

**XV Congresso Fluminense de Iniciação Científica e Tecnológica**

**28<sup>o</sup>**

Encontro de Iniciação Científica da UENF

**20<sup>o</sup>**

Circuito de Iniciação Científica do IFFluminense

**16<sup>a</sup>**

Jornada de Iniciação Científica da UFF



**U III Congresso Fluminense de Pós-Graduação**

**23<sup>a</sup>**

Mostra de Pós-Graduação da UENF

**8<sup>a</sup>**

Mostra de Pós-Graduação do IFFluminense

**8<sup>a</sup>**

Mostra de Pós-Graduação da UFF

## Sistema de coleta de dados veiculares

*Yan Nascimento Balbi, Fabrício Barros Gonçalves, Paulo Mesquita do Nascimento*

O advento da IoT (*Internet of Things*) e a introdução dos sistemas embarcados dentro de veículos automotores fomentaram o conceito de *crowdsensing*, permitindo a integração do hardware desses veículos com *software*. Desse modo, o trabalho objetiva fazer a coleta de dados veiculares usando um aplicativo mobile, através linguagem de programação Dart e com o Framework Flutter, com o intuito de fazer a comunicação com um dispositivo Bluetooth baseado no Scanner ELM327. Uma forma de saber se a implementação foi feita corretamente é a implementação de Testes Unitários de Software para as funções de conexão Bluetooth e OBD2, além das funções de decodificação e leitura do VIN (*Vehicle Identification Number*) do carro, de modo que seja feita a verificação do funcionamento de cada módulo. A metodologia utilizada ao longo do trabalho se baseou em criar testes unitários para a verificação de falhas. Antes do processo do desenvolvimento dos testes, a classe em análise foi dividida em duas, cada uma exercendo responsabilidades distintas e únicas seguindo a 1<sup>o</sup> regra do SOLID. Os testes unitários foram implementados possibilitando verificar o comportamento das funções sem a necessidade de executar todo o aplicativo. Uma vez que os testes implementam funções específicas, os recursos presentes em um dispositivo *mobile* tais como protocolos de conexão se tornam indisponíveis. Para contornar esse problema, foram desenvolvidos *Mocks*, que simulam os comportamentos que se encontram indisponíveis devido às limitações dos testes unitários, tais como uma requisição de dados. Foram desenvolvidos diversos *Mocks* para que fosse possível testar as funções da classe responsável pela conexão, que dependiam da comunicação via protocolo Bluetooth. Com os *Mocks* já implementados, através da biblioteca *Mockito*, os testes restantes foram desenvolvidos para verificar se as funções implementadas se comportam como o esperado e corrigi-las caso não o façam corretamente. O ciclo se repete até que todos os testes passem. O usos dos Testes Unitários possibilitaram encontrar diversas falhas nas funções da classe de decodificação do VIN. Assim que as falhas foram encontradas, as mesmas foram corrigidas dentro dos próprios testes e reestruturadas as suas lógicas de seus códigos correspondentes de forma correta. Quanto à classe responsável pela comunicação *Bluetooth*, foram implementados 7 testes usando mocks, todos eles passaram sem apresentar falhas, entretanto, devido ao encapsulamento da biblioteca de conexão usada, algumas funções desenvolvidas não puderam ser testadas. É importante observar que se caso o trabalho tivesse sido desenvolvido usando o princípio de TDD (*Test Driven Development*), problemas como a quebra de responsabilidade exclusiva das classes poderiam ser evitados, entretanto, a própria biblioteca de conexão usada aborda os processos de conexão e comunicação via OBD2 conjuntamente. O término do trabalho levou à conclusão que a implementação de testes é fundamental no processo de detecção de falhas, e também na documentação dos métodos implementados corretamente.

*Instituição do Programa de IC, IT ou PG: IFF BOM JESUS*  
*Eixo temático: Ciências Exatas*

ORGANIZAÇÃO E REALIZAÇÃO:



APOIO:



**XU** Congresso  
Fluminense  
de Iniciação  
Científica e Tecnológica

**28<sup>o</sup>**

Encontro de  
Iniciação  
Científica  
da UENF

**20<sup>o</sup>**

Circuito de  
Iniciação  
Científica do  
IFFluminense

**16<sup>a</sup>**

Jornada de  
Iniciação  
Científica  
da UFF



**U III** Congresso  
Fluminense de  
Pós-Graduação

**23<sup>a</sup>**

Mostra de  
Pós-Graduação  
da UENF

**8<sup>a</sup>**

Mostra de  
Pós-Graduação  
do IFFluminense

**8<sup>a</sup>**

Mostra de  
Pós-Graduação  
da UFF

*Fomento da bolsa (quando aplicável): CNPq*

## Vehicle data collection system

*Yan Nascimento Balbi, Fabrício Barros Gonçalves, Paulo Roberto Nascimento Mesquita*

The IoT ascension and the introduction of embedded systems in automotive vehicles promoted the concept of crowdsensing, allowing the integration between vehicle's hardware and software. In this way, this work aims to develop a data collect system using mobile software, based on Dart programming language and Flutter framework, in order to communicate with a ELM 327 based Bluetooth device. A way of verifying the function and methods of connection, OBD2 and VIN (Vehicle Identification Number) decodification implemented within the developed software is applying Unit Tests. The methodology adopted in the presented work was based on creating unit tests for each method used in the connection of the mobile application to verify the existence of bugs and failures. Before the development starting process, it was necessary to divide the analyzed class into two distinct classes, in a way to follow the SOLID rules. The unit tests were implemented facilitating verification of the functions behavior without the need to execute all the system. Once the tests implement specific functions, some mobile features like connection protocols will become unavailable during the testing process. To work around these problems, Mocks was developed to emulate these unavailable features. Several Mocks were implemented to emulate the Bluetooth protocol behaviors in order to perform the unit tests of these functions through the *Mockito* library. With the mocks already implemented, it was possible to finish all the remaining testings. Analyzing the used procedures, it was possible to realize several failures in the VIN decodification methods, and as soon they were found, they were corrected within the tests, in order to restructuring the logic of their corresponding codes correctly. Were implemented 7 tests to Bluetooth functions, in which all were passed without present failures, however, some functions weren't tested due encapsulation restrictions in the library. As for the discussion part, is important to realize that if the presented work had been done using the Test Driven Development (TDD) methodology, some problems such as single responsibility principle wouldn't been broken, however, the used Bluetooth protocol library addressed OBD communication and Bluetooth connection process jointly. At the end of this work led to the conclusion that the Unit Tests development is essential in the bug and failure detecting process.

ORGANIZAÇÃO E REALIZAÇÃO:



APOIO:

