

# Aplicativo Móvel para telemetria em ambiente automobilístico

\*Note: Sub-titles are not captured in Xplore and should not be used

Lucas Ferreira  
*Ciência da Computação*  
Centro Universitário IESB  
Brasília - DF, Brasil  
lucas0409lf@gmail.com

Willian Marreiros  
*Ciência da computação*  
Centro Universitário IESB  
Brasília - DF, Brasil  
william.marreiros@gmail.com

Luís Almeida  
*Ciência da Computação*  
Centro Universitário IESB  
Brasília - DF, Brasil  
anhos.luiseduardo@gmail.com

Marcos Sergio  
*Ciência da Computação*  
Centro Universitário IESB  
Brasília - DF, Brasil  
Mstjsousa@gmail.com

O presente trabalho tem por finalidade a criação de um aplicativo, em Flutter, com o intuito de exibir dados telemétricos ao veículo para o usuário, utilizando-se da comunicação com a interface ELM327 Bluetooth conectada a porta On Boarding Diagnostic (OBD2). O projeto é dividido em duas partes, sendo o servidor da aplicação desenvolvido em Java, empregando o paradigma de orientação à objetos e arquitetura MVC (Model, View, Controller). Já a segunda parte consistindo em uma interface gráfica produzida em Flutter, onde são feitas toda a parte visual que entra em contato com o usuário.

## I. INTRODUÇÃO

Dentro da cultura de esportes automobilísticos, um dos principais equipamentos do piloto é o quadro de instrumentos. Em carros de competição, esse quadro possui todas as informações necessárias para a alta performance, porém em provas de menor categoria, onde os carros são modelos comerciais modificados, o quadro de instrumentos nem sempre possui todas as informações importantes e, por muitas vezes, informações irrelevantes ou mal formatadas para o piloto.

Visando sanar esse problema, este trabalho buscará mostrar como funciona a implementação de um aplicativo móvel que apresenta um quadro de instrumentos via comunicação *blue-tooth* aliada com o uso do *scanner* automotivo elm 327 com o protocolo de comunicação OBD II (*Onboard Diagnostics*).

A boa leitura e o bom manejo das informações telemétricas é primordial quando o assunto é alta performance automobilística, porém carros de linha de produção em massa necessitam do custo reduzido.

Com o protocolo OBD II, implementado obrigatoriamente em todos os veículos produzidos na Europa e Estados Unidos da América desde 1996 e desde 2010 em veículos produzidos em território nacional, é possível recuperar quase que todas as informações relevantes para uma boa competição, mas como podemos trazer isso a um baixo custo ao piloto?

Identify applicable funding agency here. If none, delete this.

## II. CONTEXTUALIZAÇÃO

Muitas vezes, quando os componentes de monitoração do veículo em algumas ocasiões acabam nos deixando na mão, principalmente quando ocorre algum defeito no painel do carro, como marcador de combustível, temperatura do motor, medidor de velocidade, temperatura de entrada de ar, entre outros, é interessante termos uma alternativa, visto que a falta dessas informações são prejudiciais e podem levar a defeitos mecânicos, e nos piores casos, a acidentes de trânsito.

## III. PROBLEMA

Implementar uma especificação de rede sem fio de âmbito pessoal (bluetooth) utilizando a dependência flutter bluetooth serial, para que o aplicativo consiga se conectar com o bluetooth do veículo. Sabemos que com essa conexão, conseguiremos receber as informações dos componentes do veículo, mas, será que ao realizar essa conexão, conseguiríamos obter todas as informações em tempo real de forma absoluta?

## IV. OBJETIVO GERAL

- Implementar um aplicativo em plataforma android de telemetria automotiva em tempo real e coletar dados para análise de performance.

## V. OBJETIVOS ESPECÍFICOS

- Implementar um sistema de login e autenticação para armazenamentos dos dados;
- Conectar o dispositivo android ao scanner elm 327;
- Interpretar o protocolo OBD II;
- Apresentar ao usuário dados relevantes durante a prova automobilística;
- Coletar dados durante a apresentação dos mesmos;
- Gerar relatórios com os dados coletados.

## VI. REFERENCIAL TEÓRICO

O Uso da telemetria automotiva, surgiu com a intenção de auxiliar os motoristas a gerenciar o monitoramento dos componentes de seus veículos, como por exemplo, medidor de velocidade, tanque de combustível, carga de ar frio dentre

vários outros. Com o passar dos anos e com o avanço da tecnologia da informação atualmente o motorista pode monitorar seu veículo a partir de seu celular, sendo uma alternativa para quando o painel do carro estiver com defeito, ele saber gerenciar a manutenção dentre várias outras.

A telemetria automotiva, não serve só para o monitoramento do veículo pessoal, mas também como gerencial as informações de uma grande quantidade de veículos, imagine uma pista de corrida de carro, onde uma equipe possa receber em tempo real as informações de todos os veículos que estão na corrida, percebemos que auxiliaria bastante a gestão de uma gama quantidade de veículos.

Em um futuro não tão distante, é possível com que até empresas que tem como foco veículos automotivos consigam gerenciar todos de uma grande quantidade, vamos pegar por exemplo uma transportadora, que consiga ter acesso em tempo real as informações dos componentes de todos os seus caminhões em tempo real, com um sistema de gerenciamento desses aproveitando o bom uso da tecnologia 5G será uma possível implementação real.

## VII. TRABALHOS CORRELATOS

### A. Torque Pro

Torque é uma ferramenta de diagnóstico e performance OBD2 para qualquer dispositivo com sistema operacional Android. Ele permitirá que você acesse os muitos sensores dentro dos Sistema de Gerenciamento do Motor, além de permitir que você visualize e limpe os códigos de problema (TORQUE, 2022). O aplicativo utiliza da conexão com a porta OBD2 através da interface ELM327 Bluetooth, o mesmo consegue exibir ao usuário dados de sensores que não estão disponíveis no painel físico do automóvel. A seguir são apresentadas algumas telas do aplicativo Torque.



Fig. 1. Manômetros que mostram os dados dos sensores.

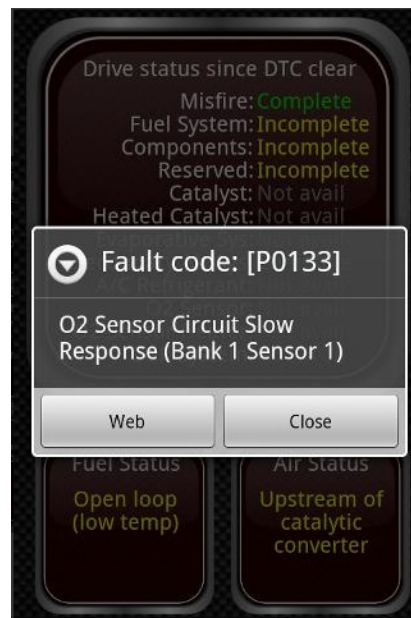


Fig. 2. Essa figura mostra a identificação de um erro apresentado por um sensor.

### B. Kotlin OBD API

É uma API leve e orientada ao desenvolvedor para consultar e analisar comando OBD. Escrito em Kotlin puro e independente de plataforma com uma interface simples e fácil de usar, apresentando diversos recursos como:

- Velocidade do veículo.
- RPM do motor.
- Posição do acelerador.
- Pressão de combustível.
- Avanço de tempo do comando.
- Temperatura da entrada de ar.
- Nível de combustível.
- Taxa de fluxo de ar.

### C. PCAN-OBD-2 Viewer

O PCAN-OBD-2 é um aplicativo que utiliza o OBD2 para diagnósticos de bordo, segundo a Peak Systems a ferramenta foi projetada para fazer sua transmissão utilizando barramento CAN. Tem uma interface que consegue monitorar sensores presentes no sistema de gerenciamento do motor que são obtidos pela PCAN-OBD-2 API, desenvolvida pela mesma empresa.

## VIII. METODOLOGIA

Para implementarmos nosso aplicativo, foi necessário a elaboração de uma conexão via bluetooth que fosse capaz de prover os dados necessários, segurança e também praticidade para o usuário. Tal infraestrutura permite ao usuário, através de um aplicativo mobile, se cadastrar, conectar ao veículo e monitorar um veículo em tempo real, como objeto de aplicação foi utilizado um Volkswagen Gol, ano 2010, modelo Ecomotion de fabricação nacional.



Fig. 3. A figura mostra a tela inicial da aplicação, onde são exibidas informações sobre o automóvel em tempo real.

## IX. RESULTADOS ESPERADOS

É esperado que o aplicativo facilite a vida de pilotos amadores, mas que também ajude, no dia a dia, entusiastas que possuem carros modificados ou que seja uma alternativa para caso a telemetria analógica esteja com defeito físico.

## X. CONCLUSÃO

Com a monitoração em tempo real, foi possível trazer dados relevantes em tempo real do carro em específico. Trazendo mais segurança em altas velocidades e praticidade de leitura dos instrumentos do veículo.

## XI. REFERÊNCIAS BIBLIOGRÁFICAS

Java. Disponível em: <https://docs.oracle.com/en/java> Acessado em: 16/03/2022.

Spring Boot Framework. Disponível em: <https://spring.io/projects/spring-boot> Acessado em: 16/03/2022.

Dart. Disponível em: <https://dart.dev/guides> Acessado em: 16/03/2022.

Flutter. Disponível em: <https://docs.flutter.dev> Acessado em: 16/03/2022.

## XII. CRONOGRAMA

- (13/04/2022) Validação do fluxo de autenticação, temas configurados, navegação entre telas, telas totalmente funcionando, e onboarding do aplicativo.
- (20/04/2022) Entrega de telas de cadastros veiculares vinculados ao usuário, Entrega da navegação entre as telas veiculares e cadastro de veículos.
- (27/04/2022) Entrega de tela de seleção de veículo, Entrega do chatbot,
- (04/05/2022) Conexão com o bluetooth.
- (11/05/2022) Primeiras conexões bluetooth, backend de coleta de dados.

- (18/05/2022) Coleta de dados do bluetooth.
- (25/05/2022) Entrega do chatbot e entra do uso de componente de mapas.
- (01/06/2022) Entregar o projeto final do aplicativo funcionando.
- (08/06/2022 - 15/06/2022) Apresentação do projeto final completamente funcionando, aplicado todos os padrões de arquitetura (MVVM), e com os princípios SOLID.