

CARLOS EDUARDO GRIVOL JÚNIOR

**Desenvolvimento de um sistema automático de detecção de irregularidades no
asfalto usando smartphones**

Trabalho de Conclusão de Curso
apresentado à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Bacharel em Engenharia

São Paulo
2017

CARLOS EDUARDO GRIVOL JÚNIOR

**Desenvolvimento de um sistema automático de detecção de irregularidades no
asfalto usando smartphones**

Trabalho de Conclusão de Curso
apresentado à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Bacharel em Engenharia

Área de Concentração: Engenharia
Elétrica com Ênfase em Sistemas
Eletrônicos

Orientador: Prof. Dr. Leopoldo Yoshioka

Co-orientadores: Eng. Rodrigo F.
Daguano e Eng. Arthur Aikawa

São Paulo

2017

Dedico este trabalho aos meus pais, Mara e Carlos

AGRADECIMENTOS

Ao Professor Leopoldo Yoshioka, pela orientação acadêmica e profissional de excelência que me forneceu ao longo da minha formação como Engenheiro. Seus estímulos foram essenciais para moldar o meu raciocínio crítico.

Aos meus co-orientadores, Rodrigo Daguano e Arthur Aikawa, pelo apoio nas decisões e troca de experiência.

Aos meus colegas Carlos Prete e Gabriel Crabbé, que me forneceram conselhos de qualidade durante o projeto.

À Ana Maria, da Biblioteca de Engenharia Elétrica da Poli-USP, pelo suporte na formatação da monografia seguindo os padrões ABNT e pela gentileza com os alunos de graduação.

À minha mãe, Mara, pelo seu suporte ao longo de toda esta fase como estudante e engenharia na Escola Politécnica da USP. Seu carinho foi fundamental para me incentivar a continuar trilhando este caminho.

Ao meu pai, Carlos, pela inspiração que me deu a seguir a carreira de Engenharia e pelo seu aconselhamento durante a minha formação. Minhas decisões foram sempre de qualidade graças à sua sabedoria.

Se eu vi mais longe, foi por estar sobre os
ombros de gigantes.

(Isaac Newton)

RESUMO

O Brasil detém cerca de 1 milhão e 700 mil km de rodovias, passando pelas esferas federal, municipal e estadual. Dependemos fortemente do transporte rodoviário e mesmo em grandes centros urbanos a infraestrutura não atende requisitos básicos de qualidade. Falta uma fiscalização constante que gere resultados reais e palpáveis para a população. Neste sentido, o projeto Cidade Lisa tem como objetivo desenvolver um sistema automático de detecção de irregularidades nas vias, permitindo assim a disponibilização de informações relevantes para os órgãos públicos competentes. O sistema de detecção automática de irregularidades no asfalto desenvolvido é uma solução tecnológica escalável que usa smartphones com redes neurais embarcadas como plataformas colaborativas na coleta de dados, usando principalmente acelerômetro e GPS. Com isto, procura-se agilizar o reparo, reduzir os custos envolvidos e satisfazer de forma geral os usuários.

Palavras-Chave: Engenharia Elétrica. Engenharia de Sistemas Eletrônicos. Sistemas Inteligentes de Transporte. Detecção automática. Smartphones. Sensores. Acelerômetro. GPS. Redes Neurais. Irregularidades no asfalto. Qualidade de pavimento.

ABSTRACT

Brazil has about 1 million and 700 thousand km of highways, passing through federal, municipal and state spheres. We depend heavily on road transport and even in large urban centers the infrastructure does not meet basic quality requirements. There is a lack of constant monitoring that produces real and tangible results for the population. In this sense, the Smooth City project aims to develop an automatic system to detect irregularities in the roads, thus allowing the availability of relevant information to the competent public agencies. The automatic asphalt irregularity detection system developed is a scalable technology solution that uses smartphones with embedded neural networks as collaborative platforms in data collection, mainly using accelerometer and GPS. With this, it is possible to expedite the repair, reduce the costs involved and satisfy users in general.

Keywords: Electrical engineering. Electronic Systems Engineering. Intelligent Transport Systems. Automatic detection. Smartphones. Sensors. Accelerometer. GPS. Neural networks. Asphalt irregularities. Quality of pavement.

LISTA DE ILUSTRAÇÕES

Figura 1 - Buraco em rua de São Paulo.....	16
Figura 2 - Operação Tapa-Buraco	18
Figura 3 - Patente da Google	19
Figura 4 - Street Bump	20
Figura 5 - Objetivos preliminares	22
Figura 6 - Eventuais soluções técnicas	23
Figura 7 - Árvore de objetivos	28
Figura 8 - Nível 0 da decomposição funcional	33
Figura 9 - Nível 1 da decomposição funcional	35
Figura 10 - Nível 2 da decomposição funcional	36
Figura 11 - Diagrama SWOT	43
Figura 12 - Análise de risco.....	44
Figura 13 - Cronograma detalhado do projeto no primeiro semestre	45
Figura 14 - Cronograma detalhado do projeto no segundo semestre.....	46
Figura 15 - Gastos com servidor	47
Figura 16 - Rede Neural	48
Figura 17 - Eixos do acelerômetro para Android.....	49
Figura 18 - Escala PASER	51
Figura 19 - Condição do pavimento e seu reparo	52
Figura 20 - Ambiente de desenvolvimento Android Studio	53
Figura 21 - Versionamento do app no GitHub.....	54
Figura 22 - Ícones do app Cidade Lisa	54
Figura 23 - Versão 1.0 do aplicativo Cidade Lisa.....	55
Figura 24 - Versão 2.0 do aplicativo Cidade Lisa.....	56
Figura 25 - Smartphone fixo ao veículo para testes.....	57
Figura 26 - Exemplo de log coletado.....	58
Figura 27 - Arquivos coletados no drive-test.....	58
Figura 28 - Arquivo gerado pelo script Java.....	59
Figura 29 - Treinamento da rede neural no MBP	60
Figura 30 - Matriz de confusão para a detecção de buracos	61
Figura 31 - Visualização da qualidade do pavimento em um mapa – Av. Corifeu	62
Figura 32 - Visualização da qualidade do pavimento em um mapa – USP	62
Figura 33 - Visualização dos buracos encontrados no trajeto dentro da USP	63
Figura 34 - Tela inicial do aplicativo	64
Figura 35 - Coleta de dados do trajeto	66
Figura 36 - Meus trajetos	67
Figura 37 - Apagar trajetos.....	68
Figura 38 - Visualização no mapa dos meus trajetos	69
Figura 39 - Escolha do tipo de mapa da cidade	70
Figura 40 - Mapa da cidade com buracos	70
Figura 41 - Mapa da cidade com qualidade	71
Figura 42 - Opção de treino da rede neural	72
Figura 43 - Painel de Controle do HostGator.com	73
Figura 44 - Gestão da base de dado no domínio	73
Figura 45 - Base de Dados do projeto Cidade Lisa	74
Figura 46 - Edição da estrutura da base MySQL	75
Figura 47 - Visualização dos dados armazenados na base.....	75
Figura 48 - Busca personalizada de dados armazenados na base	76
Figura 49 - Página PHP de conexão do app com a base de dados	77
Figura 50 - Gestão FTP de arquivos usando FileZilla	78

LISTA DE TABELAS

Tabela 1 - Requisitos de Engenharia	25
Tabela 2 - Compromissos de engenharia/marketing	26
Tabela 3 - Compromissos de engenharia	27
Tabela 4 - Geração de conceitos	29
Tabela 5 - Forças e fraquezas dos conceitos	29
Tabela 6 - Seleção de sensor	32
Tabela 7 - Seleção de unidade embarcada	32
Tabela 8 - Nível 0 da decomposição funcional	34
Tabela 9 - Módulo 1 da decomposição funcional.....	35
Tabela 10 - Módulo 2 da decomposição funcional.....	36
Tabela 11 - Sub-Módulo 1 da decomposição funcional	37
Tabela 12 - Sub-Módulo 2 da decomposição funcional	37
Tabela 13 - Sub-Módulo 3 da decomposição funcional	38
Tabela 14 - Sub-Módulo 4 da decomposição funcional	38
Tabela 15 - Sub-Módulo 5 da decomposição funcional	39
Tabela 16 - Gerenciamento do Projeto	40
Tabela 17- Orçamento	47

LISTA DE ABREVIATURAS E SIGLAS

API	Application Program Interface
GPS	Global Positioning System
ITS	Intelligent Transportation Systems
KM	Quilômetro
MBP	Multiple Back-Propagation
PASER	Pavement Surface Evaluation and Rating
PHP	Hypertext Processor
TCM	Tribunal de Contas do Município de São Paulo

SUMÁRIO

1 INTRODUÇÃO	15
1.1 IDENTIFICAÇÃO DAS NECESSIDADES	15
1.2 DECLARAÇÃO DAS NECESSIDADES	17
2 ESTADO DA ARTE	18
2.1 PESQUISA DE LEVANTAMENTO DA SITUAÇÃO	18
2.2 DECLARAÇÃO DO PROBLEMA	21
2.3 DECLARAÇÃO DOS OBJETIVOS DO PROJETO.....	21
2.3.1 Objetivos Preliminares.....	22
2.3.2 Eventuais Soluções Técnicas.....	22
3 MATERIAIS E MÉTODOS	24
3.1 ESPECIFICAÇÃO DOS REQUISITOS DE MARKETING	24
3.2 ESPECIFICAÇÃO DOS REQUISITOS DE ENGENHARIA.....	24
3.2.1 Matrizes de Compromisso	26
3.3 ÁRVORE DE OBJETIVOS.....	27
3.4 GERAÇÃO DE CONCEITOS E CRITÉRIOS PARA ESCOLHA	29
3.4.1 Geração de Conceitos.....	29
3.4.2 Seleção de Conceitos	31
3.5 DECOMPOSIÇÃO FUNCIONAL DETALHADA	33
3.5.1 Nível 0.....	33
3.5.2 Nível 1	34
3.5.3 Nível 2	36
3.5.4 Análise de Acoplamento e Coesão	39
3.6 GERENCIAMENTO DO PROJETO DETALHADO	40
3.7 AVALIAÇÃO DE RISCOS	43
3.8 CRONOGRAMA DETALHADO	44
3.9 ORÇAMENTO	47
4 IMPLEMENTAÇÃO.....	48
4.1 ESTRUTURA DA REDE NEURAL	48
4.2 ESCALA PASER	50
4.3 DESENVOLVIMENTO DO APLICATIVO	53
4.4 COLETA DE DADOS.....	55
4.5 TRATAMENTO DE DADOS	57
4.6 TREINAMENTO DA REDE NEURAL	59
4.7 CRITÉRIOS PARA DETECÇÃO DE BURACO	60
4.8 PÓS-PROCESSAMENTO DE DADOS	61
5 RESULTADOS FINAIS	64
5.1 DESENVOLVIMENTO DO APLICATIVO.....	64
5.1.1 Novo Trajeto	65
5.1.2 Meus Trajetos	66
5.1.3 Mapa da Cidade	69
5.1.4 Treino Neural.....	71
5.2 DESENVOLVIMENTO DA BASE DE DADOS.....	72
5.3 DESENVOLVIMENTO DO SERVIDOR.....	77
6 ANÁLISE.....	79
6.1 ANÁLISE DOS RESULTADOS DE MARKETING.....	79

6.2 ANÁLISE DOS RESULTADOS DE ENGENHARIA	82
7 CONCLUSÃO	84
8 TRABALHOS FUTUROS	85
REFERÊNCIAS.....	86
APÊNDICES	90
1 ATIVIDADE PRINCIPAL DO APLICATIVO ANDROID PARA A DETECÇÃO DE QUALIDADE E DE BURACOS	90
2 ATIVIDADE DE VISUALIZAÇÃO DO MAPA NO APLICATIVO ANDROID	103
3 CLASSE DE CONEXÃO DO APLICATIVO ANDROID AO SERVIDOR	108
4 PÁGINA PHP DE CONEXÃO À BASE DE DADOS	110
5 PÁGINA PHP PARA OBTER DADOS ARMAZENADOS NA BASE	111

1 INTRODUÇÃO

O projeto de formatura foi realizado individualmente sob a supervisão do Prof. Dr. Leopoldo Rideki Yoshioka e com a ajuda dos Engenheiros Rodrigo F. Daguano e Eng. Arthur Aikawa, formados pela Escola Politécnica da USP.

O título do projeto é Cidade Lisa e está situado dentro do tema de Sistemas Inteligentes de Transporte (ou mais conhecido como Intelligent Transportation Systems, ITS, do inglês).

1.1 IDENTIFICAÇÃO DAS NECESSIDADES

O Brasil detém cerca de 1 milhão e 700 mil km de rodovias, passando pelas esferas federal, municipal e estadual, porém menos de 14% delas são asfaltadas [15].

A gestão é realizada pelo poder público, que realiza investimentos insuficientes. A maior parte das cargas é transportada pelo sistema rodoviário, provocando um grande desgaste nas vias e reduzindo a qualidade da infraestrutura.

A malha rodoviária apresenta uma vascularização e uma densidade muito superiores aos outros sistemas de transporte e só não predomina na área amazônica, no qual o transporte por vias fluviais ainda apresenta grande importância [24].

São Paulo é o único estado que detém uma infraestrutura de transportes na qual as cidades do interior se encontram conectadas à capital por uma ampla rede, incluindo rodovias duplicadas, ferrovias e a hidrovia do Tietê. A cidade de São Paulo possui 17 mil km de vias.

Esta distribuição evidencia a importância econômica da região, que demanda por maior acessibilidade e melhor infraestrutura de transporte. Realizando uma observação direta do problema através de grandes meios de informação do país, é possível constatar uma situação cada vez mais alarmante.

No final de 2016, o Tribunal de Contas reprovou obras da Prefeitura de SP para tapar buracos [20]. Acima de 70% das obras precisam ser refeitas e não atendem a todos os requisitos de qualidade (Figura 1), como o índice de compactação abaixo do aceitável, qualidade da massa asfáltica inadequada ou o índice de deformação fora das especificações. O gasto por dia do serviço é de R\$ 500 mil, segundo o TCM.

Figura 1 - Buraco em rua de São Paulo



Fonte: Reprodução/TV Globo – Jornal Hoje

Fica clara a necessidade de melhoria na qualidade do material asfáltico e na sua adequada aplicação, a fim de garantir a correta destinação do dinheiro público. Além disso, há carência na gestão das licitações, na fiscalização dos serviços e de profissionais de infraestrutura de transportes [14].

Associado a tudo isso, a manutenção é na maioria dos casos inexistente ou demorada. Um levantamento na cidade de Marília, interior de São Paulo, aponta a existência de 84 mil buracos [18]. Isto corresponde a um buraco para cada três moradores da cidade.

O problema é ainda mais agravante em períodos chuvosos, deixando as capitais do país com uma quantidade alarmante de buracos [17] e pondo em risco a segurança dos cidadãos. Em Suzano por exemplo, um buraco na rua foi usado para atrair vítimas de assalto [16].

Finalmente, este problema de infraestrutura gera inúmeros problemas nos veículos que passam por buracos todos os dias. Os motoristas colecionam prejuízos, sendo que as principais peças afetadas são as buchas, pivores, pneus, suspensão, caixa

de direção e inclusive afetando o alinhamento do veículo. O prejuízo de manutenção do carro varia entre R\$800 e R\$2000 por conta das vias esburacadas [19].

1.2 DECLARAÇÃO DAS NECESSIDADES

De acordo com a Prefeitura de São Paulo, em 2015 foram utilizados R\$ 110 milhões para recapear 135,5 quilômetros de vias e tapar 294.419 buracos na cidade [14], que apresenta uma malha viária de mais de 17 mil quilômetros de vias pavimentadas. Simultaneamente, o número de solicitações de reparos em ruas e avenidas esburacadas cresceu 21% (de 54.074 para 65.459 solicitações).

Um sistema automático de detecção de irregularidades poderá garantir o reparo eficiente da infraestrutura urbana, atendendo às necessidades da população que utiliza a malha diariamente. Para isto, o sistema deve ser de fácil instalação, de simples utilização, preciso, barato e satisfatório. Estas necessidades permitem que o sistema seja utilizado em larga escala e de forma frequente, atingindo assim os objetivos planejados com um nível satisfatório de qualidade.

2 ESTADO DA ARTE

2.1 PESQUISA DE LEVANTAMENTO DA SITUAÇÃO

Atualmente a prefeitura de São Paulo tem um programa chamado Tapa-Buraco [23] (Figura 2) que busca realizar a ponte de contato entre o governo e a população. O objetivo é permitir à população um meio para relatar o local de buracos e acompanhar a execução do serviço.

Figura 2 - Operação Tapa-Buraco

zelando pela cidade

Secretaria Municipal de Coordenação das Subprefeituras
Tapa-Buraco

INÍCIO
Serviços
Áreas de Risco
Ações
Capas
Cota Regular
Cota de Lixo
Ecopontos
Gratias
Limpeza de Bueiros
Limpeza de Córregos
Piso de Pedra
Pavimentação
Problemas
Pontes e Viadutos
Rescapeamento
Tapa-Buraco
Troncos e Passarelas Subterrâneas
Varrição de Ruas

FISCALIZE E AVISE

SAC
Faça sua solicitação

SECRETARIAS
Selecione

SUBPREFEITURAS
Selecione

Este site foi criado para você acompanhar a realização do serviço Tapa-Buracos na cidade. Veja a data limite para execução do serviço (cinco dias corridos após o recebimento da ordem de serviço) e confira.

Se o serviço não tiver sido feito no prazo, ligue para 0800-37-3717 ou clique aqui e informe a Ouvidoria do Município para que a empresa prestadora do serviço seja cobrada e, se for o caso, punida.

Serviço de Tapa-Buracos na sua rua
Para verificar se houve ou como está a realização dos serviços Tapa-Buracos existentes em uma rua ou avenida (logradouro), preencha com o nome ou CEP a ser pesquisado e selecione uma das opções da lista.

Logradouro: (etc.) (Não digitar R, Av, Tr, Al)
CEP:

Pesquisar

Para solicitar a realização do serviço Tapa-Buracos, clique aqui e utilize o SAC - Sistema de Atendimento ao Cidadão.

Serviço de Tapa-Buracos no seu bairro
Para saber sobre a realização do serviço Tapa-Buracos em seu bairro, selecione na lista de opções abaixo a Subprefeitura a que pertence.

Subprefeitura: Selecione uma Subprefeitura

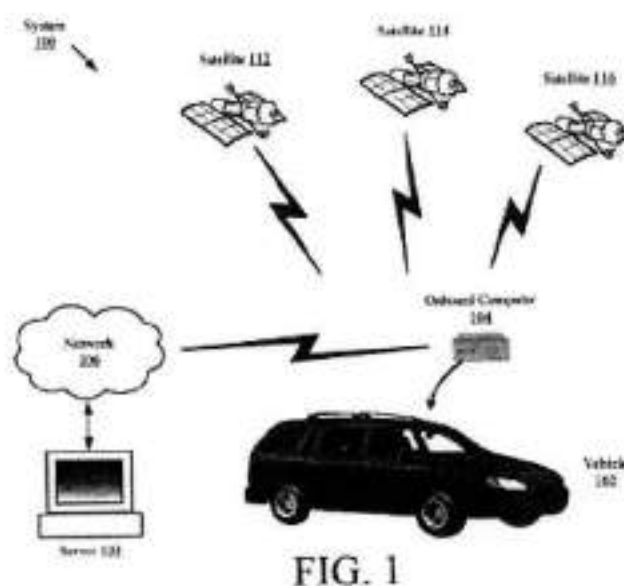
Se você identificar algum problema na realização do serviço Tapa-Buracos, clique aqui. Ajude a Prefeitura a melhorar a qualidade dos serviços prestados.

Fonte: http://www3.prefeitura.sp.gov.br/saffor_bueiros/FormsPublic/serv1TapaBuracos.aspx (2017)

O prazo de execução de obras é em teoria de 5 dias, porém o processo de relato requer o preenchimento de um longo formulário pelo cidadão, desestimulando a participação.

Além desta iniciativa, existem alguns projetos brasileiros locais como Buracos Monitor [11] e Buracos Uberlândia [12]. Neles, os moradores das cidades enviam as fotos do buraco e o local. O site então apresenta todos os buracos em um mapa feito com base na API (Application Program Interface) do Google Maps. No Estados Unidos também há iniciativas locais semelhantes, como por exemplo o Potholes in Ohio [21].

Figura 3 - Patente da Google



Fonte: Business Insider [13]

A Google registrou em 2015 uma patente de um veículo com sensores embarcados para a detecção em tempo real [13] [22] (Figura 3). Ela usaria um acelerômetro vertical e o GPS para registrar automaticamente buracos encontrados na rua. Softwares especializados poderiam depois eliminar anomalias, cancelar ou duplicar relatos e gerar relatórios específicos para cada rua.

Figura 4 - Street Bump



Fonte: <http://www.bu.edu/systems/files/2014/12/StreetBumpGraphic-1024x512-2.jpg>

Outro projeto interessante na área é o Street Bump [25] (Figura 4), um aplicativo de smartphone que envia automaticamente dados obtidos pelo acelerômetro e GPS do aparelho a um servidor central que alerta usuários sobre locais com buracos onde não foi feita manutenção. Diferentemente da patente da Google, este aplicativo depende da interação do usuário para salvar um itinerário realizado. Ele é de uso restrito na cidade de Boston, EUA, e foi um aplicativo encomendado pela prefeitura da cidade.

Observa-se, portanto, uma série de limitações nas soluções existentes, como a restrição geográfica de aplicação, falta de automação no processo de detecção, limitação de uso para certas marcas e falta de visibilidade nos resultados, pontos que o projeto Cidade Lisa procura desenvolver melhor.

Finalmente, após uma extensa pesquisa bibliográfica, uma série de artigos relevantes com pesquisas na área foram encontrados. Entre eles, pode-se citar os seguintes:

- An Intelligent System to Detect, Avoid and Maintain Potholes: A Graph Theoretic Approach [1];
- Road Condition Detection Using Smartphone Sensors: A Survey [2];
- The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring [3];
- Time Pothole Tracking System Using Android Smart Phone [4];
- Pothole Detection System using Machine Learning on Android [5];

- Road Bump and Intensity Detection using Smartphone Sensors [6];
- Real Time Pothole Detection Using Android Smartphones with Accelerometers [7];
- Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones [8];
- Analyzing Driving and Road Events via Smartphone [9];
- A Real-Time Pothole Detection Approach for Intelligent Transportation System [10].

2.2 DECLARAÇÃO DO PROBLEMA

O Brasil é um país que depende fortemente do transporte rodoviário, porém a infraestrutura não atende requisitos básicos de qualidade. Falta uma fiscalização constante que gere resultados reais e palpáveis para a população.

Apesar de existirem certas iniciativas por parte do poder público, estas são lentas, dependem demais da proatividade de certos indivíduos e não é realizada de forma automática, o que gera lentidão nos processos, custos elevados e desgaste por parte de todos os envolvidos.

As iniciativas privadas existem, entretanto, sofrem uma série de limitações. A maior parte delas se encontra limitada a zonas geográficas específicas, sofrem de restrições por questões de propriedade intelectual, são dificilmente aplicáveis na realidade brasileira ou não conseguem causar o impacto necessário na sociedade.

2.3 DECLARAÇÃO DOS OBJETIVOS DO PROJETO

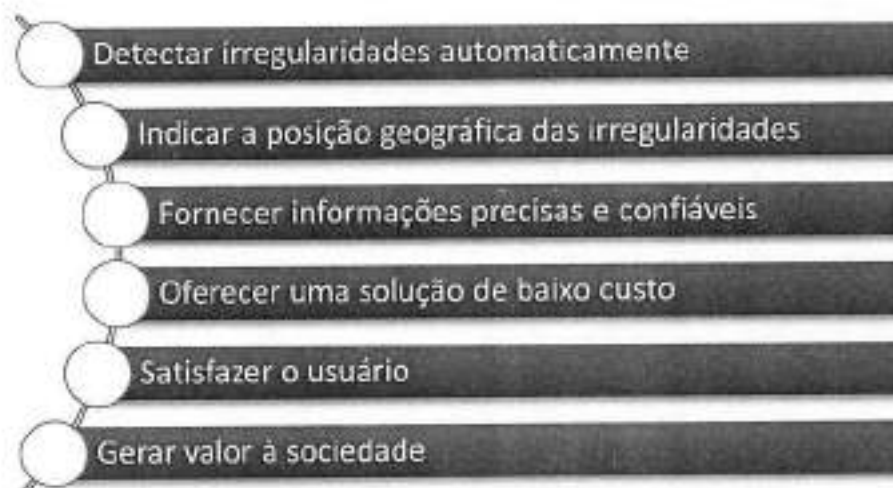
O objetivo deste projeto é desenvolver um sistema automático de detecção de irregularidades nas vias, permitindo assim a disponibilização na forma digital das informações relevantes para os órgãos públicos competentes realizarem o reparo adequado. Com isto, procura-se agilizar o processo e reduzir o custo das obras

através de uma fiscalização constante por parte da população com o uso do sistema desenvolvido pelo projeto Cidade Lisa.

2.3.1 Objetivos Preliminares

Para realizar o objetivo, podemos desdobrá-lo nos seguintes objetivos preliminares (Figura 5):

Figura 5 - Objetivos preliminares



Fonte: Autor

2.3.2 Eventuais Soluções Técnicas

Preliminarmente, a proposta é desenvolver um sistema embarcado para veículos que seja capaz de detectar irregularidades existentes nas vias, podendo assim enviar os dados técnicos relevantes à uma central.

Existem uma série de alternativas para a implementação deste sistema, que podem ser classificadas de acordo com alguns critérios (Figura 6).

Figura 6 - Eventuais soluções técnicas



Fonte: Autor

3 MATERIAIS E MÉTODOS

3.1 ESPECIFICAÇÃO DOS REQUISITOS DE MARKETING

Os requisitos de marketing do sistema são:

1. Peso leve
2. Tamanho pequeno
3. Poucas peças
4. Boa ergonomia
5. Respeito à privacidade
6. Baixo consumo de energia
7. Apresenta poucas etapas
8. Não distrai o motorista
9. Baixo custo
10. Boa confiabilidade
11. Boa precisão
12. Coleta eficiente de dados
13. Fácil envio de dados
14. Interface amigável

A justificativa detalhada e a relação com os requisitos de engenharia se encontram na próxima seção.

3.2 ESPECIFICAÇÃO DOS REQUISITOS DE ENGENHARIA

Para garantir o sucesso e a viabilidade técnica do projeto, o estabelecimento de requisitos detalhados de engenharia é fundamental. Com base na análise realizada até este ponto, encontram-se na Tabela 1 - **Requisitos de Engenharia** os requisitos de engenharia, suas relações com os requisitos de marketing levantados e as justificativas:

Tabela 1 - Requisitos de Engenharia

Requisitos de Marketing	Requisitos de Engenharia	Justificativa
10, 11, 12	A - O tempo de resposta do detector de irregularidades deve ser inferior a 3 segundos	O feedback ao usuário deve ser rápido o suficiente para garantir uma boa confiabilidade, tanto do ponto de vista técnico (posição geográfica) como pela satisfação do usuário com a obtenção de um resultado esperado.
11, 12	B - A posição geográfica da irregularidade detectada deve apresentar uma precisão de 30 metros.	De acordo com estudos recentes realizados com GPS de smartphones com tecnologia 3G [26], a precisão média obtida foi de 30 metros em zonas urbanas.
10, 12	C - A taxa de acerto na detecção deve ser superior à 75% para um usuário individual e superior à 85% na coleta coletiva.	De acordo com os quatro métodos apresentados no artigo de detecção de buracos em tempo real [8], podemos inferir que uma taxa individual de acerto de 80% é uma meta realística a ser obtida. Considerando o pós-processamento de dados, foi provado que uma taxa ainda maior pode ser obtida.
1, 2, 3, 6	D - O sistema deve drenar no máximo 800 mA.	Este consumo de bateria é aceitável do ponto de vista de dispositivos móveis (que são leves e pequenos), sendo que um modelo simples de bateria portátil apresenta em média 3400 mAh. Espera-se um trajeto médio de 2h por usuário em um dia. Este consumo moderado do sistema oferece uma boa margem de segurança se a bateria eventualmente deve ser compartilhada com outros sistemas.
1, 2	E - O peso total do sistema não deve ultrapassar 1kg e a sua maior dimensão não deve ultrapassar 30 cm.	O dispositivo deve ser portátil para permitir a mobilidade do usuário e sua fácil instalação. Um peso inferior a 1kg e um tamanho inferior a 30 cm permitem que uma pessoa de qualquer faixa etária possa carregar o dispositivo com apenas uma mão. Isto permite uma maior difusão no uso do sistema, melhorando os resultados globais de detecção.
3, 7	F - O sistema não deve necessitar de mais de 4 conexões entre diferentes partes do sistema.	Além de ser portátil, o dispositivo deve ser facilmente instalado por uma pessoa sem conhecimentos técnicos prévios. As conexões entre cabos ou etapas de ativação por exemplo não devem ser demasiadamente complexas a fim de permitir a difusão no uso do sistema, melhorando os resultados globais.
5	G - O dispositivo pedir permissão ao usuário antes de coletar seus dados privados.	O respeito à privacidade é imprescindível neste caso, pois estamos lidando com informações de alto grau de sensibilidade (como posição geográfica, por exemplo). Estudos confirmam que informações de sensores do gênero podem facilmente detectar inúmeros tipos de atividade de uma pessoa e potencialmente infringir seu direito à privacidade [2].
4	H - O design do dispositivo não deve degradar substancialmente a ergonomia do veículo.	O sistema deve ser menos invasivo quanto possível a fim de não mudar a rotina, causar desconforto, desestimular o seu uso ou ameaçar a segurança dos usuários.

6	I – O dispositivo deve ser completamente autônomo e não deve emitir sons ou imagens que distraiam o motorista durante a direção.	Durante o trajeto, o sistema deve influenciar o mínimo possível a sua rotina para não ameaçar a sua segurança nem condicionar os resultados obtidos.
9	J – O custo total fixo de desenvolvimento não deve exceder 500 reais e o custo marginal para novos usuários deve ser o mínimo possível.	Considerando exclusivamente o orçamento do autor para o projeto, o custo não deve exceder 500 reais. Ao final do projeto, quanto mais barato for para termos um novo usuário melhor, pois o incentivo é maior.
13	K – O envio dos dados à central deve ser no formato digital e não deve levar mais de 10 segundos.	Os dados só serão úteis se forem rapidamente tratados e enviados à uma central que coleta todos os dados dos usuários. O limite de 10 segundos é um tempo aceitável de espera ao final do trajeto sem que cause frustração ao usuário.
14	L – A interface deve apresentar funções minimalistas e ser acessível para qualquer pessoa.	Os comandos devem ser reduzidos ao mínimo necessário, devem ser realizados no menor tempo possível e deve apresentar um design amigável que incentive o seu uso frequente. Conhecimentos técnicos prévios não devem ser requeridos.

Fonte: Autor

3.2.1 Matrizes de Compromisso

A Tabela 2 - **Compromissos de engenharia/marketing** apresenta o estudo comparativo de correlação entre os requisitos de engenharia e marketing:

Tabela 2 - Compromissos de engenharia/marketing

		A	B	C	D	E	F	G	H	I	J	K	L
		-	+	+	-	-	-	+	+	+	-	-	-
1 - Peso	-				↑	↑↑			↑				
2 - Tamanho	-				↑	↑↑	↑↑		↑				
3 - Peças	-				↑		↑↑		↑				
4 - Ergonomia	+					↑↑	↑		↑↑				
5 - Privacidade	+		↑					↑↑		↓			↓
6 - Energia	-		↓↓		↑↑	↑	↑	↓			↑		
7 - Etapas	-		↓			↑	↑	↓		↑		↑	↑
8 - Distração	-	↑		↑			↑	↓	↑	↑↑		↑	↑
9 - Custo	-				↑	↑	↑				↑↑		↑
10 - Confiabilidade	+	↑↑		↑↑			↑	↑				↑	↑
11 - Precisão	+	↑	↑↑		↑↑	↓↓	↓↓	↑				↓	
12 - Coleta eficiente	+	↑	↑	↑↑			↓	↑		↑			
13 - Envio eficiente	+						↓	↑				↑↑	
14 - Interface	+	↑		↑				↓				↑	↑↑

Fonte: Autor

Já a Tabela 3 apresenta o estudo comparativo de correlação entre os próprios requisitos de engenharia:

Tabela 3 - Compromissos de engenharia

		A	B	C	D	E	F	G	H	I	J	K	L
		-	+	*	-	-	-	+	+	+	-	-	-
A – Tempo de resposta	-		↑↑		↓↓	↓↓				↑		↑	
B – Precisão na posição	+				↓↓	↓	↑	↑		↑	↓↓	↓↓	
C – Taxa de acerto	+				↓	↓↓				↓	↓↓	↓	↓
D – Consumo de energia	-					↑↑	↑↑			↑↑	↑		
E – Dimensionamento limite	-						↑↑		↑↑		↑	↑	↑↑
F – Número de peças	-								↑↑	↑	↑	↑	↑
G – Etapa de privacidade	+									↑		↓	↓↓
H – Design	+										↓↓		↑
I – Coleta autônoma	+										↑	↑	↑↑
J – Custo limite	-											↓	
K – Tempo de envio	-												↑
L – Interface Mínima	-												

Fonte: Autor

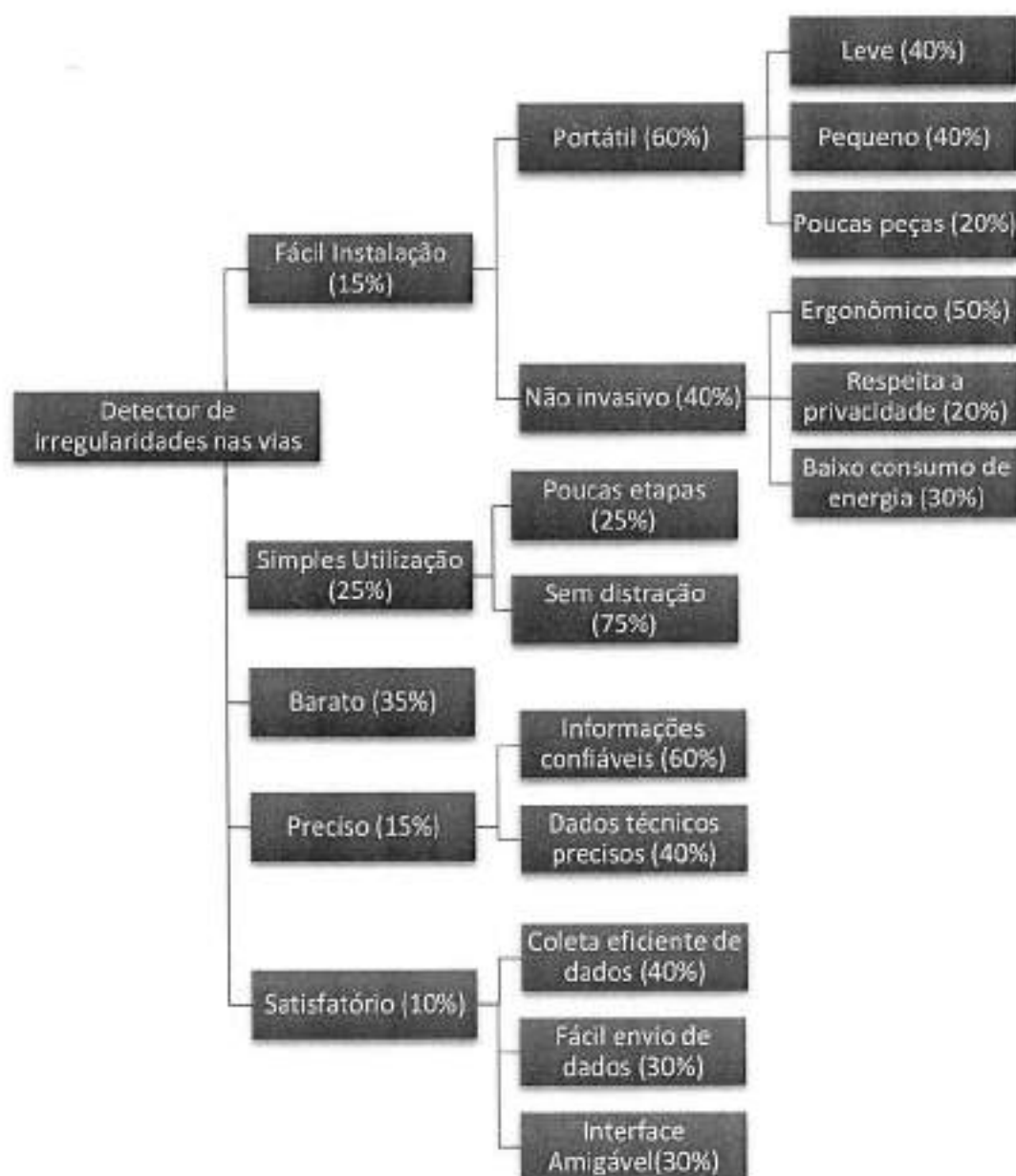
Legenda:

- + Aumentando melhora
- - Diminuindo melhora
- ↑↑ Correlação positiva forte
- ↑ Correlação positiva
- ↓↓ Correlação negativa forte
- ↓ Correlação negativa

3.3 ÁRVORE DE OBJETIVOS

Organizando de forma lógica estes requisitos, podemos ilustrar a seguinte árvore de objetivos com seus respectivos pesos percentuais de prioridade por ramo (Figura 7):

Figura 7 - Árvore de objetivos



Fonte: Autor

3.4 GERAÇÃO DE CONCEITOS E CRITÉRIOS PARA ESCOLHA

3.4.1 Geração de Conceitos

A Tabela 4 - **Geração de conceitos** mostra todos os conceitos possíveis a serem utilizados como sensor, unidade embarcada e base de dados:

Tabela 4 - Geração de conceitos

Sensor	Unidade embarcada	Base de dados
Câmera [27]	Micro controlador	Base de Dados SQL
Ultrassônico [28]	Smartphone	Amazon Web Services
Infravermelho [29]	Próprio Veículo	Google Mobile Backend
Acelerômetro [9]		

Fonte: Autor

Após esta geração de conceitos, podemos elencar as forças e fraquezas de cada opção, como podemos observar na

Tabela 5 - Forças e fraquezas dos conceitos:

Tabela 5 - Forças e fraquezas dos conceitos

Opção	Forças	Fraquezas
Câmera	- Alta precisão no reconhecimento [27]; - É completamente autônoma e eficiente na coleta [27].	- Necessidade de um alto poder de processamento para suportar o algoritmo; - Tempo de resposta lento; - Uma câmera de qualidade é cara; - Alto consumo de energia; - A instalação não é simples e deve ser feita por uma equipe técnica.

Ultrassônico	• Possibilidade de medir profundidade e largura do buraco [28]; • Tempo de resposta rápido; • Baixo custo [28].	• A instalação não é simples e deve ser feita por uma equipe técnica; • Necessidade de instalação em vários locais no veículo, com muitas peças.
Infravermelho	• É completamente autônomo e eficiente na coleta de dados [29].	• Baixíssima precisão técnica no reconhecimento da condição do asfalto [29]. • A instalação não é simples e deve ser feita por uma equipe técnica.
Acelerômetro	• Baixíssimo custo de manutenção [9]; • Tempo de resposta rápido [4]; • É completamente autônomo e eficiente na coleta de dados [6]; • É fácil de ser instalado em qualquer lugar do veículo; • É portátil e leve; • Consome pouca energia.	• Dependendo das condições, pode não apresentar uma precisão tão boa.
Micro controlador	• Pode atingir melhores performances e precisão em geral; • É leve e pequeno; • As informações são confiáveis.	• Alto custo devido à necessidade de comprar um equipamento novo para cada usuário; • Longa instalação para cada novo colaborador; • A interface não é amigável; • Apresenta várias partes e diversas etapas de ativação.
Smartphone	• Fácil instalação; • Baixíssimo custo; • Facilmente escalável para inúmeros usuários;	• Dependendo das condições de uso, pode não apresentar uma precisão tão boa.

	• É portátil e não invasivo; • Possui uma interface amigável; • Fácil envio de dados à uma central.	
Próprio Veículo	• Pode atingir melhores performances e precisão altíssima em geral;	• Alto custo de instalação por usuário; • Método muito invasivo ao veículo; • A utilização não é simples; • Dificuldade no envio dos dados.
Base de Dados SQL	• Fácil instalação; • Barato; • Requer poucos conhecimentos prévios.	• Pode não oferecer todos os recursos futuramente necessários.
Amazon Web Services	• Possui boa integração com diversas funcionalidade externas; • Oferece machine learning incluso na nuvem.	• Preço elevado; • Requer vários conhecimentos prévios.
Google Mobile Backend	• Boa integração com os outros serviços Google e Android utilizados; • Possui boa integração com diversas funcionalidade externas.	• Preço elevado; • Requer alguns conhecimentos prévios.

Fonte: Autor

3.4.2 Seleção de Conceitos

Tendo em vista estes pontos fortes e fraquezas de cada conceito, foi feita uma avaliação de conceitos com pontuação de 0 a 5 para cada folha da árvore de objetivo, como mostrado na Tabela 6 - **Seleção de sensor** e Tabela 7 - **Seleção de unidade embarcada**:

Tabela 6 - Seleção de sensor

Critério	Ponderação	Câmera	Ultrassônico	Infravermelho	Acelerômetro
1 - Peso	0,036	2	3	3	3
2 - Tamanho	0,036	1	1	1	5
3 - Peças	0,018	3	2	2	5
4 - Ergonomia	0,03	4	2	2	5
5 - Privacíd.	0,012	2	5	5	4
6 - Energia	0,018	1	5	5	5
7 - Etapas	0,0625	3	4	4	5
8 - Distração	0,1875	3	3	3	3
9 - Custo	0,35	1	2	3	5
10 - Confiab.	0,09	5	4	1	3
11 - Precisão	0,06	5	4	1	3
12 - Coleta ef.	0,04	5	4	1	3
13 - Envio ef.	0,03	2	3	3	5
14 - Interface	0,03	3	3	3	3
TOTAL		2,52	2,84	2,62	<u>4,1</u>

Fonte: Autor

Tabela 7 - Seleção de unidade embarcada

Critério	Ponderação	Micro Controlador	Smartphone	Próprio Veículo
1 - Peso	0,036	3	5	5
2 - Tamanho	0,036	3	5	5
3 - Peças	0,018	2	5	3
4 - Ergonomia	0,03	1	5	4
5 - Privacíd.	0,012	2	5	1
6 - Energia	0,018	2	3	5
7 - Etapas	0,0625	2	5	1
8 - Distração	0,1875	2	4	3
9 - Custo	0,35	1	5	1
10 - Confiab.	0,09	4	3	5
11 - Precisão	0,06	4	3	5
12 - Coleta ef.	0,04	4	3	4
13 - Envio ef.	0,03	3	5	2
14 - Interface	0,03	1	5	2
TOTAL		2,07	<u>4,4</u>	2,64

Fonte: Autor

Pode-se concluir através da metodologia de seleção adotada que a melhor escolha de conceito é o uso de smartphones com base no seu sensor acelerômetro interno. Este é o conceito que atende melhor os objetivos ponderados na árvore da seção 4. O desenvolvimento será feito com base em um aplicativo Android na linguagem Java principalmente, devido aos conhecimentos prévios do autor e à facilidade de desenvolvimento na plataforma Android que é bem mais aberta do que iOS por exemplo.

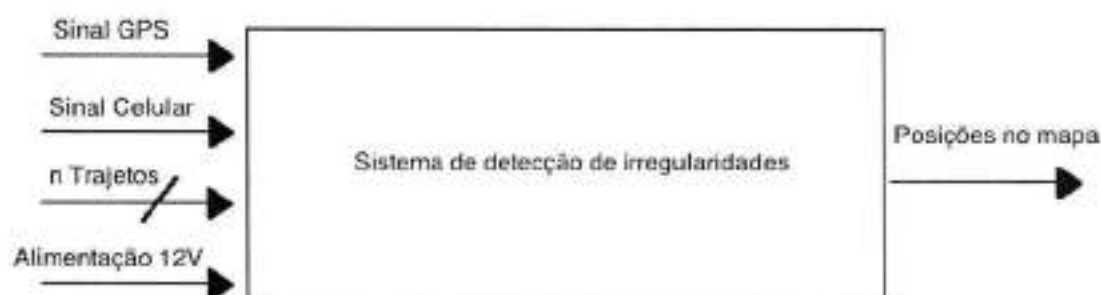
Quanto à base de dados, será utilizada uma base de dados SQL. Eventualmente, se o custo couber no orçamento e houver um aprendizado mais acelerado do que o esperado, será usado o serviço cloud da Google ou da Amazon. A adição de um serviço cloud ao projeto ajuda na escalabilidade futura de projeto em uma eventual continuação e também oferece funcionalidade adicionais interessantes do ponto de vista de performance.

3.5 DECOMPOSIÇÃO FUNCIONAL DETALHADA

3.5.1 Nível 0

O nível 0 do projeto em si encontra-se representado na Figura 8 - **Nível 0** e na Tabela 8 - **Nível 0 da decomposição funcional**:

Figura 8 - Nível 0 da decomposição funcional



Fonte: Autor

Tabela 8 - Nível 0 da decomposição funcional

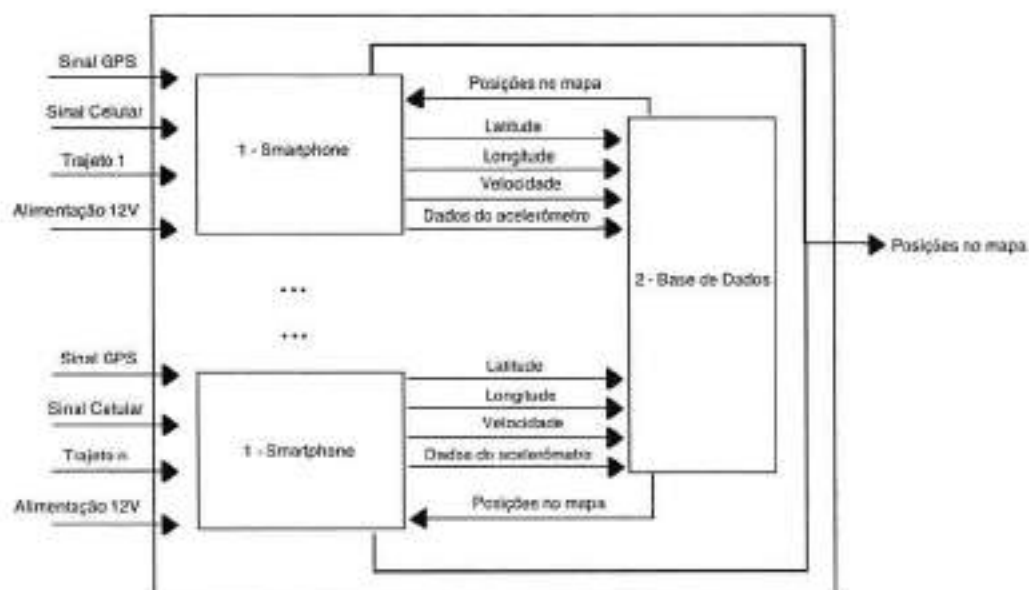
Módulo	Projeto
Entradas	Sinal GPS, sinal celular, alimentação 12V, n trajetos dos veículos
Saídas	Posições no mapa
Funcionalidade	O sistema detector se encontra acoplado e fixo no veículo. Ele coleta os sinais característicos do trajeto do usuário, recebe sinal GPS dos satélites e usa dados celulares para realizar toda a transmissão de dados. Ele pode eventualmente ser carregado pelo acendedor de cigarros do veículo.

Fonte: Autor

3.5.2 Nível 1

O nível 1 apresenta dois módulos (smartphone e base de dados) representados na Figura 9 - **Nível 1 da decomposição funcional**, Tabela 9 - **Módulo 1 da decomposição funcional** e Tabela 10 - **Módulo 2 da decomposição funcional**:

Figura 9 - Nível 1 da decomposição funcional



Fonte: Autor

Tabela 9 - Módulo 1 da decomposição funcional

Módulo 1	Smartphone Android
Entradas	- Sinal GPS; - Sinal celular; - Alimentação 12V - Trajeto do veículo; - Posições no mapa com precisão de até 30 metros.
Saídas	- Latitude; - Longitude; - Velocidade; - Dados dos 3 eixos do acelerômetro; - Posições no mapa com precisão de até 30 metros.
Funcionalidade	O smartphone coleta os sinais do trajeto através de seus sensores, recebe sinal GPS dos satélites e usa dados celulares para realizar toda a transmissão de dados. O tempo de resposta para uma irregularidade encontrada é de até 2 segundos. Ele pode eventualmente ser carregado pelo acendedor de cigarros do veículo.

Fonte: Autor

Tabela 10 - Módulo 2 da decomposição funcional

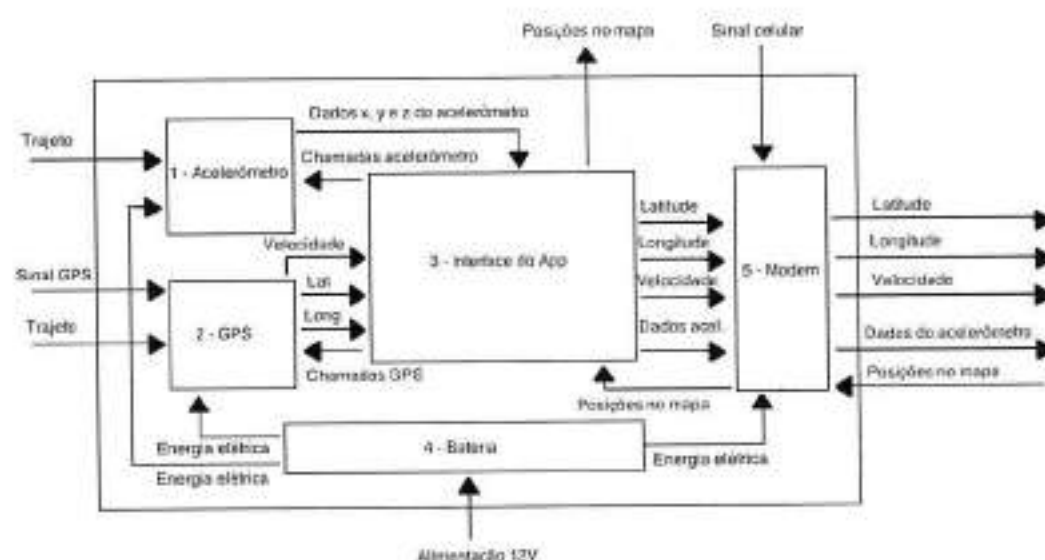
Módulo 2	Base de Dados
Entradas	• Latitude; • Longitude; • Velocidade; • Dados dos 3 eixos do acelerômetro; • Posições no mapa com precisão de até 30 metros.
Saídas	• Posições no mapa com precisão de até 30 metros.
Funcionalidade	O módulo de base de dados recebe e armazena as informações coletadas por cada smartphone. Ele devolve sob demanda a posição e os tipos de irregularidades encontradas por todos os usuários do sistema dentro de uma faixa desejada.

Fonte: Autor

3.5.3 Nível 2

O nível 2 apresenta os seguintes sub-módulos representados na Figura 10, Tabela 11, Tabela 12, Tabela 13, Tabela 14 e Tabela 15:

Figura 10 - Nível 2 da decomposição funcional



Fonte: Autor

Tabela 11 - Sub-Módulo 1 da decomposição funcional

Sub-Módulo 1	Acelerômetro
Entradas	• Energia elétrica; • Chamadas de função acelerômetro; • Trajeto.
Saídas	• Dados dos eixos x, y e z
Funcionalidade	O acelerômetro consome energia, é ativado pelo app e recebe chamadas de funções para devolver os valores de aceleração nos 3 eixos.

Fonte: Autor

Tabela 12 - Sub-Módulo 2 da decomposição funcional

Sub-Módulo 2	GPS
Entradas	• Energia elétrica; • Sinal GPS; • Chamadas de função GPS; • Trajeto.
Saídas	• Latitude; • Longitude; • Velocidade.
Funcionalidade	O GPS consome energia, é ativado pelo app e recebe chamadas de funções para devolver os valores de latitude, longitude e velocidade através do seu protocolo interno.

Fonte: Autor

Tabela 13 - Sub-Módulo 3 da decomposição funcional

Sub-Módulo 3	Interface do App
Entradas	• Latitude; • Longitude; • Velocidade; • Dados dos 3 eixos do acelerômetro; • Posições no mapa com precisão de até 30 metros.
Saídas	• Latitude; • Longitude; • Velocidade; • Dados dos 3 eixos do acelerômetro; • Posições no mapa com precisão de até 30 metros; • Chamadas de função GPS; • Chamadas de função acelerômetro.
Funcionalidade	A interface do app é responsável por reunir os dados provenientes do GPS e do acelerômetro. Após processá-los, ele mostra informações na tela e troca dados com a base de dados central a respeito das posições no mapa coletadas em outros trajetos também.

Fonte: Autor

Tabela 14 - Sub-Módulo 4 da decomposição funcional

Sub-Módulo 4	Bateria
Entradas	• Alimentação 12V
Saídas	• Energia elétrica
Funcionalidade	A bateria do smartphone pode ser carregada pelo acendedor de cigarro do carro.

Fonte: Autor

Tabela 15 - Sub-Módulo 5 da decomposição funcional

Sub-Módulo 5	Modem
Entradas	• Energia elétrica; • Sinal celular; • Latitude; • Longitude; • Velocidade; • Dados dos 3 eixos do acelerômetro; • Posições no mapa com precisão de até 30 metros.
Saídas	• Latitude; • Longitude; • Velocidade; • Dados dos 3 eixos do acelerômetro; • Posições no mapa com precisão de até 30 metros.
Funcionalidade	O modem realiza a comunicação com a base de dados através de tecnologias 3G/4G, a fim de se conectar à Internet.

Fonte: Autor

3.5.4 Análise de Acoplamento e Coesão

O smartphone não é um sistema muito acoplado, pois uma eventual falha em um módulo não se propaga a outros. Mesmo assim, como se trata de um dispositivo fechado, é difícil reprojeter um módulo separado.

O sistema também é extremamente altamente coeso, sendo fácil de testar os módulos de maneira independente (através de testes unitários) e com uma interface de controle simples (chamadas de funções).

3.6 GERENCIAMENTO DO PROJETO DETALHADO

Para garantir o bom desenvolvimento do trabalho e completar todas as etapas necessárias, foi elaborado o recurso de gerenciamento do projeto (Tabela 16), que foi seguido com sucesso e inclusive de forma adiantada:

Tabela 16 - Gerenciamento do Projeto

ID	Atividade	Descrição	Tempo cons.	Tempo rest.	Requisito	Entregas	Recursos
1	Preparo do ambiente de trabalho	Criação de repositório GitHub, instalação de Android Studio, conexão com dispositivo de teste	6h	-	-	Ambiente de trabalho pronto	Macbook pro, Motorola XT918, cabo USB, Android Studio, GitHub
2	Desenvolvimento do Sub-Módulo Acelerômetro	Aquisição de dados do Acelerômetro	4h	-	1	Sub-Módulo Acelerômetro pronto	-
3	Desenvolvimento do Sub-Módulo GPS	Aquisição de dados do GPS	3h	-	1	Sub-Módulo GPS pronto	-
4	Desenvolvimento do Sub-Módulo Interface	Desenvolvimento da Interface homem-máquina para a realização de testes	11h	-	1, 2, 3	Sub-Módulo Interface do App pronta	-
5	Teste do Sub-Módulo Acelerômetro	Testes em campos dos resultados do acelerômetro	4h	-	2	Sub-Módulo Acelerômetro aprovada	Veículo, Motorola XT918
6	Teste do Sub-Módulo GPS	Testes em campos dos resultados do GPS	2h	-	3	Sub-Módulo GPS aprovada	Veículo, Motorola XT918
7	Teste do Sub-Módulo Interface	Testes em campos da interface e seu uso	2h	-	4	Sub-Módulo Interface do App aprovada	Veículo, Motorola XT918
8	Relatório A1	Escrita e organização do Relatório A1	21h	-	1, 2, 5	Relatório A1 pronto	Microsoft Word
9	Relatório A2	Escrita e organização do	7h	-	8	Relatório A2 pronto	Microsoft Word

Relatório A2							
10	Apresentação A2	Organização da Apresentação A2 em slides	2h	-	9	Apresentação A2 pronta	Slides PPT
11	Relatório A3	Escrita e organização do Relatório A3	8h	-	9, 19	Relatório A3 pronto	Microsoft Word
12	Apresentação A3	Organização da Apresentação A2 em slides	6h	-	10, 11, 18	Apresentação A3 pronta	Slides PPT
13	Poster A3	Escrita e organização do Poster A3	4h	-	11, 12	Poster A3 pronto	Poster A3
14	Reuniões com orientador	Reuniões de acompanhamento com o orientador ao longo do ano	16h	-	-	Ata de reunião	Sala de reunião
15	Reuniões com co-orientadores	Reuniões de acompanhamento com os co-orientadores ao longo do ano	2h	-	-	Ata de reunião	Sala de reunião
16	Condicionamento de dados do acelerômetro	Anulação do efeito da gravidade, calibração automática, cálculos.	12h	-	2	Novo formato de dados do acelerômetro	Algoritmos de cálculo usando vetores
17	Design gráfico	Símbolo do app, imagens, fontes	10h	-	1	Elementos gráficos adicionados ao app	Paint, imagens web, sites especializados para Android
18	Filmagem	Filmagem dentro do veículo da app rodando no smartphone e seus resultados	2h	-	7	Vídeo de apresentação pronto	Veículo, Câmera, Motorola XT918
19	Desenvolvimento para envio de dados coletados de teste	Envio em um formato de texto para tratamento de	8h	-	5, 6	Obtenção de um dataset para validação de algoritmo	-
20	Apuração do algoritmo de detecção com elementos de machine learning	Uso do dataset obtido para treinar e testar um modelo mais apurado	20h	-	19	Novo algoritmo de detecção	Múltiplo back Propagation, Microsoft Excel
21	Desenvolvimento de mapa	Desenvolvimento de um mapa base dentro do App	25h	-	1, 19	Mapa pronto para visualização no	Google Maps API

							app	
2	Estudo de	Estudo de	7h	-	-	Proposta de	Internet,	
2	formato para	propostas de				formato de	Google	
	envio à base de	formato para				envio		
	dados	envio à base de						
	dados	dados						
2	Criação de um	Pagamento de	5h	-	-	Servidor web	Servidor web	
3	servidor web	um servidor web				disponível para	pago	
		para criação de				uso		
	página							
2	Implementação	Implementação	5h	-	23	Base de dados	Site Web	
4	de uma base de	de uma base de				SQL disponível		
	dados SQL	dados SQL no				para utilização		
		servidor para uso						
2	Desenvolvimento	Cálculos sendo	12h	-	24	Cálculos sendo	Site Web,	
5	backend do	realizados online				realizados	base de	
	servidor	pelo				online	dados SQL	
		desenvolvimento						
	backend	backend						
2	Comunicação	Comunicação	10h	-	24, 25	Comunicação	Site Web,	
6	entre o App e a	entre os dois				entre os dois	base de	
	base de dados	módulos deve ser				módulos é feita	dados SQL,	
		feita com sucesso				com sucesso	Android	
							Studio	
2	Adaptação ao	O layout deve	10h	-	21, 26	Pontos	Site Web,	
7	layout do mapa	mostrar os pontos				mostrados no	base de	
	no app	no mapa de				mapa de acordo	dados SQL,	
		acordo com				com o esperado	Android	
		diferentes					Studio	
		critérios						
2	Ajustes finais da	Ajustes finais no	20h	-	7, 27	Layout global	-	
6	interface	design do				funcionando		
		aplicativo de				como esperado		
		acordo com os						
		requisitos formais						
		do cliente						
2	Testes finais	Testes finais para	16h	-	28	O aplicativo	Veículo,	
9		validação				funciona	Câmera,	
							Motorola	
							XT918	
3	Monografia	Escrita da	40h	-	11, 29	Monografia	Microsoft	
0		monografia				pronta	Word	

Fonte: Autor

3.7 AVALIAÇÃO DE RISCOS

Pode-se estabelecer o seguinte diagrama SWOT para o projeto (Figura 11):

Figura 11 - Diagrama SWOT

	<i>Positivo</i>	<i>Negativo</i>
<i>Interno</i>	Forças • Ideia inovadora; • Aplicativo aberto e livre para o uso; • Interface amigável; • Baixo custo global; • Fácil instalação.	Fraquezas • A precisão do smartphone pode não ser suficiente; • Recursos humanos e de tempo são limitados.
<i>Externo</i>	Oportunidades • Fácil escalabilidade; • Muitos clientes potenciais disponíveis; • Excelente acessibilidade para novos usuários.	Riscos • Competidor de grande porte; • Desinteresse por parte do poder público; • Dificuldade de convencimento dos primeiros usuários.

Fonte: Autor

Além disso, os riscos do projeto podem ser classificados por impacto e probabilidade de ocorrerem, como visto na Figura 12:

Figura 12 - Análise de risco

Muito provável		• Dificuldade de convencimento dos primeiros usuários	
Provável	• GPS não oferece a precisão necessária.		• Desinteresse do poder público.
Pouco provável	• Falta de recursos humanos; • Falta de tempo.	• Acelerômetro não oferece a precisão necessária. • Problema no envio dos dados usando dados do celular	• Competidor de grande porte.
	Risco baixo	Risco moderado	Alto Risco

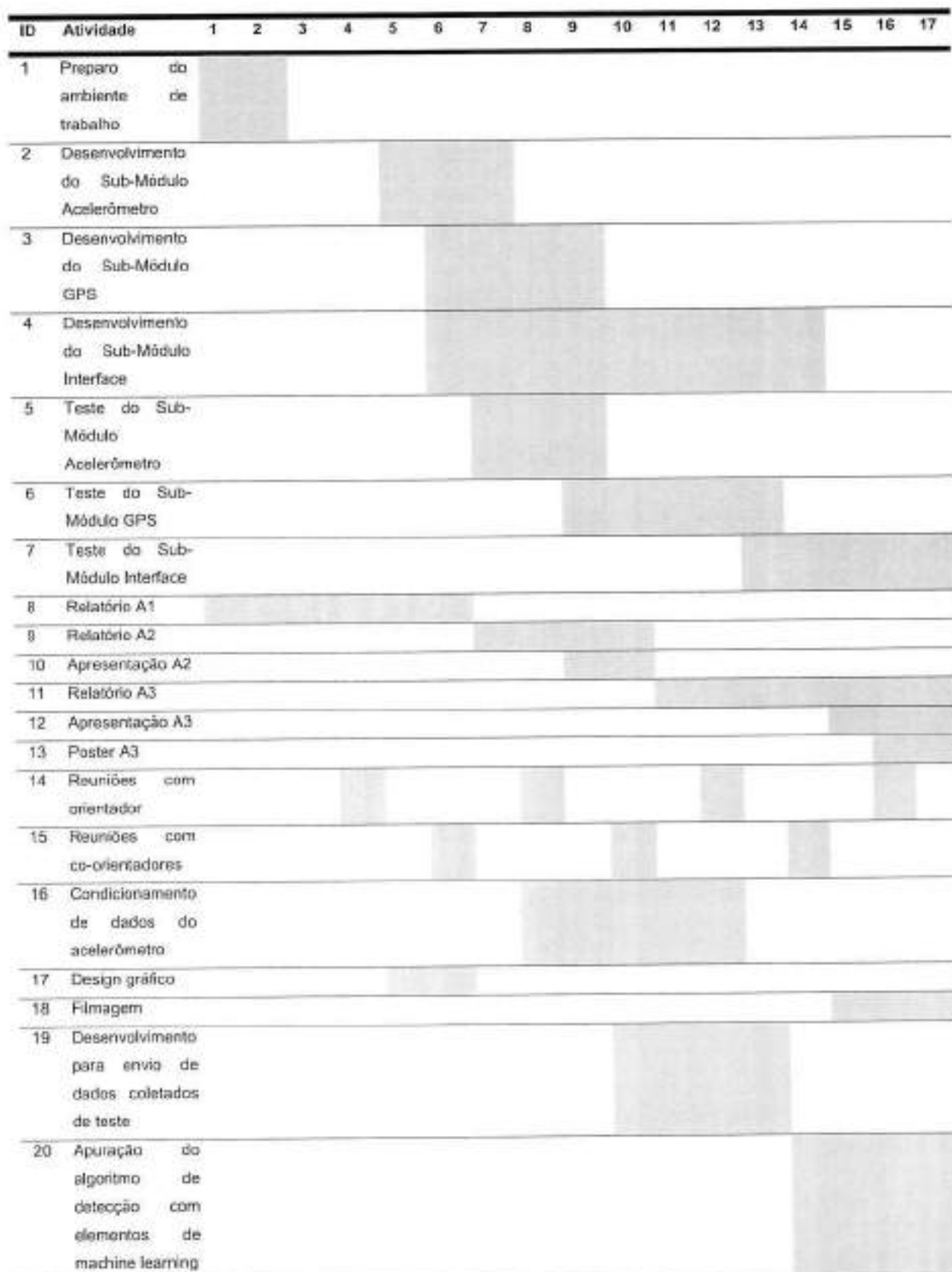
Fonte: Autor

3.8 CRONOGRAMA DETALHADO

Foi gasto um total de 300 horas durante o projeto. Isto equivale a um consumo médio de 8,1 horas por semana para o projeto de formatura.

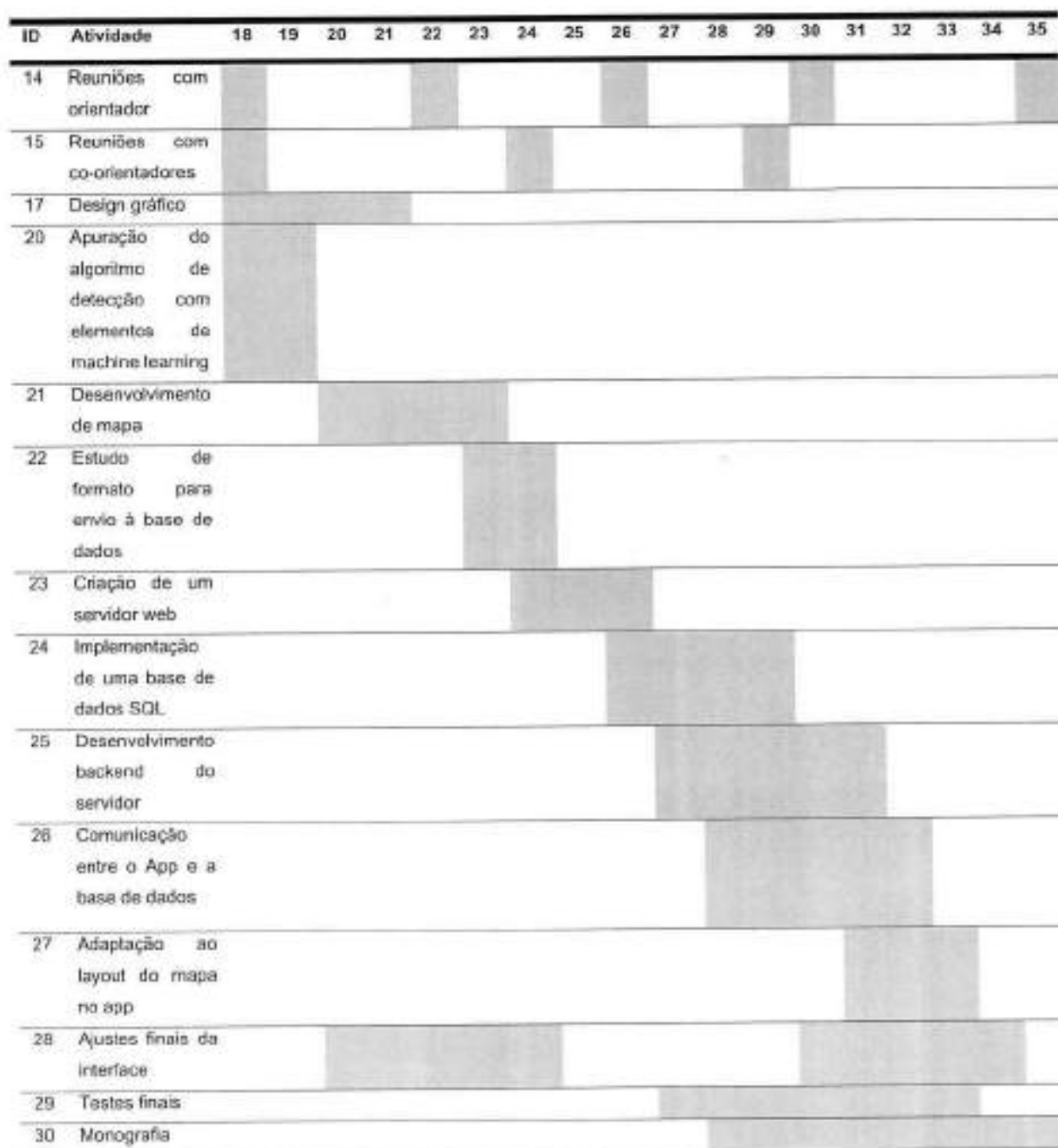
De uma forma geral, foi adotada a metodologia em espiral e ágil de desenvolvimento. Foi estabelecido o seguinte cronograma entre o primeiro e o segundo semestre de 2017 para o desenvolvimento do projeto (Figura 13 e Figura 14).

Figura 13 - Cronograma detalhado do projeto no primeiro semestre



Fonte: Autor

Figura 14 - Cronograma detalhado do projeto no segundo semestre



Fonte: Autor

3.9 ORÇAMENTO

Os principais gastos do projeto são:

- Gasolina para abastecer o carro usado nos testes;
- Registro de domínio Web (Figura 15);
- Pagamento de um servidor (Figura 15);
- Solicitação de Patente de Invenção (PI).

Figura 15 - Gastos com servidor

Invoice#	Package	Due Date	Date Paid	Amount Paid
<u>58087280</u>	SH-2556091 grivol.com	2017-07-28	2017-06-28	\$107.40
<u>58339314</u>	Domain grivol.com	2017-06-28	2017-06-28	\$15.00

Fonte: Autor

Estima-se que o projeto será testado no total durante 100 horas. Estimando uma velocidade média de 40km/h, temos um total de 4000 km percorridos com o veículo. Considerando um preço de R\$ 3,50 / L e que o carro tenha uma eficiência de 10,5 km/L, pode-se concluir que será gasto em torno de R\$ 1333 reais com os testes práticos do aplicativo. No total, o orçamento previsto será de R\$ 1934,00 (**Tabela 17**).

Tabela 17- Orçamento

Descrição dos gastos	Valor
Gasolina	R\$ 1333,00
Domínio Web	R\$ 352,00
Servidor	R\$ 49,00
Patente	R\$ 200,00
Total	R\$ 1934,00

Fonte: Autor

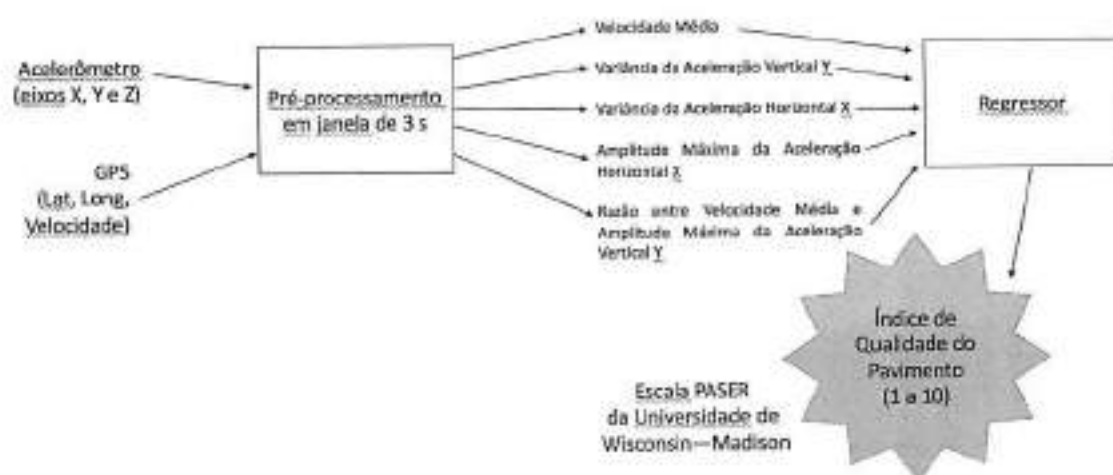
4 IMPLEMENTAÇÃO

4.1 ESTRUTURA DA REDE NEURAL

No âmbito da prova de conceito, foi elaborada uma rede neural para estimar o índice de qualidade do pavimento das vias de acordo com a escala PASER de qualidade, da Universidade de Wisconsin-Madison [29].

Para fins de abstração da solução adotada, pode-se considerar que a rede neural adotada é composta por um bloco de pré-processamento e outro bloco regressor (Figura 16).

Figura 16 - Rede Neural



Fonte: Autor

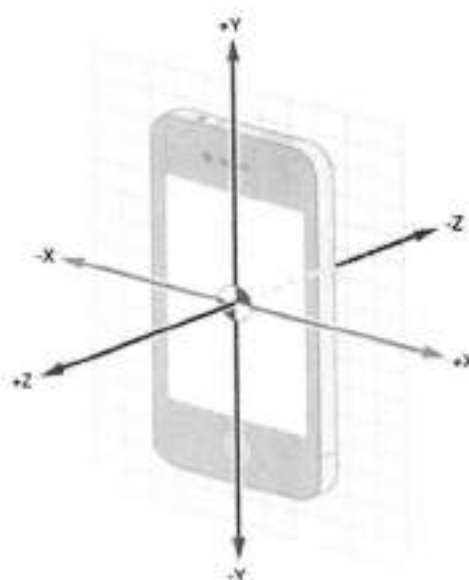
O pré-processamento envolve uma janela temporal de 3s. Isto porque a taxa de amostragem utilizada para o acelerômetro foi $\text{SENSOR_DELAY_UI} = 60\text{ms}$ e é estocado um conjunto de 50 pontos.

Entre os dois blocos se encontram as 5 entradas que foram efetivamente utilizadas na rede neural, que são:

- Velocidade Média;
- Variância da Aceleração Vertical Y;
- Variância da Aceleração Horizontal X;
- Amplitude Máxima da Aceleração Horizontal X;
- Razão entre Velocidade Média e Amplitude Máxima da Aceleração Vertical Y.

Os smartphones com sensores Android apresentam os seguintes eixos de orientação (Figura 17) :

Figura 17 - Eixos do acelerômetro para Android



Fonte: Autor

O sensor acelerômetro obtém, portanto, os valores de aceleração nos eixos X, Y e Z, sendo que o aplicativo coleta uma janela de 50 pontos por vez (totalizando 3s). A partir desta janela são calculados os valores de entrada da rede neural evidenciados anteriormente.

Associado a isto, o GPS obtém os seguintes valores com uma atualização de 0,5s:

- Velocidade;
- Latitude;
- Longitude.

A velocidade obtida serve primeiramente para calibrar os valores obtidos pelo acelerômetro. Como o veículo pode estar em uma rampa ou inclinado em uma situação normal, realiza-se uma calibração dos eixos com relação à gravidade quando a velocidade obtida pelo GPS é nula.

Isto é, quando o veículo se encontra parado, considera-se que nesta situação de equilíbrio os 3 valores de aceleração também são nulos. Esta operação deve ser feita para eliminar o efeito da gravidade sobre o sensor, que poderia contaminar as medidas com as componentes da gravidade sobre os 3 eixos.

4.2 ESCALA PASER

A escala PASER é um sistema de classificação de 1 à 10 para a condição do pavimento rodoviário desenvolvido pelo Centro de Informações de Transporte da Universidade de Wisconsin-Madison [29].

O PASER usa inspeção visual para avaliar as condições da superfície do pavimento. Quando avaliados corretamente, as classificações de PASER fornecem uma base para comparar a qualidade dos segmentos da estrada.

O método de avaliação do PASER não requer medições de angústias individuais e, portanto, as classificações do PASER não podem ser desagregadas em medidas de tipos de dificuldade específicos. A vantagem desse método é que as estradas podem ser avaliadas rapidamente, possivelmente até mesmo por meio de "levantamento de pára-brisas".

As classificações numéricas do PASER são traduzíveis para categorias de condições e opções de tratamento prescritas, conforme mostrado abaixo (Figura 18).

Figura 18 - Escala PASER

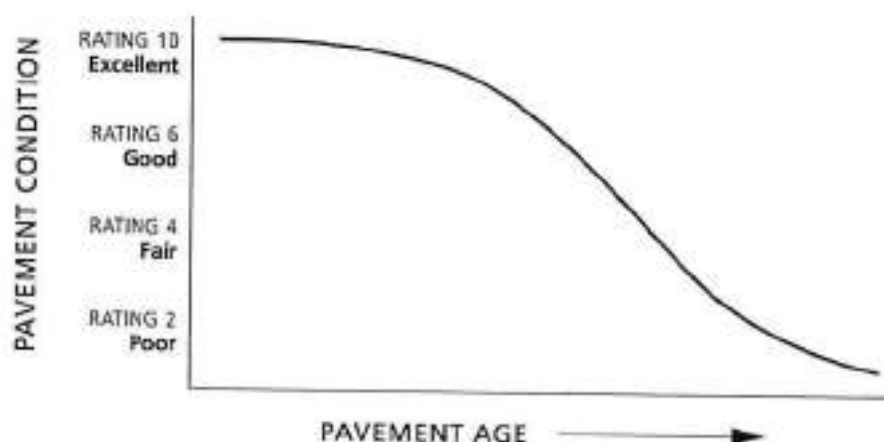
<i>Surface rating</i>	<i>Visible distress*</i>	<i>General condition/ treatment measures</i>
10 Excellent	None.	New construction.
9 Excellent	None.	Recent overlay. Like new.
8 Very Good	No longitudinal cracks except reflection of paving joints. Occasional transverse cracks, widely spaced (40' or greater). All cracks sealed or tight (open less than 1/4").	Recent sealcoat or new cold mix. Little or no maintenance required.
7 Good	Very slight or no raveling, surface shows some traffic wear. Longitudinal cracks (open 1/4") due to reflection or paving joints. Transverse cracks (open 1/4") spaced 10' or more apart, little or slight crack raveling. No patching or very few patches in excellent condition.	First signs of aging. Maintain with routine crack filling.
6 Good	Slight raveling (loss of fines) and traffic wear. Longitudinal cracks (open 1/4" - 1/2"), some spaced less than 10'. First sign of block cracking. Slight to moderate flushing or polishing. Occasional patching in good condition.	Shows signs of aging. Sound structural condition. Could extend life with sealcoat.
5 Fair	Moderate to severe raveling (loss of fine and coarse aggregate). Longitudinal and transverse cracks (open 1/2") show first signs of slight raveling and secondary cracks. First signs of longitudinal cracks near pavement edge. Block cracking up to 50% of surface. Extensive to severe flushing or polishing. Some patching or edge wedging in good condition.	Surface aging. Sound structural condition. Needs sealcoat or thin non-structural overlay (less than 2").
4 Fair	Severe surface raveling. Multiple longitudinal and transverse cracking with slight raveling. Longitudinal cracking in wheel path. Block cracking (over 50% of surface). Patching in fair condition. Slight rutting or distortions (1/2" deep or less).	Significant aging and first signs of need for strengthening. Would benefit from a structural overlay (2" or more).
3 Poor	Closely spaced longitudinal and transverse cracks often showing raveling and crack erosion. Severe block cracking. Some alligator cracking (less than 25% of surface). Patches in fair to poor condition. Moderate rutting or distortion (1" or 2" deep). Occasional potholes.	Needs patching and repair prior to major overlay. Milling and removal of deterioration extends the life of overlay.
2 Very Poor	Alligator cracking (over 25% of surface). Severe distortions (over 2" deep). Extensive patching in poor condition. Potholes.	Severe deterioration. Needs reconstruction with extensive base repair. Pulverization of old pavement is effective.
1 Failed	Severe distress with extensive loss of surface integrity.	Failed. Needs total reconstruction.

Fonte: PASER Manual [29]

O Michigan Transportation Asset Management Council selecionou o sistema de classificação PASER como o padrão estadual de relatórios de condições do pavimento.

Além disto, a escala se encontra relacionada à idade do pavimento e também às medidas adequadas a serem tomadas (Figura 19).

Figura 19 - Condição do pavimento e seu reparo



In addition to indicating the surface condition of a road, a given rating also includes a recommendation for needed maintenance or repair. This feature of the rating system facilitates its use and enhances its value as a tool in ongoing road maintenance.

RATINGS ARE RELATED TO NEEDED MAINTENANCE OR REPAIR

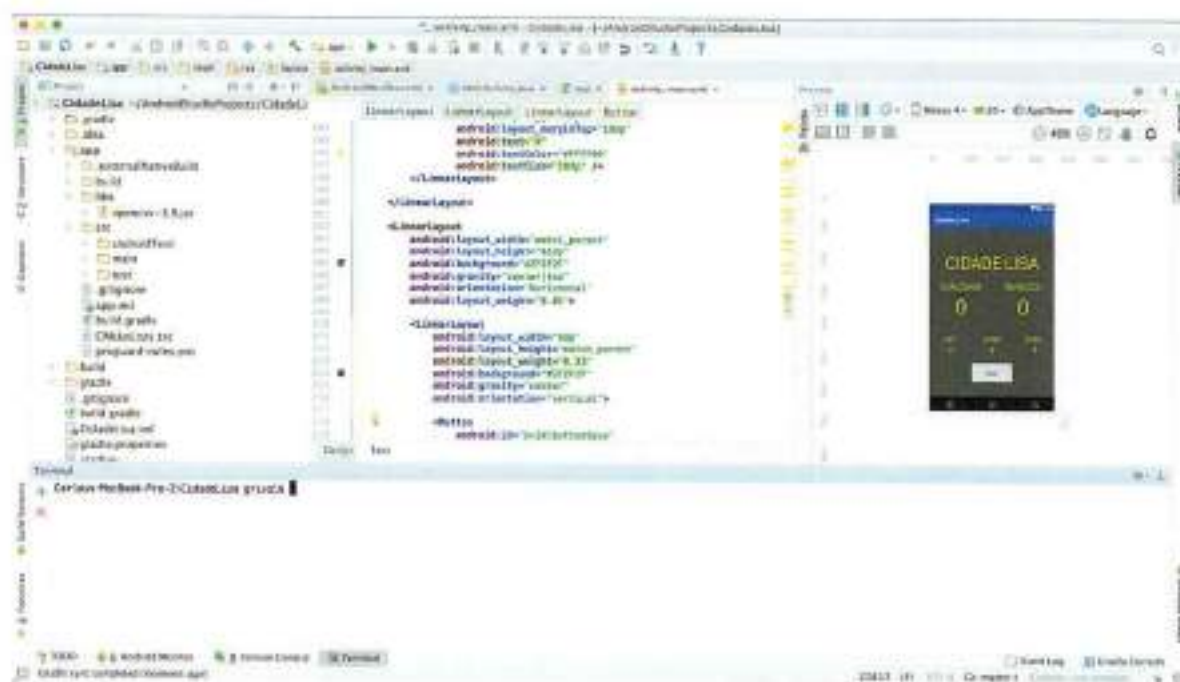
Rating 9 & 10	No maintenance required
Rating 8	Little or no maintenance
Rating 7	Routine maintenance, cracksealing and minor patching
Rating 5 & 6	Preservative treatments (sealcoating)
Rating 3 & 4	Structural improvement and leveling (overlay or recycling)
Rating 1 & 2	Reconstruction

Fonte: PASER Manual [29]

4.3 DESENVOLVIMENTO DO APLICATIVO

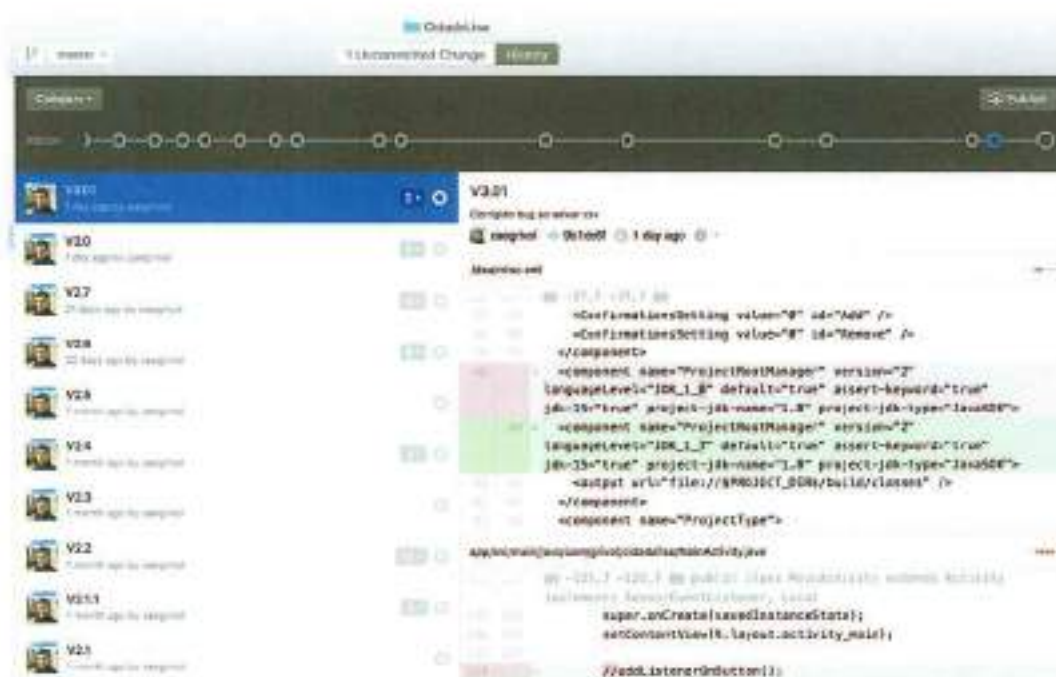
Todo o desenvolvimento do aplicativo foi feito no ambiente Android Studio (Figura 20). O versionamento do app foi realizado com GitHub desktop (Figura 21). O aplicativo se chama Cidade Lisa, se encontra na versão 3.01 e conta inclusive com ícones de app desenvolvidos sob medida (Figura 22).

Figura 20 - Ambiente de desenvolvimento Android Studio



Fonte: Autor

Figura 21 - Versionamento do app no GitHub



Fonte: Autor

Figura 22 - Ícones do app Cidade Lisa



Fonte: Autor

A versão 1.0 do app serviu como um painel de bordo para visualizar os dados coletados pelos sensores, em tempo real (Figura 23). Esta fase foi importante para se compreender o funcionamento do sistema Android e também para analisar os pontos críticos do projeto.

Figura 23 - Versão 1.0 do aplicativo Cidade Lisa



Fonte: Autor

4.4 COLETA DE DADOS

Para o treino da rede neural, foi necessário coletar uma grande quantidade de amostras em campo. Isto foi possível graças a uma interface desenvolvida exclusivamente para este fim, à partir da versão 2.0 do aplicativo (Figura 24).

Esta interface contém uma série de botões que, ao serem pressionados, salvam na memória interna do smartphone todos os dados do acelerômetro e GPS coletados na janela de amostras em questão.

Figura 24 - Versão 2.0 do aplicativo Cidade Lisa



Fonte: Autor

Para isto, o smartphone de teste utilizado foi o Motorola XT918 e o carro era um Fiesta 1.6. O smartphone foi instalado na saída de ar acima do painel do veículo com o uso de um suporte fixador (Figura 25). Desta forma, o smartphone e o carro podem ser considerados aproximadamente como um único corpo.

Figura 25 - Smartphone fixo ao veículo para testes



Fonte: Autor

4.5 TRATAMENTO DE DADOS

Após uma longa coleta de amostras no drive-test, foram extraídos da memória do smartphone todos os arquivos .csv (Figura 26) gerados pelo aplicativo nos momentos em que algum botão foi pressionado. Todos os 176 arquivos foram colocados em uma pasta juntos (Figura 27), sendo que cada um é identificado pelo tipo de evento e o seu respectivo timestamp.

Figura 26 - Exemplo de log coletado

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	AccelerationX	-0.070000005	0.100000000	-1.225	-1.072	0.95799994	-1.5899999	-1.914	-1.761	-2.18200002	-1.378	0.65	-0.345
2	AccelerationY	1.6470001	0.7279997	0.49800014	-0.15299988	2.1829996	3.6310003	1.2639999	0.30599976	-0.9200001	0.451	1.8380007	2.1070004
3	AccelerationZ	0.8049998	-1.1110001	-1.034	0.4929999	-1.3409998	-2.145	-1.7620001	-2.0400001	-0.42100012	0.03799987	-0.11590001	0.6859998
4	Speed	52.56963	52.56963	52.56963	52.56963	52.56963	52.56963	52.56963	52.56963	52.56963	52.56963	52.56963	52.56963
5	AverageSpeed	55.024258	OverOrder	0	Latitude	-23.581748333333334	Longitude	-46.89444					
6	AverageX	-0.56413995	AverageY	1.04108	AverageZ	-0.06278005							
7	MinX	-7.507	MinY	-3.5090002	MinZ	-10.3910005							
8	MaxX	3.295	MaxY	3.8310001	MaxZ	1.670							
9	StdX	1.4443568	StdY	1.1262194	StdZ	1.0070328							
10	GravityX	0.574	GravityY	8.135	GravityZ	1.183							

Fonte: Autor

Figura 27 - Arquivos coletados no drive-test

Name	^	Date Modified	Size	Kind
CLASSE1_2017-05-30 07:42:08.csv	⊙	30 May 2017, 07:42	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE1_2017-05-30 07:42:15.csv	⊙	30 May 2017, 07:42	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE1_2017-05-30 08:11:10.csv	⊙	30 May 2017, 08:11	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE1_2017-05-31 07:49:34.csv	⊙	31 May 2017, 07:49	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE1_2017-05-31 07:49:38.csv	⊙	31 May 2017, 07:49	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE1_2017-05-31 07:49:43.csv	⊙	31 May 2017, 07:49	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-29 07:41:06.csv	⊙	29 May 2017, 07:41	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-29 07:55:07.csv	⊙	29 May 2017, 07:55	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-29 18:00:30.csv	⊙	29 May 2017, 18:00	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-29 18:01:02.csv	⊙	29 May 2017, 18:01	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-29 18:01:21.csv	⊙	29 May 2017, 18:01	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-30 07:59:43.csv	⊙	30 May 2017, 07:59	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-30 08:00:42.csv	⊙	30 May 2017, 08:00	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-30 08:05:47.csv	⊙	30 May 2017, 08:05	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-30 08:06:58.csv	⊙	30 May 2017, 08:06	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-31 07:49:21.csv	⊙	31 May 2017, 07:49	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE2_2017-05-31 07:57:12.csv	⊙	31 May 2017, 07:57	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 07:54:56.csv	⊙	29 May 2017, 07:54	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 07:55:00.csv	⊙	29 May 2017, 07:55	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 07:55:29.csv	⊙	29 May 2017, 07:55	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 07:56:14.csv	⊙	29 May 2017, 07:56	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 07:56:24.csv	⊙	29 May 2017, 07:56	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 07:56:31.csv	⊙	29 May 2017, 07:56	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 17:47:57.csv	⊙	29 May 2017, 17:47	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 17:49:18.csv	⊙	29 May 2017, 17:49	3 KB	Comma Separated Spreadsheet (.csv)
CLASSE3_2017-05-29 17:49:27.csv	⊙	29 May 2017, 17:49	3 KB	Comma Separated Spreadsheet (.csv)

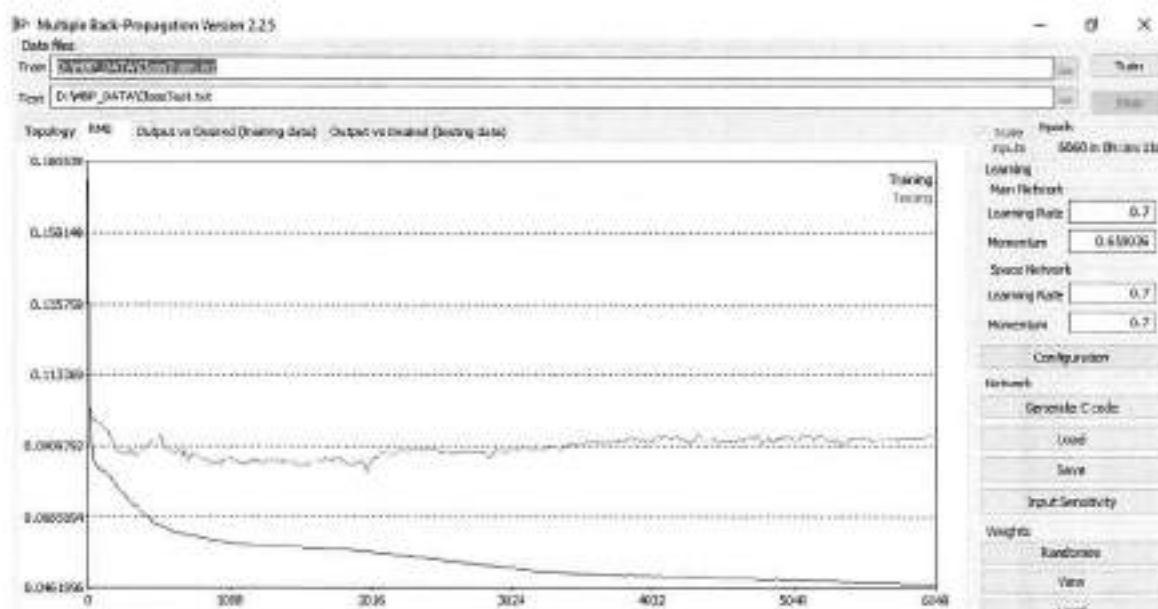
Fonte: Autor

Para o tratamento automático de dados, foi desenvolvido um script em Java que itera sobre os arquivos nesta pasta, lê os seus dados e cria um único arquivo que reúne todos os dados coletados. Este arquivo .csv gerado pode ser lido no Excel (Figura 28).

Figura 28 - Arquivo gerado pelo script Java

Min1	Min2	Max1	Max2	Min3	Min4	Max3	Max4	Min5	Min6	Max5	Max6	Min7	Min8	Max7	Max8	Min9	Min10	Max9	Max10	Min11	Min12	Max11	Max12	Min13	Min14	Max13	Max14	Min15	Min16	Max15	Max16	Min17	Min18	Max17	Max18	Min19	Min20	Max19	Max20	Min21	Min22	Max21	Max22	Min23	Min24	Max23	Max24	Min25	Min26	Max25	Max26	Min27	Min28	Max27	Max28	Min29	Min30	Max29	Max30	Min31	Min32	Max31	Max32	Min33	Min34	Max33	Max34	Min35	Min36	Max35	Max36	Min37	Min38	Max37	Max38	Min39	Min40	Max39	Max40	Min41	Min42	Max41	Max42	Min43	Min44	Max43	Max44	Min45	Min46	Max45	Max46	Min47	Min48	Max47	Max48	Min49	Min50	Max49	Max50	Min51	Min52	Max51	Max52	Min53	Min54	Max53	Max54	Min55	Min56	Max55	Max56	Min57	Min58	Max57	Max58	Min59	Min60	Max59	Max60	Min61	Min62	Max61	Max62	Min63	Min64	Max63	Max64	Min65	Min66	Max65	Max66	Min67	Min68	Max67	Max68	Min69	Min70	Max69	Max70	Min71	Min72	Max71	Max72	Min73	Min74	Max73	Max74	Min75	Min76	Max75	Max76	Min77	Min78	Max77	Max78	Min79	Min80	Max79	Max80	Min81	Min82	Max81	Max82	Min83	Min84	Max83	Max84	Min85	Min86	Max85	Max86	Min87	Min88	Max87	Max88	Min89	Min90	Max89	Max90	Min91	Min92	Max91	Max92	Min93	Min94	Max93	Max94	Min95	Min96	Max95	Max96	Min97	Min98	Max97	Max98	Min99	Min100	Max99	Max100	Min101	Min102	Max101	Max102	Min103	Min104	Max103	Max104	Min105	Min106	Max105	Max106	Min107	Min108	Max107	Max108	Min109	Min110	Max109	Max110	Min111	Min112	Max111	Max112	Min113	Min114	Max113	Max114	Min115	Min116	Max115	Max116	Min117	Min118	Max117	Max118	Min119	Min120	Max119	Max120	Min121	Min122	Max121	Max122	Min123	Min124	Max123	Max124	Min125	Min126	Max125	Max126	Min127	Min128	Max127	Max128	Min129	Min130	Max129	Max130	Min131	Min132	Max131	Max132	Min133	Min134	Max133	Max134	Min135	Min136	Max135	Max136	Min137	Min138	Max137	Max138	Min139	Min140	Max139	Max140	Min141	Min142	Max141	Max142	Min143	Min144	Max143	Max144	Min145	Min146	Max145	Max146	Min147	Min148	Max147	Max148	Min149	Min150	Max149	Max150	Min151	Min152	Max151	Max152	Min153	Min154	Max153	Max154	Min155	Min156	Max155	Max156	Min157	Min158	Max157	Max158	Min159	Min160	Max159	Max160	Min161	Min162	Max161	Max162	Min163	Min164	Max163	Max164	Min165	Min166	Max165	Max166	Min167	Min168	Max167	Max168	Min169	Min170	Max169	Max170	Min171	Min172	Max171	Max172	Min173	Min174	Max173	Max174	Min175	Min176	Max175	Max176	Min177	Min178	Max177	Max178	Min179	Min180	Max179	Max180	Min181	Min182	Max181	Max182	Min183	Min184	Max183	Max184	Min185	Min186	Max185	Max186	Min187	Min188	Max187	Max188	Min189	Min190	Max189	Max190	Min191	Min192	Max191	Max192	Min193	Min194	Max193	Max194	Min195	Min196	Max195	Max196	Min197	Min198	Max197	Max198	Min199	Min200	Max199	Max200	Min201	Min202	Max201	Max202	Min203	Min204	Max203	Max204	Min205	Min206	Max205	Max206	Min207	Min208	Max207	Max208	Min209	Min210	Max209	Max210	Min211	Min212	Max211	Max212	Min213	Min214	Max213	Max214	Min215	Min216	Max215	Max216	Min217	Min218	Max217	Max218	Min219	Min220	Max219	Max220	Min221	Min222	Max221	Max222	Min223	Min224	Max223	Max224	Min225	Min226	Max225	Max226	Min227	Min228	Max227	Max228	Min229	Min230	Max229	Max230	Min231	Min232	Max231	Max232	Min233	Min234	Max233	Max234	Min235	Min236	Max235	Max236	Min237	Min238	Max237	Max238	Min239	Min240	Max239	Max240	Min241	Min242	Max241	Max242	Min243	Min244	Max243	Max244	Min245	Min246	Max245	Max246	Min247	Min248	Max247	Max248	Min249	Min250	Max249	Max250	Min251	Min252	Max251	Max252	Min253	Min254	Max253	Max254	Min255	Min256	Max255	Max256	Min257	Min258	Max257	Max258	Min259	Min260	Max259	Max260	Min261	Min262	Max261	Max262	Min263	Min264	Max263	Max264	Min265	Min266	Max265	Max266	Min267	Min268	Max267	Max268	Min269	Min270	Max269	Max270	Min271	Min272	Max271	Max272	Min273	Min274	Max273	Max274	Min275	Min276	Max275	Max276	Min277	Min278	Max277	Max278	Min279	Min280	Max279	Max280	Min281	Min282	Max281	Max282	Min283	Min284	Max283	Max284	Min285	Min286	Max285	Max286	Min287	Min288	Max287	Max288	Min289	Min290	Max289	Max290	Min291	Min292	Max291	Max292	Min293	Min294	Max293	Max294	Min295	Min296	Max295	Max296	Min297	Min298	Max297	Max298	Min299	Min300	Max299	Max300	Min301	Min302	Max301	Max302	Min303	Min304	Max303	Max304	Min305	Min306	Max305	Max306	Min307	Min308	Max307	Max308	Min309	Min310	Max309	Max310	Min311	Min312	Max311	Max312	Min313	Min314	Max313	Max314	Min315	Min316	Max315	Max316	Min317	Min318	Max317	Max318	Min319	Min320	Max319	Max320	Min321	Min322	Max321	Max322	Min323	Min324	Max323	Max324	Min325	Min326	Max325	Max326	Min327	Min328	Max327	Max328	Min329	Min330	Max329	Max330	Min331	Min332	Max331	Max332	Min333	Min334	Max333	Max334	Min335	Min336	Max335	Max336	Min337	Min338	Max337	Max338	Min339	Min340	Max339	Max340	Min341	Min342	Max341	Max342	Min343	Min344	Max343	Max344	Min345	Min346	Max345	Max346	Min347	Min348	Max347	Max348	Min349	Min350	Max349	Max350	Min351	Min352	Max351	Max352	Min353	Min354	Max353	Max354	Min355	Min356	Max355	Max356	Min357	Min358	Max357	Max358	Min359	Min360	Max359	Max360	Min361	Min362	Max361	Max362	Min363	Min364	Max363	Max364	Min365	Min366	Max365	Max366	Min367	Min368	Max367	Max368	Min369	Min370	Max369	Max370	Min371	Min372	Max371	Max372	Min373	Min374	Max373	Max374	Min375	Min376	Max375	Max376	Min377	Min378	Max377	Max378	Min379	Min380	Max379	Max380	Min381	Min382	Max381	Max382	Min383	Min384	Max383	Max384	Min385	Min386	Max385	Max386	Min387	Min388	Max387	Max388	Min389	Min390	Max389	Max390	Min391	Min392	Max391	Max392	Min393	Min394	Max393	Max394	Min395	Min396	Max395	Max396	Min397	Min398	Max397	Max398	Min399	Min400	Max399	Max400	Min401	Min402	Max401	Max402	Min403	Min404	Max403	Max404	Min405	Min406	Max405	Max406	Min407	Min408	Max407	Max408	Min409	Min410	Max409	Max410	Min411	Min412	Max411	Max412	Min413	Min414	Max413	Max414	Min415	Min416	Max415	Max416	Min417	Min418	Max417	Max418	Min419	Min420	Max419	Max420	Min421	Min422	Max421	Max422	Min423	Min424	Max423	Max424	Min425	Min426	Max425	Max426	Min427	Min428	Max427	Max428	Min429	Min430	Max429	Max430	Min431	Min432	Max431	Max432	Min433	Min434	Max433	Max434	Min435	Min436	Max435	Max436	Min437	Min438	Max437	Max438	Min439	Min440	Max439	Max440	Min441	Min442	Max441	Max442	Min443	Min444	Max443	Max444	Min445	Min446	Max445	Max446	Min447	Min448	Max447	Max448	Min449	Min450	Max449	Max450	Min451	Min452	Max451	Max452	Min453	Min454	Max453	Max454	Min455	Min456	Max455	Max456	Min457	Min458	Max457	Max458	Min459	Min460	Max459	Max460	Min461	Min462	Max461	Max462	Min463	Min464	Max463	Max464	Min465	Min466	Max465	Max466	Min467	Min468	Max467	Max468	Min469	Min470	Max469	Max470	Min471	Min472	Max471	Max472	Min473	Min474	Max473	Max474	Min475	Min476	Max475	Max476	Min477	Min478	Max477	Max478	Min479	Min480	Max479	Max480	Min481	Min482	Max481	Max482	Min483	Min484	Max483	Max484	Min485	Min486	Max485	Max486	Min487	Min488	Max487	Max488	Min489	Min490	Max489	Max490	Min491	Min492	Max491	Max492	Min493	Min494	Max493	Max494	Min495	Min496	Max495	Max496	Min497	Min498	Max497	Max498	Min499	Min500	Max499	Max500	Min501	Min502	Max501	Max502	Min503	Min504	Max503	Max504	Min505	Min506	Max505	Max506	Min507	Min508	Max507	Max508	Min509	Min510	Max509	Max510	Min511	Min512	Max511	Max512	Min513	Min514	Max513	Max514	Min515	Min516	Max515	Max516	Min517	Min518	Max517	Max518	Min519	Min520	Max519	Max520	Min521	Min522	Max521	Max522	Min523	Min524	Max523	Max524	Min525	Min526	Max525	Max526	Min527	Min528	Max527	Max528	Min529	Min530	Max529	Max530	Min531	Min532	Max531	Max532	Min533	Min534	Max533	Max534	Min535	Min536	Max535	Max536	Min537	Min538	Max537	Max538	Min539	Min540	Max539	Max540	Min541	Min542	Max541	Max542	Min543	Min544	Max543	Max544	Min545	Min546	Max545	Max546	Min547	Min548	Max547	Max548	Min549	Min550	Max549	Max550	Min551	Min552	Max551	Max552	Min553	Min554	Max553	Max554	Min555	Min556	Max555	Max556	Min557	Min558	Max557	Max558	Min559	Min560	Max559	Max560	Min561	Min562	Max561	Max562	Min563	Min564	Max563	Max564	Min565	Min566	Max565	Max566	Min567	Min568	Max567	Max568	Min569	Min570	Max569	Max570	Min571	Min572	Max571	Max572	Min573	Min574	Max573	Max574	Min575	Min576	Max575	Max576	Min577	Min578	Max577	Max578	Min579	Min580	Max579	Max580	Min581	Min582	Max581	Max582	Min583	Min584	Max583	Max584	Min585	Min586	Max585	Max586	Min587	Min588	Max587	Max588	Min589	Min590	Max589	Max590	Min591	Min592	Max591	Max592	Min593	Min594	Max593	Max594	Min595	Min596	Max595	Max596	Min597	Min598	Max597	Max598	Min599	Min600	Max599	Max600	Min601	Min602	Max601	Max602	Min603	Min604	Max603	Max604	Min605	Min606	Max605	Max606	Min607	Min608	Max607	Max608	Min609	Min610	Max609	Max610	Min611	Min612	Max611	Max612	Min613	Min614	Max613	Max614	Min615	Min616	Max615	Max616	Min617	Min618	Max617	Max618	Min619	Min620	Max619	Max620	Min621	Min622	Max621	Max622	Min623	Min624	Max623	Max624	Min625	Min626	Max625	Max626	Min627	Min628	Max627	Max628	Min629	Min630	Max629	Max630	Min631	Min632	Max631	Max632	Min633	Min634	Max633	Max634	Min635	Min636	Max635	Max636	Min637	Min638	Max637	Max638	Min639	Min640	Max639	Max640	Min641	Min642	Max641	Max642	Min643	Min644	Max643	Max644	Min645	Min646	Max645	Max646	Min647	Min648	Max647	Max648	Min649	Min650	Max649	Max650	Min651	Min652	Max651	Max652	Min653	Min654	Max653	Max654	Min655	Min656	Max655	Max656	Min657	Min658	Max657	Max658	Min659	Min660	Max659	Max660	Min661	Min662	Max661	Max662	Min663	Min664	Max663	Max664	Min665	Min666	Max665	Max666	Min667	Min668	Max667	Max668	Min669	Min670	Max669	Max670	Min671	Min672	Max671	Max672	Min673	Min674	Max673	Max674	Min675	Min676	Max675	Max676	Min677	Min678	Max677	Max678	Min679	Min680	Max679	Max680	Min681	Min682	Max681	Max682	Min683	Min684	Max683	Max684	Min685	Min686	Max685	Max686	Min687	Min688	Max687	Max688	Min689	Min690	Max689	Max690	Min691	Min692	Max691	Max692	Min693	Min694	Max693	Max694
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Figura 29 - Treinamento da rede neural no MBP



Fonte: Autor

Foi obtido um RMS de treino igual a 0,046 e um RMS de teste de 0,091. O treinamento da rede neural convergiu muito rapidamente e não houve sobreaprendizado. O RMS obtido mostra que a rede neural gerada é de qualidade e válida para aplicação embarcada.

4.7 CRITÉRIOS PARA DETECÇÃO DE BURACO

Nesta prova de conceito, foi adicionado um algoritmo de detecção em tempo real de buracos com base nas referências bibliográficas, usando métodos apresentados em [7], [8], [9] e [10]. Em resumo, este algoritmo executa um loop também dentro de uma janela de 3s com 50 amostras e avalia os seguintes parâmetros para avaliar se o carro passou ou não em um buraco em tempo real:

- Tempo entre Detecções;
- Amplitude Total Máxima da Aceleração Horizontal X;
- Amplitude Positiva Máxima da Aceleração Vertical Y;

- Amplitude Negativa Mínima da Aceleração Vertical Y;
- Ordem entre os pontos de Máximo e Mínimo da Aceleração Vertical Y.

A prova de conceito desta detecção foi testada em conjunto com o índice de qualidade do pavimento, fornecendo as seguintes medidas de qualidade apresentadas pela matriz de confusão da Figura 30:

Figura 30 - Matriz de confusão para a detecção de buracos

		Valor Verdadeiro	
		Positivo	Negativo
Valor Previsto	Positivo	Verdadeiro Positivo VP = 30	Falso Positivo FP = 13
	Negativo	Falso Negativo FN = 4	Verdadeiro Negativo VN = 759

Trajetos de 40 minutos

Fonte: Autor

A partir desta matriz de confusão, podemos calcular os seguintes parâmetros:

- **Sensibilidade** = $VP / (VP + FN) = 89,19\%$
- **Precisão** = $VP / (VP + FP) = 76,74\%$
- **Especificidade** = $VN / (VN + FP) = 98,70\%$
- **Taxa de Acerto** = $(VP + VN) / (VP + VN + FP + FN) = 98,26\%$

Observa-se que a qualidade obtida se encontra acima do esperado, de acordo com os requisitos de engenharia estabelecidos anteriormente no projeto.

4.8 PÓS-PROCESSAMENTO DE DADOS

Ao final da rota de teste e validação da prova de conceito, foi salvo um arquivo .csv com todos os índices de qualidade do pavimento coletados. Este arquivo foi

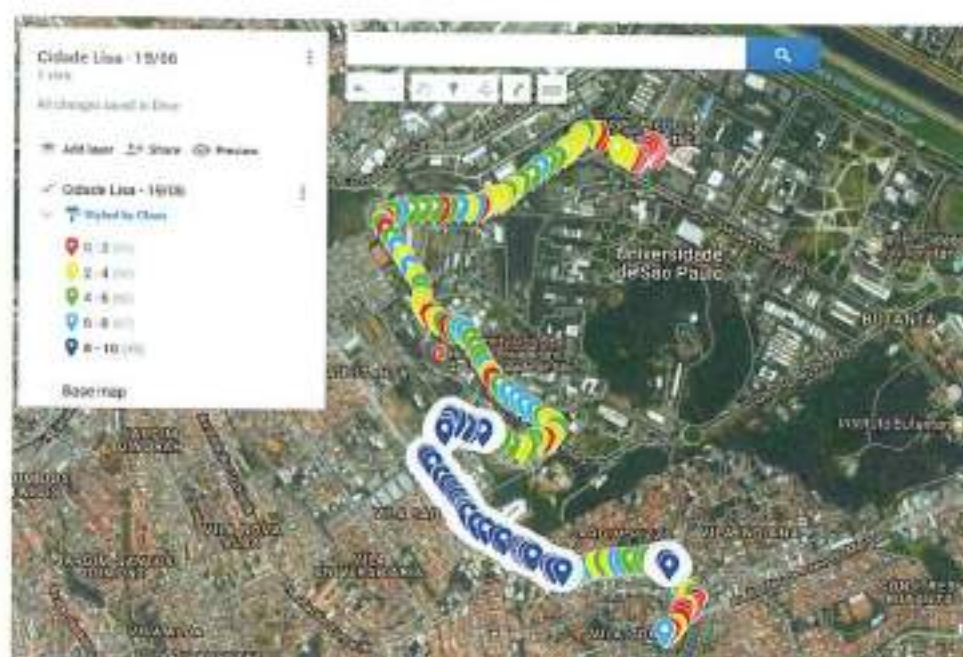
importado no Google Maps e foi possível assim visualizar em um mapa de escala colorida a qualidade do pavimento por onde o veículo passou (Figura 31 e Figura 32).

Figura 31 - Visualização da qualidade do pavimento em um mapa – Av. Corifeu



Fonte: Autor

Figura 32 - Visualização da qualidade do pavimento em um mapa – USP



Fonte: Autor

Além disso, temos também a opção de visualizar os buracos encontrados no caminho e detectados em tempo real, conforme a Figura 33.

Figura 33 - Visualização dos buracos encontrados no trajeto dentro da USP



Fonte: Autor

5 RESULTADOS FINAIS

A prova de conceito do projeto consistia basicamente em uma única tela que coletava os dados do trajeto, mostrava em tempo real o resultado e salvava uma compilação dos dados na memória SD do smartphone, que é de difícil acesso. Para chegarmos em um estágio de distribuição do produto, foi necessário o desenvolvimento de uma série de etapas de estruturação do projeto, como previsto no cronograma de atividades. Elas se dividem principalmente em desenvolvimento do aplicativo, do servidor, da base de dados e todas as interfaces que ligam estes elementos.

5.1 DESENVOLVIMENTO DO APLICATIVO

O aplicativo Cidade Lisa conta com um ícone na tela Home do smartphone que contém o logotipo do projeto (Figura 34). Ao iniciar o aplicativo, temos uma tela com o menu principal do app. Nele é possível:

Figura 34 - Tela inicial do aplicativo



Fonte: Autor

- **Novo Trajeto:** Coleta automaticamente os dados de qualidade e de buracos das vias, que são enviados à base de dados do projeto Cidade Lisa;
- **Meus Trajetos:** Visualiza o histórico de trajetos salvos no aplicativo e seleciona um para ver no mapa, sendo possível tanto o mapa de qualidade como o de buracos;
- **Mapa da Cidade:** Visualiza o mapa de qualidade ou de buracos com os dados coletados por todos os usuários que contribuíram para o projeto, sendo este o principal objetivo e forma de entrega do resultado do projeto;
- **Treino Neural:** Possibilita o treino de uma nova rede neural a ser treinada a partir do arquivo salva na memória SD do smartphone. Esta opção é restrita ao desenvolvimento do projeto e não estará abertamente visível no produto a ser distribuído.

5.1.1 Novo Trajeto

Assim que a tela de Novo Trajeto é aberta, é iniciado o processo de coleta de dados (Figura 35). Como medida de robustez do projeto, os primeiros 5 segundos de coleta são desprezados. Após este curto período, os dados do GPS são recebidos a cada 0,5 segundos e os dados dos 3 eixos do acelerômetro são coletados a cada 60 ms.

Estes dados dos sensores são salvos em vetores temporários. Além disto, uma série de cálculos são executados em tempo real dentro de uma janela com intervalo de 3 segundos, produzindo como resultado as “features” que servem como entrada na rede neural.

Desta forma, se a velocidade do veículo em movimento superar os 20 km/h, ele passa a reportar para o servidor os resultados da coleta. Esta condição foi necessária a fim de garantir uma qualidade ainda melhor nos resultados.

Sob esta condição, a cada 3 segundos o aplicativo reporta um resultado de qualidade observada. Ele acessa a base de dados e salva o ponto em questão, incluindo sua identificação de usuário. Da mesma forma, quando um buraco é

detectado no trajeto, o aplicativo reporta os dados enviando o ponto para a base de dados, porém em uma outra tabela.

Figura 35 - Coleta de dados do trajeto



Fonte: Autor

Ao clicar no botão "Salvar", o procedimento é finalizado e o usuário do aplicativo retorna ao menu principal.

5.1.2 Meus Trajetos

A segunda opção do menu principal é o item Meus Trajetos (Figura 36). Dentro desta opção, pode-se escolher através de uma janela pop-up a visualização do mapa de qualidade ou do mapa de buracos. Ao clicar, temos a respectiva lista completa com os trajetos salvos pelo usuário no aplicativo, sendo que o trajeto salvo mais recente se encontra no topo.

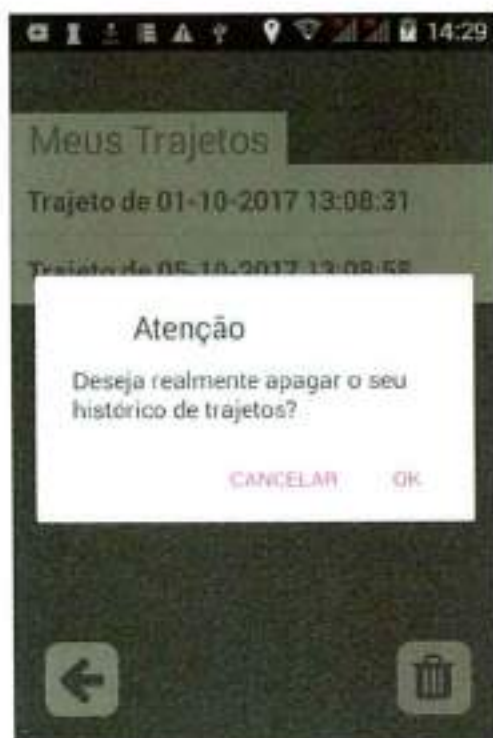
Figura 36 - Meus trajetos



Fonte: Autor

Nesta tela também é possível deletar todos os trajetos ao clicarmos no botão com o símbolo de uma lixeira (Figura 37). Isto apaga os dados apenas da memória interna do celular e não da base de dados. Esta opção existe para liberar espaço de memória consumida pelo aplicativo.

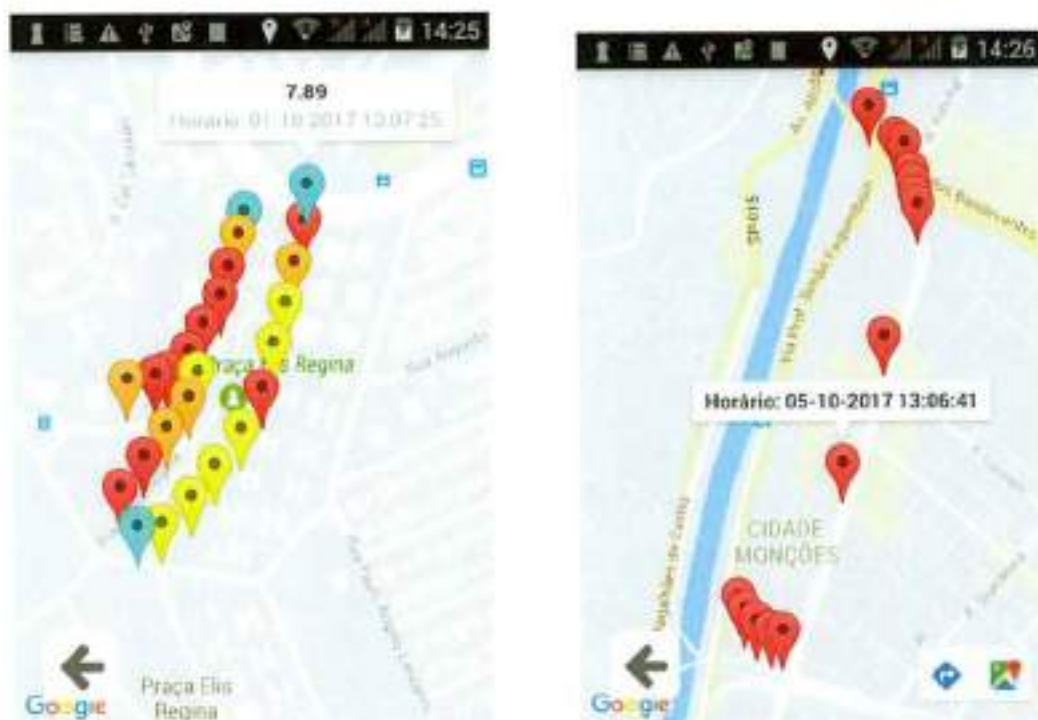
Figura 37 - Apagar trajetos



Fonte: Autor

Escolhendo o trajeto específico, é aberto o mapa com todos os pontos salvos (Figura 38). Ao clicar em um ponto, uma caixa de texto indica o horário preciso em que houve a detecção. No caso no mapa de qualidade à esquerda, a escala de cor indica a qualidade, sendo o vermelho o pior e azul o de melhor qualidade. O valor exato da escala PASER obtido pode ser visto ao clicarmos sobre o ponto.

Figura 38 - Visualização no mapa dos meus trajetos



Fonte: Autor

5.1.3 Mapa da Cidade

A terceira opção do menu principal é o item Mapa da Cidade. Dentro desta opção, pode-se escolher através de uma janela pop-up a visualização do mapa de qualidade ou do mapa de buracos (Figura 39). Ao clicar, o aplicativo carrega no mapa todo os pontos que se encontram na base de dados. Desta forma, os pontos coletados por diferentes usuários ao longo do tempo são compilados e mostrados juntos no mapa (Figuras 40 e 41).

Figura 39 - Escolha do tipo de mapa da cidade



Fonte: Autor

Figura 40 - Mapa da cidade com buracos



Fonte: Autor

Figura 41 - Mapa da cidade com qualidade



Fonte: Autor

5.1.4 Treino Neural

A quarta e última opção do menu principal possibilita o treino de uma nova rede neural a ser treinada a partir do arquivo salva na memória SD do smartphone (Figura 42). Esta opção é restrita ao desenvolvimento do projeto e não estará abertamente visível no produto a ser distribuído.

Nesta opção, ao clicarmos em um dos botões, um arquivo que contém todas as informações necessárias para o treino da rede neural é salvo na memória. O nome do arquivo contém a descrição do respectivo botão.

Figura 42 - Opção de treino da rede neural



Fonte: Autor

5.2 DESENVOLVIMENTO DA BASE DE DADOS

Foi comprado pela HostGator.com um servidor e o domínio de endereço www.grivol.com. Além de possibilitar hospedagem de um site, também é possível criar nele bases de dados e outras ferramentas diretamente no painel de controle (Figura 43).

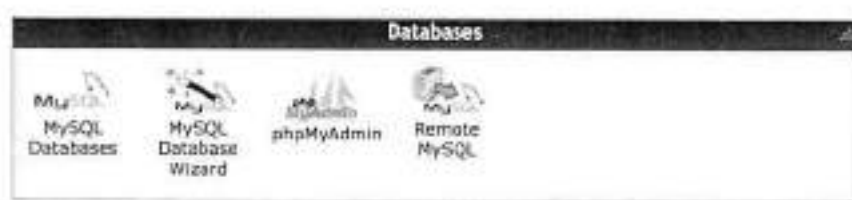
A ferramenta utilizada para a criação da base de dados do projeto Cidade Lisa foi MySQL Database Wizard, enquanto para a gestão foi utilizado phpMyAdmin (Figura 44). Desta forma, foi escolhido o uso de uma base de dados do tipo relacional SQL, mais especificamente MySQL.

Figura 43 - Painel de Controle do HostGator.com



Fonte: Autor

Figura 44 - Gestão da base de dado no domínio

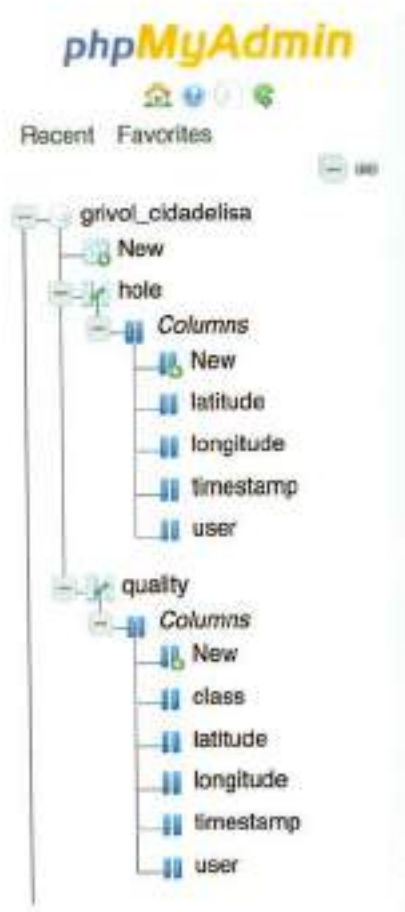


Fonte: Autor

Foi criada uma base de dados chamada `grivol_cidadelisa` (Figura 45), contendo duas tabelas: uma para armazenar os dados de qualidade (chamada `quality`) e outra para os dados de buracos encontrados nas vias (chamada `hole`).

Cada uma delas apresenta colunas de latitude, longitude, identidade do usuário e data. A tabela de qualidade apresenta uma coluna suplementar que contém o valor de qualidade da escala PASER, que vai de 0 a 10. Desta forma, foram configurados os tipos corretos das variáveis necessárias para as tabelas.

Figura 45 - Base de Dados do projeto Cidade Lisa



Fonte: Autor

Dentro da ferramenta phpMyAdmin, é possível editar a estrutura da base de dados (Figura 46), administrar configurações avançadas e inclusive visualizar os dados armazenados (Figura 47).

Figura 46 - Edição da estrutura da base MySQL

The screenshot shows the MySQL Workbench interface for editing the structure of a table named 'user'. The table has five columns: 'user' (varchar(10), primary key), 'timestamp' (varchar(20)), 'latitude' (varchar(10)), 'longitude' (varchar(10)), and 'class' (varchar(10)). The 'user' column is the primary key and has a unique index. The 'timestamp' column has a unique index. The 'latitude' and 'longitude' columns have unique indexes. The 'class' column has a unique index. The 'user' column is also the primary key of the table.

Below the table structure, the 'Information' tab is selected, showing the following data:

Space usage		Row statistics	
Data	21.1 KB	Format	dynamic
Index	1 KB	Collation	latin1_general_ci
Total	22.1 KB	Rows	100
		Row length	100 B
		Row size	100 B
		Creation	09-10-2017 13:05:08
		Last update	09-10-2017 13:05:08
		Last check	09-10-2017 13:05:08

Fonte: Autor

Figura 47 - Visualização dos dados armazenados na base

user	timestamp	latitude	longitude	class
466928017d47ace5	01-10-2017 13:05:08	-23.576281888888887	-46.73047499999999	2.0
466928017d47ace5	01-10-2017 13:06:05	-23.576408333333333	-46.730628888888886	2.11
466928017d47ace5	30-09-2017-18:58:26	-23.576347	-46.732886	8.9
466928017d47ace5	30-09-2017-18:58:26	-23.576312	-46.732612	6.0
466928017d47ace5	30-09-2017-18:58:26	-23.576324	-46.732317	6.0
466928017d47ace5	30-09-2017-18:58:26	-23.576393	-46.731377	2.3
466928017d47ace5	01-10-2017 13:06:11	-23.576110000000003	-46.73018888888889	2.06
466928017d47ace5	01-10-2017 13:06:14	-23.575915000000002	-46.73024166666667	2.01
466928017d47ace5	01-10-2017 13:06:17	-23.575735	-46.730188333333334	2.0
466928017d47ace5	01-10-2017 13:06:20	-23.575619999999997	-46.73011499999999	4.93
466928017d47ace5	01-10-2017 13:06:23	-23.57537	-46.730060333333335	8.34
466928017d47ace5	01-10-2017 13:07:25	-23.57518	-46.729621888888886	7.89
466928017d47ace5	01-10-2017 13:07:28	-23.575423333333333	-46.72964	2.0
466928017d47ace5	01-10-2017 13:07:31	-23.575703333333333	-46.729715000000005	3.32
466928017d47ace5	01-10-2017 13:07:35	-23.575988888888886	-46.72978499999999	5.61
466928017d47ace5	01-10-2017 13:07:38	-23.576234999999997	-46.72987	5.34
466928017d47ace5	01-10-2017 13:07:41	-23.576523333333333	-46.72995666666667	2.03
466928017d47ace5	01-10-2017 13:07:44	-23.576793333333333	-46.730101888888886	5.4
466928017d47ace5	01-10-2017 13:07:47	-23.577028888888886	-46.730296666666666	6.58
466928017d47ace5	01-10-2017 13:07:50	-23.577231888888886	-46.730471666666666	6.22
466928017d47ace5	01-10-2017 13:07:53	-23.577414999999999	-46.730678333333333	5.88
466928017d47ace5	01-10-2017 13:07:56	-23.577429999999996	-46.730853333333336	8.68
466928017d47ace5	01-10-2017 13:08:06	-23.577181555555556	-46.730988888888886	2.0
466928017d47ace5	01-10-2017 13:08:09	-23.576971888888887	-46.730798333333333	2.03
466928017d47ace5	01-10-2017 13:08:12	-23.576788333333333	-46.730643333333333	3.2

Fonte: Autor

Além disso, a ferramenta possibilita a criação de buscas avançadas direto do navegador (Figura 48). Neste exemplo da Figura 48, podemos observar todos os pontos coletados que apresentam uma qualidade inferior a 4 na escala PASER.

Figura 48 - Busca personalizada de dados armazenados na base

Showing rows 0 - 24 (50 total, Query took 0.0004 seconds.)

SELECT * FROM 'quality' WHERE 'class' < 4.0

1 > >> Show all Number of rows: 25 Filter rows: Search this table

+ Options

user	timestamp	latitude	longitude	class
466928017d47ace5	01-10-2017 13:06:08	-23.576281666666667	-46.730474999999999	2.0
466928017d47ace5	01-10-2017 13:06:06	-23.576408333333333	-46.730526666666666	2.11
466928017d47ace5	30-09-2017-18:58:28	-23.576393	-46.731877	2.3
466928017d47ace5	01-10-2017 13:06:11	-23.576110000000003	-46.730699999999999	2.06
466928017d47ace5	01-10-2017 13:06:14	-23.575915000000002	-46.730241666666667	2.01
466928017d47ace5	01-10-2017 13:06:17	-23.575735	-46.730188333333334	2.0
466928017d47ace5	01-10-2017 13:07:28	-23.575423333333333	-46.72964	2.0
466928017d47ace5	01-10-2017 13:07:31	-23.575703333333333	-46.729715000000006	3.32
466928017d47ace5	01-10-2017 13:07:41	-23.576523333333333	-46.729586666666667	2.03
466928017d47ace5	01-10-2017 13:08:06	-23.577181666666668	-46.730866666666667	2.0
466928017d47ace5	01-10-2017 13:08:09	-23.576971666666667	-46.730798333333333	2.03
466928017d47ace5	01-10-2017 13:08:12	-23.576768333333337	-46.730643333333333	3.2
466928017d47ace5	01-10-2017 13:08:23	-23.576435	-46.73071	2.0

Fonte: Autor

Este procedimento é útil no projeto porque permite a busca por dados de uma certa classe presente em uma região específica por exemplo. O resultado pode inclusive ser exportado para outros formatos com ótima compatibilidade, como planilhas por exemplo.

5.3 DESENVOLVIMENTO DO SERVIDOR

O elemento que conecta o aplicativo Android à base de dados é a página PHP (Figura 49) que foi colocada no servidor usando o programa FileZilla (Figura 50).

Figura 49 - Página PHP de conexão do app com a base de dados



Fonte: Autor

6 ANÁLISE

6.1 ANÁLISE DOS RESULTADOS DE MARKETING

Após a implementação do projeto Cidade Lisa e o seu uso extensivo em testes de validação em campo, pode-se realizar uma avaliação dos resultados sob a perspectiva dos requisitos de marketing que foram estabelecidos na fase de concepção do projeto:

1. **Peso leve:** O peso do produto é na realidade o próprio peso do smartphone no qual o aplicativo foi instalado, o que fica a critério do usuário. Sendo assim, pode-se dizer que o produto final apresentará entre 120 e 200 gramas de peso total, que é a faixa média dos smartphones comercializados atualmente no mercado. Isto quer dizer que este requisito é respeitado com uma boa margem.
2. **Tamanho pequeno:** Assim como o peso, o tamanho do produto final equivale ao tamanho do próprio smartphone no qual o aplicativo foi instalado, o que fica a critério do usuário. Sendo assim, pode-se dizer que a tela do produto final apresentará um tamanho entre 4,5 e 5,5 polegadas, que é a faixa média dos smartphones comercializados atualmente no mercado. Isto quer dizer que o produto pode ser facilmente transportado e instalado em diferentes veículos.
3. **Poucas peças:** O produto final apresenta uma única peça, que no caso é o smartphone onde o aplicativo foi instalado. O smartphone é fixado no veículo com um suporte. Eventualmente ele será carregado no acendedor de cigarro do veículo. Desta forma, o requisito de poucas peças é atendido integralmente.
4. **Boa ergonomia:** A presença do smartphone no veículo não prejudica a ergonomia do veículo e a sua interface com o usuário foi inteiramente desenvolvida no sentido de agregar valor à experiência do usuário. O único requisito necessário para a instalação do smartphone durante a coleta de dados é o suporte que o fixe verticalmente no veículo em movimento, o que representa uma adequação ao conceito de boa ergonomia e uma invasão praticamente nula.

5. **Respeito à privacidade:** Não há necessidade de cadastro prévio para utilizar o aplicativo e todas as informações enviadas ao servidor são anônimas. Em nenhum momento outros usuários conseguem acessar dados privados sobre os outros colaboradores do projeto, sendo que a única informação que identifica a origem dos dados é um número serial que é único para cada smartphone e aplicativo instalado. Este número serial não tem nenhuma utilidade prática para terceiros, é usado apenas para controle interno e cruzamento de dados entre diferentes usuários.
6. **Baixo consumo de energia:** Após extensivos testes de exaustão no uso do aplicativo, foi possível observar que o smartphone apresenta uma boa autonomia durante o uso, mesmo com a dependência no uso do GPS e modem para troca de dados com o servidor. Um fato que melhora ainda mais no baixo consumo é a redução da luminosidade da tela, que foi o maior responsável pelo consumo energético. Executando exclusivamente o aplicativo Cidade Lisa, pode-se concluir que ele foi responsável por apenas 26% do consumo da bateria durante o trajeto, sendo a maior parte devido ao próprio sistema Android. Este requisito foi bem atendido e pode ficar ainda melhor levando em consideração a possibilidade de carregamento do smartphone no acendedor de cigarro do veículo durante o próprio trajeto.
7. **Apresenta poucas etapas:** Para ativar a coleta de dados, basta selecionar o ícone do aplicativo Cidade Lisa na tela principal e clicar em "Novo Trajeto". Ao final do percurso, basta clicar em "Finalizar Trajeto" para fechar a tela de coleta e encerrar o procedimento. Isto representa um total de 3 cliques ao longo de todo o processo, o que atende ao requisito com um número mínimo de etapas. Todo o processo de coleta em si é inteiramente autônomo e não necessita de qualquer interferência do usuário.
8. **Não distrai o motorista:** O aplicativo Cidade Lisa não emite sons nem animações durante a coleta de dados no trajeto, garantindo o requisito de não distrair o motorista e garantindo assim a segurança. Além disso, a coleta é inteiramente autônoma e em nenhum momento necessita da intervenção do motorista. A tela durante a coleta exibe simplesmente indicadores numéricos em tempo real que são discretos e não atrapalham o motorista.
9. **Baixo custo:** O aplicativo em si ficará disponível gratuitamente para os usuários baixarem. Quanto ao custo de desenvolvimento, ele foi

extremamente baixo por ter sido desenvolvido individualmente e sem necessidade de compras. O único custo que existirá em trabalhos futuros é o de manutenção do domínio e do servidor onde a base de dados fica armazenada, que é um custo baixíssimo com relação ao impacto positivo gerado pelo projeto.

- 10. Boa confiabilidade:** O aplicativo funciona de uma forma extremamente robusta a falhas. Além disso, por ter etapas e opções muito bem definidas e claras, transmite uma sensação de confiabilidade aos usuários. Com relação à robustez, pode-se citar por exemplo a verificação de validade da posição geográfica antes do envio ao servidor, realinhamento das coordenadas do acelerômetro ao longo do trajeto, aviso de não existência de pontos em um trajeto coletado, etc.
- 11. Boa precisão:** A precisão dos sensores foi ajustada tendo como objetivo maximizar a qualidade dos resultados sem prejudicar o trade-off com outros requisitos importantes para o projeto, como o consumo de bateria. A precisão obtida no produto final foi decididamente de ótima qualidade e o requisito amplamente atendido. Pode-se afirmar isso primeiramente pela precisão obtida com os dados geográficos, que se encontram efetivamente centralizados na pista por onde o veículo passou e correspondem realmente aos pontos de interesse. Além disso, a granularidade do acelerômetro foi o suficiente para atender às expectativas do usuário na detecção de buracos e avaliação da qualidade das vias.
- 12. Coleta eficiente de dados:** A coleta autônoma de dados apresenta uma alta eficiência tanto pela necessidade nula de intervenção do motorista como pelo uso inteligente dos recursos, através de uma coleta em intervalos bem definidos ou em caso de uma detecção específica.
- 13. Fácil envio de dados:** Os dados são enviados de forma assíncrona para não congestionar o buffer de dados. Desta forma é garantido que não haverá uma grande perda de dados como seria no caso de envio de todos os dados ao final do trajeto, havendo a possibilidade de perda de conexão celular.
- 14. Interface amigável:** A interface agrega valor à experiência do usuário e é convidativa. Detalhes como fundo de tela que remete à imagem de uma rua lisa, slogan do projeto, botões arredondados e ícones bem representativos tornam a interface amigável e cumprem as expectativas.

6.2 ANÁLISE DOS RESULTADOS DE ENGENHARIA

- A. Tempo de resposta:** O tempo de resposta do aplicativo é sempre igual ou inferior a 3 segundos, pela construção da própria solução. A rede neural que detecta as irregularidades recebe como entrada os dados do acelerômetro obtidos em uma janela de 3 segundos (equivalente a 50 pontos com intervalo de 60 ms). Como resultado, sempre teremos um tempo de resposta inferior a 3 segundos e com uma média de 1,5 segundos (centro da janela temporal de detecção).
- B. Precisão na posição:** A precisão estática do GPS é por definição equivalente 4,9 metros [31]. Levando em consideração velocidades entre 0 e 90 km/h (velocidade limite em ambientes urbanos), um tempo médio de resposta de 1,5 segundos (centro da janela temporal de entrada) e o uso da posição geográfica obtido no centro da janela temporal, temos no pior caso uma precisão equivalente a 23,65 metros, que é inferior aos 30 metros estabelecidos inicialmente. Desta forma, o requisito é atendido com uma margem de segurança.
- C. Taxa de acerto:** Após os testes de validação do produto, foi obtida uma sensibilidade de 89,19%, precisão de 76,74%, especificidade de 98,70% e taxa de acerto de 98,26%. Isto é, a taxa de acerto individual por si só já é superior à taxa de acerto coletiva de 85% estabelecida inicialmente como requisito de engenharia e aos 75% individuais.
- D. Consumo de energia:** O smartphone usado na validação foi o Motorola RAZR D1 XT918, que apresenta 1785 mAh de bateria. Nos testes, foi usado exclusivamente o aplicativo Cidade Lisa. O tempo médio de duração total da bateria nos testes exaustivos foi de 615 minutos, o que equivale a 10,26 horas. O consumo de energia foi, portanto, de 174 mA, que é bem inferior ao estipulado como requisito de engenharia de 800 mA.
- E. Dimensionamento limite:** Qualquer smartphone Android é mais leve do que 1 kg e apresenta dimensões inferiores a 30 cm, portanto este requisito foi amplamente respeitado.
- F. Número de peças:** O produto final apresenta uma única peça, que no caso é o smartphone onde o aplicativo foi instalado. O smartphone é fixado no veículo com um suporte. Eventualmente ele será carregado no acendedor de

cigarro do veículo. Desta forma, no pior caso teremos um total de 3 conexões, que é inferior às 4 conexões estipuladas como requisito de engenharia.

- G. Etapa de privacidade:** Ao iniciar o aplicativo pela primeira vez é solicitada a permissão para a coleta de dados do GPS e acelerômetro do smartphone. Isto é feito através do uso de permissões que foram incluídas no Android Manifest, garantindo a privacidade e a concordância com os termos de uso.
- H. Design:** O produto final não é invasivo porque está instalado no próprio smartphone do colaborador, sem a necessidade de equipamento extra no veículo. Não causa desconforto nem desestimula porque o uso do aparelho já faz parte de sua rotina. Já a segurança é respeitada através do processo de coleta de dados autônoma e com poucas etapas de ativação.
- I. Coleta autônoma:** A coleta é inteiramente autônoma e não depende da intervenção do motorista em momento nenhum. Além disso, o aplicativo em execução não emite sons nem exibe imagens que possam causar distração.
- J. Custo limite:** O custo total fixo de desenvolvimento não excedeu os 500 reais estipulados, sendo que o custo efetivo total foi de 401 reais (servidor e domínio web). Já o custo marginal para novos usuários será nulo, porque toda a infraestrutura será o bastante para manter o volume previsto de usuários na plataforma.
- K. Tempo de envio:** Devido à escolha de realizar a transmissão dos dados em rajadas e de forma assíncrona, o tempo de envio é praticamente nulo e definitivamente bem inferior aos 10 segundos estipulados inicialmente. Esta decisão foi tomada para garantir a robustez do sistema e também para evitar um tempo de espera excessivo ao final da coleta.
- L. Interface Mínima:** Toda a interface do aplicativo foi desenhada desde o início para ser minimalista e permitir o uso de pessoas provenientes de diferentes contextos, não apenas especialistas técnicos. Conhecimentos técnicos prévios não são requeridos. Os comandos foram reduzidos ao mínimo necessário e o produto final é responsivo. O design também é amigável e remete ao objetivo do projeto de melhorar a infraestrutura das vias.

7 CONCLUSÃO

O objetivo do projeto Cidade Lisa de desenvolver um sistema automático de detecção de irregularidades nas vias foi atingido. O produto final entregue pelo projeto é o aplicativo Android, a plataforma que permite o funcionamento do sistema e principalmente o algoritmo de detecção automática da qualidade, que realiza operações em tempo real usando redes neurais.

O impacto do projeto na sociedade é a prova de que é sim possível a disponibilização na forma digital das informações relevantes para os órgãos públicos competentes realizarem o reparo adequado. Com isto, será possível agilizar o processo e reduzir o custo das obras através de uma fiscalização constante por parte da população com o uso do sistema desenvolvido pelo projeto Cidade Lisa.

Um sistema automático de detecção de irregularidades poderá garantir o reparo eficiente da infraestrutura urbana, atendendo às necessidades da população que utiliza a malha diariamente. Por ser de fácil instalação, de simples utilização, preciso, barato e satisfatório, o aplicativo Cidade Lisa oferece uma ótima alternativa para as grandes metrópoles, permitindo o uso em larga escala, de forma frequente e com um nível satisfatório de qualidade.

De acordo com a Prefeitura de São Paulo, em 2015 foram utilizados R\$ 110 milhões para recapear 135,5 quilômetros de vias e tapar 294.419 buracos na cidade [14], que apresenta uma malha viária de mais de 17 mil quilômetros de vias pavimentadas. Será possível manter o ritmo de recapeamentos e aumentar a oferta para seguir o aumento anual de 21% na demanda sem variar muito o custo de manutenção da plataforma. Se trata, portanto, de uma solução tecnológica escalável.

O projeto Cidade Lisa teve sucesso porque foi bem planejado e seguiu o método de projetos em Engenharia ensinado ao longo do curso na Poli-USP. A estruturação do trabalho em objetivos centrais e o seu desdobramento em requisitos de marketing e engenharia foram decisivos no desenvolvimento do produto final, sendo a implementação uma consequência natural do método usado na concepção do projeto.

8 TRABALHOS FUTUROS

Este trabalho serve como um primeiro passo para uma plataforma colaborativa de grande porte para agilizar o reparo das vias nas cidades, pois o projeto foi capaz de entregar a tecnologia central que será responsável por tornar a ideia possível.

Como próximos passos, será possível a adaptação do aplicativo Cidade Lisa para diversas marcas e modelos de smartphones, possibilitando a distribuição para o público em geral. Além disso, será dado o primeiro passo em direção à apresentação do projeto às entidades competentes do governo.

REFERÊNCIAS

- [1] BALAKUNTALA, S.; VENKATESH, S. An Intelligent System to Detect, Avoid and Maintain Potholes: A Graph Theoretic Approach. In: INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND UBIQUITOUS NETWORKING (ICMU), 7. **Proceedings...** 6-8 jan. 2014.
- [2] CHUGH, G.; BANSAL, D.; SOFAT, S. Road Condition Detection Using Smartphone Sensors: A Survey. International. **Journal of Electronic and Electrical Engineering**, v. 7, n. 6, p. 595-602, 2014. ISSN 0974-2174
- [3] ERIKSSON, J.; GIROD, L.; HULL, B.; NEWTON, R.; MADDEN, S.; BALAKRISHNAN, H. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS AND SERVICES (MOBISYS 2008), 6. **Proceedings...** Breckenridge, Estados Unidos, jun. 2008.
- [4] GORADE, P.; KARDE, D.; KHILARE, P.; SONTAKKE, A.; KEDAR, R. M. Real Time Pothole Tracking System Using Android Smart Phone. **International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)**, v. 3, ed. 5, maio 2014.
- [5] KALRA, N.; CHUGH, G.; BANSAL, D. Analyzing Driving and Road Events via Smartphone. **International Journal of Computer Applications**, ISSN 0975-8887, v. 98, no. 12, jul. 2014.
- [6] KULKARNI, A.; MHALGI, N.; GURNANI, S.; GIRI, N. Pothole Detection System using Machine Learning on Android. **International Journal of Emerging Technology and Advanced Engineering**, ISSN 2250-2459, v. 4, ed. 7, jul. 2014.
- [7] LANJEWAR, B. U.; SAGAR, R.; PAWAR, R.; KEDKHAR, J.; GOSAVI, K. Road Bump and Intensity Detection using Smartphone Sensors. **International Journal of Innovative Research in Computer and Communication Engineering**, v. 4, ed. 5, maio 2016.

- [8] MEDNIS, A.; STRAZDINS, G.; ZVIEDRIS, R.; KANONIRS, G.; SELAVO, L. Real Time Pothole Detection Using Android Smartphones with Accelerometers. In: IEEE INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING IN SENSOR SYSTEMS AND WORKSHOPS (DCOSS), 7. **Proceedings...** 2011.
- [9] MOHAN, P.; PADMANABHAN, V. N.; RAMJEE, R.; MURPHY, M. **Nericell**: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. Microsoft Research India, Bangalore. 2008.
- [10] WANG, H. W.; CHEN, C. H.; CHENG, D. Y.; LIN, C. H.; LO, C. C. A Real-Time Pothole Detection Approach for Intelligent Transportation System. **Mathematical Problems in Engineering**, v. 2015, id 869627, 2015.
- [11] **BURACOS MONITOR**. Disponível em: <<http://www.buracosmonitor.com.br/>> Acesso em: 19 mar. 2017.
- [12] **BURACOS UBERLÂNDIA**. Disponível em: <<https://buracosuberlandia.com.br/>> Acesso em: 19 mar. 2017.
- [13] BUSINESS INSIDER. **google-has-patented-a-way-to-hunt-down-potholes-2015-8**. Disponível em: <<http://www.businessinsider.com/google-has-patented-a-way-to-hunt-down-potholes-2015-8>> Acesso em: 19 mar. 2017.
- [14] SÃO PAULO. ESTADÃO. **Por que há tantos buracos em São Paulo?**. Disponível em: <<http://sao-paulo.estadao.com.br/noticias/geral,por-que-ha-tantos-buracos-em-sao-paulo,10000023824>> Acesso em: 18 mar. 2017.
- [15] Fundação Dom Cabral. **O Brasil não tem planejamento em infraestrutura de transporte**. Disponível em: <<http://www.fdc.org.br/blogespacodialogo/Lists/Postagens/Post.aspx?ID=363>> Acesso em: 19 mar. 2017.
- [16] RIO DE JANEIRO. GLOBO. G1 Notícias. **Buraco na rua é usado para atrair vítimas de assalto em Suzano**. Disponível em: <<http://g1.globo.com/sp/mogi-das-cruzes-suzano/noticia/2017/01/buraco-na-rua-e-usado-para-atrair-vitimas-de-assalto-em-suzano.html>> Acesso em: 18 mar. 2017.

[17] RIO DE JANEIRO. GLOBO. G1 Notícias. **Chuvas continuam deixando ruas e avenidas de Teresina cheias de buracos e alagadas.** Disponível em:

<<http://g1.globo.com/sp/bauru-marilia/bom-dia-cidade/videos/t/edicoes/v/levantamento-aponta-existencia-de-84-mil-buracos-em-marilia/5704799/>> Acesso em: 18 mar. 2017.

[18] RIO DE JANEIRO. GLOBO. G1 Notícias. **Marília tem um buraco para cada três moradores, aponta levantamento.** Disponível em:

<<http://g1.globo.com/sp/bauru-marilia/bom-dia-cidade/videos/t/edicoes/v/levantamento-aponta-existencia-de-84-mil-buracos-em-marilia/5704799/>> Acesso em: 18 mar. 2017.

[19] RIO DE JANEIRO. GLOBO. G1 Notícias. **Motoristas colecionam prejuízos por conta de buracos nas ruas da capital.** Disponível em:

<<http://g1.globo.com/pi/piaui/pitv-1edicao/videos/v/motoristas-colecionam-prejuizos-por-conta-de-buracos-nas-ruas-da-capital/5716833/>> Acesso em: 18 mar. 2017.

[20] RIO DE JANEIRO. GLOBO. G1 Notícias. **Tribunal de Contas reprova obras da Prefeitura de SP para tapar buracos.** Disponível em: <<http://g1.globo.com/sao-paulo/noticia/prefeitura-de-sp-gasta-r-500-mil-por-dia-para-tapar-buracos-diz-tcm.ghml>> Acesso em: 18 mar. 2017.

[21] **GOOGLE POTHOLE IN OHIO.** Disponível em:

<https://www.google.com/maps/d/viewer?mid=1m4kLDJgUvgID0Zp7Bd6w4iwm8fc&hl=en_US&ll=40.48416217194374%2C-81.47903344999997&z=11> Acesso em: 19 mar. 2017.

[22] DIGITAL TRENDS. **Google patents a system for cars that maps potholes.** Disponível em: <<http://www.digitaltrends.com/android/google-pothole-detection/>> Acesso em: 19 mar. 2017.

[23] SÃO PAULO SPTRANS. **Operação Tapa-Buraco.** Disponível em: <http://www3.prefeitura.sp.gov.br/saffor_bueiros/FormsPublic/serv1TapaBuracos.aspx#ancoraMapa> Acesso em: 19 mar. 2017.

[24] IBGE. **IBGE mapeia a infraestrutura dos transportes no Brasil**. Disponível em: <<http://www.brasil.gov.br/infraestrutura/2014/11/ibge-mapeia-a-infraestrutura-dos-transportes-no-brasil>> Acesso em: 19 mar. 2017.

[25] **STREET BUMP**. Disponível em: <<http://www.streetbump.org>> Acesso em: 19 mar. 2017.

[26] ZANDBERGEN, P. A. Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning. **Transactions in GIS**, 13: 5–25. doi:10.1111/j.1467-9671.2009.01152.x

[27] KOCH, C. Pothole detection in asphalt pavement images, **Advanced Engineering Informatics**, v. 25, ed. 3, ago. 2011, p. 507–515

[28] MADLI, R.; HEBBAR, S.; PATTAR, P.; GOLLA, V. Automatic Detection and Notification of Potholes and Humps on Roads to Aid Drivers, **IEEE Sensors Journal**, v. 15, ed. 8, ago. 2015

[28] JENGO, C. M.; HUGHES, D. Pothole Detection and Road Condition Assesment Using Hyperspectral Imagery. In: ASPRS 2005 Annual Conference. **Proceedings...** março 2005.

[29] JENGO, C. M.; HUGHES, D. **Pavement Surface Evaluation and Rating Manual**, Transportation Information Center. University of Wisconsin-Madison, 2002.

[30] **Multiple Back-Propagation**. Disponível em: <<http://mbp.sourceforge.net/>>. Acesso em: 18 jun. 2017.

[31] VAN DIGGELEN, F.; ENGE, P. The World's first GPS MOOC and Worldwide Laboratory using Smartphones, In: PROCEEDINGS OF THE 28TH INTERNATIONAL TECHNICAL MEETING OF THE SATELLITE DIVISION OF THE INSTITUTE OF NAVIGATION (ION GNSS+ 2015), 28. **Proceedings...** Tampa, Florida, Setembro 2015, p. 361-369.

APÊNDICES

1 ATIVIDADE PRINCIPAL DO APLICATIVO ANDROID PARA A DETECÇÃO DE QUALIDADE E DE BURACOS

```
package com.grivol.cidadelisa;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.graphics.Typeface;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.provider.Settings;
import android.util.Log;
import android.widget.Button;
import android.widget.TextView;
import java.io.OutputStreamWriter;
import java.util.ArrayList;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;

public class MainActivity extends Activity implements SensorEventListener,
LocationListener {

    private double tStartDelay = 5.0;
    private double tDetectionDelay = 3;
    public int WindowsSize = 50;
    private float stDefSup = 2.5f;

    //Android ID
    String androidId = "";

    //GPS's variables
    private double lat;
    private double lon;
    private float speed;

    //Accelerometer's variables
    private float accX = 0f;
    private float accY = 0f;
    private float accZ = 0f;
    private float gravityX = 0f;
```

```

private float gravityY = 0f;
private float gravityZ = 0f;
private float stDevX = 0f;
private float stDevY = 0f;
private float stDevZ = 0f;

//Window's variables
private ArrayList<Float> windowX = new ArrayList<Float>(WindowSize);
private ArrayList<Float> windowY = new ArrayList<Float>(WindowSize);
private ArrayList<Float> windowZ = new ArrayList<Float>(WindowSize);
private ArrayList<Float> speedWindow = new
ArrayList<Float>(WindowSize);
private float maxWindowX = -20;
private float maxWindowY = -20;
private float maxWindowZ = -20;
private float minWindowX = 20;
private float minWindowY = 20;
private float minWindowZ = 20;
private float avWindowX = 0;
private float avWindowY = 0;
private float avWindowZ = 0;
private float avSpeed = 0;

//Sensors' variables
public MediaPlayer mp1;
private SensorManager sensorManager;
private Sensor accelerometer;
private LocationManager locationManager;

//Time variables
private long tStart = 0l;
private long tCurrent = 0l;
private long tLastSave = 0l;
private long tHole = 0l;

//Counters
private int yMinIndex = 0;
private int yMaxIndex = 0;
private int hCounter = 0;

//Booleans
private boolean yEventOrder = false;

//TextView variables
private TextView HolesCounter, QualityIndex;
private TextView Latitude, Longitude, Speed;
private TextView textViewBack;

//Arrays for storing data
private ArrayList<Double> latSave = new ArrayList<Double>();
private ArrayList<Double> longSave = new ArrayList<Double>();
private ArrayList<String> timeSave = new ArrayList<String>();
private ArrayList<Double> classSave = new ArrayList<Double>();
private ArrayList<Double> holeLatSave = new ArrayList<Double>();
private ArrayList<Double> holeLongSave = new ArrayList<Double>();
private ArrayList<String> holeTimeSave = new ArrayList<String>();

//Buttons
private Button buttonSave;
private Button buttonBack;

```

```

//Typeface
private Typeface typeface;

//Neural Network variables
double mainWeights[] = {11.411386187777799, 3.198382123002161,
6.220682208897731, -1.750444386037554, 4.178002790795290,
2.294005827339134, 0.375270214662382, -7.963293638062140, -
0.936312816092373, 5.929689706550804, -6.297595659477760,
5.319516302892770, 6.660232084938935, -4.207308646106393,
10.950996199128838, 1.449000352746968, 0.263685464441822,
6.034585701344841, 0.193800270471542, 2.365582136730427, 0.338376564432226,
-1.582670080157836, -0.963852868606893, 0.158621884830477, -
5.816942301627740, -10.809506351256930, 2.258349631783364, -
3.931995564773921, -14.880825884692486, 3.920914856456162, -
3.217856693696738, 7.220537867497002, -5.431294054099764,
9.558448269457022, -3.385783291020965, 3.779349183528723, -
1.035054541619723, 0.116074601302356, -1.151807412170484,
1.655795565287583, 1.762011507174072, 5.205344439083789, -
1.399586194675045, 0.820078563132718, 1.094578213536831, -
0.477726761527458, -1.425124295617327, 0.907062852447141, -
0.593348793599483, 5.094516480894536, 5.337488198170610, -
3.039005587150732, -7.160532484729357, -4.425346117552123, -
0.512616505421930, -0.083993498089911, -0.167065996316647,
3.001453167923719, -0.451968622757537, -2.530152953579713,
2.871155732795901, -4.762295691986123, 3.530266930960352,
1.168172596460838, -10.033030596977333, -1.064548741642245};
double[] hiddenLayer1outputs = new double[5];
double[] hiddenLayer2outputs = new double[5];
int c;
double[] inputs = new double[5];
double output = 5d;

@Override
public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    addListenerOnButton();

    typeface = Typeface.createFromAsset(getAssets(), "fontawesome-
webfont.ttf");

    textViewBack = (TextView)findViewById(R.id.back);
    textViewBack.setTypeface(typeface);

    locationManager = (LocationManager)
    getSystemService(Context.LOCATION_SERVICE);

    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 500,
    0, this);

    initializeViews();

    //Initialize timer
    tStart = System.currentTimeMillis();

    //Initialize window

```

```

        for (int j = 0; j < WindowsSize; j++) {
            windowX.add(0f);
            windowY.add(0f);
            windowZ.add(0f);
            speedWindow.add(0f);
        }

        //Initialize sound player
        mp1 = MediaPlayer.create(MainActivity.this, R.raw.glass);

        //Initialize accelerometer
        sensorManager = (SensorManager)
        getSystemService(Context.SENSOR_SERVICE);

        if (sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER) !=
        null) {
            // success! we have an accelerometer
            accelerometer =
            sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
            sensorManager.registerListener(this, accelerometer,
            SensorManager.SENSOR_DELAY_UI);
        } else {
            // fail! we dont have an accelerometer!
        }

        androidId = Settings.Secure.getString(getContentResolver(),
        Settings.Secure.ANDROID_ID);
    }

    @Override
    public void onLocationChanged(Location location) {
        lat = location.getLatitude();
        lon = location.getLongitude();
        speed = location.getSpeed()*(3.6f);
    }

    @Override
    public void onProviderDisabled(String provider) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras)
    {
    }

    //Initialize Text Views
    public void initializeViews() {
        HolesCounter = (TextView) findViewById(R.id.HolesCounter);
        QualityIndex = (TextView) findViewById(R.id.QualityIndex);
        Latitude = (TextView) findViewById(R.id.Latitude);
        Longitude = (TextView) findViewById(R.id.Longitude);
        Speed = (TextView) findViewById(R.id.Speed);
    }

```

```

}

//onResume() registers the accelerometer for listening the events
protected void onResume() {
    super.onResume();
    sensorManager.registerListener(this, accelerometer,
SensorManager.SENSOR_DELAY_UI);
}

//onPause() unregisters the accelerometer for stop listening the events
protected void onPause() {
    super.onPause();
    sensorManager.unregisterListener(this);
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}

@Override
public void onSensorChanged(SensorEvent event) {

    //Recalibrate default gravity when the vehicle is stopped
    if (avSpeed == 0){
        gravityX = event.values[0];
        gravityY = event.values[1];
        gravityZ = event.values[2];
    }

    // Get the change of the x,y,z values of the accelerometer
    tCurrent = System.currentTimeMillis();
    accX = event.values[0] - gravityX;
    accY = event.values[1] - gravityY;
    accZ = event.values[2] - gravityZ;

    // Cleans current values
    getCleanValues();

    // Displays the current x,y,z accelerometer values
    getCurrentValues();

    // Displays the max x,y,z accelerometer values
    checkHole();

    // Prints values on the screen
    setText();

    //Execute every 3 seconds
    double runTime = (tCurrent - tLastSave) / 1000.0;
    if((runTime > 3.0) && (avSpeed > 20) && !(Double.isNaN(output))){

        // Gets the quality class of the road
        calculateClass();

        //Add to file
        if(!(Double.isNaN(lat)) && !(Double.isNaN(lon))){
            latSave.add(lat);
            longSave.add(lon);
            classSave.add(Math.round(output*100.0)/100.0);
        }
    }
}

```

```

        SimpleDateFormat dateformat = new SimpleDateFormat("dd-MM-
yyyy HH:mm:ss");
        String timeInString = dateformat.format(new Date());
        timeSave.add(timeInString);
        sign(androidId, timeInString, Double.toString(lat),
Double.toString(lon), Double.toString(Math.round(output*100.0)/100.0));
    }

    tLastSave = System.currentTimeMillis();
}

}

//Reset variables
public void getCleanValues() {

    windowX.remove(0);
    windowY.remove(0);
    windowZ.remove(0);
    speedWindow.remove(0);

    windowX.add(accX);
    windowY.add(accY);
    windowZ.add(accZ);
    speedWindow.add(speed);

    maxWindowX = -20;
    maxWindowY = -20;
    maxWindowZ = -20;

    minWindowX = 20;
    minWindowY = 20;
    minWindowZ = 20;

    avWindowX = 0f;
    avWindowY = 0f;
    avWindowZ = 0f;

    stDevX = 0f;
    stDevY = 0f;
    stDevZ = 0f;
}

//Calculate the window's max, min and average values
public void getCurrentValues() {

    int i = 0;
    for (i = 0; i < WindowsSize; i++){

        if (windowX.get(i) > maxWindowX) {
            maxWindowX = windowX.get(i);
        }
        if (windowY.get(i) > maxWindowY) {
            maxWindowY = windowY.get(i);
            yMaxIndex = i;
        }
        if (windowZ.get(i) > maxWindowZ) {
            maxWindowZ = windowZ.get(i);
        }
        if (windowX.get(i) < minWindowX) {

```

```

        minWindowX = windowX.get(i);
    }
    if (windowY.get(i) < minWindowY) {
        minWindowY = windowY.get(i);
        yMinIndex = i;
    }
    if (windowZ.get(i) < minWindowZ) {
        minWindowZ = windowZ.get(i);
    }

    avWindowX = avWindowX + windowX.get(i);
    avWindowY = avWindowY + windowY.get(i);
    avWindowZ = avWindowZ + windowZ.get(i);
    avSpeed = avSpeed + speedWindow.get(i);
}

avWindowX = avWindowX / WindowsSize;
avWindowY = avWindowY / WindowsSize;
avWindowZ = avWindowZ / WindowsSize;
avSpeed = avSpeed / WindowsSize;

if(yMaxIndex > yMinIndex){
    yEventOrder = true;
}
else{
    yEventOrder = false;
}

for (i = 0; i < WindowsSize; i++){
    stDevX = (float) (stDevX + Math.pow((windowX.get(i)-
avWindowX), 2));
    stDevY = (float) (stDevY + Math.pow((windowY.get(i)-
avWindowY), 2));
    stDevZ = (float) (stDevZ + Math.pow((windowZ.get(i)-
avWindowZ), 2));
}

stDevX = (float) Math.sqrt(stDevX / WindowsSize);
stDevY = (float) Math.sqrt(stDevY / WindowsSize);
stDevZ = (float) Math.sqrt(stDevZ / WindowsSize);
}

//Check if we just passed a pothole
public void checkHole() {

    //Calculate time elapsed between each event

    double elapsedTime = (tCurrent - tStart) / 1000.0;
    double holeTime = (tCurrent - tHole) / 1000.0;

    //Analyse road conditions

    boolean iTimeCheck = (elapsedTime > tStartDelay);
    boolean strongStDev = (stDevY > stDefSup);

    //Analyse if we passed a pothole

    boolean hTimeCheck = (elapsedTime > tStartDelay) && (holeTime >
tDetectionDelay);
    boolean unbalancedCheck = (Math.abs(maxWindowX - minWindowX) > 6);

```

```

        boolean dropCheck = (((maxWindowY - avWindowY) > 4)
            && ((avWindowY - minWindowY) > 4));

        //Check if we have a pothole
        if (hTimeCheck && iTimeCheck && unbalancedCheck && dropCheck &&
yEventOrder) {
            hCounter++;
            tHole = System.currentTimeMillis();
            mp1.start();
            if(!(Double.isNaN(lat)) && !(Double.isNaN(lon))){
                holeLatSave.add(lat);
                holeLongSave.add(lon);
                SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-
yyyy HH:mm:ss");
                String timeInString = dateFormat.format(new Date());
                holeTimeSave.add(timeInString);
                sign2(androidId, timeInString, Double.toString(lat),
Double.toString(lon));
            }
        }

        //Print values on the screen
        public void setText() {

            HolesCounter.setText(Integer.toString(hCounter));
            if(avSpeed < 10){
                QualityIndex.setText("-");
            }
            else{
                QualityIndex.setText(String.format("%.1f", output));
            }

            Latitude.setText(String.format("%.6f", (float) lat));
            Longitude.setText(String.format("%.6f", (float) lon));
            Speed.setText(String.format("%.2f", speed));
        }

        //Gets the quality class of the road
        public void calculateClass() {

            int pointer = 0;

            //Inputs of the neural network

            inputs[0] = avSpeed;
            inputs[1] = stDevX;
            inputs[2] = stDevY;
            inputs[3] = maxWindowX - minWindowX;
            inputs[4] = (maxWindowY - minWindowY)/avSpeed;

            //Get the output
            inputs[0] = -1.0 + (inputs[0] - 2.963879300000000) /
24.704923349999998;
            inputs[1] = -1.0 + (inputs[1] - 0.213978370000000) /
1.287507515000000;
            inputs[2] = -1.0 + (inputs[2] - 0.268840850000000) /
1.771910175000000;

```

```

    inputs[3] = -1.0 + (inputs[3] - 0.9190000600000000) /
7.603999969999999;
    inputs[4] = -1.0 + (inputs[4] - 0.083176128746388) /
0.276267007385270;

    //Hidden Layer 1 outputs
    hiddenLayer1outputs[0] = mainWeights[pointer];
    for(c = 0; c < 5; c++) {
        pointer++;
        hiddenLayer1outputs[0] += mainWeights[pointer] * inputs[c];
    }
    hiddenLayer1outputs[0] = 1.0 / (1.0 + Math.exp(-
hiddenLayer1outputs[0]));

    pointer++;
    hiddenLayer1outputs[1] = mainWeights[pointer];
    for(c = 0; c < 5; c++) {
        pointer++;
        hiddenLayer1outputs[1] += mainWeights[pointer] * inputs[c];
    }
    hiddenLayer1outputs[1] = 1.0 / (1.0 + Math.exp(-
hiddenLayer1outputs[1]));

    pointer++;
    hiddenLayer1outputs[2] = mainWeights[pointer];
    for(c = 0; c < 5; c++) {
        pointer++;
        hiddenLayer1outputs[2] += mainWeights[pointer] * inputs[c];
    }
    hiddenLayer1outputs[2] = 1.0 / (1.0 + Math.exp(-
hiddenLayer1outputs[2]));

    pointer++;
    hiddenLayer1outputs[3] = mainWeights[pointer];
    for(c = 0; c < 5; c++) {
        pointer++;
        hiddenLayer1outputs[3] += mainWeights[pointer] * inputs[c];
    }
    hiddenLayer1outputs[3] = 1.0 / (1.0 + Math.exp(-
hiddenLayer1outputs[3]));

    pointer++;
    hiddenLayer1outputs[4] = mainWeights[pointer];
    for(c = 0; c < 5; c++) {
        pointer++;
        hiddenLayer1outputs[4] += mainWeights[pointer] * inputs[c];
    }
    hiddenLayer1outputs[4] = 1.0 / (1.0 + Math.exp(-
hiddenLayer1outputs[4]));

    //Hidden Layer 2 outputs
    pointer++;
    hiddenLayer2outputs[0] = mainWeights[pointer];
    for(c = 0; c < 5; c++) {
        pointer++;
        hiddenLayer2outputs[0] += mainWeights[pointer] *
hiddenLayer1outputs[c];
    }
    hiddenLayer2outputs[0] = 1.0 / (1.0 + Math.exp(-
hiddenLayer2outputs[0]));

```

```

        pointer++;
        hiddenLayer2outputs[1] = mainWeights[pointer];
        for(c = 0; c < 5; c++) {
            pointer++;
            hiddenLayer2outputs[1] += mainWeights[pointer] *
hiddenLayer1outputs[c];
        }
        hiddenLayer2outputs[1] = 1.0 / (1.0 + Math.exp(-
hiddenLayer2outputs[1]));

        pointer++;
        hiddenLayer2outputs[2] = mainWeights[pointer];
        for(c = 0; c < 5; c++) {
            pointer++;
            hiddenLayer2outputs[2] += mainWeights[pointer] *
hiddenLayer1outputs[c];
        }
        hiddenLayer2outputs[2] = 1.0 / (1.0 + Math.exp(-
hiddenLayer2outputs[2]));

        pointer++;
        hiddenLayer2outputs[3] = mainWeights[pointer];
        for(c = 0; c < 5; c++) {
            pointer++;
            hiddenLayer2outputs[3] += mainWeights[pointer] *
hiddenLayer1outputs[c];
        }
        hiddenLayer2outputs[3] = 1.0 / (1.0 + Math.exp(-
hiddenLayer2outputs[3]));

        pointer++;
        hiddenLayer2outputs[4] = mainWeights[pointer];
        for(c = 0; c < 5; c++) {
            pointer++;
            hiddenLayer2outputs[4] += mainWeights[pointer] *
hiddenLayer1outputs[c];
        }
        hiddenLayer2outputs[4] = 1.0 / (1.0 + Math.exp(-
hiddenLayer2outputs[4]));

        //Get final output
        pointer++;
        output = mainWeights[pointer];

        for(c = 0; c < 5; c++) {
            pointer++;
            output += mainWeights[pointer] * hiddenLayer2outputs[c];
        }
        output = 1.0 / (1.0 + Math.exp(-output));
        output = 1.0000000000000000 + (output - 0.000000) *
4.0000000000000000;

        output = 2 * output;
    }

    public void addListenerOnButton() {
        buttonBack = (Button) findViewById(R.id.back);

```

```

buttonBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        buttonBack.setAlpha(0.85f);
        Intent i = new Intent(MainActivity.this, MainMenu.class);
        startActivity(i);
    }
});

buttonSave = (Button) findViewById(R.id.buttonSave);

buttonSave.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View arg0) {

        buttonSave.setAlpha(0.85f);

        //Save quality indicators
        String[] strTime = new String[timeSave.size()];
        String[] strLat = new String[latSave.size()];
        String[] strLong = new String[longSave.size()];
        String[] strClass = new String[classSave.size()];

        //Save holes' data
        String[] strHoleTime = new String[holeTimeSave.size()];
        String[] strHoleLat = new String[holeLatSave.size()];
        String[] strHoleLong = new String[holeLongSave.size()];

        for (int j = 0; j < timeSave.size(); j++) {
            strTime[j] = String.valueOf(timeSave.get(j));
            strLat[j] = String.valueOf(latSave.get(j));
            strLong[j] = String.valueOf(longSave.get(j));
            strClass[j] = String.valueOf(classSave.get(j));
        }

        for (int j = 0; j < holeTimeSave.size(); j++) {
            strHoleTime[j] = String.valueOf(holeTimeSave.get(j));
            strHoleLat[j] = String.valueOf(holeLatSave.get(j));
            strHoleLong[j] = String.valueOf(holeLongSave.get(j));
        }

        //Create JSON object with the quality indicators
        JSONObject jsonObj = new JSONObject();

        try {
            jsonObj = createJson(strTime, strLat, strLong,
strClass);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        //Write JSON in the internal storage
        SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-
yyyy HH:mm:ss");
        String millisInString = dateFormat.format(new Date());
        writeJsonToFile("Trajeto de
"+millisInString+"_0.json", jsonObj.toString());
    }
});

```

```

        //Create JSON object with the holes's data
        jsonObj = new JSONObject();

        try {
            jsonObj = createJsonHoles(strHoleTime, strHoleLat,
strHoleLong);
        } catch (JSONException e) {
            e.printStackTrace();
        }

        //Write JSON in the internal storage
        writeJsonToFile("Trajeto de
"+millisInString+"_H.json", jsonObj.toString());

        //Finalize saving and exit
        Toast.makeText(getApplicationContext(), "Trajeto Salvo",
Toast.LENGTH_SHORT).show();
        Intent i = new Intent(MainActivity.this, MainMenu.class);
        startActivity(i);
    }

    });
}

    public void sign(String user, String time, String latitude, String
longitude, String classe){
        new SignupActivity(this).execute(user, time, latitude, longitude,
classe);
    }

    public void sign2(String user, String time, String latitude, String
longitude){
        new SignupActivity2(this).execute(user, time, latitude, longitude);
    }

    public JSONObject createJson(String[] strTime, String[] strLat,
String[] strLong, String[] strClass) throws JSONException {
        JSONObject jsonObj = new JSONObject();
        JSONArray markerObj = new JSONArray();
        JSONObject dataObj;

        for (int i = 1; i <= strTime.length; i++){
            dataObj = new JSONObject();
            dataObj.put("timestamp", strTime[i-1]);
            dataObj.put("latitude", strLat[i-1]);
            dataObj.put("longitude", strLong[i-1]);
            dataObj.put("class", strClass[i-1]);
            markerObj.put(dataObj);
        }
        jsonObj.put("markers", markerObj);

        return jsonObj;
    }

    public JSONObject createJsonHoles(String[] strTime, String[] strLat,
String[] strLong) throws JSONException {
        JSONObject jsonObj = new JSONObject();
        JSONArray markerObj = new JSONArray();

```

```

JSONObject dataObj;

for (int i = 1; i <= strTime.length; i++){
    dataObj = new JSONObject();
    dataObj.put("timestamp",strTime[i-1]);
    dataObj.put("latitude",strLat[i-1]);
    dataObj.put("longitude",strLong[i-1]);
    markerObj.put(dataObj);
}
jsonObj.put("markers", markerObj);

return jsonObj;
}

private void writeJsonToFile(String filename, String data) {
    try {
        OutputStreamWriter outputStreamWriter = new
        OutputStreamWriter(openFileOutput(filename, Context.MODE_PRIVATE));
        outputStreamWriter.write(data);
        outputStreamWriter.close();
    }
    catch (IOException e) {
        Log.e("Exception", "File write failed: " + e.toString());
    }
}
}

```

2 ATIVIDADE DE VISUALIZAÇÃO DO MAPA NO APLICATIVO ANDROID

```

package com.grivol.cidadelisa;

import android.content.Intent;
import android.graphics.Typeface;
import android.os.AsyncTask;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptor;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class MapActivity extends FragmentActivity implements
    GoogleMap.OnMarkerClickListener,
    OnMapReadyCallback {

    private Button buttonBack;
    private Typeface typeface;
    private TextView textViewBack;
    private GoogleMap mMap;
    private JSONArray jsonArray;
    private JSONObject jsonTmpObj;
    private BitmapDescriptor descrIcon;
    private String mapKind, mapInfo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Bundle extras = getIntent().getExtras();
        if (extras != null) {
            mapKind = (String) extras.get("mapType");
            mapInfo = (String) extras.get("mapInfo");

```

```

    }

    setContentView(R.layout.activity_map);

    addListenerOnButton();

    typeface = Typeface.createFromAsset(getAssets(), "fontawesome-
webfont.ttf");

    textViewBack = (TextView) findViewById(R.id.back);
    textViewBack.setTypeface(typeface);

    SupportMapFragment mapFragment =
        (SupportMapFragment)
        getSupportFragmentManager().findFragmentById(R.id.fragment);
    mapFragment.getMapAsync(this);
}

/**
 * Called when the map is ready.
 */
@Override
public void onMapReady(GoogleMap map) {
    mMap = map;
    try {
        jsonArray = new JSONArray(mapInfo);
    } catch (JSONException e) {
        e.printStackTrace();
    }
    Boolean validMarker = false;
    for (int i = 0; i < jsonArray.length(); i++) {
        try {
            jsonTmpObj = jsonArray.getJSONObject(i);
        } catch (JSONException e) {
            e.printStackTrace();
        }
        try {
            if (mapKind.equals("Q")) {
                switch (jsonTmpObj.getInt("class")) {
                    case 0:
                        descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_VIOLET);
                        break;
                    case 1:
                        descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED);
                        break;
                    case 2:
                        descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED);
                        break;
                    case 3:
                        descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ORANGE);
                        break;
                    case 4:
                        descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_ORANGE);
                        break;

```

```

        case 5:
            descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_YELLOW);
            break;
        case 6:
            descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_YELLOW);
            break;
        case 7:
            descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_CYAN);
            break;
        case 8:
            descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_CYAN);
            break;
        case 9:
            descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE);
            break;
        case 10:
            descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE);
            break;
        default:
            descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_VIOLET);
            break;
    }
    mMap.addMarker(new MarkerOptions()
        .position(new
LatLng(jsonTmpObj.getDouble("latitude"),
jsonTmpObj.getDouble("longitude")))
        .title(jsonTmpObj.getString("class"))
        .snippet("Horário: " +
jsonTmpObj.getString("timestamp"))
        .icon(descrIcon));
    validMarker = true;
    } else {
        descrIcon =
BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED);
        mMap.addMarker(new MarkerOptions()
            .position(new
LatLng(jsonTmpObj.getDouble("latitude"),
jsonTmpObj.getDouble("longitude")))
            .title("Horário: " +
jsonTmpObj.getString("timestamp"))
            .icon(descrIcon));
        validMarker = true;
    }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    }
    if (validMarker) {
        try {
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new
LatLng(jsonTmpObj.getDouble("latitude"),
jsonTmpObj.getDouble("longitude")), 16.0f));

```

```

        } catch (JSONException e) {
            e.printStackTrace();
        }
    } else {
        Toast.makeText(getApplicationContext(), "Nenhum resultado
encontrado!", Toast.LENGTH_SHORT).show();
    }
}

/**
 * Called when the user clicks a marker.
 */
@Override
public boolean onMarkerClick(final Marker marker) {

    // Retrieve the data from the marker.
    Integer clickCount = 0;

    // Check if a click count was set, then display the click count.
    if (clickCount != null) {
        clickCount = clickCount + 1;
        marker.setTitle(clickCount.toString());
        Toast.makeText(this,
            marker.getTitle() +
            " has been clicked " + clickCount + " times.",
            Toast.LENGTH_SHORT).show();
    }

    // Return false to indicate that we have not consumed the event and
    that we wish
    // for the default behavior to occur (which is for the camera to
    move such that the
    // marker is centered and for the marker's info window to open, if
    it has one).
    return false;
}

private String readJsonFromFile(String filename) {

    String ret = "";

    try {
        InputStream inputStream = openFileInput(filename);

        if (inputStream != null) {
            InputStreamReader inputStreamReader = new
InputStreamReader(inputStream);
            BufferedReader bufferedReader = new
BufferedReader(inputStreamReader);
            String receiveString = "";
            StringBuilder stringBuilder = new StringBuilder();

            while ((receiveString = bufferedReader.readLine()) != null)
            {
                stringBuilder.append(receiveString);
            }

            inputStream.close();
            ret = stringBuilder.toString();
        }
    }
}

```

```

    } catch (FileNotFoundException e) {
        Log.e("login activity", "File not found: " + e.toString());
    } catch (IOException e) {
        Log.e("login activity", "Can not read file: " + e.toString());
    }

    return ret;
}

public void addListenerOnButton() {
    buttonBack = (Button) findViewById(R.id.back);

    buttonBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
            buttonBack.setAlpha(0.85f);
            Intent i = new Intent(MapActivity.this, MainMenu.class);
            startActivity(i);
        }
    });
}
}

```

3 CLASSE DE CONEXÃO DO APLICATIVO ANDROID AO SERVIDOR

```

package com.grivol.cidadelisa;

/**
 * Created by grivol on 30/09/2017.
 */

import android.content.Context;
import android.os.AsyncTask;
import android.view.View;
import android.widget.Toast;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;

public class SignupActivity extends AsyncTask<String, Void, String> {

    private Context context;

    public SignupActivity(Context context) {
        this.context = context;
    }

    protected void onPreExecute() {

    }

    @Override
    protected String doInBackground(String... arg0) {
        String userName = arg0[0];
        String timeStamp = arg0[1];
        String latitude = arg0[2];
        String longitude = arg0[3];
        String classNumber = arg0[4];

        String link;
        String data;
        BufferedReader bufferedReader;
        String result;

        try {
            data = "?user=" + URLEncoder.encode(userName, "UTF-8");
            data += "&timestamp=" + URLEncoder.encode(timeStamp, "UTF-8");
            data += "&latitude=" + URLEncoder.encode(latitude, "UTF-8");
            data += "&longitude=" + URLEncoder.encode(longitude, "UTF-8");
            data += "&class=" + URLEncoder.encode(classNumber, "UTF-8");

            link = "http://www.grivol.com/signup.php" + data;
            URL url = new URL(link);
            HttpURLConnection con = (HttpURLConnection)
url.openConnection();

```

```

        bufferedReader = new BufferedReader(new
InputStreamReader(con.getInputStream()));
        result = bufferedReader.readLine();
        return result;
    } catch (Exception e) {
        return new String("Exception: " + e.getMessage());
    }
}

@Override
protected void onPostExecute(String result) {
    String jsonStr = result;
    if (jsonStr != null) {
        try {
            JSONObject jsonObj = new JSONObject(jsonStr);
            String query_result = jsonObj.getString("query_result");
            if (query_result.equals("SUCCESS")) {
                //Toast.makeText(context, "Data inserted successfully.
Signup successful.", Toast.LENGTH_SHORT).show();
            } else if (query_result.equals("FAILURE")) {
                //Toast.makeText(context, "Data could not be inserted.
Signup failed.", Toast.LENGTH_SHORT).show();
            } else {
                //Toast.makeText(context, "Couldn't connect to remote
database.", Toast.LENGTH_SHORT).show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
            //Toast.makeText(context, "Error parsing JSON data.",
Toast.LENGTH_SHORT).show();
        }
    } else {
        //Toast.makeText(context, "Couldn't get any JSON data.",
Toast.LENGTH_SHORT).show();
    }
}
}

```

4 PÁGINA PHP DE CONEXÃO À BASE DE DADOS

```
<?php
$con=mysqli_connect("localhost","grivol_carlos","*****","grivol_cidadelisa");
if (mysqli_connect_errno($con))
{
    echo '{"query_result":"ERROR"}';
}

$userName = $_GET['user'];
$timeStamp = $_GET['timestamp'];
$lat = $_GET['latitude'];
$long = $_GET['longitude'];
$classNumber = $_GET['class'];

$result = mysqli_query($con,"INSERT INTO quality (user, timestamp, latitude, longitude, class)
VALUES ('$userName', '$timeStamp', '$lat', '$long', '$classNumber')");

if($result == true) {
    echo '{"query_result":"SUCCESS"}';
}
else{
    echo '{"query_result":"FAILURE"}';
}
mysqli_close($con);
?>
```

5 PÁGINA PHP PARA OBTENHA DADOS ARMAZENADOS NA BASE

```

<?
//these are the server details
$servername = "localhost";
$username = "grivol_carlos";
$password = "*****";
$dbname = "grivol_cidadelisa";

//creating a new connection object using mysqli
$conn = new mysqli($servername, $username, $password, $dbname);

//if there is some error connecting to the database
//with die we will stop the further execution by displaying a message causing the error
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

//if everything is fine

//creating an array for storing the data
$qualities = array();

//this is our sql query
$sql = "SELECT user, timestamp, latitude, longitude, class FROM quality;";

//creating an statement with the query
$stmt = $conn->prepare($sql);

//executing that statement
$stmt->execute();

//binding results for that statement
$stmt->bind_result($user, $timestamp, $latitude, $longitude, $class);

//looping through all the records
while($stmt->fetch()){

    //pushing fetched data in an array
    $temp = [
        'user'=>$user,
        'timestamp'=>$timestamp,
        'latitude'=>$latitude,
        'longitude'=>$longitude,
        'class'=>$class
    ];

    //pushing the array inside the hero array
    array_push($qualities, $temp);
}

//displaying the data in json format
echo json_encode($qualities);

```