído o tempo de trajeto que é resultado da seguinte operação (TT / VV) onde TT representa o tamanho o trajeto e a variável VV representa a velocidade do veículo.

Caso o valor retornado para TS seja maior ou igual a 0 isso significa que que o veículo poderá passar pelo semáforo específico em caso negativo o mesmo ficará em uma fila aguardando a execução do próximo ciclo.

A velocidade do veículo (notada como VV) é definida como:

$$VV = VTV + (VTV * FM)$$

Onde VTV é a velocidade total permitida na via em que o veículo está trafegando somado a um percentual de variância que é calculado pela multiplicação de VTV pelo FM (fator motorista) que é um valor randômico entre -0.5 e 0.5, esse percentual tem como objetivo adicionar características únicas ao veículo criando variações comportamentais o que ocorreria em um cenário real.

3.7.2. Camada de Análise dos Dados dos Sensores

A camada de análise dos dados dos sensores tem como intuito capturar os dados gerados pelos sensores através da captura do streaming dos dados gerados na Seção 3.7.1, após a captura desses dados os mesmos passaram pelo processo de Map Reduce assim tendo como resultado a velocidade média de cada trecho. Para efetuar essa tarefa foi selecionado o módulo Spark Streaming pertencente ao ecossistema Spark devido ao fato de o mesmo ter sido desenhado para lidar com um alto volume de dados, o mesmo atua sobre o princípio de micro batches de processamento.

A Figura 8 abaixo descreve o comportamento desse processo, onde são captados os arquivos de texto conforme forem sendo gerados em uma pasta específica no sistema pelo processo descrito na Seção 3.7.1 do Capítulo 3, após isso o mesmo irá processar os dados e obter como resultado a velocidade média de cada trecho essa informação será persistida em uma base de dados orientada a grafos.



Figura 8 - Processo de Captura dos Arquivos Gerados para obter a Velocidade Média

O algoritmo desenvolvido segue as seguintes etapas de funcionamento:

É iniciado um serviço no Spark que inicia um monitoramento em streaming em uma pasta específica, esta pasta é alimentada constantemente pelo processo descrito na Seção 3.7.1.

Para cada batch gerado pela captura de dados do streaming é executado uma implementação Map Reduce.

O processo efetuado pelo script pode ser descrito pela seguinte notação:

$$VM = SVV / NV$$

Onde VM é a velocidade média que é equivalente a divisão entre a soma das velocidades dos veículos que é denominada como SVV dividida pelo número de veículos no trecho denominado como NV. Uma observação é que a velocidade média do trecho estará em constante mudança já que diversos fatores poderiam impactar nas variáveis SVV e NV.

Como citado anteriormente as informações obtidas serão persistidas na base de dados em grafos, esse modelo de banco de dados foi selecionado especificamente devido a sua semelhança com a abstração proposta para a virtualização da cidade onde cada cruzamento entre vias é representado por um node enquanto o relacionamento entre eles é a representação do trecho do trajeto específico. Essa semelhança pode ser observada na Figura 9:

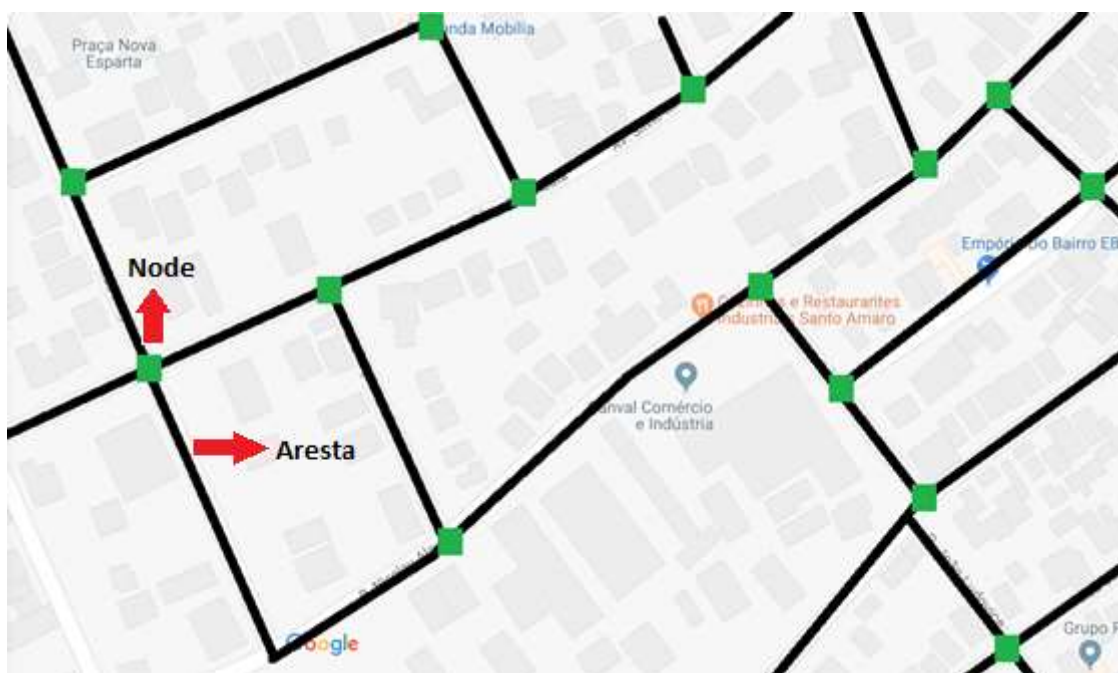


Figura 9 - Definição de node e aresta. Adaptado de (Google Maps, 2018)

3.7.3. Camada de Definição e Otimização de Trajeto

A camada de definição de otimização do trajeto executa os dois conceitos descritos no capítulo, ambos são executados sequencialmente, inicialmente ocorre a seleção do caminho mais propício descrito na subseção **Definição do melhor caminho**, onde são selecionados todos os possíveis trajetos e através de um processo iterativo é calculado o caminho com o menor tempo.

O cálculo do melhor caminho se baseia no algoritmo de Dijkstra, onde ocorre uma varredura de todas as possíveis combinações de caminhos entre uma determinada origem e destino e calcula a melhor possibilidade de trajeto de acordo com as métricas escolhidas, no nosso caso selecionamos a velocidade média dos dados coletados de sensores por via como esse indicador.

Existe a possibilidade de que a execução desse algoritmo possa resultar em um grande período de tempo de execução, ultrapassando assim a janela de tempo desejada, para evitar esse comportamento foi definido uma métrica de timeout de execução do mesmo. Além disso também foi definido uma regra de corte que age quando um total de 45 tentativas repetidas não resultam em um caminho com uma melhor otimização.

Devido ao fato de o algoritmo ser desenhado para tratar as rotas de uma forma granular onde as vias são divididas em pedaços delimitados pelos cruzamentos a topologia comporta a redefinição da rota, essa peculiaridade pode ser útil em momentos em que comportamentos imprevistos como uma tempestade impacte o modelo.

Após a seleção do caminho mais propício ocorre o processo descrito na subseção **Otimização do trajeto**, onde é validado se a velocidade média da via é inferior à velocidade máxima da mesma, caso seja detectado esse cenário o algoritmo inicia um processo com o objetivo aumentar o escoamento dessas vias, esse processo é dividido em duas etapas sendo elas o aumento do tempo dos chamados semáforos “vermelhos” localizados nas intersecções responsáveis por direcionar fluxo a via e ao mesmo tempo o aumento do tempo dos chamados semáforos “verdes” no trajeto trafegado pelos veículos de emergência, como resultado desta operação é obtido um aumento na velocidade média da via.

Para tal comportamento o algoritmo se baseia na seguinte operação:

$$PVO = VMV - VM$$

Onde *PVO* representa a velocidade que pode ser otimizada no trajeto que é o resultado da seguinte operação $VMV - VM$ onde *VMV* representa a velocidade máxima que um veículo pode atingir na via enquanto *VM* é a velocidade média da via que foi descrita no subcapítulo Camada de Análise dos Dados dos Sensores. Caso o resultado dessa operação seja superior a 0 isso significa que existe a possibilidade de executar os processos de otimização do trajeto.

A Figura 10 abaixo fornece uma visualização high level onde inicialmente o algoritmo da seleção da melhor rota é alimentado por duas origens de dados sendo elas os dados referentes aos trajetos e os chamados pesos que é o resultado de uma operação do cálculo da velocidade média da via ambos armazenados no Neo4j e os dados referentes aos setups dos semáforos (localização, vias afetadas e tempo dedicado para cada ciclo de abertura e fechamento). Esses dados são carregados em um algoritmo que como citado anteriormente tem como objetivo definir o caminho mais propício em relação a

velocidade do trajeto. Após isso o resultado é direcionado para uma segunda etapa do algoritmo que citado anteriormente tem como objetivo a otimização do trajeto através de uma customização temporária no setup dos semáforos.

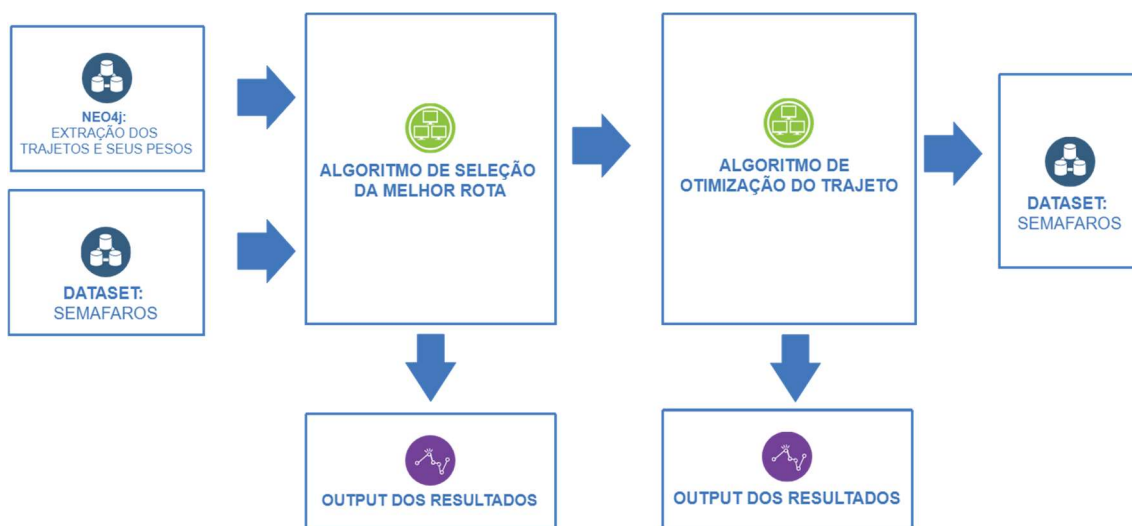


Figura 10 - Visualização do Processo para Definição da Melhor Rota

3.7.4. Topologia

O modelo terá como base duas fontes de alimentação de dados sendo elas:

- ✓ Dados gerados pelos sensores de movimento instalados nos semáforos.
- ✓ Dados gerados por sensores de aproximação configurados para detectar a aproximação de veículos de emergência.

Os dados captados pelos sensores de movimento têm como objetivo estimar a sobrecarga de fluxo em determinados pontos e com isso auxiliar o modelo a estimar o caminho mais performático.

Os dados captados pelos sensores de aproximação têm como objetivo detectar a aproximação dos veículos de emergência e a sua velocidade com isso estimando o tempo de trajeto para o próximo semáforos, além disso o mesmo terá autonomia para ignorar a instrução da central caso seja detectado algum erro na estimativa. Por exemplo, caso um veículo de emergência tenha um adiantamento de 30 segundos no tempo previsto de trajeto entre um semáforo e outro, o mesmo deverá entrar em contato com os demais e solicitar uma “prioridade emergencial

de fluxo”. Como a cada aproximação do veículo privilegiado será disparado uma notificação para a central o modelo será automaticamente ajustado.

3.7.5. Métricas de Qualidade

As métricas de qualidade descrevem especificamente um atributo de projeto ou produto e como o processo de controle da qualidade o medirá. (Bernardo, André: 2015).

Este artigo utiliza de ferramentas de Big Data e técnicas de análise de dados para obtenção de informações qualitativas sobre o funcionamento da rede de transportes de veículos de emergência, descrevemos uma metodologia para o cálculo de tempo no deslocamento desses veículos a partir de dados massivos de deslocamento de tais veículos e de dados abertos.

Após a exploração inicial dos dados, é possível definir algumas métricas para avaliar a qualidade dos dados fornecidos e a confiança dos resultados obtidos ao fim do projeto:

- Diminuição do tempo de deslocamento.
- Priorização do deslocamento por vias menos congestionadas.
- Tempo previsto/planejado de viagem de um ponto A ou ponto B.
- Intervalo médio de tempo entre os semáforos.
- Velocidade Média

Entre as métricas que merecem ser detalhadas está o índice de confiabilidade no cálculo da velocidade média extraída dos cenários analisados. Este índice depende da quantidade de carros passando pelos trechos analisados e quantidade de semáforos ali existentes, a análise fornece como resultado, um sumário com a média da diferença entre a frequência real e a planejada ao aplicar a utilização do algoritmo no cenário, tão quanto os desvios padrões dos valores.

4. ESTUDO DE CASO

O estudo de caso desenvolvido para este artigo foi necessário para comprovar as teses levantadas de forma prática. Para isto foi elaborado um protótipo com as características do mundo real, inclusive com situações de trânsito reais experimentadas em uma região da cidade de São Paulo.

Como local para os experimentos foi selecionado a região de bairro de Perdizes, essa seleção foi baseada nos seguintes critérios:

- ✓ Tráfego de veículos variável possibilitando o estudo de múltiplos cenários
- ✓ Concentração relevante de semáforos
- ✓ Múltiplas possibilidades de trajetos

Outro motivo que levou a essa decisão foi da existência de um estádio de futebol utilizado para uma grande gama de eventos variando desde partidas de futebol a shows, trazendo à tona uma variação de cenários caóticos para análise. As informações de distância, latitude e longitude do trajeto foram extraídas da API Routes & Directions do Google e as informações dos semáforos na via foram extraídas do projeto Sinal Verde da CET, esse projeto tem como objetivo listar todos os semáforos em operação na cidade de São Paulo. A Figura 11 traz ilustrativamente como estão dispostos os semáforos no projeto Sinal Verde (Sinal Verde, 2018).

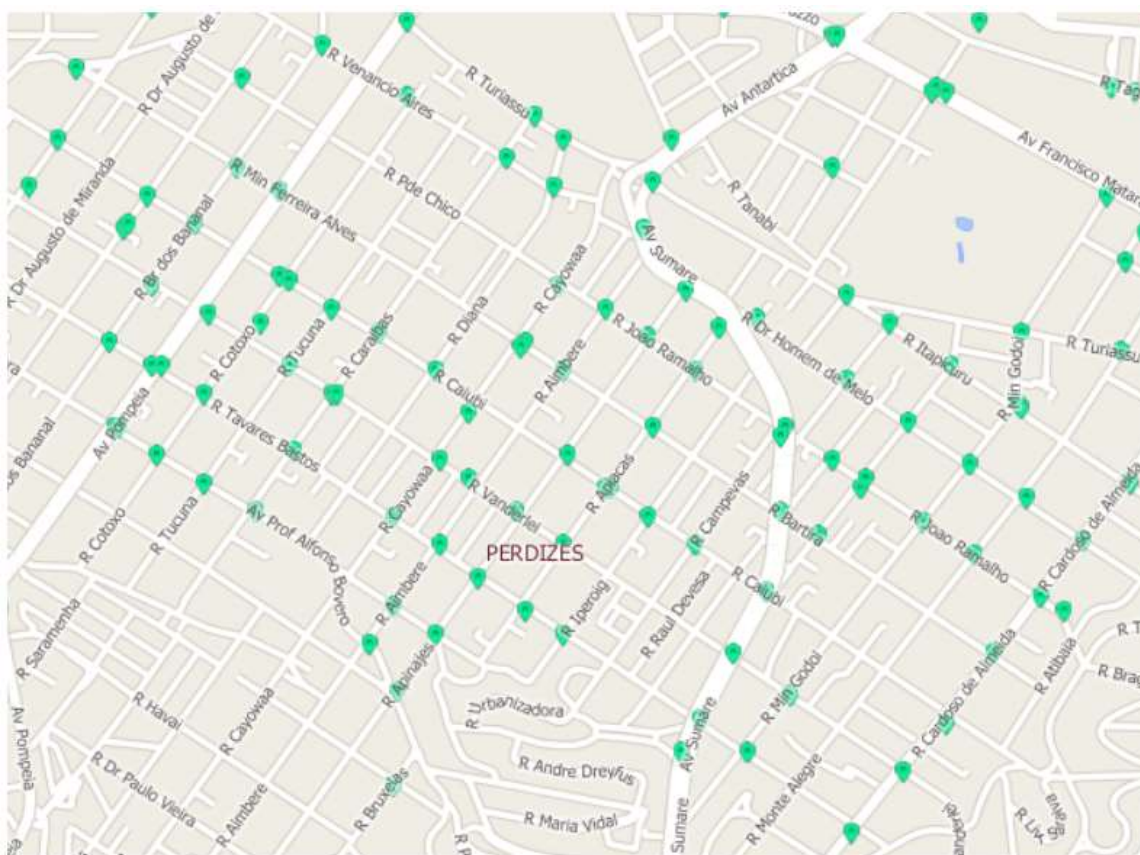


Figura 11 - Projeto Sinal Verde da CET: Semáforos em operação no trajeto. Extraído de: **(Sinal Verde, 2018)**

Para simular os sensores nos cruzamentos de todas as vias do perímetro analisado foi necessário incluir manualmente em todos os cruzamentos de vias com semáforos o que seriam estes sensores. Isso foi possível através de um algoritmo desenvolvido pela equipe especificamente para esta finalidade. A Figura 12 ilustra onde estão localizados estes sensores após a inserção através dos algoritmos.

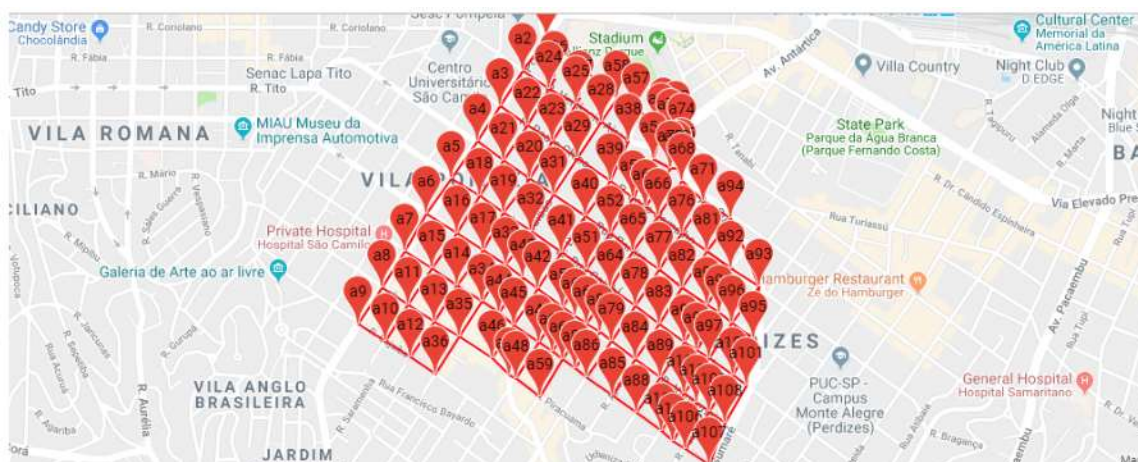


Figura 12 - Mapeamento de Interseções das vias e de seus trajetos. Adaptado de: **(Google Maps, 2018)**

Ao todo foram mapeados 367 trechos de trajetos todos interligados no mínimo a uma das 107 intersecções monitoradas. Para este protótipo quando maior a quantidade de trechos mapeados melhor, para uma maior eficácia do modelo de decisão de trajetos. Certas dificuldades foram superadas, como a capacidade de processar a grande quantidade de trechos o que fere uma grande quantidade de dados. Foi necessário atualizar o algoritmo para um melhor funcionamento com grande quantidade de dados para que fosse possível transpor essa barreira de processamento. A Figura 13 demonstra como ficaram os grafos com os trajetos levantados nessa etapa.

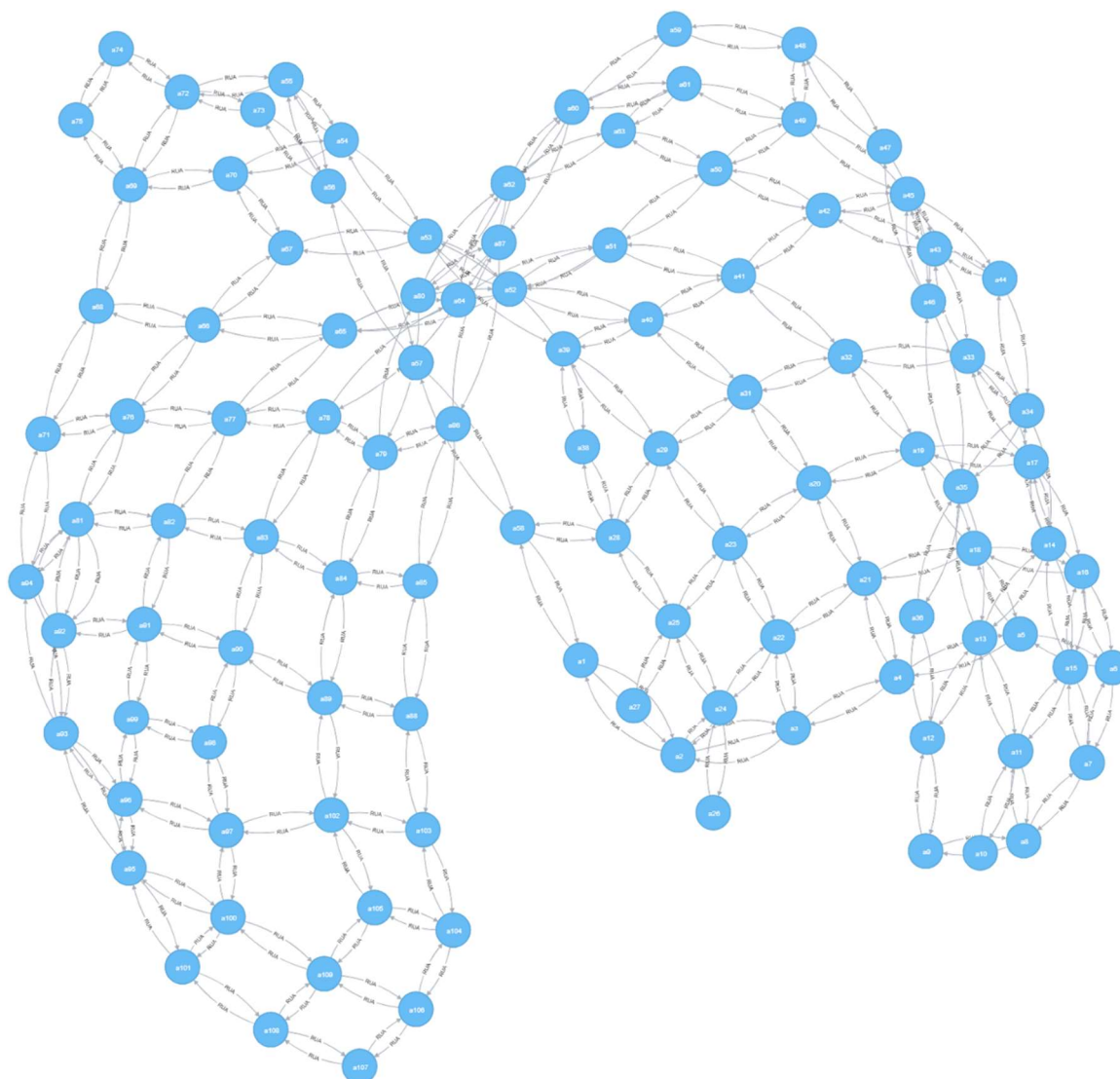


Figura 13 - Visualização do grafo referente a região estudada.

A Figura 13 traz o grafo com as conexões entre os trechos do setor que foi selecionado para a análise. O trecho selecionado para o acompanhamento foi entre as Rua Turiassú e a Rua Ministro Ferreira Alves. Esse trecho foi selecionado principalmente pela grande possibilidade de trajetos como avenidas, ruas e também pelo fato de grande impacto gerado pela movimentação de automóveis e pedestres em dias de evento no Allianz Parque.

4.1. Comparação com o Waze

O Waze, hoje um dos melhores players do mercado de navegação por GPS, é baseado no conceito de crowdsourcing social e com aproximadamente 65 milhões de usuários conectados mensalmente, transformou cada usuário em um participante na agregação de informações sobre o trânsito nas cidades e estradas (News, 2018). Ao usar o aplicativo, você concorda em dizer continuamente onde está, dado fundamental para que os servidores centrais realizem cálculos com mais precisão e ajudem a encontrar o melhor caminho, na realidade, a combinação de latitude e longitude, informada pelo próprio celular, auxilia na montagem de um mapa vivo e sofisticado. Devido a limitações na extração dos dados originados pelo Waze, não foi possível efetuar uma comparação precisa sobre o tempo de trajeto já que na própria API do Waze não foi possível obter tal informação. Devido a esta impossibilidade de obter dados da plataforma do Waze foi realizada uma análise comparativa de trajeto apenas. A tabela comparativa na Figura 14 demonstra a diferença de trajetos entre os mesmos pontos, no lado esquerdo o trajeto sugerido pelo Waze e no lado direito o trajeto sugerido pelo protótipo do algoritmo. Para este experimento o algoritmo do protótipo utilizou dados da API do Google Maps / Routes.

| Trajetos do waze | Trajetos do Algoritmo |
|---|---|
| R. Turiassu Av. Sumaré R. Dr. Homem de Melo R. Cayowaá R. João Ramalho R. Cotoxó | Rua Turiassú Rua Cayowaá Avenida Sumaré Rua Doutor Homem de Melo Rua João Ramalho Rua Aimberê Rua Bartira Rua Caiubi Rua Aimberê Rua Vanderlei Rua Cayowaá Rua João Ramalho Rua Doutor Homem de Melo Rua Cayowaá Rua Turiassú Rua Palestra Itália Avenida Pompéia Rua Cotoxó Rua Padre Chico Rua Ministro Ferreira Alves Rua Cotoxó |

Figura 14 - Tabela de Comparação do Trajeto

Na Figura 15 é demonstrado o trajeto sugerido pelo Waze para os pontos entre a Rua Cotoxó e Rua Turiassú. Apesar dos trajetos sugeridos entre o algoritmo do protótipo e o Waze apresentarem quase o mesmo tempo de deslocamento entre os pontos, é interessante perceber que o trajeto do Waze traz a passagem por menos vias do que o trajeto do algoritmo do protótipo. Na Figura 15 é possível ver o trajeto sugerido pelo algoritmo do protótipo entre os mesmos pontos, partindo da Rua Cotoxó até a Rua Turiassú.

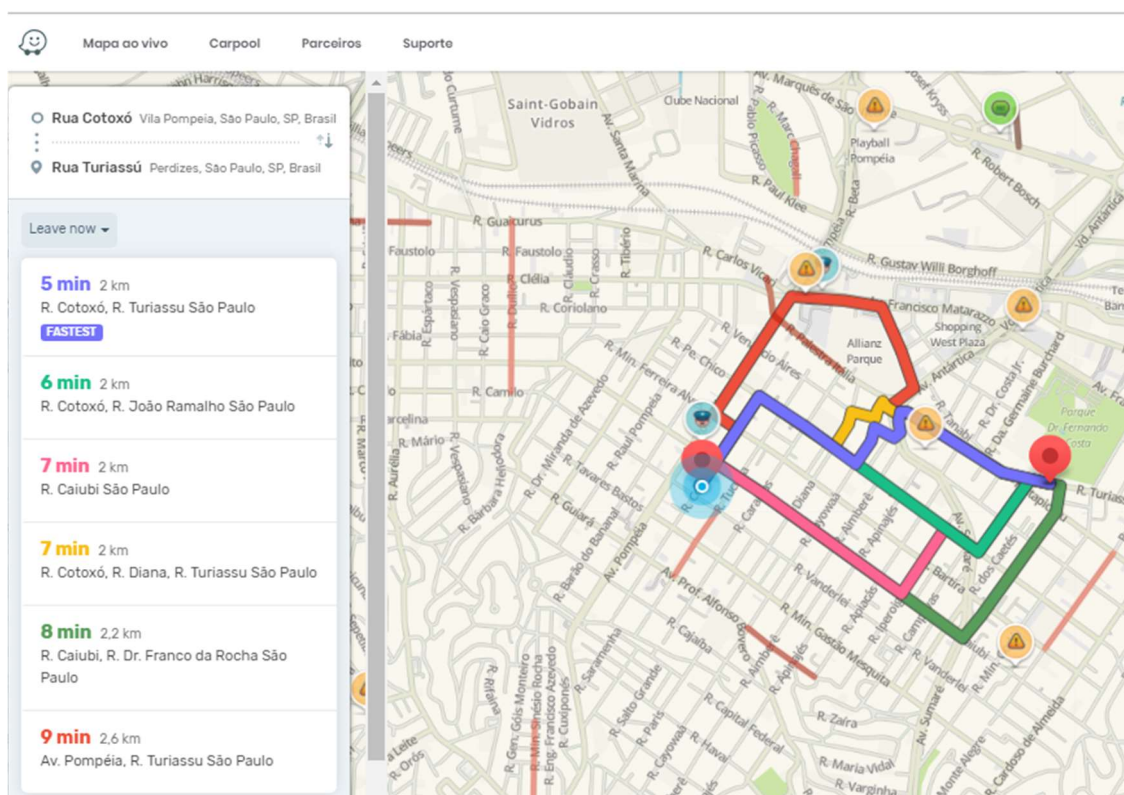


Figura 15 - Trajetos sugeridos. Adaptado de: (Waze, 2018)

4.2. Comparação com o Google Maps / Routes

O Google Maps é um outro player do mercado de navegação por GPS. É um serviço de pesquisa e visualização de mapas e imagens de satélites da terra gratuito (Mourão, 2014). Nele é possível obter informações como rotas, transporte público e etc. Foram levados em consideração três cenários para a análise comparativa. Um cenário com baixo fluxo de automóveis, um cenário com médio fluxo de automóveis e um cenário com um altíssimo fluxo de automóveis (fluxo

caótico). Abaixo são descritos mais detalhadamente cada um dos três cenários escolhidos para a análise:

Cenário 1 – Baixo Fluxo

Esse cenário foi emulado no dia 23 de Outubro de 2018, tanto a extração do tempo estimado do trajeto como a extração dos dados da API ocorreram às 10:23hrs (Horário de Brasília). O trajeto escolhido para esta análise foi entre as ruas Turiassú e Ministro Ferreira Alves. O dia analisado não teve uma interferência negativa do clima já que era um dia ensolarado. Tanto o algoritmo como o Google Maps sem a opção dos semáforos ativos apresentou o mesmo tempo de trajeto estimado, isso se deve ao fato de ambos os algoritmos terem como base a mesma fonte de alimentação. A Figura 16 traz o trajeto sugerido pelo Google Maps.

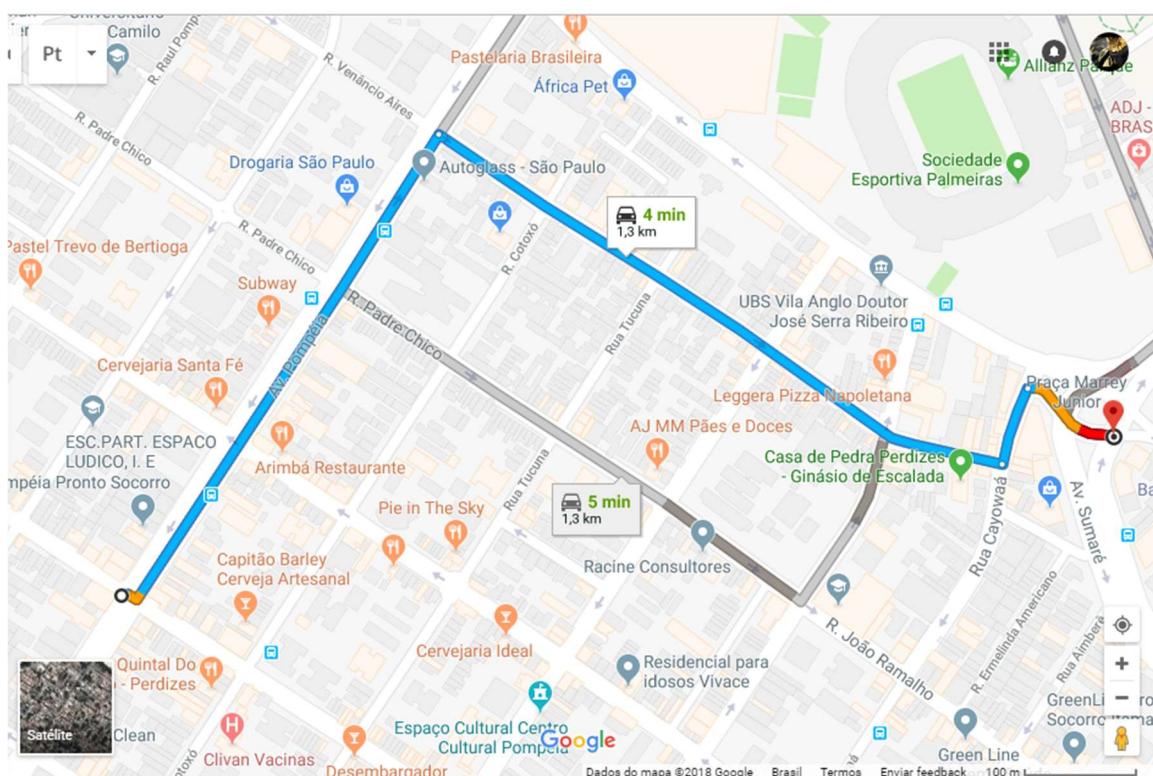


Figura 16 - Trajeto analisado no cenário 1. Extraído de: (Google Maps, 2018)

Cenário 2 – Médio Fluxo

O segundo cenário foi analisado no dia 24 de Novembro de 2018 em uma Segunda-Feira às 19:00hrs (Horário de Brasília), novamente não houve nenhum impacto negativo devido ao clima, o mesmo trajeto teve um aumento no tempo do trajeto em cerca de 25% em comparação com o primeiro cenário pelo fato do horário ser considerado de pico de movimentação de automóveis na região. A Figura 17 mostra o trajeto sugerido pelo Google Maps.

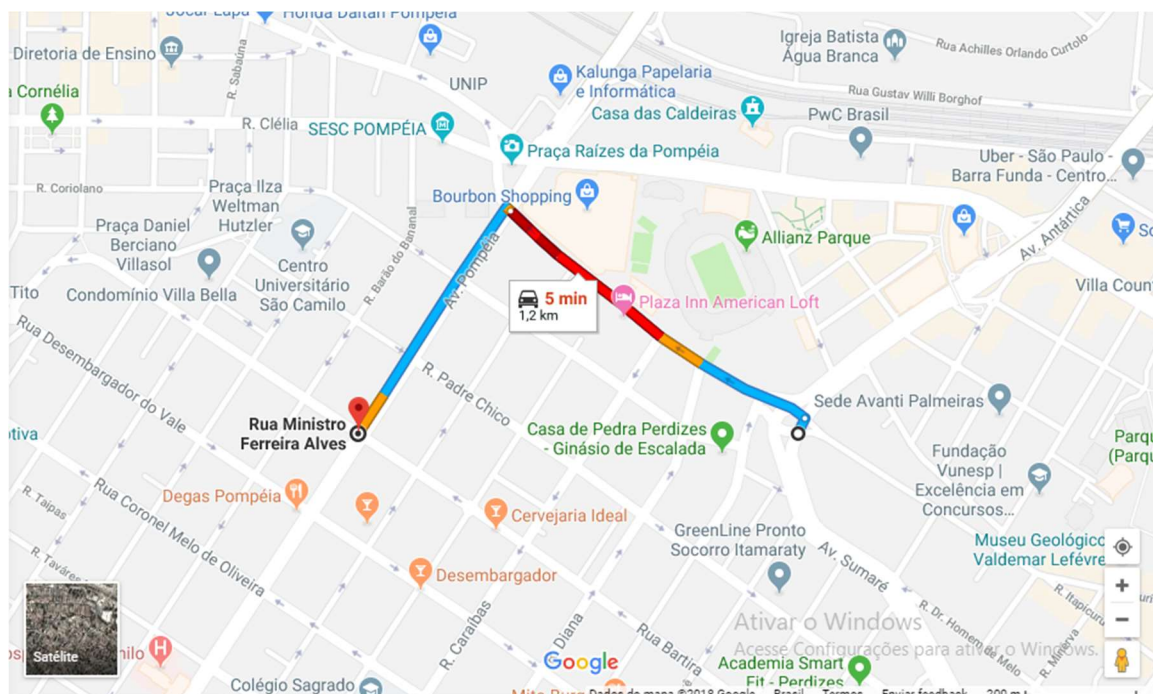


Figura 17 - Trajeto analisado no cenário 2. Extraído de (Google Maps, 2018)

Cenário 3 – Fluxo Caótico

O objetivo desse cenário foi analisar o comportamento do modelo mediante a uma situação de alto fluxo devido a fatores únicos, esse cenário foi capturado no dia 29 de Setembro de 2018 às 20:00hrs (Horário de Brasília), o motivo da escolha desse horário para a análise foi o fato de que ocorreria uma apresentação do *Tenor Andrea Bocelli* às 21:00hrs (Horário de Brasília) na arena Allianz Parque e isso teria um impacto direto no fluxo de veículos na região. Nesse cenário em questão houve um aumento médio de 125% no tempo de trajeto do mesmo

percurso selecionando no cenário 1 e não houve nenhum impacto negativo pelo clima. A Figura 18 mostra o trajeto sugerido pelo Google Maps.

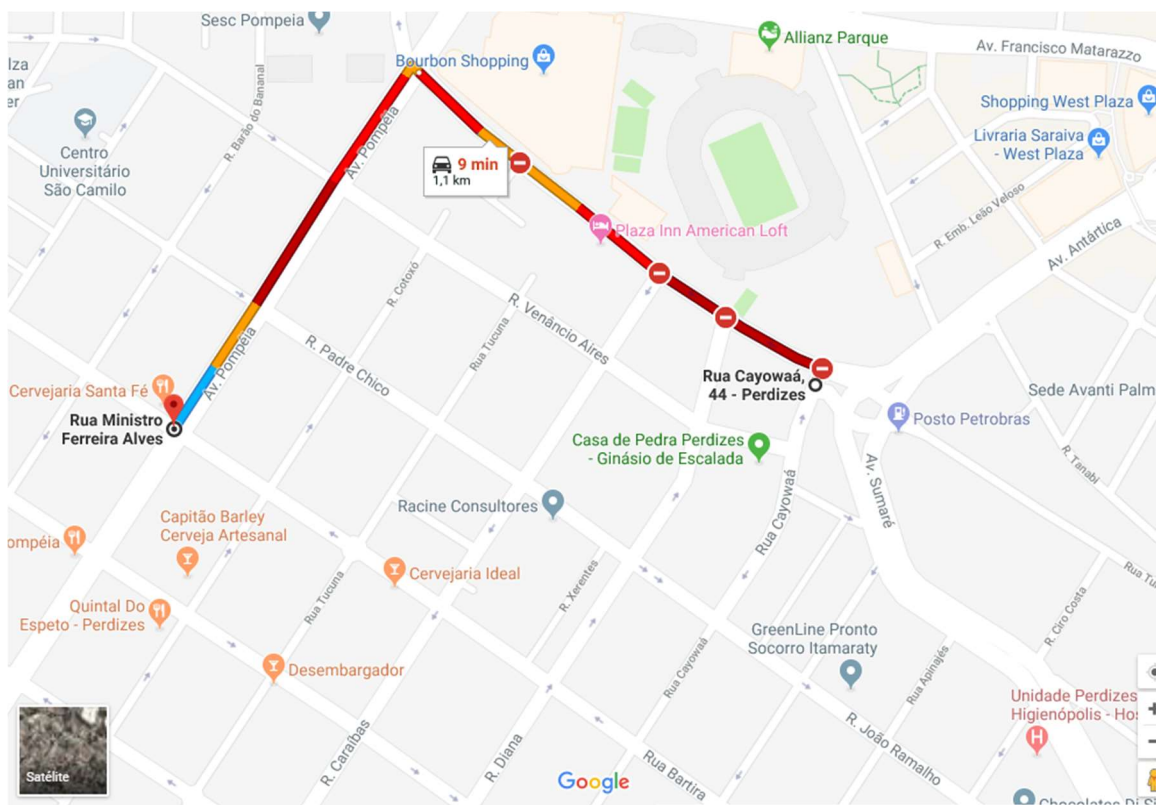


Figura 18 - Trajeto analisado no cenário 3. Extraído de (Google Maps, 2018)

4.3. Comparação com outros Algoritmos

A comparação com outros algoritmos visa a comparação com outros players de mercado. Apesar destes outros serviços tenham sido cotados para essa análise comparativa, como, por exemplo, o Here Maps, em todos foi encontrado o mesmo problema ao comparar com Waze, que devido à falta da mesma fonte de dados não foi possível obter um insight confiável referente a performance do mesmo. Devido a isso os esses exemplos foram excluídos do estudo devido à falta de contribuição das informações necessárias para andamento das análises.

5. AVALIAÇÃO DOS RESULTADOS

A seção de avaliação dos resultados apresenta os resultados obtidos e as conclusões referentes a cada cenário de estudos descritos anteriormente no Capítulo 4, em todos os casos analisados foi atingido a métrica de sucesso previamente definidas na Seção 3.7.5 do Capítulo 3. Também foi detectado a presença de certos fatores que impactaram o desempenho do algoritmo. Os resultados detalhados estão descritos a seguir:

5.1. Comparação com o Waze

O capítulo de comparação com o Waze demonstra os resultados obtidos com relação ao estudo citado na Seção 4.1 do Capítulo 4, como citado anteriormente devido à falta de uma base de dados para a comparação fidedigna do Waze, não foi possível efetuar uma comparação de performance entre os mesmos. Devido a isso o estudo não se focou na velocidade do mesmo e sim nos comportamentos de ambas as abordagens, a Tabela 1 a seguir descreve os dados do percurso obtidos pelo algoritmo. A Figura 19 demonstra o tempo de trajeto retornado pelo aplicativo Waze durante o mesmo percurso.

| Rua | Sensor | Tamanho | Velocidade Média (KM /H) | Tempo Estimado (s) |
|--------------------------|--------|---------|--------------------------|--------------------|
| Rua Turiassú | 288 | 47 | 8.1391304347826 | 14.413333333333341 |
| Rua Cayowaá | 302 | 125 | 18.75 | 20.134228187919465 |
| Avenida Sumaré | 212 | 28 | 12.2133333333333 | 6.374367622259831 |
| Rua Doutor Homem de Melo | 236 | 177 | 10.986206896552 | 43.6851063829779 |
| Rua João Ramalho | 206 | 57 | 14.4 | 11.4 |
| Rua Aimberê | 80 | 167 | 14.94 | 32.42718446601941 |
| Rua Aimberê | 275 | 117 | 13.989473684211 | 23.946140035905998 |
| Rua Bartira | 229 | 166 | 13.885714285714 | 34.17647058823585 |

| | | | | |
|-----------------------------|-----|-----|-----------------|--------------------|
| Rua Caiubí | 197 | 117 | 15.087272727273 | 22.53940455341473 |
| Rua Caiubí | 248 | 166 | 15.821052631579 | 30.77073170731699 |
| Rua Aimberê | 172 | 55 | 8.7 | 16.097560975609756 |
| Rua Vanderlei | 269 | 58 | 6.6545454545455 | 20.361702127659484 |
| Rua Cayowaá | 242 | 168 | 15.03 | 32.46376811594203 |
| Rua Cayowaá | 225 | 166 | 12.991304347826 | 36.018867924528486 |
| Rua Cayowaá | 39 | 164 | 12.991304347826 | 35.584905660377544 |
| Rua João Ramalho | 158 | 167 | 12.96 | 36.30434782608696 |
| Rua Doutor Homem de Melo | 37 | 114 | 12.825 | 24.986301369863014 |
| Rua Cayowaá | 165 | 24 | 11.828571428571 | 5.6000000000000156 |
| Rua Turiassú | 35 | 86 | 18.9 | 13.76 |
| Rua Palestra Itália | 189 | 358 | 14.648275862069 | 70.62585034013593 |
| Avenida Pompéia | 0 | 158 | 13.73023255814 | 32.82125603864646 |
| Avenida Pompéia | 1 | 161 | 19.862068965517 | 24.703703703703958 |
| Avenida Pompéia | 81 | 161 | 15.869724770642 | 29.76929601357113 |
| Avenida Pompéia | 57 | 124 | 14.283870967742 | 24.961038961038867 |
| Rua Cotoxó | 58 | 165 | 16.155555555556 | 30.067491563553883 |
| Rua Padre Chico | 66 | 163 | 16.8 | 28.76470588235294 |
| Rua Ministro Ferreira Alves | 55 | 163 | 10.917757009346 | 40.419466975665706 |
| Rua Cotoxó | 24 | 129 | 13.142857142857 | 27.73720136518795 |

Tabela 1 – Dados do percurso obtidos pelo algoritmo.

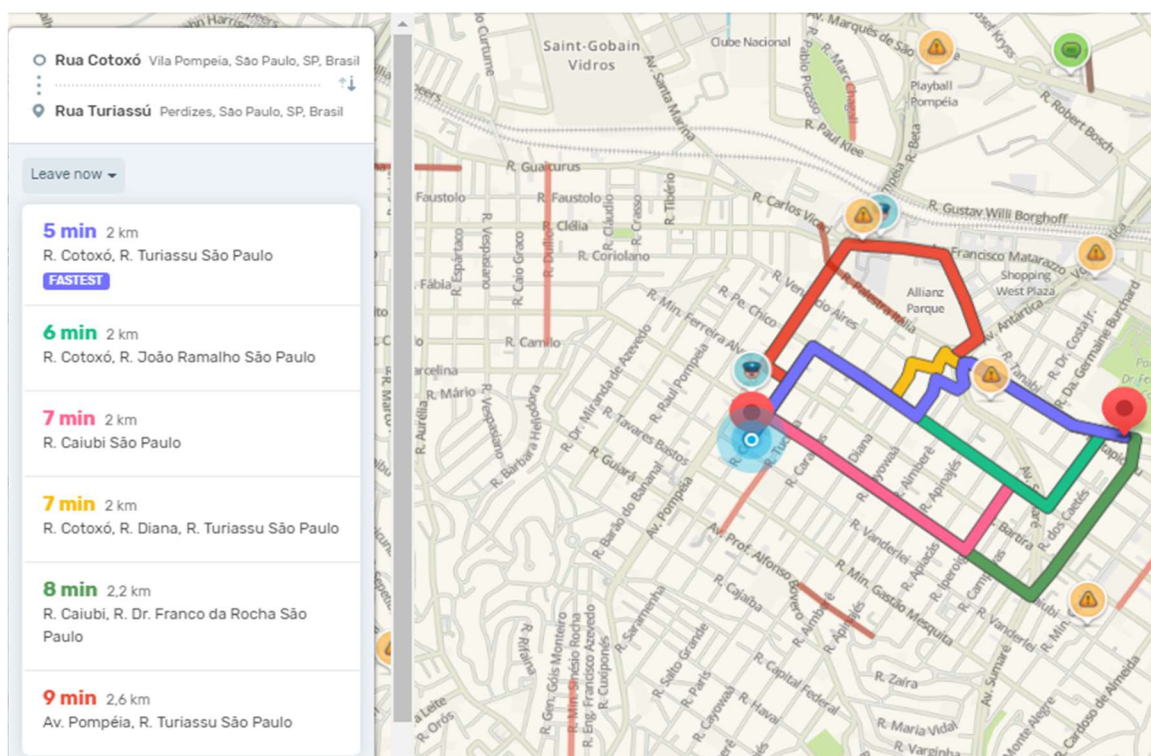


Figura 19 - Trajetos sugeridos. Adaptado de: (Waze, 2018)

Como resultado pode-se observar uma discrepância na quantidade de vias indicadas pelo aplicativo Waze e pelo algoritmo do protótipo, em uma proporção de 18 curvas indicadas algoritmo para 3 curvas indicadas pelo Waze, essa diferença pode ser resultado da não consideração da perda de velocidade sofrida pelo veículo em uma curva por conta do algoritmo do protótipo. Após esse teste foi efetuado uma nova calibragem no algoritmo, porem devido as limitações na comparação com o algoritmo do Waze, o mesmo não foi utilizado para novos testes.

5.2. Comparação com o Google Maps / Routes

Os seguintes resultados apresentados se referem aos estudos de caso apresentados na Seção 4.2 do Capítulo 4, os mesmos se encontram na ordem respectiva.

5.2.1. Cenário 1 – Baixo Fluxo

O cenário 1 visa obter dados de análise em um dia de baixo fluxo de trânsito na região, durante essa análise houve um empenho não só em definir a performance do algoritmo com relação ao serviço do Google Maps Routes mas também de definir qual o impacto da camada de direcionamento do trânsito.

A Tabela 2 que se encontra abaixo ilustra a velocidade média do trajeto selecionado pelo algoritmo, sem a utilização do algoritmo de gerenciamento do trânsito.

| Caminho | Velocidade (km / h) | Tempo de Trajeto (s) |
|---|---------------------|----------------------|
| Rua Turiassú | 8.13913043478 | 14.4133333333 |
| Rua Cayowaá | 12.391416309 | 9.00483091787 |
| Rua Cayowaá | 11.8285714286 | 5.6 |
| 1210 (<i>Nome não extraído via api</i>) | 19.3090909091 | 18.5428571429 |
| Rua Turiassú | 18.9 | 13.76 |
| Rua Palestra Itália | 14.6482758621 | 70.6258503401 |
| Avenida Pompéia | 13.7302325581 | 32.8212560386 |
| Avenida Pompéia | 19.8620689655 | 24.7037037037 |
| Avenida Pompéia | 20.3586206897 | 24.3419689119 |
| Rua Ministro Ferreira Alves | 18.7875 | 27.8190954774 |

Tabela 2 – Velocidade média do trajeto selecionado pelo algoritmo.

Após a ativação do algoritmo que faz os semáforos darem a preferência aos veículos de emergência em trânsito o resultado obtido foi um aumento de cerca de 10% de economia de tempo no trajeto total além de uma redução média de 12 segundos por trajeto valvulado por um ou mais semáforos sendo equivalente a uma redução média de 70%. As informações detalhadas do trajeto sugerido, tempo, velocidades médias para este cenário podem ser avaliadas na Tabela 3.

| Caminho | Velocidade (km / h) | Tempo de Trajeto (s) | Tempo de Trajeto Economizado | Tempo de Trajeto (s) Total |
|----------------------------------|---------------------|----------------------|------------------------------|----------------------------|
| Rua Turiassú | 8.13913043478 | 14.4133333333 | 0 | 14.4133333333 |
| Rua Cayowaá | 12.391416309 | 9.00483091787 | 0 | 9.00483091787 |
| Rua Cayowaá | 11.8285714286 | 5.6 | 0 | 5.6 |
| 1210 (Nome não extraído via api) | 19.3090909091 | 18.5428571429 | 0 | 18.5428571429 |
| Rua Turiassú | 18.9 | 13.76 | 0 | 13.76 |
| Rua Palestra Itália | 14.6482758621 | 70.6258503401 | 0 | 70.6258503401 |
| Avenida Pompéia | 13.7302325581 | 32.8212560386 | 9.0 | 23.8212560386 |
| Avenida Pompéia | 19.8620689655 | 24.7037037037 | 9.6 | 15.1037037037 |
| Avenida Pompéia | 20.3586206897 | 24.3419689119 | 17.4 | 6.9419689119 |
| Rua Ministro Ferreira Alves | 18.7875 | 27.8190954774 | 16.8 | 11.0190954774 |

Tabela 3 – Informações detalhadas do trajeto sugerido, tempo e velocidade média para o Cenário 1 – Baixo Fluxo

Comparando esses dois resultados obtidos foi possível constatar que houve um ganho considerável de tempo após a ativação do algoritmo de gerenciamento do trânsito que embora tenha atingido apenas 4 das 10 interseções do trajeto já que as demais não possuíam um semáforo a ser gerenciado houve um ganho considerável de cerca de 10% no resultado final.

5.2.2. Cenário 2 – Médio Fluxo

O cenário 2 há um foco em avaliar a performance do algoritmo em um cenário de médio fluxo de tráfego na região estudada, como resultado da experimentação neste cenário foi possível obter um aumento de performance de 25% no tempo total de trajeto. Se comparado ao cenário 1 houve um aumento de cerca de 150% na performance do algoritmo, essa diferença indica uma certa superioridade de desempenho do mesmo com relação ao aumento de veículos no trajeto. As

informações detalhadas do trajeto sugerido, tempo, velocidade média para este cenário pode ser avaliada na Tabela 4.

| Caminho | Velocidade (km / h) | Tempo de Trajeto (s) | Tempo de Trajeto Economizado | Tempo de Trajeto (s) Total |
|---|---------------------|----------------------|------------------------------|----------------------------|
| Rua Turiassú | 5.94156521739 | 17.7329396325 | 0 | 17.7329396325 |
| Rua Cayowaá | 12.391416309 | 9.00483091787 | 0 | 9.00483091787 |
| Rua Cayowaá | 12.391416309 | 9.00483091787 | 0 | 9.00483091787 |
| Rua Cayowaá | 11.8285714286 | 5.6 | 0 | 5.6 |
| 1210 (<i>Nome não extraído via api</i>) | 19.3090909091 | 18.5428571429 | 0 | 18.5428571429 |
| Rua Turiassú | 18.9 | 13.76 | 0 | 13.76 |
| Rua Palestra Itália | 14.6482758621 | 70.6258503401 | 0 | 70.6258503401 |
| Avenida Pompéia | 13.7302325581 | 32.8212560386 | 9 | 23.8212560386 |
| Avenida Pompéia | 19.8620689655 | 24.7037037037 | 9.6 | 15.1037037037 |
| Avenida Pompéia | 20.3586206897 | 24.3419689119 | 17 | 24.3419689119 |
| Rua Ministro Ferreira Alves | 18.7875 | 27.8190954774 | 16.8 | 27.8190954774 |

Tabela 4 - Informações detalhadas do trajeto sugerido, tempo e velocidade média para o Cenário 2 – Médio Fluxo

Na Figura 20 é possível observar um *heatmapper* realizado na área analisada. *Heatmapper* são mapas de calor com representações visuais de cores que mostram exatamente onde as pessoas estão interagindo fisicamente ou em um site, por exemplo. Com essa informação, você pode inserir conteúdo estratégico nesses locais que atraem mais pessoas ou visitantes e, assim, ajudar a aumentar sua taxa de conversão para um exemplo de site. Ajuda a compreender melhor a distribuição da informação de acordo com os pontos que o seu público mais interage. Os mapas de calor são basicamente representações visuais de dados, permitem gravar o que as pessoas fazem com o mouse ou onde clicam mais,

quantificar essas informações e exibi-las de forma visualmente atraente. (Marketing, 2018)

Na Figura 20 este recurso foi utilizado para demonstrar a distribuição dos veículos no momento da análise, a escala utilizada seguiu a seguinte escala de cores, verde para regiões com baixa concentração de veículos e vermelhos para demonstrar as regiões com maior concentração de veículos.

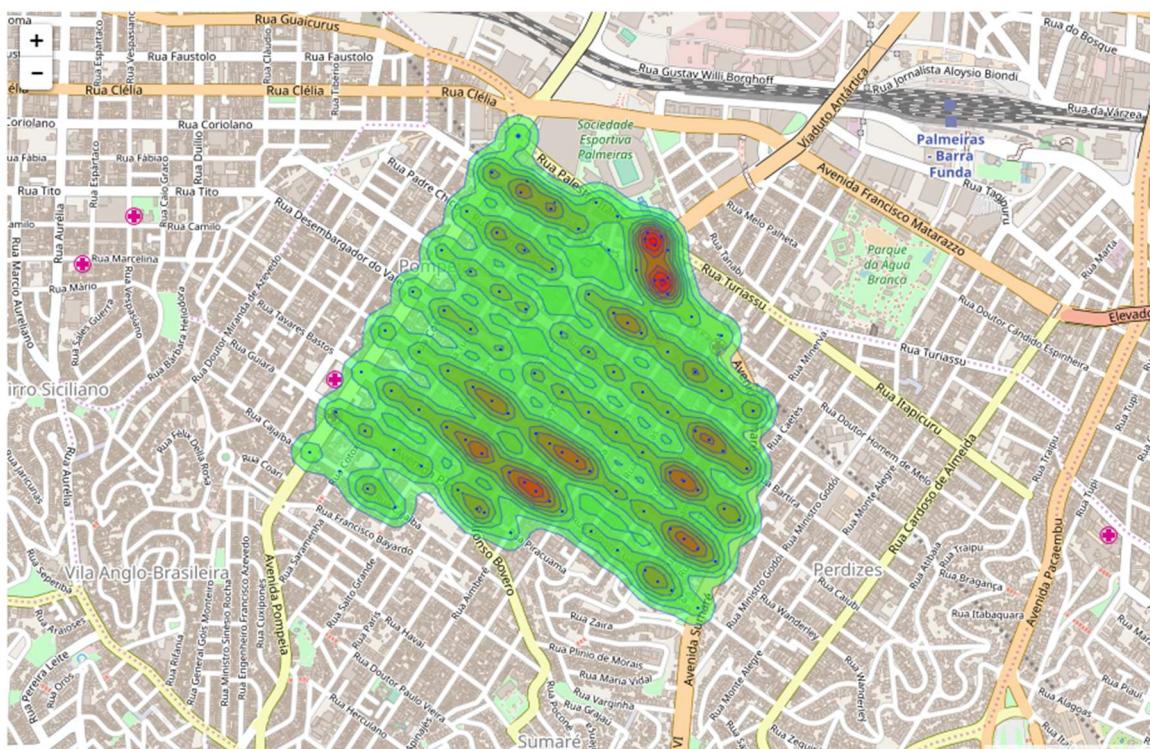


Figura 20 - Concentração de veículos por região análise do cenário 2. Adaptado de (Heatmapper, 2018)

Algo que foi observado foi que nesse estudo o algoritmo teve uma maior performance nos pontos com uma maior concentração de veículos, isso se deve ao fato dos mesmos terem se concentrado em pontos com uma maior incidência de semáforos.

5.2.3. Cenário 3 – Fluxo Caótico

O cenário 3 foi focado em analisar um cenário com um alto fluxo de tráfego de veículos, para tal foi necessário selecionar um cenário que exigisse ao extremo da aplicação.

Como resultado foi obtido um aumento de cerca de 16% no tempo total do trajeto, se comparado com os resultados obtidos entre o cenário 1 e o cenário 2 analisados, é possível observar que embora exista um aumento de cerca de 60% em relação ao resultado do cenário 1, houve um declínio em relação a performance do cenário 2 de cerca de 36%.

Isso indica que embora o algoritmo tenha obtido resultados satisfatórios em todas as emulações a existência de fatores externos podem vir a impactar negativamente no resultado do tempo de trajeto, embora que em nenhum dos cenários tenha sido obtido um resultado inferior a não implementação do mesmo.

O resultado obtido pode ser analisado detalhadamente na Tabela 5 que se encontra abaixo:

| Caminho | Velocidade (km / h) | Tempo de Trajeto (s) | Tempo de Trajeto Economizado | Tempo de Trajeto (s) Total |
|----------------------------------|---------------------|----------------------|------------------------------|----------------------------|
| Rua Turiassú | 0.964728077918 | 37.0668300744 | 12.6 | 24.4668300744 |
| Rua Cayowaá | 1.02626752452 | 31.126604598 | 12 | 19.126604598 |
| Rua Cayowaá | 4.02900510222 | 11.3251988749 | 0 | 11.3251988749 |
| 1210 (Nome não extraído via api) | 19.3435318078 | 18.5150221665 | 10.8 | 7.7150221665 |
| Rua Turiassú | 1.37505325203 | 62.2304896684 | 33.4 | 22.8304896684 |
| Rua Palestra Itália | 5.27236535319 | 145.260023533 | 48.0 | 97.260023533 |
| Avenida Pompéia | 1.52172643979 | 111.05630234 | 35.0 | 76.05630234 |
| Avenida Pompéia | 8.27513635461 | 49.1110150304 | 17.4 | 31.7110150304 |
| Rua Ministro Ferreira Alves | 0.605742303227 | 148.083252633 | 64.8 | 83.283252633 |

Tabela 5 - Informações detalhadas do trajeto sugerido, tempo e velocidade média para o Cenário 3 – Fluxo Caótico

Na Figura 21 é mostrado o *Heatmapper* para a área selecionada para este cenário de análise. A escala da Figura seguiu a seguinte escala de cores, verde para regiões com baixa concentração de veículos e vermelho para demonstrar as regiões com uma maior concentração de veículos.

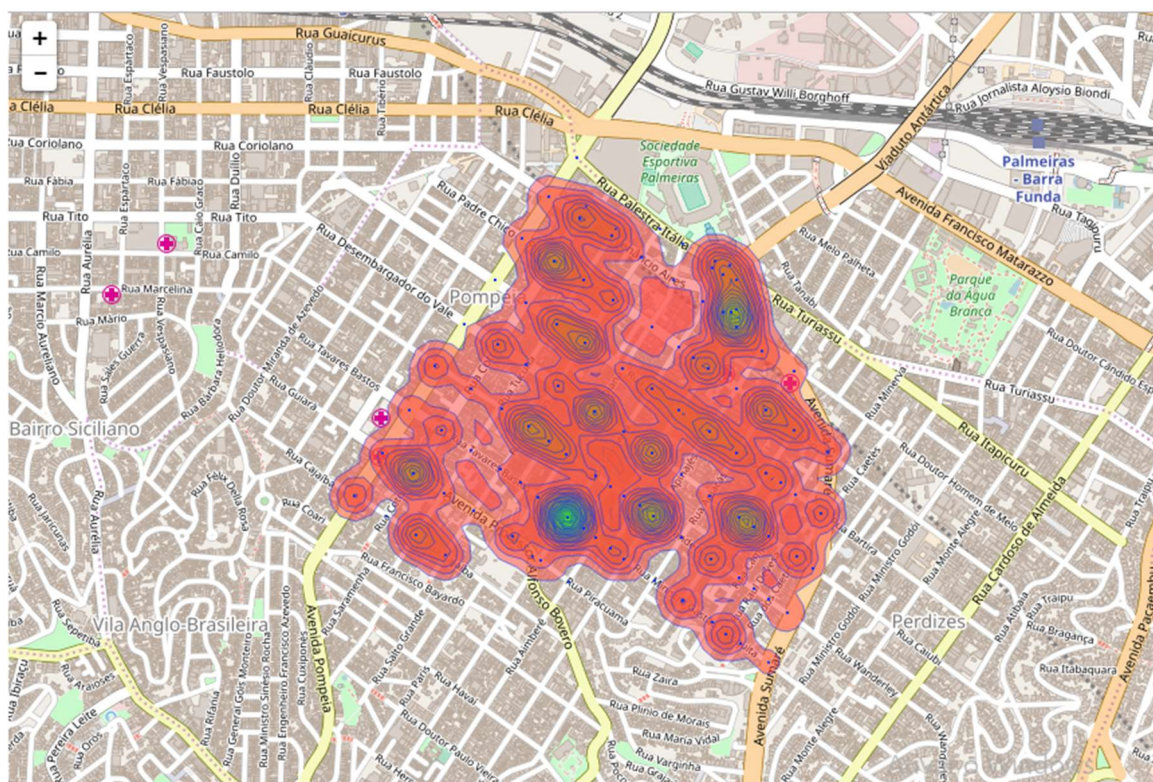


Figura 21 - Concentração de veículos por região análise do cenário 3. Adaptado de (Heatmapper, 2018)

Pode-se observar uma alta concentração de veículos, demonstrados com a cor vermelha. Isso indica que o fluxo de automóveis está bem acima do normal, isso se deve não só ao show ocorrido na arena Allianz Parque como também a grande concentração de casas noturnas na região.

6. CONCLUSÃO

O artigo desenvolveu um estudo de caso aplicado a vida real com o intuito de comprovar que com o uso de tecnologias de Big Data é possível resolver grandes problemas urbanos da atualidade. O problema principal levantado neste artigo foi a demora no deslocamento de veículos de emergência por grandes centros urbanos, devido principalmente a falta do uso de tecnologias, que já se encontram disponíveis, para resolver tais problemas que consequentemente levará ao salvamento de inúmeras vidas, já que o principal requisito em situações de emergência médica é o atendimento especializado no menor tempo possível.

O desenvolvimento deste artigo se deu com base na necessidade de agilizar o deslocamento de veículos de emergência até os pontos de atendimentos, e com isso salvar mais vidas. Para a comprovação das hipóteses levantadas foi criado um framework baseado nas hipóteses teóricas para que fosse possível testar com dados levantados do mundo real. Para os testes no ambiente virtual foram usados dados do mundo real de bairros da cidade de São Paulo em três cenários diferentes e com isso foi possível obter os seguintes resultados.

Foram levantados dados da região oeste de São Paulo, região esta que possui hospitais, casas noturnas e um estádio onde são realizados jogos e eventos diversos sendo possível levantar cenários diferentes. Conforme a Seção 5.2 do Capítulo 5 que traz os cenários levantados no estudo, com o cenário 1 de baixo fluxo foi possível obter uma diminuição de até 10% do tempo de deslocamento quando o algoritmo que gerencia os semáforos foi ativado. No cenário 2 de médio fluxo de automóveis foi possível obter uma diminuição de até 25% do tempo de deslocamento. Enquanto no cenário 3 fluxo caótico, um dia em que há eventos na região trazendo uma quantidade elevada de automóveis o tempo de deslocamento foi diminuído em até 16% em relação a veículos comuns que se utilizam do algoritmo de gerenciamento dos semáforos.

É possível perceber o aumento da performance do algoritmo do cenário 1 para o cenário 2. Isso se deve ao aumento do número de automóveis nas vias, fazendo com que o algoritmo que gerencia os semáforos pudesse atuar de forma a melhorar o escoamento na via onde o veículo de emergência estava transitando.

Para o cenário 1, apesar de contar com a mesma estratégia dos outros cenários, como existiam poucos veículos na via estes transitavam em velocidade semelhante aos veículos de emergência. Também foi possível perceber que no cenário 3 houve um decréscimo de performance em relação ao cenário 1 apesar de haver um aumento no número de veículos em circulação. Isto se deve ao fato de ter havido evento nos arredores durante o levantamento dos dados, isso faz com que fatores externos influenciem bastante no desempenho do algoritmo, como número elevado de pedestres circulando ou fechamento de vias.

Os resultados obtidos em três cenários diferentes permitiram levar a conclusão que o algoritmo projetado para a solução do problema levantado na Seção 1.1.1 do Capítulo 1 deste artigo atendeu de forma satisfatória o principal requisito que era a diminuição do tempo de deslocamento de veículos de emergência através do uso de tecnologias de Big Data.

6.1. Limitações

No processo de análise das informações obtidas e de comparações com algoritmos concorrentes, demonstrado na Seção 4.2 do Capítulo 4 deste artigo foram observadas algumas limitações referentes a arquitetura e ao algoritmo do protótipo em si. A primeira limitação quanto a arquitetura foi o uso excessivo de memória do banco de dados Neo4J, que foi utilizado neste projeto. Este é um comportamento conhecido de bancos de dados de grafos, mas que deve ser observado conforme a sua escala for aumentando durante a implementação. Outro ponto que foi observado durante os testes comparativos foi o comportamento diferente dos algoritmos do Google Routes e do algoritmo do projeto quanto a sugestão de rotas mais rápidas. O algoritmo do Google Routes durante a sugestão calcula sua rota levando em consideração curvas, penalizando rotas com muitas curvas fazendo com que o veículo ande por um tempo maior por retas sem desacelerar o automóvel. O algoritmo implementado neste projeto não leva em consideração as curvas, apesar de se sair mais rápido quando ativado a estratégia de semáforos do algoritmo. Esse comportamento pode ser levado em consideração em melhorias futuras do algoritmo, elevando sua performance na

sugestão de rotas e consequentemente fazendo com que os veículos de emergência de locomovam ainda mais rapidamente entre os pontos.

6.2. Trabalhos Futuros

O estudo de caso foi baseado em cenário emulado com dados reais. Seguindo um planejamento prévio o próximo passo a ser executado é a implementação do modelo apresentado. O primeiro ponto a ser observado para esta implementação é conseguir a reprodução do mesmo design do framework em um cenário real. Essa reprodução pode necessitar de ajustes o que demandaria alterações físicas no framework e novos testes em ambientes simulados para a observância de anomalias advindas destas alterações.

A implementação do modelo deste artigo também demanda hardware. Este hardware deve ser calculado com base na necessidade que o protótipo de implementação irá precisar. Este é um ponto muito importante pois não necessariamente o mesmo hardware utilizado para os testes deste artigo será o suficiente para uma operação em grande escala.

E por fim, uma parte importante desta implementação é a junção com sistemas já existentes de controle de tráfego, bem como com sistemas de controle de semáforos que já existem ou que necessariamente deverão ser implantados para que este protótipo possa ser executado. Este artigo não entrou em tais detalhes por não ser parte do escopo inicial do trabalho, mas que para a implementação em cenário real, todos os itens citados nesta seção devem ser abordados com bastante atenção para que o protótipo consiga ser executado com sucesso em grande escala em um grande centro urbano para que possa efetivamente ser de grande ajuda na resolução do problema de demora do atendimento dos veículos de emergência em geral.

REFERÊNCIAS BIBLIOGRÁFICAS

Apache Spark. **Apache Spark**. 2018. Disponível em: < <https://spark.apache.org/>> Acesso em 31 out. 2018.

Araújo, Julyver. **Circulação de Veículos de Emergência**. 2015. Disponível em: <<http://www.ctbdigital.com.br/artigo-comentarista/440>> Acesso em: 31 out. 2018.

Atzoria, Luigi; IERAB, Antonio; Morabito, Giacomo. **The Internet of Things: A survey. Computer Networks**. Elsevier B.V: Catania, 2010.

BARDIN, Laurence. **Análise de conteúdo**. EDIÇÕES 70: Lisboa, 1977.

Bernardo, André. **Fundamentos em Gerenciamento de Projetos**. 2015. Disponível em:<<https://amauroboliveira.files.wordpress.com/2015/11/fundamentos-em-gerenciamento-de-projetos.pdf>> Acesso em 24 out. 2018.

BRASIL, Ministério da Saúde do. **Política Nacional de atenção às urgências**. Editora do Ministério da Saúde: Brasília, 2006.

Commons, Wikimedia. **Dijkstra algorithm example 2.svg**, 2008. Disponível em:< https://commons.wikimedia.org/wiki/File:Dijkstra_algorithm_example_2.svg> Acesso em: 02 nov. 2018.

CET. **Pesquisa de Monitoração da Mobilidade**. 2016. Disponível em: < <http://www.cetsp.com.br/media/574933/2016volumevelocidade.pdf> > Acesso em: 09 dez. 2018.

Conte, Nelson. **O problema do caixeiro viajante, teoria e aplicações**. 2002 Disponível em:< <https://lume.ufrgs.br/handle/10183/118198>> Acesso em: 13 out. 2018.

Da república, Presidência. **LEI Nº 9.503, DE 23 DE SETEMBRO DE 1997**. 1997. Disponível em: < http://www.planalto.gov.br/ccivil_03/LEIS/L9503.htm> Acesso em: 31 out. 2018.

DB-Engines. **DB-Engines Ranking of Graph DBMS**. 2018. Disponível em: <<https://db-engines.com/en/ranking/graph+dbms>> Acesso em: 21 out. 2018.

de Carvalho, Bruno Miguel Pacheco Saraiva. **Algoritmo de Dijkstra**. 2008. Disponível em: <<https://student.dei.uc.pt/~brunomig/cp/Artigo.pdf>> Acesso em: 14 out. 2018.

DETRAN SP. **Frota de Veículos em SP – por tipo de veículo**. 2018. Disponível em:<<https://www.detran.sp.gov.br/wps/wcm/connect/portaldetran/detran/detran/e-statisticastrnsito/sa-frotaveiculos/d28760f7-8f21-429f-b039-0547c8c46ed1>> Acesso em: 01 ago. 2017.

Dos Anjos, Marina; Almeida, Morgana; Neto, Oscar. **Trânsito é o maior culpado na demora de atendimento de serviços de emergência. 2010.** Disponível em: <<http://www.metodista.br/ronline/noticias/cidades/2010/10/trânsito-e-o-maior-culpado-na-demora-de-atendimento-de-servicos-de-emergencia>>. Acesso em: 9 dez. 2017.

ELLIOT, P. **An approach to integrated rescue. 2000.** Disponível em: <http://www.abepro.org.br/biblioteca/enegep2005_enegep0902_1716> Acesso em: 24 Jun. 2018.

Galdino, Natanael. **Big Data: Ferramentas e Aplicabilidade.** Disponível em: <<https://docplayer.com.br/52038854-Big-data-ferramentas-e-aplicabilidade.html>> Acesso em: 20 out. 2018.

Google Maps, **Google Maps. 2018.** Disponível em: <<https://www.google.com.br/maps/>> Acesso em: 02 out. 2018.

Heatmapper, **Heatmapper. 2018.** Disponível em: <<http://heatmapper.ca/geocoordinate/>> Acesso em: 02 out. 2018.

Marketing, Analítico. **O que é mapa de calor (heat map) e por que você deve aplicar em seu site. 2018.** Disponível em: < <https://marketinganalitico.com.br/o-que-e-mapa-de-calor-heat-map/>> Acesso em 06 nov. 2018.

MOELLER, Bruce J. **Obstacles to Measuring Emergency Medical Services Performance.** EMS Management Journal: [s.l.], 2004.

MongoDB. **What is MongoDB?. 2018.** Disponível em: <<https://www.mongodb.com/what-is-mongodb>> Acesso em: 29 out. 2018.

Mortari, Rosane. **Tempo de resposta de um serviço de atendimento móvel de urgência. 2015.** Disponível em: <<https://www.lume.ufrgs.br/bitstream/handle/10183/129481/000976890.pdf?sequence=1>> Acesso em: 13 jun. 2018.

Mourão, Átilla. **Google Maps é um serviço de pesquisa e visualização de mapas. 2014** Disponível em: < <https://prezi.com/fkhxwksbtuzs/google-maps-e-um-servico-de-pesquisa-e-visualizacao-de-mapas/> > Acesso em: 06 dez. 2018.

Neo4j. **Top Ten Reasons for Choosing Neo4j. 2018.** Disponível em: < <https://neo4j.com/top-ten-reasons/>> Acesso em: 28 out. 2018.

Penchikala, Siri. **Big Data com Apache Spark - Parte 1: Introdução. 2015.** Disponível em: <<https://www.infoq.com/br/articles/apache-spark-introduction>> Acesso em: 21 out. 2018.

Python. **General Python FAQ. 2018.** Disponível em: < <https://docs.python.org/2/faq/general.html> > Acesso em: 05 dez. 2018.

News, Revista. **Saiba mais sobre o Waze, app de navegação GPS. 2018.** Disponível em: < <http://revistanews.com.br/2018/03/15/saiba-mais-sobre-o-waze-app-de-navegacao-por-gps/>> Acesso em 06 dez. 2018.

Rolnik, Raquel; Klintowitz Danielle. **Mobilidade na cidade de São Paulo. 2011.** Disponível em: < http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40142011000100007&lng=pt&tlng=pt> Acesso em: 15 out. 2018.

Santos, Bruno; Silva, Lucas; Celes, Clayson; Neto, João Borges; Peres, Bruna; Vieira, Marcos Augusto; Vieira, Luiz Filipe. **Internet das coisas: da teoria à prática. 2016.** Disponível em:<<http://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>> Acesso em: 14 out. 2018.

Scaringella, Roberto Salvador. **A crise da mobilidade urbana em São Paulo. 2001.** Disponível em: <http://www.scielo.br/scielo.php?pid=S0102-88392001000100007&script=sci_arttext&tlng=pt> Acesso em: 15 out. 2018.

SeeBot, **SeeBot. 2017.** Disponível em: < <https://seebot.com.br> > Acesso em: 27 nov. 2017.

Silva, Mauro **Hospital Nossa Senhor da Conceição Perguntas Frequentes. 2018.** Disponível em: <<http://www.hnsc.org.br/perguntas-frequentes/90>> Acesso em: 31 out. 2018.

Schadt, Eric E.; Linderman, Michael D.; Sorenson, Joe; Lee, Lawrence, Lee; Nolan, Garry P. **Computational solutions to large-scale data management and analysis. Nature Reviews Genetics. 2011.** Disponível em: < <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3124937/> > Acesso em: 06 nov. 2018.

Shen, John; Mikko H. Lipasti. **Modern Processor Design: Fundamentals of Superscalar Processors.** Waveland Press: Long Grove, 2013.

Silva, Naldiana Cerqueira; Nogueira, Lidya Tolstenko. **AVALIAÇÃO DE INDICADORES OPERACIONAIS DE UM SERVIÇO DE ATENDIMENTO MÓVEL DE URGÊNCIA. 2012.** Disponível em: < <https://revistas.ufpr.br/cogitare/article/viewFile/29287/19037> > Acesso em: 09 dez. 2018.

SILVEIRA, Fernando Lang da. **A filosofia da ciência de Karl Popper: o racionalismo crítico.** Caderno catarinense de ensino de física: Florianópolis, 1989.

Silveira, Marcio; Marcolin, Carla Bonato; De Freitas, Henrique Mello. **ANÁLISE DA INTERAÇÃO DO WAZE NAS CONDIÇÕES DO TRÂNSITO NA CIDADE DE SÃO PAULO. 2015.** Disponível em: < https://www.researchgate.net/profile/Henrique_Freitas2/publication/311794139_A

nalise_da_Interacao_do_Wase_nas_Condicoes_do_Transito_na_Cidade_de_Sa
o_Paulo/links/585ad00d08ae329d61f14749/Analise-da-Interacao-do-Wase-nas-
Condicoes-do-Transito-na-Cidade-de-Sao-Paulo.pdf > Acesso em: 06 dez. 2018.

Sinal Verde. **Projeto Sinal Verde. 2018.** Disponível em:
<<http://cetsp1.cetsp.com.br/sinalverde/osSemaforosAgora.aspx>> Acesso em 21
out. 2018.

Stackoverflow. **Developer Survey Results. 2017.** Disponível em:
<<https://insights.stackoverflow.com/survey/2017#technology>> Acesso em: 21 out.
2018.

Tani, Valter Zanela. **SAD baseado em caminhos mínimos e
georreferenciamento: uma ferramenta de apoio ao serviço de regulação
médica. 2003.** Disponível em: <
<https://repositorio.ufsc.br/handle/123456789/86285> > Acesso em: 12 set. 2018.

Waze. **Waze. 2018.** Disponível em: < <https://www.waze.com/pt-BR>> Acesso em:
15 out. 2018.

APÊNDICE A – MANUAL DO USUÁRIO

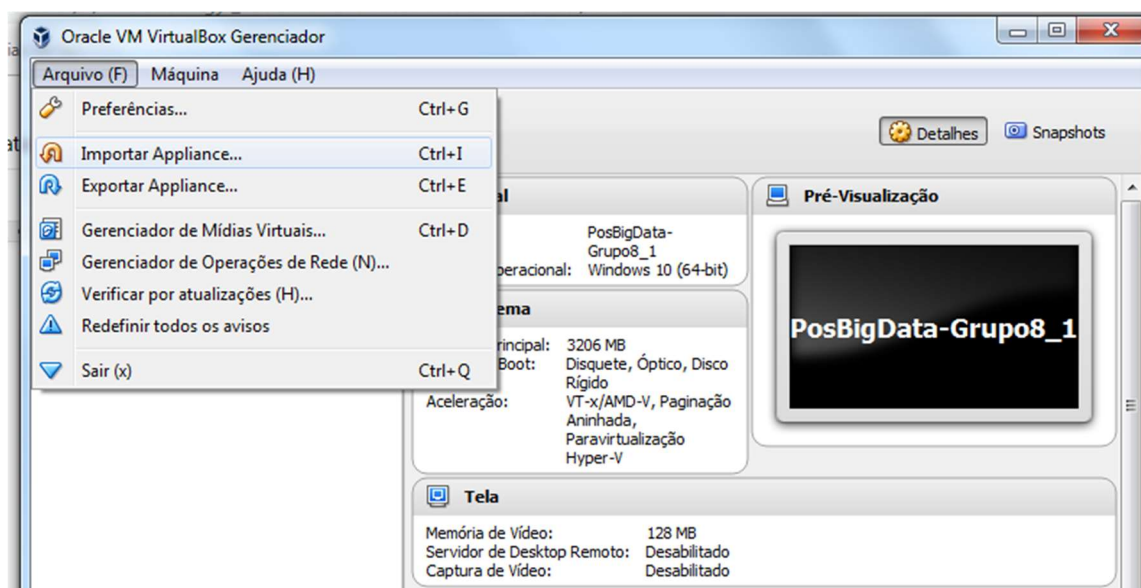
Essa seção é dedicada a explicar o funcionamento prático do sistema a fim de permitir a sua reprodução por terceiros. Para tal foi disponibilizado uma exportação da virtualização utilizada contendo o conjunto de ferramentas utilizado totalmente configurado. Para facilitar o entendimento este manual foi dividido em subseções que vão desde a importação da máquina virtual até a execução e monitoramento dos resultados dos algoritmos.

A1. Importando a Máquina virtual

A máquina virtual foi criada no Oracle VM VirtualBox, onde foi gerada uma imagem do sistema operacional Windows 10 de arquitetura 64 bit's, a máquina virtual foi entregue junto com esta monografia. Essa ferramenta foi escolhida devido a sua simplicidade de configuração e a sua estabilidade já que a mesma é mantida. Todo o processo descrito neste manual foi executado na versão 5.1.28 do Oracle VM VirtualBox, é aconselhado a utilização desta versão ou de superiores no momento da replicação do experimento.

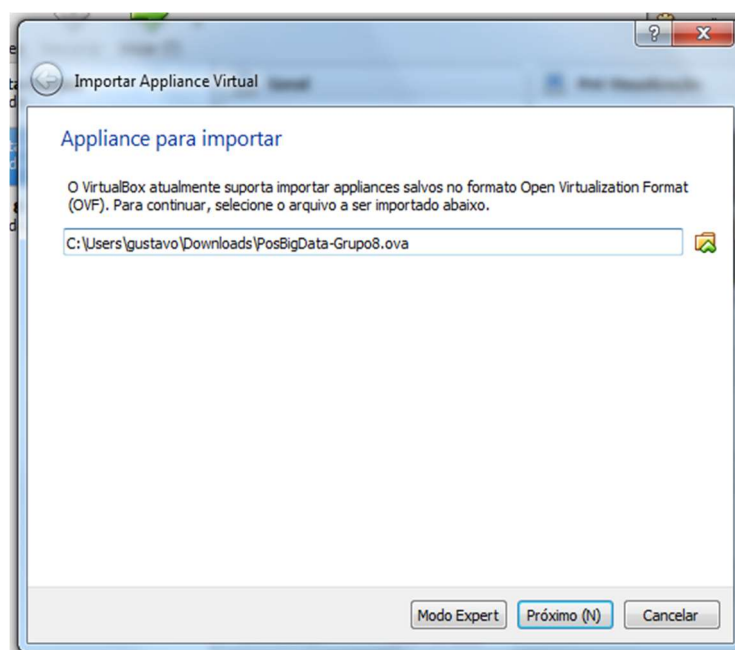
No experimento foi utilizado uma máquina com um processador Intel(R) Core (TM) i7-4500U, a máquina utilizada possuía 16 GB de memória RAM, embora essa virtualização possa ser implementada em máquinas com um menor poder computacional é aconselhado a utilização de máquinas com uma arquitetura próxima a descrita, isso se deve principalmente ao alto consumo de memória por parte do Neo4J. Todo o experimento foi conduzido em uma máquina virtual com conexão à internet, a conexão à internet é recomendável para a execução de todos os passos deste manual. As recomendações desta seção devem ser seguidas de modo que se possa garantir a execução de todo o procedimento em ambiente semelhante, evitando problemas por fatores de hardware.

O primeiro passo é a importação do Appliance, para isso será necessário iniciar o Virtualbox, clicar no item de menu **Arquivo (F)** e depois na opção **Importar Appliance ...**, é possível também efetuar esse procedimento através do atalho de teclado **Ctrl+I**, esse procedimento pode ser visto na Figura A1 abaixo:



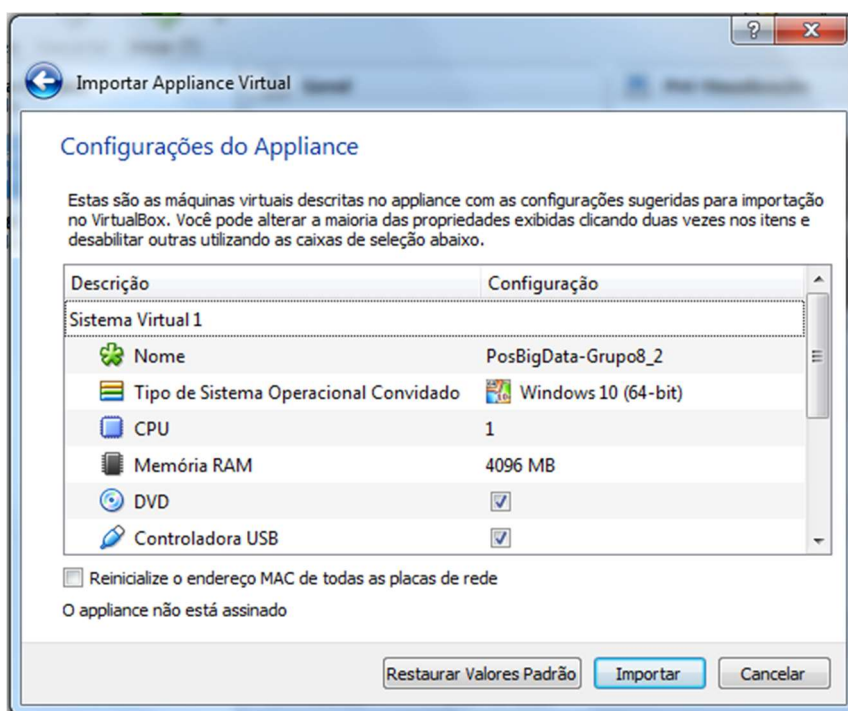
A 1 - Importação do Appliance

Após esse procedimento será exibido uma janela solicitando o caminho do arquivo a ser importado, selecione o arquivo **PosBigData-Grupo8.ova** (Este arquivo foi entregue à instituição juntamente com a dissertação) e clique no botão **Próximo (N)** como demonstrado na Figura A2.



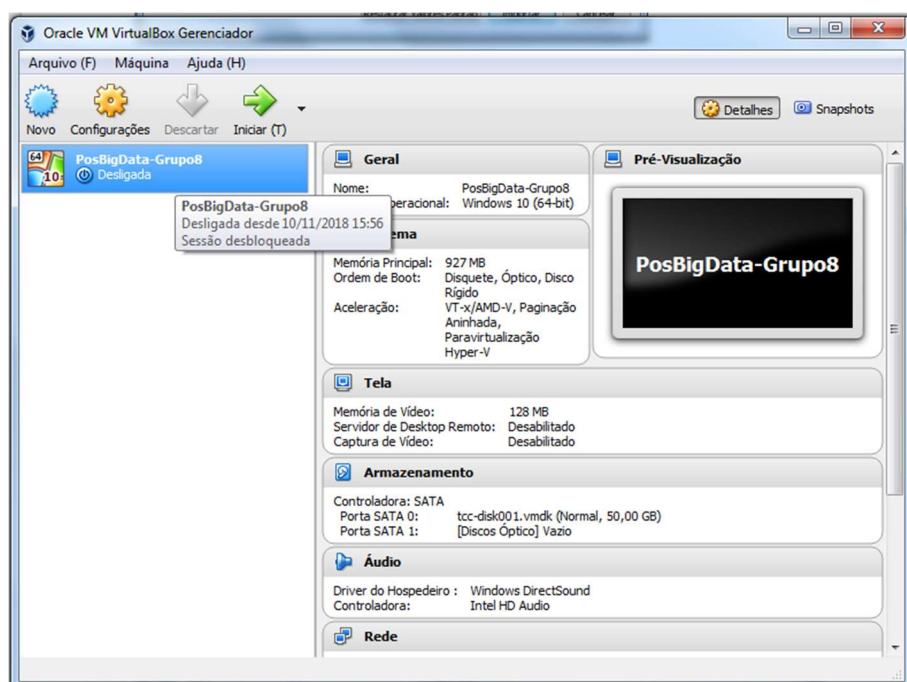
A 2 – Caminho do arquivo que deve ser importado

Após isso será exibido a nova janela com as especificações técnicas da virtualização, nessa etapa não será necessária nenhuma modificação nos parâmetros. Clique no botão Importar e aguarde o carregamento da imagem como demonstrado na Figura A3.



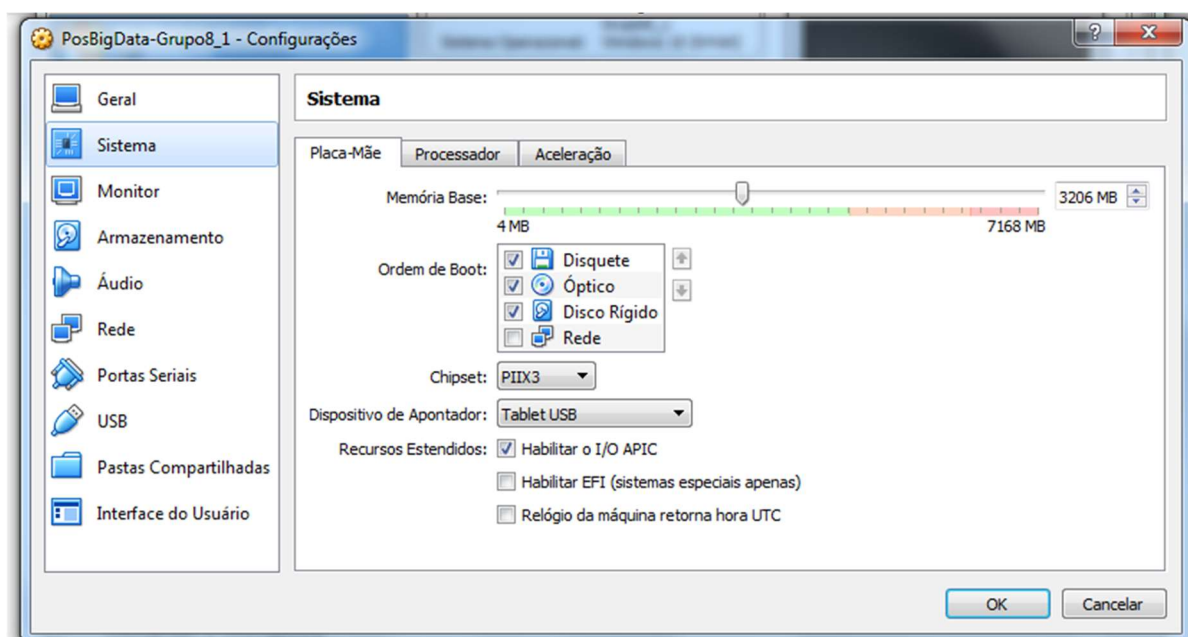
A 3 – Especificações Técnicas da Virtualização

Após esse processo a máquina virtual ficará visível no canto esquerdo da tela inicial como demonstrado na Figura A4, selecione a máquina virtual e clique na opção de **Iniciar (T)**, esse processo irá iniciar a execução da virtualização do sistema operacional.



A 4 – Tela inicial da VirtualBox Gerenciador

Caso sejam apresentados problemas ligados ao uso excessivo da memória no momento da inicialização execute o seguinte procedimento, selecione a imagem **PosBigData-Grupo8** clicando com o mouse sobre a mesma, após isso clique no botão superior de **Configurações** será exibido uma nova janela. Nessa janela clique no item Sistema localizado no menu lateral e depois na aba **Placa-Mãe**, após isso será possível reduzir o valor em **Memória Base**, como demonstrado na Figura A5 abaixo.



A 5 – Configuração do Sistema

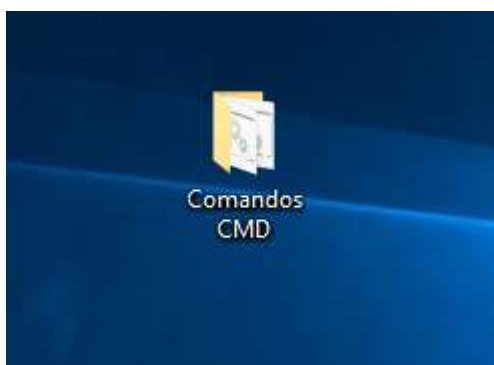
Devido ao fato do ferramental Neo4J utilizar grandes reservas de memória que já são parcialmente comprometidas pela emulação de um segundo SO, não é aconselhado a execução desse ferramental em uma máquina hospedeira com uma configuração inferior a 8 GB de memória.

A2. Introdução a arquitetura

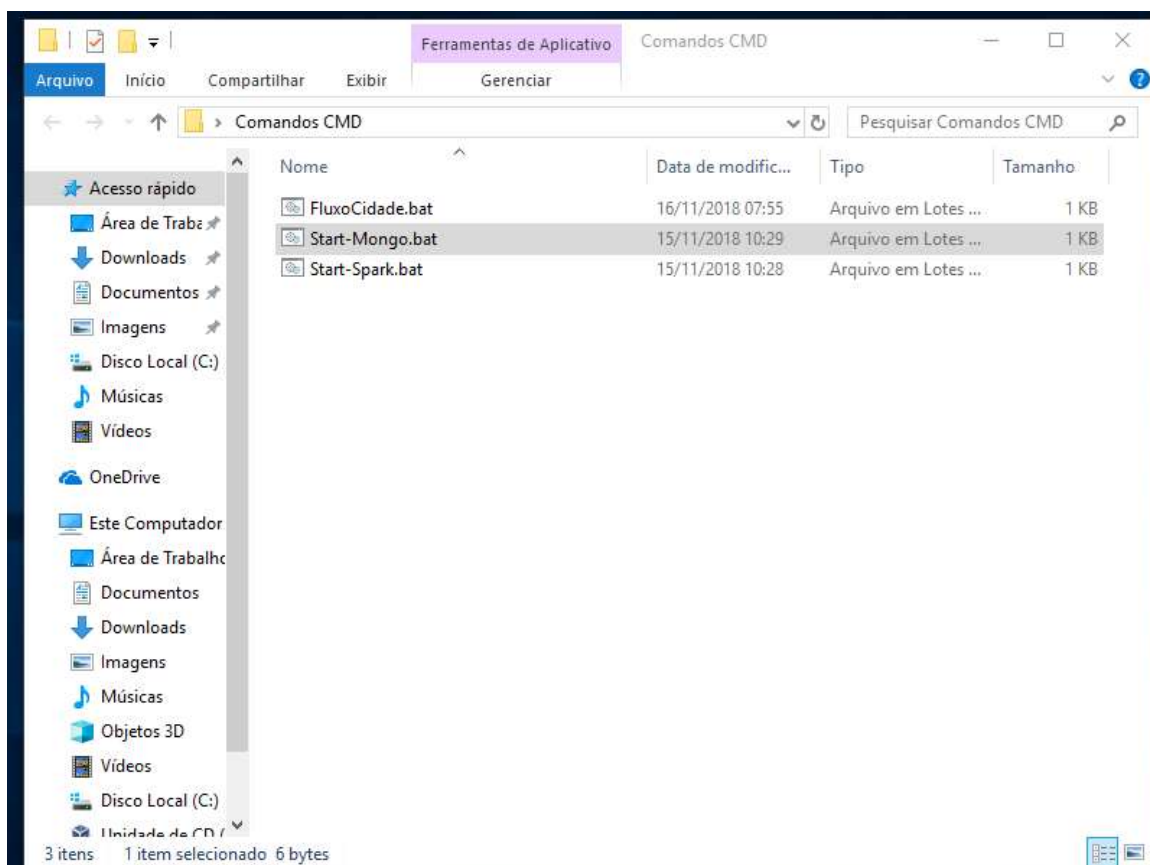
Como citado anteriormente foi utilizado uma virtualização do sistema operacional Windows 10, ao ligar a máquina o sistema operacional irá se logar automaticamente no usuário **Big Data**, caso solicite uma senha, pode-se prosseguir deixando o campo em branco (sem senha), esse comportamento se trata de um bug específico da versão do sistema operacional.

Com o intuito de facilitar as operações todos os arquivos executados foram adicionados a área de trabalho. Também foram disponibilizados arquivos da extensão .bat, esses arquivos foram criados inicialmente para o MS-DOS com o intuito da automatização de tarefas e foram herdados pelos sistemas operacionais Windows. O objetivo desses arquivos é evitar qualquer interação com o console eliminando a necessidade de certos conhecimentos técnicos por parte de qualquer um que venha a replicar o experimento, todos esses arquivos se

encontram na pasta **Comandos CMD** localizados na área de trabalho, como ilustrado na Figura A 6 e na Figura A7.



A 6 – Arquivos .bat

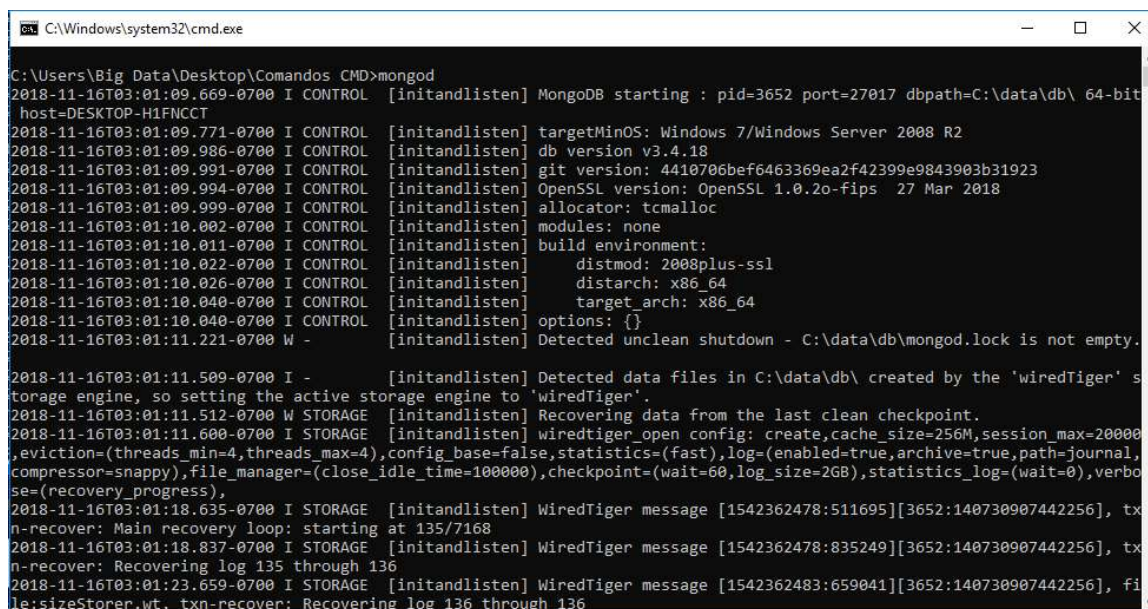


A 7 – Arquivos disponíveis dentro da pasta Comandos CMD

A3. Iniciando o MongoDB

Na pasta **Comandos CMD** localizada na área de trabalho execute o seguinte arquivo **Start-Mongo.bat**, ele irá iniciar a base de dados MongoDB

automaticamente. Caso você queira fazer esse procedimento manualmente inicie o prompt de comando e digite o seguinte comando **mongod**, esse comando irá iniciar a instância do MongoDB como demonstrado na Figura A8 abaixo:



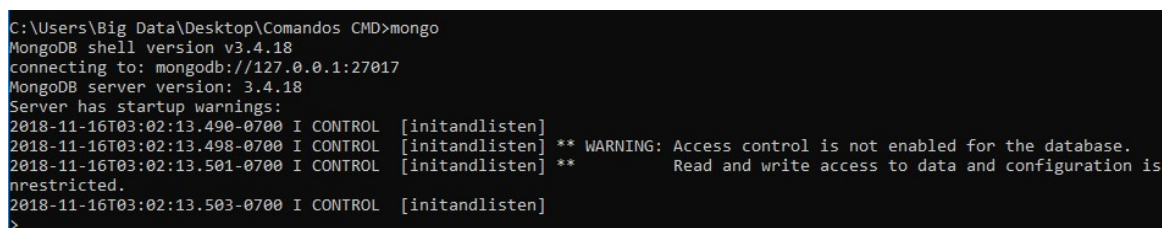
```

C:\Windows\system32\cmd.exe
C:\Users\Big Data\Desktop\Comandos CMD>mongod
2018-11-16T03:01:09.669-0700 I CONTROL [initandlisten] MongoDB starting : pid=3652 port=27017 dbpath=C:\data\db\ 64-bit
host=DESKTOP-H1FNCCT
2018-11-16T03:01:09.771-0700 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2018-11-16T03:01:09.986-0700 I CONTROL [initandlisten] db version v3.4.18
2018-11-16T03:01:09.991-0700 I CONTROL [initandlisten] git version: 4410706bef6463369ea2f42399e9843903b31923
2018-11-16T03:01:09.994-0700 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.2o-fips 27 Mar 2018
2018-11-16T03:01:09.999-0700 I CONTROL [initandlisten] allocator: tcmalloc
2018-11-16T03:01:10.002-0700 I CONTROL [initandlisten] modules: none
2018-11-16T03:01:10.011-0700 I CONTROL [initandlisten] build environment:
2018-11-16T03:01:10.022-0700 I CONTROL [initandlisten] distmod: 2008plus-ssl
2018-11-16T03:01:10.026-0700 I CONTROL [initandlisten] distarch: x86_64
2018-11-16T03:01:10.040-0700 I CONTROL [initandlisten] target_arch: x86_64
2018-11-16T03:01:10.040-0700 I CONTROL [initandlisten] options: {}
2018-11-16T03:01:11.221-0700 W - [initandlisten] Detected unclean shutdown - C:\data\db\mongod.lock is not empty.
2018-11-16T03:01:11.509-0700 I - [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger' s
storage engine, so setting the active storage engine to 'wiredTiger'.
2018-11-16T03:01:11.512-0700 W STORAGE [initandlisten] Recovering data from the last clean checkpoint.
2018-11-16T03:01:11.600-0700 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=256M,session_max=20000
,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,
compressor=snappy),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),statistics_log=(wait=0),verbo
se=(recovery_progress),
2018-11-16T03:01:18.635-0700 I STORAGE [initandlisten] WiredTiger message [1542362478:511695][3652:140730907442256], tx
n-recover: Main recovery loop: starting at 135/7168
2018-11-16T03:01:18.837-0700 I STORAGE [initandlisten] WiredTiger message [1542362478:835249][3652:140730907442256], tx
n-recover: Recovering log 135 through 136
2018-11-16T03:01:23.659-0700 I STORAGE [initandlisten] WiredTiger message [1542362483:659041][3652:140730907442256], fi
le:sizeStorer.wt, txn-recover: Recovering log 136 through 136

```

A 8 – Inicializando o mongod via linha de comando

Caso haja interesse em entender e/ou visualizar os dados persistidos no Mongo é possível efetuar isso através do comando **mongo** em um segundo terminal do Windows. Uma observação é que todas as bases de dados foram criadas com seus usuários padrões de conexão, essa medida foi tomada com o intuito de facilitar a transparência dos processos durante os testes neste ambiente isolado, em um cenário real as mesmas teriam uma topologia de segregação de acessos com credenciais robustas além de outros protocolos de segurança em todas as comunicações. A Figura A9 abaixo demonstra o procedimento citado:



```

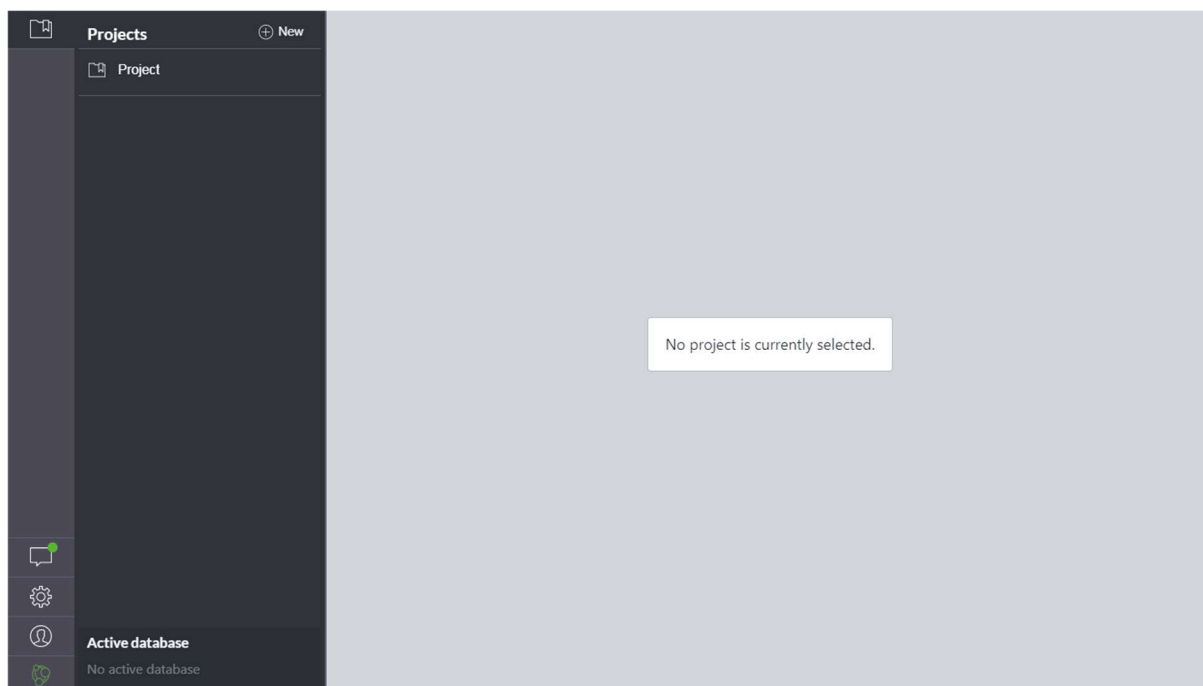
C:\Users\Big Data\Desktop\Comandos CMD>mongo
MongoDB shell version v3.4.18
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.18
Server has startup warnings:
2018-11-16T03:02:13.490-0700 I CONTROL [initandlisten]
2018-11-16T03:02:13.498-0700 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2018-11-16T03:02:13.501-0700 I CONTROL [initandlisten] ** Read and write access to data and configuration is
unrestricted.
2018-11-16T03:02:13.503-0700 I CONTROL [initandlisten]
>

```

A 9 – Aplicação de regras de controle de acesso aos arquivos

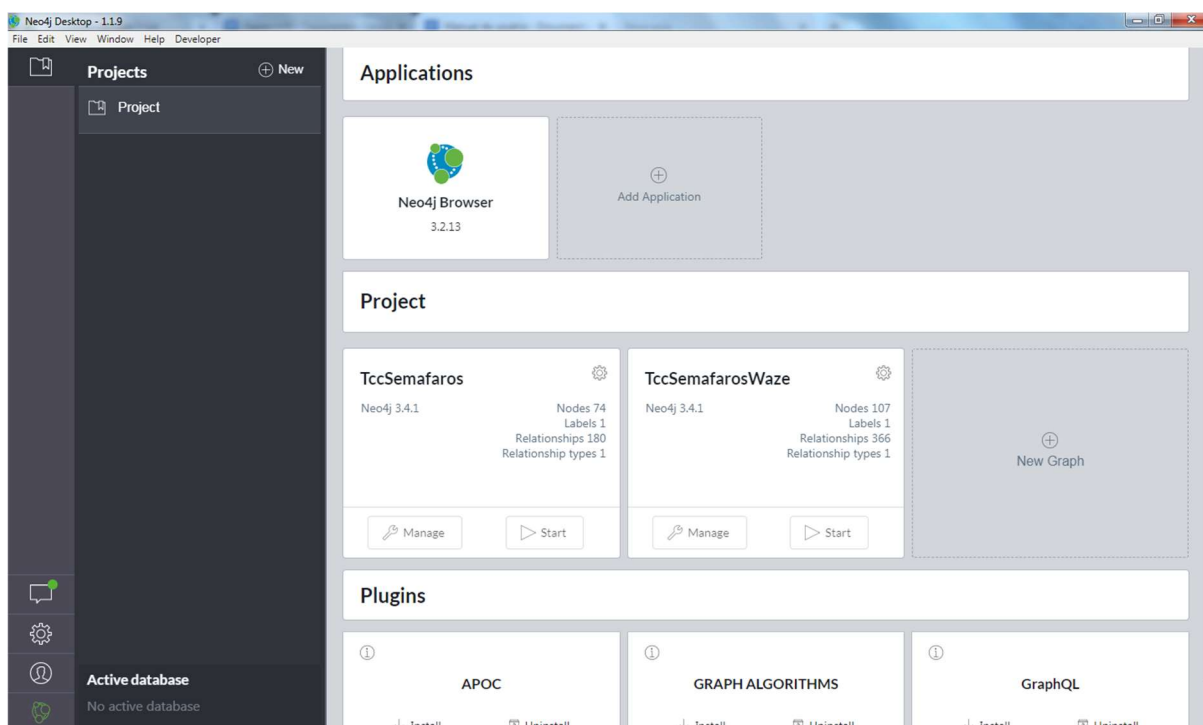
A4. Iniciando o Neo4j

Clique no ícone **Neo4j Desktop** localizado na área de trabalho, após a inicialização do software clique na opção **Projectct** localizada no menu lateral esquerdo Projects como demonstrado na Figura A10 abaixo.



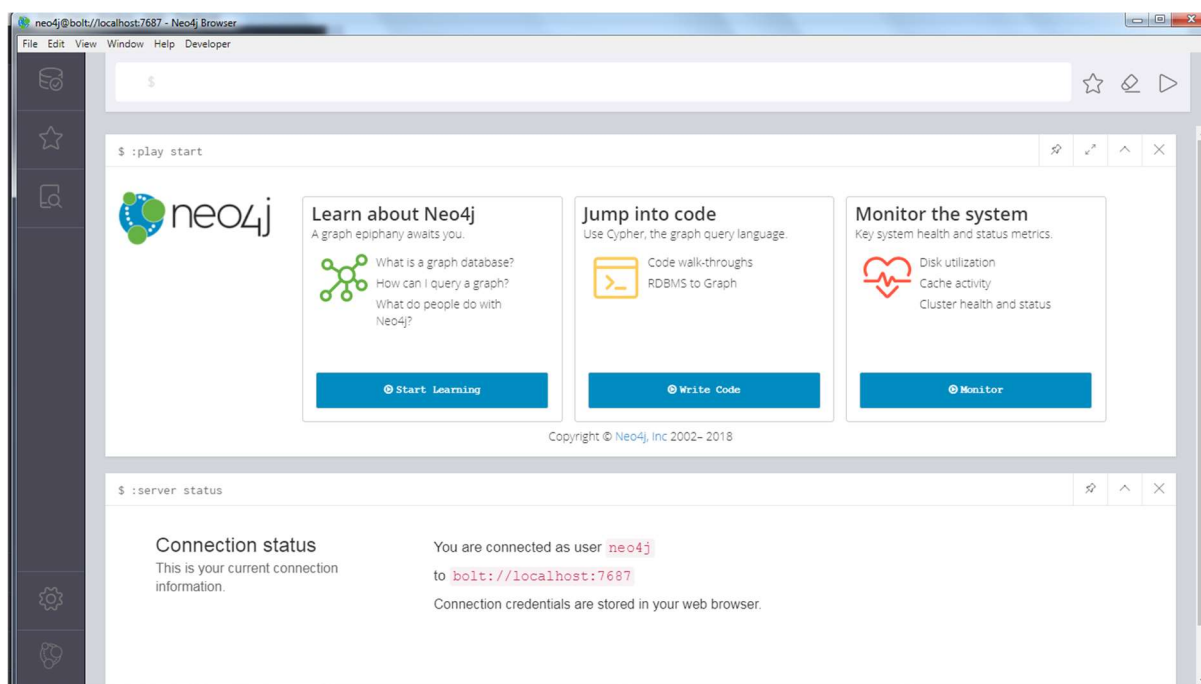
A 10 – Inicialização do Neo4j

Após isso será exibido algumas opções no lado direito, clique no botão **Start** localizado dentro do projeto **TccSemafaros**, aguarde alguns instantes até esse mesmo botão ser substituído por um outro botão com a opção de **Stop**. Após isso a base já estará inicializada como evidenciado na Figura A11.



A 11 – Painel de inicialização do serviço

Caso haja interesse em entender e/ou visualizar os dados é possível efetuar isso clicando na opção **Neo4J Browser** localizada na tela inicial, após isso será exibido uma janela como a demonstrada na imagem A12. Nela será possível executar instruções cypher queries.

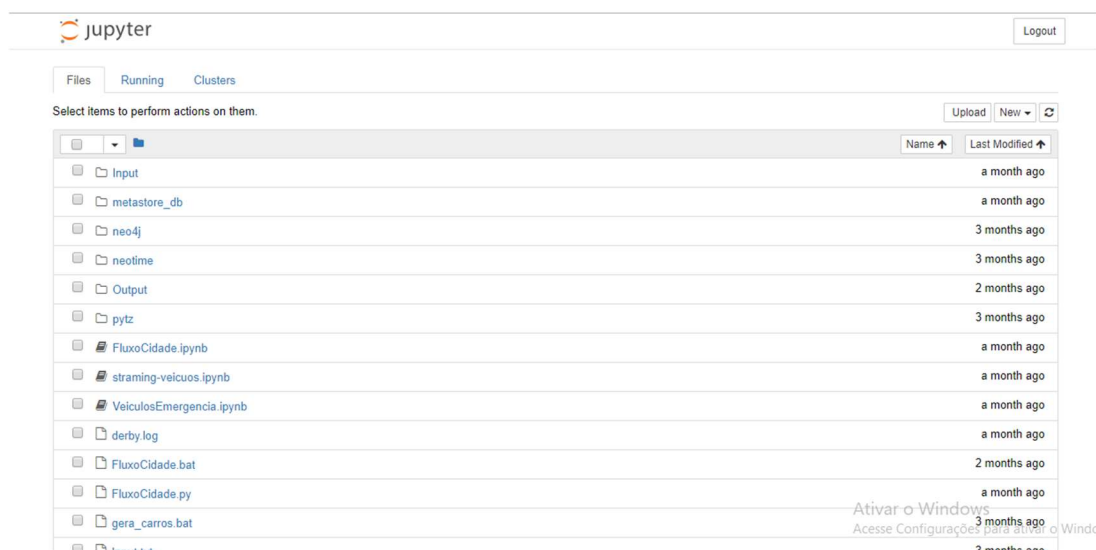


A 12 – Tela de navegação do Neo4j

A5. Iniciando o Spark

Após a conclusão das etapas anteriores volte para a área de trabalho, e entre na pasta **Comandos CMD**, nela haverá um arquivo chamado de **Start-Spark.bat**, clique com o botão direito sobre ele e depois na opção executar como administrador, caso o arquivo seja executado normalmente sem a permissão de administrador haverá conflitos de execução que inviabilizaram o resto do experimento.

Aguarde alguns segundos até a inicialização do Jupyter Notebook no navegador Microsoft Edge. O Jupyter Notebook é uma aplicação web que permite o compartilhamento de documentos e a execução de scripts, ele possui diversas funcionalidades que facilitam a visualização da execução e dos resultados dos scripts implementados. A Figura A13 abaixo mostra a tela inicial do Jupyter, nela podemos observar os arquivos presentes no diretório, existem três arquivos principais que são os que efetuaram os processos citados no subcapítulo **3.7 Arquitetura Física**.



A 13 – Tela inicial do Jupyter

Observação: Caso haja a necessidade ou o interesse de efetuar esse procedimento manualmente, você pode abrir um novo prompt de comando no modo administrador, e dentro da pasta **Spark-Scripts** executar o comando **pyspark**, após o fim da execução deste comando o Jupyter Notebook será executado no navegador como no procedimento citado anteriormente.

A6. Executando o algoritmo da Camada de Emulação do Trânsito

Após isso volte pasta **Comandos CMD**, execute o arquivo **FluxoCidade.bat**, esse arquivo irá disparar um comando que irá executar o arquivo **FluxoCidade.py** no interpretador do python esse script ficará executando durante todo os demais procedimentos como demonstrado na Figura A14 abaixo:

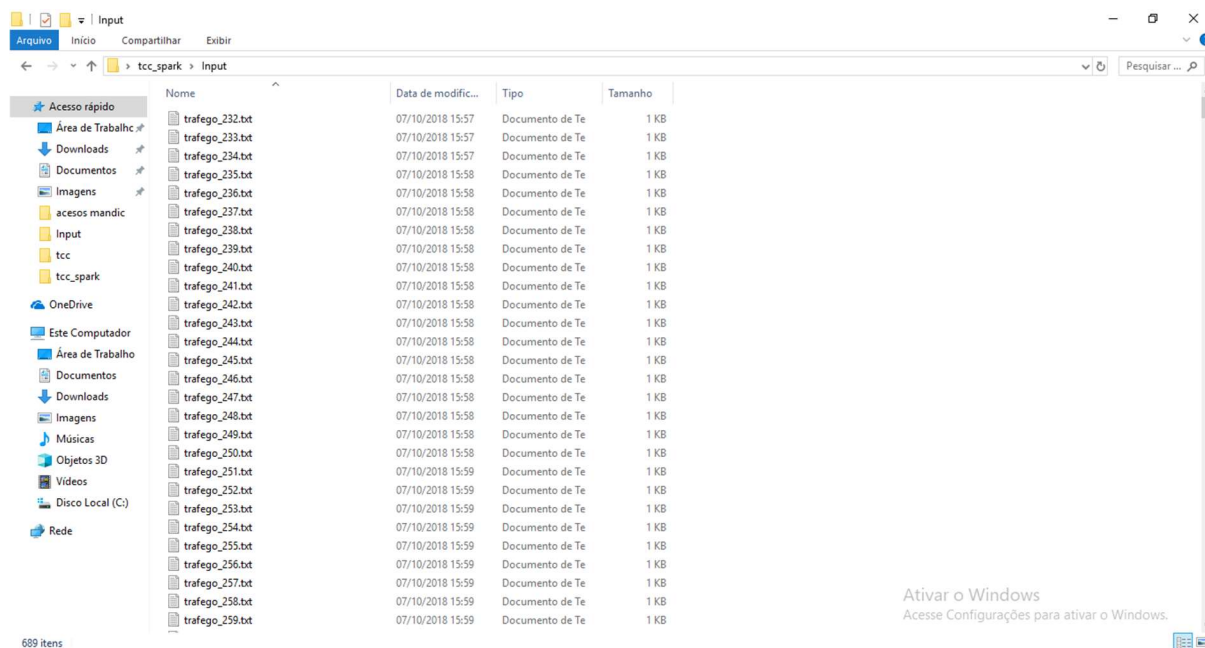
```

C:\Windows\System32\cmd.exe
C:\Users\Infinity\Desktop\tcc_spark>python FluxoCidade.py
FluxoCidade.py:80: SyntaxWarning: name 'filedata' is assigned to before global declaration
global filedata
('-----', 0, 24)
('-----', 0, 4)
1 - Semafaro aberto na via 0
=====
('0', '4')
=====
('-----', 0, 1)
1 - Semafaro aberto na via 0
=====
('0', '1')
=====
('-----', 1, 26)
('-----', 1, 2)
('-----', 1, 0)
1 - Semafaro aberto na via 1
=====
('1', '0')
=====
('-----', 2, 28)
('-----', 2, 5)
('-----', 2, 1)
('-----', 2, 3)
1 - Semafaro aberto na via 2
=====
('2', '3')
=====

```

A 14 – Executando o algoritmo via linha de comando

O motivo de não executarmos esse processo pelo Jupyter Notebook foi devido ao fato de o mesmo utilizar o pyspark como motor para a execução, e isso geraria um bloqueio nos demais processos já que essa aplicação não consegue executar mais de um script por vez. Como resultado o algoritmo irá gerar a massa de dados emulando o comportamento dos sensores, na pasta **Input** localizada em **C:\Users\Big Data\Desktop\tcc_spark**, a Figura A15 abaixo demonstra esse resultado, é possível observar que ao longo do tempo novos arquivos serão armazenados nesse diretório, eles serão a base do nosso processamento em streaming que será descrito a seguir.



A 15 – Pasta de armazenamento dos dados gerados

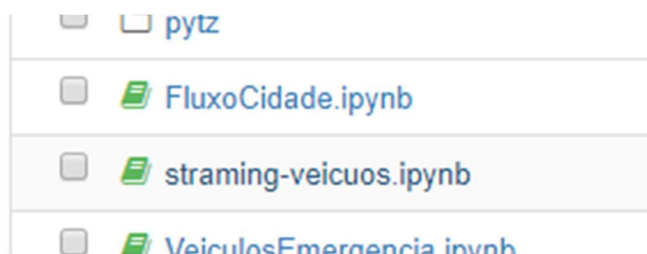
A7. Executando o algoritmo da Camada de Análise dos Dados dos Sensores

Após a ativação do script **FluxoCidade.bat** descrito no subcapítulo **Executando o algoritmo da Camada de Emulação do Trânsito**, iremos prosseguir com a etapa de captura dos dados de tráfego dos veículos e do seu processamento através de uma arquitetura de streaming, o funcionamento desse script foi descrito nos subcapítulos **3.5 Definição do Melhor Caminho** e **3.6 Otimização do Trajeto**. Inicialmente iremos voltar para o navegador **Microsoft Edge**, na aba contendo a tela inicial do Jupyter Notebook como demonstrado na Figura A16 abaixo:



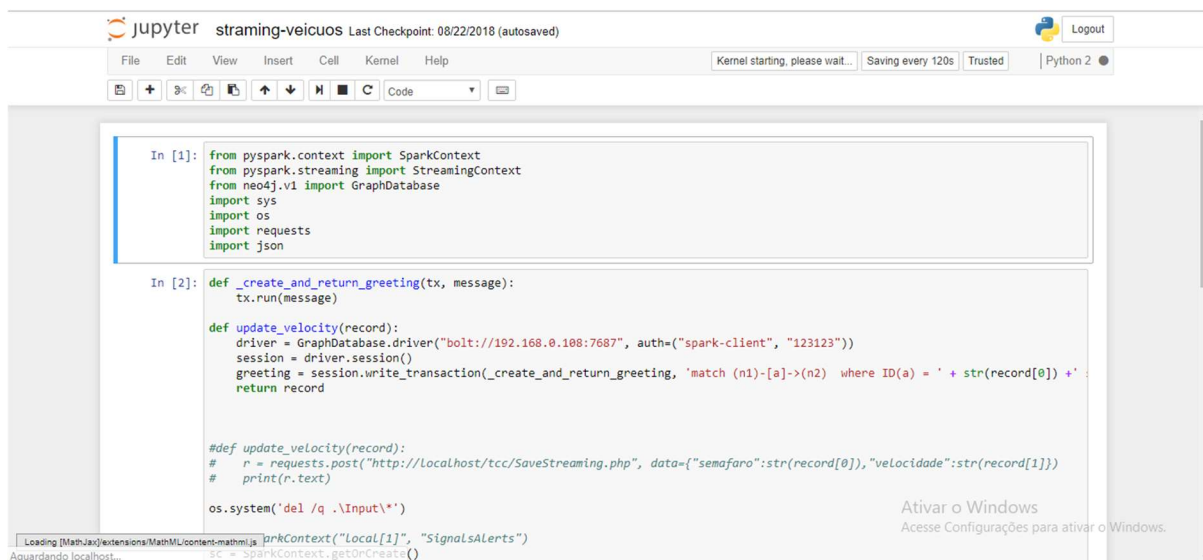
A 16 – Arquivos armazenados no Jupyter

Na tela inicial do Jupyter haverá um link de um arquivo com o nome **straming-veiculos.ipnb** como ilustrado na Figura A17 abaixo, abra ele em uma nova guia clicando com o botão direito do mouse e depois em “**Abrir em uma nova guia**” ou pelo atalho de teclado **Ctrl + I** e clicando com o botão esquerdo do mouse.



A 17 – Arquivos Jupyter

Na nova guia será exibido uma tela semelhante à Figura A18, essa é a tela de edição de documentos do Jupyter Notebook, podemos dividir essa tela em duas etapas, na parte superior se encontra-se uma grande quantidade de opções para o gerenciamento dos arquivos, a seguir uma breve explicação sobre algumas funções da ferramenta.

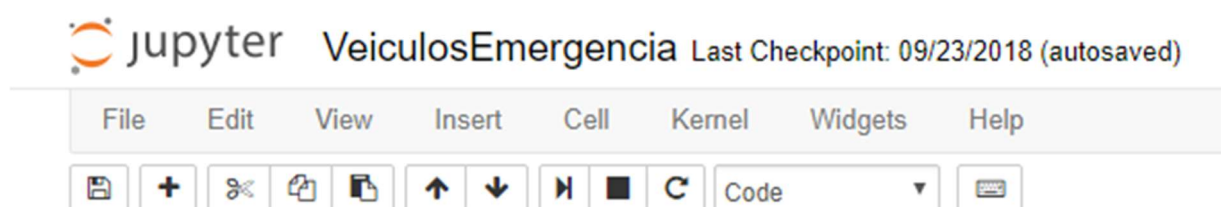


A 18 – Edição de documento no Jupyter

A Figura A19 abaixo mostra a parte superior na ferramenta onde ficam concentradas diversas opções de ações e configurações da ferramenta, na parte superior destacado no menu na cor cinza.

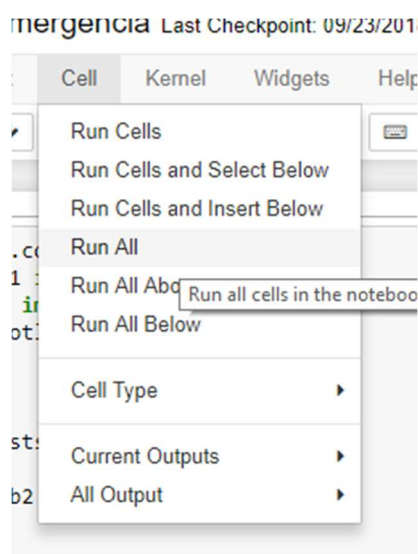
- **File:** Aba dedicada ao gerenciamento do arquivo, nessa aba é possível abrir, salvar, renomear, imprimir o documento entre outras funções.
- **Edit:** Aba dedicada ao gerenciamento dos itens do documento, na topologia do Jupyter Notebook todos os trechos em um documento são tratados como uma célula específica isso permite com que cada processo seja isolado, dentre as principais funções se encontram opções de copiar, colar, recortar e mesclar as células.
- **View:** Opções relacionadas a customização da exibição dos widgets do editor.
- **Insert:** Aba dedicada a inserção das células no documento.
- **Cell:** Aba dedicada a execução das células, nesta aba se encontra uma opção chamada de “Run All”, essa opção executa todas as células do notebook.
- **Kermel:** Aba dedicada a ações do kernel, nela é possível reiniciar, interromper e trocar o kernel de execução.
- **Widgets:** Aba responsável pelo gerenciamento de plugins.
- **Help:** Manuais de usuário

Abaixo desse menu se encontram as opções de acesso rápido da ferramenta, as suas funções respectivamente da esquerda para a direita são: salvar o documento, adicionar uma célula, recortar, copiar, colar, mover a célula para cima, mover a célula para baixo, executar a célula selecionada, interromper a execução e recarregar a tela, ao lado desse último botão existem dois componentes onde o primeiro é uma caixa de seleção, nela é definida quais características a célula terá, por exemplo, a mesma pode ser um código executável ou um trecho de texto explicativo, ao lado se encontra uma paleta de comandos avançados da ferramenta.



A 19 – Menus e ferramentas disponíveis na tela do Jupyter

Clique no item do menu superior Cell após isso na opção de **Run All**, isso irá executar todas o script de todas as células como na imagem A20.



A 20 – Executando os scripts

Para acompanhar a execução do processamento em streaming mova o scroll da página até a parte inferior, note que após a terceira célula o haverá um output semelhante ao da imagem A21 abaixo, nele é possível conferir o relatório dos dados processados.

```

(90, 170)
-----
Time: 2018-10-07 16:03:15
(22, 150)
-----
Time: 2018-10-07 16:03:20
(90, 170)
-----
Time: 2018-10-07 16:03:30
(22, 150)
-----
Time: 2018-10-07 16:03:35

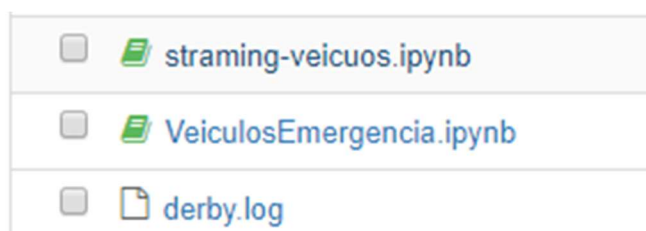
```

A 21 – Processamento em streaming

A8. Executando o algoritmo Camada de Definição e otimização do Trajeto

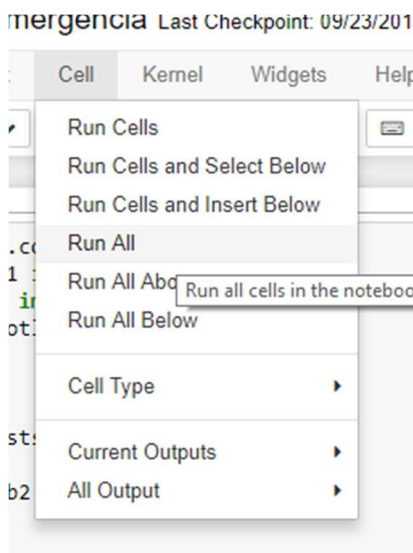
Após o término do processo descrito na Seção A6 é possível utilizar o script responsável pela seleção e otimização do melhor trajeto, o funcionamento detalhado desse script descrito no subcapítulo .

Volte para aba contendo a tela inicial do Jupyter Notebook no navegador, após isso abra o arquivo **VeiculosEmergencia.ipynb** em uma nova guia clicando com o botão direito do mouse e depois em “**Abrir em uma nova guia**” ou pelo atalho de teclado **Ctrl + I** e clicando com o botão esquerdo do mouse como demonstrado na imagem A22.



A 22 – Arquivos Jupyter

Na nova guia clique no item do menu superior Cell após isso na opção de Run All, isso irá executar todas o script de todas as células como demonstrado na imagem A23.



A 23 – Executando os scripts

Esse processo irá iniciar o algoritmo de otimização do trajeto, após a sua execução será obtido o seguinte resultado, onde como demonstrado na imagem A24 primeiro será exibido o percurso selecionado e após isso os pontos a serem otimizados nos semáforos.

```
=====
Mapeando as rotas
=====
trajeto 0 tempo estimado 73 segundos

--> Selecionou o caminho 0 pois possui o menor tempo de trajeto

trajeto 1 tempo estimado 77 segundos
trajeto 2 tempo estimado 85 segundos
trajeto 3 tempo estimado 91 segundos
trajeto 4 tempo estimado 82 segundos
trajeto 5 tempo estimado 88 segundos
trajeto 6 tempo estimado 89 segundos
trajeto 7 tempo estimado 88 segundos
trajeto 8 tempo estimado 88 segundos
trajeto 9 tempo estimado 84 segundos
trajeto 10 tempo estimado 86 segundos
trajeto 11 tempo estimado 87 segundos
trajeto 12 tempo estimado 83 segundos
trajeto 13 tempo estimado 86 segundos
trajeto 14 tempo estimado 83 segundos
trajeto 15 tempo estimado 83 segundos
trajeto 16 tempo estimado 83 segundos
trajeto 17 tempo estimado 89 segundos
```

A 24 - Resultado final