



Aplicação para Rastreio de Viaturas

Licenciatura em Engenharia Informática

Bruno Miguel Ferreira Vieira, nº 2191820

Pieter Henrique Huizinga Aparício, nº 2191763

Leiria, julho de 2023

Aplicação para Rastreio de Viaturas

Licenciatura em Engenharia Informática

Bruno Miguel Ferreira Vieira, nº 2191820

Pieter Henrique Huizinga Aparício, nº 2191763

Trabalho de Projeto da unidade curricular de Projeto Informático realizado sob a orientação do Professor Doutor Sílvio Priem Mendes, da Professora Doutora Anabela Moreira Bernardino e do Professor Doutor Paulo Jorge Gonçalves Loureiro.

Leiria, julho de 2023

Agradecimentos

Dada a conclusão do projeto, consideramos de importância salientar os nossos agradecimentos a todas as pessoas e entidades que contribuíram para que este projeto tivesse decorrido com sucesso.

Começamos por agradecer os docentes orientadores, Sílvio Priem Mendes, Anabela Moreira Bernardino e Paulo Jorge Gonçalves Loureiro. Agradecemos a sua disponibilidade, apoio e acompanhamento do projeto, tendo organizado reuniões semanais, para a análise do estado do desenvolvimento, com o lançamento de novos objetivos e desafios.

Agradecemos também a todos os docentes que ao longo do decorrer do curso contribuíram de forma indireta para este projeto, com conhecimento partilhado/transmitido durante o leccionamento de outras unidades curriculares, o que por si teve um impacto positivo neste projeto.

Deixamos também os nossos agradecimentos aos colegas de curso e amigos, que deram o seu contributo de variadas formas, desde opiniões construtivas, sugestões e testando o projeto nos vários componentes.

Gostaríamos também de fazer um agradecimento especial pela contribuição do estudante André Alberto Paulo Duivenvoorden, que ajudou a colocar o projeto em modo produtivo e alojado no servidor.

Por fim, mas de igual importância, deixamos os nossos agradecimentos à instituição de ensino, que nos acolheu durante quatro anos, que nos ofereceu uma excelente qualidade de ensino, com excelentes condições e cujos docentes contribuíram para o nosso desenvolvimento profissional e pessoal.

Resumo

O presente relatório tem como objetivo apresentar e descrever o projeto desenvolvido no âmbito da unidade curricular de Projeto Informático, que decorreu ao longo do segundo semestre do terceiro ano do curso de Licenciatura em Engenharia Informática, do Instituto Politécnico de Leiria, no ano letivo 2022/2023.

O projeto desenvolvido tem como título “Aplicação para Rastreio de Viaturas” e teve como objetivo principal, o desenvolvimento de uma aplicação móvel para permitir o rastreamento de viaturas com base nos dados provenientes dos sensores (giroscópio, acelerómetro e GPS) embutidos num dispositivo móvel e a submissão destes mesmos dados via uma API *RESTful* para um servidor. No contexto deste projeto, estes dados capturados pelos sensores pretendem ser utilizados para a análise de acidentes rodoviários, de forma a dar a entender o comportamento das viaturas envolvidas, isto é, conseguir detetar por exemplo uma travagem brusca ou leve, uma guinada para qualquer direção ou porventura uma aceleração brusca, entre outros.

O projeto é composto por três grandes componentes. O primeiro componente é o desenvolvimento de duas aplicações móveis *Android* semelhantes para públicos-alvo diferentes que têm como função principal, a captura de informação dos sensores numa viagem. O segundo componente é a realização de testes, isto é, com o auxílio de duas viaturas foram simulados um conjunto de comportamentos com o intuito de dar a entender como os sensores reagem face a estas situações e condições. Estes testes incluem, por exemplo, comportamentos agressivos ou menos agressivos relacionados com travagens, acelerações, guinadas, entre outros. A terceira componente é o desenvolvimento de uma plataforma *web*, cujo objetivo passa por disponibilizar uma forma de gerir e analisar os dados provenientes das aplicações móveis, ou seja, os dados dos sensores durante as viagens efetuadas.

Com a conclusão do projeto, foi possível entender o valor destes dados provenientes dos sensores embutidos nos dispositivos móveis, uma vez que conseguem representar variados comportamentos durante a condução de uma viatura e podem ser valiosos para complementar uma análise de um acidente rodoviário real.

Palavras-chave: Android, Desenvolvimento *web*, Sensores, Giroscópio, GPS, Acelerómetro

Abstract

This current report aims to present and describe the development of the project within the scope of the curricular unit of “*Projeto Informático*”, which took place during the second semester of the third year of the Degree in Computer Engineering, at the Polytechnic Institute of Leiria, in the academic year 2022/2023.

The project developed has the title “Application for Vehicle Tracking” (“*Aplicação para Rastreio de Viaturas*”) and its main objective, is the development of an application, that allows the tracking of vehicles, based on information gathered by sensors, which are embedded in mobile devices, and the subsequent submission of this data to a server via a RESTful API. Due to the context of this project, the captured data by the sensors can be used for the analysis of road accidents involving motor vehicles. It is mainly helpful to understand the vehicles’ behaviour and how they were driven. For instance, being able to detect behaviors such as sudden braking, sudden acceleration, or a swerve, among others.

This project consists of three major components. The first component is the development of two similar Android mobile applications, but for different target audiences, with the main functionality as to gather data from the sensors during a trip. The second component in which tests were carried out, simulating certain driving behaviors on public roads (never putting safety at risk), by using two vehicles, with the aim of understanding how such behaviors and the surrounding conditions could impact and shape the data gathered by the sensors built into a mobile device. These tests include, but not only, aggressive, or non-aggressive behaviours related to, braking, accelerating, and swerving, among others. A third component is the development of a web platform, which main purpose is to provide a way to manage and analyse all the data received from the mobile applications, that is, all data gathered from the sensors built into the mobile devices during a trip.

Having completed this project, it is now possible to understand the value of the data gathered from these built-into mobile device sensors, since, with this data, it is possible to distinguish different driving behaviors and it could potentially be very valuable for the analysis of real road accidents involving motor vehicles.

Keywords: Android, Web development, sensors, gyroscope, accelerometer, GPS

Índice

| | |
|--|-----|
| Agradecimentos | ii |
| Resumo | iii |
| Abstract | iv |
| Lista de Figuras | vii |
| Lista de tabelas | ix |
| Lista de siglas e acrónimos..... | x |
| 1. Introdução..... | 1 |
| 1.1. Enquadramento do tema..... | 1 |
| 1.2. Objetivos..... | 2 |
| 1.3. Fases de desenvolvimento do projeto | 2 |
| 1.4. Estrutura do documento | 3 |
| 2. Metodologia..... | 5 |
| 2.1. <i>Scrum</i> | 5 |
| 2.2. Reuniões..... | 6 |
| 2.3. <i>Jira</i> | 7 |
| 2.4. <i>Bitbucket</i> | 8 |
| 3. Ferramentas e tecnologias utilizadas | 9 |
| 3.1. HTML..... | 9 |
| 3.2. CSS..... | 9 |
| 3.3. <i>JavaScript</i> | 9 |
| 3.4. Java | 9 |
| 3.5. <i>Laravel/ PHP</i> | 10 |
| 3.6. <i>Vue.js</i> | 10 |
| 3.7. <i>Core UI</i> | 11 |
| 3.8. <i>MySql</i> | 11 |
| 3.9. <i>ObjectBox</i> | 11 |
| 3.10. <i>Volley</i> | 12 |
| 3.11. <i>Axios</i> | 12 |
| 3.12. <i>Google Maps API</i> | 12 |
| 3.13. <i>Docker</i> | 13 |
| 3.14. <i>Sail</i> | 13 |
| 3.15. <i>Postman</i> | 13 |
| 3.16. <i>Bitbucket</i> | 13 |
| 3.17. <i>Jira</i> | 14 |
| 3.18. <i>Power BI</i> | 14 |
| 3.19. <i>Inkscape</i> | 14 |
| 3.20. <i>Android Studio</i> | 15 |
| 3.21. <i>Visual Studio Code</i> | 15 |

| | | |
|------|---|----|
| 4. | Análise | 16 |
| 4.1. | Análise concorrencial | 16 |
| 4.2. | Modelo de dados | 20 |
| 4.3. | Análise de <i>datasets</i> | 23 |
| 4.4. | Análise de projetos anteriores | 26 |
| 4.5. | Análise de sensores dos dispositivos móveis e da API..... | 27 |
| 5. | Arquitetura do sistema..... | 31 |
| 6. | Desenvolvimento..... | 33 |
| 6.1. | Aplicação móvel <i>Android</i> | 33 |
| 6.2. | Aplicação <i>Web</i> | 45 |
| 7. | Testes | 50 |
| 7.1. | Estratégia..... | 50 |
| 7.2. | Objetivos..... | 50 |
| 7.3. | Considerações | 51 |
| 7.4. | Tipos de testes | 52 |
| 7.5. | Descrição dos testes | 54 |
| 8. | Instalação/configuração | 73 |
| 8.1. | Plataforma web | 73 |
| 8.2. | Aplicação móvel <i>Android</i> | 76 |
| 9. | Conclusões e trabalho futuro..... | 78 |
| 9.1. | Conclusões | 78 |
| 9.2. | Trabalho futuro | 79 |
| 10. | Bibliografia | 81 |

Lista de Figuras

| | |
|--|----|
| Figura 1 - Ecrã de gravação para a aplicação "My Track" | 17 |
| Figura 2 - Ecrã de <i>record</i> para a aplicação "Strava" | 18 |
| Figura 3 - Ecrãs de demonstração da aplicação "Sensor Logger" | 19 |
| Figura 4 - Ecrã de recolha de dados da aplicação "Sensor Data" | 20 |
| Figura 5 - Diagrama Entidade-Relacionamento para a aplicação <i>Android cliente</i> | 21 |
| Figura 6 - Diagrama Entidade-Relacionamento para a aplicação <i>Android admin/testes</i> | 22 |
| Figura 7 - Diagrama Entidade-Relacionamento para a aplicação <i>web</i> | 23 |
| Figura 8 - Estrutura JSON de um <i>dataset</i> | 24 |
| Figura 9 - Estrutura de um <i>dataset</i> em CSV | 25 |
| Figura 10 - Representação dos eixos de um sensor de aceleração (retirado de [29])..... | 28 |
| Figura 11 - Representação dos eixos de um sensor de giroscópio (retirado de [30])..... | 28 |
| Figura 12 - Arquitetura do sistema | 31 |
| Figura 13 - Ecrã de registo de dispositivo | 34 |
| Figura 14 - Ecrã de iniciar captura | 34 |
| Figura 15 - Ecrã que permite parar a captura..... | 35 |
| Figura 16 - Ecrã de detalhes da aplicação..... | 35 |
| Figura 17 - Mensagem de sincronização automática | 36 |
| Figura 18 - Ecrã de registo de dispositivo | 39 |
| Figura 19 - Ecrãs que permitem visualizar os dados dos sensores e localização | 40 |
| Figura 20 - Ecrã de criação da <i>label</i> do <i>dataset</i> | 41 |
| Figura 21 - Ecrã para a criação de uma nova palavra-chave | 41 |
| Figura 22 - Ecrã que permite eliminar uma <i>label</i> | 42 |
| Figura 23 - Opção de <i>re-record</i> | 42 |
| Figura 24 - Opção de sincronização | 43 |
| Figura 25 - Página de definições | 43 |
| Figura 26 - Listagem de <i>datasets</i> gravados | 44 |
| Figura 27 - Opção para exportação de <i>datasets</i> | 44 |
| Figura 28 - Opção de sinalizar dados de captura | 45 |

| | |
|---|----|
| Figura 29 - Página de listagem de <i>datasets</i> | 46 |
| Figura 30 - Página de detalhes de um dataset | 46 |
| Figura 31 - <i>Dashboard</i> | 48 |
| Figura 32 - Página de <i>about</i> | 49 |
| Figura 33 - Posicionamento do dispositivo móvel durante os testes | 52 |
| Figura 34 - Eixos de leitura do sensor acelerómetro (retirado de [31]) | 55 |
| Figura 35 - Eixos de leitura do sensor giroscópio (retirado de [32]) | 55 |
| Figura 36 - Gráfico do <i>dataset</i> acceleration aggressive 50kmh-28-04-2023-16-53-01 | 56 |
| Figura 37 - Gráfico do dataset brake aggressive 50kmh-26-04-2023-13-06-03..... | 58 |
| Figura 38 - Gráfico do dataset brake aggressive 50kmh-26-04-2023-13-34-51..... | 59 |
| Figura 39 - Percursos e informação dos testes de travagem antes de cruzamento..... | 60 |
| Figura 40 - Gráfico do dataset acceleration aggressive 50kmh turn right 90°-05-05-2023-20-56-08 | 62 |
| Figura 41 - Percursos e informação dos testes de saída do cruzamento com velocidade constante (sem parar) | 63 |
| Figura 42 - Gráfico do dataset constant 40kmh turn right 90°-05-05-2023-20-04-13..... | 64 |
| Figura 43 - Percursos e informação dos testes de rotunda pequena..... | 65 |
| Figura 44 - Gráficos do dataset 30kmh small roundabout 3rd exit-27-05-2023-12-25-28 | 66 |
| Figura 45 - Percursos e informação dos testes de rotunda média | 68 |
| Figura 46 - Gráficos do dataset 40kmh medium roundabout 2nd exit-17-05-2023-19-36-35 | 69 |
| Figura 47 - Percursos e informação dos testes de rotunda grande | 70 |
| Figura 48 - Gráfico do dataset 40kmh big roundabout 4th exit-27-05-2023-11-19-43 | 71 |
| Figura 49 - Contentor "rastreiodeviaturas" a correr..... | 74 |
| Figura 50 - Comando <i>npm run serve</i> executado com sucesso | 75 |
| Figura 51 - Painel login | 76 |
| Figura 52 - Selecionar pasta da aplicação móvel..... | 77 |
| Figura 53 - Executar aplicação no emulador | 77 |

Lista de tabelas

| | |
|--|----|
| Tabela 1 - Tabela representativa das reuniões realizadas | 6 |
| Tabela 2 - Características das viaturas utilizadas | 51 |

Lista de siglas e acrónimos

| | |
|------|---|
| ABS | <i>Antilock Braking System</i> |
| API | <i>Application Programming Interface</i> |
| APK | <i>Android Package Kit</i> |
| CSS | <i>Cascading Style Sheets</i> |
| CSV | <i>Comma-separated values</i> |
| ESTG | <i>Escola Superior de Tecnologia e Gestão</i> |
| GPS | <i>Global Positioning System</i> |
| HTML | <i>HyperText Markup Language</i> |
| IDE | <i>Integrated Development Environment</i> |
| IoT | <i>Internet of Things</i> |
| JSON | <i>JavaScript Object Notation</i> |
| MVC | <i>Model–View–Controller</i> |
| PHP | <i>Hypertext Preprocessor</i> |
| SQL | <i>Structured Query Language</i> |
| UI | <i>User Interface</i> |
| WSL | <i>Windows Subsystem for Linux</i> |
| XML | <i>Extensible Markup Language</i> |

1. Introdução

O presente relatório tem como objetivo principal apresentar e descrever o projeto desenvolvido no âmbito da unidade curricular de Projeto Informático, do curso de Licenciatura em Engenharia Informática, do Instituto Politécnico de Leiria, no ano letivo 2022/2023.

A unidade curricular de Projeto Informático teve a duração íntegra do segundo semestre (cerca de quatro meses e doze dias), tendo iniciado a 2 de março de 2023 com uma reunião de *kick-off* com os docentes orientadores, dando início ao projeto, e terminado a 14 de julho de 2023 com a entrega do projeto realizado.

O projeto foi desenvolvido no tema de “Aplicações para Rastreio de Viaturas”, teve como base dois projetos desenvolvidos por outros estudantes em anos letivos anteriores. Foi realizado sob orientação dos docentes orientadores Sílvio Priem Mendes, Anabela Moreira Bernardino e Paulo Jorge Gonçalves Loureiro.

Teve como objetivo principal, o desenvolvimento de uma aplicação móvel, que permite o rastreamento de viaturas com base nos dados provenientes dos sensores embutidos num dispositivo móvel e a submissão destes mesmos dados via uma API (*Application Programming Interface*) *RESTful*. para um servidor. Sendo que, no contexto deste projeto, esses dados capturados pelos sensores possam ser utilizados para a análise de acidentes rodoviários, de forma a dar a entender o comportamento das viaturas envolvidas, isto é, conseguir detetar por exemplo uma travagem brusca ou leve, uma guinada para qualquer direção ou porventura uma aceleração brusca, entre outros.

Na secção 1.1 é feito um enquadramento ao tema. Na secção 1.2 são descritos os objetivos. Na secção 1.3 são apresentadas as fases de desenvolvimento do projeto. Por fim, na Secção 1.4 é apresentada a estrutura do documento.

1.1. Enquadramento do tema

Os acidentes rodoviários são um perigo nas vias públicas, o óbvio intuito é de reduzir o número destes incidentes. Contudo, por vezes é difícil entender o que levou a um acidente,

quais foram os fatores e por vezes não há testemunhas que possam descrever as condições em que decorreu um acidente.

É desta forma importante fazer um esforço para entender o comportamento, detalhado, de uma viatura antes de o acidente acontecer. É esperado que esta informação possa reduzir a quantidade de acidentes rodoviários ou de meramente entender os sucedidos.

1.2. Objetivos

A unidade curricular de Projeto Informático tem o objetivo de colocar em prática um conjunto de competências adquiridas e desenvolvidas durante o decorrer do curso, assim como também de proporcionar a aquisição de novas competências e incentivar o trabalho de investigação/exploração das atividades propostas. Acaba também por proporcionar uma perspetiva do que é possível alcançar com o conhecimento e competências adquiridas, e até mesmo dar a conhecer um tema novo, real e desafiante.

O objetivo principal deste projeto é determinar qual o tipo de comportamento que os sensores nos dispositivos móveis *Android* apresentam, face a determinados comportamentos de condução de uma viatura. Pretende-se também verificar se é possível destacar padrões ao analisar a variação de velocidade, atitude na condução, entre outras. É por isso interessante entender qual o seu valor, utilidade e capacidade de auxiliar na análise de acidentes rodoviários e eventualmente evitá-los.

1.3. Fases de desenvolvimento do projeto

O projeto é constituído por três componentes principais:

- **Aplicação *Android*:** Engloba o desenvolvimento de duas aplicações móveis *Android* semelhantes, mas para públicos-alvo diferentes.

Uma das aplicações é destinada à realização de testes, tendo à disposição um conjunto de opções e informações sobre a leitura dos sensores do dispositivo móvel, como também as funcionalidades base de gravação de uma viagem e o envio desses dados para um servidor via uma API *RESTful*.

A outra aplicação é destinada ao utilizador final, que apenas permite iniciar a gravação de viagens e realizar o envio automático desses dados para o servidor via uma API *RESTful*.

Ambas as aplicações móveis fazem a gravação de viagens tirando partido dos dados provenientes dos sensores (giroscópio, acelerómetro e GPS (*Global Positioning System*)) embutidos no dispositivo móvel.

- **Fase de testes:** Foram realizados testes, isto é, com o auxílio de duas viaturas foram simulados nas vias públicas (nunca colocando em risco a segurança), um conjunto de comportamentos com o intuito de dar a entender como os sensores reagem face a estas situações e condições.

Estes testes incluem, por exemplo, comportamentos, agressivos ou menos agressivos relacionados com travagens, acelerações, guinadas, entre outros.

- **Plataforma web:** Representa o desenvolvimento de uma plataforma *web*, com uma tecnologia *frontend* baseada na *template CoreUI* (que por si faz uso da tecnologia *Vue.js*) e uma tecnologia *backend Laravel*.

O objetivo desta plataforma passa por disponibilizar uma forma de gerir e analisar os dados provenientes das aplicações móveis, ou seja, os dados dos sensores durante as viagens efetuadas.

1.4. Estrutura do documento

O documento está dividido em nove capítulos principais:

- **1. Introdução:** O capítulo atual, que visa introduzir o projeto, dar a conhecer o tema, objetivos e dar uma abordagem inicial ao projeto realizado;
- **2. Metodologia:** Visa descrever a metodologia aplicada e usada, a forma como foi planeado e organizado o desenvolvimento do projeto;
- **3. Ferramentas e tecnologias utilizadas:** Tem como objetivo apresentar e brevemente descrever as ferramentas e tecnologias utilizadas;
- **4. Análise:** Pretende dar a conhecer as várias análises efetuadas, quer uma análise de componentes desenvolvidas, análise de projetos anteriores e uma análise dos sensores utilizados, entre outros;
- **5. Arquitetura do sistema:** Pretende dar uma visão global do projeto desenvolvido, entender como as várias componentes se interligam e dar a entender as suas dimensões;
- **6. Desenvolvimento:** Tem como objetivo descrever o desenvolvido do projeto desenvolvido para cada uma das componentes;

- **7. Testes:** Pretende dar a conhecer os testes que foram realizados, assim como apresentar a sua análise;
- **8. Instalação/configuração:** Tem como objetivo descrever os passos necessários para a instalação/configuração do projeto desenvolvido;
- **9. Conclusão e trabalho futuro:** Último capítulo que visa apresentar as conclusões alcançadas e direções para trabalho futuro.

2. Metodologia

Neste capítulo pretende-se descrever a metodologia utilizada no decorrer do desenvolvimento do projeto. Na Secção 2.1 é descrita o tipo de metodologia utilizada para gestão do projeto, na Secção 2.2 são descritas as reuniões, na Secção 2.3 a plataforma *Jira* e por fim, na Secção 2.4 a ferramenta de controlo de versões *Bitbucket*.

2.1. *Scrum*

O *Scrum* é uma metodologia ágil de gestão de projetos amplamente utilizada no desenvolvimento de software.

Baseia-se na divisão do trabalho em *sprints*, com duração de uma a quatro semanas, em que a equipa entrega incrementos do produto a desenvolver.

A metodologia pretende fomentar a colaboração, a flexibilidade e a adaptação às mudanças, permitindo que as equipas se ajustem rapidamente para realizar entregas de versões do projeto de forma eficiente e com elevada qualidade [1].

Foram adotados para este projeto os seguintes elementos padrões da metodologia ágil *Scrum*:

- Planeamento da Sprint: Seleção de um conjunto de tarefas a ser implementadas;
- Distribuição das tarefas pelos elementos da equipa: Atribuição de tarefas consoante ao elemento mais apropriado para a realização da mesma;
- Tarefas conjuntas separadas em subtarefas: Tarefas de grande complexidade podem ser atribuídas sobre a forma de subtarefas a vários elementos da equipa;
- Revisão da sprint: As funcionalidades implementadas são verificadas com a equipa de forma a obtêm *feedback*.

A utilização desta metodologia permitiu identificar o estado do projeto ao longo do decorrer do mesmo, permitindo identificar se este estava numa fase avançada ou se tinha tarefas em atraso em relação às atividades que tinham sido definidas durante a fase de planeamento do produto.

2.2. Reuniões

A parte fundamental do planeamento foram as reuniões realizadas como forma de acompanhamento ao projeto, tendo como intervenientes os desenvolvedores (nós, estudantes) e os docentes orientadores.

Estas reuniões foram no geral realizadas com a frequência de uma vez por semana, com o objetivo principal, de analisar o trabalho realizado durante a semana anterior à reunião, assim como de discutir e planear as novas fases a realizar até à seguinte reunião.

Foram úteis para identificar problemas com as implementações e analisar os dados obtidos pelas várias fases das aplicações, assim como para discutir a implementação gráfica dos vários componentes.

Foram realizadas no total 13 reuniões, cujas atividades se encontram abaixo na Tabela 1:

Tabela 1 - Tabela representativa das reuniões realizadas

| <i>Número</i> | <i>Data</i> | <i>Atividades</i> |
|---------------|---------------------|--|
| 1 | 2 de março de 2023 | Planeamento das atividades iniciais. |
| 2 | 13 de março de 2023 | Registar dispositivo, registar <i>dataset</i> , analisar o funcionamento de sensores |
| 3 | 20 de março de 2023 | Filtros e definições da aplicação <i>android</i> , velocidade máxima, exportação de dados |
| 4 | 27 de março de 2023 | Apresentação na aplicação dos mínimos e máximos que os sensores registaram, localização, estruturar projeto <i>web</i> |
| 5 | 17 de abril de 2023 | Corrigir formatação dos ficheiros exportados, realização de testes de travagem |
| 6 | 24 de abril de 2023 | Realização de testes, formatação do ficheiro JSON |
| 7 | 2 de maio de 2023 | Formatar valores de saída dos sensores, testes com curvas 90º |
| 8 | 8 de maio de 2023 | Sinalizar gravação, realização de testes em rotundas, aceleração, desaceleração, testes de atrito |

| | | |
|----|---------------------|---|
| 9 | 15 de maio de 2023 | Envio de <i>dataset</i> , planeamento de atividades posteriores |
| 10 | 29 de maio de 2023 | Filtros aplicação <i>web</i> , modificar formatação dos <i>timestamps android</i> , definições para exportação de <i>datasets</i> |
| 11 | 5 de junho de 2023 | Adicionar <i>bearing</i> , suporte para <i>datasets</i> antigos, botão de play para <i>web</i> |
| 12 | 12 de junho de 2023 | Páginas com <i>about</i> , aplicação de cliente, opções de velocidade na função de <i>play</i> . |
| 13 | 5 de julho de 2023 | Cenário para defesa, relatório |

2.3. *Jira*

Foi realizado um planeamento através da plataforma *Jira*, com recurso a *sprints*, que ajudaram a distribuir as várias tarefas a realizar por cada semana.

Cada *sprint* teve a duração de uma ou duas semanas e foram organizadas da seguinte forma:

- **Sprint 0:**

Analisar *software* fornecido, analisar *software* da aplicação móvel.

- **Sprint 1:**

Registrar dispositivo, visualizar os dados dos sensores.

- **Sprint 2:**

Criar *datasets*, escolher velocidade de captura.

- **Sprint 3:**

Filtrar sensores, visualizar *min/max* dos sensores.

- **Sprint 4:**

Exportar *datasets* em json/csv, *labels* com *autocomplete*.

- **Sprint 5:**

Opção de *re-record*, opção de sinalizar partes da captura.

- **Sprint 6:**

Filtragem de *datasets web*, mostrar detalhes de *dataset*.

- **Sprint 7:**

Apresentar dados de *datasets* no mapa, filtrar *outliers*.

- **Sprint 8:**

Apresentar rota do *dataset*, utilizar o *bearing* para atualizar a rotação do carro.

- **Sprint 9:**

Funcionalidade *play* dos *datasets* selecionados no mapa, opção de importar *datasets* a partir da plataforma *web*.

- **Sprint 10:**

Opção para abrir detalhes de um *dataset* a partir do mapa, opção para descarregar um *dataset*.

A distribuição das várias atividades foi realizada tendo em conta com a familiaridade com que cada elemento tinha com os elementos necessários à realização da atividade, tentando assim que as atividades pudessem ser realizadas da forma mais eficiente possível e tentando sempre manter a realização de atividades em paralelo para que o projeto não ficasse parado à espera de que uma atividade fosse concluída.

2.4. *Bitbucket*

As várias versões do projeto foram registadas com recurso a ferramenta de controlo de versões *Bitbucket*, que nos permitiu fazer uma separação do projeto por aplicações através de repositórios, que por sua vez separavam cada atividade em ramos dentro do respetivo repositório.

O projeto foi separado em três repositórios:

- Aplicação *Web*;
- Aplicação *Android testes/admin*;
- Aplicação *Android cliente*.

Cada elemento tinha as suas tarefas separadas independentemente dos outros elementos permitindo realizar tarefas de código separadamente, facilitando assim a leitura e implementação nas novas tarefas.

Todas as atividades passaram por um processo de *review* e aprovação por parte dos elementos da equipa de desenvolvimento garantindo assim a qualidade do código e da funcionalidade implementada.

3. Ferramentas e tecnologias utilizadas

Este capítulo tem como objetivo apresentar e descrever as ferramentas, tecnologias e bibliotecas principais utilizadas durante o desenvolvimento do projeto.

3.1. HTML

O HTML (*HyperText Markup Language*) é uma linguagem de marcação utilizada no desenvolvimento de páginas *web*.

Tem um papel fundamental na construção de páginas *web*, estruturando a página e trabalhando em conjunto com linguagens de programação como o *JavaScript* e mecanismos que fornecem estilos como o CSS (*Cascading Style Sheets*) para poder entregar um produto final estruturado e interpretável ao utilizador da aplicação *web* [2].

3.2. CSS

Mecanismo que fornece um conjunto de estilos a aplicar a elementos HTML ou, através das *tags* “*<style>*”, que permite apresentar o conteúdo da aplicação *web* de uma forma visual apelativa e fazer a separação entre o formato e o conteúdo, de um documento [3].

3.3. JavaScript

O *JavaScript* é uma linguagem de programação utilizada no desenvolvimento de páginas *web*. É uma das principais tecnologias utilizadas em navegadores *web* e faz parte dos três elementos principais para o desenvolvimento *web* e *front-end* em conjunto com o HTML, CSS e o PHP (*Hypertext Preprocessor*) [4].

Na sua base existem os scripts que tem um papel fundamental na conversão de páginas estáticas para páginas de elementos dinâmicos.

3.4. Java

O *Java* é uma linguagem de programação orientada a objetos geralmente utilizada em aparelhos moveis e IoT (*Internet of Things*).

Utiliza XML (*Extensible Markup Language*) para a criação de páginas dinâmicas para a *web*, e tem como principal vantagem a sua capacidade de adaptação a diversos dispositivos [5] como por exemplo:

- *Tablets*;
- Computadores;
- *SmartWatches*;
- Telemóveis.

3.5. *Laravel/ PHP*

Laravel é uma *framework* de desenvolvimento *web* em PHP, que fornece uma estrutura robusta e abrangente para a construção de aplicações *web* eficientes e escaláveis [6]. Segue o padrão de arquitetura MVC (*Model–View–Controller*) [7], o que ajuda a separar os componentes das aplicações, tornando-os mais organizados e fáceis de manter.

O uso desta *framework* foi obrigatório, dado que a *template* fornecida pelo *Core UI* requere o uso da mesma, como base para a configuração da API e base de dados. Esta *framework* fornece como base várias funcionalidades, como sistemas de autenticação, sistemas de *roles*, migrações e *seeds*.

Na realização da aplicação *web* foi utilizado a *framework* *Laravel* na versão 8.83.27 com PHP na versão 8.0.

3.6. *Vue.js*

O *Vue.js* é uma *framework* de *JavaScript* usado para construir UI's interativas e reativas em aplicações da *web*.

O *Vue.js* permite que os programadores transformem as interfaces em componentes reutilizáveis e fornece uma sintaxe simples e intuitiva para a criação desses componentes. Possibilita rastrear as dependências entre os dados e a UI, permitindo atualizações automáticas e eficientes sempre que os dados são modificados [8].

Esta *framework* faz parte dos componentes base necessários a utilização da *template* do *Core UI*, o que tornou a escolha do *Vue.js* obrigatória. A *framework* permite que os dados dos vários componentes usados na *template*, possam ser atualizados de forma dinâmica quando estes são utilizados [8].

3.7. Core UI

O *Core UI* disponibiliza um conjunto de interfaces, componentes, estilos e *layouts* pré-construídos que servem de base para a criação de UI modernas e atraentes. Apresenta um kit de UI de código aberto, com recurso a componentes *bootstrap* [9].

Esta ferramenta foi proposta pelos docentes orientadores e foi essencial à aplicação *web*, pois permitiu poupar muito tempo na configuração inicial da aplicação, nomeadamente na configuração de utilizadores, sistema de *login/registo*. Apresentou um *layout* base, no qual é possível inserir as funcionalidades da aplicação de forma fácil e fazendo uso dos componentes fornecidos pelo *Core UI*, podendo assim voltar o foco para a funcionalidade, em vez da construção de interfaces/*layouts*.

3.8. MySql

O *MySQL* é um sistema que permite a gestão de base de dados relacional. O *MySQL* permite armazenar, organizar e gerir grandes volumes de dados de forma eficiente. Utiliza a linguagem *SQL* (*Structured Query Language*) para obter e manipular dados armazenados na base de dados [10].

Este sistema foi utilizado como base de dados para a aplicação *web*, pois este vinha configurado por base na *template* do *Core UI* fornecida.

3.9. ObjectBox

O *ObjectBox* é uma biblioteca que disponibiliza uma base de dados orientada a objetos para desenvolvimento de aplicações móveis.

Ao contrário das bases de dados tradicionais, o *ObjectBox* foi projetado especificamente para o armazenamento de objetos [11].

O *ObjectBox* foi escolhido como base de dados para aplicação *Android*, pois apresenta as seguintes vantagens em relação aos seus concorrentes:

- Alta velocidade;
- Baixa sobrecarga de memoria;
- Sincronização eficiente;

- Disponibilização de painel de *admin* que permite a visualização dos dados e a sua estrutura.

3.10. *Volley*

O *Android Volley* é uma biblioteca de rede que facilita a execução de solicitações HTTP e o processamento das respostas. Algumas das principais características do *Android Volley* incluem [12]:

- Análise automática de respostas;
- Tratamento de erros;
- Pedidos em segundo plano;
- Cancelar pedidos;
- Pedidos em simultâneo;
- Sistema de fila de pedidos.

3.11. *Axios*

O *Axios* é um cliente HTTP utilizado através do *node.js* ou do *browser* que permite o envio e receção de pedidos HTTP através de *promises* e fornece as seguintes funcionalidades [13]:

- Intercetar pedidos e respostas;
- Cancelar pedidos;
- Gerir *timeouts*;
- Permite pedidos com body no formato JSON, *Multipart/FormData*, *Url encoded form*;
- Gerir os limites de tráfego.

3.12. *Google Maps API*

A *Google Maps API* é uma API que permite incorporar dados de mapas e informações de localização do *google maps* em projetos [14].

Com a *Google Maps API*, os programadores podem criar aplicações que apresentam mapas interativos, traçam rotas, fornecem direções de condução, estimam o tempo de viagem, oferecem sugestões de localização próximas e muito mais.

3.13. Docker

O *Docker* é uma plataforma de código aberto que permite automatizar o processo de criação/installação e execução de aplicações em *containers* [15].

Os *containers* são ambientes isolados e autossuficientes, onde é possível criar um pacote de aplicação juntamente com as suas dependências e configurá-lo de maneira consistente, independentemente do ambiente em que é executado [15].

3.14. Sail

O *Laravel Sail* é uma ferramenta, que simplifica o uso do *Docker*, para o desenvolvimento de aplicações *Laravel*, fornecendo uma maneira fácil de configurar e gerir ambientes de desenvolvimento locais. Facilita também a execução de aplicações *Laravel* em *containers Docker*, agilizando o processo de configuração do ambiente de desenvolvimento e garantindo uma experiência de desenvolvimento consistente entre diferentes máquinas [16].

3.15. Postman

O *Postman* é uma ferramenta de colaboração e desenvolvimento que permite testar, documentar e depurar APIs de forma eficiente. Permite também criar coleções e ambientes, testes automatizados e gerar documentação da API. O *Postman* foi usado como ferramenta de teste para os componentes da API, em fase de testes e como meio de depuração para os erros encontrados durante a implementação. Foi assim uma ferramenta essencial no suporte à implementação do código para a aplicação *web* [17].

3.16. Bitbucket

A tecnologia *Bitbucket*, foi escolhida como sistema de controle de versões por fornecer uma interface gráfica através da *web* que permite uma utilização mais eficiente das ferramentas disponibilizadas, apresenta formas mais eficientes nos requisitos resolução de conflitos e criação de *pull requests* e disponibiliza ainda a integração com ferramentas de planeamento de projeto [18].

3.17. *Jira*

A tecnologia *Jira* escolhida permite o monitoramento de tarefas e acompanhamento de projetos garantindo a gestão de todas as suas atividades num único lugar, foi escolhida pela sua capacidade de interação com a ferramenta de controlo de versões escolhida (*Bitbucket*) facilitando o planeamento das aplicações e o acompanhamento do estado das suas respetivas tarefas através de ligações entre as tarefas e os repositórios que contém a resolução da respetiva tarefa [19].

3.18. *Power BI*

O *Microsoft Power BI* é um serviço de análise de dados que permite transformação e modelação de dados, visualizações interativas e análise avançada de dados [20].

Esta ferramenta foi utilizada como forma de analisar os dados provenientes dos testes realizados permitindo filtrar os mesmos, analisar detalhadamente cada atributo e fazer uma previsualização dos dados de forma interativa num componente de mapa, o que permitiu fazer uma análise na fase inicial da qualidade dos dados e testar as fases iniciais da aplicação ajudando a encontrar vários erros de dados e de registo.

Foi também uma forma de poder partilhar os vários resultados dos dados de testes com os docentes orientadores para ser possível discutir os resultados obtidos e entender se estes estavam dentro do padrão e que era possível identificar os padrões de testes pretendidos.

3.19. *Inkscape*

O *Inkscape* é um software de edição de gráficos vetoriais que permite manipulação de imagens vetoriais, disponibiliza ferramentas de desenho, edição de camadas, texto e tipografia, ferramentas de transformação, entre outros [21].

Esta ferramenta permitiu realizar uma legenda detalhada para a imagem que explica o trajeto de cada teste introduzindo a rota percorrida pelo veículo durante cada teste e uma legenda que permite identificar a que teste pertence cada traçado.

3.20. *Android Studio*

O *Android Studio* é um IDE (*Integrated Development Environment*) que fornece uma série de ferramentas e recursos que facilitam a criação, teste e depuração de aplicações móveis

O *Android Studio* foi escolhido como IDE para a realização da aplicação *Android* pois é o IDE que apresenta a documentação mais extensa e detalhada e a que tem compatibilidade com o um maior número de bibliotecas e *plugins*, apresenta a possibilidade de realização de aplicações em linguagem *Java* e *Kotlin* e uma forma organizada e estruturada de apresentar o seu sistema de pastas [22].

Tem ainda suporte nativo a recursos essenciais a este projeto como o acesso a sensores, localização e GPS.

3.21. *Visual Studio Code*

O *Visual Studio Code* é um editor de código-fonte projetado para ser um ambiente de desenvolvimento leve, altamente extensível e personalizável, adequado para uma variedade de linguagens de programação e projetos. Inclui depuração integrada, controlo de versões, terminal integrado e suporte para várias linguagens [23].

4. Análise

Neste capítulo é apresentada a análise concorrencial (ver Secção 4.1), o modelo de dados (ver Secção 4.2), a análise de *datasets* (ver Secção 4.3), a análise aos projetos anteriores (ver Secção 4.4) e por fim, a análise de sensores dos dispositivos móveis e da API (ver Secção 4.5).

4.1. Análise concorrencial

Antes de iniciar o planeamento inicial do projeto, foi necessário perceber que tipo de aplicações com objetivos comuns ao do projeto a realizar estavam disponíveis para a plataforma *Android*. Estas aplicações representariam mais tarde a concorrência, pelo que deveriam ser analisadas para melhor entender que funcionalidades oferecem e de que forma estas são apresentadas aos seus utilizadores.

Foram encontradas as seguintes aplicações que se consideram semelhante o suficiente para ter impacto na abordagem que iria ser tida em atenção no desenvolvimento das aplicações android para este projeto:

“*My Track*”: é uma aplicação disponível para *Android* (ver Figura 1), que permite as seguintes funcionalidades:

- Gravar uma rota;
- Editar uma rota;
- Agrupar rotas;
- Compartilhar uma rota;
- Importar e exportar rotas;
- Visualizar rotas anteriores [24].

Esta aplicação apresenta várias funcionalidades que vão de encontro às pretendidas para este projeto, como por exemplo, o facto de registar as várias rotas e poder exportar os dados das mesmas. Contudo, por outro lado, não faz registo de dados de sensores, pelo que não tem a capacidade de fazer a avaliação do tipo de manobras que ocorreram durante os percursos gravados, não apresenta ainda a possibilidade de alterar a frequência de registo e efetua a exportação dos dados de forma desorganizada e de difícil compreensão.

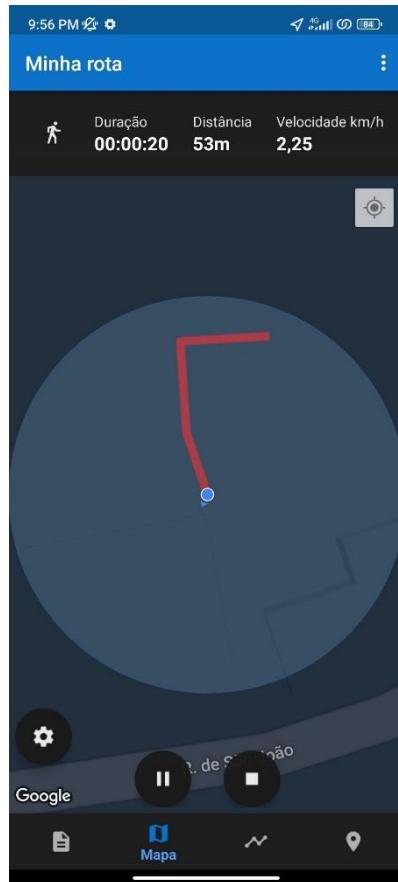


Figura 1 - Ecrã de gravação para a aplicação "My Track"

“Strava”: é uma empresa que disponibiliza uma aplicação para dispositivos *iPhone* e *Android* que permite rastrear as rotas percorridas durante a realização de atividade física. Esta aplicação recolhe informação relativa à velocidade, distância percorrida, velocidade média (ver Figura 2). Faz também a recolha de dados de sensores, como por exemplo, do sensor de batimento cardíaco. Disponibiliza ainda a partilha das rotas percorridas com outros utilizadores o que permite ver as várias rotas no mapa em simultâneo [25].

Esta aplicação não tem como objetivo o suporte para a recolha de dados de viagens de uma viatura, pelo que o registo das mesmas apresenta vários problemas na recolha de dados. Não faz leitura de dados de sensores de aceleração e giroscópio, não tendo então capacidade de analisar e classificar os tipos de manobras realizados durante as viagens. Não permite exportar os dados relativos a viagem e não possibilita a alteração da velocidade de captura dos registo efetuados durante a viagem.

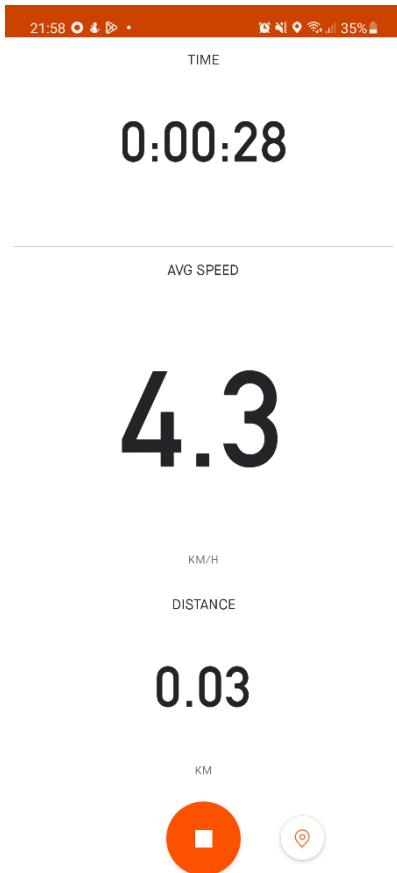


Figura 2 - Ecrã de record para a aplicação "Strava"

"Sensor Logger": É uma aplicação para dispositivo *Android* e *Apple* (*Iphone*, *Apple Watch*), que permite guardar dados de vários sensores e da localização através de GPS.

Esta aplicação tem a capacidade de gerar gráficos em tempo real, com os dados de cada sensor, mostrar dados detalhados de cada sensor e a sua descrição (ver Figura 3), fazer gravação dos dados dos sensores e exportar dados gravados em vários formatos.

Inclui ainda ferramentas de suporte à análise de dados e promove a utilização dos dados gerados para a investigação através de uma página de grupos de estudo.

Oferece uma aplicação de complemento através do uso do *Smart watch* que permitem ao utilizador inicializar as suas recolhas de dados de forma mais prática e eficiente e um painel através da *web* onde o utilizador poderá ver em tempo real os gráficos gerados através dos dados dos sensores de forma mais ampliada e através de um ecrã de maior resolução facilitando assim a leitura de dados [26].

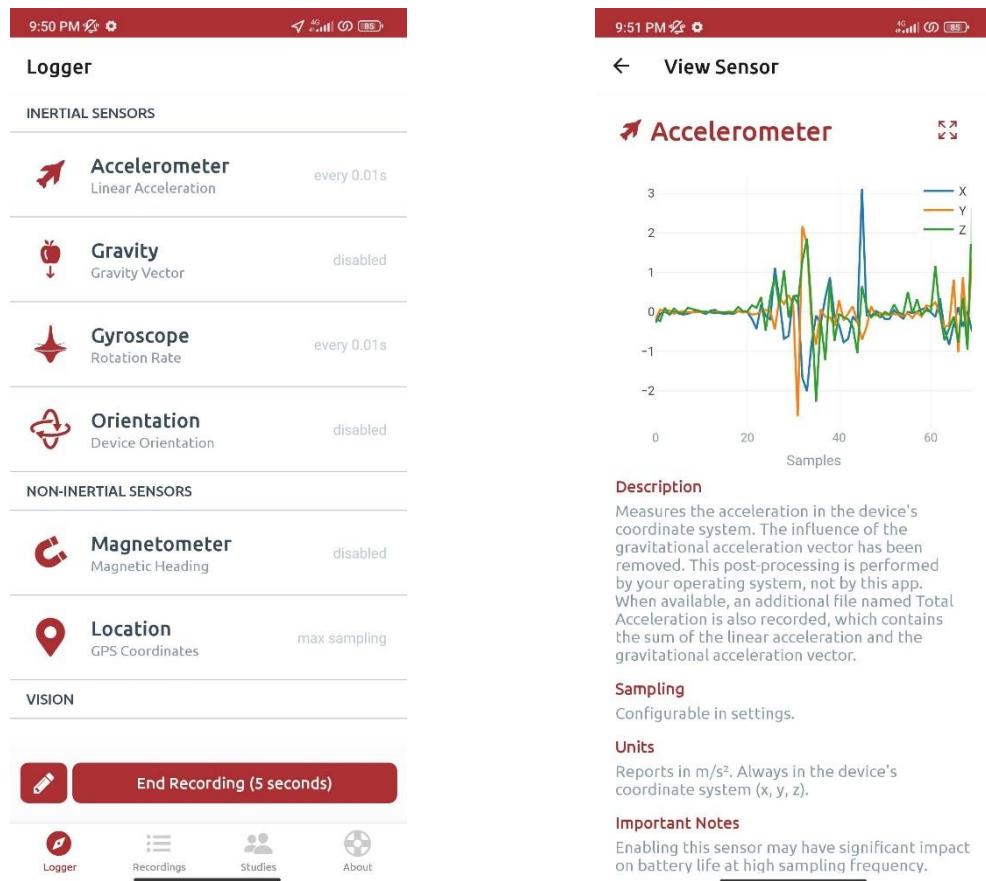


Figura 3 - Ecrãs de demonstração da aplicação "Sensor Logger"

“Sensor Data”: Permite registrar, guardar e avaliar os dados obtidos pelos sensores integrados do seu *smartphone* ou *tablet* (ver Figura 4). Inclui suporte para sensores de acelerómetro, giroscópio, campo magnético, luz, proximidade, pressão, humidade e/ou temperatura.

Além disso, oferece suporte a sensores compostos, como frequência cardíaca, contador de passos, detetor de passos, vetor de rotação, gravidade, aceleração linear e sensores não calibrados.

Permite guardar os dados automaticamente no dispositivo ou através do *Google Drive* num arquivo “.txt”, delimitado por tabulações para permitir uma análise mais aprofundada.

Permite alterar a frequência de amostragem, a duração da gravação e o número de sensores utilizados em simultaneamente [27].

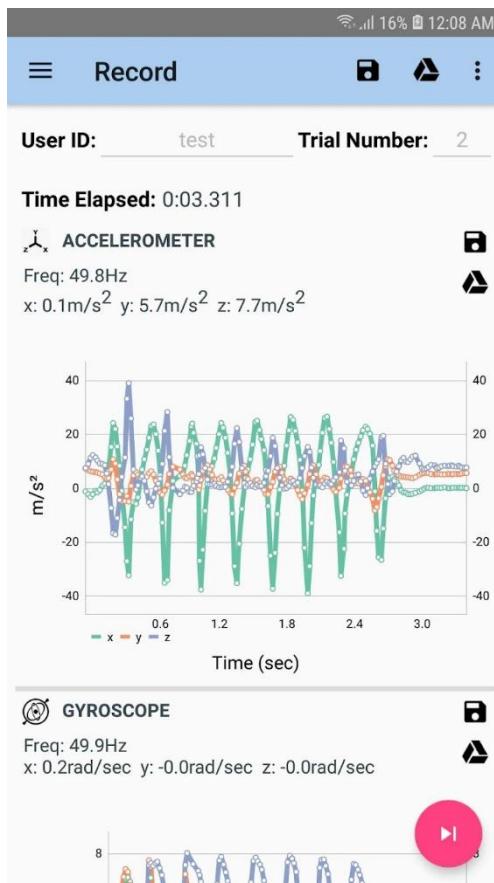


Figura 4 - Ecrã de recolha de dados da aplicação "Sensor Data"

4.2. Modelo de dados

Neste capítulo pretende-se descrever os modelos de dados utilizados para as aplicações android e para a aplicação web. Na Secção 4.2.1 é descrita a estrutura para os dados de ambas as aplicações moveis android, e na secção 4.2.2 a estrutura para o modelo de dados utilizado na aplicação *web*.

4.2.1. Modelo de dados aplicação Android

Na Figura 5 está representada a estrutura da base de dados utilizada pela aplicação cliente sendo esta uma variação simplificada da estrutura representada na Figura 6 utilizada pela aplicação de *admin/testes*.

Esta estrutura tem como entidade central a tabela de *datasets* que regista o nome do *dataset* (*label*), a data e hora da sua criação e o seu estado de sincronização com a API (sincronizado, não sincronizado). Regista ainda o tipo de aplicação que realizou o registo do respetivo *dataset* (cliente, *admin/testes*). Esta tabela relaciona-se com a tabela de *tags*, que regista as definições de captura do *dataset*, incluindo a velocidade de captura e o estado dos

diversos filtros (gravidade, *low-pass filter*). Relaciona-se também com a tabela de *devices*, que regista toda a informação referente ao dispositivo, nomeadamente os seus sensores. Por fim, também se relaciona com a tabela *capturedData* que guarda todos os registos efetuados pelos sensores e os dados de localização (latitude, longitude, *bearing*, *speed*), durante o período em que a captura de dados para aquele *dataset* esteve ativa.

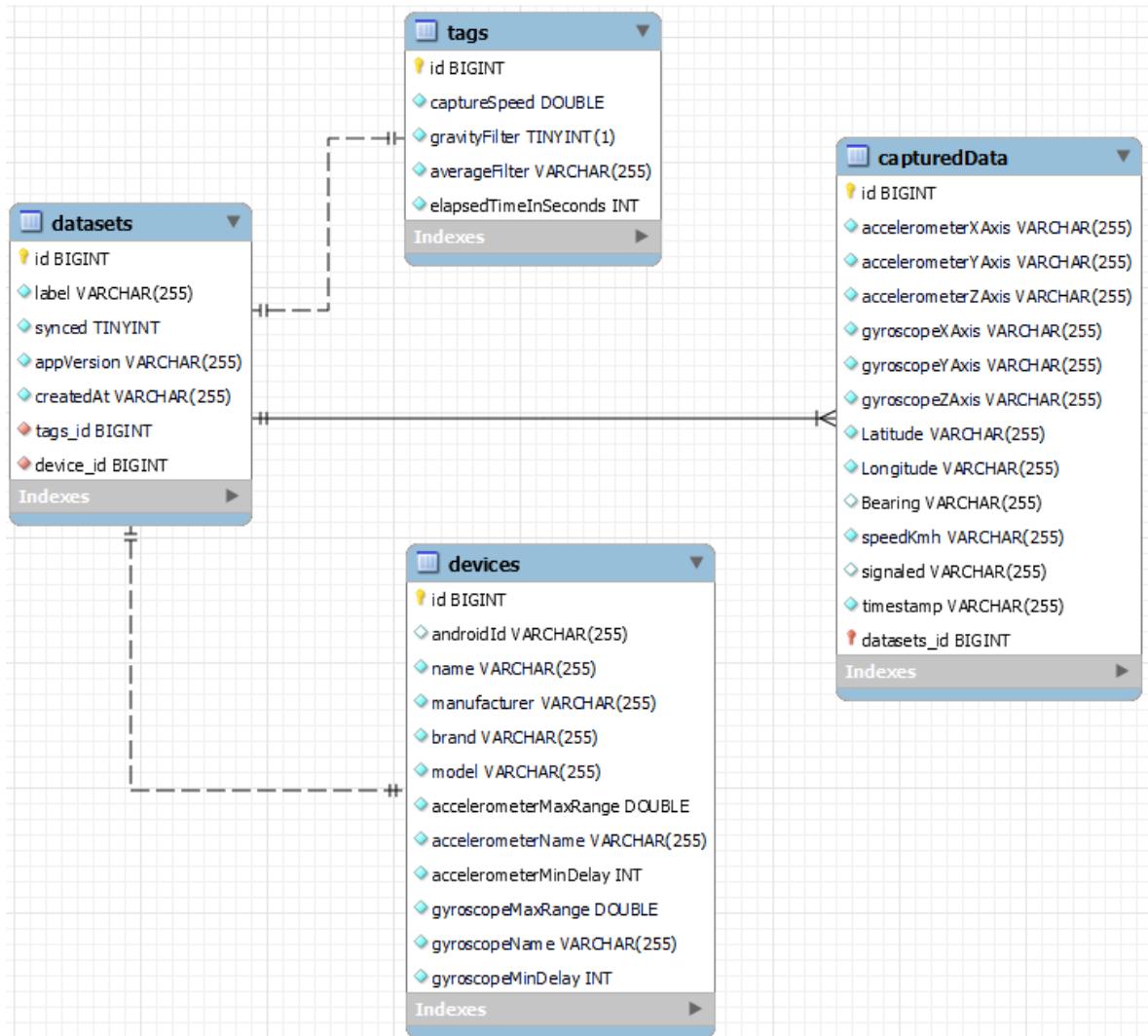
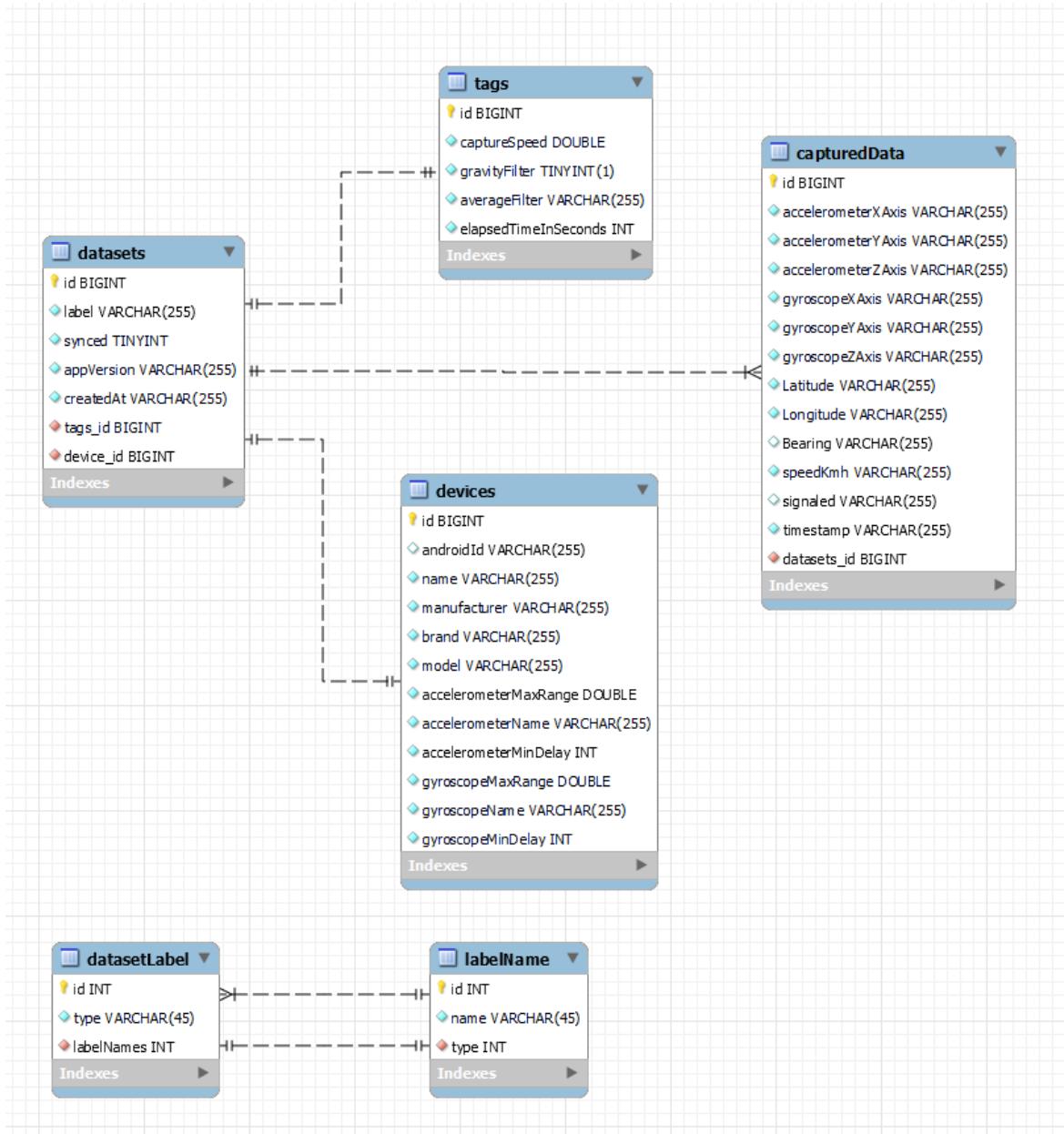


Figura 5 - Diagrama Entidade-Relacionamento para a aplicação *Android cliente*

Na Figura 6 é representado a estrutura utilizada como base de dados na aplicação *admin/testes*, que segue a mesma estrutura referida acima. No entanto, esta também inclui duas novas tabelas que auxiliam a gerar a *label* do *dataset*, fornecendo um conjunto de categorias de *labels* (*Motion*, *Steering*, *Behaviour*), que contêm várias *labels* predefinidas (10km/h, agressivo, 90°), que permitem ao utilizador da aplicação criar uma *label* complexa com apenas alguns cliques.

Figura 6 - Diagrama Entidade-Relacionamento para a aplicação *Android admin/testes*

4.2.2. Modelo de dados aplicação web

Na Figura 7 está representado a estrutura de dados utilizada na aplicação *web*, sendo que esta estrutura é baseada na estrutura da aplicação de cliente (ver Figura 5) mas adiciona à tabela de *datasets*, os atributos *deleted_at*, *created_at* e *updated_at*, que registam as datas das alterações feitas ao *dataset*.

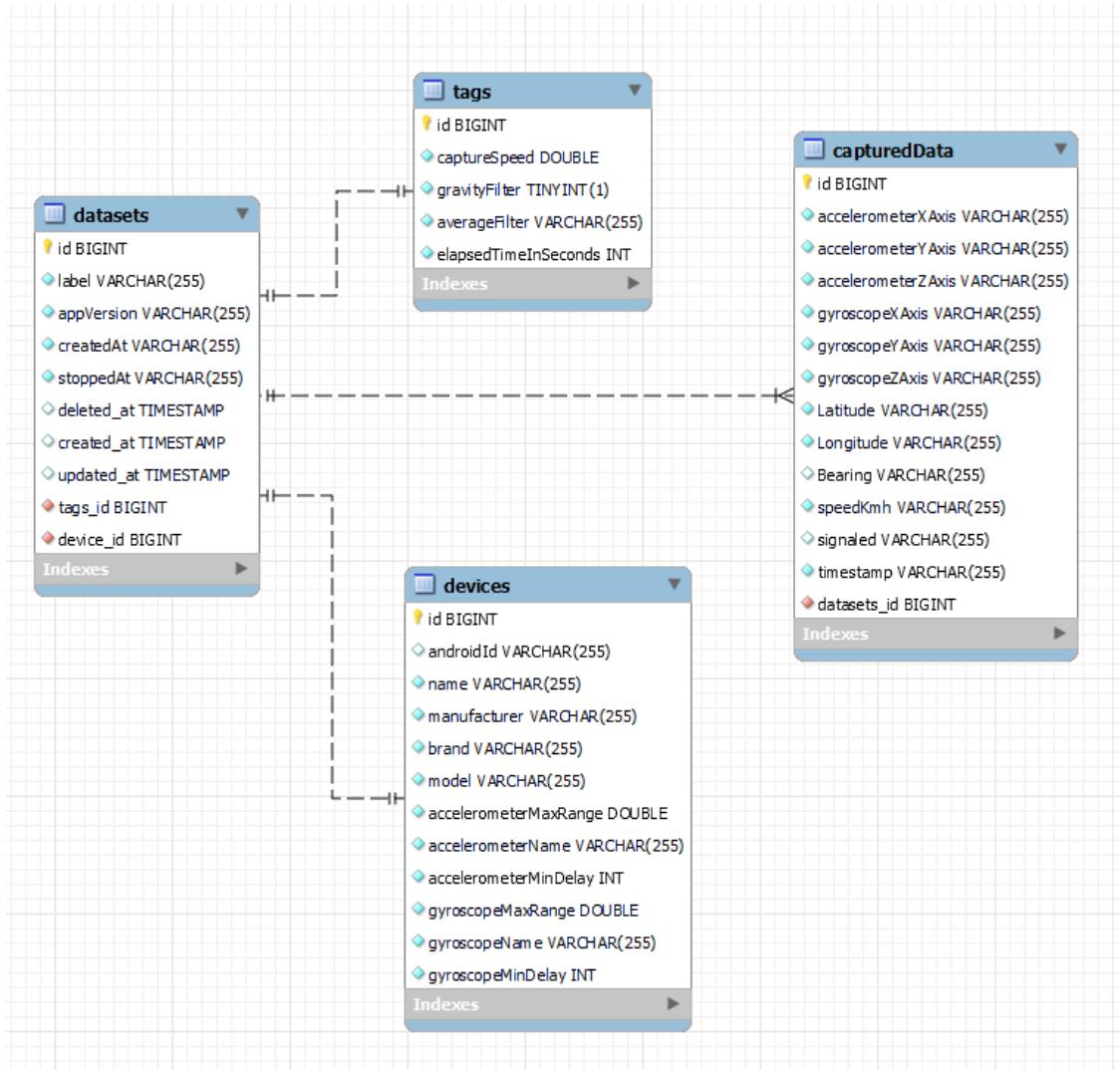


Figura 7 - Diagrama Entidade-Relacionamento para a aplicação web

4.3. Análise de *datasets*

Nesta secção é abordada a estrutura do elemento principal de recolha de dados (*dataset*) e de todos os seus elementos. É também apresentado um exemplo da estrutura de um *dataset* em formato JSON (*JavaScript Object Notation*) (ver Figura 8) e outro em formato CSV (Comma-separated values) (ver Figura 9).

4.3.1. Estrutura JSON

O *dataset* em formato JSON é composto por quatro partes fundamentais:

- **Informação do dataset:** Contém informação do *dataset* (*label* e data de criação);
- **Tags:** Apresenta os dados escolhidos através das definições;
- **Device:** Apresenta informação referente ao dispositivo e aos seus sensores;

- **CapturedData:** Apresenta os dados recolhidos pelos sensores e de localização.

```

1  {
2      "id": 35,
3      "label": "30kmh small roundabout 1st exit",
4      "createdAt": "2023-05-27 12:11:44.869",
5      "tags": {
6          "id": 35,
7          "captureSpeed": 250,
8          "gravityFilter": true,
9          "averageFilter": "Low-Pass Filter",
10         "elapsedTimeInSeconds": 9
11     },
12     "device": {
13         "id": 1,
14         "androidId": "2ccc978eef8e0f22",
15         "name": "Pieter",
16         "manufacturer": "samsung",
17         "brand": "samsung",
18         "model": "SM-A515F",
19         "accelerometerMaxRange": 78.45320129394531,
20         "accelerometerName": "LSM6DSL Accelerometer",
21         "accelerometerMinDelay": 2000,
22         "gyroscopeMaxRange": 17.45326042175293,
23         "gyroscopeName": "LSM6DSL Gyroscope",
24         "gyroscopeMinDelay": 2000
25     },
26     "capturedData": [
27         {
28             "id": 2395,
29             "accelerometerXAxis": -1.418927908,
30             "accelerometerYAxis": -0.694663823,
31             "accelerometerZAxis": 0.773861885,
32             "gyroscopeXAxis": 0.042760566,
33             "gyroscopeYAxis": -0.01038471,
34             "gyroscopeZAxis": 0.124005646,
35             "Latitude": 39.74207434337586,
36             "Longitude": -8.812896776944399,
37             "speedKmh": 29.1,
38             "signaled": null,
39             "timestamp": "2023-05-27 12:11:44.921"
40         },
41         {
42             "id": 2396,
43             "accelerometerXAxis": -1.630367517,
44             "accelerometerYAxis": -0.178227782,
45             "accelerometerZAxis": -0.629752159,
46             "gyroscopeXAxis": 0,
47             "gyroscopeYAxis": -0.051312681,
48             "gyroscopeZAxis": 0.135001212,
49             "Latitude": 39.74207434337586,
50             "Longitude": -8.812896776944399,
51             "speedKmh": 29.1,
52             "signaled": null,
53             "timestamp": "2023-05-27 12:11:45.176"
54         }
55     ]
}

```

Figura 8 - Estrutura JSON de um dataset

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|----------------------|--------------|--------------|---------------|---------------|--------------------|-------------|----------|--------------|-------------------|-------|--------------------|--------------------|--------------------|----------------|----------------|----------------|
| label | createdAt | captureSpeed | gravityFilter | averageFilter | elapsedTime | androidId | name | manufacturer | brand | brand | accelerometerXAxis | accelerometerYAxis | accelerometerZAxis | gyroscopeXAxis | gyroscopeYAxis | gyroscopeZAxis |
| 30kmh small roundab | 11:44.9 | 250 | TRUE | Low-Pass Fil | 9.2ccc978eef8e0f22 | samsung | samsung | SM-A515F | 78.45320129394531 | 2000 | 17.45326042175293 | 2000 | LSM6DSL | LSM6DSL | LSM6DSL | |
| timestamp | | | | | | | | | | | | | | | | |
| 27/05/2023 11:44.921 | -1.418927908 | -0.694663823 | 0.773861885 | 0.04276057 | -0.01038471 | 0.12400565 | 39.74207 | -8.8129 | 29.1 | null | | | | | | |
| 27/05/2023 11:45.176 | -1.630367517 | -0.178227782 | -0.629752159 | 0 | -0.051312681 | 0.13500121 | 39.74207 | -8.8129 | 29.1 | null | | | | | | |
| 27/05/2023 11:45.427 | -0.862673402 | -0.179376543 | -0.445918083 | 0.0109557 | -0.0274889 | 0.14294247 | 39.74202 | -8.81284 | 28.5 | null | | | | | | |
| 27/05/2023 11:45.578 | -0.354654588 | 0.282754868 | 0.02382374 | 0.02382374 | 0.14110987 | 39.74202 | -8.81284 | 28.5 | null | | | | | | | |
| 27/05/2023 11:45.947 | 0.395710826 | -0.301070094 | 0.662752151 | 0.00671952 | 0.00366519 | 0.1459979 | 39.74202 | -8.81284 | 28.5 | null | | | | | | |
| 27/05/2023 11:46.198 | 0.534773707 | -0.293063164 | -0.922137342 | -0.0030543 | -0.0207694 | 0.15699236 | 39.74202 | -8.81284 | 28.5 | null | | | | | | |
| 27/05/2023 11:46.448 | 0.016829342 | -0.301813364 | -0.922137342 | -0.0030543 | -0.0207694 | 0.15699236 | 39.74196 | -8.81278 | 27.7 | null | | | | | | |
| 27/05/2023 11:46.708 | 0.911335767 | -0.826909781 | 0.461242676 | -0.0177151 | -0.0561996 | 0.12889257 | 39.74196 | -8.81278 | 27.7 | null | | | | | | |
| 27/05/2023 11:46.959 | 0.11190924 | -0.442334533 | -0.127848625 | 0.03298672 | 0.00549779 | 0.09285152 | 39.74196 | -8.81278 | 27.7 | null | | | | | | |
| 27/05/2023 11:47.209 | -0.24837321 | 0.817950368 | 0.1076014519 | -0.0171042 | -0.0812451 | 0.06658431 | 39.74196 | -8.81278 | 27.7 | null | | | | | | |
| 27/05/2023 11:47.461 | -0.184759855 | -0.064348936 | -0.473161697 | -0.0262672 | -0.0555887 | 0.01343904 | 39.74193 | -8.81272 | 25.3 | null | | | | | | |
| 27/05/2023 11:47.714 | 0.071869791 | 0.311932445 | 0.455918312 | -0.0293215 | -0.0523441 | 0.0523441 | 39.74193 | -8.81272 | 25.3 | null | | | | | | |
| 27/05/2023 11:47.964 | 2.123336315 | -0.180481076 | -0.127251625 | 0.00488692 | 0.03970624 | -0.0836885 | 39.74193 | -8.81272 | 25.3 | null | | | | | | |
| 27/05/2023 11:48.216 | 0.475056589 | -0.116009951 | -0.369391441 | 0.0580322 | -0.0482584 | -0.1747075 | 39.74193 | -8.81272 | 25.3 | null | | | | | | |
| 27/05/2023 11:48.466 | 2.884114742 | -0.441191316 | 0.19814774 | -0.1288926 | 0.07941248 | -0.2461787 | 39.74193 | -8.81265 | 23.1 | null | | | | | | |
| 27/05/2023 11:48.718 | 2.726148605 | 0.182634115 | 0.53617582 | -0.0256563 | -0.0360411 | 0.3714061 | 39.74193 | -8.81265 | 23.1 | null | | | | | | |
| 27/05/2023 11:48.972 | 0.671584129 | 0.292858958 | 0.37905888 | -0.07452355 | 0.05192355 | -0.4557055 | 39.74193 | -8.81265 | 23.1 | null | | | | | | |
| 27/05/2023 11:49.230 | 0.215510368 | 2.0629282 | -0.1325578 | 0.13439035 | -0.5247332 | 0.3974187 | -8.81259 | 20.9 | null | | | | | | | |
| 27/05/2023 11:49.482 | 0.042318344 | -0.564054608 | 0.772481918 | -0.0250455 | 0.01343904 | -0.54648918 | 39.74187 | -8.81259 | 20.9 | null | | | | | | |
| 27/05/2023 11:49.733 | 0.751258373 | -0.823885918 | -0.676999092 | 0.1148427 | 0.08185598 | -0.5326745 | 39.74187 | -8.81259 | 20.9 | null | | | | | | |
| 27/05/2023 11:49.985 | 0.057741642 | -1.248344898 | 0.895137787 | -0.06900278 | 0.04276057 | -0.5681047 | 39.74187 | -8.81259 | 20.9 | null | | | | | | |
| 27/05/2023 11:50.238 | 0.664840221 | -1.027275085 | 0.218286514 | 0.00549779 | 0.08979719 | -0.5723807 | 39.74187 | -8.81259 | 20.9 | null | | | | | | |
| 27/05/2023 11:50.489 | 0.353258133 | 0.772666931 | -0.0513127 | 0.03420845 | -0.5650504 | 0.3974183 | -8.81255 | 17.9 | null | | | | | | | |
| 27/05/2023 11:50.741 | 0.145707607 | -1.327056527 | 0.413094521 | -0.0189388 | 0.05681047 | -0.5705481 | 39.74183 | -8.81255 | 17.9 | null | | | | | | |
| 27/05/2023 11:50.995 | 0.367152214 | -1.191453564 | 0.665036201 | 0.0433714 | 0.00733038 | -0.5821546 | 39.74183 | -8.81255 | 17.9 | null | | | | | | |
| 27/05/2023 11:51.247 | 0.24902153 | -0.430236459 | 0.706525803 | -0.0800233 | -0.0048869 | -0.5632178 | 39.74179 | -8.81257 | 21.9 | null | | | | | | |
| 27/05/2023 11:51.498 | -0.746078968 | -0.486767948 | 0.322633743 | -0.0879646 | 0.01832596 | -0.5253441 | 39.74179 | -8.81257 | 21.9 | null | | | | | | |
| 27/05/2023 11:51.753 | 0.661320686 | -0.411632627 | -0.104777336 | -0.0714712 | -0.0061087 | -0.5057964 | 39.74179 | -8.81257 | 21.9 | null | | | | | | |
| 27/05/2023 11:52.004 | 0.535007477 | 0.329638332 | -0.658975601 | -0.0702495 | 0.05497787 | -0.4734206 | 39.74179 | -8.81257 | 21.9 | null | | | | | | |
| 27/05/2023 11:52.258 | -1.198420048 | -0.210102007 | 0.907342911 | 0.01038471 | 0.04092797 | -0.3723287 | 39.74175 | -8.81263 | 25.5 | null | | | | | | |
| 27/05/2023 11:52.511 | -1.004306793 | -0.039007634 | 0.221868515 | -0.0201582 | 0.04581489 | -0.2223549 | 39.74188 | -8.81261 | 0 | null | | | | | | |
| 27/05/2023 11:52.764 | -0.448925018 | -0.336526781 | 0.084182794 | 0.00671952 | -0.0042761 | -0.181427 | 39.74188 | -8.81261 | 0 | null | | | | | | |

Figura 9 - Estrutura de um dataset em CSV

4.3.2. Estrutura CSV

O *dataset* em formato CSV é composto por duas partes fundamentais:

- **Cabeçalho:** Contém os elementos correspondentes ao *dataset*, dispositivo e *tags*;
- **Corpo do documento:** Contém a lista de dados capturados para o respetivo *dataset*.

4.3.3. Elementos do *dataset*

O *dataset* é composto pelos seguintes elementos:

- ***label*:** Nome identificador do *dataset*;
- ***createdAt*:** Data e hora em que o *dataset* foi criado;
- ***captureSpeed*:** Frequência de captura em milissegundos;
- ***gravityFilter*:** Quando ativo o acelerómetro remove o efeito da gravidade sobre os seus valores;
- ***averageFilter*:** Filtros que permitem suavizar os valores obtidos pelos sensores;
- ***elapseTimeInSeconds*:** Tempo de duração da captura do *dataset* em segundos;
- ***androidID*:** Identificador exclusivo atribuído a um dispositivo *Android*;
- ***name*:** Nome escolhido para o dispositivo na primeira instalação da aplicação;
- ***manufacturer*:** Nome do fabricante do dispositivo;
- ***brand*:** Marca do dispositivo;
- ***model*:** Modelo do dispositivo;
- ***accelerometerMaxRange*:** Valor máximo que o acelerómetro do dispositivo pode atingir;
- ***accelerometerName*:** Nome do sensor acelerómetro;
- ***accelerometerMinDelay*:** Taxa mínima da leitura do sensor em milissegundos;
- ***gyroscopeMaxRange*:** Valor máximo que o giroscópio do dispositivo pode atingir;
- ***gyroscopeName*:** Nome do sensor giroscópio;
- ***gyroscopeMinDelay*:** Taxa mínima da leitura do sensor em milissegundos;
- ***accelerometerXAxis*:** Valor obtido pelo acelerómetro em m/s² para o eixo X;
- ***accelerometerYAxis*:** Valor obtido pelo acelerómetro em m/s² para o eixo Y;

- **accelerometerZAxis**: Valor obtido pelo acelerómetro em m/s² para o eixo Z;
- **gyroscopeXAxis**: Valor obtido pelo giroscópio em radianos/s para o eixo X;
- **gyroscopeYAxis**: Valor obtido pelo giroscópio em radianos/s para o eixo Y;
- **gyroscopeZAxis**: Valor obtido pelo giroscópio em radianos/s para o eixo Z;
- **latitude**: Valor da latitude obtida em graus decimais;
- **longitude**: Valor da longitude obtida em graus decimais;
- **speedmh**: Velocidade atual na unidade de km/h;
- **signaled**: Atributo que identifica o momento fulcral em que uma manobra ocorreu;
- **timestamp**: Data e hora a que os dados de sensor se referem.

4.4. Análise de projetos anteriores

Este projeto foi realizado tendo por base dois projetos realizados por dois grupos de estudantes no contexto da disciplina de projeto informático.

Foi realizado uma análise inicial a ambos os projetos, que consistiu em realizar o processo de instalação dos mesmos seguindo o respetivo manual de utilização disponibilizado no relatório que acompanhava os projetos.

Foi realizada uma análise detalhada de todas as funcionalidades para que fosse possível obter uma lista de funcionalidades presentes nos projetos e tentar detetar potenciais problemas presentes nessas funcionalidades.

Após análise foram detetados vários problemas com a aplicação *Android* disponibilizada, desde dados de sensores posicionados incorretamente, opções de definições que não realizavam corretamente a sua função, e dados de GPS que não representavam a posição atual corretamente, mas sim um valor que deveria ter sido considerado como erro de leitura.

Foi decidido em conjunto com os docentes orientadores que as aplicações *Android* a realizar não teriam por base o código disponibilizado através do projeto realizado anteriormente e que seriam realizadas aplicações novas de raiz.

A análise à aplicação *web* disponibilizada revelou a qualidade de implementação da mesma pelo que em conjunto com os docentes orientadores foi decidido que seriam utilizados vários componentes da mesma para a realização da nova aplicação *web*.

Este projeto tinha como objetivo o desenvolvimento de um conjunto de componentes com base nos projetos realizados anteriormente pelos outros dois grupos. Contudo, estes componentes foram desenvolvidos de raiz, com aspectos melhorados, corrigidos e com funcionalidades novas.

4.5. Análise de sensores dos dispositivos móveis e da API

Após análise dos elementos fornecidos (projetos anteriores e relatórios), foram analisados os elementos fundamentais às aplicações, neste caso sensores (acelerómetro, giroscópio), mapas e GPS.

No caso dos sensores, foi elaborada uma análise que inclui a pesquisa detalhada do seu funcionamento, os seus parâmetros de funcionamento (eixos, limites, taxa de atualização) e as várias formas possíveis da inclusão dos mesmos nas aplicações onde seriam utilizados.

A documentação disponibilizada através do *Android Developer* ([28]), facilitou a utilização e entendimento relativo à utilização dos sensores. A documentação explica com precisão o funcionamento dos sensores necessários e a forma correta de implementação dos mesmos nas aplicações *Android*. O facto de incluir estes sensores de forma nativa e disponibilizar diversas informações e parâmetros relativos aos sensores utilizados, permitiu que a implementação destes elementos fosse realizada de forma planeada e eficiente.

4.5.1. Sensor de aceleração

Sensor de aceleração (ver Figura 10), usado para medir a aceleração própria de um sistema, utilizado pela aplicação para registar e identificar a velocidade que o dispositivo atingiu num determinado eixo em determinado momento, o que permite identificar acelerações e desacelerações assim como movimentos laterais.

O sensor de aceleração permite medir forças em três dimensões (X, Y, Z). É utilizado como forma de detetar a orientação do dispositivo e interações por gestos.

Métricas: Aceleração (m/s^2), ângulo de inclinação (graus), intensidade de vibração.

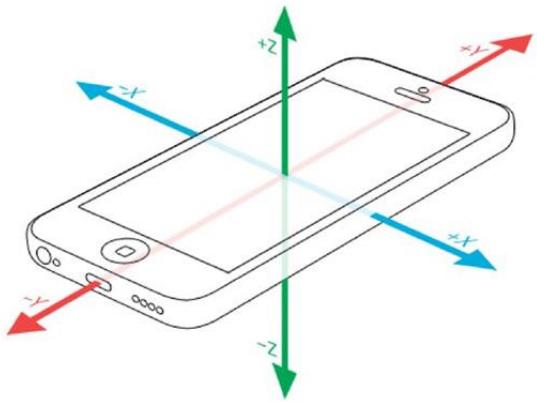


Figura 10 - Representação dos eixos de um sensor de aceleração (retirado de [29])

4.5.2. Sensor de giroscópio

Sensor giroscópio (ver Figura 11), usado para medir a velocidade angular própria de um sistema. É utilizado pela aplicação para registar com mais precisão manobras em que o veículo necessita de realizar uma mudança de direção de forma mais leve e faseada.

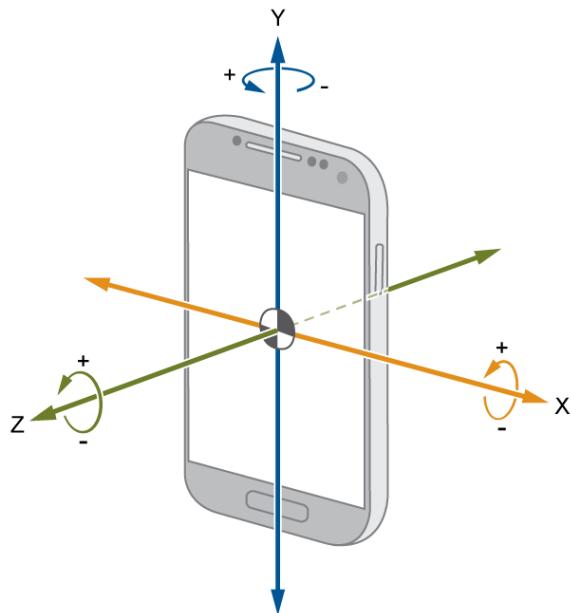


Figura 11 - Representação dos eixos de um sensor de giroscópio (retirado de [30])

Este sensor é utilizado em complemento do sensor de aceleração, para identificar manobras com mudança de direção de forma faseada, o que seria mais dificilmente analisado através dos valores obtidos pelo sensor de aceleração.

Este sensor oferece também um complemento à informação obtida através dos valores de *bearing*.

O sensor de giroscópio permite medir velocidade angular ou a taxa de rotação de um dispositivo para três eixos (X, Y, Z). É usado para rastrear a orientação do dispositivo e fornecer deteção de movimento de forma mais precisa.

Métricas: Taxa de rotação (radianos/s).

4.5.3. Localização

A localização é usada para obter informação relativa à posição do dispositivo. Utilizado pela aplicação para registar: latitude, longitude, velocidade e *bearing*, de forma a poder situar um dispositivo e consequentemente um veículo no espaço e tempo.

A localização permite, em conjunto com os dados dos sensores, entender e analisar detalhadamente dados recolhidos em vários veículos e interpretar os vários detalhes antecedentes a situações de perigo ou mesmo de acidente.

Métricas: Latitude/Longitude (graus decimais), precisão(m), velocidade(m/s), *bearing* (graus).

4.5.4. API *Android* para sensores

A API de Sensores do *Android* é uma interface de programação disponibilizada pela plataforma *Android*, que permite aos programadores aceder e utilizar os vários sensores embutidos disponíveis nos seus dispositivos.

Os sensores são componentes de *hardware* que medem diversas grandezas físicas, como aceleração, rotação, intensidade de luz e proximidade.

A API de Sensores fornece uma forma padronizada de interagir com estes sensores e obter os seus dados. Disponibiliza várias classes e interfaces que permitem realizar as seguintes tarefas:

Descoberta de sensores: Através da classe “*SensorManager*”, é possível descobrir quais os sensores disponíveis num dispositivo *Android*.

Recolher dados dos sensores: Após identificar o sensor desejado, pode-se registar um *listener* usando a classe “*SensorManager*”.

A API disponibiliza eventos e *callbacks* específicos de cada sensor para lidar com a informação recebida.

O evento disponibilizado apresenta ao utilizador um *array* de valores que corresponde aos valores medidos pelo sensor naquele momento.

Precisão dos sensores: A API fornece informações acerca da precisão de cada sensor, assim como a sua capacidade máxima de atualização (nome do sensor, *delay* mínimo entre recolhas de dados e intervalo de valores no qual o sensor opera).

Cada sensor apresenta através da classe “*SensorManager*” uma lista de informações referentes ao sensor físico (características do *hardware*).

4.5.5. API android para localização

A API de Localização do *Android*, é uma interface de programação disponibilizada pela plataforma *Android*, que permite aos programadores obter informações sobre a localização geográfica dos dispositivos *Android*.

Esta API possibilita aceder a dados de localização provenientes de GPS, rede móvel e *wi-fi*.

Esta API fornece através do objeto “*location*” informação detalhada da localização do dispositivo (latitude, longitude, velocidade atual, *bearing*, altitude e nome do provedor de serviço).

A API de Localização disponibiliza várias classes e interfaces que permitem realizar as seguintes tarefas:

Obtenção da localização atual: Através da classe “*LocationManager*”, é possível solicitar atualizações da localização atual do dispositivo.

Solicitação de atualizações de localização: permite solicitar atualizações periódicas ou por eventos sempre que a localização do dispositivo muda.

Deteção de mudança de localização: A API de Localização permite detetar quando o dispositivo se move para uma nova localização.

5. Arquitetura do sistema

Na Figura 12 é representado o esquema referente à arquitetura do sistema que resume os vários componentes do projeto e as interações que ocorrem entre eles.

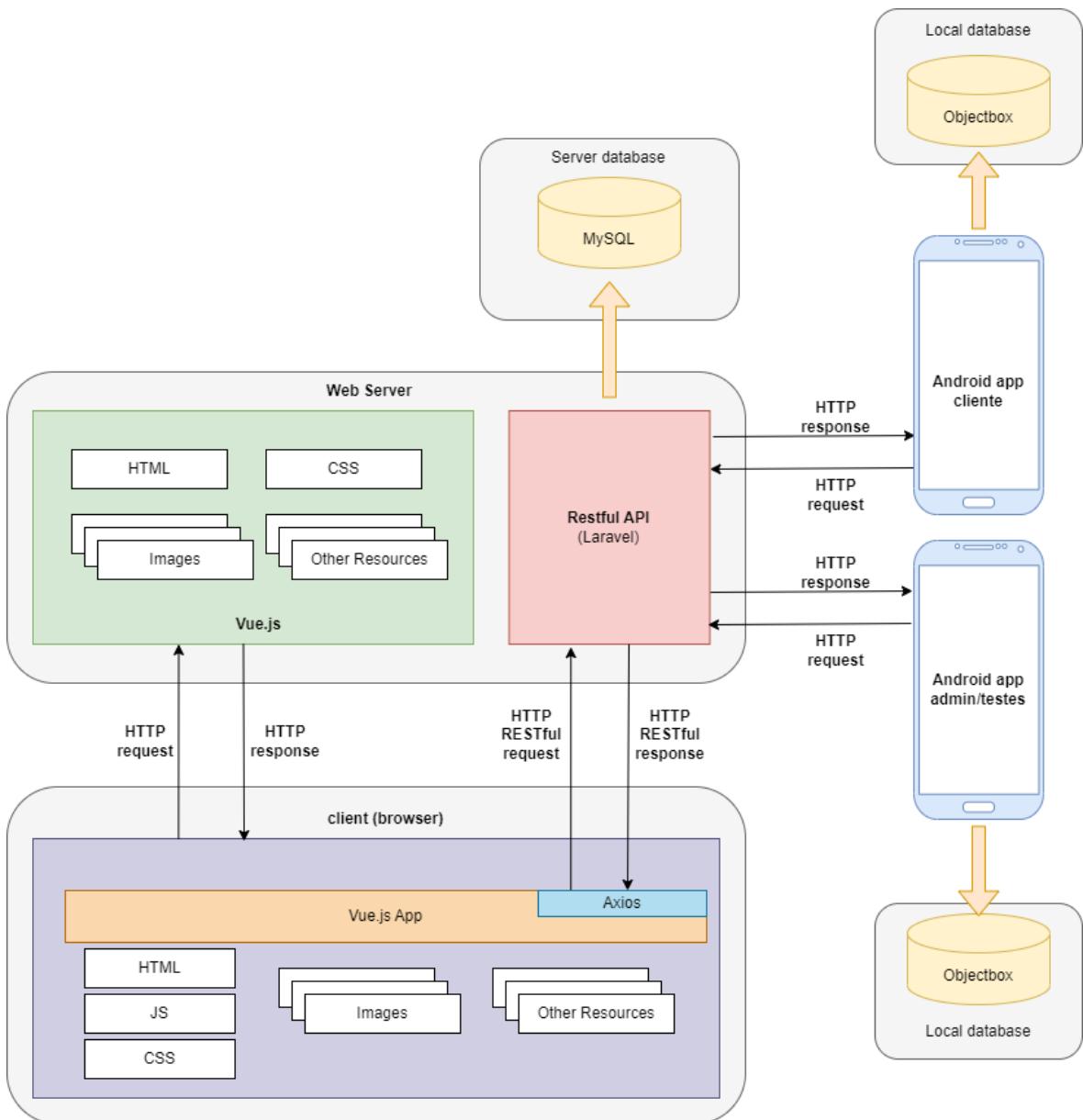


Figura 12 - Arquitetura do sistema

O projeto divide-se em dois grandes componentes principais:

- **Plataforma Web:**

- **Client (browser):** Aplicação *Vue.js* é constituída por um conjunto de componentes. Comunica com a API *RESTful (Laravel)* para receber e

atualizar os dados que são apresentados nas vistas. Esta comunicação é efetuada por meio de pedidos HTTP *RESTful*, usando a biblioteca *Axios*.

- **API RESTful (Laravel)**: Constituída por um conjunto de modelos e controladores que gerem os pedidos que são recebidos e comunicam com um servidor de base de dados relacional *MySQL* (localizado num servidor *web*), para o armazenamento de dados.

- **Aplicações móveis Android**: Cada aplicação usa uma base de dados local, *Objectbox*, onde são guardados localmente os dados. Faz uso de uma ligação à API *RESTful (Laravel)* para armazenar a informação enviada na base de dados *MySQL* (localizado num servidor *web*).

As aplicações apresentam um processo de sincronização automático que deteta se é possível efetuar a conexão à API. Se for possível é enviado um pedido à API que irá realizar a sincronização dos *datasets* guardados localmente no dispositivo móvel para a base de dados disponibilizada pela plataforma *web*.

As aplicações utilizam a biblioteca *Volley* para realizar o envio dos pedidos HTTP para a API no servidor *web* e para receber a resposta do pedido.

A resposta aos pedidos HTTP recebida através do *Volley* é então analisada para determinar se o pedido foi realizado com sucesso ou se existe a necessidade de voltar a fazer o pedido.

6. Desenvolvimento

Neste capítulo serão abordadas as duas componentes de desenvolvimento.

Componente *Android* (ver Secção 6.1), que descreve as funcionalidades desenvolvidas para a aplicação de testes/*admin* e para a aplicação de cliente e a interação que esta tem com a aplicação *web*.

Componente de aplicação *web* (ver Secção 6.2), que descreve as várias funcionalidades desenvolvidas, incluindo uma API *Restfull* que providencia a ligação entre a base de dados e aplicação de *frontend* em *Vue.js*.

6.1. Aplicação móvel *Android*

6.1.1. Aplicação Cliente

A aplicação é destinada ao utilizador final e providencia funcionalidades de utilização mais práticas para que o utilizador se possa focar apenas no realizar das suas gravações de dados, não tendo assim de ter qualquer conhecimento sobre sensores ou dos vários componentes presentes na secção de configuração da aplicação de testes.

- **Registar o dispositivo:** O utilizador regista o seu dispositivo na primeira vez que instala a aplicação, introduzindo o nome que pretende dar ao seu dispositivo (ver Figura 13).
- **Iniciar uma captura:** Permite ao utilizador iniciar um registo que irá gerar um *dataset* automaticamente, onde será guardado a informação recolhida dos sensores e da localização (ver Figura 14).
- **Parar captura:** Permite ao utilizador finalizar a recolha de dados que iniciou (ver Figura 15).
- **Página de informações:** Página que apresenta ao utilizador todos os detalhes relacionados com o projeto e a aplicação (ver Figura 16).

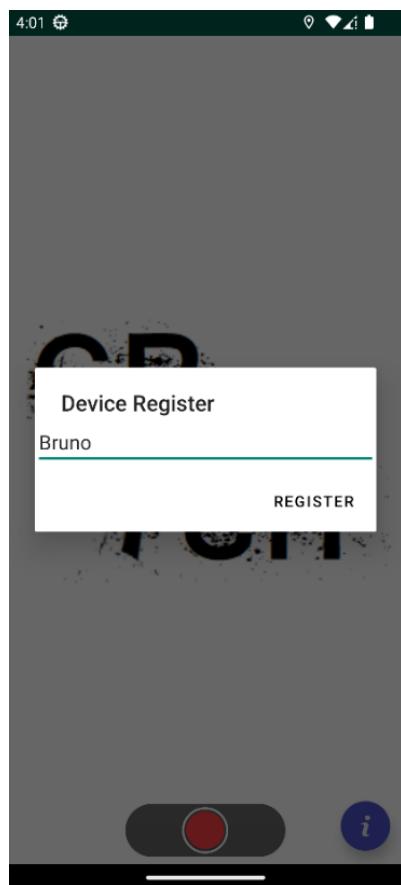


Figura 13 - Ecrã de registo de dispositivo



Figura 14 - Ecrã de iniciar captura



Figura 15 - Ecrã que permite parar a captura

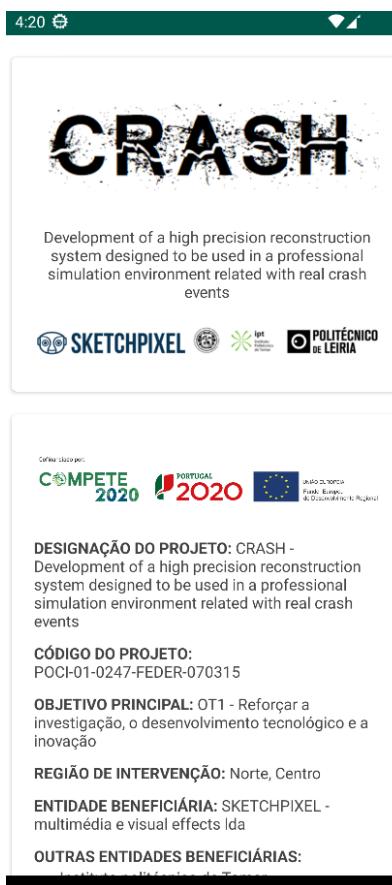


Figura 16 - Ecrã de detalhes da aplicação

- **Sincronização automática de dados:** Função que sincroniza os *datasets* automaticamente, esta função é ativada sempre que o utilizador termina de realizar uma capturada, de forma a enviar os novos dados para a aplicação *web*. Esta funcionalidade tem a capacidade de detetar problemas de rede ou problemas com o envio dos dados para aplicação *web*, através da API e efetuar um novo pedido assim que estes problemas sejam detetados como corrigidos pela aplicação (ver Figura 17).



Figura 17 - Mensagem de sincronização automática

6.1.2. Aplicação de testes/admin

A aplicação é destinada à realização de testes e fornece ferramentas para a visualização em tempo real de dados de sensores, opções para a frequência de captura, filtragem de dados, visualização de *datasets* e exportação de dados em formato JSON e CSV.

- **Registrar o dispositivo:** O utilizador regista o seu dispositivo na primeira vez que instala a aplicação introduzindo o nome que pretende dar ao seu dispositivo (ver Figura 18).

- **Visualizar dados em tempo real dos sensores:** Permite ao utilizador visualizar os dados obtidos em tempo real pelos sensores de aceleração giroscópio, visualizar o nome dos sensores que o dispositivo possui e os valores máximos e mínimos que os sensores atingiram até ao momento, como é possível visualizar na Figura 19.
Permite visualizar a localização atual do dispositivo através dos dados de latitude/longitude, a velocidade a que este se encontra, assim como a velocidade média e máxima que atingiu.
- **Criar *label* para *dataset*:** Ecrã que disponibiliza ao utilizador várias palavras-chave, agrupadas por grupos, para facilitar a construção de uma *label* genérica, que identifique o tipo de manobras presentes no *dataset* que irá ser guardado. Permite ao utilizador gerar uma *label* para o *dataset*, clicando nas várias palavras-chave que deseja incluir, ou, eventualmente escrevendo manualmente o nome que desejar, utilizando a caixa de texto no topo do ecrã (ver Figura 20).
- **Criar uma *label*:** Permite ao utilizador criar uma palavra-chave (*label*) dentro de um grupo à sua escolha, para que a possa passar a utilizar de forma mais prática (ver Figura 21).
- **Remover uma *label*:** Permite ao utilizador remover uma palavra-chave (*label*), realizando um clique longo sobre o nome da *label* a remover e confirmando que pretende eliminar a mesma (ver Figura 22).
- **Re-record:** Opção que permite ao utilizador iniciar uma nova captura utilizando a *label* da última captura. Permite que o utilizador não perca tempo, em caso de ter ocorrido algum problema com a captura anterior, ou que este apenas queira realizar um novo teste utilizando a *label* anterior.
Elimina a necessidade de ter de selecionar novamente as palavras-chaves desejadas ou ter de introduzir a *label* manualmente (ver Figura 23).

- **Sincronização:** Opção que permite ao utilizador realizar a sincronização dos *datasets* que ainda não foram enviados para a aplicação *web*, esta funcionalidade tem a capacidade de detetar problemas de rede ou problemas com o envio dos dados para aplicação *web* através da API e efetuar um novo pedido logo que estes problemas sejam detetados como corrigidos pela aplicação (ver Figura 24).
- **Página de definições:** Página que permite ao utilizador mudar definições da aplicação, incluindo a velocidade de captura, filtro de gravidade (sensor de aceleração utiliza ou não o cálculo através da gravidade), *low-pass filter* e opções de extração (ver Figura 25).
- **Página de visualização de dados gravados:** Página que permite ao utilizador verificar alguns dados dos *datasets* que guardou, nomeadamente a sua *label* a data de registo, o número de registo e o estado de sincronização. Nesta página o utilizador tem como opções, a exportação de um ou vários *datasets* sendo estes exportado conforme as definições que o utilizador selecionou na sua página de definições, e a eliminação dos *datasets* presentes na lista (ver Figura 26).
- **Exportação de dados:** Opção que permite ao utilizador exportar os dados dos seus *datasets* guardados em formato JSON e/ou CSV podendo realizar esta exportação separando cada *dataset* num ficheiro ou exportando um único ficheiro com a informação de todos os *datasets* selecionados (ver Figura 27).
 - **Sinalizar momentos da captura:** Permite ao utilizador sinalizar momentos específicos durante a gravação de dados que este considere de maior importância para análise posterior (ver Figura 28).

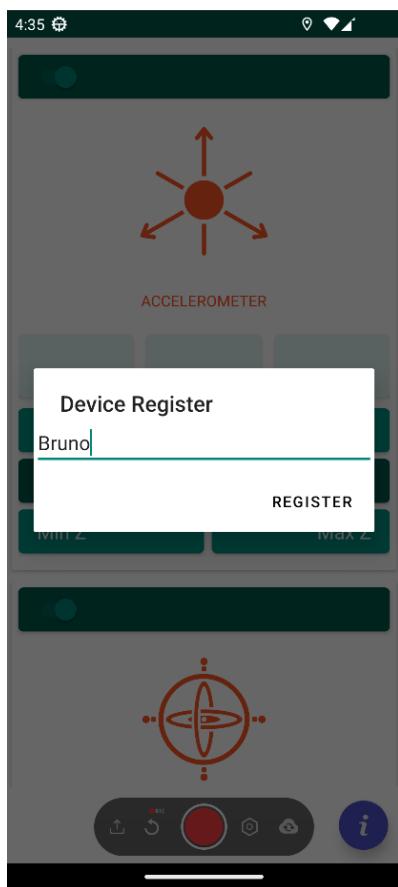


Figura 18 - Ecrã de registo de dispositivo

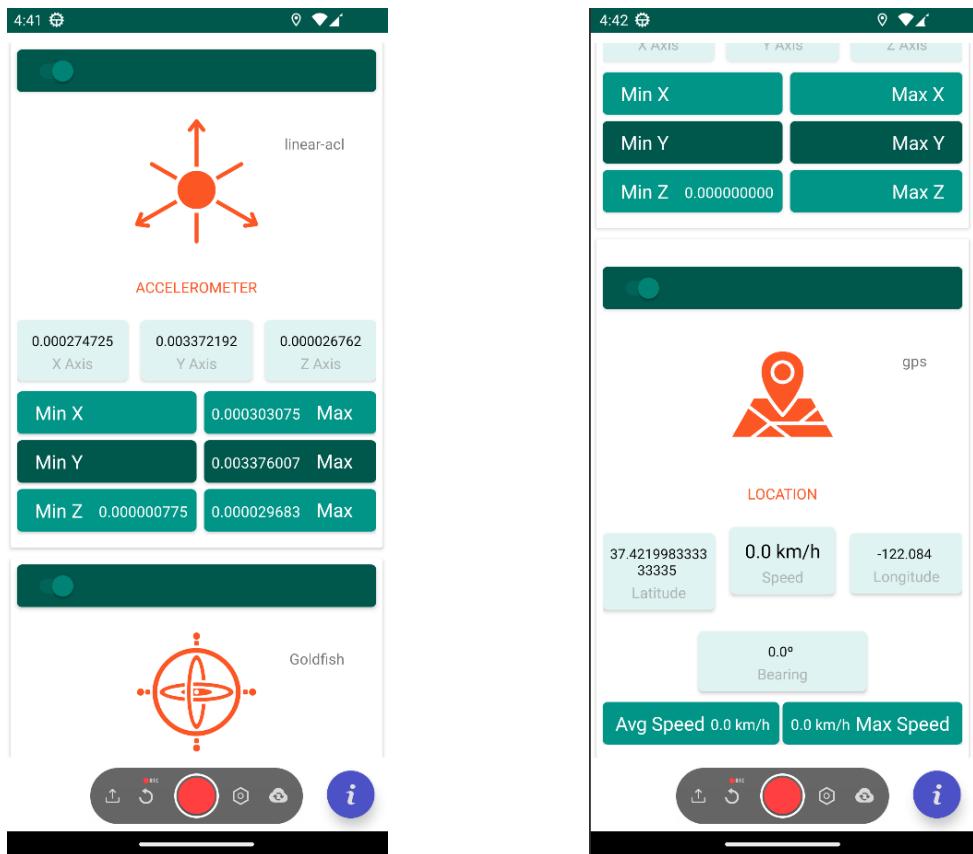


Figura 19 - Ecrãs que permitem visualizar os dados dos sensores e localização

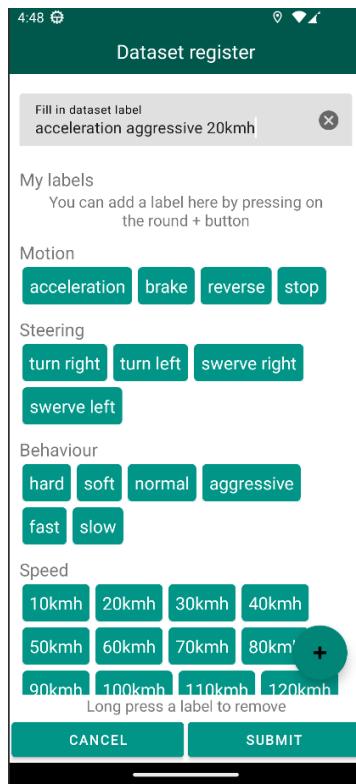


Figura 20 - Ecrã de criação da *label* do dataset

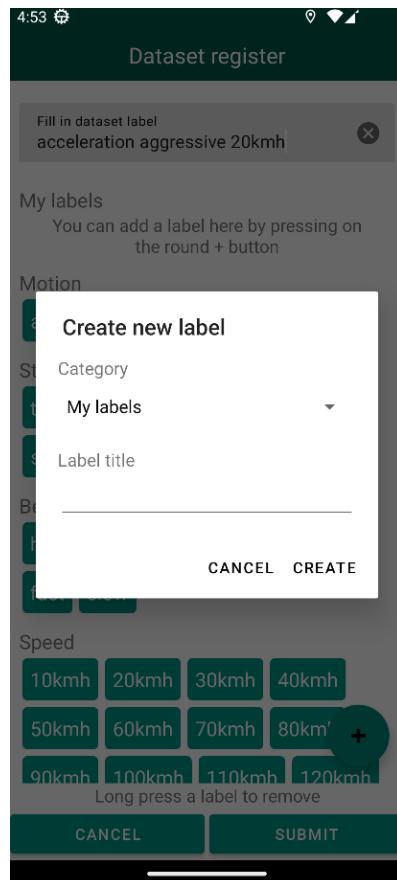


Figura 21 - Ecrã para a criação de uma nova palavra-chave

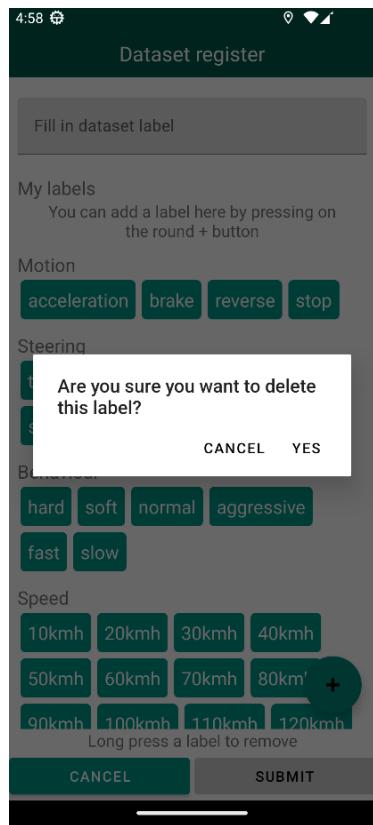


Figura 22 - Ecrã que permite eliminar uma *label*



Figura 23 - Opção de *re-record*



Figura 24 - Opção de sincronização

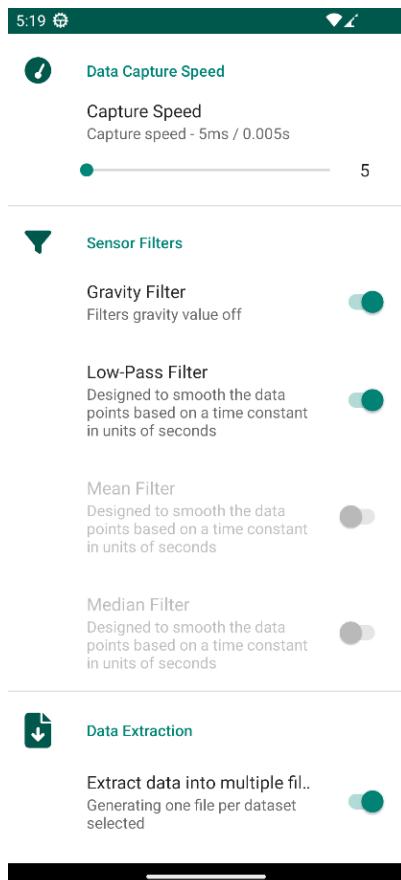


Figura 25 - Página de definições

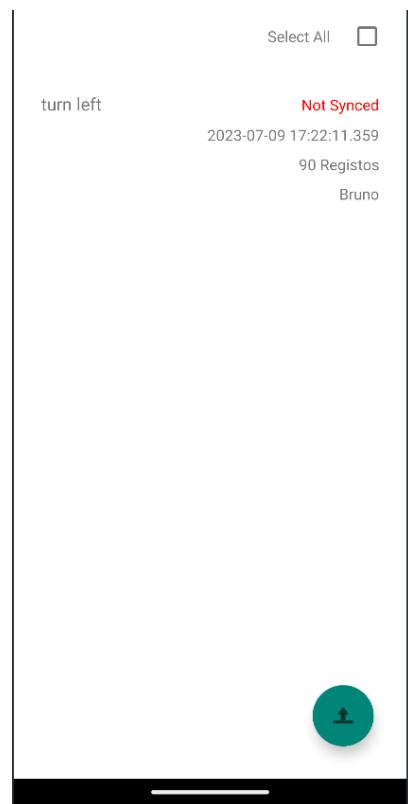


Figura 26 - Listagem de *datasets* gravados

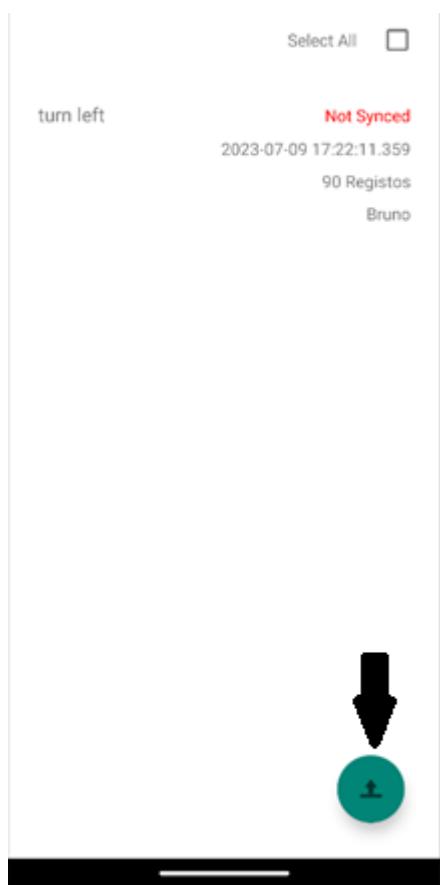


Figura 27 - Opção para exportação de *datasets*

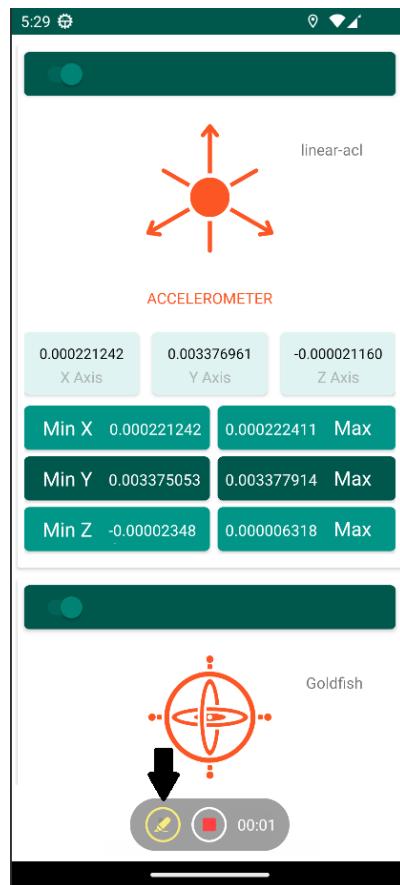


Figura 28 - Opção de sinalizar dados de captura

6.2. Aplicação Web

- **Página de listagem de datasets:** Esta página permite ao utilizador ver uma listagem dos *datasets* que estão disponíveis, importar novos *datasets*, filtrar e pesquisar a listagem pelos diversos atributos, eliminar *datasets*, transferir os dados do dataset em formato JSON e aceder aos detalhes de um *dataset* (ver Figura 29).

| Label | User device name | Application | captureSpeed (ms) | Duration (s) | Created At | Ended at | Actions |
|--|------------------|-------------|-------------------|--------------|-------------------------|-------------------------|---------|
| 8df562e32bb82f93-05-07-2023-14-53-57-575 | M0hamede | Client | 250 | 2 | 2023-07-05 14:53:57.606 | 2023-07-05 14:53:59.143 | |
| 8df562e32bb82f93-05-07-2023-14-54-20-212 | M0hamede | Client | 250 | 1 | 2023-07-05 14:54:20.238 | 2023-07-05 14:54:21.011 | |
| 8df562e32bb82f93-05-07-2023-14-54-36-157 | M0hamede | Client | 250 | 1 | 2023-07-05 14:54:36.177 | 2023-07-05 14:54:36.703 | |
| 8df562e32bb82f93-05-07-2023-14-54-43-742 | M0hamede | Client | 250 | 2 | 2023-07-05 14:54:43.760 | 2023-07-05 14:54:45.050 | |

Total datasets: 4

Import new dataset(s) (.json)

Figura 29 - Página de listagem de datasets

- **Detalhes do dataset:** Página que apresenta informação detalhada do respetivo *dataset*, incluindo os dados do dispositivo, *tags*, e dados de captura. Permite a visualização dos dados de captura assinalados no mapa através de marcadores, realizar o *download* em formato JSON dos dados e eliminar o *dataset* (ver Figura 30).

Figura 30 - Página de detalhes de um dataset

- **Dashboard:** Página principal da aplicação que permite visualizar as rotas correspondentes aos *datasets*, que correspondem aos parâmetros de filtragem selecionados (ver Figura 31**Error! Reference source not found.**). Permite a exportação da lista de filtros e dos *datasets* que correspondem à filtragem.

Apresenta um mapa onde é possível visualizar a rota percorrida em cada *dataset*, ao clicar num dos elementos presentes no mapa (*icon* do carro), é possível visualizar informação obtida num determinado momento da captura.

A visualização dos diferentes momentos da captura pode ser feita movendo o *slider* ou utilizando a opção de *play*, que irá automaticamente demonstrar os trajetos percorridos pelos elementos no mapa durante o período selecionados através dos filtros pré-selecionados. Esta opção de *play* apresenta ainda, várias velocidades para que o utilizador possa selecionar o nível de velocidade/precisão com que deseja visualizar as movimentações dos veículos, e uma opção de pausa para que seja possível analisar um momento específico pelo período que for necessário.

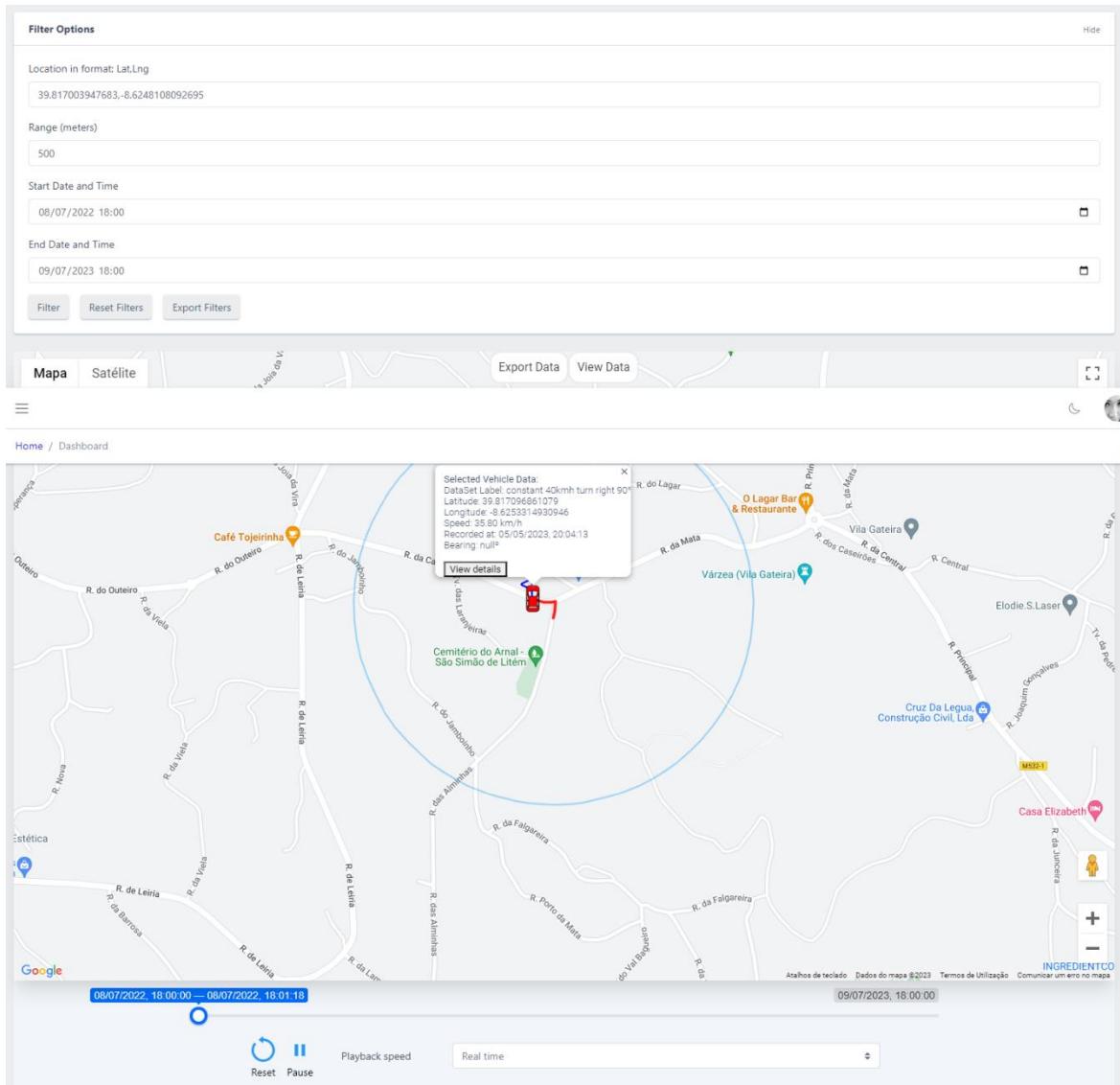


Figura 31 - Dashboard

- **Página de about:** Página que apresenta ao utilizador todos os detalhes relacionados com o projeto e a aplicação (ver Figura 32).



Figura 32 - Página de *about*

7. Testes

Neste capítulo serão abordados os testes efetuados com a aplicação móvel desenvolvida, que deu início a 23 de abril de 2023 e terminou com o fim do desenvolvimento do projeto. É uma fase/capítulo de grande relevância, uma vez que representa uma das três principais fases deste projeto, dado que foi com esta atividade que foi possível tirar as conclusões principais.

7.1. Estratégia

O processo de realização de testes decorreu sempre segundo uma estratégia com três fases. Uma primeira fase, em que decorreu uma discussão em conjunto com os docentes orientadores, de forma a determinar qual o teste a ser realizado, quais os seus objetivos e condições a ter em conta.

Uma segunda fase, em que se realizou o teste propriamente dito, em que um elemento conduzia a viatura enquanto o outro elemento estava encarregue de iniciar a captura dos dados na aplicação, apontar informações relevantes e gerir processo e organização desse conjunto de testes. No final de cada série de testes foi sempre elaborada uma análise superficial aos dados, com o objetivo de encontrar anomalias e porventura repetir algum teste.

Por fim, uma terceira fase que, novamente em conjunto com os docentes orientadores, que se deveu à análise dos *datasets* obtidos, a análise de eventuais anomalias nos dados, analisar/entender como os dados dos sensores variam consoante as condições a que foram expostos e analisar com os dados podem eventualmente variar consoante a viatura que realizou o teste.

7.2. Objetivos

O objetivo desta fase do projeto passa por entender, com a realização de testes, como os valores reproduzidos pelos sensores embutidos no dispositivo móvel reagem consoante o ambiente e condições às quais são expostos. Pretende conseguir também analisar se existe algum padrão consoante os comportamentos manifestados durante a condução de uma viatura. É desta forma considerado o objetivo principal a análise do valor que os dados que os sensores reproduzem durante a condução de uma viatura representam.

7.3. Considerações

Antes de iniciar a fase de testes, foi considerado que seria interessante realizar cada teste com duas viaturas, ao invés de apenas com uma viatura. Isto porque, dado que estas viaturas apresentam características diferentes, também iriam apresentar variações diferentes nos dados capturados pelos sensores do dispositivo móvel e que, portanto, iria enriquecer os resultados obtidos nesta fase. Para este efeito, foram utilizadas as seguintes viaturas: **BMW e36 318tds** (1995) e **Honda Jazz 1.2 LS COOL** (2006), com as características apresentadas na Tabela 2.

Tabela 2 - Características das viaturas utilizadas

| Características | Viaturas | BMW e36 | Honda Jazz |
|--|----------|----------------------|----------------------|
| Sistema de ABS | | Não tem | Tem |
| Cilindrada | | 1665 cm ³ | 1246 cm ³ |
| Potencia máxima | | 66 Kw | 57 Kw |
| Tipo de combustível | | Gasóleo | Gasolina |
| Regime nominal | | 4400 min-1 | 5700 min-1 |
| Regime do motor | | 2000 min-1 | 4275 min-1 |
| Tara/Peso(vazio) | | 1265 kg | 1139 kg |
| Distancia entre eixos | | 2700 mm | 2450 mm |
| Suspensão (observado com base na condução) | | Mole | Dura |

Com o objetivo de assegurar a qualidade das capturas de dados durante os testes, foram tidos em consideração vários aspetos. Foi tido em atenção colocar o dispositivo móvel sempre no mesmo sítio em todos os testes efetuados (ver Figura 33). Como se verifica, o dispositivo móvel foi colocado no ponto central da viatura, na posição e orientação mais indicada, devidamente acomodado com fita-cola para evitar eventuais deslocações do mesmo.

Foi tido em atenção em usar sempre o mesmo dispositivo móvel para a realização dos testes, assegurar que todas as funcionalidades no dispositivo móvel se mantinham, por exemplo, assegurar que a funcionalidade GPS estava ativada e que a ligação *wifi* estava desativada. Foi necessário também assegurar que a aplicação instalada se mantinha com as

definições estabelecidas desde o primeiro teste efetuado, definições estas como: velocidade de captura e tipo de filtro ativado. Assegurar que dentro de um subconjunto de testes, estes seriam efetuados aproximadamente nas mesmas condições. E por fim, sempre que possível, tentar que os testes fossem efetuados nas mesmas condições climatéricas, dando mais relevância às condições do piso.



Figura 33 - Posicionamento do dispositivo móvel durante os testes

Apesar de todos os testes terem sido efetuados na via pública, a segurança foi considerada prioritária, não colocando em perigo a nossa segurança, nem a dos outros condutores ou cidadãos, evitando situações de risco, procurando sítios e horas com a menor circulação possível e respeitando o código da estrada.

7.4. Tipos de testes

Foram realizados sete tipos de testes, sendo que cada tipo de teste é composto por um conjunto de variações, por exemplo, ser realizado a velocidades diferentes, com comportamentos de condução diferentes, ser realizado dentro de uma distância específica, entre outros.

No total foram realizados cerca de 283 testes individuais, cerca de 141 testes únicos por cada viatura. Contudo, devido a problemas mecânicos numa das viaturas não foi possível realizar todos os testes do tipo “travagem antes de cruzamento” (pertencentes aos testes de cruzamento). Além destes testes, também foram realizados numa fase inicial e depois de

forma rotineira um conjunto de testes, com o intuito de entender o comportamento dos sensores e detetar anomalias na aplicação.

Abaixo encontram-se todos os tipos de testes realizados:

- **Testes de aceleração:**
 - Aceleração até atingir uma distância: Partindo de uma posição inicial parada, realizar uma aceleração do tipo [agressiva, normal, suave] numa linha reta, sem inclinação, até uma distância de [10m, 20m, 30m, 40m, 50m];
 - Aceleração até atingir velocidade: Partindo de uma posição inicial parada, realizar uma aceleração do tipo [agressiva, normal, suave] numa linha reta, sem inclinação, até perfazer uma velocidade de [10km/h, 20km/h, 30km/h, 40km/h, 50km/h].
- **Testes de aceleração e desaceleração:**
 - Em curva suave: A circular a uma velocidade de 80km/h, numa curva suave, realizar uma travagem do tipo normal, consistente e constante até a viatura parar;
 - Em linha reta: A circular a uma velocidade de 80km/h, em linha reta, realizar uma travagem do tipo normal, consistente e constante até a viatura parar.
- **Testes de travagem**: A circular a uma velocidade de [10km/h, 20km/h, 30km/h, 40km/h, 50km/h], realizar uma travagem do tipo [agressiva, normal, suave] até parar a viatura;
- **Testes de velocidade constante**: Circular a uma velocidade constante de [10km/h, 20km/h, 30km/h, 40km/h, 50km/h], em linha reta, durante uma distância de [50m, 100m];
- **Testes de fricção em linha reta**: Circular a uma velocidade de [10km/h, 20km/h, 30km/h, 40km/h, 50km/h, 60km/h], colocar a caixa de mudanças da viatura no ponto neutro e deixar a viatura parar por si, sem redução de marcha nem travar;
- **Testes de cruzamento**: Realizado numa via com uma saída acentuada com 90° [à esquerda, à direita]:

- Travagem antes de cruzamento: Aproximar a saída com uma velocidade de [10km/h, 20km/h, 30km/h, 40km/h], travar de forma [agressiva, normal, suave] até pouco antes da saída e até a viatura parar. De seguida, acelerar de forma [agressiva, normal, suave] e completar o percurso;
 - Saída do cruzamento com velocidade constante (sem parar): A saída do cruzamento é efetuada sem parar ou abrandar, com uma velocidade de [10km/h, 20km/h, 30km/h, 40km/h].
- **Testes em rotundas**: Circular com uma velocidade constante de [10km/h, 20km/h, 30km/h, 40km/h], entrar na rotunda e sair na [1º saída, 2º saída, 3º saída, 4º saída]:
 - Rotunda pequena: Diâmetro do centro de 34.4 metros, com três vias de circulação e quatro saídas;
 - Rotunda média: Diâmetro do centro de 24.5 metros, com 1 via de circulação e três saídas;
 - Rotunda grande: Diâmetro do centro de 13 metros, com 1 via de circulação e três saídas.
 - **Outros testes**: Testes realizados numa fase inicial, com o intuito de entender o comportamento dos sensores e detetar anomalias na aplicação:
 - Testes de aceleração;
 - Testes de travagens;
 - Testes de viagens citadinas curtas;
 - Testes de viagens de maior duração (cerca de 1 hora);
 - Testes de guinadas: para esquerda e para a direita;
 - Testes em dois tipos de pavimento: asfalto e calçada.

O aspeto interessante desta variedade de testes passa por poder analisar um conjunto de vários fatores em várias condições.

7.5. Descrição dos testes

Nos testes apresentados adiante nesta secção, como mencionado anteriormente e visível na Figura 33, o dispositivo móvel encontra-se posicionado no ponto central da viatura, na posição horizontal e com o topo do dispositivo a apontar para a frente da viatura. Com o dispositivo móvel nesta posição, é esperado que a leitura dos eixos do sensor de acelerómetro

sejam os apresentados na Figura 34 e que os do sensor de giroscópio sejam os apresentados na Figura 35.

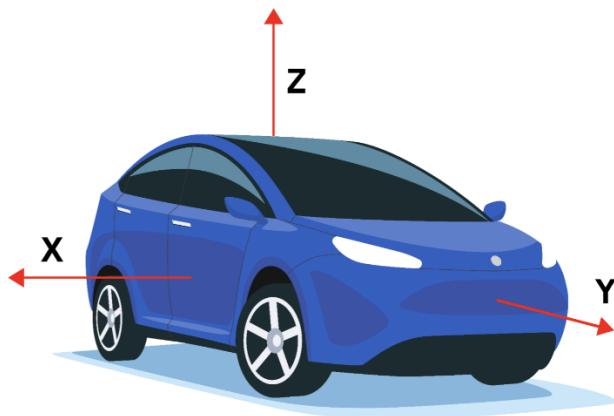


Figura 34 - Eixos de leitura do sensor acelerómetro (retirado de [31])

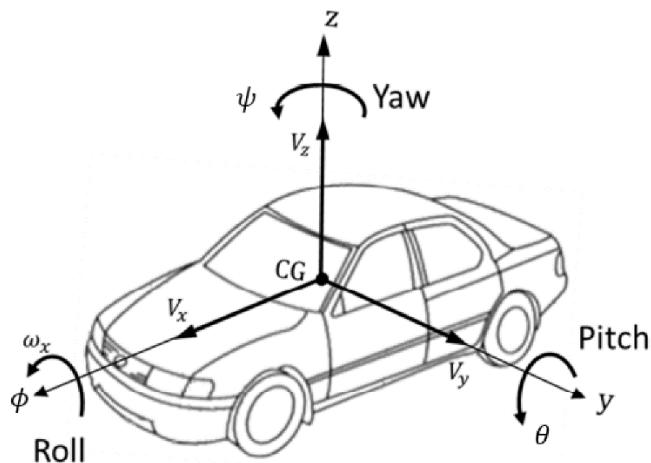


Figura 35 - Eixos de leitura do sensor giroscópio (retirado de [32])

Os testes adiante apresentados são uma pequena porção do total de testes realizados (referido anteriormente), contudo, o conjunto de todos os testes encontra-se junto com a submissão deste relatório.

7.5.1. Testes de aceleração

Estes testes pretendem estudar os dados provenientes dos sensores face ao comportamento de aceleração. Foi referido anteriormente o conjunto de variações efetuadas para este tipo de teste.

Para este tipo de testes foi tido em atenção a escolha de uma localização plana e a direito para garantir a menor interferência possível. É esperado que neste tipo de testes o sensor acelerómetro seja o mais relevante, já que este mede a aceleração linear.

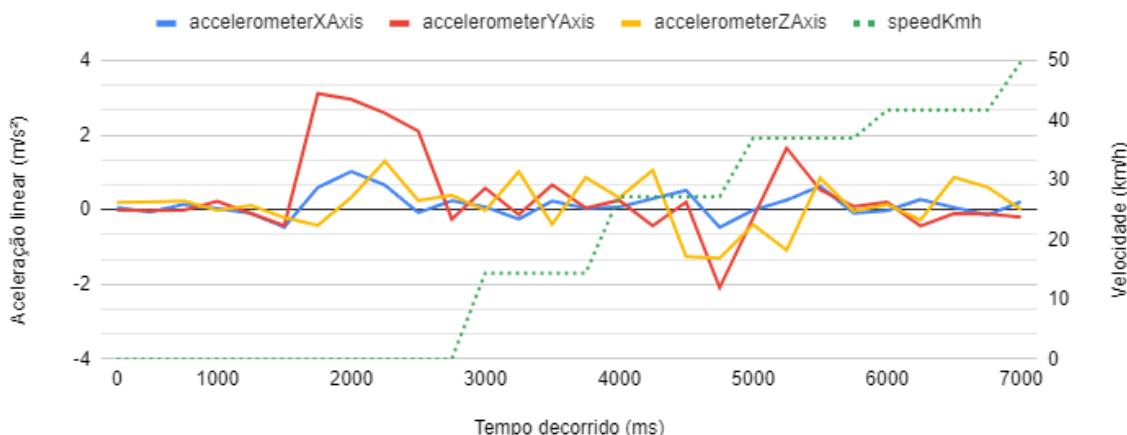
De entre os testes realizados, foi selecionado um teste de aceleração até atingir uma velocidade de 50km/h, realizado pela viatura *Honda Jazz* (ver Tabela 2):

- Dataset “acceleration aggressive 50kmh-28-04-2023-16-53-01”:

Como indica o título, este teste foi realizado até atingindo uma velocidade de 50km/h com uma aceleração agressiva.

Na Figura 36Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica no eixo Y (a vermelho), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas. No gráfico que diz respeito aos dados do sensor de giroscópio, parece haver uma mudança drástica em todos os eixos, contudo, estes valores representam ruído na gravação.

Dados do sensor acelerómetro



Dados do sensor giroscópio

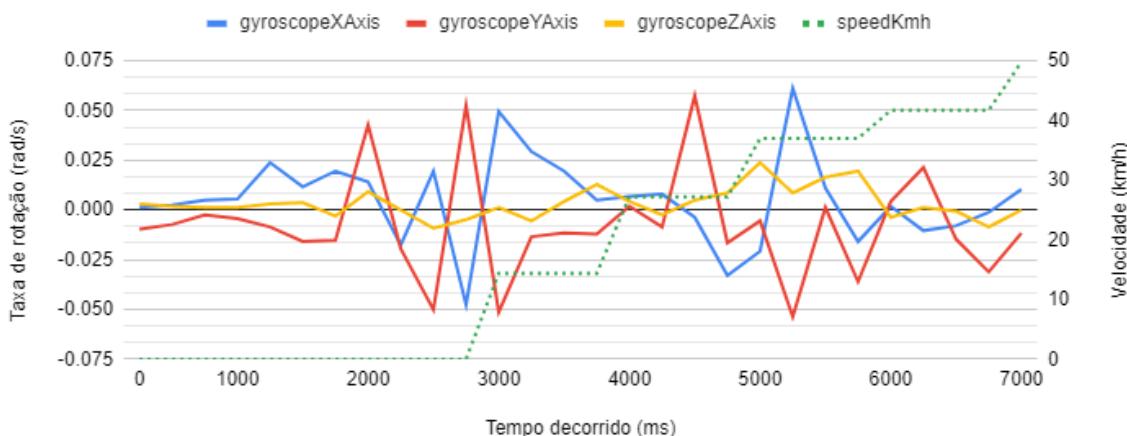


Figura 36 - Gráfico do dataset acceleration aggressive 50kmh-28-04-2023-16-53-01

Em mais detalhe, no gráfico que diz respeito aos dados do sensor de acelerómetro, é observável uma súbita alteração no eixo Y, que se deve à aceleração súbita, depois os valores estabilizam, sendo possível detetar trocas de mudança com descidas de valores neste eixo. É visível uma aceleração agressiva, dado que os valores no eixo Y aumentam subitamente para um patamar alto.

É também de notar que os valores apresentados nos gráficos relativos á velocidade apresentam um atraso na leitura, sendo por isso não tão evidente quando comparado com os valores dos sensores.

7.5.2. Testes de travagem

Estes testes pretendem estudar os dados provenientes dos sensores face ao comportamento de travagem. Foi referido anteriormente o conjunto de variações efetuadas para este tipo de teste.

Para este tipo de testes foi tido em atenção a escolha de uma localização plana e a direito para garantir a menor interferência possível. É esperado que neste tipo de testes o sensor acelerómetro seja o mais relevante, já que este mede a aceleração linear.

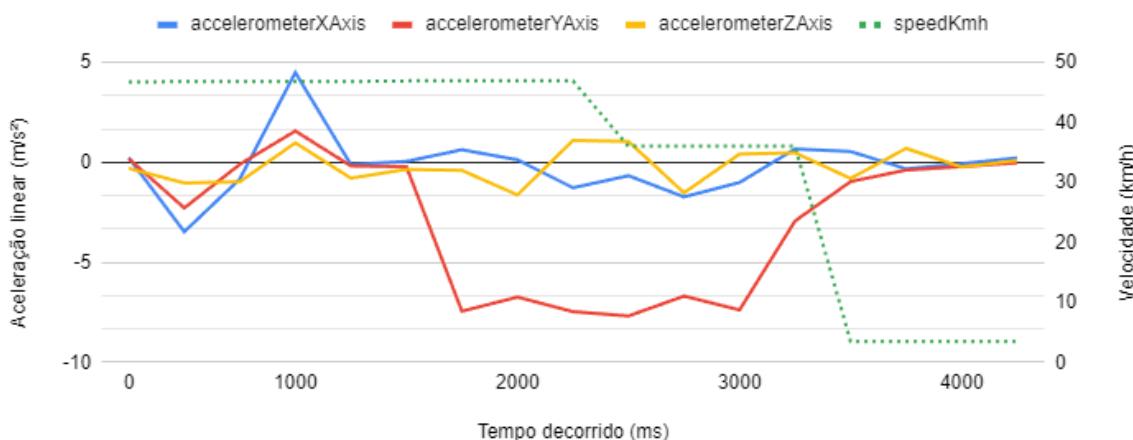
De entre os testes realizados, foram selecionados dois testes do mesmo tipo, mas com viaturas diferentes (Tabela 2):

- Dataset “brake aggressive 50kmh-26-04-2023-13-06-03” com Honda Jazz:

Como indica o título, este teste foi realizado a uma velocidade de 50km/h, quando se deu uma travagem agressiva.

Na Figura 37Figura 36Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica no eixo Y (a vermelho), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas. No gráfico que diz respeito aos dados do sensor de giroscópio, parece haver uma mudança drástica em todos os eixos, contudo, estes valores representam ruído na gravação.

Dados do sensor acelerómetro



Dados do sensor giroscópio

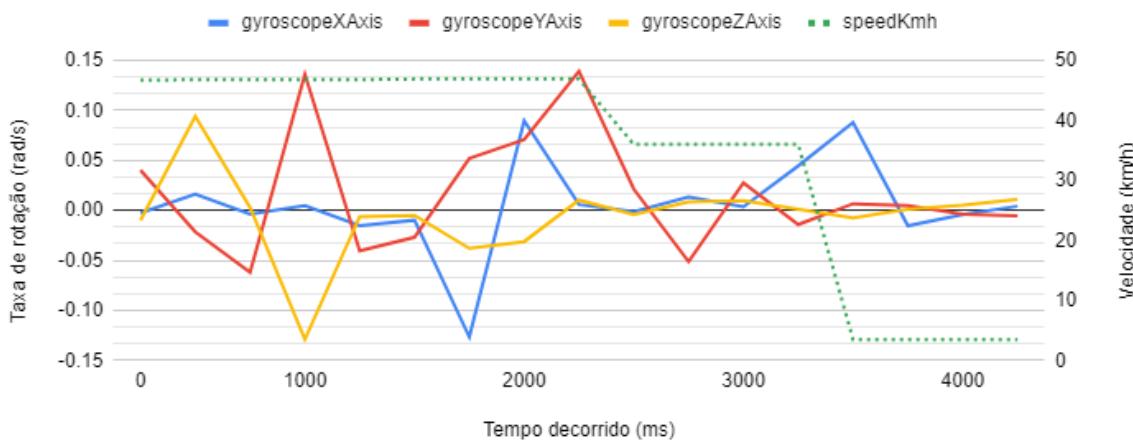


Figura 37 - Gráfico do dataset brake aggressive 50kmh-26-04-2023-13-06-03

É observável uma súbita alteração no eixo Y, que se deve à travagem súbita, depois os valores estabilizam, sendo possível presumir que o sistema de ABS (*Antilock Braking System*) foi ativado, porque a travagem é intervalada, apesar de o pedal do travão ter sido pressionado ao seu máximo. É visível uma travagem agressiva, dado que os valores no eixo Y descem subitamente para um patamar muito baixo.

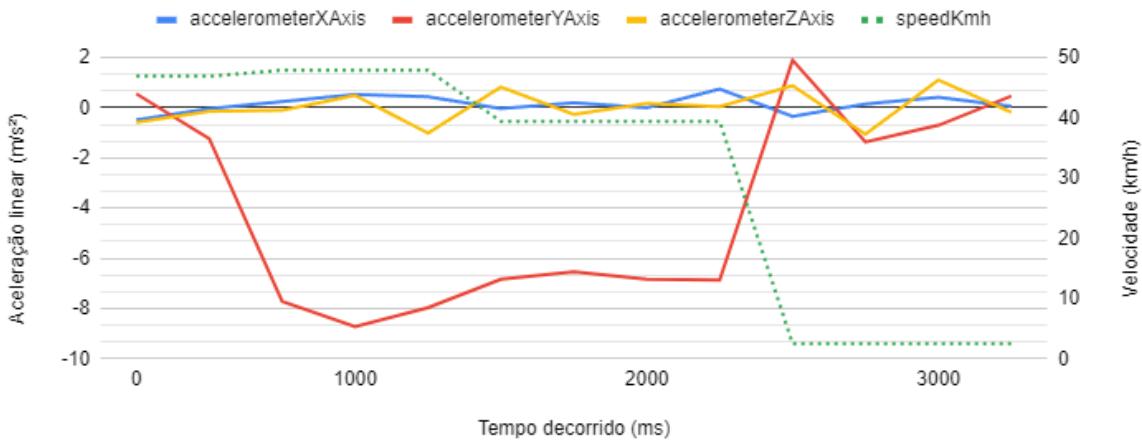
- Dataset “brake aggressive 50kmh-26-04-2023-13-34-51” com BMW e36:

Como indica o título, este teste foi realizado a uma velocidade de 50km/h, quando se deu uma travagem agressiva.

Na Figura 38Figura 37Figura 36Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica no eixo Y

(a vermelho), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas. No gráfico que diz respeito aos dados do sensor de giroscópio, parece haver uma mudança drástica em todos os eixos, contudo, estes valores representam ruído na gravação.

Dados do sensor acelerómetro



Dados do sensor giroscópio

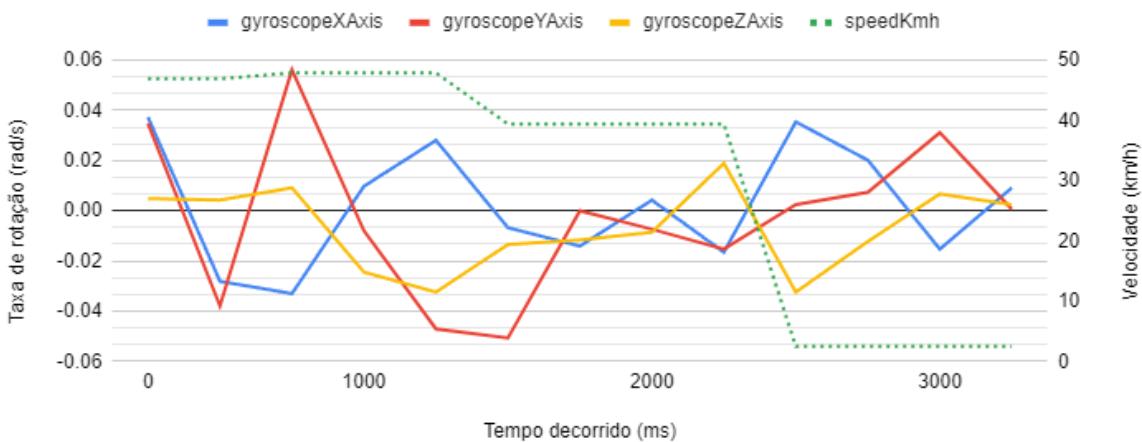


Figura 38 - Gráfico do dataset brake aggressive 50kmh-26-04-2023-13-34-51

É observável uma súbita alteração no eixo Y, que se deve à travagem súbita, depois os valores estabilizam, sendo possível presumir que as rodas da viatura bloquearam (por não ter sistema ABS), porque a travagem é de certa forma contínua. Incluído outras características da viatura, esta apresentou valor máximos superiores ao da viatura *Honda Jazz*. É visível uma travagem agressiva, dado que os valores no eixo Y descem subitamente para um patamar muito baixo.

7.5.3. Testes de cruzamento

Estes testes foram realizados numa via com uma curva/saída de ângulo 90° e pretendem estudar os dados provenientes dos sensores face ao comportamento de giroscópio. Foi referido anteriormente o conjunto de variações efetuadas para este tipo de teste.

É esperado que neste tipo de testes o sensor giroscópio seja o mais relevante, já que este mede a aceleração linear.

7.5.3.1. Travagem antes de cruzamento

Este conjunto de testes foi realizado na localidade de Arnal (junto à Igreja de São Simão de Litém), no dia 5 de maio de 2023. Na Figura 39 encontram-se os percursos realizados, como também algumas informações adicionais no canto inferior direito.



Figura 39 - Percursos e informação dos testes de travagem antes de cruzamento

De entre os testes realizados, foi selecionado um *dataset*, realizados pela viatura *BMW e36* (ver Tabela 2), que representa uma aceleração seguida de uma mudança de direção:

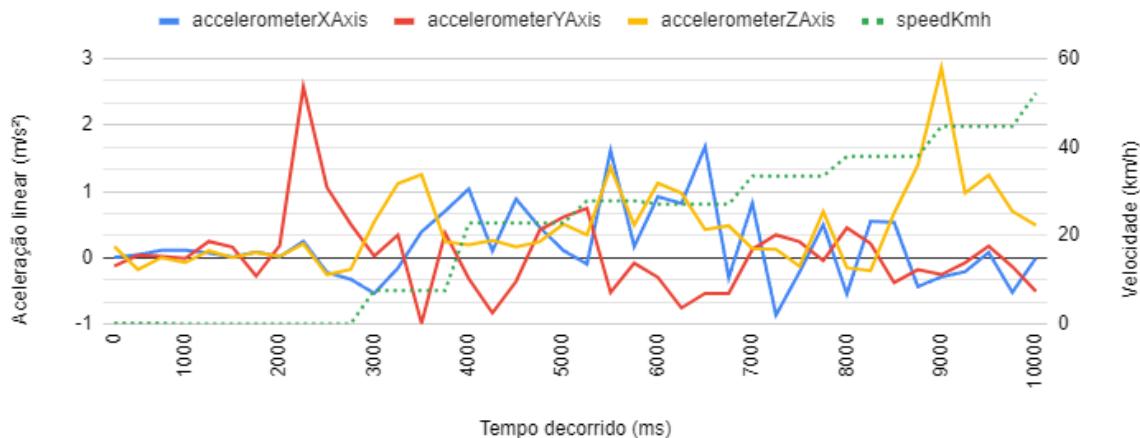
- Dataset “*acceleration aggressive 50kmh turn right 90°-05-05-2023-20-56-08*”:

Como indica o título, este teste foi realizado a partir de uma posição parada, realizando uma aceleração agressiva, mudando de direção à direita, até atingir uma velocidade de 50km/h (cujo percurso se encontra assinalado a vermelho na Figura 39).

Na Figura 40Figura 38Figura 37Figura 36Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica no eixo Y (a vermelho), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas. No gráfico que diz respeito aos dados do sensor de giroscópio, parece haver uma mudança drástica em todos os eixos, contudo, estes valores representam ruído na gravação.

Neste gráfico é possível verificar uma mudança drástica no eixo Y do sensor de giroscópio (a amarelo), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas.

Dados do sensor acelerómetro



Dados do sensor giroscópio

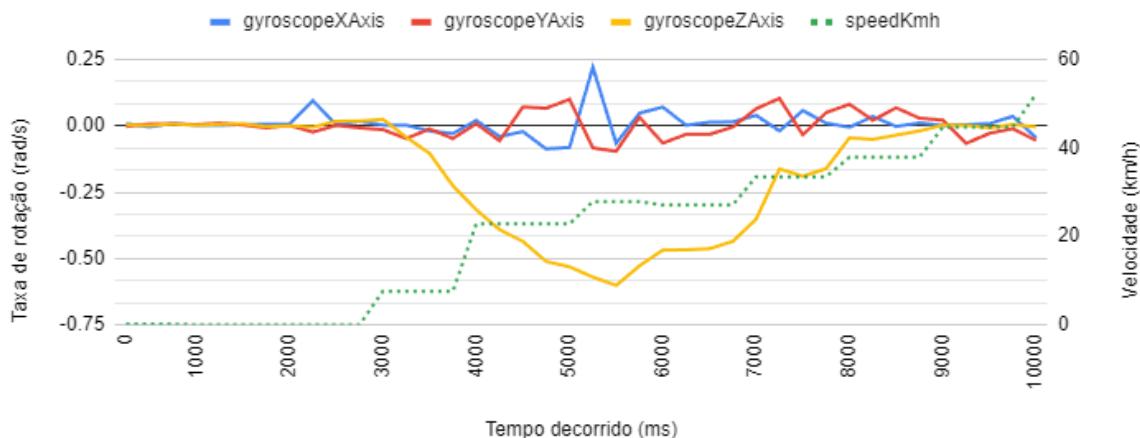


Figura 40 - Gráfico do dataset acceleration aggressive 50kmh turn right 90°-05-05-2023-20-56-08

Em mais detalhe é observável uma súbita alteração no eixo Z, que se deve à súbita mudança de direção, depois os valores estabilizam. É visível que este comportamento é agressivo, dado que diminuem subitamente para um patamar muito baixo.

7.5.3.2. Saída do cruzamento com velocidade constante (sem parar)

Este conjunto de testes foi realizado na localidade de Arnal (junto à Igreja de São Simão de Litém), no dia 5 de maio de 2023. Na Figura 41 encontram-se os percursos realizados, como também algumas informações adicionais no canto inferior direito.



Figura 41 - Percursos e informação dos testes de saída do cruzamento com velocidade constante (sem parar)

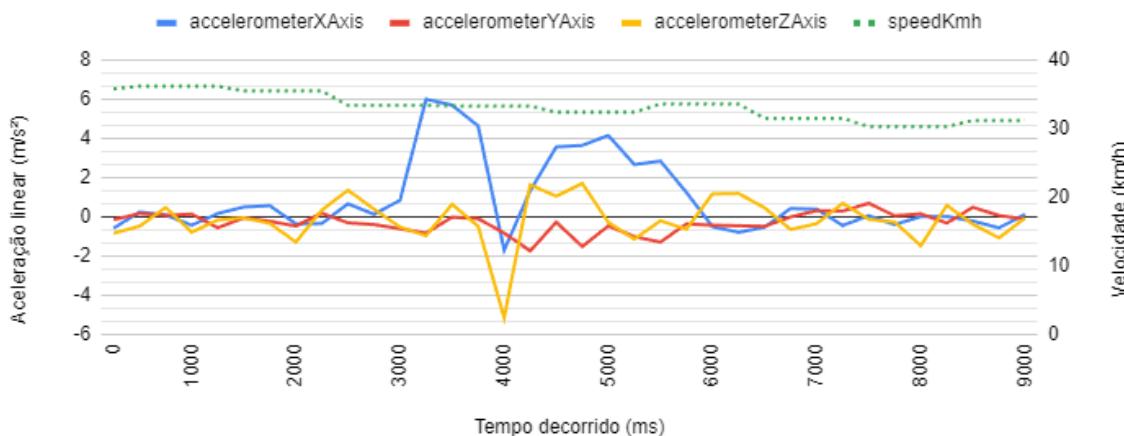
De entre os testes realizados, foi selecionado um *dataset*, realizado pela viatura *Honda Jazz* (Tabela 2):

- Dataset “constant 40kmh turn right 90°-05-05-2023-20-04-13” com Honda Jazz:

Como indica o título, este teste foi realizado a uma velocidade de 40km/h, quando se deu mudança de direção à direita (cujo percurso se encontra assinalado a vermelho na Figura 41).

Na Figura 42Figura 40Figura 38Figura 37Figura 36Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica no eixo X (a azul) e no eixo Z (a amarelo), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas. No gráfico que diz respeito aos dados do sensor de giroscópio, parece haver uma mudança drástica no eixo Z (a amarelo), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas.

Dados do sensor acelerómetro



Dados do sensor giroscópio

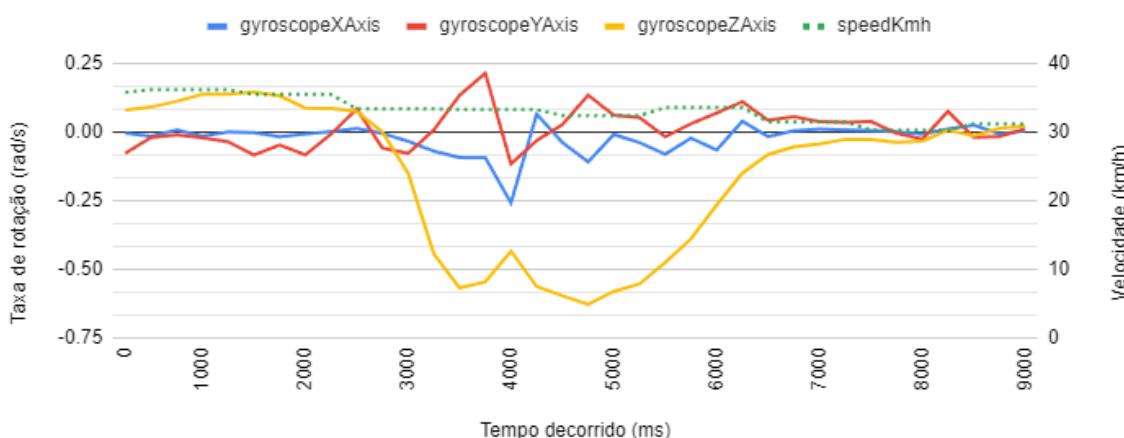


Figura 42 - Gráfico do dataset constant 40kmh turn right 90°-05-05-2023-20-04-13

Em mais detalhe é observável uma súbita alteração no eixo Z do sensor giroscópio, que se deve à súbita mudança de direção, depois os valores estabilizam. É visível que este comportamento é agressivo, dado que diminuem subitamente para um patamar muito baixo.

7.5.4. Testes em rotundas

Estes testes pretendem estudar os dados provenientes dos sensores face à condução de uma viatura em três rotundas de dimensões diferentes: grande, pequena e média. Além das variações de dimensão das rotundas, foi estudado o comportamento da circulação dentro de uma rotunda a várias velocidades e variando a saída escolhida.

É esperado que neste tipo de testes o sensor giroscópio seja o mais relevante, já que este mede rotações do dispositivo móvel.

Neste tipo de testes foi tido em consideração a correta circulação dentro de rotundas, como consta no Código da Estrada.

7.5.4.1. Rotunda pequena

Este conjunto de testes foi realizado na zona de Capuchos (junto ao edifício Convento de Santo António dos Capuchos), na cidade de Leiria, no dia 27 de maio de 2023. Na Figura 43 encontram-se os percursos realizados, como também algumas informações adicionais no canto inferior direito.

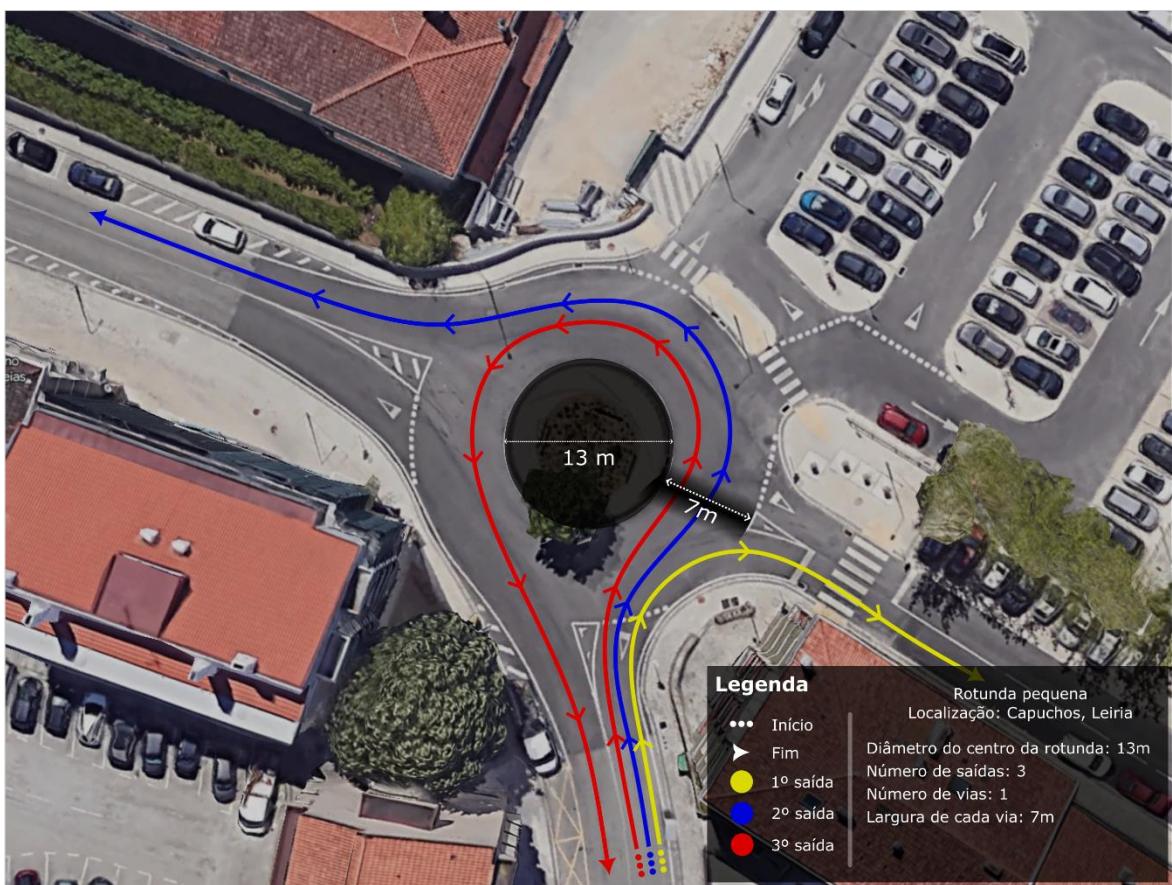


Figura 43 - Percursos e informação dos testes de rotunda pequena

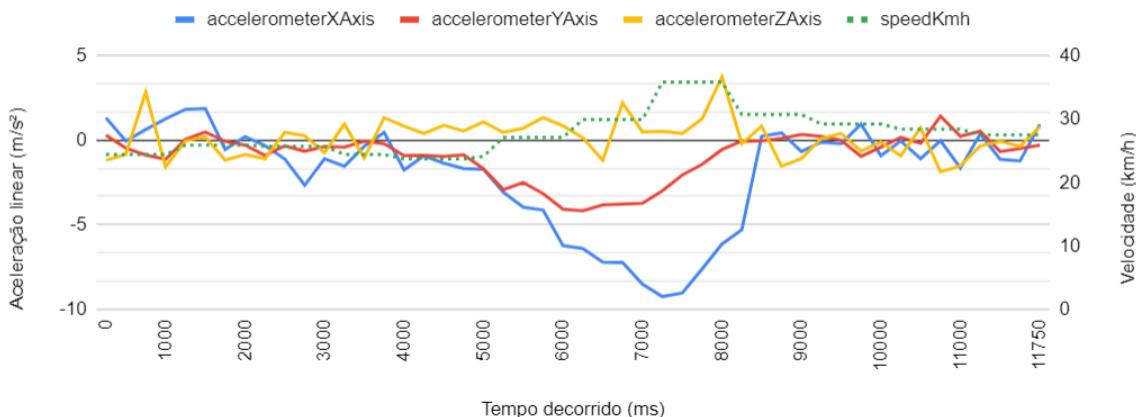
De entre os testes realizados, foi analisado um *dataset*, realizado pela viatura *Honda Jazz* (ver Tabela 2):

- Dataset “30kmh small roundabout 3rd exit-27-05-2023-12-25-28”:

Como indica o título, este teste foi realizado a uma velocidade de 30km/h, contornando a rotunda e saindo na terceira saída (cujo percurso se encontra assinalado a vermelho na Figura 43).

Na Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica em todos os eixos. No gráfico que diz respeito aos dados do sensor de giroscópio, é possível verificar uma mudança drástica no eixo Z (a amarelo), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas.

Dados do sensor acelerómetro



Dados do sensor giroscópio

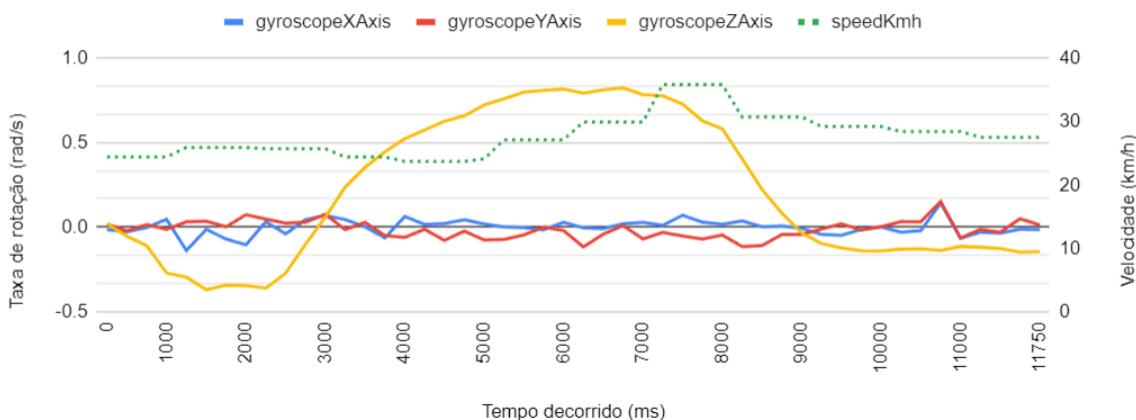


Figura 44 - Gráficos do dataset 30kmh small roundabout 3rd exit-27-05-2023-12-25-28

Em mais detalhe, no gráfico que diz respeito aos dados do sensor de acelerómetro, é possível observar que existe uma grande variação dos dados, em todos os eixos, que se apresenta no intervalo de tempo decorrido entre os 3500ms e os 8000ms aproximadamente. Desta grande variação é possível concluir o seguinte:

- O valor do eixo X sofreu uma variação para valores negativos bastante elevada, o que é justificável pelo facto de esta rotunda pequena ser sido contornada com 30km/h;

- O valor do eixo Y sofreu uma variação para valores negativos, o que é justificável pelo facto de esta rotunda pequena ser sido contornada com 30km/h;
- O eixo Z sofreu uma variação para valores positivos que se transmite nas forças aplicadas no momento do contornar da rotunda;

Em mais detalhe, no gráfico que diz respeito aos dados do sensor de giroscópio, é possível observar que existem duas grandes variações do eixo Z, que divide visualmente o gráfico em três intervalos:

- Intervalo de tempo decorrido entre os 0ms e os 3000ms aproximadamente:
Representa a entrada na rotunda (uma ligeira curva à direita), que neste eixo apresenta valores negativos;
- Intervalo de tempo decorrido entre os 3000ms e os 9000ms aproximadamente:
Representa o contorno da rotunda, realizando uma curva á esquerda, que neste caso apresenta valores bastante elevados;
- Intervalo de tempo decorrido entre os 9000ms e os 11750ms aproximadamente:
Representa a saída na rotunda (uma curva ligeira à direita), que neste eixo apresenta valores negativos.

É possível concluir que esta manobra foi realizada com um comportamento agressivo, dado o eixo X do sensor de acelerómetro e o eixo Z do sensor de giroscópio sofreram uma variação com valores bastante altos.

7.5.4.2. Rotunda média

Este conjunto de testes foi realizado na localidade de Arnal (junto café/bar “O Lagar Bar & Restaurante”), do distrito de Leiria, no dia 17 de maio de 2023. Na Figura 45 encontram-se os percursos realizados, tanto como algumas informações adicionais no canto inferior direito.

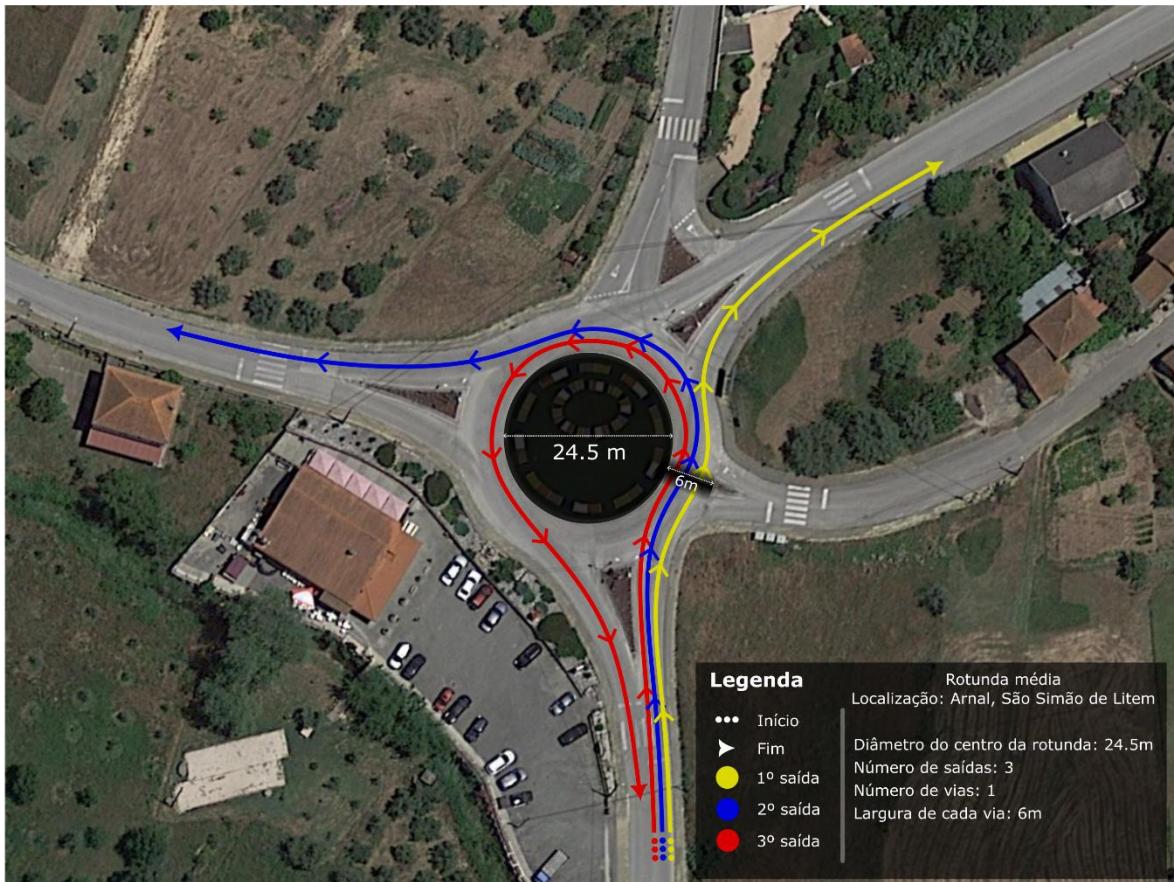


Figura 45 - Percursos e informação dos testes de rotunda média

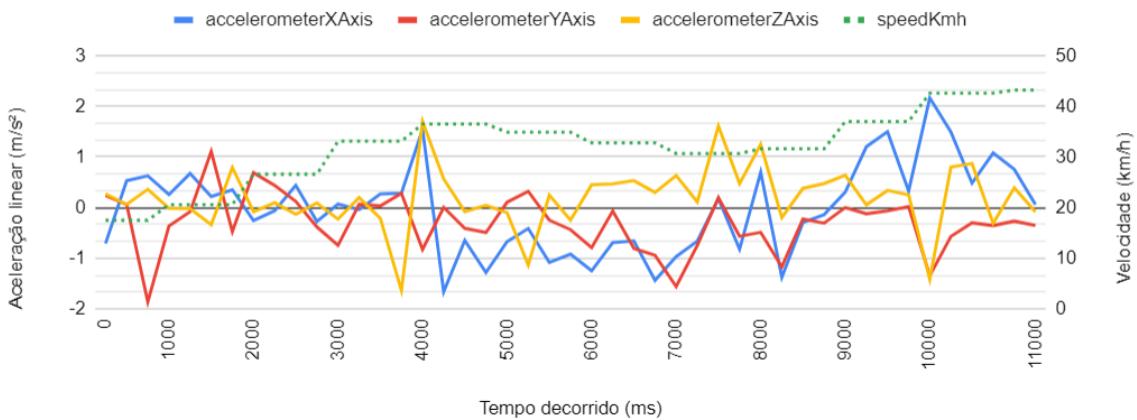
De entre os testes realizados, foi selecionado um *dataset*, realizado pela viatura *BMW e36* (Tabela 2):

- Dataset “40kmh medium roundabout 2nd exit-17-05-2023-19-36-35”:

Como indica o título, este teste foi realizado a uma velocidade de 40km/h, contornando a rotunda e saindo na segunda saída (cujo percurso se encontra assinalado a azul na Figura 45).

Na Figura 46Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica em todos os eixos. No gráfico que diz respeito aos dados do sensor de giroscópio, é possível verificar uma mudança drástica no eixo Z (a amarelo), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas.

Dados do sensor acelerómetro



Dados do sensor giroscópio

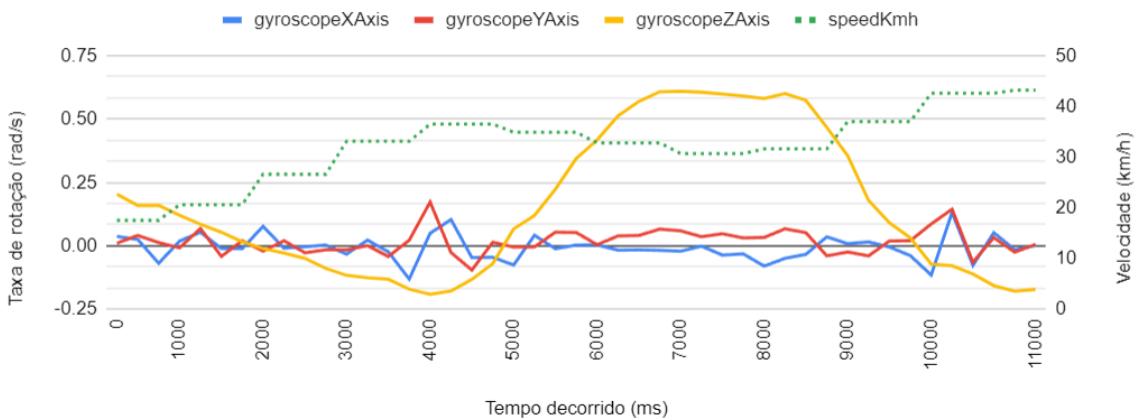


Figura 46 - Gráficos do dataset 40kmh medium roundabout 2nd exit-17-05-2023-19-36-35

Em mais detalhe, no gráfico que diz respeito aos dados do sensor de giroscópio, é possível observar que existem duas grandes variações do eixo Z, que divide visualmente o gráfico em quatro intervalos:

- Intervalo de tempo decorrido entre os 0ms e os 2000ms aproximadamente:
Representa o trajeto antes de iniciar a entrada na rotunda (visível na Figura 45);
- Intervalo de tempo decorrido entre os 2000ms e os 5000ms aproximadamente:
Representa a entrada na rotunda (uma leve curva à direita), que neste eixo apresenta valores negativos;
- Intervalo de tempo decorrido entre os 5000ms e os 10000ms aproximadamente:
Representa o contorno da rotunda, realizando uma curva à esquerda, que neste caso apresenta valores bastante elevados. É possível também visualizar um momento em que os dados são constantes, isto significa que o contornar da rotunda foi realizado sem grandes oscilações;

- Intervalo de tempo decorrido entre os 10000ms e os 11000ms aproximadamente:
Representa a saída na rotunda (uma curva ligeira à direita), que neste eixo apresenta valores negativos.

Em mais detalhe é também possível observar que a entrada na rotunda foi realizada com aproximadamente 35km/h, no delinear da rotunda esta velocidade oscila devido à tentativa de manter uma velocidade constante. Este comportamento é traduzido nos valores que o sensor de acelerómetro apresenta.

É possível concluir que esta manobra foi realizada com um comportamento agressivo, dado os eixos do sensor de acelerómetro e o eixo Z do sensor de giroscópio sofreram uma variação com valores bastante altos.

7.5.4.3. Rotunda grande

Este conjunto de testes foi realizado perto da “Startup Leiria”, na cidade de Leiria, no dia 27 de maio de 2023. Abaixo, na Figura 47, encontram-se os percursos realizados, como também algumas informações adicionais no canto inferior direito.

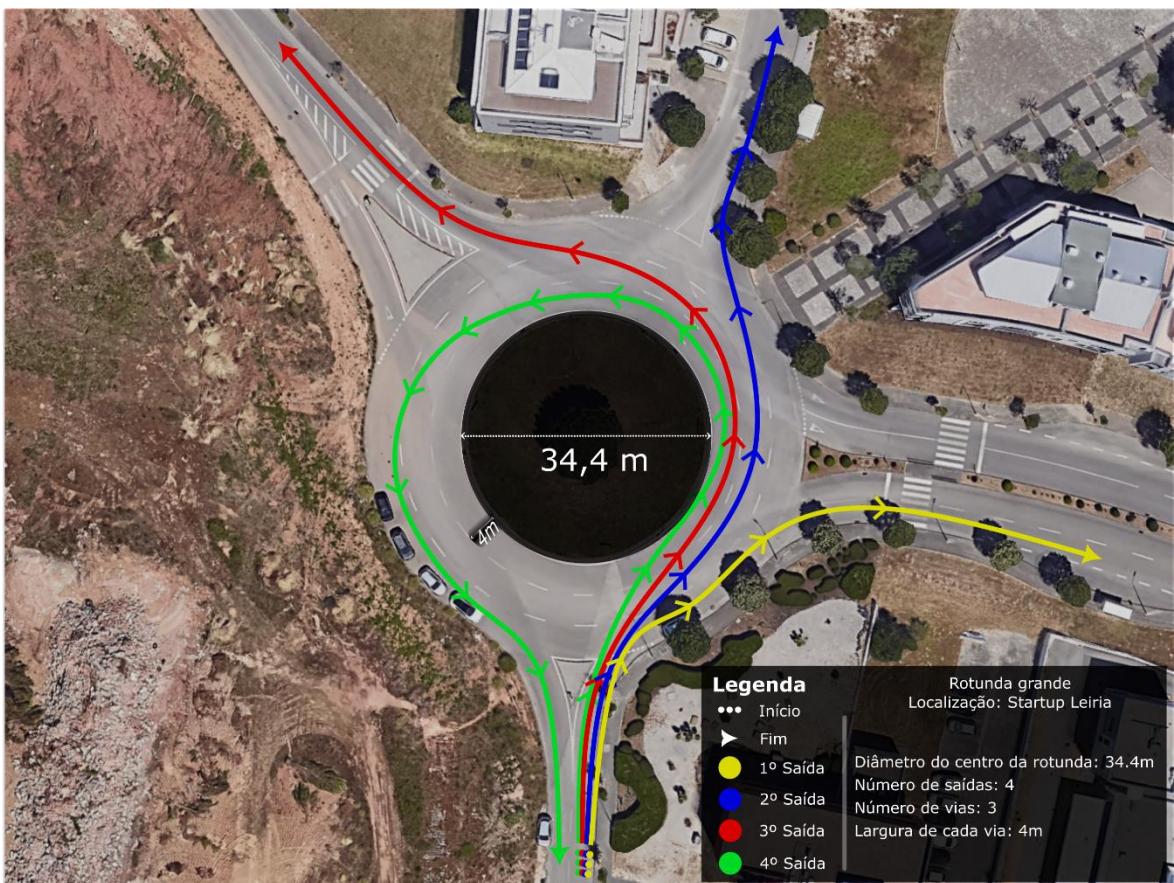


Figura 47 - Percursos e informação dos testes de rotunda grande

De entre os testes realizados, foi selecionado um *dataset*, realizado pela viatura *Honda Jazz* (ver Tabela 2):

- Dataset “40kmh big roundabout 4th exit-27-05-2023-11-19-43”:

Como indica o título, este teste foi realizado a uma velocidade de 40km/h, contornando a rotunda e saindo na quarta saída (cujo percurso se encontra assinalado a verde na Figura 47).

Na Figura 48Figura 46Figura 44 são apresentados os gráficos correspondentes aos dados dos sensores de acelerómetro e giroscópio do *dataset*. No gráfico que diz respeito aos dados do sensor de acelerómetro, é possível verificar uma mudança drástica em todos os eixos. No gráfico que diz respeito aos dados do sensor de giroscópio, é possível verificar uma mudança drástica no eixo Z (a amarelo), enquanto os restantes eixos apesar de sofrerem variações, não foram tão drásticas.

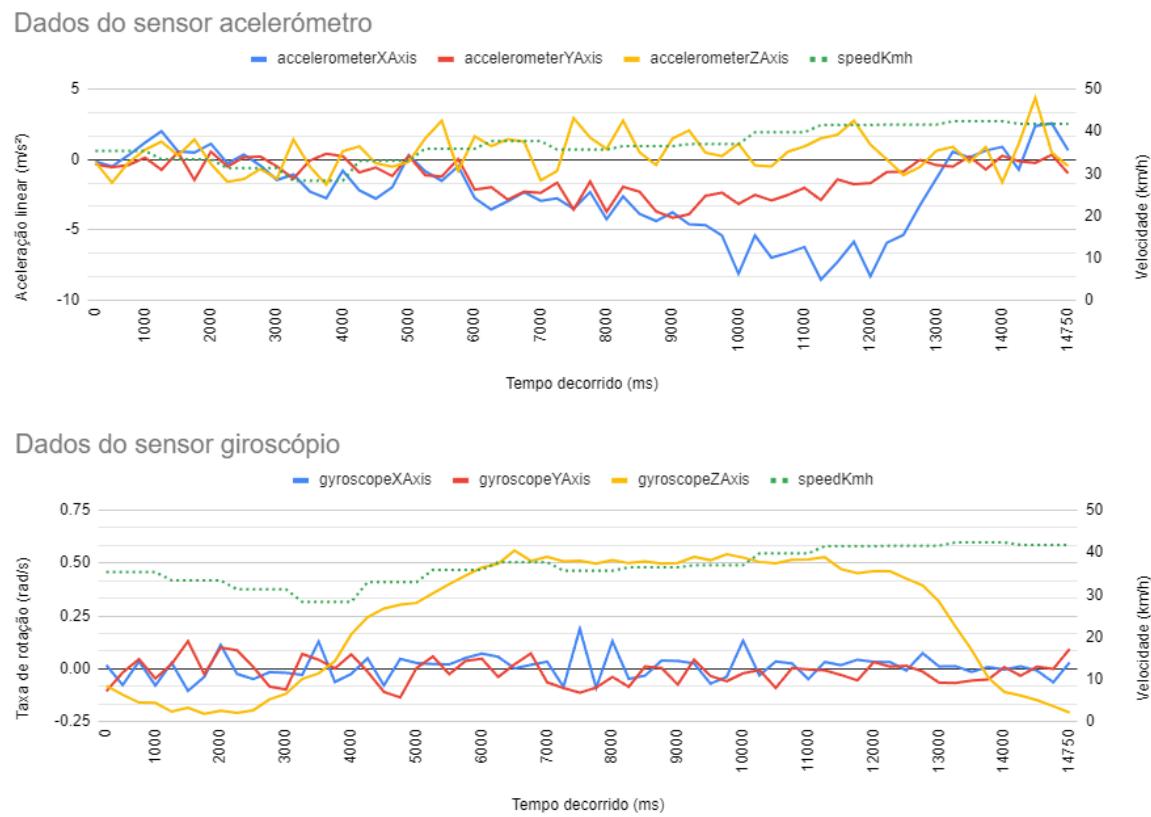


Figura 48 - Gráfico do dataset 40kmh big roundabout 4th exit-27-05-2023-11-19-43

Em mais detalhe, no gráfico que diz respeito aos dados do sensor de giroscópio, é possível observar que existem duas grandes variações do eixo Z, que divide visualmente o gráfico em três intervalos:

- Intervalo de tempo decorrido entre os 0ms e os 3800ms aproximadamente:
Representa a entrada na rotunda (uma leve curva à direita), que neste eixo apresenta valores negativos;
- Intervalo de tempo decorrido entre os 3800ms e os 13800ms aproximadamente:
Representa o contorno da rotunda, realizando uma curva à esquerda, que neste caso apresenta valores bastante elevados. É possível também visualizar um momento em que os dados são constantes, isto significa que o contornar da rotunda foi realizado sem grandes oscilações;
- Intervalo de tempo decorrido entre os 13800ms e os 14750ms aproximadamente:
Representa a saída na rotunda (uma leve curva à direita), que neste eixo apresenta valores negativos.

8. Instalação/configuração

Neste capítulo pretende-se fornecer informação que permita colocar os vários elementos do projeto em execução em ambiente local. Na Secção 8.1.1 é apresentado um conjunto de informação e comandos necessários para correr a aplicação *web*, na secção 8.1.2 é apresentado a informação necessária à realização do processo de login e da forma de acesso as restantes funcionalidades da aplicação e por fim na secção 8.2 é descrito o processo de instalação das aplicações moveis através dos vários processos disponíveis.

8.1. Plataforma web

8.1.1. Configuração da plataforma *web*

Para a configuração da plataforma *web* é necessário ter em atenção um conjunto de dependências e ferramentas necessárias antes de realizar a instalação. **Dependências e ferramentas necessárias:**

- *Docker Desktop*: Possível instalar a partir do seguinte *link*, <https://www.docker.com/products/docker-desktop/>;
- *Ubuntu wsl*: Caso o sistema operativo onde será efetuada a configuração seja *Microsoft Windows*, é possível realizar a instalação desta ferramenta via *Microsoft Store* (pelo *link*, <https://apps.microsoft.com/store/detail/ubuntu/9PDXGNCFSCZV>). Esta instalação é necessária para ser possível executar comandos que interajam com o *Docker* usando a linha de comandos do *Ubuntu* no sistema operativo *Microsoft Windows*;
- Versão de *PHP 8.1*: A instalar no sistema operativo *Ubuntu* ou no *Ubuntu wsl*;
- Versão *Python 3.0* ou mais recente;
- *Composer*: A instalar no sistema operativo *Ubuntu* ou no *Ubuntu wsl*;
- *Node.js*: instalar via o *link* <https://nodejs.org/en>.

Após verificar que as versões das dependências coincidem e que as ferramentas se encontram instaladas, é necessário descompactar o ficheiro “.zip” incluído na submissão do projeto para um local a gosto, preferencialmente numa pasta na raiz do sistema operativo, para evitar quaisquer conflitos.

O seguinte conjunto de passos ordenados indica o processo de **configuração para o servidor da plataforma web**:

1. Inicializar o *Docker Desktop*;
2. Através do terminal *Ubuntu/Ubuntu wsl*, entrar na pasta onde anteriormente foram descompactados os ficheiros, entrar na pasta “*webapp*” e por fim entrar na pasta “Rastreio de viaturas”;
3. Executar o comando “*composer install*”;
4. Executar o comando “*alias sail='[-f sail] && sh sail || sh vendor/bin/sail'*”;
5. Executar o comando “*sail up -d*” e verificar no *Docker Desktop* que o contentor foi inicializado com sucesso, deverá ter um aspeto semelhante ao da Figura 49;

| NAME | IMAGE | STATUS | PORT(S) | STARTED | ACTIONS |
|--------------------------------|---------------------|---------------|--------------------|--------------|---------|
| rastreiodeviaturas - | | Running (2/2) | | | |
| mysql-1 e61e9fac123c | mysql/mysql-server | Running | 3306:3306 | 1 minute ago | |
| laravel.test-1 d0be9d4c8566 | sail-8.2/app:latest | Running | 5173:5173 80:80 | 1 minute ago | |

Figura 49 - Contentor “rastreiodeviaturas” a correr

6. Executar o comando “*php artisan migrate:fresh*”;
7. Executar o comando “*php artisan db:seed*”.

Tendo os passos anteriores sido executados com sucesso, significa que agora o servidor está operacional, falta, contudo, configurar a componente *frontend*. O seguinte conjunto de passos ordenados indica o processo de **configuração para a componente *frontend***:

1. Através do terminal *Ubuntu/Ubuntu wsl*, voltar para a pasta onde foram descompactados os ficheiros, voltar a abrir a pasta “*webapp*” e de seguida abrir a pasta “*coreui*”;
2. Executar o comando “*npm install*”;
3. Executar o comando “*npm run serve*” e após breves instantes deverá ser apresentada a mensagem de sucesso “*DONE Compiled successfully (...)*”, algo semelhante ao apresentado na Figura 50.

```
DONE | Compiled successfully in 15582ms

App running at:
- Local: http://localhost:8080/
- Network: http://192.168.1.77:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

Figura 50 - Comando *npm run serve* executado com sucesso

Para iniciar o servidor e a componente *frontend* do zero após ter a configuração, basta:

1. Inicializar o *Docker Desktop*;
2. Através do terminal *Ubuntu/Ubuntu wsl*, entrar na pasta onde anteriormente foram descompactados os ficheiros, entrar na pasta “*webapp*”:
 - a. Entrar na pasta “*Rastreio de viaturas*”, executar o comando “*sail up*” ou “*sail up -d*” e verificar que no *Docker Desktop* o contentor se encontra a correr, como se apresenta na Figura 49.
 - b. Entrar na pasta “*coreui*”, executar o comando “*npm run serve*” e verificar que se apresenta uma mensagem semelhante à da Figura 50.

8.1.2. Utilização

Após ter o contentor “rastreiodeviaturas” a correr no *Docker Desktop* (como na Figura 49) a correr e ter a componente *frontend coreui* a correr (como na Figura 50), basta no browser inserir o *url* que se encontra no momento de iniciar a componente *frontend* (ver Figura 50) denominado por “Local” e será apresentada o painel de login (ver Figura 51). Para efetuar o login e aceder à restante plataforma *web* são precisas credenciais, podendo ser utilizadas as seguintes credenciais:

Username: admin@admin.com

Password: password

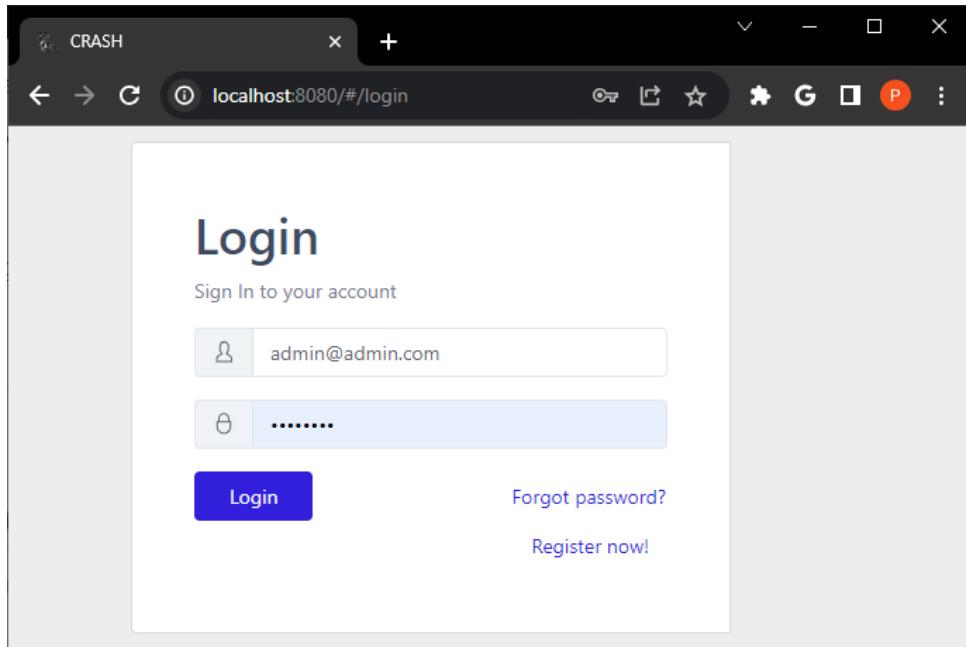


Figura 51 - Painel login

8.2. Aplicação móvel *Android*

Nota importante: Para ter acesso a todas as funcionalidades da aplicação móvel, é necessário ter o contentor “*rastreiodeviaturas*” no *Docker Desktop* a correr (configuração encontra-se na Secção 8.1.1**Error! Reference source not found.**), para assim a aplicação conseguir comunicar com o servidor.

8.2.1. Executar a aplicação no dispositivo móvel

Para qualquer uma das aplicações com extensão a “.apk”, basta copiá-las para a memória do dispositivo móvel e instalar (poderá ser necessário permitir a instalação de aplicações desconhecidas nas definições do dispositivo).

8.2.2. Executar a aplicação a partir do código fonte submetido

Para analisar e/ou executar qualquer uma das aplicações móveis, é necessário ter em atenção um conjunto de dependências e ferramentas necessárias:

- Instalar o *Android Studio*: via <https://developer.android.com/studio>;
- Versão *Gradle 7.5*;
- Versão igual ou superior do *SDK 21 (Android 5.0 Lollipop)*

Após verificar que as versões das dependências coincidem e que as ferramentas se encontram instaladas, é necessário descompactar o ficheiro “.zip” incluído na submissão do

projeto para um local a gosto, preferencialmente numa pasta na raiz do sistema operativo, para evitar quaisquer conflitos.

Por fim, basta seguir os seguintes passos para poder executar a aplicação no *Android Studio*:

1. Abrir o *Android Studio*;
2. Abrir a pasta “androidadminapp” ou “androidclientapp” (como indicado na Figura 52);
3. Aguardar pela sincronização das dependências.
4. Carregar no botão “Run ‘app’” (como na Figura 53) e aguardar que o emulador inicie.

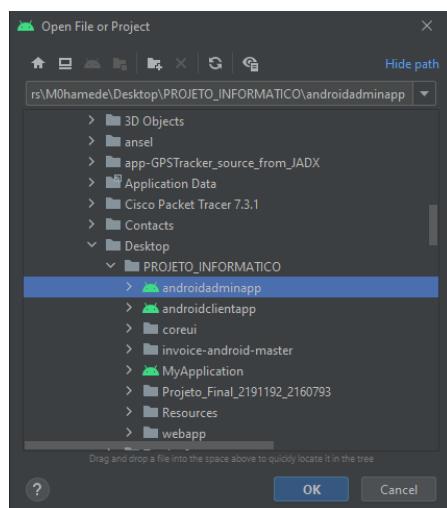


Figura 52 - Selecionar pasta da aplicação móvel

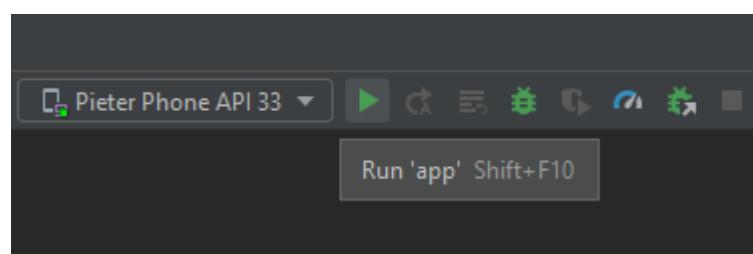


Figura 53 - Executar aplicação no emulador

9. Conclusões e trabalho futuro

Neste capítulo pretende-se recapitular os principais pontos abordados ao longo do projeto, destacar as descobertas e os resultados obtidos e propor pesquisas futuras que possam ser exploradas com base nos resultados/descobertas obtidos no estudo atual.

Na Secção 9.1 é feito uma abordagem do trabalho desenvolvido, os objetivos alcançados e o estado final do projeto, na secção 9.2**Error! Reference source not found.** é abordado algumas das funcionalidades extra que gostaríamos de ter implementado nesta versão do projeto e que podem vir a ser um implementadas em projetos/versões futuras.

9.1. Conclusões

Com a conclusão desta fase de desenvolvimento do projeto, desenvolvemos duas aplicações móveis desenvolvidas, de raiz, com sucesso. Estas duas aplicações permitiram-nos a realização da componente de testes e a sua respetiva análise, o que permitiu chegar ao objetivo principal deste projeto. Por sua vez, a plataforma web desenvolvida permitiu-nos entender como os dados GPS podem modelar um trajeto e dessa forma entender como é possível analisar um eventual acidente rodoviário.

A fase de testes realizada foi especialmente interessante dado que permitiu interagir com as componentes anteriormente desenvolvidas, possibilitou também abstrair do desenvolvimento uma aplicação/solução e passar a analisar os dados provenientes dos sensores embutidos no dispositivo móvel.

Durante o desenvolvimento deste projeto, foi possível obter informação valiosa ao explorar os temas de comportamentos ao volante, segurança automóvel, padrões de condução e inclusive deteção de danos no veículo tendo sido na sua totalidade uma experiência interessante e enriquecedora.

Um dos principais resultados alcançados foi o entender dos valores dos sensores utilizados no projeto. Foi possível compreender como esses valores modelam a condução de um veículo e como podem ser utilizados para identificar comportamentos específicos a partir dos dados capturados. Além disso, ficou claro, como esses valores dos sensores podem ter outros propósitos além da condução, expandindo o potencial de aplicação dos mesmos.

Este projeto ajudou a compreender o valor dos dados dos sensores, assim como, entender como o do ambiente, as condições e o comportamento durante os testes afetam os valores dos sensores. A partir dessas informações, foi possível realizar uma análise aos dados e identificar padrões, viabilizando a deteção de comportamentos de condução.

Este projeto proporcionou informação valiosa sobre a importância dos dados dos sensores, o impacto real do projeto e a capacidade de identificar padrões nos comportamentos de condução. Estas descobertas contribuem para o desenvolvimento de sistemas de deteção de acidentes mais avançado, sistemas de segurança mais eficientes, ajudando na prevenção de acidentes rodoviários, na melhoria da segurança no trânsito e na deteção de avarias nos vários sistemas dos veículos.

A estratégia de trabalho em grupo permitiu alcançar os objetivos de forma rápida e eficaz, assim como também permitiu ultrapassar as dificuldades confrontadas. O trabalho em grupo permitiu a partilha de ideias, por vezes o uso do processo de *brainstorming*, comparação de várias soluções para uma implementação, divisão de tarefas de forma estratégica e possibilitou uma gestão e planeamento do projeto bem estruturada.

Com o decorrer do projeto foi possível aplicar bastantes conhecimentos adquiridos em outras unidades curriculares do curso, conhecimentos como o desenvolvimento de aplicações *web* com recurso a uma API *RESTful*, uso de ferramentas de controlo de versões, planeamento e gestão de projetos, entre outros. Além do uso destas ferramentas já conhecidas e dominadas, a exploração de novas tecnologias e ferramentas provou ser uma experiência interessante. Foi explorado o funcionamento dos sensores embutidos no dispositivo móvel, o uso de ferramentas de visualização interativa de dados, o uso de novas ferramentas de gestão de base de dados, entre outros.

9.2. Trabalho futuro

Foram identificadas algumas funcionalidades que podem enriquecer ainda mais este projeto:

9.2.1. Aplicação *Android*:

- Editar uma rota (trocar nome associado a rota, data...);
- Agrupar as rotas por grupo (Ex: guardar vários registos de viagens à Figueira da Foz);

- Realizar o *plot* de gráficos que representem os dados em tempo real dos sensores;
- Apresentar dados estatísticos para os *datasets*;
- Importar *datasets*(rotas) para o dispositivo móvel;
- Visualizar rotas na aplicação *Android*;
- *Playback* de rotas na aplicação;
- Anúncio de voz (diz os valores atuais de velocidade, localização aproximada);
- Carta dinâmica de velocidade e altitude (gráfico).

9.2.2. Aplicação web:

- Mostrar dados estatísticos referentes aos dados do *dataset*;
- Selecionar *datasets* para visualização;
- Opções avançadas de filtragem na listagem de *datasets*;
- Página de *logs* que regista as entradas de novos *datasets*;
- *Plot* de gráficos com informações de vários elementos em um só gráfico;

Identificámos também que seria interessante usar os *datasets* provenientes dos testes, num algoritmo de classificação, para desta forma conseguir identificar um conjunto de comportamentos.

10. Bibliografia

- [1] “Scrum,” [Online]. Available: <https://www.atlassian.com/agile/scrum>. [Acedido em 03 2023].
- [2] “HTML,” [Online]. Available: <https://en.wikipedia.org/wiki/HTML>. [Acedido em 07 2023].
- [3] “CSS,” [Online]. Available: <https://en.wikipedia.org/wiki/CSS>. [Acedido em 07 2023].
- [4] “PHP,” [Online]. Available: <https://www.php.net>. [Acedido em 06 2023].
- [5] “Java programming language,” [Online]. Available: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)). [Acedido em 07 2023].
- [6] “Laravel,” [Online]. Available: <https://laravel.com>. [Acedido em 06 2023].
- [7] “MVC Laravel,” [Online]. Available: <https://laravel.com/docs/10.x/releases>. [Acedido em 06 2023].
- [8] “VueJS,” [Online]. Available: <https://vuejs.org>. [Acedido em 06 2023].
- [9] “CoreUI,” [Online]. Available: <https://coreui.io>. [Acedido em 06 2023].
- [10] “MySQL,” [Online]. Available: <https://www.mysql.com>. [Acedido em 05 2023].
- [11] “ObjectBox,” [Online]. Available: <https://docs.objectbox.io>. [Acedido em 05 2023].
- [12] “Volley,” [Online]. Available: <https://developer.android.com/training/volley>. [Acedido em 07 2023].

- [13] “Axios,” [Online]. Available: <https://axios-http.com/docs/intro>. [Acedido em 05 2023].
- [14] “Google Maps,” [Online]. Available: <https://www.google.com/maps>. [Acedido em 06 2023].
- [15] “Docker,” [Online]. Available: <https://www.docker.com>. [Acedido em 05 2023].
- [16] “Laravel Sail,” [Online]. Available: <https://laravel.com/docs/10.x/sail>. [Acedido em 05 2023].
- [17] “Postman,” [Online]. Available: <https://www.postman.com>. [Acedido em 05 2023].
- [18] “Bitbucket,” [Online]. Available: <https://bitbucket.org>. [Acedido em 03 2023].
- [19] “Jira,” [Online]. Available: <https://www.atlassian.com/software/jira>. [Acedido em 03 2023].
- [20] “PowerBi,” [Online]. Available: <https://powerbi.microsoft.com>. [Acedido em 04 2023].
- [21] “Inkscape,” [Online]. Available: <https://inkscape.org>. [Acedido em 07 2023].
- [22] “Android Studio,” [Online]. Available: <https://developer.android.com/studio>. [Acedido em 04 2023].
- [23] “Visual Studio Code,” [Online]. Available: <https://code.visualstudio.com>. [Acedido em 05 2023].
- [24] “My Track,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.zihua.android.mytracks>. [Acedido em 03 2023].
- [25] “Strava,” [Online]. Available: <https://www.strava.com>. [Acedido em 03 2023].

- [26] “Sensor Logger,” [Online]. Available: <https://www.tszheichoi.com/sensorlogger>. [Acedido em 03 2023].
- [27] “Sensor Data,” [Online]. Available: <https://play.google.com/store/apps/details?id=com.matlabgeeks.gaitanalyzer>. [Acedido em 03 2023].
- [28] “Android Developer,” [Online]. Available: <https://developer.android.com>. [Acedido em 04 2023].
- [29] “Mobile phone three-axis coordinate system,” ResearchGate, [Online]. Available: https://www.researchgate.net/figure/Mobile-phone-three-axis-coordinate-system_fig1_352259249. [Acedido em 07 2023].
- [30] “Measure rotational speed around X, Y, and Z axes in rad,” MathWorks, [Online]. Available: <https://www.mathworks.com/help/supportpkg/android/ref/gyroscope.html>. [Acedido em 07 2023].
- [31] “Source Android,” [Online]. Available: <https://source.android.com/docs/core/interaction/sensors/sensor-types?hl=pt-br>. [Acedido em 07 2023].
- [32] “ResearchGate,” [Online]. Available: https://www.researchgate.net/figure/Vehicle-Axis-System-ISO-8855-2011_fig1_332561752. [Acedido em 7 2023].
- [33] “Android,” [Online]. Available: <https://www.android.com>. [Acedido em 03 2023].
- [34] “Composer,” [Online]. Available: <https://getcomposer.org>. [Acedido em 05 2023].
- [35] “Google Cloud,” [Online]. Available: <https://console.cloud.google.com>. [Acedido em 06 2023].

- [36] “Google Maps Platform,” [Online]. Available: <https://developers.google.com/maps>. [Acedido em 06 2023].
- [37] “NodeJS,” [Online]. Available: <https://nodejs.org/en/>. [Acedido em 06 2023].
- [38] “Sensores Android,” [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_overview. [Acedido em 04 2023].