

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**DRTRACK: MONITORAMENTO DA AGRESSIVIDADE DE  
MOTORISTAS ATRAVÉS DE DISPOSITIVO ANDROID**

**LEONARDO REZENDE**

**BLUMENAU**  
**2019**

**LEONARDO REZENDE**

**DRTRACK: MONITORAMENTO DA AGRESSIVIDADE DE  
MOTORISTAS ATRAVÉS DE DISPOSITIVO ANDROID**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Dalton Solano dos Reis - Orientador

**BLUMENAU  
2019**

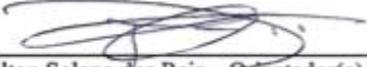
**DRTRACK: MONITORAMENTO DA AGRESSIVIDADE DE  
MOTORISTAS ATRAVÉS DE DISPOSITIVOS ANDROID**

Por

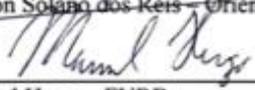
**LEONARDO REZENDE**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

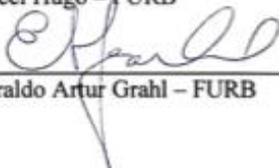
Presidente:

  
\_\_\_\_\_  
Prof(a). Dalton Solano dos Reis – Orientador(a), FURB

Membro:

  
\_\_\_\_\_  
Prof(a). Marcel Hugo – FURB

Membro:

  
\_\_\_\_\_  
Prof(a). Everaldo Artur Grahl – FURB

Blumenau, 11 de dezembro de 2019

Dedico este trabalho a meus familiares, amigos e professores que de alguma forma contribuíram na elaboração desse projeto.

## **AGRADECIMENTOS**

À minha mãe, irmão e namorada por todo o apoio, paciência e sermões nessa caminhada.

À minha família e amigos pela compreensão e incentivo nos momentos que precisei.

Ao meu orientador Prof. Dalton Solano dos Reis pela paciência, persistência e ensinamentos para a conclusão dessa etapa.

Pode se encontrar a felicidade mesmo nas horas mais sombrias, se a pessoa se lembrar de acender a luz.

J. K. Rowling

## RESUMO

Este trabalho tem como objetivo o desenvolvimento de uma ferramenta para monitoramento da condução de veículos. A ferramenta, nomeada DrTrack, faz a análise de aspectos como aceleração, frenagem, curvas, velocidade e geolocalização para dar notas e traçar o perfil de condução do usuário com a finalidade de apontar pontos onde a pilotagem pode ser mais segura. Foi utilizada a plataforma do Delphi 10.3 para desenvolvimento do aplicativo de celular. Para desenvolver o *web service* foi utilizada a plataforma AWS Lambda, juntamente com o DynamoDB. O banco de dados utilizado no aplicativo é o SQLite. A ferramenta foi testada juntamente com o aplicativo Trânsito+gentil, correlato deste trabalho, para fazer a validação da análise de perfil do usuário. A comparação permitiu concluir que os objetivos deste trabalho foram alcançados e apontou os pontos a serem melhorados.

Palavras-chave: Monitoramento. Geolocalização. Aceleração. Curvas. Velocidade.

## **ABSTRACT**

This work aims to develop a vehicle driving monitoring tool. The tool, named DrTrack, analyzes aspects such as acceleration, braking, cornering, velocity and geolocation to grade and profile the user's driving profile to pinpoint points where riding can be safer. We used the Delphi 10.3 platform for mobile application development. To develop the web service, the AWS Lambda platform was used, together with DynamoDB. The database used in the application is SQLite. The tool was tested in conjunction with the Trânsito+gentil application, correlated of this work, to validate the user profile analysis. Comparison allowed us to conclude that the objectives of this work were achieved and pointed out the points to be improved.

Key-words: Tracking. Geolocation. Acceleration. Curves. Velocity.

## LISTA DE FIGURAS

Figura 1 – Representação da triangulação do GPS.....	15
Figura 2 - Eixos do acelerômetro .....	17
Figura 3 - Representação de vetores perpendiculares entre si.....	17
Figura 4- Funcionamento do <i>AWS Lambda</i> .....	19
Figura 5 - Dispositivo montado (acoplado) no caminhão .....	21
Figura 6 - Tela de análise da viagem.....	22
Figura 8 - Dispositivo instalado no veículo.....	23
Figura 9 - Tela Inicial Trânsito+gentil .....	24
Figura 10 - Diagrama de classes.....	28
Figura 11 - Diagrama de sequência .....	29
Figura 12 - Modelo de Entidade e Relacionamento .....	30
Figura 13 - Diagrama de casos de uso .....	31
Figura 14 - Estrutura de arquivos do aplicativo .....	32
Figura 15 - Tela de login .....	40
Figura 16 - Cadastro de usuário.....	41
Figura 17 - Tela principal .....	42
Figura 18 - Tela nova viagem.....	43
Figura 19 - Finalizar viagem .....	44
Figura 20 - Tela Minhas Viagens .....	45
Figura 21 - Configuração do sistema.....	45
Figura 22 - Comparação entre menus DrTrack e Trânsito+gentil.....	47
Figura 23 - Comparação entra velocidade no velocímetro e DrTrack.....	48
Figura 24 - Comparação das notas do DrTrack e Trânsito+gentil. ....	48
Figura 25 - Comparação entre os mapas gerados no DrTrack e Trânsito+gentil .....	49

## LISTA DE QUADROS

Quadro 1- Fórmula de Harvesine .....	16
Quadro 2 - Equação de distância entre duas coordenadas.....	16
Quadro 3- Teorema de Pitágoras .....	18
Quadro 4 - Comparativo dos trabalhos correlatos .....	24
Quadro 5 - Conexão com banco de dados .....	33
Quadro 6 - Configurando Permissões.....	33
Quadro 7 - Solicitação de permissão .....	34
Quadro 8 - Obtenção dos dados do acelerômetro .....	34
Quadro 9 - Obtenção dos dados de localização .....	35
Quadro 10 - Cálculo da distância entre coordenadas .....	35
Quadro 11 - Desenho da rota traçada no mapa.....	36
Quadro 12 - Consulta velocidade máxima via.....	37
Quadro 13 - Cálculo nota velocidade .....	38
Quadro 14 - Procedimento para média mascarada 3 .....	39
Quadro 15 - Análise de classificação das curvas.....	39
Quadro 16 - Comparativo entre trabalhos correlatos e o DrTrack .....	46
Quadro 17 - Tabela Acelerometro .....	56
Quadro 18 - Tabela Configuracao .....	56
Quadro 19 - Tabela Localizacao.....	56
Quadro 20 - Tabela Usuario .....	57
Quadro 21 - Tabela Viagem.....	57

## **LISTA DE ABREVIATURAS E SIGLAS**

ANFAVEA – Associação Nacional de Veículos Automotores

API – Application Programming Interface

CSS – Cascading Style Sheets

CTB – Código de Trânsito Brasileiro

DPVAT – Danos Pessoais Causados por veículos Automotores de via Terrestre

GPS – Global Position System

GSM – Global System for Mobile

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IPEA – Instituto de Pesquisas Espaciais Aplicadas

JSON – JavaScript Object Notation

MDF – Medium Density Fiberboard

OMS – Organização Mundial da Saúde

REST – REpresentational State Transfer

RF – Requisitos Funcionais

RNF – Requisitos Não Funcionais

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>14</b>
2.1 GEOLOCALIZAÇÃO EM APLICAÇÕES MÓVEIS .....	14
2.2 SENSORES INERCIAIS .....	16
2.3 ACELERAÇÃO, FRENAGEM E VELOCIDADE .....	18
2.4 ARQUITETURA SERVERLESS .....	18
2.4.1 AWS Lambda.....	19
2.5 TRABALHOS CORRELATOS.....	19
2.5.1 Monitoramento da agressividade na direção de caminhões através de acelerômetro e GPS .....	20
2.5.2 Aplicação para monitoramento veicular em tempo real .....	22
2.5.3 Trânsito+gentil .....	23
2.5.4 Correlação entre os trabalhos relacionados .....	24
<b>3 DESENVOLVIMENTO.....</b>	<b>26</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	26
3.2 ESPECIFICAÇÃO .....	27
3.2.1 Diagrama de classes .....	27
3.2.2 Diagrama de sequência .....	28
3.2.3 Modelo de Entidade e Relacionamento.....	29
3.2.4 Diagrama de casos de uso .....	30
3.3 IMPLEMENTAÇÃO .....	31
3.3.1 Técnicas e ferramentas utilizadas.....	31
3.3.2 Operacionalidade da implementação .....	40
3.4 RESULTADOS E DISCUSSÕES.....	46
<b>4 CONCLUSÕES.....</b>	<b>51</b>
4.1 EXTENSÕES .....	52
<b>REFERÊNCIAS.....</b>	<b>53</b>
<b>APÊNDICE A – DICIONÁRIO DE DADOS.....</b>	<b>56</b>

## 1 INTRODUÇÃO

Um estudo coordenado e revelado pela Associação Nacional de Veículos Automotores (ANFAVEA), estima que a frota de veículos brasileira vai sofrer um aumento de 140% e deve chegar a 95,2 milhões em 2034 (ANFAVEA, 2014, p. 1). Os veículos se tornaram uma necessidade para empresas, e podem representar prejuízo ao negócio, dependendo da utilização e gestão feita da frota (ROTA EXATA, 2015).

No ano de 2012, os acidentes de trânsito foram a oitava maior causa de morte no mundo, causando 1,3 milhões de mortes (WORLD HEALTH ORGANIZATION, 2014, p.1). Segundo a Organização Mundial da Saúde (OMS), estima-se que em 2020 pode-se ter 1,9 milhões do mortes no trânsito (WAISELFISZ, 2012). Segundo o Instituto de Pesquisa Econômica Aplicada (IPEA) no Brasil o número de mortes e pessoas gravemente feridas no trânsito ultrapassa os 150 mil por ano. Estima que os custos destes acidentes sejam de R\$ 28 bilhões por ano (BACCHIERI; BARROS, 2011, p.1).

Segundo a OMS o Brasil é o quinto país mais violento no trânsito no mundo. Os dados apontados pelo Danos Pessoais Causados por Veículos Automotores de Via Terrestre (DPVAT) mostram que são, em média, 150 mortes por dia no país. Conforme a Guarda municipal de Itapetininga (SP), 90% dos acidentes são causados por imprudência do motorista ou pedestre. Segundo Catarina Nanini, especialista em trânsito da guarda municipal, “a vida e os sonhos de muitas vítimas poderiam ser mantidos caso o cuidado e prudência fosse um hábito comum daqueles que compõem o trânsito” (GLOBO, 2013, p.1).

Tendo esse cenário de crescimento da frota de veículos brasileira e o alto número de acidentes e mortes no trânsito provenientes da imprudência, esse trabalho apresenta o desenvolvimento de uma ferramenta para monitoramento do perfil de condução de motoristas. Através da análise desse monitoramento, será possível os condutores analisarem sua forma de condução e se conscientizarem para uma direção mais segura.

### 1.1 OBJETIVOS

O objetivo é disponibilizar uma ferramenta para dispositivos Android para captação e análise de dados referentes a condução dos motoristas.

Os objetivos específicos são:

- a) obter os dados de rota do motorista;
- b) obter dados de condução do motorista (velocidade, força de curvas e aceleração);
- c) fazer a análise dos dados das viagens traçando um perfil para o motorista.

## 1.2 ESTRUTURA

Esta monografia está dividida em quatro capítulos. No primeiro capítulo é apresentada a introdução ao assunto do trabalho e descrito os objetivos do trabalho. O segundo capítulo destina-se a fundamentação teórica explorando conceitos de geolocalização, sensores inerciais, *serverless*, aceleração, frenagem, velocidade e os trabalhos correlatos. No terceiro capítulo é demonstrado o desenvolvimento do aplicativo, listando os requisitos, apresentando diagramas, técnicas e ferramentas utilizadas, operacionalidade do sistema e os resultados obtidos. Por fim, o quarto capítulo descreve as conclusões do trabalho e apresenta sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os principais assuntos utilizados na realização deste trabalho. Eles foram divididos em seis partes, a seção 2.1 aborda a geolocalização em aplicações móveis, a seção 2.2 apresenta os sensores vetoriais, a seção 2.3 apresenta conceitos relacionados a aceleração, velocidade e frenagem, na seção 2.4 é exposto os conceitos de *serverless* e a seção 2.5 apresenta os trabalhos correlatos a este.

### 2.1 GEOLOCALIZAÇÃO EM APLICAÇÕES MÓVEIS

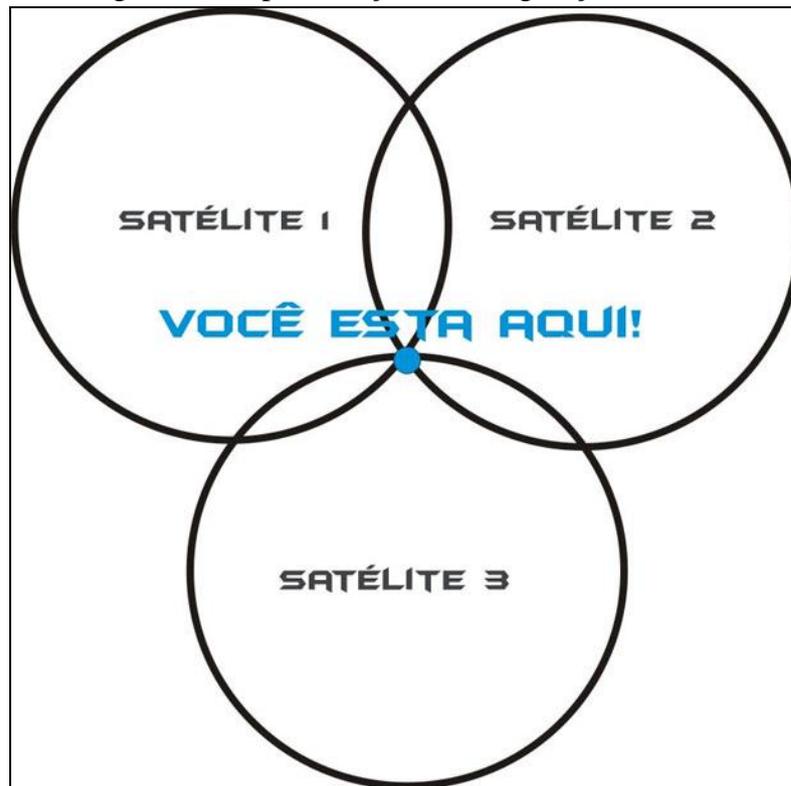
Geolocalização consiste na identificação da localização de pessoas/estabelecimentos em um objeto, o qual pode ser um radar, um GPS e até mesmo um celular (LONDERO et al., 2015). Conforme Bohrer (2011, p.50), quando se desenvolve aplicativos que fazem uso de geolocalização, é possível optar por qual meio se deseja obter as informações. Um desses meios é o GPS, que é uma tecnologia muito precisa em espaços abertos.

Segundo Martins (2009, p.1), “O GPS (Global Positioning System - Sistema de Posicionamento Global) é um aparelho que teve sua origem no Departamento de Defesa dos Estados Unidos. Sua função é identificar a localização de um aparelho chamado de receptor GPS”. Os receptores têm função de captar o sinal dos satélites, e fazendo alguns cálculos o receptor consegue determinar a posição (MARTINS, 2009, p.1). Para o funcionamento do GPS são necessários três componentes. O componente espacial é composto pelos satélites. O componente de controle são as estações de controle dos satélites. O último componente é o receptor GPS (MARTINS, 2009, p.1).

Segundo Martins (2009, p.1), os satélites possuem um relógio interno, quando um sinal é emitido, o satélite envia a hora que o sinal saiu. Estes sinais são enviados por ondas de rádio, que viajam na velocidade da luz. Cronometrando o tempo em que o sinal levou para chegar é possível saber a distância que se está do satélite.

Garrett (2011, p.1) afirma que três satélites são necessários para indicar a posição, um quarto satélite indica a altitude em relação ao nível do mar. Após obter o tempo de resposta de cada satélite, é calculada a distância de cada um dos três. Existe apenas um local no espaço onde as três distâncias irão ser as distâncias calculadas. Este é o ponto onde o dispositivo receptor está. A Figura 1 apresenta uma representação de como a triangulação utilizada pelo GPS funciona.

Figura 1 – Representação da triangulação do GPS



Fonte: Garrett (2018, p.1).

Conforme Cesani e Dranka (2012, p.124), “de maneira geral, os dispositivos móveis (mobile devices) incluem uma série de aparelhos, como celulares, smartphones, tablets, dentre outros”. Os dispositivos que utilizam GPS contam com bases de dados completas com informações de posicionamento, pontos de referência, rotas alternativas e posicionamento de radares de trânsito (CESANI; DRANKA, 2012, p.124).

O Google Maps Android API é uma Application Programming Interface (APIs) que fornece um serviço para acesso a localização em dispositivos Android. Essa API fornece também acesso a informações de rotas e dados de tráfego, bem como permite acesso a mapas e sobreposição de formas e marcadores no mapa (GOOGLE MAPS, 2018, p.1). A ferramenta do Google já implementa as funções necessárias para obtenção da localização através de sua API (SILVA, 2017, p.54).

Segundo Ribeiro Júnior et al. (2013), a função de Harvesine é bastante usada em sistemas de navegação, fornecendo a distância entre dois pontos de uma esfera utilizando coordenadas geográficas (latitude e longitude), realizando uma aproximação da terra como uma esfera perfeita e gerando um erro médio de 0,3% nos resultados do cálculo. O Quadro 1 demonstra a equação de Harvesine.

Quadro 1- Fórmula de Harvesine

$$\text{haversine}(\theta) \equiv \sin^2\left(\frac{\theta}{2}\right)$$

Fonte: adaptado de Ribeiro Júnior et al. (2013, p. 7).

Considerando dois pontos de uma esfera de raio  $R$ , com latitude e longitudes  $(l_1, \Delta_1)$ ,  $(l_2, \Delta_2)$ , respectivamente, a distância  $d$  é definida pela equação demonstrada no Quadro 2.

Quadro 2 - Equação de distância entre duas coordenadas

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{l_2 - l_1}{2}\right) + \cos(l_1) \cos(l_2) \sin^2\left(\frac{\Delta_2 - \Delta_1}{2}\right)}\right)$$

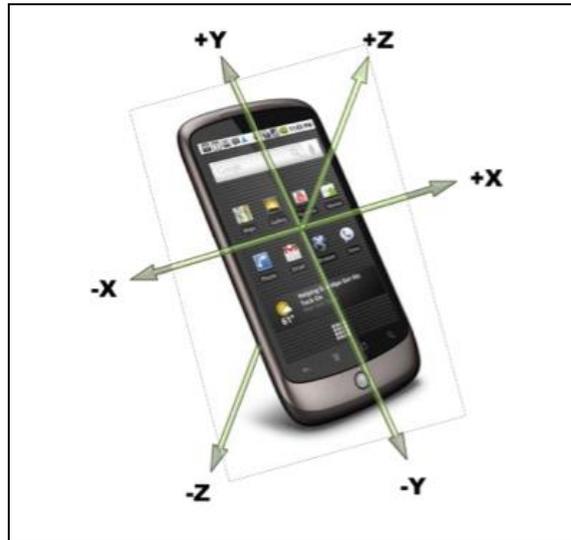
Fonte: adaptado de Ribeiro Júnior et al. (2013, p. 7).

## 2.2 SENSORES INERCIAIS

Segundo Torres (2014, p.1), “são chamados de sensores inerciais aqueles que têm por objetivo perceber os efeitos da ação de forças que provoquem uma mudança do estado inercial de sistemas sobre os quais estas forças são exercidas [...]”. Ainda segundo o autor, os acelerômetros são sensores “capazes de aferir a aceleração linear na direção de um eixo referencial” (TORRES, 2014, p.1).

A aceleração medida pelo sensor é diferente de velocidade, geralmente é mensurada em força  $G$ , que em resumo é a aceleração sentida como peso. Os acelerômetros mais comuns são os piezoelétrico, acelerômetro por indução magnética e acelerômetro por capacitância. Eles geralmente possuem três eixos:  $x$ ,  $y$ ,  $z$ . Por eles é possível detectar a orientação do dispositivo, os resultados são os dados de *pitch* (inclinação) e *roll* (rotação) (PAULA, 2015, p.1). A Figura 2 apresenta os eixos do acelerômetro de um dispositivo móvel.

Figura 2 - Eixos do acelerômetro

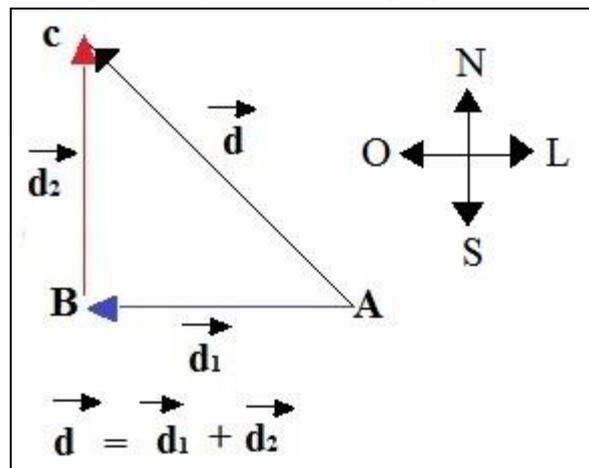


Fonte: Laboratório Imobilis (2013).

Os Giroscópios são sensores que medem a velocidade angular em torno de um eixo referencial (TORRES, 2014). Segundo Diniz (2010, p.1), “o giroscópio é, de fato, uma peça para complementar o acelerômetro [...]”. O giroscópio quantifica precisamente aceleração e velocidade angular nos eixos x, y e z. A utilização do giroscópio juntamente com os dados do acelerômetro aumenta a precisão dos dados obtidos (DINIZ, 2010).

De acordo com Teixeira (2018, p.1), “dois vetores são perpendiculares quando fazem um ângulo de 90° entre si”. A Figura 3 mostra a representação da soma de vetores.

Figura 3 - Representação de vetores perpendiculares entre si



Fonte: Teixeira (2018, p.1).

Ainda conforme Teixeira (2018, p.1), o deslocamento do ponto A até o ponto C é dado pelo vetor resultante d. A figura formada é um triângulo retângulo, e os vetores d1 e d2 são os catetos e o vetor d é a hipotenusa. Nesse caso a soma dos vetores d1 e d2 é dada pelo teorema de Pitágoras, apresentado no Quadro 3.

Quadro 3- Teorema de Pitágoras

$$d^2 = d_1^2 + d_2^2$$

Fonte: Teixeira (2018, p.1).

### 2.3 ACELERAÇÃO, FRENAGEM E VELOCIDADE

O artigo 43 do Código de Trânsito Brasileiro (CTB) afirma que ao regular a velocidade do veículo, o condutor deve obedecer aos limites máximos de velocidade da via. Ainda segundo o CTB, no artigo 61 está indicado que “a velocidade máxima permitida para a via será indicada por meio de sinalização, obedecidas suas características técnicas e as condições de trânsito” (BRASIL, 1997, p.33). Para o condutor que ultrapassar a velocidade da via, o artigo 218 do CTB define as infrações e penalidades a serem aplicadas ao condutor caso ultrapasse o limite de velocidade em até 20% (vinte por cento), até 50% (cinquenta por cento) ou acima de 50% (cinquenta por cento) (BRASIL, 1997, p.69).

Nunes (2019, p.1) aponta que o excesso de velocidade corresponde a maior parte das multas de transido emitidas em Santa Catarina. Ainda segundo Nunes (2019, p.1), “a principal violação é a de motoristas que transitam nas vias com velocidade superior à máxima permitida em até 20%.”

Segundo Moreira (2017), a direção agressiva, com mudanças bruscas de velocidade podem causar danos aos pneus, amortecedores e pastilhas de freio. Ainda segundo Moreira (2017), o aumento do giro do motor, além de aumentar o consumo de combustível, acaba desgastando o motor.

### 2.4 ARQUITETURA SERVERLESS

Segundo Stigler (2017) arquitetura *serverless*, também conhecido como Function as a Service (FaaS), é uma tecnologia que fornece um provedor de nuvem que retira a necessidade de gerenciamento de recursos operacionais de um servidor. Essa arquitetura elimina a necessidade de sistemas de execução contínua e viabiliza a criação de aplicativos escaláveis. Ainda conforme Stigler (2017), utilizando *serverless* é possível distribuir aplicativos funcionais e escalonáveis na metade do tempo necessário para construí-los do início.

Conforme Braga e Silva (2018), computação *serverless* é um novo paradigma de computação em nuvem. Ela descreve um modelo e arquitetura em que pequenos códigos são executados na nuvem sem controlar os recursos onde são executados. Segundo a Amazon (2019), uma arquitetura sem servidor, ou *serverless*, permite que o desenvolvedor se

concentre no produto em que trabalha, removendo a responsabilidade sobre gerenciamento de servidores ou tempo de execução. Ainda conforme Amazon (2019), isso permite a recuperação de tempo e energia que pode ser investido no desenvolvimento de produtos com escalabilidade e confiabilidade.

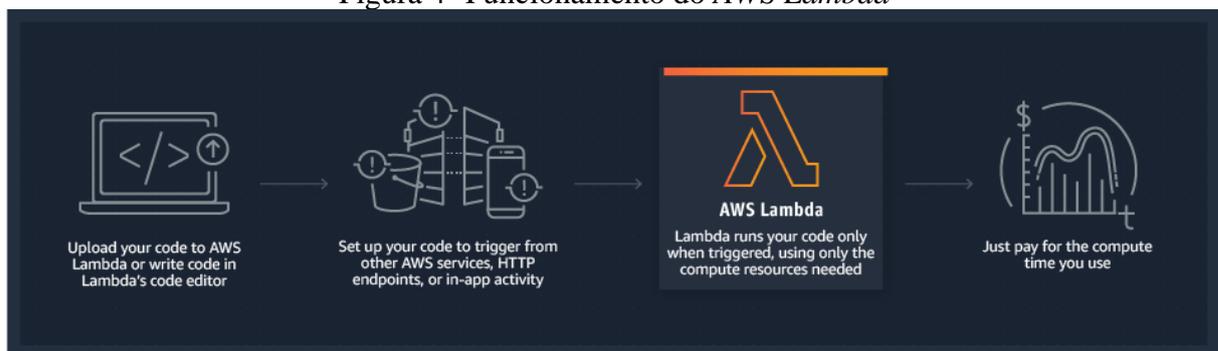
#### 2.4.1 AWS Lambda

Segundo Klems (2018), Amazon Web Services (AWS) é uma coleção de *web services* que formam uma plataforma de computação em nuvem que pode ajudar a tornar uma empresa mais eficiente. Ainda conforme Klems (2018), o AWS traz muitos benefícios, como alta flexibilidade, facilidade no uso, ótima relação custo-benefício, alta performance e escalabilidade.

O AWS Lambda permite que você execute códigos sem provisionar ou gerenciar servidores. Você paga apenas pelo tempo de computação que utilizar. Não haverá cobranças quando o seu código não estiver em execução (AMAZON, 2019). Ainda segundo a Amazon (2019), o AWS Lambda pode executar automaticamente código em resposta a vários eventos, como solicitações HyperText Transfer Protocol (HTTP) por meio do Amazon API Gateway, modificações de objetos armazenados na nuvem ou atualizações de tabela no banco de dados.

A Figura 4 apresenta o fluxo de funcionamento do AWS Lambda. No primeiro estágio há o carregamento do código para o AWS Lambda, após isso a configuração do gatilho que acionará o código. O terceiro estágio apresenta a execução do código quando o gatilho for acionado e por último o pagamento pelo uso do servidor na execução do código.

Figura 4- Funcionamento do *AWS Lambda*



Fonte: Amazon (2019).

## 2.5 TRABALHOS CORRELATOS

Neste capítulo são apresentados três trabalhos correlatos do trabalho desenvolvido. Os três são voltados para o gerenciamento e obtenção de informações de veículos. A seção 2.5.1 apresenta o projeto de Schlag (2017) na área de monitoramento da agressividade na direção

de caminhões através de acelerômetro e GPS. Na seção 2.5.2 é apresentado o desenvolvimento de aplicação para monitoramento veicular em tempo real, por Silva (2017). A seção 2.5.3 apresenta o aplicativo Trânsito+gentil (2018), da empresa Porto Seguro Auto, que propõe um jogo com o monitoramento do perfil do motorista.

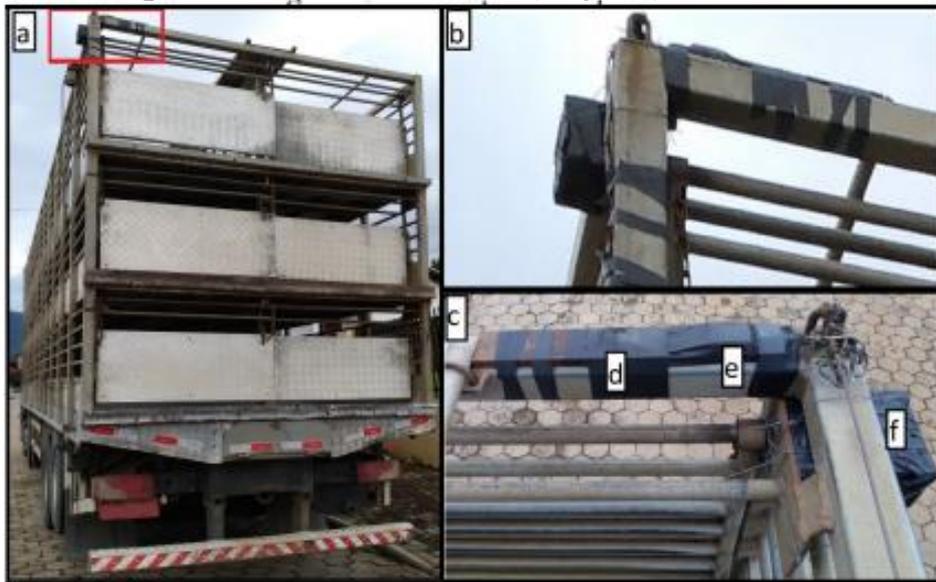
#### 2.5.1 Monitoramento da agressividade na direção de caminhões através de acelerômetro e GPS

O trabalho desenvolvido por Schlag (2017) apresenta o desenvolvimento de uma ferramenta que objetiva monitorar a agressividade de motoristas de caminhões, sobretudo a execução de uma manobra denominada “quebra de asa”. Para isso o sistema faz uso de um microcontrolador ESP8266, dois módulos de acelerômetro, Sistema de Posicionamento Global (Global Positioning System, GPS) e Global System for Mobile (GSM). Além disso, o sistema ainda possui um servidor Web.

O GPS detecta a velocidade e posição geográfica do caminhão, enquanto o acelerômetro é utilizado para detectar a inclinação do veículo. O servidor classifica os dados recebidos do dispositivo instalado no veículo e identifica se o motorista está dirigindo de forma perigosa.

Schlag (2017) utilizou no desenvolvimento do dispositivo o material Medium Density Fiberboard (MDF) ultra para desenvolver uma caixa para proteção e fixação. Segundo o autor esse material oferece boa resistência a água, sol e impactos. Para o plano de dados do cartão SIM808, foi escolhido o plano Infinity Pré Web 50 da empresa de telecomunicações TIM. O software embarcado no dispositivo foi desenvolvido na linguagem C++. A Figura 5 mostra o dispositivo montado acoplado no caminhão.

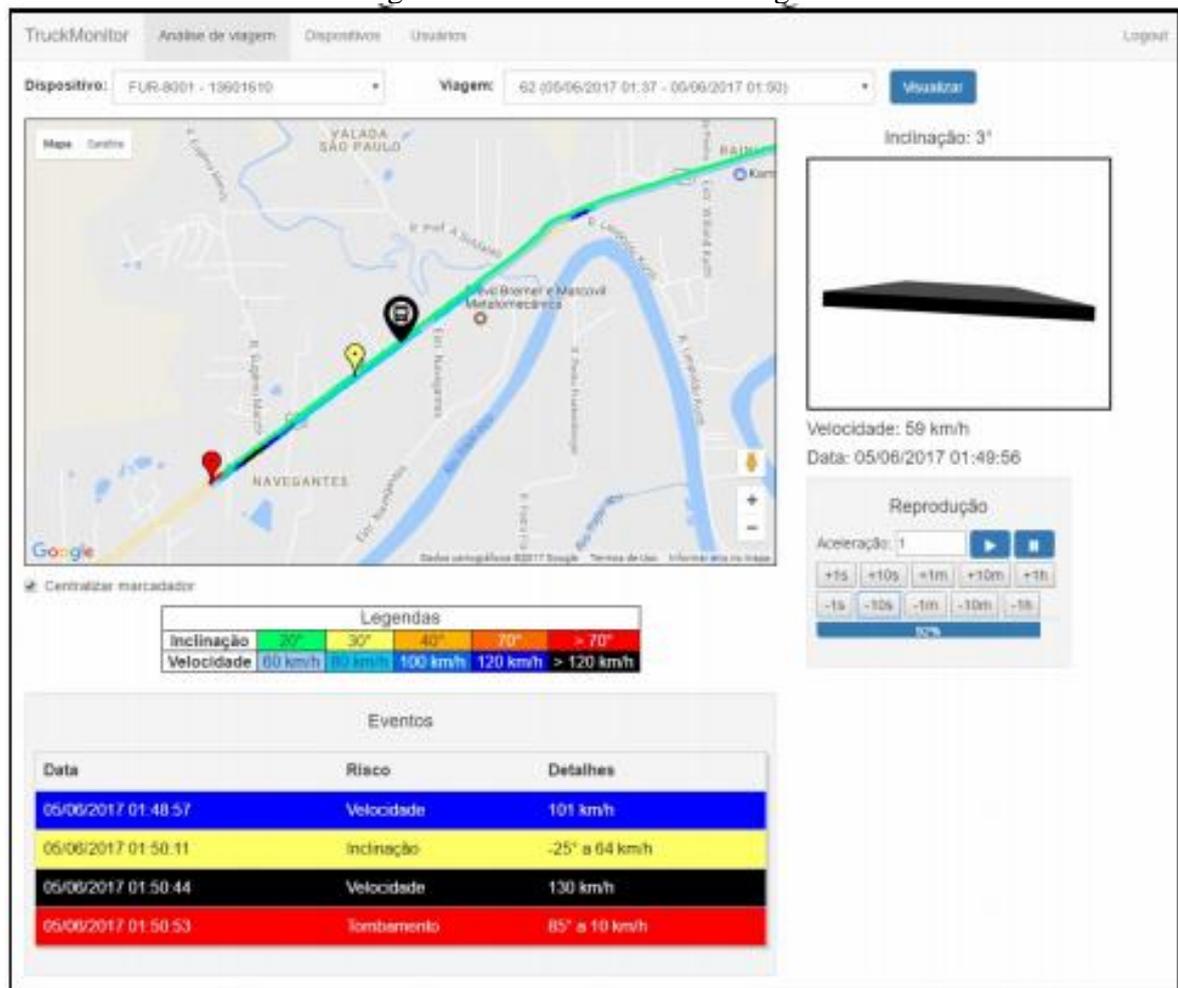
Figura 5 - Dispositivo montado (acoplado) no caminhão



Fonte: Schlag (2017).

No desenvolvimento do servidor Web da aplicação, Schlag (2017) utilizou a linguagem Java e o *framework* Spark para o *back-end*. Para a parte *front-end* ele utilizou as linguagens Javascript, HyperText Markup Language (HTML) e Cascading Style Sheets (CSS), junto com o *framework* AngularJS e as Application Programming Interface (API) do Google Maps e *WebGL*. O banco de dados utilizados para armazenamento dos dados coletados foi o PostgreSQL. A Figura 6 mostra a tela de análise da viagem em uma página Web com os dados colhidos pelo dispositivo.

Figura 6 - Tela de análise da viagem



Fonte: Schlag (2017).

### 2.5.2 Aplicação para monitoramento veicular em tempo real

O trabalho de Silva (2017) mostra o desenvolvimento de uma aplicação para monitorar veículos em tempo real. A aplicação informa possíveis falhas mecânicas ou furtos no veículo. Segundo Silva (2017), o principal objetivo da aplicação é informar a localização do automóvel, obter imagens e disponibilizar os dados da porta On-Board Diagnostic (OBD). O dispositivo desenvolvido ainda é capaz de enviar mensagens de e-mail e mensagens de texto em caso de falha no automóvel. Os dados captados pelo dispositivo embarcado são disponibilizados em uma aplicação móvel. A Figura 7 mostra o dispositivo instalado no veículo.

Figura 7 - Dispositivo instalado no veículo



Fonte: Silva (2017).

No trabalho foi utilizada a placa Raspberry Pi Zero W, juntamente com um módulo GPS, um adaptador Bluetooth e uma câmera. A câmera utilizada foi a WAVGAT Raspberry Pi Camera Rev 1.3 para captar as imagens internas do veículo. Para conexão foi utilizado um modem 3G ZTE MF626.

O sistema embarcado do dispositivo foi desenvolvido utilizando a linguagem Python. Para o sistema móvel foi utilizada a linguagem Typescript. A comunicação entre o dispositivo móvel e o sistema embarcado foram feitas utilizando HTTP com a formatação JSON.

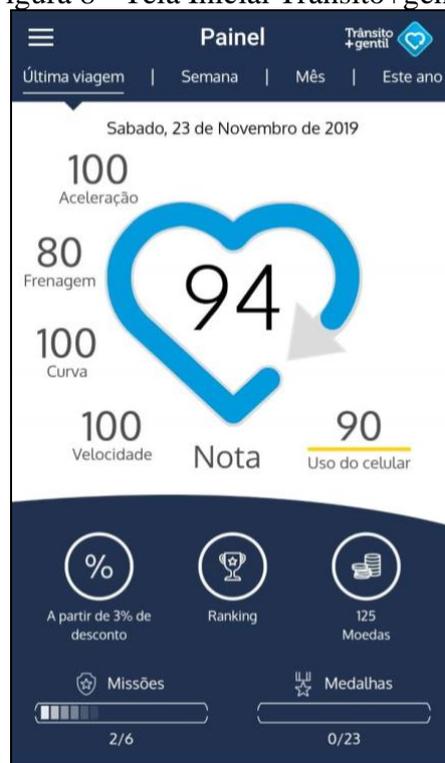
### 2.5.3 Trânsito+gentil

O Trânsito+gentil é um aplicativo desenvolvido pela empresa Porto Seguro Auto. O aplicativo “auxilia na dirigibilidade e incentiva o usuário a ser um motorista mais consciente, traduzindo comportamento em desconto no seguro” (TRÂNSITO+GENTIL, 2018, p.1).

O aplicativo tem como proposta ser um jogo para auxiliar a dirigir com mais segurança. É oferecida premiação para os motoristas que estiverem melhor no ranking. Para controlar o ranking, conforme o motorista pilotar, ele vai recebendo moedas. São ainda oferecidos missões e desafios diariamente ao usuário (TRÂNSITO+GENTIL, 2018, p.1).

O aplicativo está disponível para Android e iOS, podendo ser baixado pelas respectivas lojas. O monitoramento do motorista é constante, utilizando os sensores do celular. A Figura 8 apresenta a tela inicial do aplicativo Trânsito+gentil.

Figura 8 - Tela Inicial Trânsito+gentil



Fonte: digitalizado pelo autor.

#### 2.5.4 Correlação entre os trabalhos relacionados

No Quadro 4 é apresentada a comparação entre os trabalhos correlatos. As linhas representam características relacionadas e as colunas representam os trabalhos relacionados com o sistema desenvolvido.

Quadro 4 - Comparativo dos trabalhos correlatos

Características	Schlag (2017)	Silva (2017)	Trânsito+gentil (2018)
gerenciamento para vários veículos	Não	Não	Não
dispositivo para captação dos dados do veículo	Sim	Sim	Não
relatório de informações coletadas do veículo	Sim	Sim	Sim
monitoramento de perfil do motorista	Sim	Não	Sim
utilização do GPS para localização do veículo	Sim	Sim	Sim
cadastro de veículos	Não	Não	Não
interface da aplicação	Web	Android	Android, iOS
informação de condução em tempo real para o condutor	Não	Não	Não

Fonte: Elaborado pelo autor.

Se observa no Quadro 4, que tanto Schlag (2017) quanto Silva (2017) optaram por um dispositivo dentro do veículo para obtenção dos dados do veículo. Ambos desenvolveram um dispositivo que ficasse no veículo e enviasse os dados coletados para algum outro dispositivo para visualização. Schlag (2017) fez uso da plataforma Web para dispor os dados coletados. Silva (2017) utilizou-se da plataforma Android para apresentar os dados.

Schlag (2017) apresenta os relatórios dos dados captados através da plataforma Web, Silva (2017) apresenta os dados diretamente na tela do dispositivo móvel Android. O Trânsito+gentil (2018) apresenta os dados de perfil de motorista nas duas plataformas em que funciona (Android e iOS).

Os dispositivos de Schlag (2017) e Silva (2017) captam a localização do veículo através do módulo GPS que utilizam. Já o Trânsito+gentil (2018) capta os dados diretamente através do sensor GPS do celular. Schlag (2017) e o Trânsito+gentil (2018) fornecem um relatório com a rota percorrida e Silva (2017) utiliza o GPS para armazenar as localizações quando solicitado. Schlag (2017), Silva (2017) e Trânsito+gentil (2018) possuem a característica de fazer o monitoramento do veículo em tempo real, sendo que Schlag (2017) e o Trânsito+gentil (2018) monitoram a agressividade e padrão de condução, e Silva (2017) capta os dados de funcionamento mecânico do veículo ou furto.

### 3 DESENVOLVIMENTO

Neste capítulo são apresentadas as etapas relacionadas ao desenvolvimento da ferramenta. A seção 3.1 aborda o levantamento de informações e requisitos que envolvem a ferramenta. A seção 3.2 apresenta a especificação dos requisitos e diagramas desenvolvidos para ajudar na fase de desenvolvimento. A seção 3.3 expõe as técnicas e ferramentas utilizadas para a implementação, além disso mostra ainda a operacionalidade da implementação. Por fim, a seção 3.4 expõe a análise dos resultados e a relação entre os trabalhos correlatos e a ferramenta DrTrack desenvolvida, bem como a validação da ferramenta em comparação com o aplicativo Trânsito+gentil.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

Tendo em vista os problemas enfrentados pela falta de prudência dos motoristas ao dirigir, foi desenvolvida uma ferramenta que ajudasse a analisar o perfil do motorista, tendo como base os itens citados na seção 2.3 e **Error! Reference source not found.** para traçar os objetos da análise dos dados. Para efetuar essa análise, os dados são coletados através de sensores de um dispositivo Android. O objetivo é fazer a análise de dados referentes a aceleração, frenagem, curvas e velocidades do motorista, e com isso gerar um perfil para o motorista. A partir disso a análise foi dividida em três grupos, descritos a seguir:

- a) aceleração e frenagem: que será formado pela análise dos dados obtidos do sensor acelerômetro do dispositivo;
- b) velocidade: que será formado pela análise de dados de velocidade obtidos através do cálculo de deslocamento, fornecido pelo sensor do GPS, pelo tempo do trajeto;
- c) curvas: que será formado pelos dados obtidos do sensor acelerômetro do dispositivo.

A análise unificada desses grupos possibilitará traçar um perfil de condução do motorista, apontando a agressividade do condutor. A ferramenta trará dados reais sobre a sua forma de condução dos veículos para os condutores.

A seguir estão listados os Requisitos Funcionais (RF) atendidos pelo DrTrack:

- a) manter o cadastro de usuários;
- b) permitir acesso por login e senha;
- c) disponibilizar informação da localização do condutor;
- d) calcular automaticamente velocidade do veículo conduzido;
- e) captar automaticamente dados de inclinação e movimento do veículo conduzido;

- f) armazenar dados das viagens efetuadas pelo condutor (aceleração, velocidade, curvas e rota);
- g) efetuar análise dos dados captados para determinar perfil de condução do motorista;
- h) manter dados de velocidade máxima das vias para a análise da velocidade.

Os Requisitos Não Funcionais (RNF) atendidos pelo DrTrack são:

- a) o sistema móvel irá executar na plataforma Android;
- b) o sistema móvel será desenvolvido na linguagem Delphi para Android;
- c) os dados no dispositivo Android serão armazenados no banco de dados SQLite;
- d) na nuvem o banco de dados ficará armazenado no banco de dados DynamoDB;
- e) será utilizada a plataforma AWS Lambda para fazer a integração dados com a nuvem.

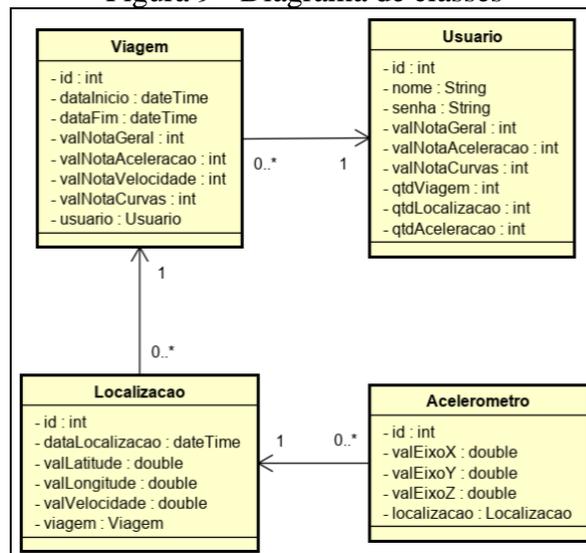
## 3.2 ESPECIFICAÇÃO

Esta seção apresenta a especificação do desenvolvimento da ferramenta, contendo, o diagrama de classes, diagrama de sequência, o Modelo de Entidade e Relacionamento (MER) e o diagrama de casos de uso.

### 3.2.1 Diagrama de classes

A Figura 9 apresenta o diagrama de classes de domínio do DrTrack, que mostra como as classes são estruturadas no desenvolvimento da ferramenta. Elas são utilizadas para instanciar objetos que serão persistidos no banco de dados do dispositivo. As classes de modelo utilizadas no desenvolvimento são: `Usuario`, `Acelerometro`, `Viagem` e `Localizacao`. A relação entre classes e atributos é apresentada na Figura 9.

Figura 9 - Diagrama de classes

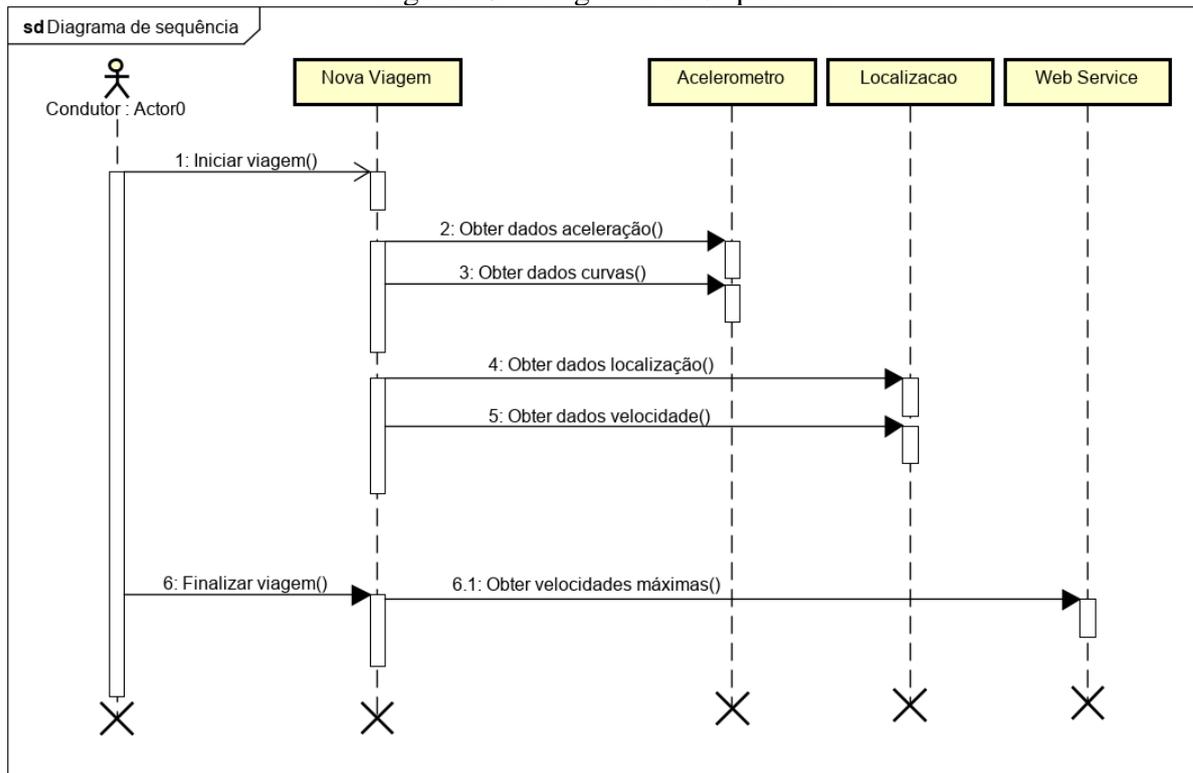


Fonte: elaborado pelo autor.

### 3.2.2 Diagrama de sequência

A Figura 10 apresenta o diagrama de sequência, que lista a sequência de passos que são executados no processo principal da ferramenta para monitorar a viagem. O diagrama detalha a linha de vida do ator Usuário, da tela para Nova Viagem, das classes Aceleracao e Localizacao e do Web Service. O diagrama começa com a ação de iniciar viagem do Usuário, após isso a tela Nova Viagem irá Obter os dados da aceleração, Obter os dados de curvas da classe Aceleração e Obter dados localização e Obter dados velocidade da classe Localizacao. Quando o Usuário executar a ação de Finalizar Viagem, na tela Nova Viagem, a tela irá chamar o processo para Obter as velocidades máximas do Web Service.

Figura 10 - Diagrama de sequência

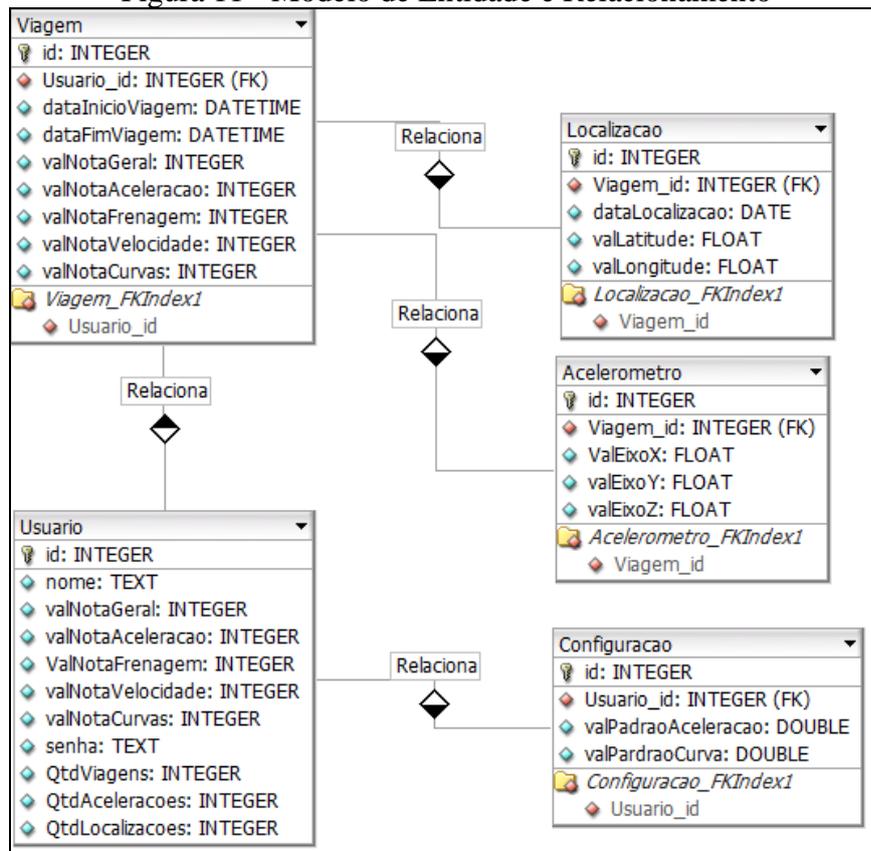


Fonte: elaborado pelo autor.

### 3.2.3 Modelo de Entidade e Relacionamento

Nessa subseção é apresentado o Modelo de Entidade e Relacionamento (MER) da ferramenta desenvolvida. A Figura 11 detalha a estrutura do banco de dados do DrTrack. As tabelas armazenadas no banco de dados são: *Usuario*, *Viagem*, *Localizacao*, *Acelerometro* e *Configuracao*. São listadas as tabelas, com seus respectivos atributos. A identificação dos atributos é feita da seguinte forma: a chave amarela representa os atributos que são chaves primárias, os atributos com losango vermelho são chaves estrangeiras e os atributos com losango azul são atributos comuns. O dicionário de dados referente a esse MER está listado no apêndice A.

Figura 11 - Modelo de Entidade e Relacionamento

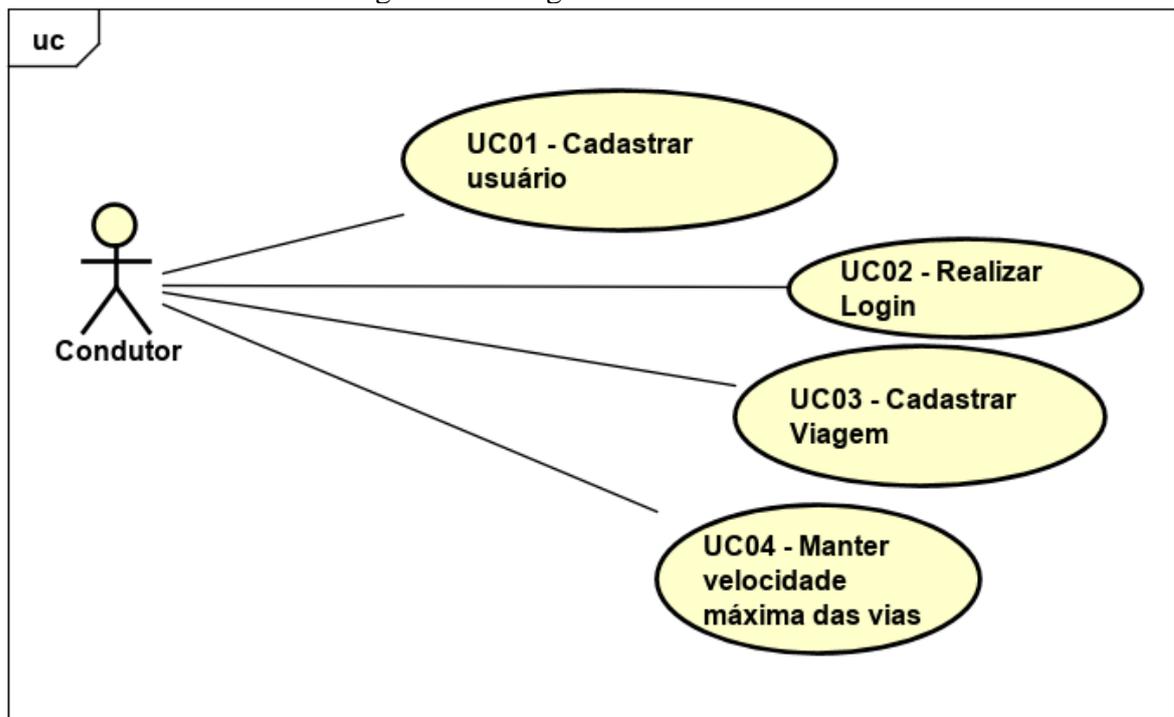


Fonte: elaborado pelo autor.

### 3.2.4 Diagrama de casos de uso

A Figura 12 apresenta o diagrama de casos de uso da ferramenta desenvolvida. O diagrama é composto por um ator, Usuário, e quatro casos de uso. O caso de uso UC01 - Cadastrar usuário, se refere ao cadastro e manutenção de usuário no banco de dados. O caso de uso UC02 - Realizar login, se refere a ação de login efetuada pelo Usuário para acessar o sistema. UC03 - Cadastrar viagem, é o caso de uso que se refere ao cadastro e manutenção de viagem e seus respectivos dados no banco de dados. O caso de uso UC04 - Manter velocidade máxima das vias se refere ao *web service* criado para cadastro e consultas de velocidade máxima para as vias.

Figura 12 - Diagrama de casos de uso



Fonte: elaborado pelo autor.

### 3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas para o desenvolvimento da ferramenta (seção 3.3.1), bem como a sua operacionalidade da implementação (seção 3.3.2). Também é descrito o desenvolvimento da ferramenta, apresentando trechos de códigos-fontes das principais rotinas desenvolvidas.

#### 3.3.1 Técnicas e ferramentas utilizadas

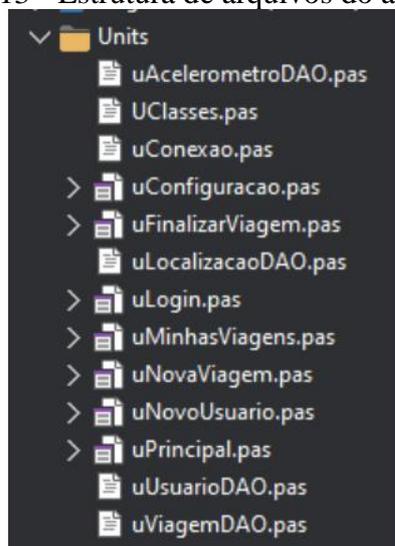
Esta seção apresenta as técnicas e ferramentas utilizadas para o desenvolvimento dessa ferramenta. Para a construção deste trabalho foram utilizadas as seguintes técnicas e ferramentas:

- a) Astah Community para desenho dos diagramas de classe, sequência e casos de uso;
- b) DBDesigner 4 para a modelagem do banco de dados do dispositivo móvel;
- c) SQLite como banco de dados no dispositivo móvel;
- d) Delphi 10.3 para o desenvolvimento da ferramenta para Android;
- e) Google Maps API para visualização e desenho do mapa;
- f) AWS Lambda para criação do *web service* que armazena e retorna as velocidades máximas das vias;
- g) AWS API Gateway para fazer tratamento das requisições HTTP que chegam ao AWS Lambda;

- h) linguagem Node.js para o *web service* montado;
- i) DynamoDB como banco de dados para armazenar as velocidades máximas das vias;
- j) fórmula de Haversine para calcular a distância entre coordenadas geográficas.

O aplicativo para dispositivos Android foi desenvolvido utilizando a linguagem de programação Delphi, na versão 10.3, que gera código nativos para o Android e contém uma série de componentes prontos para integração com os sensores do dispositivo. A estrutura dos arquivos dentro do projeto está distribuída conforme apresentado na Figura 13.

Figura 13 - Estrutura de arquivos do aplicativo



Fonte: elaborado pelo autor.

A *unit* `UClasses` contém a declaração das classes da ferramenta, correspondente a camada de modelo. As *units* `uConfiguracao`, `uFinalizarViagem`, `uLogin`, `uMinhasViagens`, `uNovaViagem`, `uNovoUsuario` e `uPrincipal` correspondem as telas do sistema, ou seja, a camada de visualização. As *units* `uAcelerometroDAO`, `uLocalizacaoDAO`, `uUsuarioDAO`, `uViagemDAO` são correspondentes a camada de controle do aplicativo no dispositivo. A *unit* `uConexao` é a camada de conexão com o banco de dados, nela também é feita a criação e atualização das tabelas do banco de dados.

A conexão com o banco de dados SQLite é feita através dos componentes `FireDac` que são nativos do Delphi, e proporcionam uma série de componentes para conexão e manipulação de dados. O Quadro 5 mostra o trecho de código que inicia a conexão com o banco de dados.

Quadro 5 - Conexão com banco de dados

```

48 DBConnection:= TFDConnection.Create(nil);
49 try
50   DBConnection.Params.Values['DriverID'] := 'SQLite';
51   DBConnection.LoginPrompt := False;
52   {$IF DEFINED (ANDROID) || (IOS)}
53     DBConnection.Params.Values['Database'] := TPath.Combine(TPath.GetDocumentsPath, 'drTrack.db');
54   {$ENDIF}
55   {$IF DEFINED (MSWINDOWS)}
56     DBConnection.Params.Values['DataBase'] := '{CAMINHO_DB}';
57   {$ENDIF}
58   DBConnection.Connected := True;

```

Fonte: elaborado pelo autor.

O objeto `DBConnection` é o componente que fará a conexão com o banco de dados. Na linha 50 é atribuído o *driver* do SQLite para o objeto. Na linha 53 é feita a atribuição do caminho do banco de dados no ambiente Android. Na linha 56 é feita a atribuição do caminho do banco de dados no ambiente Windows, utilizado somente para testes. Na linha 58 é aberta a conexão efetivamente.

O Android possui uma política de permissões para acesso de algumas informações do dispositivo. Para o acesso a localização, por exemplo, é necessário solicitar ao sistema operacional se o aplicativo possui permissão de acesso a esse recurso. Essas permissões são solicitadas no arquivo `AndroidManifest.xml`. O Quadro 6 exibe a configuração das permissões necessárias da ferramenta.

Quadro 6 - Configurando Permissões

```

10 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
11 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
12 <uses-permission android:name="android.permission.WAKE_LOCK" />

```

Fonte: elaborado pelo autor.

Nas linhas 10 e 11 do Quadro 6 estão listadas as permissões para acesso à localização do dispositivo. A linha 12 solicita permissão à função de manter a tela do dispositivo ligada para a visualização da viagem. As políticas de permissão do Android determinam ainda que a permissão para acesso da localização é uma permissão crítica, e requer permissão expressa do usuário, para isso a solicitação da permissão de acesso a localização deve ser feita quando ela for necessária. No Quadro 7 está o trecho de código que faz a solicitação para o usuário para permissão de acesso ao sensor de localização.

Quadro 7 - Solicitação de permissão

```

128 PermissionsService.RequestPermissions([JStringToString(TJManifest_permission.JavaClass.ACCESS_FINE_LOCATION)],
129 procedure(const APermissions: TArray<string>; const AGrantResults: TArray<TPermissionStatus>)
130 var
131     fNovaViagem : TFNovaViagem;
132 begin
133     if (Length(AGrantResults) = 1) and (AGrantResults[0] = TPermissionStatus.Granted) then
134     begin
135         try
136             fNovaViagem := TFNovaViagem.Create(self);
137             fNovaViagem.Show;
138         except on E: Exception do
139             showMessage(e.Message);
140         end;
141     end
142     else
143     begin
144         ShowMessage('Location permission not granted');
145     end;
146 end)

```

Fonte: elaborado pelo autor.

Neste trecho de código é feita a requisição de permissão para o usuário no método `RequestPermissions`, que recebe como atributo um procedimento que testa se a permissão foi concedida, abre a tela para nova viagem (linhas 136 e 137) ou retorna mensagem de erro caso a permissão não tenha sido concedida (linha 144).

Para obter os dados dos sensores, foram utilizados os componentes padrões do Delphi. O acelerômetro possui o componente `TMotionSensor` para manipulação. Este componente retorna a força que está agindo nos eixos X, Y e Z na unidade g (força da gravidade). Para a captação dos dados foi adicionado um temporizador que aciona a cada 500 milissegundos e grava a informação do sensor de acelerômetro. O Quadro 8 mostra o trecho que é executado na ação do temporizador para gravar os valores dos eixos X, Y e Z na lista de acelerações instanciada na memória. A função `abs` retorna o valor em módulo, para que esse sensor controle a aceleração e frenagem como uma só nota, sem diferenciar se a força é positiva (aceleração) ou negativa (frenagem).

Quadro 8 - Obtenção dos dados do acelerômetro

```

373     aceleracao := TAcelerometro.Create;
374     aceleracoes.Add(aceleracao);
375     aceleracao.setValEixoX(abs(senAcelerometro.Sensor.AccelerationX));
376     aceleracao.setValEixoY(abs(senAcelerometro.Sensor.AccelerationY));
377     aceleracao.setValEixoZ(abs(senAcelerometro.Sensor.AccelerationZ));
378     if localizacao <> nil then
379         aceleracao.setLocalizacao(localizacao);
380

```

Fonte: elaborado pelo autor.

O sensor de localização do dispositivo é acessado pelo componente `TLocationSensor`. Esse componente possui uma propriedade em que é configurada a distância que deve ser percorrida para que seja acionado o evento `LocationChange`. Foi atribuído à propriedade `distancia` o valor de dez metros, ou seja, a cada dez metros percorridos, o sistema irá gravar um novo ponto de localização geográfica, com latitude e longitude. O Quadro 9 mostra a atribuição de valores para a lista de localizações da viagem.

Os valores de `aLatitude` e `aLongitude` vem do sensor de localização, o valor de `now` é a data e hora do dispositivo.

Quadro 9 - Obtenção dos dados de localização

```

276 |   localizacao := TLocalizacao.Create;
277 |   localizacao.setDataLocalizacao(now);
278 |   localizacao.setValLatitude(aLatitude);
279 |   localizacao.setValLongitude(aLongitude);
280 |   localizacao.setViagem(viagem);

```

Fonte: elaborado pelo autor.

O sensor de localização possui algumas falhas, retornando coordenadas que não correspondem às coordenadas reais do dispositivo. Para remover esse erro, são considerados somente as coordenadas a uma distância de até 25 (vinte e cinco) metros da coordenada anterior. Para isso é necessário calcular a distância entre duas coordenadas, e para esse cálculo foi utilizada a fórmula de Haversine (ver seção 2.1). Para o cálculo de distância de coordenadas geográficas, é utilizada uma aproximação do raio médio terrestre de 6.371 quilômetros. O Quadro 10 apresenta o cálculo da distância entre duas coordenadas geográficas, identificadas por latitude e longitude.

Quadro 10 - Cálculo da distância entre coordenadas

```

376 | function sgn(a: real): real;
377 | begin
378 |   if a < 0 then
379 |     result := -1
380 |   else
381 |     result := 1;
382 | end;
383 |
384 | function atan2(y, x: real): real;
385 | begin
386 |   if x > 0 then
387 |     result := arctan(y/x)
388 |   else
389 |     if x < 0 then
390 |       result := arctan(y/x) + pi
391 |     else
392 |       result := pi/2 * sgn(y);
393 | end;
394 | begin
395 |   lat1 := latitude*pi/180;
396 |   lat2 := getValLatitude*pi/180;;;
397 |   lng1 := longitude*pi/180;;;
398 |   lng2 := getValLongitude*pi/180;;;
399 |
400 |   Lat := lat1 - lat2;
401 |   Lng := lng1 - lng2;
402 |
403 |   Val := sin(Lat / 2) * sin(Lat / 2) + cos(lat2) * cos(lat1) * sin(Lng / 2) * sin(Lng / 2);
404 |   Val := 2 * ATan2(sqrt(Val), sqrt(1 - Val));
405 |
406 |   Result:= (r * Val)*1000;
407 | end;

```

Fonte : elaborado pelo autor.

O cálculo de distância entre duas coordenadas também é utilizado para calcular a velocidade do veículo. É utilizada a fórmula simples de velocidade média para isso, dividindo

a distância percorrida pelo tempo. A cada coordenada nova é calculado a velocidade do deslocamento para fornecer ao usuário essa informação quase em tempo real.

Para fazer a construção e desenho do mapa na ferramenta, foi utilizado o Google Maps API. Ele possui os componentes prontos para exibição do mapa e seus recursos, o Delphi 10.3 possui também componentes que facilitam a integração com a API. Para o serviço do Google Maps API funcionar, é necessário configurar a chave da API no `AndroidManifest.xml`.

No Delphi o componente `TMapView` é utilizado para a exibição dos mapas do Google Maps. É possível adicionar formas geométricas e desenhar linhas, passando as coordenadas geográficas para que sejam pintados no mapa. O Quadro 11 apresenta o código para fazer o desenho da rota traçada no mapa.

Quadro 11 - Desenho da rota traçada no mapa

```

246     if localizacaol <> nil then
247     begin
248         myline := TMapPolylineDescriptor.Create(TArray<FMX.Maps.TMapCoordinate>(coordenadas));
249         myline.StrokeColor := TAlphaColorRec.Blue;
250         myline.StrokeWidth := 20;
251         mapa.AddPolyline(MyLine);
252     end;
253
254     if circuloMapa <> nil then
255         circuloMapa.Remove;
256
257     circulo := TMapCircleDescriptor.Create(coordenadaMapa, 5);
258     circulo.StrokeWidth := 5;
259     circulo.StrokeColor := TAlphaColorRec.Blue;
260     circulo.FillColor := TAlphaColorRec.Aquamarine;
261     circuloMapa := mapa.AddCircle(circulo);
262
263     circulo := TMapCircleDescriptor.Create(coordenadaMapa, 2);
264     circulo.StrokeWidth := 5;
265     circulo.StrokeColor := TAlphaColorRec.red;
266     circulo.FillColor := TAlphaColorRec.red;
267     mapa.AddCircle(circulo);
268 end;

```

Fonte: elaborado pelo autor.

As linhas de 248 até 251 fazem o desenho da linha da rota traçada pelo veículo no mapa. Para criar essa linha é passado uma lista de coordenadas geográficas, que são os pontos onde o veículo passou. Nas linhas 257 até 267 é feito o desenho de um pequeno círculo que representa a posição atual do veículo no mapa, ou seja, a coordenada mais recente a ser adicionada na lista de coordenadas da viagem.

Quando o usuário finaliza a viagem a ferramenta calcula as notas para a viagem. É feito o cálculo de notas para aceleração, velocidade, curvas e a nota geral da viagem. A análise da velocidade é feita utilizando as velocidades máximas das vias já cadastradas no *web service* da ferramenta. Para fazer a consulta das velocidades máximas foi criado um *web service* utilizando AWS Lambda. Como linguagem de programação no *web service* foi utilizado o Node.js 8.10. Para armazenar os dados foi utilizado o banco de dados DynamoDB,

que também possui integração facilitada com o AWS Lambda. O Quadro 12 apresenta o trecho de código no AWS Lambda que faz a consulta da velocidade máxima da via, recebendo a latitude e longitude como parâmetro.

Quadro 12 - Consulta velocidade máxima via

```

21
22 async function read_velocidadeMaxima(aLatitude, aLongitude) {
23   var params, params1;
24
25   params = {
26     TableName: table,
27     ProjectionExpression: "latitude, longitude, Velocidade",
28
29     FilterExpression: "begins_with(#lat, :latv) and begins_with(#lon, :lonv)",
30     ExpressionAttributeNames: {
31       "#lat": "latitude",
32       "#lon" : "longitude",
33     },
34     ExpressionAttributeValues: {
35       ":latv": aLatitude,
36       ":lonv" : aLongitude
37     }
38   };
39   params1 = {
40     TableName: table,
41   };
42
43   try {
44     const data = dynamoDB.scan(params).promise();
45     return data;
46     //return { statusCode: 200, body: JSON.stringify(data) };
47   } catch (error) {
48     return {
49       statusCode: 400,
50       error: `Could not fetch: ${error.stack}`
51     };
52   }
53 }

```

Fonte: elaborado pelo autor.

Após obter as velocidades máximas das coordenadas, a ferramenta calcula a nota para velocidade utilizando o CTB (seção 2.3) como base. Para isso se utilizou as definições das multas em caso de excesso de velocidade. O Quadro 13 apresenta o cálculo das notas em relação a velocidade nas coordenadas.

Quadro 13 - Cálculo nota velocidade

```

47  if aLocalizacoes[ilaco].getValVelocidade <= retornarVelocidadeMaximaVia(aLocalizacoes[ilaco]) then
48  begin
49      result := result + 100;
50  end
51  else
52  begin
53      if aLocalizacoes[ilaco].getValVelocidade <= (retornarVelocidadeMaximaVia(aLocalizacoes[ilaco]) + 7) then
54      begin
55          result := result + 75;
56      end
57      else
58      begin
59          if aLocalizacoes[ilaco].getValVelocidade <= (retornarVelocidadeMaximaVia(aLocalizacoes[ilaco]) +
60              (retornarVelocidadeMaximaVia(aLocalizacoes[ilaco]) * 0.2)) then
61          begin
62              result := result + 50;
63          end
64          else
65          begin
66              if aLocalizacoes[ilaco].getValVelocidade <= (retornarVelocidadeMaximaVia(aLocalizacoes[ilaco]) +
67                  (retornarVelocidadeMaximaVia(aLocalizacoes[ilaco]) * 0.5)) then
68              begin
69                  result := result + 25;
70              end
71              end;
72          end;
73      end;
74  end;
75  result := round(result / aLocalizacoes.Count);

```

Fonte: elaborado pelo autor.

As coordenadas com velocidade abaixo da velocidade máxima vão receber a nota 100 (linha 49). Caso seja superior a velocidade máxima em até 7 km/h, que é a tolerância segundo o CTB, a coordenada receberá nota 75 (linha 55). Se for maior que 7 km/h acima da máxima e superior em até 20% da velocidade máxima a nota será 50 (linha 62). Caso seja superior a máxima em até 50% a nota será 25 (linha 69). Acima de 50% da velocidade máxima a coordenada receberá nota 0. Com as notas atribuídas para cada coordenada é feito uma média simples para obter a nota geral da velocidade (linha 75).

Para calcular a nota para a aceleração, foi necessário analisar os eixos X, Y e Z do dispositivo (ver seção 2.2). Foi definido que o dispositivo irá ficar sempre na posição retrato, portanto, os eixos em que incidem a aceleração do veículo são os eixos Y e Z. A aceleração resultante nesses dois eixos é calculada com a soma de vetores, que é feita através do teorema de Pitágoras. A gravidade também está em ação nos eixos, o valor da aceleração da gravidade é desconsiderado, apenas a variação da força de aceleração em relação ao estado de velocidade constante é considerada.

A classificação da aceleração foi feita em quatro níveis: normal, alta, muito alta e extrema. Quando a variação da aceleração em relação ao valor padrão é em até 5% a aceleração é classificada como normal. Aceleração com variação entre 5% e 11% é classificada como alta. A variação entre 11% e 17% é classificada como muito alta. Acima de 17% a aceleração é extrema. Os percentuais foram estabelecidos através de testes comparando com o Trânsito+gentil. Após atribuir os valores, foi aplicado um filtro de média com máscara

3, para amenizar os ruídos da aceleração que ficam fora do desvio padrão. O Quadro 14 mostra o procedimento para a aplicação da máscara de média na lista de acelerações.

Quadro 14 - Procedimento para média mascarada 3

```

185 procedure filtroMediaMascara3(lista : TList<double>);
186 var
187     numeroAtual, numeroAnterior, numeroPosterior : Double;
188     laco: Integer;
189 begin
190     if lista.Count > 1 then
191     begin
192         numeroAtual := lista[0];
193         numeroPosterior := lista[1];
194         lista[0] := (numeroAtual + numeroPosterior)/2;
195
196         for laco := 1 to lista.Count -2 do
197         begin
198             numeroAtual := lista[laco];
199             numeroAnterior := lista[laco-1];
200             numeroPosterior := lista[laco+1];
201             lista[laco] := (numeroAtual + numeroAnterior + numeroPosterior)/3;
202         end;
203         numeroAtual := lista[lista.Count-1];
204         numeroAnterior := lista[lista.Count-2];
205         lista[laco] := (numeroAtual + numeroAnterior)/2;
206     end;
207 end;

```

Fonte: elaborado pelo autor.

Para o cálculo da nota de curvas, foi utilizado o eixo X do sensor de acelerômetro. A classificação é semelhante à aceleração. Curvas de até 0,4 g são classificadas como normal. curvas de 0,4 g até 0,7 g são classificadas como fortes. De 0,7 g até 1 g as curvas são classificadas como muito fortes. Acima de 1 g a classificação da curva é extrema. Os valores das forças por níveis foram estabelecidos através de testes comparando com o Trânsito+gentil. O Quadro 15 apresenta a classificação das curvas conforme listado anteriormente.

Quadro 15 - Análise de classificação das curvas

```

281 function TAcelerometro.getTipoCurva: TTipoAceleracao
282 Const
283     CURVA_NORMAL = 0.4;
284     CURVA_FORTE = 0.7;
285     CURVA_MUITO_FORTE = 1;
286 begin
287     result := taExtrema;
288     if Abs(getValEixoX) <= CURVA_NORMAL then
289         result := taNormal
290     else
291         if Abs(getValEixoX) <= CURVA_FORTE then
292             result := taForte
293         else
294             if Abs(getValEixoX) <= CURVA_MUITO_FORTE then
295                 result := taMuitoForte;
296     end;

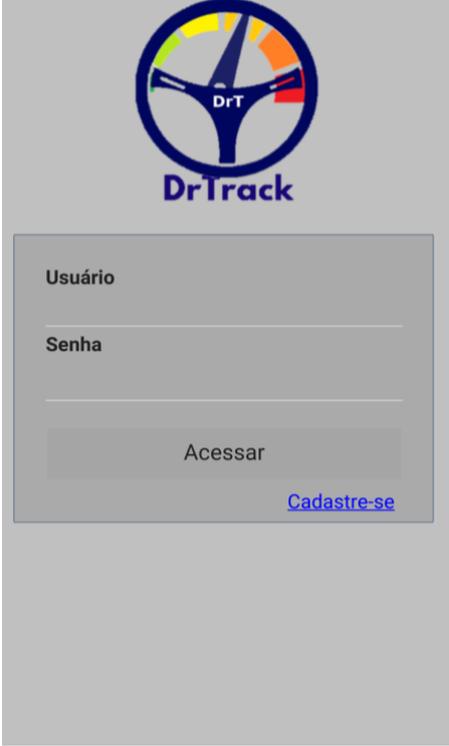
```

Fonte: elaborado pelo autor.

### 3.3.2 Operacionalidade da implementação

Nesta seção é apresentado o funcionamento da ferramenta DrTrack através de capturas das telas, seguindo o roteiro padrão de operação do sistema para um novo usuário. A primeira tela ao acessar o aplicativo é a tela de Login, que está apresentada na Figura 14.

Figura 14 - Tela de login



Usuário

Senha

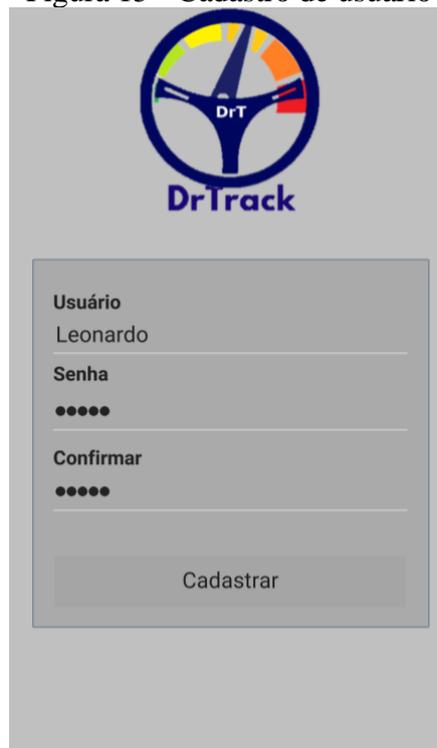
Acessar

[Cadastre-se](#)

Fonte: elaborado pelo autor.

A tela possui os campos para o usuário informar **Usuário** e **Senha** para efetuar o login. Clicando no botão `cadastre-se` uma nova tela é aberta para cadastrar um novo usuário. A Figura 15 mostra a tela para cadastro de usuário.

Figura 15 - Cadastro de usuário



The image shows a registration form for 'DrTrack'. At the top is the logo, which consists of a steering wheel with a speedometer and the text 'DrT' and 'DrTrack'. Below the logo is a form with three input fields: 'Usuário' containing 'Leonardo', 'Senha' with five black dots, and 'Confirmar' with five black dots. A 'Cadastrar' button is positioned below the 'Confirmar' field.

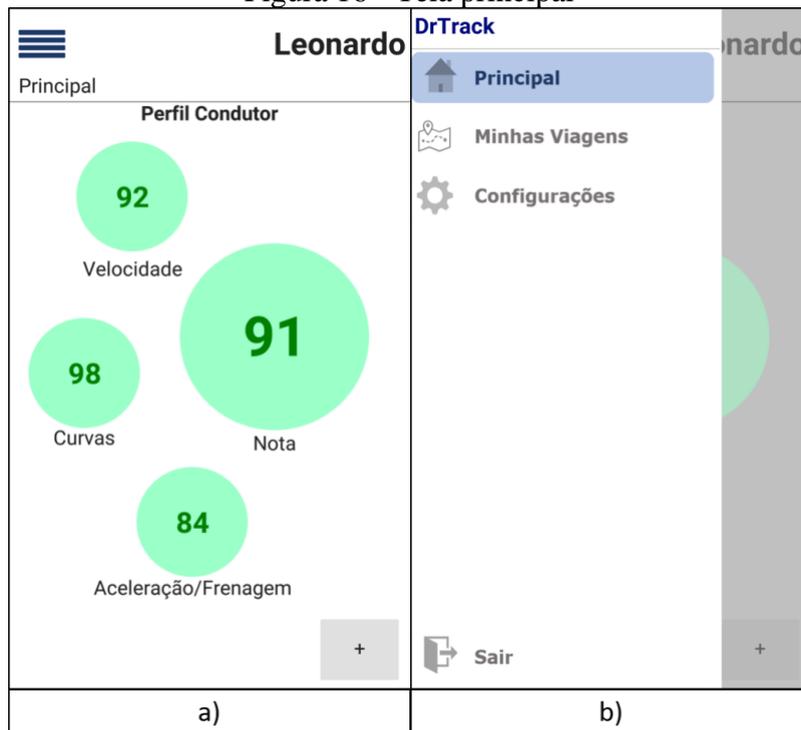
Fonte: elaborado pelo autor.

O cadastro de usuário solicita um nome de usuário que será usado para efetuar o login, a senha e sua confirmação. Ao clicar em cadastrar o usuário será cadastrado e retornará para a tela de login.

Ao efetuar o login caso usuário ou senha estejam incorretos uma mensagem de erro é exibida na tela e o login não é efetuado. Se o login for efetuado com sucesso o usuário será redirecionado para a tela principal do sistema, apresentada na Figura 16.

Na tela principal (Figura 16 a) está destacado no centro as notas médias do condutor. O perfil do condutor é composto por uma nota para aceleração e frenagem, uma nota para velocidade, uma nota para curvas e a nota geral, que é a média dos 3 atributos (aceleração/frenagem, velocidade e curvas). A cor dos círculos e dos números das notas de cada atributo varia de acordo com seu valor. Notas mais baixas, por exemplo ficam com a cor vermelha.

Figura 16 - Tela principal



Fonte: elaborado pelo autor.

No canto superior esquerdo está o botão para o menu (Figura 16 b), que contém as opções para abrir a tela consulta de viagens (Minhas Viagens), a tela de Configurações e o botão para sair da conta do usuário. Na tela principal, existe ainda o botão para cadastrar nova viagem, no canto inferior direito. A Figura 17 apresenta a captura da tela de nova viagem.

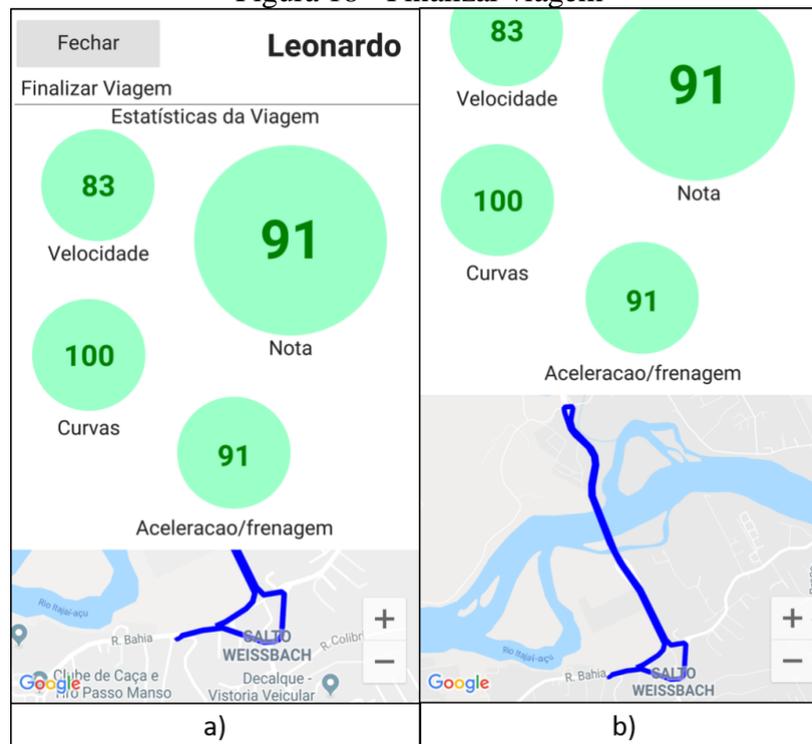
Figura 17 - Tela nova viagem



Fonte: elaborado pelo autor.

A tela nova viagem vai atualizando os dados em tempo real. O campo *Curva*, na parte superior, mostra a classificação da curva que está sendo feita. No campo *Velocidade* é apresentado a velocidade atual do deslocamento em quilômetros por hora (km/h). O campo *Aceleração* apresenta a classificação da força de aceleração ou frenagem que está agindo no momento. O mapa na parte inferior da tela apresenta o traçado da rota conforme o deslocamento do usuário com o dispositivo. Ao clicar no botão *Finalizar*, a tela de resumo da viagem é exibida, conforme mostra a Figura 18.

Figura 18 - Finalizar viagem



Fonte: elaborado pelo autor.

As estatísticas da viagem são exibidas da mesma forma que o perfil de condutor é exibido na tela principal, com notas para velocidade, curvas, aceleração e frenagem. Por fim a média das notas dos atributos resulta na nota geral. Na parte inferior da tela é exibido no mapa o trajeto completo da viagem. Esse mapa pode ser ampliado arrastando os atributos para cima, conforme mostra a Figura 18b. As viagens já efetuadas podem ser consultadas através do botão no menu principal. A tela de consulta de viagens é apresentada na Figura 19.

A tela *Minhas Viagens* exibe todas as viagens efetuadas pelo usuário, identificadas através da data e hora de início da viagem. Ao clicar sobre uma viagem, as estatísticas da viagem são exibidas, utilizando a tela já apresentada na Figura 18.

Figura 19 - Tela Minhas Viagens

Leonardo	
Minhas Viagens	
Viagem 21/11/2019	20:20:42 >
Viagem 21/11/2019	20:30:25 >
Viagem 21/11/2019	20:31:01 >
Viagem 21/11/2019	20:31:29 >
Viagem 21/11/2019	20:31:55 >
Viagem 21/11/2019	20:32:33 >
Viagem 21/11/2019	20:32:52 >
Viagem 21/11/2019	22:08:40 >
Viagem 21/11/2019	22:09:40 >
Viagem 21/11/2019	22:45:55 >
Viagem 22/11/2019	07:57:44 >

Fonte: elaborado pelo autor.

Existe ainda uma tela de configuração do sistema que está representada na Figura 20. O campo *Velocidade Máxima* serve para cadastrar uma nova velocidade máxima para uma via. Ao clicar no botão *Adicionar Velocidade Máxima*, a ferramenta joga o valor informado no campo para o *web service*, atribuindo essa velocidade para a localização atual do usuário. O botão *Limpar Banco de Dados* é usado para apagar todos os dados do banco de dados, caso seja necessário para fazer algum teste.

Figura 20 - Configuração do sistema

Leonardo	
Configurações	
Velocidade Máxima	50
Adicionar Velocidade Máxima	
Limpar Dados	

Fonte: elaborado pelo autor.

### 3.4 RESULTADOS E DISCUSSÕES

Com base nos trabalhos correlatos apresentados na seção 2.5, a ferramenta desenvolvida apresenta semelhanças com os três trabalhos. O Quadro 16 apresenta as semelhanças entre os trabalhos correlatos e o DrTrack.

Quadro 16 - Comparativo entre trabalhos correlatos e o DrTrack

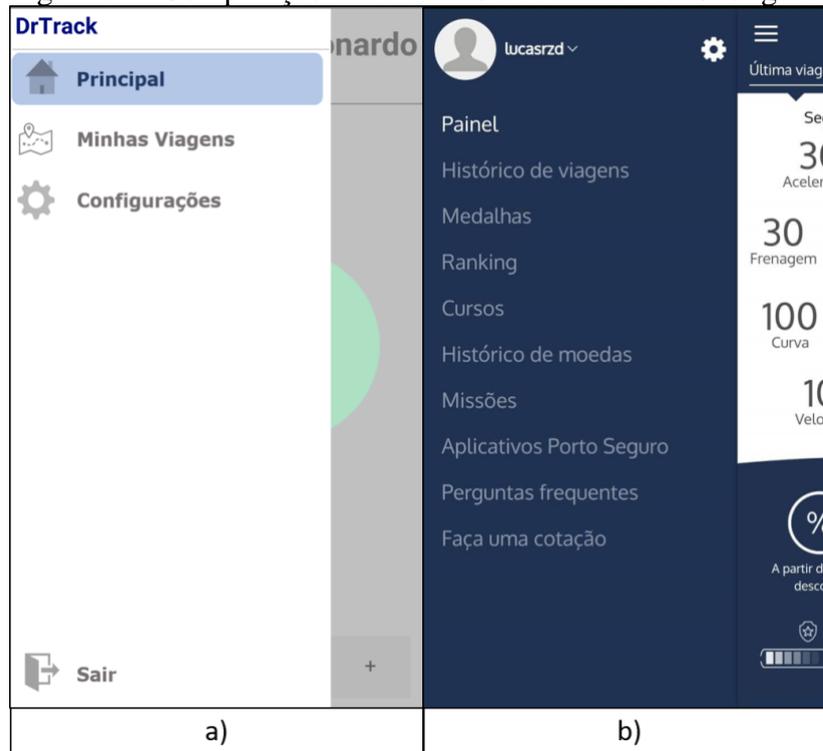
Características	Schlag (2017)	Silva (2017)	Trânsito+gentil (2018)	<b>DrTrack</b>
gerenciamento para vários veículos	Não	Não	Não	Não
dispositivo para captação dos dados do veículo	Sim	Sim	Não	Não
relatório de informações coletadas do veículo	Sim	Sim	Sim	Sim
monitoramento de perfil do motorista	Sim	Não	Sim	Sim
utilização do GPS para localização do veículo	Sim	Sim	Sim	Sim
cadastro de veículos	Não	Não	Não	Não
interface da aplicação	Web	Android	Android, IOS	Android
informação de condução em tempo real para o condutor	Não	Não	Não	Sim

Fonte: elaborado pelo autor.

Os trabalhos de Schlag (2017), Trânsito+gentil (2018) e o DrTrack fazem o monitoramento do perfil do motorista. Schlag (2017), Silva (2017), Trânsito+gentil (2018) e o DrTrack utilizam o sensor de GPS para obter a localização do veículo. Somente o DrTrack apresenta informações para o condutor em tempo real sobre a sua condução. A comparação com os correlatos apontou a importância de um processo para cadastro e controle de motoristas através de uma ferramenta web.

Os testes das funcionalidades da ferramenta foram feitos em comparação ao aplicativo Trânsito+gentil, correlato deste trabalho. Com as funcionalidades semelhantes e por ser um aplicativo que já está inserido no mercado, o Trânsito+gentil se mostrou uma ótima ferramenta para fazer a validação do DrTrack. Nos testes foram utilizados dois celulares com Android na versão 9. O primeiro celular possuía instalado o DrTrack, o segundo possuía o aplicativo Trânsito+gentil. A Figura 21 mostra a comparação entre os menus das duas ferramentas.

Figura 21 - Comparação entre menus DrTrack e Trânsito+gentil

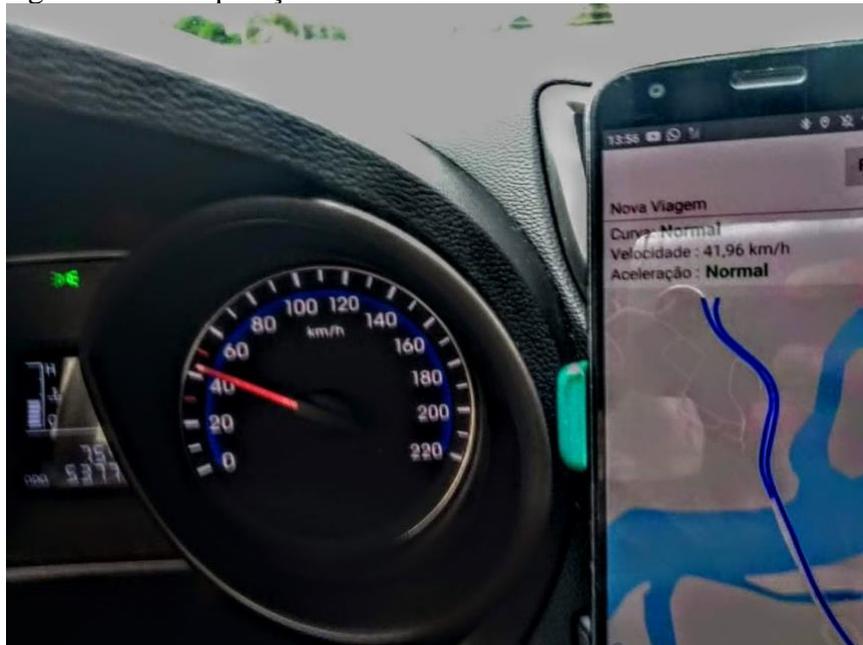


Fonte: elaborado pelo autor.

Como é possível visualizar na Figura 21b, o aplicativo Trânsito+gentil, por propor um jogo como forma de monitoramento, possui a função de medalhas, ranking, missões e moedas, que acabam não oferecendo vantagem para a proposta do DrTrack por ser uma ferramenta para empresas. As duas ferramentas apresentam a opção de visualizar o histórico de viagens. O aplicativo Trânsito+gentil oferece ainda a opção de cursos de pilotagem que pode ser uma opção interessante para agregar ao DrTrack para melhorar a condução dos usuários.

Para validar a obtenção da velocidade atual do veículo, foram analisados os valores para velocidade dados pelo DrTrack com o velocímetro do carro. Os resultados foram muito semelhantes com o valor apontado pelo velocímetro, apresentando variações pequenas devido a algumas imprecisões do sensor de GPS do dispositivo. A Figura 22 mostra uma foto tirada para comparação da velocidade do velocímetro com a velocidade no DrTrack.

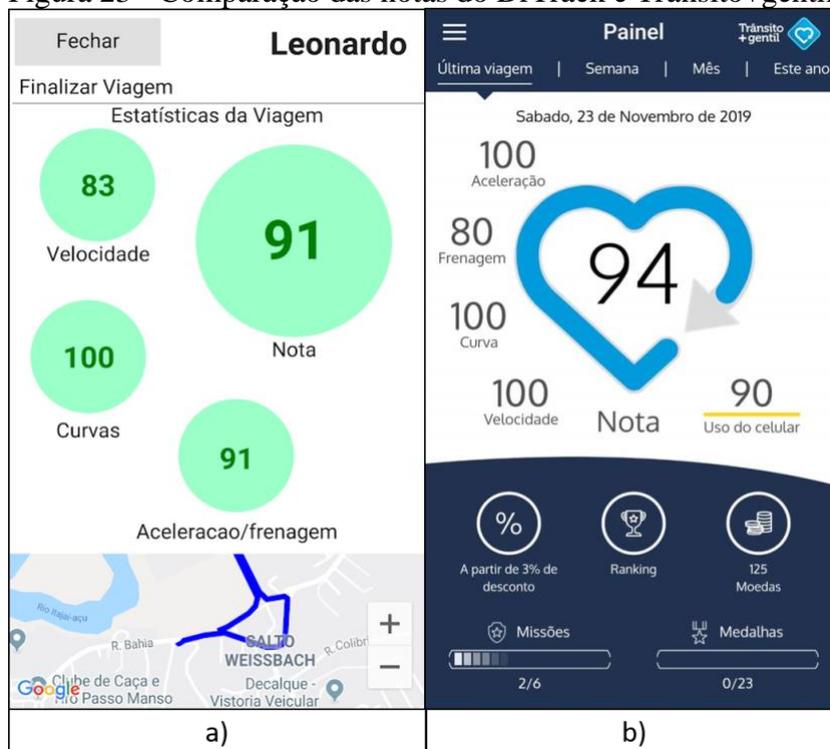
Figura 22 - Comparação entre velocidade no velocímetro e DrTrack



Fonte: elaborado pelo autor.

A análise do perfil do condutor foi também validada utilizando o aplicativo Trânsito+gentil. Foi comparada a nota de cada uma das ferramentas para a mesma viagem feita, e o mapa com o caminho gerado. A Figura 23 apresenta as notas das duas ferramentas para a mesma viagem.

Figura 23 - Comparação das notas do DrTrack e Trânsito+gentil.



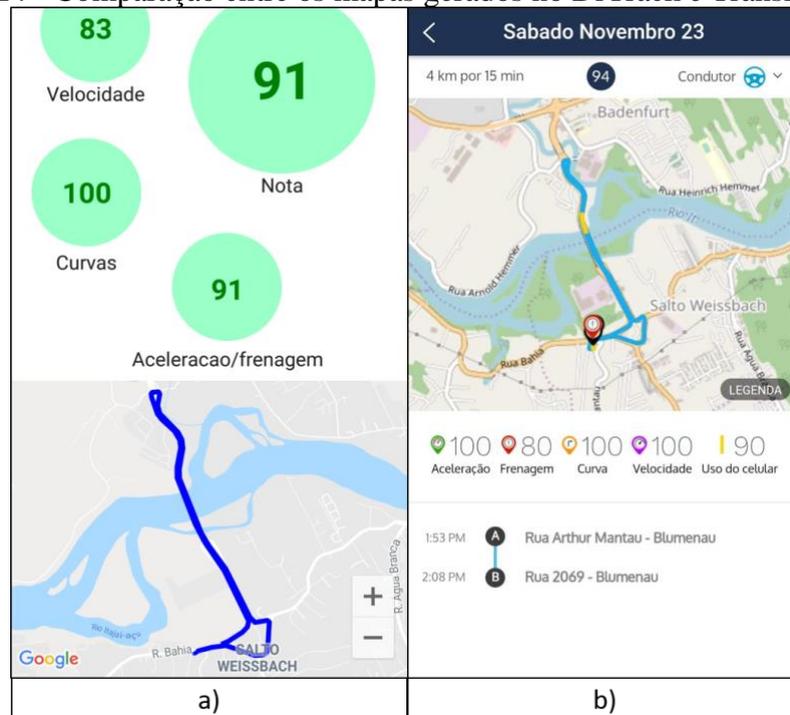
Fonte: elaborado pelo autor.

A comparação entre a análise feita entre as duas ferramentas apresentam resultados bem semelhantes. O valor da nota de curvas foi igual, sendo a pontuação máxima. Os valores de frenagem e aceleração são tratados de forma diferente entre as duas ferramentas. No DrTrack estas características foram agrupadas em uma só nota, enquanto o Trânsito+gentil possui notas individuais para as duas características. Fazendo a média entre as notas de aceleração e frenagem do Trânsito+gentil se obtém o valor 90 muito semelhante a nota 91 apontada pelo DrTrack.

A análise velocidade apresentou a maior variação entre as duas ferramentas. Enquanto o Trânsito+gentil avaliou como 100 a nota para a velocidade, o DrTrack avaliou como 83. Essa variação está ligada ao fato de ter se desenvolvido um *web service* para obter as velocidades máximas. O método utilizado foi por aproximação, fazendo com que a velocidade máxima retornada fosse a da localização mais próxima, podendo ser de uma rua lateral ou transversal da via que o condutor está trafegando. Para resolver esse problema poderia ser alterado o *web service* para identificar as velocidades máximas pelo trecho da via.

O mapa gerado ao final da viagem com o trajeto percorrido também foi comparado entre os dois aplicativos. O resultado foi o mesmo entre os dois. Mostrando o caminho exato percorrido na viagem. A Figura 24 apresenta a comparação entre os mapas das duas ferramentas.

Figura 24 - Comparação entre os mapas gerados no DrTrack e Trânsito+gentil



Fonte: elaborado pelo autor.

O mapa gerado pelas duas ferramentas possui o mesmo desenho, porém o Trânsito+gentil aponta os pontos onde a condução foi mais perigosa. O Trânsito+gentil mostra ainda a rota percorrida por extenso, mostrando as ruas em que o condutor passou. Esses recursos podem ser importantes para o usuário analisar onde foram os principais problemas da viagem.

O aplicativo apresentou lentidão para finalizar algumas viagens. Como a requisição para a obtenção da velocidade máxima da localização é feita via *web service*, caso o usuário esteja sem internet no momento da finalização da viagem o aplicativo demora a responder, visto que ele tenta fazer várias conexões para obter a velocidade. Caso a viagem seja muito longa, a análise pode demorar devido grande número de requisições feitas. Um dos contornos aos problemas seria a verificação de conexão com a internet antes de tentar fazer a conexão. Outra melhoria seria a exportação dos dados para a nuvem, e fazer a análise da viagem direto no servidor, para diminuir o processamento direto no dispositivo.

Em relação ao desenvolvimento, a ferramenta utilizada para desenvolver o aplicativo, Delphi 10.3, se mostrou muito eficiente e produtiva. O desenho de telas é facilitado, podendo desenhar arrastando e fixando os componentes da tela. Para desenvolvedores que conhecem a linguagem Delphi, a curva de aprendizado é muito mais curta, visto que é possível fazer um aplicativo com muita facilidade usando somente a linguagem já conhecida. A integração com os sensores do dispositivo é muito fácil e intuitiva, os componentes do Delphi 10.3 facilitam a parte de comunicação e controle dos sensores. A depuração do código pode ser feita com o aplicativo rodando diretamente no dispositivo e com os dados dos sensores.

O uso do AWS Lambda para a criação do *web service* se mostrou uma boa escolha. O serviço é escalável, devido a montagem da plataforma. O desenvolvimento e configuração foi muito simples, tendo em poucos minutos o serviço rodando. As respostas as requisições feitas são rápidas, com exceção da primeira requisição, pois o gerenciamento do AWS Lambda desativa o código após um tempo sem ser utilizado.

## 4 CONCLUSÕES

Este trabalho apresenta o desenvolvimento e testes de uma ferramenta para o monitoramento da agressividade de condução de motoristas através de um dispositivo Android. Com base nos objetivos traçados para a ferramenta, é possível afirmar que os objetivos foram atingidos, já que a ferramenta grava os dados da rota feita pelo motorista, obtém os dados referentes a velocidade, aceleração, frenagem e curvas e faz a análise desses dados para gerar o perfil do motorista.

Com base nas comparações feitas com uma ferramenta que já está no mercado, o Trânsito+gentil, o aplicativo mostrou que o método de análise referente a velocidade, aceleração e curvas foram corretos, apresentado grande semelhança. Apesar da eficácia dos métodos, a avaliação da velocidade pode ser melhorada alterando o método de obtenção da velocidade máxima das vias. A comparação apontou também a falta de mostrar no mapa os pontos onde houve maior agressividade do condutor. A opção para incluir cursos para o condutor também poderia ser incluída.

A análise dos dados coletados pode ser melhorada com o processamento dos dados na nuvem, removendo essa função do aplicativo, com isso aumentaria a velocidade de resposta do aplicativo, possibilitando ainda o desenvolvimento de uma plataforma web para monitorar os condutores por parte da empresa.

As ferramentas utilizadas para o desenvolvimento do trabalho se mostraram muito úteis. Tanto o Delphi 10.3, utilizado para desenvolvimento do aplicativo, quanto o AWS Lambda utilizado para o desenvolvimento do *web service*, facilitaram o desenvolvimento da ferramenta, aumentando a agilidade de desenvolvimento, a escalabilidade e tornando a curva de aprendizado dessas ferramentas muito curta, devido a facilidade de uso.

É possível concluir que este trabalho apresenta uma ferramenta que faz o monitoramento e análise da condução de motoristas. As análises dos aspectos da condução (aceleração, velocidade e curvas) funcionaram corretamente com base na comparação com outra ferramenta. Com base nos dados fornecidos pela ferramenta, condutores podem controlar melhor sua forma de condução, podendo tomar medidas para preservação de seus veículos. Além disso esse trabalho apresenta contribuição para a sociedade visto que o monitoramento da condução é possível traçar medidas para conscientização e aprendizagem de uma forma mais segura de pilotagem, possibilitando a diminuição de acidentes e mortes no trânsito, bem como gastos com seguro e manutenção de veículos.

#### 4.1 EXTENSÕES

Como sugestões para possíveis extensões ao trabalho desenvolvido citam-se:

- a) criação de plataforma web para controle das empresas dos seus motoristas;
- b) fazer a análise do perfil em um serviço na nuvem para melhorar a performance da análise;
- c) melhorar a busca de velocidades máximas das vias para aumentar a precisão da análise da velocidade;
- d) adicionar função para inserir cursos para melhorar a condução dos motoristas;
- e) mostrar na rota no mapa os pontos de maior agressividade para ajudar a identificar os pontos de risco da rota;
- f) criar lista com ruas da rota feita pelo condutor para identificar mais facilmente a rota efetuada.

## REFERÊNCIAS

- AMAZON. **AWS Lambda**. Seattle, 2019. Disponível em: <<https://aws.amazon.com/pt/lambda/>>. Acesso em: 14 nov. 2019.
- ANFAVEA. Associação Nacional dos FABricantes de VEículos Automotores. **Anfavea apresenta estudo sobre o mercado brasileiro de autoveículos em 2034**. São Paulo: Anfavea, 2014. Disponível em: <[http://www.anfavea.com.br/docs/06%2011%2014\\_PressRelease\\_Estudo\\_2034.pdf](http://www.anfavea.com.br/docs/06%2011%2014_PressRelease_Estudo_2034.pdf)>. Acesso em: 23 de abril de 2018.
- BACCHIERI, Giancarlo; BARROS, Aluísio J. D.. Acidentes de trânsito no Brasil de 1998 a 2010: muitas mudanças e poucos resultados. **Revista Saúde Pública**, Pelotas, v. 45, n. 5, p.949-963, 04 ago. 2011.
- BOHRER, Fernando José. **Serviço de geolocalização para plataforma android**. 2011. 60 f. TCC (Graduação) - Curso de Engenharia da Computação, Centro Universitário Univates, Lajeado, 2014. Disponível em: <<http://www.univates.br/bdu/handle/10737/519>> . Acesso em: 22 de abril de 2018.
- BRAGA, Felipe Augusto; SILVA, Marcos Alberto Lopes. Implementação de Serviços em Nuvem Baseado no Conceito de Serverless. **E-Rac**, Uberlândia, ano 2018, v. 8, n. 1, p. 1. Disponível em: <<http://www.computacao.unetri.edu.br/erac/index.php/erac/article/view/1299/908>>. Acesso em: 2 dez. 2019.
- BRASIL. Lei nº 9.503/97, de 23 de setembro de 1997. Dispositivos Constitucionais Pertinentes Lei no 9.503/97 (Código de Trânsito) Normas Correlatas. **Código de Trânsito Brasileiro**, [S. l.], 23 set. 1997.
- CESANI, Eric; DRANKA, Lucas. Diretrizes para desenvolvimento de Aplicativo GPS em dispositivos móveis para ciclistas de Curitiba. **InfoDesign**, São Paulo, v. 9, n. 2, p. 123-138. 2012.
- DINIZ, Victor. **Hardware: entenda o Giroscópio**. 2010. Disponível em: <<http://ipodschool.com/hardware-entenda-o-giroscopio/>>. Acesso em: 23 de abril de 2018.
- GARRETT, Felipe. **Saiba o que é GPS e como funciona**. 2011. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2011/12/como-funciona-o-gps.html>>. Acesso em: 22 de abril de 2018.
- GLOBO**. 90% dos acidentes de trânsito ocorrem por imprudência, diz GM. Globo, Itapetininga, 09 abr. 2013. Disponível em: <<http://g1.globo.com/sao-paulo/itapetininga-regiao/noticia/2013/04/90-dos-acidentes-de-transito-ocorrem-por-imprudencia-diz-gm.html>>. Acesso em: 23 de abril de 2018.
- GOOGLE MAPS APIS. **Google Maps para todas as plataformas**. 2018. Disponível em: <<https://developers.google.com/maps/?hl=pt-br>>. Acesso em: 23 de abril de 2018.
- KLEMS, Markus (Ed.). **AWS Lambda Quick Start Guide: Learn how to build and deploy serverless applications on AWS**. Birmingham: Packt Publishing Ltd, 2018. 184 p.
- LABORATÓRIO IMOBILIS. **Acelerômetros: Uso Em Celulares E Detecção De Velocidade**. Parte 1. 2013. Disponível em: <<http://www.decom.ufop.br/imobilis/acelerometros-uso-em-celulares-e-deteccao-de-velocidade-parte-1/>>. Acesso em: 23 de abril de 2018.

- LONDERO, Fabrício Tonetto et al. Opção de marketing para sistemas Android por meio de geolocalização. **Disciplinarum Scientia| Naturais e Tecnológicas**, v. 16, n. 1, p. 109-130, 2015. Disponível em: <<http://periodicos.unifra.br/index.php/disciplinarumNT/article/view/1365>>. Acesso em: 22 de abril de 2018.
- MARTINS, Elaine. **Como funciona o GPS?**. Tecmundo, 2009. Disponível em: <<https://www.tecmundo.com.br/gps/2562-como-funciona-o-gps-.htm>>. Acesso em: 22 de abril de 2018.
- MOREIRA, David. **Controle de velocidade: Acelerar muito pode causar problemas?**. Belo Horizonte: Decla Track, 12 jun. 2017. Disponível em: <<https://declatrack.com.br/2017/06/12/control-de-velocidade-acelerar-pode-causar-problemas/>>. Acesso em: 2 dez. 2019.
- NUNES, Samuel. Excesso de velocidade é a principal infração cometida no trânsito em Santa Catarina. **Diário Catarinense**, Florianópolis, p. 1, 17 abr. 2019. Disponível em: <<https://www.nsctotal.com.br/noticias/excesso-de-velocidade-e-a-principal-infracao-cometida-no-transito-em-santa-catarina>>. Acesso em: 2 dez. 2019.
- PAULA, Fábio Oliveira de. **Sensores IMU – Uma Abordagem Completa – Parte 1**. Ouro Preto: Laboratório Imobilis, 22 fev. 2015. Disponível em: <<http://www.decom.ufop.br/imobilis/sensores-imu-uma-abordagem-completa-parte-1/>>. Acesso em: 2 dez. 2019.
- RIBEIRO JÚNIOR, José Geraldo et al. Sistema para monitoramento descentralizado de trânsito baseado em redes veiculares infraestruturadas. In: 31º SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 10., 2013, Brasília. **Anais eletrônicos...** Brasília: Royal Tulip Brasilia Alvorada, 2010. Disponível em: <<https://www.gta.ufrj.br/ftp/gta/TechReports/JCC13.pdf>>. Acesso em 29 de novembro de 2019.
- ROTA EXATA. **Infográfico: 3 passos para gestão dos veículos**. [S. l.], 2015. Disponível em: <<https://www.rotaexata.com.br/blog/3-passos-para-gestao-dos-veiculos>>. Acesso em: 23 de abril de 2018.
- SCHLAG, Fredy. **Monitoramento da agressividade na direção de Caminhões através de acelerômetro e gps**. 2017. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- SILVA, Maicon Machado Gerardi da. **Aplicação para monitoramento veicular em tempo real**. 2017. 109 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- STIGLER, Maddie. **Beginning Serverless Computing: Developing with Amazon Web Services, Microsoft Azure, and Google Cloud**. New York: Apress, 2017. 216 p.
- TEIXEIRA, Mariane Mendes. **Operações com vetores**. 2018. Disponível em: <<https://brasilescola.uol.com.br/fisica/operacoes-com-vetores.htm>>. Acesso em 29 de novembro de 2019.
- TORRES, Henrique. **Sensores Inerciais - Parte 1**. 2014. Disponível em: <<https://www.embarcados.com.br/sensores-inerciais-parte-1/>>. Acesso em: 23 de abril de 2018.

TRÂNSITO+GENTIL. **Trânsito+gentil**, 2018. Disponível em:  
<<https://www.transitomaisgentil.com.br/>. Acesso em: 3 dez. 2019.>

WASELFISZ, Julio Jacobo. **Mapa da violência 2012**: caderno complementar 2: Acidentes de Trânsito. São Paulo: Instituto Sangari, 2012. Disponível em:  
<[http://mapadaviolencia.org.br/pdf2012/mapa2012\\_transito.pdf](http://mapadaviolencia.org.br/pdf2012/mapa2012_transito.pdf)>. Acesso em: 23 de abril de 2018.

WORLD HEALTH ORGANIZATION. **The top 10 causes of death**. 2014. Disponível em:  
<<http://www.who.int/mediacentre/factsheets/fs310/en/>>. Acesso em: 23 de abril de 2018.

## APÊNDICE A – Dicionário de Dados

Este Apêndice apresenta a descrição do dicionário de dados da ferramenta DrTrack e as entidades do banco de dados, apresentadas na seção 3.2.3. Para os campos de texto foi utilizado o tipo `varchar`, para os campos numéricos `int` e `number`, para os campos de data e hora foi usado o tipo `datetime`. O Quadro 17 apresenta o dicionário de dados da tabela `Acelerometro`.

Quadro 17 - Tabela `Acelerometro`

TABELA: <code>Acelerometro</code>			
ATRIBUTO	TIPO/TAMANHO	OBRIGATORIEDADE	OBSERVAÇÕES
<code>id</code>	<code>int(10)</code>	<code>sim</code>	<code>chavePrimária</code>
<code>viagem_ID</code>	<code>int(10)</code>	<code>sim</code>	<code>Chave estrangeira</code>
<code>valEixoX</code>	<code>number(15,6)</code>	<code>não</code>	
<code>valEixoY</code>	<code>number(15,6)</code>	<code>não</code>	
<code>valEixoZ</code>	<code>number(15,6)</code>	<code>não</code>	

Fonte: elaborado pelo autor.

O Quadro 18 apresenta o dicionário de dados da tabela `Configuracao`.

Quadro 18 - Tabela `Configuracao`

TABELA: <code>Configuracao</code>			
ATRIBUTO	TIPO/TAMANHO	OBRIGATORIEDADE	OBSERVAÇÕES
<code>id</code>	<code>int(10)</code>	<code>sim</code>	<code>chave primária</code>
<code>id_UsuarioConectado</code>	<code>int(10)</code>	<code>não</code>	<code>chave estrangeira</code>
<code>valPadraoAceleracao</code>	<code>number(7,5)</code>	<code>não</code>	
<code>valPadraoCurva</code>	<code>number(7,5)</code>	<code>não</code>	

Fonte: elaborado pelo autor.

O Quadro 19 apresenta o dicionário de dados da tabela `Localizacao`.

Quadro 19 - Tabela `Localizacao`

TABELA: <code>Localizacao</code>			
ATRIBUTO	TIPO/TAMANHO	OBRIGATORIEDADE	OBSERVAÇÕES
<code>id</code>	<code>int(10)</code>	<code>sim</code>	<code>chave primária</code>
<code>viagem_ID</code>	<code>int(10)</code>	<code>sim</code>	<code>chave estrangeira</code>
<code>dataLocalizacao</code>	<code>dateTime</code>	<code>não</code>	
<code>valLatitude</code>	<code>number(10,6)</code>	<code>não</code>	
<code>valLongitude</code>	<code>number(10,6)</code>	<code>não</code>	

Fonte: elaborado pelo autor

O Quadro 20 apresenta o dicionário de dados da tabela `Usuario`.

Quadro 20 - Tabela Usuario

TABELA: Usuario			
ATRIBUTO	TIPO/TAMANHO	OBRIGATORIEDADE	OBSERVAÇÕES
id	int(10)	sim	Chave primária
nome	varchar(100)	não	
valNotaGeral	int(10)	não	
valNotaAceleracao	int(10)	não	
valNotaFrenagem	int(10)	não	
valNotaVelocidade	int(10)	não	
valNotaCurvas	int(10)	não	
senha	varchar(100)	não	
QtdViagens	int(10)	não	
QtdAceleracoes	int(10)	não	
QtdLocalizacoes	int(10)	não	

Fonte: elaborado pelo autor.

O Quadro 21 apresenta o dicionário de dados da tabela viagem.

Quadro 21 - Tabela Viagem

TABELA: Viagem			
ATRIBUTO	TIPO/TAMANHO	OBRIGATORIEDADE	OBSERVAÇÕES
id	int(10)	sim	Chave primária
usuario_ID	int(10)	sim	chave estrangeira
dataInicioViagem	datetime	não	
dataFimViagem	dateTime	não	
valNotaGeral	int(10)	não	
valNotaAceleracao	int(10)	não	
valNotaFrenagem	int(10)	não	
valNotaVelocidade	int(10)	não	
valNotaCurvas	int(10)	não	

Fonte: elaborado pelo autor.