

**mAPEAMENTO DAS CONDIÇÕES ASFÁLTICAS por meio dA AQUISIÇÃO DE DADOS VIA APLICATIVO ANDROID COM USO DE ALGORITMO DE APRENDIZADO DE MÁQUINA**

Lucas Cavalcanti Adorno

Projeto de Graduação apresentado ao Curso de

Engenharia Eletrônica e de Computação da Escola

Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientador: Flávio Luís de Mello

Rio de Janeiro

Março de 2018

**mAPEAMENTO DAS CONDIÇÕES ASFÁLTICAS por meio dA AQUISIÇÃO DE DADOS VIA APLICATIVO ANDROID COM USO DE ALGORITMO DE APRENDIZADO DE MÁQUINA**

Flávio Luís de Mello

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE DO CURSO DE ENGENHARIA ELETRÔNICA E DE COMPUTAÇÃO DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO ELETRÔNICO E DE COMPUTAÇÃO

Autor:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lucas Cavalcanti Adorno

Orientador:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Flávio Luís de Mello, D. Sc.

Examinador:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Heraldo Luis Silveira Almeida, D. Sc.

Examinador:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

XX

Rio de Janeiro – RJ, Brasil

Março de 2018

Declaração de Autoria e de Direitos

Eu, *Lucas Cavalcanti Adorno*, CPF *147294527-12*, autor da monografia *MAPEAMENTO DAS CONDIÇÕES ASFÁLTICAS POR MEIO DA AQUISIÇÃO DE DADOS VIA APLICATIVO ANDROID COM USO DE ALGORITMO DE APRENDIZADO DE MÁQUINA*, subscrevo para os devidos fins, as seguintes informações:

1. O autor declara que o trabalho apresentado na disciplina de Projeto de Graduação da Escola Politécnica da UFRJ é de sua autoria, sendo original em forma e conteúdo.

2. Excetuam-se do item 1. Eventuais transcrições de texto, figuras, tabelas, conceitos e ideias, que identifiquem claramente a fonte original, explicitando as autorizações obtidas dos respectivos proprietários, quando necessárias.

3. O autor permite que a UFRJ, por um prazo indeterminado, efetue em qualquer mídia de divulgação, a publicação do trabalho acadêmico em sua totalidade, ou em parte. Essa autorização não envolve ônus de qualquer natureza à UFRJ, ou aos seus representantes.

4. O autor pode, excepcionalmente, encaminhar à Comissão de Projeto de Graduação, a não divulgação do material, por um prazo máximo de 01 (um) ano, improrrogável, a contar da data de defesa, desde que o pedido seja justificado, e solicitado antecipadamente, por escrito, à Congregação da Escola Politécnica.

5. O autor declara, ainda, ter a capacidade jurídica para a prática do presente ato, assim como ter conhecimento do teor da presente Declaração, estando ciente das sanções e punições legais, no que tange a cópia parcial, ou total, de obra intelectual, o que se configura como violação do direito autoral previsto no Código Penal Brasileiro no art.184 e art.299, bem como na Lei 9.610.

6. O autor é o único responsável pelo conteúdo apresentado nos trabalhos acadêmicos publicados, não cabendo à UFRJ, aos seus representantes, ou ao(s) orientador(es), qualquer responsabilização/ indenização nesse sentido.

7. Por ser verdade, firmo a presente declaração.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lucas Cavalcanti Adorno

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Escola Politécnica – Departamento de Eletrônica e de Computação

Centro de Tecnologia, bloco H, sala H-217, Cidade Universitária

Rio de Janeiro – RJ CEP 21949-900

Este exemplar é de propriedade da Universidade Federal do Rio de Janeiro, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es).

**DEDICATÓRIA**

Dedico a elaboração do trabalho à minha família e à comunidade acadêmica, em especial ao meu orientador Flávio Luis de Mello por estar sempre disposto em ajudar e de elaborar um projeto importante para a sociedade como um todo.

**AGRADECIMENTO**

Agradeço a todas as pessoas presentes nesses longos anos de caminhada na faculdade. É notório que este foi o momento da minha vida de maior transformação psicológica e caráter como um todo. Por todos os momentos presenciados na UFRJ, sou grato a todos que participaram direta e indiretamente.

**RESUMO**

O presente trabalho tem como finalidade desenvolver um aplicativo para dispositivos Android que permita, de forma intuitiva, a captura de informações das vias pavimentadas. A identificação dos eventos existentes nos trajetos percorridos, como buracos e quebra-molas, será realizada de forma automatizada através do uso de um algoritmo de aprendizado de máquina. Além disso, não menos importante, será desenvolvido um serviço público hospedado na internet que possibilite o consumo por qualquer pessoa de quaisquer interesses nos dados adquiridos pelo aplicativo mobile. O projeto como um todo utiliza as linguagens de programação Python, C# e Java. O aplicativo é desenvolvido para smartphones Android acima da versão Kitkat (4.x). Neste, é concentrado a lógica para captura de dados do sensor acelerômetro e também das coordenadas do GPS presente no dispositivo móvel. A etapa do processamento dos dados se dá através de processos desenvolvidos em C# que estão hospedados na *cloud* e também de algoritmos em Python responsável por toda a lógica de *machine learning*. A respeito do acesso aos registros salvos no banco de dados, foi criado uma *Web API* em C# com a IDE Visual Studio 2017 que realizará o intermédio entre o dado e o usuário. Nessa aplicação web, o usuário poderá acessar dados de quaisquer obstáculos, em qualquer intervalo de tempo, de uma localidade específica e outras funcionalidades que serão apresentadas ao longo do documento. Neste trabalho, são abordados tópicos a respeito das linguagens de programação, tecnologias empregadas, assim como os conceitos matemáticos e estatísticos manuseados para a construção do projeto. Por fim, são descritas e apresentadas as conclusões a respeito dos resultados obtidos, além da descrição de todas as etapas construídas para o funcionamento da aplicação.

Palavras-Chave: aprendizado de máquina, captura de obstáculos na pista, identificação de obstáculos na pista, mapeamento da qualidade asfáltica, linguagem de programação, aplicativo Android, desenvolvimento WEB API.

**ABSTRACT**

Insert your abstract here. Insert your abstract here. Insert your abstract here. Insert your abstract here. Insert your abstract here.

Key-words: word, word, word.

**SIGLAS**

API - Interface de Programação de Aplicações

DNIT - Departamento Nacional de Infraestrutura de Transportes.

FTP - Protocolo de Transferência de Arquivos

HTTP - Protocolo de Transferência de Hipertexto

IDE - Ambiente de Desenvolvimento Integrado

IEEE - Instituto de Engenheiros Eletricistas e Eletrônicos

IoT - Internet das Coisas

JSON - Notação de Objetos JavaScript

MBA - Mestre em Administração de Negócios

OMS - Organização Mundial da Saúde

REST - Transferência de Estado Representacional

RPC - Chamada de Procedimento Remoto

SMTP - Protocolo de Transferência de Correio Simples

SOAP - Protocolo Simples de Acesso a Objetos

UFRJ - Universidade Federal do Rio de Janeiro

URL - Localizador Uniforme de Recursos

WSDL - Descrição de Linguagem de Web Services

XML - Linguagem Extensível de Marcação Genérica

**Sumário**

[Introdução 1](#_Toc505672263)

[1.1 – Tema 1](#_Toc505672264)

[1.2 – Delimitação 1](#_Toc505672265)

[1.3 – Justificativa 2](#_Toc505672266)

[1.4 – Objetivos 3](#_Toc505672267)

[1.5 – Metodologia 3](#_Toc505672268)

[1.6 – Descrição 4](#_Toc505672269)

[Fundamentação Teórica 4](#_Toc505672270)

[2.1 – Linguagem Python 5](#_Toc505672271)

[2.2 – Linguagem C# 6](#_Toc505672272)

[2.3 – Linguagem Java 7](#_Toc505672273)

[2.4 – Web API 9](#_Toc505672274)

[2.5 – Sensor Acelerômetro 13](#_Toc505672275)

[2.6 – Normalização 16](#_Toc505672276)

[2.7 – Trabalhos Relacionados 17](#_Toc505672277)

[Tecnologias na Cloud 24](#_Toc505672278)

[3.1 – Amazon Web Service 25](#_Toc505672279)

[3.2 – Amazon Elastic Compute Cloud (EC2) 27](#_Toc505672280)

[3.3 – Amazon Simple Queue Service (SQS) 30](#_Toc505672281)

[Aquisição dos Dados 30](#_Toc505672282)

[Projeto Lunar 30](#_Toc505672283)

[Conclusões 30](#_Toc505672284)

[Bibliografia 31](#_Toc505672285)

**Lista de Figuras**

[Figura 1X.X LEGENDA AQUI 6](#_Toc505843170)

[Figura 2X.X LEGENDA AQUI 8](#_Toc505843171)

[Figura 3X.X LEGENDA AQUI 9](#_Toc505843172)

[Figura 4X.X LEGENDA AQUI 10](#_Toc505843173)

[Figura 5X.X LEGENDA AQUI 11](#_Toc505843174)

[Figura 6X.X LEGENDA AQUI 12](#_Toc505843175)

[Figura 7X.X LEGENDA AQUI 12](#_Toc505843176)

[Figura 8X.X LEGENDA AQUI 14](#_Toc505843177)

[Figura 9X.X LEGENDA AQUI 15](#_Toc505843178)

[Figura 10X.X LEGENDA AQUI 15](#_Toc505843179)

[Figura 11X.X LEGENDA AQUI 16](#_Toc505843180)

[Figura 12X.X LEGENDA AQUI 20](#_Toc505843181)

[Figura 13X.X LEGENDA AQUI 22](#_Toc505843182)

[Figura 14X.X LEGENDA AQUI 23](#_Toc505843183)

[Figura 15 X.X LEGENDA AQUI 26](#_Toc505843184)

[Figura 16 X.X LEGENDA AQUI 28](#_Toc505843185)

[Figura 17 X.X LEGENDA AQUI 29](#_Toc505843186)

[Figura 18 X.X LEGENDA AQUI 29](#_Toc505843187)

**Lista de Tabelas**

|  |  |
| --- | --- |
| 2.1 – Casos de ataques aos computadores da Intranet . . . . . . . . . . . . . . . . . . . . . . | 1 |

**Capítulo 1**

# Introdução

1.1 – Tema

O tema deste projeto é a utilização do sensor acelerômetro e do GPS de sistemas operacionais Android para captura dos obstáculos em vias pavimentadas e a criação de um serviço público web para facilitar, aos usuários, o acesso aos dados. Neste sentido, o problema a ser resolvido é a criação de uma aplicação de baixo custo capaz de capturar as informações necessárias e também de identificar automaticamente os eventos percebidos, com o propósito de realizar a medição das condições de rodagem de uma malha rodoviária utilizando smartphones.

1.2 – Delimitação

Ao que se refere ao processo da captura dos dados, o público-alvo é divido em dois diferentes grupos de usuários. Os usuários comuns de dispositivos móveis que possuem os recursos necessários no aparelho que querem ampliar a base de dados de forma voluntária. O segundo grupo está relacionado com os usuários que trabalham para algum serviço público e que almejam elaborar estudos de baixo custo a respeito do mapeamento das condições das vias pavimentadas contribuindo com a inserção de novos registros ao banco de dados.

No que tange na utilização do serviço web para acessar os dados, é destinado aos funcionários de serviços públicos, corpo discente e docente de quaisquer instituições de ensino, como também interessados em trabalhar com os dados para algum projeto pessoal, empresarial ou governamental.

O tipo de rua a ser analisado também contém algumas restrições. Somente estradas pavimentadas serão estudadas pela aplicação. Pistas de terra e paralelepípedo não se enquadram no escopo do projeto, devendo ser objeto de trabalho a parte.

Apesar desta delimitação, toda a sociedade poderá tirar proveito dos resultados obtidos pela aplicação que permitirá a elaboração de um mapa da qualidade asfáltica de qualquer região que tenha sido percorrida por algum usuário pelo menos uma única vez e, consequentemente, facilitará a decisão de medidas para realizar as correções necessárias nos trajetos registrados pelo projeto.

1.3 – Justificativa

É sabido por todos os brasileiros que a qualidade das estradas pavimentadas no Brasil não é sinônimo de orgulho ou de um trabalho bem feito. Deixando os motivos de lado, constantemente são encontrados, mesmo nas estradas recém-inauguradas, problemas como fissuras, deformações e buracos.

Além da má impressão que passa ao se deparar com estes eventos, o principal problema é a contribuição direta para o número de acidentes em vias, tanto no caso brasileiro como para os outros países do mundo. Segundo um estudo publicado em 2015 pela *World Health Organization* (OMS) **[X]**, o Brasil, em 2013, ficou na primeira posição com o maior número de mortes no trânsito a cada 100 mil habitantes considerando os países da América do Sul, seja por problemas na via ou devido à imprudência dos motoristas. Ao mesmo momento, o Brasil é destacado por aplicar numerosas leis de controle de risco.

O governo brasileiro, mais especificamente o DNIT, é o responsável pela regularização e normas de manutenção das vias. Além disso, este departamento realiza o mapeamento das condições dos pavimentos. O processo é custoso e exige uma série de ferramentas, como também etapas penosas para escanear as estradas. O interesse é público, porém os dados adquiridos são privados.

A aplicação desenvolvida neste projeto, inspirada nos apontamentos de Laudelino Amaral de Oliveira Lima [2], apresentará uma forma de custo ínfimo para permitir o mapeamento das regiões que possuem vias pavimentadas e, ao mesmo tempo, os dados poderão ser analisados, observados por qualquer cidadão interessado.

1.4 – Objetivos

O trabalho, na sua essência, possui três objetivos primordiais que são: elaboração de um aplicativo mobile, desenvolvimento de algoritmo de aprendizado de máquina e a criação de um serviço web que seja o intermediário entre o usuário final e a base de dados. Assim, os objetivos específicos do trabalho são:

1. Criação de uma aplicação Android para a coleta dos dados dos sensores acelerômetro e GPS.
2. Criação de processos internos para processar os dados enviados pelo smartphone e salvá-los em um banco de dados.
3. Estudo e aplicação de serviços na nuvem para a hospedagem tanto dos processos como do banco de dados.
4. Criação de um serviço web para facilitar o acesso dos usuários aos dados.

1.5 – Metodologia

O foco inicial foi em como e quais dados deveriam ser capturados para que a base de dadostornasse o projeto utilizável. A aprendizagem foi iniciada estudando os sensores acelerômetro e giroscópio presentes em sistemas operacionais Android pelo fato de fornecer as acelerações nos eixos x, y e z do dispositivo em que está sendo observado. O sistema operacional Android foi escolhido devido a sua presença abundante no mercado mundial.

Foi realizada uma aplicação teste para os dispositivos Android com o intuito de verificar se estes componentes eletrônicos eram sensíveis o suficiente para a medição dos dados do projeto. Constatado a alta sensibilidade, os testes para captura e análise dos dados foram iniciados.

O celular com aplicação instalada foi inserido em um suporte para smartphone no ar-condicionado permanecendo-o na posição vertical. Com o auxílio de uma câmera fixada no capô do automóvel para realizar a filmagem de todo o percurso, tornou-se possível uma validação e comparação dos dados capturados com as imagens do vídeo adquirido.

Anteposto à elaboração dos gráficos após aquisição dos dados, foram realizadas pesquisas em artigos científicos relacionados ao tema proposto, a fim de obter uma maior clareza sobre os métodos existentes para mapeamento das malhas de rodagem tanto no Brasil como nos demais países. Além disso, iniciou-se um estudo para encontrar o melhor método de aprendizado de máquina que atendesse as exigências do projeto.

Após a pesquisa e a elaboração dos gráficos comparando as informações dos sensores com os eventos que estavam ocorrendo no automóvel nos mesmos instantes de tempo, foi constatado a viabilidade da realização de um mapeamento de baixo custo através de algoritmos de aprendizagem de máquina para detectar, de forma automatizada, os obstáculos nas vias pavimentadas.

A partir da viabilidade do mapeamento ser constatada tanto de modo manual como automatizado, permitiu dar a continuidade para as próximas etapas e também ao projeto como um todo. Baseado nesta factibilidade, deu início à elaboração de uma plataforma Web, mais especificamente de uma Web API, permitindo a qualquer usuário solicitar e estudar os dados simplesmente chamando uma URL em um formato específico.

1.6 – Descrição

**Em desenvolvimento....**

**Capítulo 2**

# Fundamentação Teórica

Para uma melhor compreensão dos assuntos abordados no projeto, são apresentados nas seções seguintes os conceitos fundamentais e suas premissas. Nas seções 2.1, 2.2 e 2.3 são dedicadas à apresentação das linguagens de programação Python, C# e Java respectivamente, nas quais foram utilizadas ao longo de toda a aplicação. Na seção 2.4 é apresentado o conceito de Web API e alguns exemplos com a utilização dos métodos principais de requisições HTTP. Em seguida, é realizada um panorama sobre o sensor acelerômetro e como será empregado no projeto. Adiante, é apresentado a concepção de uma das técnicas de normalização conhecida como *Sliding Window* e o porquê de utilizar técnicas do gênero em base de dados cruas. Por fim, é efetuada uma análise e uma breve descrição da monografia do ex-aluno de MBA Laudelino Amaral de Oliveira Lima, como também periódicos da IEEE relacionados ao escopo do projeto.

2.1 – Linguagem Python

Criada em 1991 por Guido Van Rossum, Python é uma linguagem de programação de alto nível, tipagem dinâmica e orientada a objeto. Reconhecida pela sintaxe clara, excelente documentação e com uma comunidade ativa, Python vem sendo empregada em uma ampla variedade de projetos de quaisquer complexidades.

A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código à velocidade ou expressividade. Combina uma sintaxe clara e concisa com os recursos nativos e também de pacotes desenvolvidos por terceiros já que o seu modelo de desenvolvimento é comunitário e aberto as demais pessoas que queiram contribuir para a comunidade.

Python torna-se uma das linguagens mais utilizadas pela comunidade científica devido as qualidades mencionadas. Existem bibliotecas específicas para cálculos científicos e plotagem de gráficos fazendo pouco uso de linhas de código. A comunidade desenvolveu e continuam desenvolvendo o *SciPy* assim como *Numpy*, *Pandas*, *Matplotlib* que são ferramentas *open-source* voltadas para o meio acadêmico. A utilização também é abundante fora do meio científico. Por apresentar uma sintaxe clara, a linguagem é comumente apresentada para estudo em disciplinas de introdução à programação nas Universidades e cursos técnicos tornando-se porta de entrada dos estudantes ao ambiente da programação.

A biblioteca *Scipy* será necessária para os interessados em utilizar as classes e os métodos referentes ao aprendizado de máquina. Ao realizar a instalação do pacote do *Scipy*, uma série de outros importantes módulos são adicionados ao Python instalado no computador do usuário. O *Numpy* incrementa o Python com um suporte para trabalhar com matrizes multidimensionais, operações matemáticas complexas, cálculos entre outros conceitos da Engenharia. O módulo *Pandas* auxilia na manipulação e análise dos dados, nas manipulações nas tabelas e na estrutura dos dados.

Um dos módulos incorporados na biblioteca *Scipy* e que, no escopo em questão, serve para trabalhar em um sistema de detecção automática de obstáculos é o *Scikit-Learn*. Na figura **X.X** é apresentada a página inicial deste módulo.

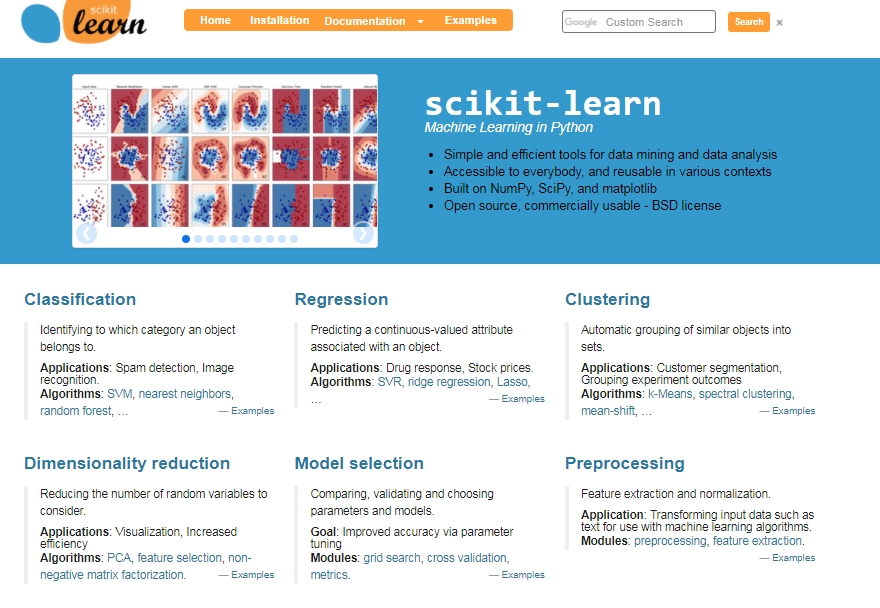


Figura 1X.X LEGENDA AQUI

O *Scikit-Learn,* rico em sua documentação,contém toda a base de informação para a realização de algoritmos de classificação, regressão, redução da dimensionalidade, pré-processamento e outros processos importantes para os desenvolvedores que querem trabalhar com algoritmos de aprendizado de máquina.

2.2 – Linguagem C#

Inicialmente alcunhada de *Cool* por Anders Hejlsberg e por outros integrantes da equipe formada pela Microsoft, a linguagem foi apresentada e renomeada nos anos 2000 no evento *Professional Developers Conference* (PDC) que reúne os desenvolvedores de software da Microsoft para apresentar as novas tecnologias do momento.

O C# é uma linguagem orientada a objeto que faz parte da plataforma .NET. Contém uma tipagem forte e é amplamente utilizada quando se trata da construção de softwares, sistemas operacionais e jogos em geral.

A IDE oficial para a elaboração de códigos em C# é a plataforma da própria Microsoft chamada Visual Studio. Mesmo possuindo suporte nativo para outras linguagens como Visual Basic (VB), C e C++ a ferramenta é completa para os desenvolvedores em C# que buscam criar serviços Desktop como também para a Web.

O serviço Web elaborado no projeto foi implementado com a linguagem C# através do Visual Studio Community 2017. O serviço foi implementado utilizando o framework Web API. Apesar de existir versões com uma maior quantidade de recursos como o Ultimate, optamos pelo Communitypor possuir todos os recursos necessários e gratuitos para o desenvolvimento da aplicação.

2.3 – Linguagem Java

Alcunhada de Oak no início de seu desenvolvimento, Java foi criado na mesma época que a linguagem Python. Desenvolvido pela Sun Microsystems e pertencido atualmente à Oracle, a linguagem nasceu com a finalidade de ser executada em diversas plataformas de hardwares diferentes [6]. Para obter esse funcionamento distribuído em sistemas operacionais distintos, a compilação de códigos em Java gera arquivos objetos chamados de *byte-codes*, e estes podem ser executados por meio de interpretadores desenvolvidos para cada tipo de plataforma [6].

Lançada nos anos 90, Java ainda tem uma presença massiva no universo dos softwares desktops e mobiles. Segundo IEE Spectrum [7] no ano de 2017 o Java ocupa a terceira posição nas linguagens mais utilizados do Mundo.

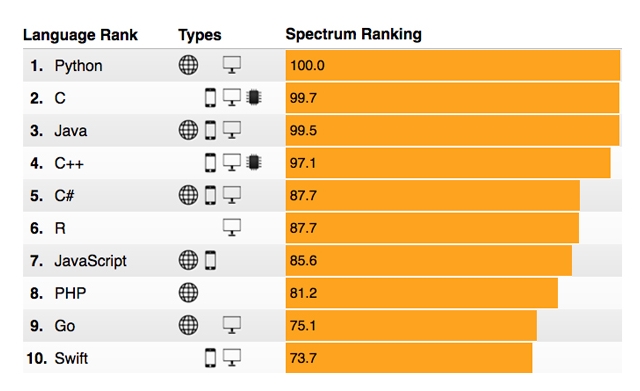


Figura 2X.X LEGENDA AQUI

No caminho oposto da linguagem Python, a linguagem da Oracle não é *open-source*. O fato de não ser aberta ao público para desenvolver em conjunto, não impede que as bibliotecas se mantenham atualizadas e ricas em conteúdo. Java, por ter sido desenvolvida a um tempo considerável, possui bibliotecas muito bem consolidadas e uma grande quantidade de material disponível na internet para estudo.

No universo Mobile a situação não é diferente. Os aplicativos dos dispositivos móveis com o sistema operacional Android podem ser desenvolvidos utilizando outras linguagens, porém o recomendado pela própria Google, criadora do sistema Android, é em Java através da IDE oficial Android Studio. De acordo com a IDC Quarterly Mobile Phone Tracker [8], a presença de aparelhos Android no mercado é aproximadamente de 85% nos primeiros meses de 2017. A figura **X.x** ilustra a diferença da presença e a sua evolução dos diferentes sistemas mobile no mercado mundial.

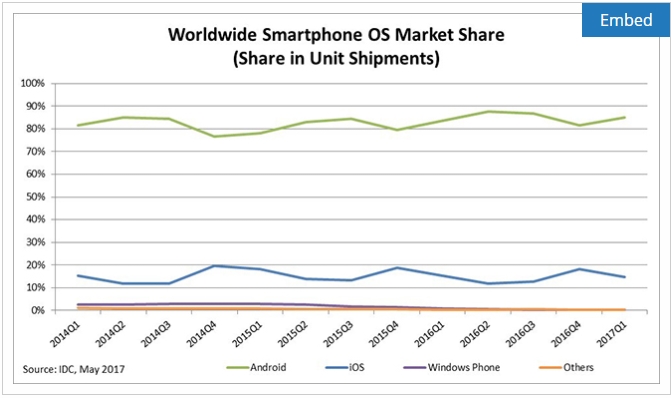


Figura 3X.X LEGENDA AQUI

Em virtude da maior presença de dispositivos Android na sociedade, o aplicativo construído para a captura de dados do sensor acelerômetro e das coordenadas do GPS foi para smartphones Android, através da utilização da IDE oficial da Google em conjunto com a linguagem recomendada.

2.4 – Web API

A maioria das aplicações existentes utilizam serviços conectados à Internet. O motivo pode variar desde uma simples conexão com um banco de dados para consultar uma certa informação, ou mesmo pelo fato da aplicação como um todo estar hospedada na nuvem.

Hoje, consegue-se acessar um serviço qualquer por diversas plataformas e sistemas operacionais distintos. O uso de smartphones, por exemplo, vem crescendo exponencialmente e ultrapassando o próprio computador como ferramenta para acessar uma aplicação [9]. Com base nesse conhecimento, é razoável que os desenvolvedores de software levem em consideração estruturar um projeto para que o próprio funcione tanto em sistemas Windows, Mac, Linux e seus derivados.

Um dos serviços lançados pela Microsoft focados na questão de desenvolvimento para linguagens de programação e para sistemas operacionais diferentes é o framework Web API. Fazendo uso deste framework em uma aplicação, não haverá distinção se o cliente está acessando através de um computador Linux, Mac OS X, dispositivo móvel ou até mesmo de aplicações desenvolvidas em outras linguagens de programação. O Web API independe do que está por dentro da “caixa preta” do cliente. São estabelecidas algumas regras para a conexão e para a configuração dos serviços HTTP construídos, permitindo que todas as funcionalidades do software funcionem.

Para realizar a conexão entre cliente e servidor são utilizados basicamente, como protocolo de comunicação, o SOAP e/ou REST. Na tabela **X.x** é apresentada uma comparação entre os dois tipos de comunicação:

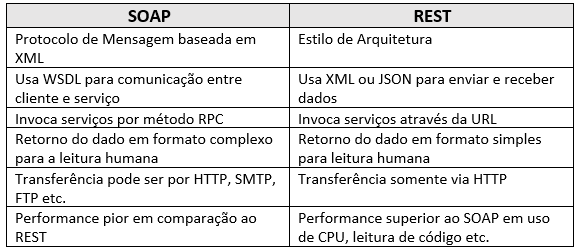


Figura 4X.X LEGENDA AQUI

O protocolo de transferência mais utilizado por desenvolvedores de softwares que necessitam realizar requisições Web em seus produtos é o HTTP. Este protocolo possui métodos bem definidos, sendo os principais listados na Tabela **X.x**.

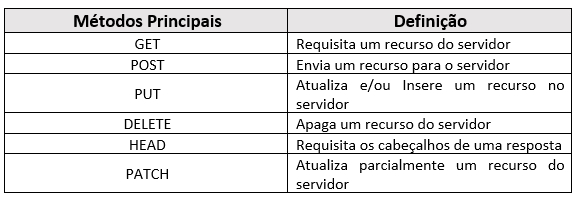


Figura 5X.X LEGENDA AQUI

Para exemplo de demonstração, será analisado um breve funcionamento de um Web Service público que, ao informar o nome do país, são retornados nomes de cidades do mesmo. O hostdesta aplicação aberta a qualquer usuário possui a seguinte URL: www.webservicex.net/globalweather.asmx/GetCitiesByCountry

Em requisições GET, os parâmetros devem ser passados na construção do próprio link do serviço. Para separar o que é o direcionamento para um determinado método da sua aplicação e quais são os parâmetros, é necessário o uso do símbolo “?”. Ademais, os parâmetros devem ser organizados em pares de chave-valor.

Neste Web Service, o nome do parâmetro que será preenchido com o nome do país é “CountryName”. Portanto, a construção final da URL escolhendo “United States” como base ficará no seguinte formato:

www.webservicex.net/globalweather.asmx/GetCitiesByCountry?CountryName=United States

Os dados podem ser retornados via XML ou JSON. Neste caso, o desenvolvedor deste serviço público web optou pela tecnologia XML. A seguir está uma parte do retorno do envio da URL acima.



Figura 6X.X LEGENDA AQUI

Aparentemente o XML retornou uma codificação fácil de ser interpretada, mas não é comum de acontecer. Neste caso ocorreu devido a passagem de somente 1 parâmetro e que o mesmo possui uma estrutura simples.

A respeito do método POST que é amplamente utilizado, as informações não são mais visíveis e expressas pelo próprio link no navegador. Os parâmetros serão inseridos no *Body* (corpo) da requisição. A seguir consta como deve ser a construção do *Header* e *Body* da solicitação por um determinado recurso que se encontra no servidor.



Figura 7X.X LEGENDA AQUI

As informações contidas no retângulo de cor vermelha correspondem a construção do *Header* (cabeçalho) e a parte de fora do *Body*. Diferentemente da requisição GET que o próprio usuário utiliza a própria URL do serviço para informar os parâmetros, no POST o interessado deverá preencher a solicitação do recurso através de uma aplicação.

2.5 – Sensor Acelerômetro

O acelerômetro é um dispositivo eletrônico utilizado para medir as acelerações nos eixos x, y e z em relação a gravidade do ambiente. Todo o deslocamento de massa implica induzir uma velocidade, direção e sentido. A partir da variação da velocidade obtida, tem-se como resultado a aceleração independentemente da direção aplicada.

Uma questão relevante é saber interpretar os dados deste tipo de sensor e por isso é preciso conhecer seu sistema referencial. O posicionamento dos eixos do sensor acelerômetro em dispositivos móveis é apresentado na Figura **X.X**.

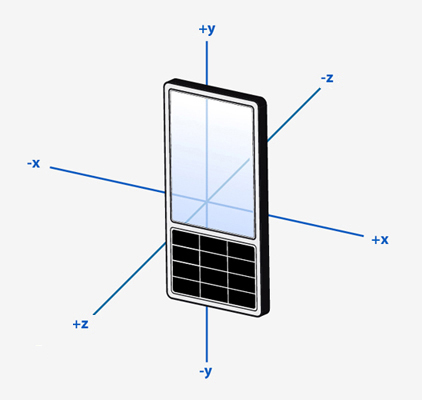


Figura 8X.X LEGENDA AQUI

Este sensor, presente na maioria dos smartphones comercializados, é amplamente utilizado em sistemas de posicionamento, inclinação e de vibração.

O seu princípio baseia-se na Segunda Lei de Newton que relaciona a força resultante aplicada em um corpo com a massa e a velocidade. Conhecendo o valor da força imposta no sensor, podemos obter a variação da velocidade em relação ao tempo.

Existem diversas formas de construir um acelerômetro. Cada forma parte de um princípio físico distinto para obter o mesmo resultado que é a medição da aceleração. Alguns tipos de acelerômetros mais utilizados são: Capacitivo, Piezoelétrico e Piezoresistivo [1].

O acelerômetro capacitivo baseia-se no movimento que uma massa de prova, suspensa por molas, faz entre duas placas fixas ao substrato do sensor. Quando há uma aceleração sendo aplicada, as capacitâncias entre a massa de prova e cada uma das placas variam antagonicamente. Isto é, enquanto uma capacitância possui uma variação positiva, a outra terá uma perda em valor absoluto. Na Figura **X.X** apresenta uma ilustração esquemática de como se comporta este tipo de sensor. Nesta figura, quando há uma aceleração sendo aplicada, a massa de prova fica mais próxima da placa fixa que consta o capacitor C2. Como a distância entre esses dois objetos será menor, o capacitor C2 terá uma maior capacitância. A variação desta é proporcional à aceleração.

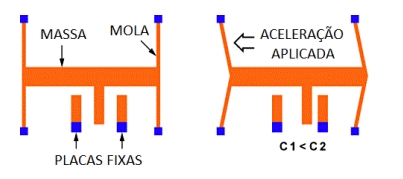


Figura 9X.X LEGENDA AQUI

Este tipo de acelerômetro também está presente nos smartphones modernos. Como apresentado nas Figuras **X.X e X.X** , o circuito é composto de um elemento móvel, o *Seismic Mass*, que vibra sobre a base (*Housing*) em caso de movimentação do dispositivo móvel. Em cada vibração, as aletas do *Seismic Mass* se aproximam dos capacitores, alterando a capacitância. Os circuitos medem o diferencial da tensão e retornam o mesmo para a saída do acelerômetro. Cada valor é proporcional a movimentação das aletas.

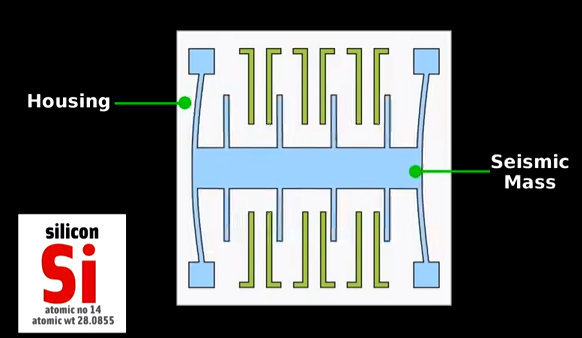


Figura 10X.X LEGENDA AQUI

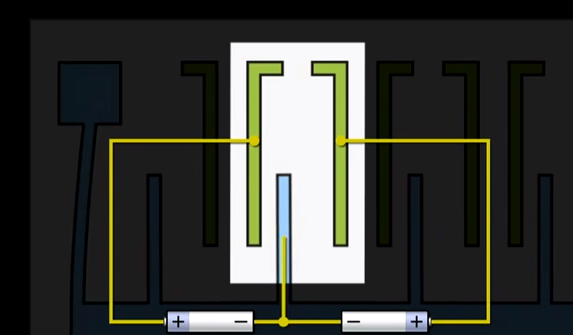


Figura 11X.X LEGENDA AQUI

O acelerômetro do tipo Piezoelétrico remete a capacidade de alguns cristais gerarem tensão elétrica por resposta a uma pressão mecânica [xx]. Normalmente, há uma massa presa a um cristal piezoelétrico e quando há uma aceleração no sistema (smartphone) a massa presa ao cristal acaba gerando uma deformação no cristal e este deslocamento gera um sinal elétrico proporcional à força aplicada. Análogo ao Piezoelétrico, o Piezoresistivo é relacionado à variação da resistência do circuito ao ocorrer uma deformação do cristal através da massa acoplada.

No projeto, esse dispositivo eletrônico será utilizado para capturar irregularidades nas estradas pavimentadas através dos deslocamentos no espaço impostos no veículo e consequentemente no smartphone também.

2.6 – Normalização

Os dados capturados pelo sensor acelerômetro assim como em outros sensores possuem ruídos vinculados. Uma técnica bastante utilizada para as pessoas que trabalham com dados sujeitos à ruídos é aplicação de técnicas de normalização.

A normalização consiste em suavizar os ruídos presentes no sinal que está sendo capturado ou em uma base de dados já existente. Para amenizar esses dados inconvenientes, aplica-se a seguinte fórmula.

A variável de saída “Z” é o dado normalizado, “x” o valor que se deseja suavizar o ruído, “” como a média dos dados e “” o desvio padrão do sinal.

Existem alguns tipos de normalização e os dois principais são: normalização completa e por janela. A completa consiste em calcular a média e desvio padrão de todo o conjunto de dados e aplicá-los na fórmula acima para encontrar o vetor de saída “Z”. Este método é apropriado quando o sinal tende a ser uniforme com a adição de um ruído ao mesmo.

A normalização por janela também conhecida como *Sliding Window* contempla um curto intervalo de amostras, o que reflete em valores de média e desvio padrão somente para o escopo observado. O sinal de saída irá conter picos mais destacados devido a remoção ou suavização dos ruídos existentes. A amplitude dos picos dependerá do valor ponderado do desvio padrão em relação à média do sinal.

**Inserir referência bibliográfica**

2.7 – Trabalhos Relacionados

Este trabalho dá continuidade à pesquisa de Laudelino [2] que fornece uma base de dados importante para se ter um entendimento melhor de como o órgão federal brasileiro realiza atualmente a classificação das ruas pavimentadas no Brasil. Ele cunhou o termo “coeficiente lunar” como uma variável para quantificar ou melhor, para qualificar as vias pavimentadas. Além disso, explicita todos os itens que são considerados pelo DNIT, desde fenda, afundamento, até por tipos específicos de cada item como por exemplo “Trinca do tipo Couro de Jacaré”. Além de abordá-los, também é informada uma tabela contendo todos os detalhes oficiais a respeito das classificações desses defeitos nas vias. Ademais, é informado como é feito a identificação das condições dos pavimentos citando o Deflectógrafo Lacroix, veículos específicos utilizados e o processamento de imagem embarcado. A monografia é recomendada para os leitores que querem aprofundar mais para entender o processo de mapeamento das ruas pavimentadas no Brasil.

O desenvolvimento do projeto Lunar também se baseou no artigo científico publicado pelos autores Christian Koch e Ioannis Brilakis [3]. Este artigo apresenta resultados dos estudos envolvendo a análise de imagens de vias pavimentadas com o auxílio de uma câmera de alta velocidade embarcada em um veículo. O conhecimento foi extraído através de 120 imagens sendo 50 para treino dos algoritmos e 70 para teste a fim de aperfeiçoar os códigos elaborados.

O estudo foi motivado ao se depararem que, no ano de 2009, o Departamento de Transporte dos Estados Unidos da América utilizava métodos manuais para detectar as fissuras, buracos, remendos e outros tipos de obstáculos nas vias pavimentadas das regiões dos EUA e do Canadá. A coleta e o armazenamento dos dados eram através do uso de câmeras acopladas em veículos do departamento, porém a sua análise era feita manualmente através de técnicos especializados. Os resultados eram influenciados pela subjetividade e pela experiência dos avaliadores.

Desde então, esforços foram dedicados tanto pelos autores como também por outros laboratórios na área de detecção automática de buracos e outros obstáculos nas vias pavimentadas. A detecção, segundo Koch e Brilakis [3], é dividida nas seguintes etapas: reconstrução do pavimento em 3D, emprego de digitalização laser, computer *stereo-vision* e utilização de acelerômetros. Enquanto as duas primeiras etapas sofrem com alto custo de equipamentos e de esforço computacional, as demais são destinadas para levantamento de mais informações para o aumento de precisão e da confiabilidade dos dados.

A respeito do uso de acelerômetros, descobriram que abordagens baseadas exclusivamente em vibrações poderiam fornecer resultados errados, informando buracos por engano ou falsos negativos. Buracos localizados no centro de uma pista muitas vezes não são atingidos pelas rodas dos carros e, portanto, não podem ser reconhecidos pelo sensor. No entanto, estes obstáculos devem ser capturados visto que são defeitos de pavimentos e particularmente danosos aos motociclistas. Caso contrário, o mapeamento dos obstáculos das estradas não vai possuir uma precisão e uma confiabilidade significante.

O trabalho de Koch e Brilakis se baseia nas propriedades geométricas das regiões defeituosas dos formatos dos buracos utilizando técnicas de regressão e de processamento de imagem. Foram identificadas três características principais referentes à aparência visual dos *potholes*. São elas:

1. Um buraco inclui uma ou mais sombras que são mais escuras do que a área circundante.
2. A forma de um buraco é aproximadamente elíptica.
3. A textura da superfície dentro de um buraco é mais áspera e granulosa em relação a uma superfície intacta.

Com base nessas informações, o modelo proposto foi dividido na segmentação da imagem, extração da forma e comparação com superfícies próximas na mesma imagem para obter um diagnóstico final. Como resultado, obtiveram uma acurácia de 85.9%, precisão de 81.6% e sensibilidade de 86.1%.

Já Adarkwa e Attoh-Okine [4] chamam atenção sobre o uso da fatorização Tensorial para realizar a classificação dos obstáculos nos pavimentos. O artigo considerou somente as rachaduras longitudinais e transversais devido serem de um fenômeno de defeitos nas vias de maior ocorrência. Os danos causados por estes obstáculos atingem a cada de US$ 10 bilhões anuais no EUA o que explica o porquê do Departamento de Transporte dos Estados Unidos da América enfatizam o monitoramento e a manutenção das calçadas.

O estudo desenvolvido foi realizado em Matlab utilizando o Matlab Tensor Toolbox 2.5. Há quatro etapas importantes a serem citadas:

1. Pré-processamento e formação de *Training Tensor*
2. Pré-processamento de dados de teste
3. Decomposição do valor Singular da Ordem Superior do *Training Tensor*
4. Classificação

O conjunto é composto de 1830 imagens a serem pré processadas, dividindo-se em grupo com e sem defeitos.

O pré-processamento envolve a conversão das imagens do formato RGB em *grayscale* e convertida para 100 pixels tanto de largura como altura. Para realizar o treinamento e o teste de validação, o conjunto de imagens é então dividido em dois grupos principais, são eles: rachaduras transversais e longitudinais. Em seguida, imagens de cada classe são empilhadas, separadamente, uma após a outra para formar o *Training Tensor*.

Se, após a aplicação do algoritmo, a imagem for classificada contendo fissuras longitudinais, o nome do arquivo recebia a letra “l” e caso contrário a letra “t”. No entanto, destaca-se que o nível de precisão do algoritmo mudou com as mudanças no número de imagens nos conjuntos de treinamento pois é evidente que a precisão não depende apenas do número de imagens no conjunto de treinamento, mas também da variabilidade interna entre as classes.

Nitsche, et al. [5] realizaram uma abordagem diferente sobre como processar as informações das vias pavimentadas. Eles utilizaram como base para a captura, os sensores acelerômetros e sensores que medem a velocidade e o comportamento das rodas dos carros de passeio.

Foi criado, pelos próprios autores, um índice que analisa o nível de rugosidades das vias estudadas. Apelidado de *weighted longitudinal profile* (*wLP*), conseguiram detectar os fenômenos de rugosidade com precisão utilizando diferentes algoritmos como *SVM*, *MLP* e *Random Florest*.

Neste artigo, a elaboração das fórmulas para obtenção dos resultados e os testes aplicados foram baseados em exatamente três tipos de fenômenos diferentes de rugosidades nas vias pavimentadas. São elas: Rugosidade Genérica, Rugosidade Pseudo-Periódica e Irregularidade Pontuais. Na Figura **X.x** pode-se notar a diferença de comportamento de cada variação de obstáculo que foi incorporado ao campo de pesquisa.

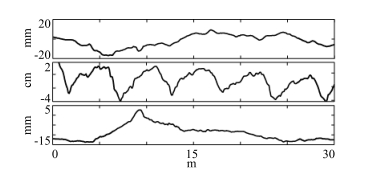


Figura 12X.X LEGENDA AQUI

**Colocar na Legenda: Gráficos Rugosidade Genérica, Pseudo- Períodica e Pontuais.**

Um outro intuito deste trabalho foi também apresentar um comparativo da eficácia de cada modelo de aprendizado de máquina aplicado, explicando as vantagens e as desvantagens de cada modelo estudado.

O método proposto pelos autores permite que o monitoramento da rede rodoviária seja alcançado por carros de passageiros convencionais e não em veículos especializados como as outras pesquisas indicam e se baseiam.

Além do parâmetro *wLP* existe um outro indicador para a medição da rugosidade nas estradas. Criado no Brasil em 1982 através da parceria do Banco Mundial com equipes de pesquisa Brasileiros e de outros países, o Índice de Rugosidade Internacional (IRI) foi elaborado com o intuito de unificar os parâmetros utilizados nos diferentes países para determinar a rugosidade nas vias.

Sayers e Karamihas[10], em Setembro de 1998, realizaram um estudo chamado “*The Little Book of Profiling*” que fornece todas as informações básicas para o entendimento das medições e interpretações dos perfis rodoviários visto a tamanha relevância dos indicadores como o IRI para a realização do mapeamento de condições asfálticas. Iniciando desde a história da elaboração deste parâmetro até uma análise detalhada sobre o seu funcionamento, o IRI é apresentado como o primeiro índice de perfil amplamente utilizado, onde o método de análise é planejado para o trabalho com diferentes tipos de perfiladores tornando-o um índice reproduzível através de outros meios de cálculo.

Ainda nesse estudo elaborado em 1998, o IRI é reconhecido como o único parâmetro de alcance global que sumariza a qualidade asfáltica, podendo ser utilizado para o cálculo de custo operacional, cargas dinâmicas inseridas nas rodas (isto é, danos nas estradas causados por veículos pesados ou por outros motivos que colocam em risco os veículos que trafegam na região) e condições gerais da superfície. Na Figura **X.X** apresenta os intervalos de IRI para diferentes tipos de rodovia e velocidade.

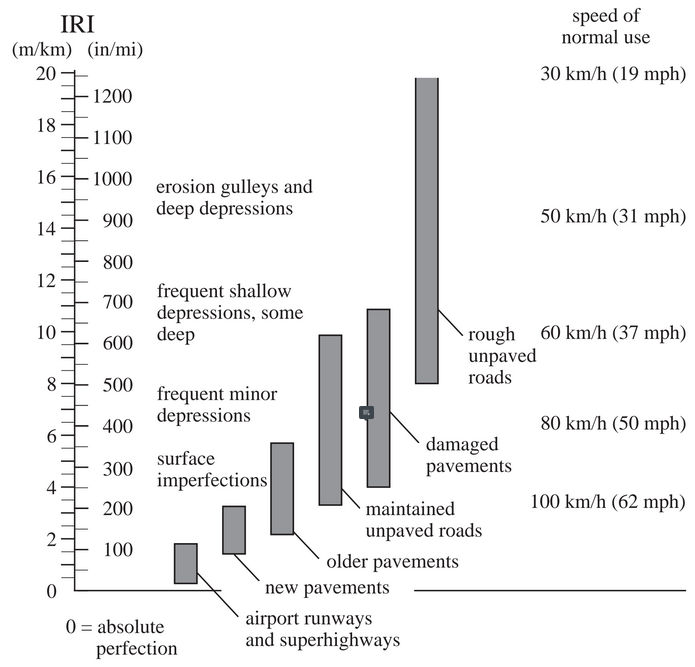


Figura 13X.X LEGENDA AQUI

O modelo que o algoritmo do IRI utiliza como base é chamado de “*quarter-car”*. Esta construção teórica é exatamente o que o seu nome implica. O esquemático está apresentado na Figura X.X abaixo.

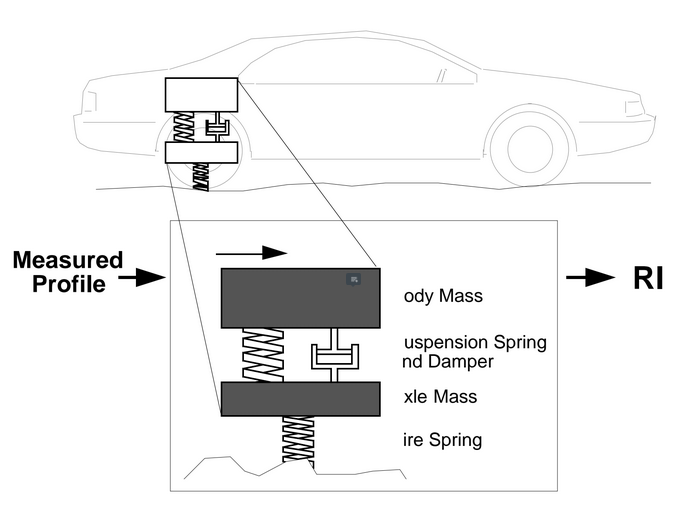


Figura 14X.X LEGENDA AQUI

O filtro “*quarter-car*” calcula a simulação da deflexão da suspensão de um sistema mecânico. Mesmo tratando-se de uma simulação, os valores coincidem com as respostas físicas tanto para veículos de passeios e caminhões ao trafegarem por tipos diferentes de vias pavimentadas.

Uma outra característica importante do Índice Internacional de Rugosidade apontada por Sayers e Karamihas é o fato dele ser linearmente proporcional à rugosidade, o que significa que se ocorrer uma alteração dos valores de um perfil medido, então o IRI alterará exatamente na mesma porcentagem.

Vargas [11], em 2010, também elabora um estudo que apresenta os aspectos e as considerações importantes para a determinação das condições das estradas com base no cálculo do IRI. Em termos mais simples, Vargas descreve este parâmetro da seguinte forma: “O IRI é um modelo matemático que calcula o acúmulo de força aplicada na suspensão de um veículo de passeio percorrendo uma estrada em um intervalo de tempo”.

Vargas expõe a relação do índice com as consequências das vias asfálticas. As variações significativas do IRI estão diretamente relacionadas com o maior custo de manutenção para a administradora da pista como também para os motoristas que frequentam essa região.

O intervalo para as estradas pavimentadas é entre 0 a 12 m/km (metros acumulados por quilômetro percorrido) aonde o valor 0 representar uma superfície perfeitamente uniforme e 12 uma estrada intransitável. O perfil de um percurso recém-inaugurado possui um valor próximo a zero, visto que igual a zero é extremamente difícil no ponto de vista construtivo. Uma vez calculado o índice para uma determinada região, a rugosidade do pavimento tende a diminuir em função da circulação de veículos.

O autor cita alguns métodos para realizar a medição das variáveis que o IRI exige para poder ser quantificado. Estes métodos diferem na precisão e na velocidade dos resultados. Alguns citados foram o Perfilógrafo, Perfilômetro Inercial a Laser e Nível e mira Topográfica.

Como resultado do estudo, Vargas obtém que uma diminuição de 10% no valor do IRI de uma estrada aumenta a eficiência do combustível em aproximadamente 1,91 quilômetros por litro. Além disso, um acréscimo de 50% na diminuição de rugosidade equivale na elevação de 15% na vida útil do pavimento.

Diferente do meio proposto pelo projeto Lunar, os meios utilizados pelo Vargas e assim como os demais autores são necessários uso de equipamentos extras para realizar a medição dos índices para fornecerem valores quantitativos a respeito da qualidade asfáltica ou até mesmo de veículos especializados como é o caso do Adarkwa e Attoh-Okine [4] e Koch e Brilakis [3].

**Capítulo 3**

# Tecnologias na Cloud

Para uma melhor compreensão de toda a infraestrutura implementada no projeto, é necessária uma breve explicação sobre os serviços na nuvem que foram utilizados e também dos proprietários desses serviços de modo geral. Na seção 3.1, é exposto a empresa de cloud computing escolhida e os seus principais serviços para o escopo da aplicação desenvolvida. Adiante, nas seções 3.2 e 3.3, são abordados com mais detalhes os serviços utilizados.

3.1 – Amazon Web Service

Quando o assunto é empresas de cloud computing, três grandes nomes se destacam. São eles: Amazon Web Services (AWS), Microsoft Azure e Google Cloud Platform. Tratam-se de tecnologia recente mas, ao mesmo tempo, já possuem milhares de serviços robustos na nuvem.

A mais antiga dessas empresas multinacionais mencionadas acima é a AWS. Lançada oficialmente em 2016 **[X.X]**, esta empresa é um provedor de serviços online para websites e também para aplicações cliente servidor baseado nas nuvens. Pode ser acessada via HTTP, protocolo REST ou SOAP. Oferece serviços de banco de dados, análise de dados, redes, dispositivos móveis, ferramentas de gerenciamento, IoT, armazenamento, aluguel de servidor entre outros. **[https://aws.amazon.com/pt/products/]**

Esta provedora de serviço possui um plano gratuito (*free-tier*) que possui duração de 1 ano e que contempla benefícios mais do que suficientes para executar aplicações testes. Dentre alguns dos serviços fornecidos gratuitamente **[https://aws.amazon.com/pt/free/#legal]** estão: 750 horas por mês para alugar máquinas na Cloud, 5 GB de armazenamento de objetos, 750 horas por mês de uso de banco de dados e 1.000.000 de solicitações ao Amazon Simple Queue Service (SQS).

A maioria dos serviços presentes na AWS possuem um plano de pagamento chamado pay-as-you-go, isto é, pagamento conforme o uso. O custo será dado pelos serviços individuais que estão sendo utilizados, pelo instante que os utilizar, e sem necessidade de contratos de usufruir por um determinado período de tempo.

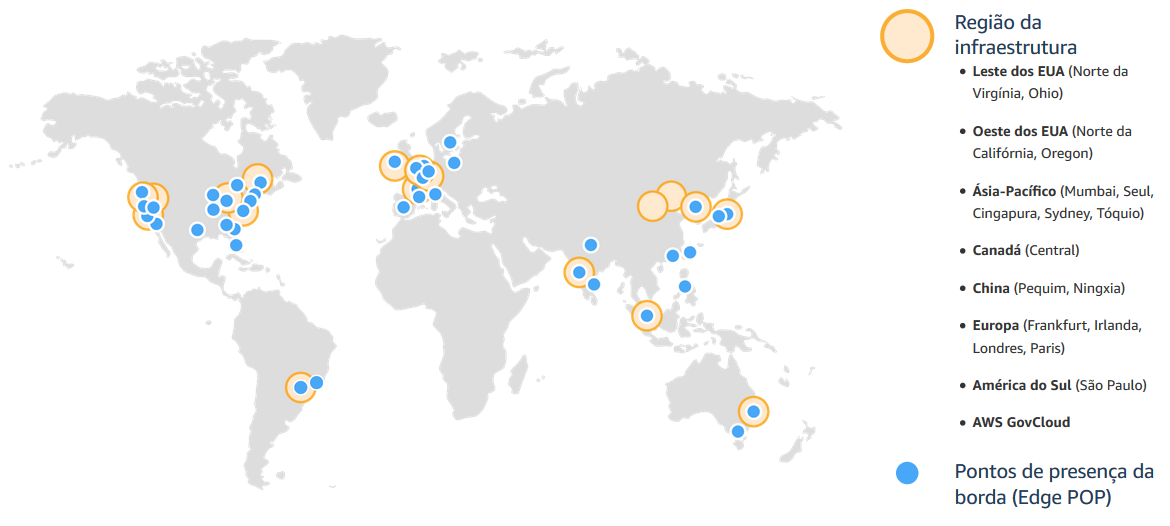


Figura 15 X.X LEGENDA AQUI

A Amazon opera datacenters espalhados ao redor do mundo. Os locais são compostos por regiões e zonas de disponibilidade. Cada região é uma área geográfica separada, totalmente independente, e contém vários locais isolados conhecidos como zonas de disponibilidade. O fato das regiões serem completamente isoladas é o que proporciona a maior tolerância a falhas e estabilidade possível. Além desses benefícios, uma outra qualidade não menos importante é a possibilidade de distribuir uma aplicação entre várias zonas de disponibilidade, o que a torna escalável o quanto financeiramente aceitar.

Ao fazer referência ao termo “escalabilidade”, um dos serviços fornecidos pela AWS que o projeto Lunar irá utilizar é o Amazon Simple Queue Service (SQS). Este serviço que, permite escalar um processo, será explicado com maior gama de detalhes na seção 3.3. Um outro recurso a ser utilizado pelo Lunar é o Amazon Elastic Compute Cloud (EC2). Esse, permite que os usuários aluguem computadores virtuais para rodar os processos de seu interesse. Os computadores virtuais, denominados de instâncias, podem ser criados, lançados, parados e terminados conforme necessário, pagando por hora pelas máquinas que estão em uso.

3.2 – Amazon Elastic Compute Cloud (EC2)

O EC2 [**https://aws.amazon.com/pt/ec2/pricing/**] é a parte central da plataforma da AWS. É nele que os usuários irão alugar instâncias (máquinas virtuais) para a hospedagem dos processos. Para atender todos os perfis de usuário, a plataforma tem disponível três tipos distintos de instâncias e centenas de configurações diferentes para as máquinas em cada grupo.

As instâncias são divididas em Spot, Reservada e Dedicada. Cada uma delas possui públicos alvos diferentes. Iniciando pela Spot, as instâncias Spot [**https://aws.amazon.com/pt/ec2/spot/**] do Amazon EC2 são capacidade computacional extra disponível na Nuvem AWS. Elas possuem um valor até 90% mais baixo em relação às Dedicadas. Em contrapartida, as máquinas podem ser interrompidas a qualquer momento com uma notificação prévia de dois minutos, caso o EC2 estiver precisando desse poder de processamento. Como consequência, as instâncias Spot são recomendadas para aplicativos que têm períodos de início e de término flexíveis, aplicativos que são viáveis somente por preços computacionais muito baixos e usuários com necessidades computacionais pontuais.

A respeito das instâncias reservadas **[https://docs.aws.amazon.com/pt\_br/AWSEC2/latest/UserGuide/ec2-reserved-instances.html]**, elas podem possuir um valor até 75% mais significativo em relação às Dedicadas. Ao contrário das máquinas do tipo Spot, as reservadas não podem ser terminadas quando o EC2 precisar do poder computacional alocado pelas mesmas. Elas não instâncias físicas, mas sim um desconto aplicado ao uso de instância dedicada. O usuário pode reservar determinados atributos de uma instância física e operar durante um determinado prazo de tempo. Após a expiração do tempo, caso queira continuar utilizando, o preço a ser pago será de uma instância dedicada.

As instâncias Dedicadas referem-se à uma instância física totalmente dedicada a um único cliente. Recomendas para aplicações que não podem ser interrompidas, as Dedicadas são as mais caras, mas que trazem uma maior confiabilidade no quesito de estabilidade. As cobranças das máquinas, assim como na Spot e na Reservada, são por segundo de uso. Vale ressaltar que há uma taxa extra de 2 (dois) dólares por hora caso tenha uma instância dedicada em uma das regiões disponíveis pela AWS.

As máquinas, além de possuírem diferentes tipos de alugueis, também possuem famílias, aonde cada família é voltada para um uso específico de uso. Existem 5 famílias atualmente [**https://aws.amazon.com/pt/ec2/instance-types/**]. São elas: Uso Geral, Otimizadas para Computação, Otimizadas para memória, Computação acelerada, Otimizada para armazenamento. As instâncias, da mesma família, também podem possuir tamanhos diferentes que variam de nano, micro, small, medium, large, xlarge, 2xlarge etc.

Abaixo segue uma imagem retirada do próprio site da AWS referente a informações das máquinas do grupo “Otimizadas para memória”.



Figura 16 X.X LEGENDA AQUI

A respeito do processo de levantamento de uma máquina pelo serviço EC2, ao criar um cadastro em um plano gratuito (free-tier) ou pago no console da AWS, e entrar no painel do serviço EC2, teremos acesso à algumas informações como as instâncias que estão rodando, os volumes extras adquiridos e outros dados interessantes como mostra a figura X.X abaixo.

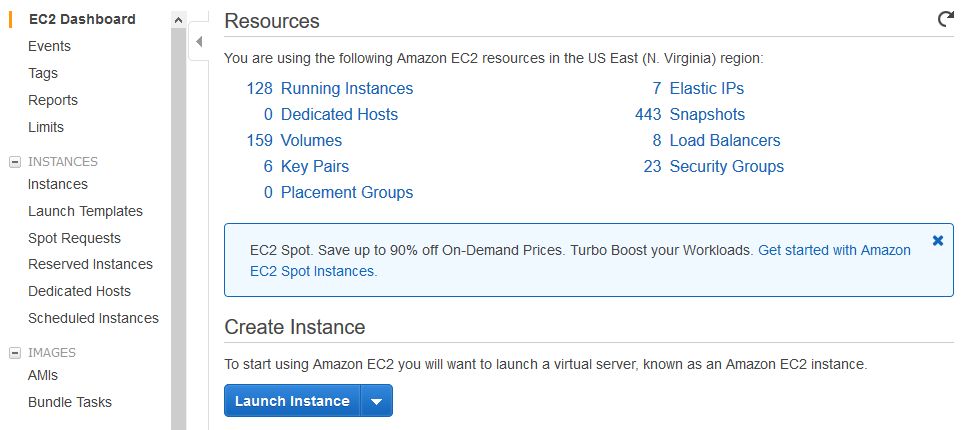


Figura 17 X.X LEGENDA AQUI

Ao clicar em “Launch Instance”, nas próximas janelas será possível manipular o sistema operacional da máquina, o volume root, configurações de acesso, bloquear ou permitir portas de conexão entre outros fatores. Além disso, um serviço que AWS disponibiliza para acelerar a produtividade é o lançamento de uma instância já com uma pré configuração definida, isto é, softwares instalados e já configurados. Para isso, durante as etapas para iniciar uma nova máquina, haverá uma opção para informar se deseja inserir uma Amazon Machine Image (AMI). Existem tanto imagens gratuitas quanto pagas. É aconselhável ir nas seções “AWS Marketplace” e “Community AMIs” e procurar por palavras chaves contendo nome dos softwares que pretende utilizar após o aluguel da máquina.

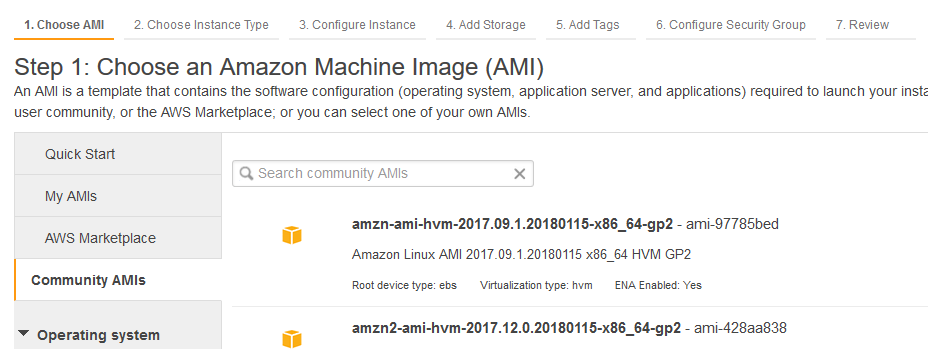


Figura 18 X.X LEGENDA AQUI

3.3 – Amazon Simple Queue Service (SQS)

O serviço SQS da AWS oferece hospedagem de filas altamente escalável para armazenamento de mensagens, com tamanho máximo de 256 kb cada, na medida em que trafegam entre aplicativos ou micro serviços. Movimenta dados entre os componentes distribuídos do aplicativo e ajuda a desacoplá-los. Este é um papel fundamental para permitir escalonar serviços menores que compõe um projeto grande e de alta complexidade.

O serviço possui dois tipos de filas. A fila FIFO (*First-In-First-Out*) e a Standard (Padrão). Há duas diferenças primordiais entres estes tipos que são: Ordenamento da Entrega e a Duplicidade de Processamento.

Na FIFO, a ordem que as mensagens são enviadas e recebidas é preservada. Ademais, garante que uma mensagem seja entregue uma única vez, permanecendo disponível até que o consumidor a processe e a exclua. Não existem duplicações introduzidas neste tipo de fila.

A respeito do tipo Standard, as mensagens poderão ser entregues em uma ordem diferente da qual elas foram enviadas. Segundo a questão de duplicidade, podem existir um número infinito de cópias de mensagens dentro da mesma fila, atribuindo a responsabilidade do usuário de criar uma lógica dentro de sua aplicação para realizar o tratamento dessas mensagens repetidas.

O custo para a criação de fila é gratuito. O valor é cobrado por número de solicitações (primeiro 1 milhão é gratuito) enviadas e também por gigabytes de transferência de dados caso seja para uma instância EC2 de uma região distinta da qual a fila foi criada. O tipo *first-in-first-out*, por possuir os benefícios de sem duplicidade e a entrega de forma ordenada, possui o dobro do custo em relação a fila padrão.

**Capítulo 4**

# Projeto Lunar

O projeto foi implementado visando torná-lo escalável. As principais etapas foram divididas em pequenos processos distintos para facilitar a manutenção, além de diminuir a complexidade do código como um todo. A Figura **X.x** mostra como está estruturado a aplicação Lunar no quesito do envio de dados pelo usuário até a inserção ao banco de dados.

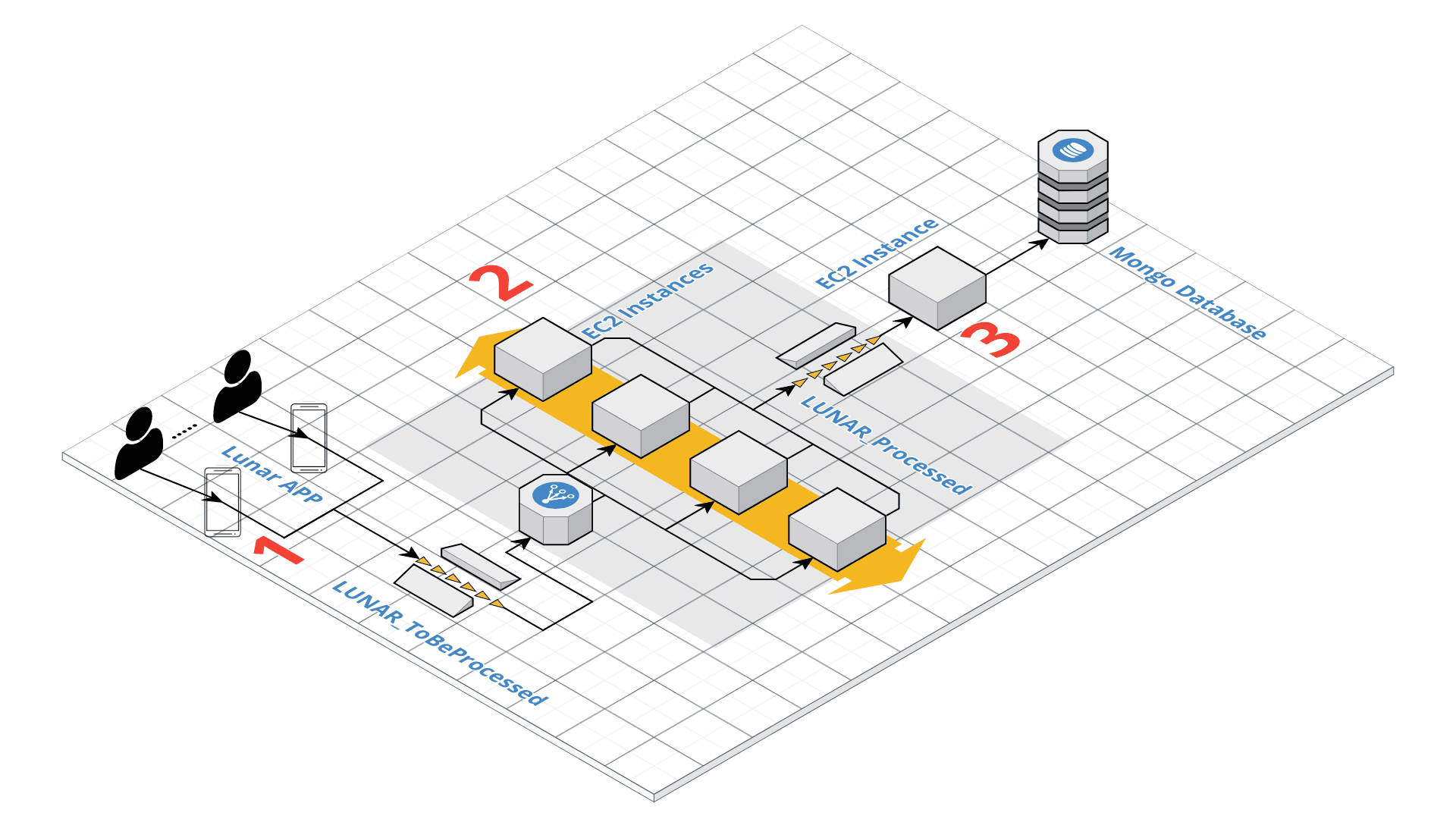


Figura 19 Inserir Legenda aqui

A estruturação do projeto é fracionada em três principais etapas. A primeira delas refere-se à comunicação do aplicativo android Lunar com uma *standard queue* chamada “LUNAR\_ToBeProcessed”. A transmissão de dados entre essas duas ferramentas está apresentada na figura a seguir.

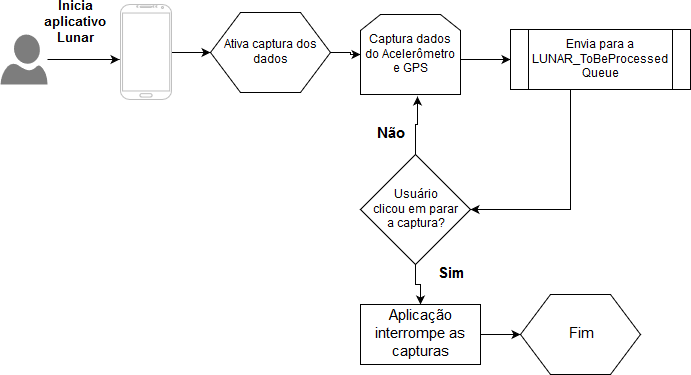


Figura 20 Inserir legenda aqui

O processo relacionado a captura dos dados dos sensores ocorre a cada 30 milissegundos, isto é, a cada 50 centímetros percorridos pelo veículo caso a velocidade instantânea seja de aproximadamente 60 km/h. As informações cruas são enviadas em formato JSON para a fila “LUNAR\_ToBeProcessed”.

A segunda etapa é o estágio responsável pelo processamento tanto no quesito de extrair mais informações a partir das coordenadas do GPS, como também para realizar as validações com o objetivo de remover os dados caracterizados como inválidos. Como esta parte do projeto depende exclusivamente da quantidade de dados armazenada em uma fila da AWS, ela foi estruturada de modo que tornasse viável a escalabilidade, podendo, portanto, executar a quantidade de processos da segunda etapa o quanto for necessário para atender a demanda da entrada.

A terceira etapa é hospedada em uma máquina distinta daquela usufruída pelos processos responsáveis pelo tratamento e aquisição de mais dados. É o segmento do projeto aonde ocorre a leitura das mensagens hospedadas na fila “LUNAR\_Processed” e o envio das mesmas para uma coleção do banco de dados não relacional MongoDb.

Após a informação percorrer as três etapas, ela estará disponível para ser requisitada pelo usuário através de uma requisição HTTP do tipo GET. A construção de uma Web API foi desenvolvida para facilitar a comunicação, principalmente a requisição, com os registros presentes na base de dados.

4.1 Aquisição dos Dados

Para facilitar o entendimento de como foi realizado a aquisição dos dados, este capítulo foi dividido em três etapas. Na seção 4.1.1 é apresentado quais dados são capturados e o motivo de terem sido escolhidos. Já na seção 4.1.2, é esclarecido o método de captura implementado e os aparatos utilizados para torna-lo possível. Na 4.1.3 é realizada uma análise diagnóstica com base nos gráficos gerados com base nos dados adquiridos. Por fim, na última seção deste capítulo, é informado como é realizada a exposição das informações para usuários comuns como também para agentes de instituições governamentais.

4.1.1 – Tipos de Dados

Anterior a etapa da aquisição dos dados, é entendermos quais tipos de dados são necessários para que torne plausível o objetivo principal e, se possível, também o encaminhamento das metas a serem cumpridas a longo prazo que foram descritas na seção de “Trabalhos Futuros”. Após ter todos os objetivos bem definidos, torna-se fácil a tarefa de identificar quais dados devem ser capturados.

A respeito da detecção de obstáculos nas vias pavimentadas, foi elaborada uma análise prévia para verificar se o sensor Acelerômetro presente em dispositivos mobile, seria sensível o suficiente para registrar as diferenças sutis das oscilações provocadas pela via no automóvel. Observada a viabilidade, a conclusão foi de capturar os dados gerados pelo sensor Acelerômetro.

Em relação ao mapeamento dos dados, é viável se e somente se forem capturadas informações relacionadas ao GPS. Para isso, no código desenvolvido para o aplicativo Android, foi desenvolvida a lógica para salvar as informações referentes a Latitude e a Longitude em tempo real.

Além dos dados já mencionados, outros, não menos importantes, são capturados como: a inclinação em graus do dispositivo no momento da aquisição; tempo em milissegundos desde que foi iniciada a aplicação; data e hora corrente. A informação referente à inclinação é para possibilitar novos estudos diferentes daqueles realizados neste projeto. Já sobre o cronômetro e hora que está marcando no aparelho servirá como auxílio para a realização do processamento e exposição dos dados.

4.1.2 – Método de Captura

A construção de uma aplicação teste para smartphone Android foi o primeiro passo para tornar possível a captura dos dados mencionados na seção 4.1.1. Antes de realizar a aquisição, foi estudado a necessidade de incluir mais equipamentos tecnológicos para poder realizar o treinamento do modelo de machine learning.

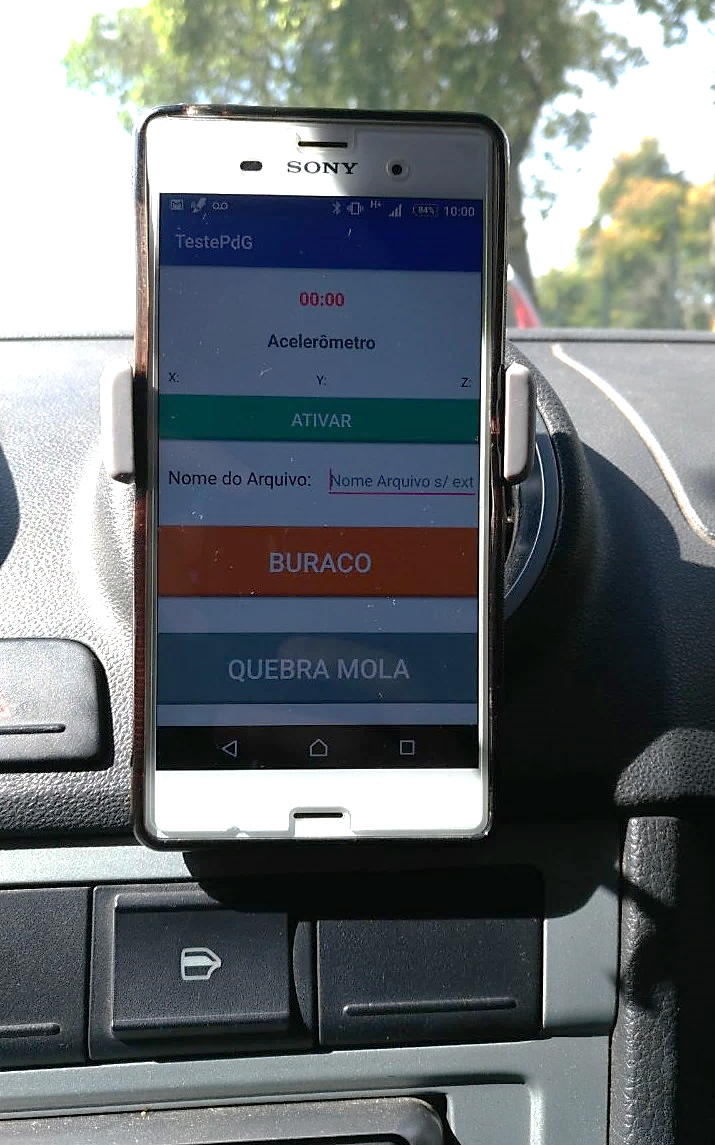


Figura 21 INSERIR LEGENDA AQUI

A aplicação teste além de contar com o botão “Ativar”, também possui os botões de “BURACO” e “QUEBRA MOLA” para registrar os eventos dos obstáculos de forma manual. Um suporte de celular foi utilizado para diminuir a variação da inclinação do dispositivo e também para manter um padrão a ser seguido na aquisição dos dados.

Um outro auxílio muito importante para realizar a validação dos eventos registrados foi a utilização de uma câmera no capô do automóvel para produzir a gravação de todo o trajeto percorrido.



Figura 22 INSERIR LEGENDA AQUI

A colaboração conjunta de uma câmera filmando o trajeto e os eventos serem inseridos de forma manual, possibilitou uma análise visual para checar a viabilidade de desenvolver um algoritmo de aprendizagem de máquina que detecte, de forma automatizada, os obstáculos nas vias pavimentadas.

A respeito da base de dados utilizada para os testes, o automóvel percorreu vias de diferentes pontos da cidade e do estado do Rio de Janeiro. Foram capturados quebra-molas de diferentes tipos, buracos de baixa, média e alta profundidade.

4.1.3 – Análise Diagnóstica

Após realizada a captura dos dados, inicia-se a elaboração de uma análise diagnóstica a partir dos gráficos gerados pelos valores dos três eixos do sensor acelerômetro em cada instante do tempo. É realizado uma busca para identificar as relações de causa e consequência percebidas ao longo das amostras para entender o porquê de ter ocorrido picos e depressões em certos momentos da coleta.

Anterior a geração dos gráficos, foi realizada uma normalização em todo o conjunto das amostras utilizadas com o intuito de reduzir o ruído e de deixar mais nítido as consequências geradas após o automóvel sofrer o impacto de um certo tipo de obstáculo.

As imagens a seguir apresentam o comportamento do sensor acelerômetro no eixo x, y e z em um trajeto percorrido na Cidade Universitária do Rio de Janeiro.

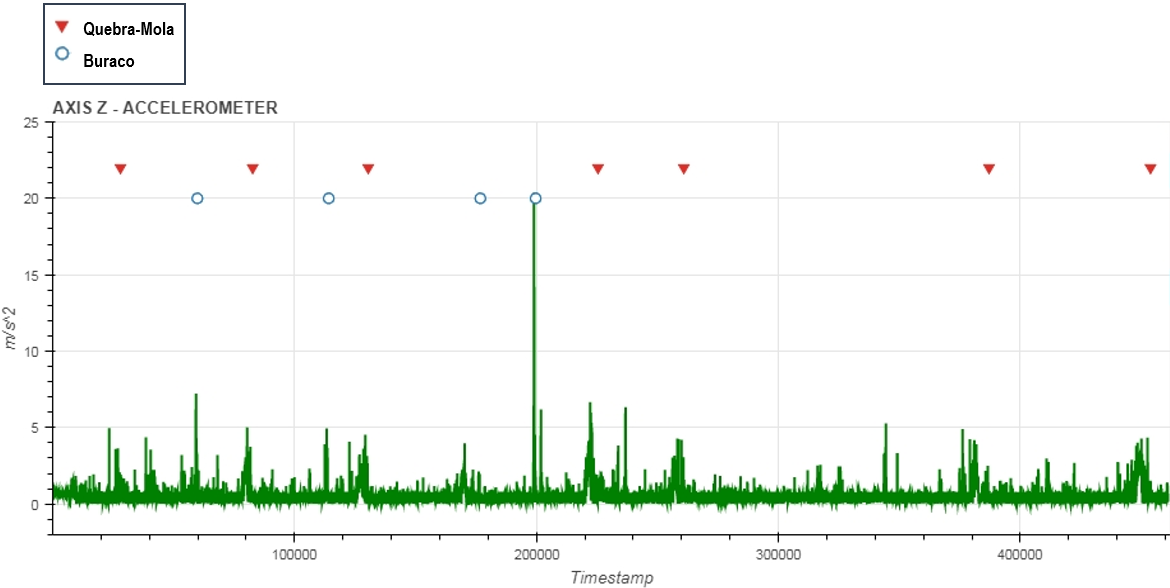


Figura 23 INSERIR LEGENDA AQUI

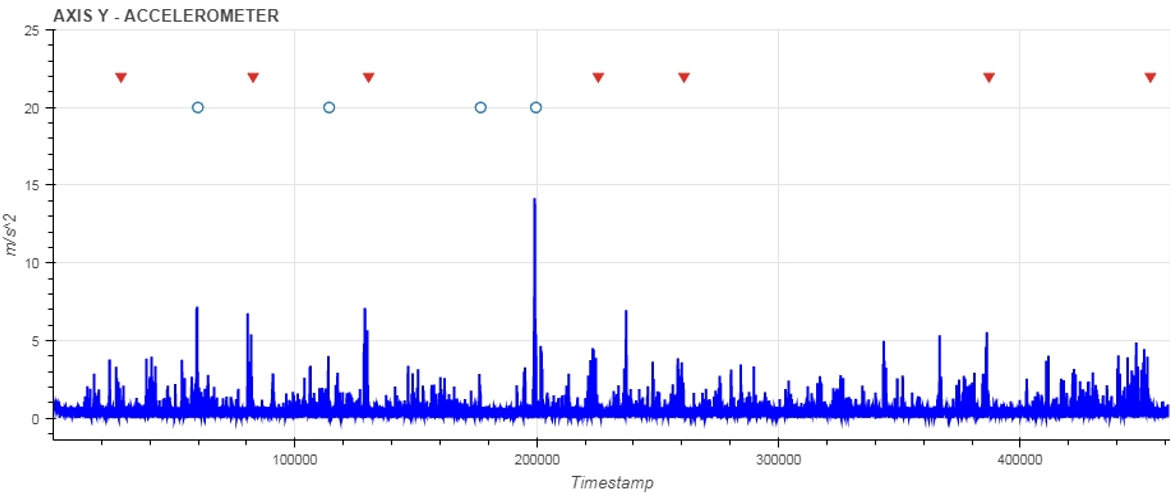


Figura 24 INSERIR LEGENDA AQUI

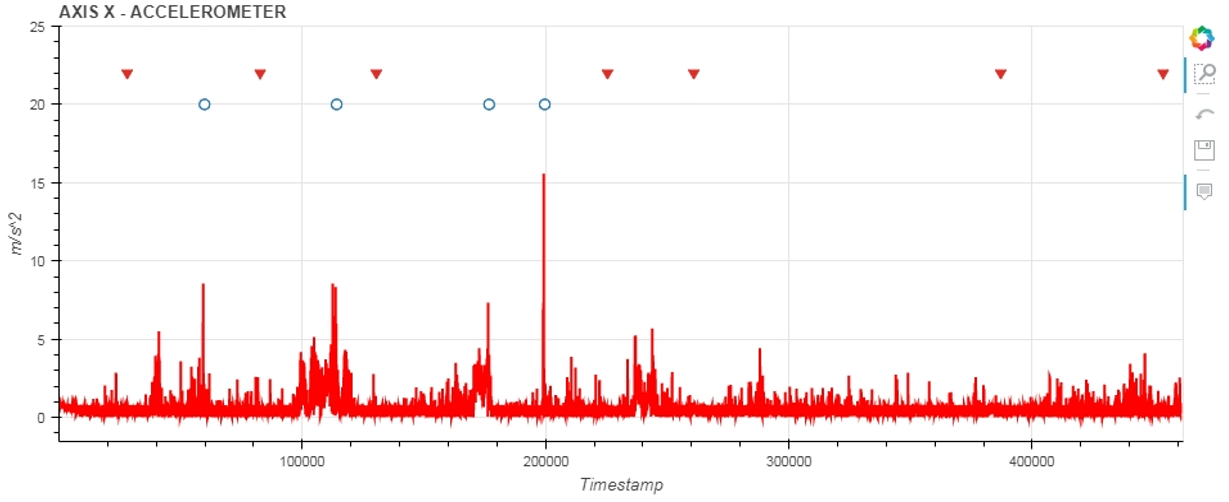


Figura 25 INSERIR LEGENDA AQUI

As consequências geradas pelos eventos se tornam nítidas no conjunto de dados normalizados. Os picos com as intensidades diferentes estão diretamente relacionados com a força do impacto que cada evento foi capaz de gerar no automóvel. Isto vale também para as reações distintas entre cada eixo do sensor. Dependendo de como o automóvel passou por cima de um obstáculo, as assinaturas em cada eixo tendem a apresentar valores diferentes. Os picos menores, mas que não podem ser ignorados, foram praticados por imperfeições existentes nas vias pavimentadas.

O algoritmo de aprendizagem de máquina deve ser capaz de saber diferençar picos ocasionados por eventos do interesse e aqueles por ondulações da pista. Uma das técnicas possíveis é a inserção de um *threshold* para realizar essa distinção.

A Figura **X.x** apresenta um caso que é utilizado a imagem fornecida pela câmera presa ao capô do carro e a assinatura provocada pelo evento destacado em um dos gráficos gerados.



Figura 26 INSERIR LEGENDA AQUI

O ícone triangular significa um evento de quebra-mola, já o círculo de um buraco. O pico de maior intensidade foi gerado pelo buraco presente na imagem da esquerda na Figura **X.x**. Entre o primeiro quebra-mola e buraco, foi constado um pico significativo. Este auge em questão está relacionado com uma faixa existente assim que o veículo entra em uma rotatória, momentos depois da imagem capturada e apresentada acima. A faixa contém desgastes, desníveis e esses detalhes foram capturados com sucesso pelo sensor presente no dispositivo móvel utilizado.

4.2 Processamento dos Dados

**Em desenvolvimento...**

4.3 Exposição dos Dados

Dados capturados, analisados e processados. Todas as etapas necessárias para poder enfim, compartilhar com os usuários comuns e agentes de instituições governamentais informações relacionadas à qualidade das vias pavimentadas de uma determinada região.

O intermédio entre o usuário e os registros presentes no banco de dados se dá através uma Web API. O serviço web está disponibilizado em uma instância da AWS, podendo ser usufruída através do emprego de parâmetros na própria URL de consulta. O usuário poderá acessar os dados, em formato JSON, de acordo com o interesse. Os parâmetros disponíveis e que podem ser combinados são: output, latitude, longitude, city, state, numberofdays e limit.

O parâmetro “output” é do tipo inteiro e aceita 0, 1 ou 2 como valor. É o atributo responsável por apresentar informações referentes a nenhum evento detectado (0), eventos de buraco (1) e eventos de quebra-molas (2).

A “latitude” e “longitude”, que são do tipo double, permitem aos usuários a realizarem filtros em busca de registros de um local específico. Aliada as coordenadas do GPS, os parâmetros “city” e “state” permitem a possibilidade de aplicar filtros mais genéricos como cidade ou estado de modo geral.

Os campos “numberofdays” e “limit” são designados a fornecerem um limite quantitativo e de informações recentes aos usuários, permitindo assim uma maior customização e leque de possibilidades de obter os dados.

A Figura **X.x** apresenta um exemplo de uma consulta de acordo com os seguintes filtros: **xxx**

**FIGURA AQUI**

**Capítulo 5**

# Trabalhos Futuros

**Em desenvolvimento...**

**Capítulo 6**

# Conclusões

**Em desenvolvimento...**

# Bibliografia

[0] Relatório da World Health Organization aqui

[1] \_\_\_\_\_\_, Wikipedia, a enciclopédia livre. Disponível em < https://pt.wikipedia.org/wiki/Acelerômetro>. Acesso em 24 de Agosto de 2017.

[2] Lima, L.A.O. “Sistema de detecção das condições asfálticas das ruas através do uso passivo de smartphones”: Monografia do MBA em Tecnologia da Informação - Executivo, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2016.

[3] Koch, Christian; Brilakis, Ioanis. “Pothole detection in asphalt pavement images*”.* Advanced Engineering Informatics, v.25, n.3, pp.507-515, 2011. doi: 10.1016/j.aei.2011.01.002

[4] Adarkwa, Amanor; Attoh-Okine, Nii. “Pavement crack classification based on tensor factorization”. Construction and Bulding Materials, v.48, pp.853-857, 2013. doi: 10.1016/j.conbuildmat.2013.07.091

[5] Nitsche, Philippe; Stutz, Rainer; Kammer, Michael; Maurer, Peter. “Comparison of Machine Learning Methods for Evaluating Pavement Roughness Based on Vehicle Response”, Journal of Computing in Civil Engineering, v.28, n.4, 2014. doi: 10.10161/(ASCE)CP.1943-5487.0000285

[6] Fedeli, Peres, Polloni. “Introdução à ciência da computação”, 2ª Edição. São Paulo: Cengage Learning, 2010.

[7] Cass, Stephen. “The 2017 Top Programming Languages”. Disponível em < http://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>. Acesso em 29 de Agosto de 2017

[8] \_\_\_\_\_\_, “Smartphone OS Market Share, 2017 Q1”, http://www.idc.com/promo/smartphone-market-share/os, Acesso em 28 de Agosto de 2017)

45-52

[9] \_\_\_\_\_\_, “Zenith forecasts 75% of internet use will be mobile in 2017”. Disponível em < https://www.zenithmedia.com/mobile-forecasts-75-internet-use-will-mobile-2017/ >. Acesso em 28 de Agosto de 2017

[10] Sayers, Michael W.; Karamihas, Steven M. “The Little Book of Profiling. Basic Information about Measuring and Interpreting Road Profilers”, pp.45-52. The Regent of the University of Michigan. Setembro de 1998.

[11]