

Universidade do Vale do Rio dos Sinos - Unisinos - Ciências Exatas e Tecnológicas

Desenvolvimento de Aplicações para Dispositivos Móveis

## **NASTABIL: Um Modelo de Aplicativo Para Coleta e Envio de Dados em Veículos**

André Gandolfi<sup>1</sup>

Alex Roehrs<sup>2</sup>

**CONTEXTO:** Nos últimos anos é possível perceber um grande crescimento no uso de dispositivos móveis e acompanhando esta evolução os usuários de veículos demandam por maior conectividade. O aumento da presença da eletrônica embarcada nos veículos permite coletar informações de diversos sistemas do veículo. Coleta de dados juntamente com conectividade permite o monitoramento remoto das informações, ou seja, telemetria. **PROBLEMA:** Monitoramento remoto de veículos é uma prática frequente em países desenvolvidos, todavia o mesmo não acontece em países emergentes. O principal motivo é o elevado custo dos sistemas de telemetria. **SOLUÇÃO:** Este trabalho propõe um modelo de sistema de telemetria simples e de baixo custo. **MÉTODO PROPOSTO:** No protótipo foi utilizado um módulo para coletar dados do veículo através de uma interface CAN (*Controller Area Network*). Um dispositivo móvel com sistema *Android*, por meio de uma conexão *Bluetooth*, recebe estes dados, adiciona informação de geolocalização e acelerômetro cria arquivos com as informações e os envia para um servidor. **CONCLUSÃO:** A análise do protótipo inserido em um cenário mostrou-se viável no estudo do caso aplicado e estimulam o aprofundamento dos estudos sobre sistemas de telemetria utilizando dispositivos móveis para coleta de dados.

Palavras-chave: Veículos. *Bluetooth*. *Android*. CAN. Coleta de dados. Telemetria

---

<sup>1</sup> André Gandolfi, Graduado em Ciências da Computação, UCS. Email: gandolfi10@outlook.com.

<sup>2</sup> Prof. Msc. Alex Roehrs, Mestre em Computação Aplicada com atuação na área de Análise e Desenvolvimento de Aplicações para Dispositivos Móveis. E-mail: alexr@unisinos.br

## 1 INTRODUÇÃO

Com a redução do custo dos dispositivos móveis tornou-se viável usar estes dispositivos como IHM (Interface Homem Máquina). Podemos disponibilizar um *smartphone* para automação residencial, usar um *tablet* fixo em um táxi apresentando um mapa apontando a geolocalização de clientes ou usar um *smartphone* como uma câmera remota que podemos visualizar de qualquer outro dispositivo com conexão à internet. Estes são exemplos de aplicações que os dispositivos móveis não são de uso pessoal, mas utilizados como computadores distribuídos para coleta de dados ou utilizando a tela e alto-falante para informar ao usuário.

Sistemas de coleta de dados em veículos não são algo novo, já sendo muito utilizados em países desenvolvidos, mas estes sistemas são em sua maioria de alto custo, o que dificulta a utilização em países menos desenvolvidos. Um dos fatores que mais encarece um computador automotivo é a conectividade, ou seja, agregar 3G/4G ou wifi (rede local sem fios baseado no padrão IEEE 802.11). Outro fator dispendioso em um computador automotivo é agregar um sistema de navegação por satélite (SOUZA, 2015).

Uma abordagem para conseguirmos um custo viável para o sistema é distribuir as funcionalidades entre dois dispositivos. Um dispositivo que conta com comunicação CAN (módulo CAN) coleta os dados e envia por *Bluetooth* para o *smartphone*. A geolocalização fica a cargo do *smartphone* assim como o envio das informações através da internet.

### 1.1 Motivação

Em geral, sistemas de telemetria veicular são utilizados para os seguintes propósitos:

- Posicionamento em tempo real: possibilita o acompanhamento da posição atual do veículo, permite também verificar se o veículo está na rota correta e se está adiantado ou atrasado.
- Diagnóstico remoto: verificar remotamente variáveis do sistema, com o intuito de diagnosticar uma possível falha ou mal-uso do veículo.

- Análise da condução do veículo: através de sensores de força G, dados de RPM (rotações por minuto), posição do pedal do acelerador e posição do pedal de freio, é possível analisar a maneira que o veículo está sendo conduzido, com isso seria possível punir maus condutores e premiar bons condutores (DE OLIVEIRA, 2016).

Para transferência de dados para o servidor, duas maneiras se destacam, por 3G/4G, e por wifi. A primeira opção permite o recebimento das informações em tempo real, em contrapartida este método tem um custo contínuo referente ao plano de dados da operadora, o qual pode ser significativo para o sistema. A transferência por wifi, é feita quando o veículo está ao alcance de alguma rede que o dispositivo está previamente configurado, apesar de ser mais atrativo financeiramente, este método não permite a análise em tempo real.

A comunicação CAN com o veículo enriquece ainda mais o sistema, possibilitando a coleta de informações de nível do tanque de combustível, posição do pedal do acelerador, posição do pedal de freio, RPM, entre outros.

## **1.2 Contribuições e definição do problema**

Em países desenvolvidos, normalmente o valor de um sistema de telemetria em comparação com o valor do veículo é pouco significativo, sendo assim simples de justificar o uso, além disso alguns países proporcionam aporte financeiro às iniciativas que comprovadamente modernizem o transporte público, muito diferente do que encontramos em países menos desenvolvidos (DAWSON, 2016).

A grande barreira para a utilização massiva deste sistema em países como Índia, China, todos os países latinos e africanos é o custo e o retorno financeiro que o sistema pode proporcionar. Esta proposta visa oferecer uma opção de menor custo do que a maioria dos sistemas oferecidos no mercado, além disso o usuário teria maior liberdade na escolha do dispositivo móvel fundamentado nos recursos que o equipamento oferece e no custo.

Este trabalho apresenta uma alternativa de baixo custo de telemetria veicular utilizando um *tablet* ou *smartphone* para transmitir os dados para um servidor, de modo que posteriormente possam ser analisados.

### 1.3 Questão de pesquisa

Frente aos problemas expostos, busca-se responder à questão de pesquisa definida como base deste trabalho: *“Como seria uma abordagem com um dispositivo móvel utilizado como um computador automotivo para coleta de dados em um veículo?”*

A questão fundamenta o trabalho na busca de uma alternativa aos sistemas de telemetria veiculares de mercado, que em sua maioria utilizam dispendiosos computadores automotivos.

### 1.4 Objetivos

O principal objetivo do estudo e do protótipo é verificar a possibilidade de implementar um sistema de telemetria utilizando um dispositivo móvel, basicamente coletando dados em um veículo e persistindo em um servidor. Como meta secundária, utilizar a tela do dispositivo móvel como IHM para informar ao motorista do veículo a forma que está conduzindo, afim de orientá-lo a conduzir de forma mais suave objetivando maior conforto, segurança e economia de combustível.

## 2 REFERENCIAL TEÓRICO

Para auxiliar na compreensão deste artigo, são abordados nesta seção os seguintes assuntos: Computação Móvel, Internet das Coisas (IoT – *Internet of Things*), *Android*, *Bluetooth*, Sistemas de Tráfego Inteligentes, CAN (*Controller Area Network*), Módulo HC-05 e o Microcontrolador STM32F105.

### 2.1 Computação Móvel

Computação Móvel é um novo paradigma computacional. Considerado a quarta revolução computacional, precede os centros de processamento nos anos sessenta, os terminais dos anos setenta e rede de computadores dos anos oitenta (PROSTT, 2013).

A mobilidade aliada à comunicação sem fio possibilitou o suporte computacional contínuo, a qualquer momento e em qualquer lugar, o que chamamos de Computação Ubíqua (BARBOSA, 2007). Um dos campos de pesquisa que aplica

os conceitos da Computação Móvel é a automotiva, em duas frentes, os sistemas inter-veiculares, e as estradas inteligentes (JUNG, 2005). Este trabalho apresenta um sistema de Computação Móvel inter-veicular para coleta de dados.

## **2.2 Internet das Coisas (IoT – Internet of Things)**

A Internet das Coisas está crescendo rapidamente no âmbito das comunicações sem fio. O conceito é basicamente a presença de computadores distribuídos em torno de nós como sensores, atuadores, *tablets* e outros. (ATZORI, 2010).

O conceito da Internet das Coisas é muito próximo ao conceito de Computação Móvel, o que difere, é que além de comunicação a qualquer momento e em qualquer lugar, agrega o conceito de comunicação com qualquer coisa (MARQUES, 2015).

A grande discussão sobre o assunto Internet das Coisas é o impacto que terá sobre vários aspectos da vida e no comportamento dos usuários. Uma série de aplicações surgem com este conceito, como vida assistida e *e-health* (Saúde Digital), estes são alguns exemplos onde esta ideia irá desempenhar um papel protagonista em um futuro próximo (ATZORI, 2010).

O setor automotivo e o transporte público são áreas que já estão sentindo os impactos da Internet das Coisas. São estas áreas que serão abordadas neste trabalho, o qual apresenta uma abordagem para envio dos valores coletados dos sensores presentes no veículo.

## **2.3 Plataforma *Android***

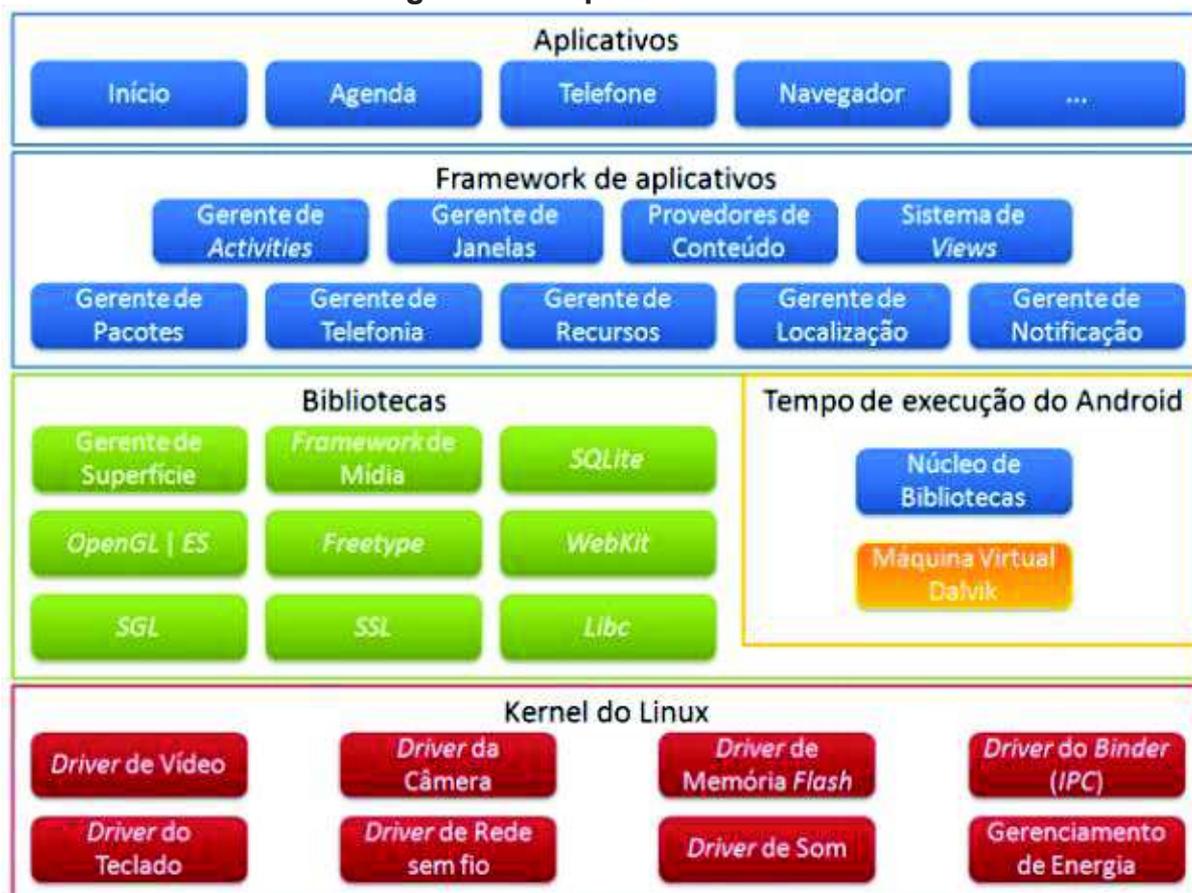
O *Android* é o sistema operacional móvel do Google e atualmente é líder mundial no segmento. Além do Google outros gigantes se juntaram no desenvolvimento de toda a plataforma. Este grupo é chamado de OHA (*Open Handset Alliance*) e entre estes gigantes podemos citar, Samsung, Intel, Sony entre outros (LECHETA, 2015).

A plataforma *Android* é de código aberto, por isso pode ser adaptado pelos fabricantes de dispositivos móveis e pelos desenvolvedores de aplicativos. Esta

plataforma está sempre em evolução já que possui uma comunidade grande, muita ativa e onde frequentemente presenciamos inovações (PEREIRA, 2009).

A figura 1 apresenta a arquitetura do *Android* que está dividida em quatro camadas: *Linux Kernel*, Bibliotecas, Camada de execução, *Framework* e as aplicações.

**Figura 1 – Arquitetura Android**



Fonte [OTSUKA, 2015]

## 2.4 Bluetooth

*Bluetooth* é uma tecnologia muito popular para comunicação sem fio para pequenas distâncias e está presente na imensa maioria dos dispositivos móveis fabricados atualmente. O propósito desta tecnologia é disponibilizar um padrão de comunicação sem fio amigável, para curta distância e de baixo custo (PIETERSE, 2014).

Para que ocorra a conexão, um dispositivo mestre no estado *inquiry* inicia uma procura para detectar dispositivos no raio de cobertura e descobrir os endereços. A próxima etapa é o estado de *page*, onde são enviados pacotes de *paging request* aos dispositivos e aguarda pacotes de *paging response* de dispositivos *slave* que estão no estado *pagescan*. O dispositivo *master* envia um *master response* para iniciar a conexão que poderá ser confirmado pelo dispositivo *slave*. Este processo chamamos de pareamento (VASCONCELOS, 2013).

## 2.5 Sistemas de Tráfego Inteligentes

Sistemas de tráfego inteligentes (ITS) tem o objetivo de melhorar o congestionamento rodoviário e gestão do transporte público, pela análise das informações derivadas de telemetria. A crescente exigência do público por informações sobre o tráfego e horários precisos forçou a implantação de infraestruturas para gerenciamento de tráfego em larga escala. (FRIESEN, 2015).

Para que um sistema de tráfego inteligente funcione, é necessária uma rede de dispositivos que compõem o sistema. Para estes sistemas funcionarem de maneira automática, e que a intervenção humana seja a menor possível é necessário algum tipo de comunicação sem fio (SOUZA, 2015).

## 2.6 CAN (*Controller Area Network*)

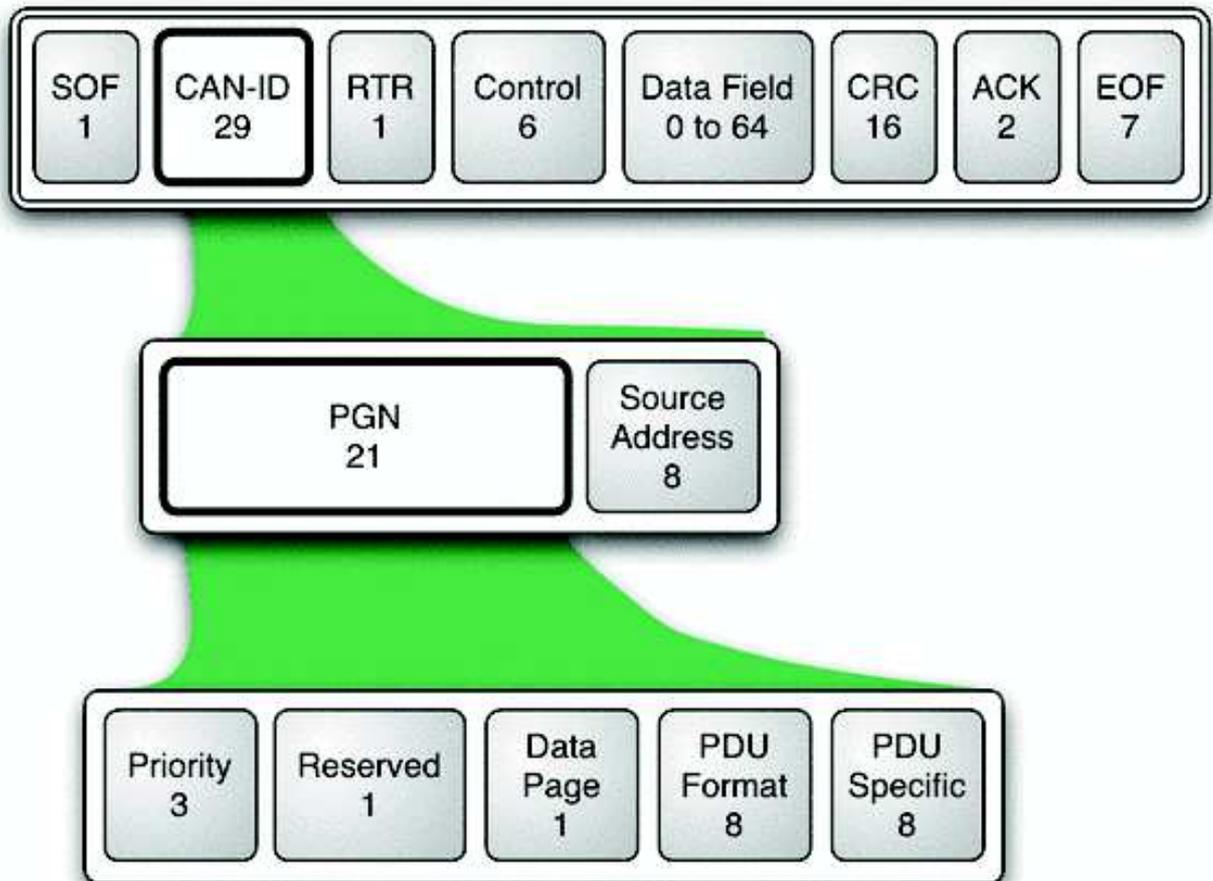
CAN é uma rede inicialmente desenvolvida para interligar componentes de controle de veículos. Com a evolução dos veículos a complexidade dos sistemas eletroeletrônicos embarcados vêm aumentando, e interligar todos os componentes com fios dedicados se tornou inviável. A rede CAN permitiu uma enorme redução no cabeamento, além disso permitiu interligar diversos componentes utilizando apenas um par de fios (DI NATALE, 2012).

A adoção massiva do barramento CAN pela indústria automotiva fomentou a produção de semicondutores para esta tecnologia. Atualmente grandes fabricantes de semicondutores como ST, NXP, Freescale, entre outras desenvolvem e produzem diversas soluções para redes CAN, o que indica que esta tecnologia continuará sendo fortemente adotada (DI NATALE, 2012).

CAN é uma comunicação serial, *multimaster* e protocolo *multicast*, o que significa que a rede está livre para que qualquer módulo envie uma mensagem como *master*. Uma mensagem CAN deve conter 8 bytes de dados, o identificador da mensagem descreve o conteúdo da mensagem, o módulo de origem e o destino na rede. A velocidade de transmissão máxima é 1 Mbit/s, mas quanto maior a velocidade, menor a distância suportada entre os módulos (COOK, 2007). A velocidade mais utilizada para transmissão em veículos é de 250 Kbit/s.

Atualmente nos setores automotivos o protocolo mais utilizado sobre a rede CAN é o J1939. A figura 2 mostra como os bits estão organizados no protocolo J1939.

**Figura 2 – Organização do pacote do protocolo SAE J1939**



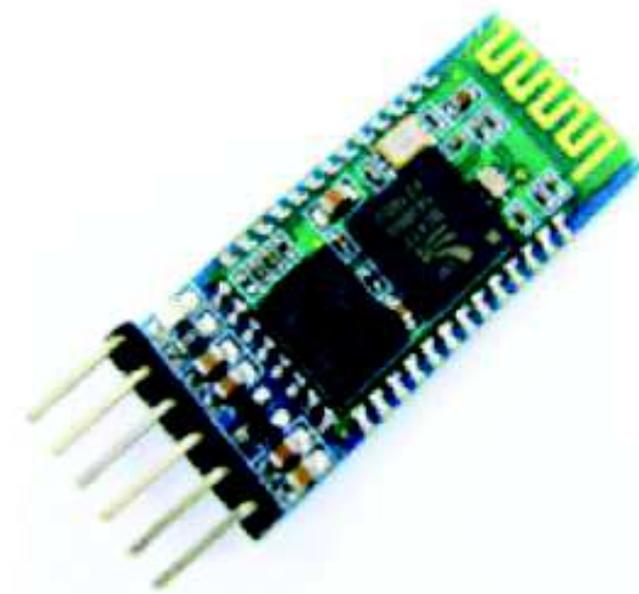
Fonte: [BAXEVANIS, 2012]

Este trabalho coleta parte dos dados em um protocolo J1939 sobre uma rede CAN. Os principais valores a serem lidos na rede CAN são: velocidade, rotações por minuto (RPM), posição do pedal do acelerador e nível de combustível.

## 2.7 Módulo *Bluetooth* HC-05

O HC-05 é um módulo para conversão entre *Bluetooth* e serial nível TTL (*Transistor–transistor logic*). O módulo possui dois modos, um para configuração, o qual é possível enviar comandos AT para alterar parâmetros, e outro modo, automático que utiliza os parâmetros configurados para estabelecer a conexão. (RODRIGUES, 2016). A Figura 3 apresenta o módulo HC-05.

**Figura 3 – Módulo HC-05**



Fonte: [RODRIGUES, 2016]

## 2.8 Microcontrolador STM32F105

O microcontrolador STM32F105, da ST Microelectronics, possui um núcleo Cortex-M3 projetado pela ARM. O conjunto de instruções utilizadas denomina-se Thumb-2, que alia desempenho de 32 bits com tamanho reduzido do arquivo binário. A série STM32F1xx denomina-se *Performance line* e atinge frequências de *clock* de até 72MHz, STM32F105VC possui encapsulamento de 100 pinos, memória *flash* de 256KB e 64KB de memória RAM (DOS REIS, 2011).

Este microcontrolador tem um ótimo custo benefício e se destaca com diversos periféricos, entre eles destaca-se a existência de duas portas CAN, o que torna interessante coletar dados por uma porta CAN, tratar os dados, e enviar pela outra porta CAN.

### 3 TRABALHOS RELACIONADOS

Esta seção descreve quatro trabalhos relacionados à coleta de dados utilizando dispositivos móveis.

#### 3.1 *Development of the Eco-Driving and Safe-Driving Components using Vehicle Information*

O foco deste trabalho é utilizar informações coletadas do veículo para gerar informação de *eco-driving* (economia de combustível) e *safe-driving* (condução segura). A informação do veículo é obtida através de comunicação CAN pela interface OBD (*On-Board Diagnostics*). Estas informações são enviadas para o dispositivo móvel por *Bluetooth*, este por sua vez mostra os resultados na tela do dispositivo *Android* (YUN, 2012).

O objetivo é que os motoristas possam mudar o estilo e comportamento por eles próprios, e checar as informações de economia e direção segura que o aplicativo disponibiliza. A figura 4 mostra as telas principais do aplicativo *Android*.

Figura 4 – Telas do *Eco-driving*



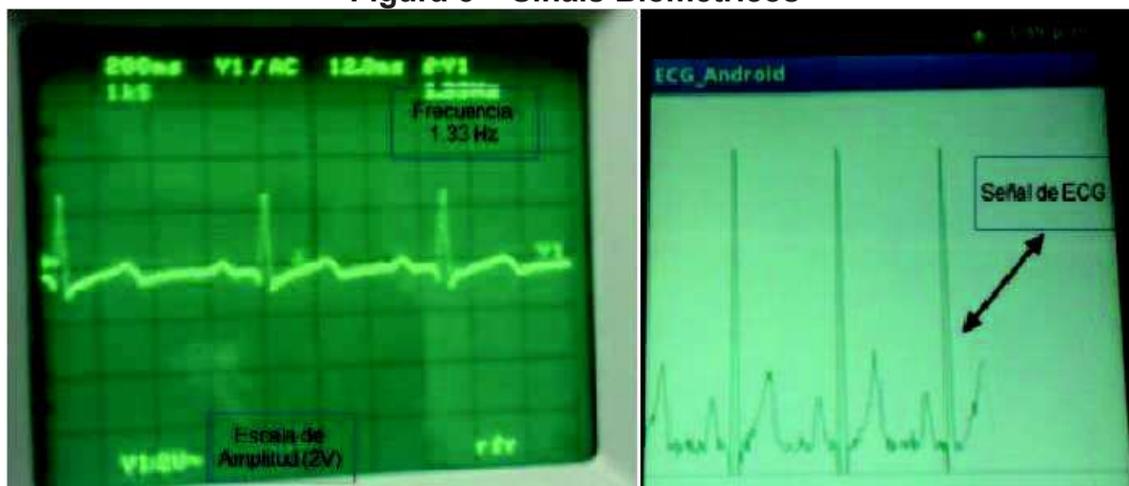
Fonte: [YUN, 2012]

#### 3.2 *Transmisión de señales ECG a sistema móvil con sistema Android vía Bluetooth.*

Este projeto aborda a coleta dos dados biométricos, a digitalização dos dados, o envio através de *Bluetooth*, sua recepção e a visualização em um software implementado em *Android*. Este sistema de telemetria permite monitorar constantemente parâmetros do paciente prevenindo futuras complicações e aumentando a eficiência do trabalho médico. (DE LA CRUZ, 2015).

O artigo descreve a importância do monitoramento contínuo de pacientes e apresenta um protótipo com módulo *Bluetooth* para transmissão de sinais de eletrocardiograma. A Figura 5 mostra os sinais coletados através de um circuito eletrônico, medido através de um osciloscópio (imagem à esquerda) e também sendo mostrado pelo aplicativo *Android* (imagem à direita).

**Figura 5 – Sinais Biométricos**



Fonte: [DE LA CRUZ, 2015]

### **3.3 Research and Development of Hybrid Electric Vehicles CAN-Bus Data Monitor and Diagnostic System through OBD-II and Android-Based Smartphones**

Veículos híbridos são cada vez mais populares, e a demanda de monitoramento remoto está crescendo. Por possuir duas unidades de potência estes veículos precisam de uma estratégia para o uso eficiente da energia. O autor ainda destaca, a importância de compartilhar informações entre veículo e o centro de serviço para aprimorar a eficiência. E considerando o transporte público, o monitoramento do status e da *performance* aumenta a segurança e eficiência para a frota (YANG, 2013).

A figura 6 mostra as telas do sistema. As duas primeiras telas mostram em um mapa a posição e o trajeto do veículo respectivamente. A terceira tela indica o status em tempo real e a quarta tela representa as configurações do aplicativo.

Figura 6 – Telas do Idrive



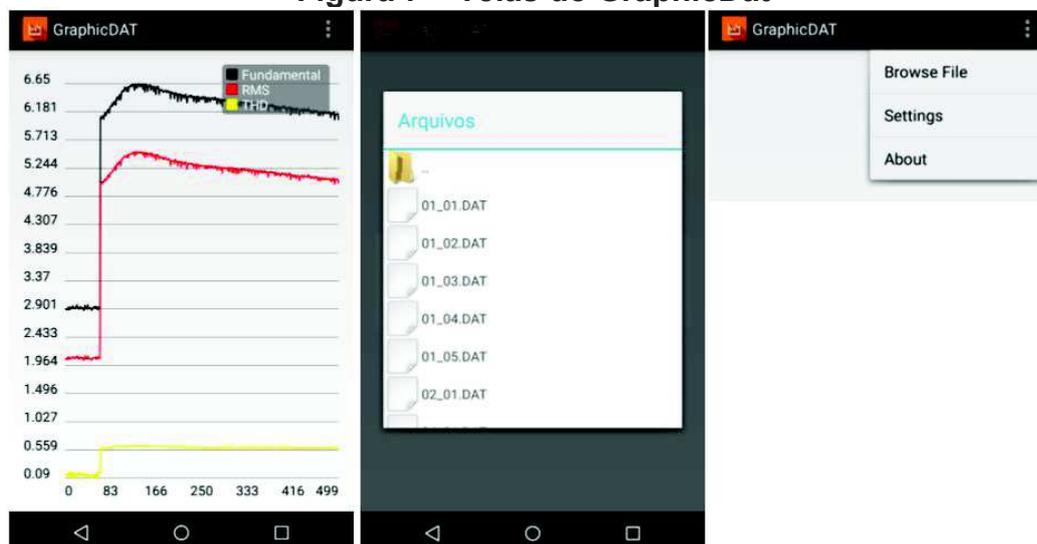
Fonte: [YANG, 2016]

### 3.4 MONITORAMENTO DE VARIÁVEIS ELÉTRICAS DE UM SISTEMA FOTOVOLTAICO COM ARDUINO E ANDROID.

Este trabalho apresenta um sistema de baixo custo para análise da corrente do inversor de um sistema fotovoltaico. O protótipo utiliza a plataforma *Arduino* para coletar e calcular os parâmetros necessários e um aplicativo *Android* foi desenvolvido para apresentar os resultados. O sistema apresentou resultados semelhantes ao analisador de sinal comercial usado como referência, portanto o sistema se mostrou confiável e viável para ser utilizado. (EPIFANIO, 2015)

Apesar deste trabalho não ter relação com veículos, a ideia de um sistema de baixo custo como alternativa a um sistema comercial é semelhante a proposta do Nastabil. A figura 7 mostra as telas do aplicativo desenvolvido para *Android*.

Figura 7 – Telas do GraphicDat



Fonte: [EPIFANIO, 2015]

### 3.5 Comparação Entre os Modelos

A tabela 1 mostra o comparativo entre os trabalhos relacionados à coleta de dados utilizando dispositivos móveis.

**Tabela 1 – Comparativo entre os modelos**

<b>Trabalho / Característica</b>	<i>3.1 Eco-Driving</i>	<i>3.2 Sinais EGC</i>	<i>3.3 Veiculo Híbrido</i>	<i>3.4 Fotovoltaico</i>	<i>Nastabil</i>
<b>Área da aplicação</b>	Setor Automotivo	Saúde	Setor Automotivo	Setor Energético	Setor Automotivo
<b>Plataforma dispositivo</b>	<i>Android</i>	<i>Android</i>	<i>Android</i>	<i>Android</i>	<i>Android</i>
<b>Envio de informação para web</b>	Não	Não	Sim	Não	Sim
<b>Dispositivo coletor de dados</b>	Dispositivo OBD-II	<i>Arduino</i>	Dispositivo OBD-II	<i>Arduino</i>	Dispositivo próprio

Fonte: Elaborada pelo Autor

## 4 METODOLOGIA

Esta seção apresenta a metodologia empregada no desenvolvimento do trabalho.

### 4.1 Delineamento da Pesquisa

Com a intenção de apresentar uma alternativa de baixo custo para coleta de dados em veículos, com foco em transporte público, foi iniciada uma pesquisa sobre trabalhos relacionados à coleta de dados utilizando dispositivos móveis. Os requisitos do projeto foram embasados nos sistemas existentes no mercado que utilizam computadores automotivos.

O estudo de caso representa uma investigação empírica e compreende um método abrangente da coleta e da análise de dados. Neste contexto, foi aplicado o

método de estudos de caso único como abordagem quantitativa e qualitativas de pesquisa (YIN, 2015).

Como técnica de coleta de dados, foi distribuído o aplicativo para diversos usuários, desta maneira, o aplicativo Nastabil coletou informações do dispositivo móvel que estava instalado e enviou para o servidor. Para avaliação do protótipo foi utilizado estudo de cenário.

## 5 PROTÓTIPO E TECNOLOGIAS UTILIZADAS

O modelo Nastabil tem o objetivo de educar o condutor e ao mesmo tempo persistir os dados coletados em um servidor para futuras análises. O aplicativo registra os dados coletados via *Bluetooth*, agrega neste registro informação de acelerômetro e posicionamento e guarda estes dados em um arquivo JSON que é armazenado na memória do dispositivo. Isto ocorre a cada período de tempo pré-determinado ou através de algum evento, por exemplo, por um movimento brusco, detectado pelo acelerômetro do dispositivo. Periodicamente o dispositivo testa se existe conexão com a internet. Se sim, envia todos os arquivos JSON para o servidor. Ao verificar o sucesso no envio do arquivo o mesmo é apagado do dispositivo.

Com os arquivos JSON persistidos no servidor, o usuário poderá interpretar e visualizar estes dados de diversas formas, como por exemplo, um aplicativo *web mobile* para consultar a posição do veículo em tempo real, ou um site com gráficos de consumo de combustível para o proprietário dos veículos consultar, um programa *desktop* com o desenho do trajeto em um mapa, entre outros exemplos de aplicações possíveis com o uso dos dados coletados.

### 5.1 Avaliação Inicial

A ideia inicial era o uso da porta USB para comunicação entre o dispositivo microcontrolado e o dispositivo móvel. O microcontrolador stm32 possui o periférico USB OTG, assim, com um programa exemplo de comunicação USB, foi conectado um *smartphone* ao microcontrolador, e com o uso do aplicativo *UsbTerminal* para *Android* instalado no *smartphone*, poderia verificar os dados que o dispositivo microcontrolado enviou. No teste com o dispositivo Sony Xperia M4, não foi possível estabelecer a conexão pela USB. Em função do resultado, realizou-se o teste com um

*smartphone* Samsung Galaxy Young Duos, mas, da mesma forma, não foi possível estabelecer a conexão pela USB.

Em uma pesquisa mais detalhada das razões do insucesso ao conectar por USB, verificou-se que a porta USB do dispositivo móvel precisa ser OTG (*On-The-Go*). Esta funcionalidade permite conectar um dispositivo microcontrolado como *Device*, e o dispositivo móvel como *Host*. A funcionalidade OTG está presente em poucos dispositivos, e geralmente em dispositivos de maior custo, o que limitaria o projeto, e prejudicaria o objetivo de ser uma solução para coleta de dados de baixo custo. Com a conclusão de que a comunicação por USB não seria o ideal para o projeto, *Bluetooth* passou a ser o foco da pesquisa.

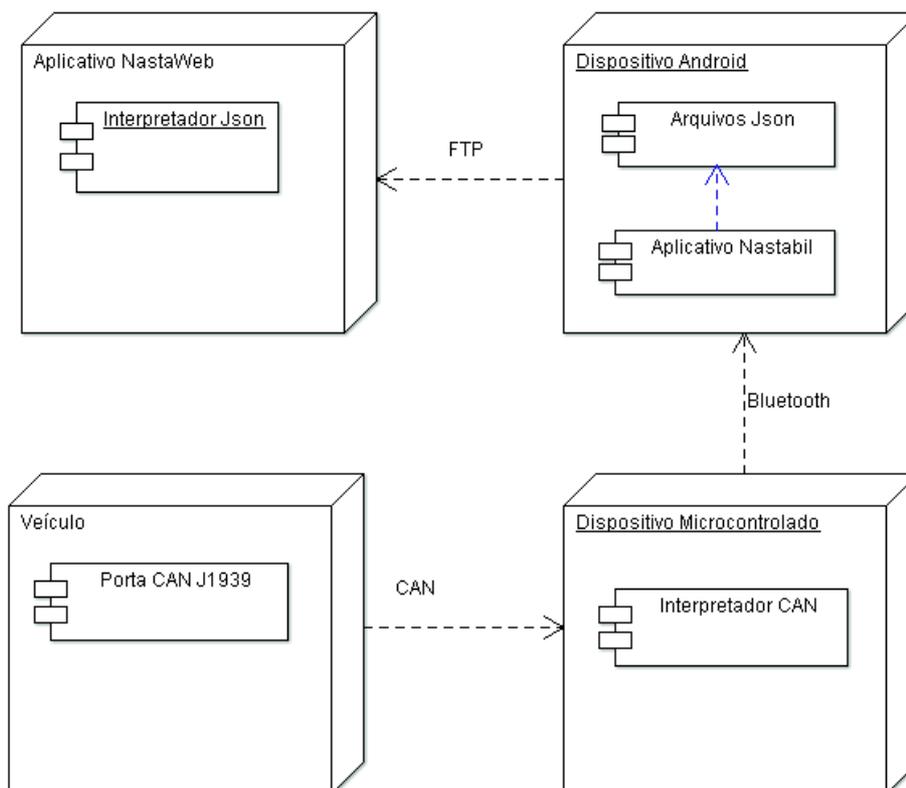
A grande vantagem do uso do *Bluetooth* para este modelo é que praticamente todos os dispositivos *Android* possuem este recurso, o que permite a escolha do dispositivo móvel mais adequado para o projeto, levando em consideração custo e recursos.

## 5.2 Arquitetura do Projeto

O protótipo foi desenvolvido utilizando a linguagem Java com o auxílio da IDE *Android Studio*, para enviar os arquivos JSON com os dados coletados para o servidor foi utilizado a biblioteca Apache FTP (do inglês FTP - *File Transfer Protocol*). Além disso, o aplicativo Nastabil utilizou *Bluetooth*, acelerômetro e GPS.

Para apresentar os dados coletados, foi desenvolvido o NastaWeb, o qual lê os arquivos JSON e apresenta de uma forma que o usuário possa visualizar os dados coletados pelos dispositivos móveis que possuem o aplicativo Nastabil instalado. A Figura 8 apresenta a arquitetura do protótipo Nastabil.

**Figura 8 – Arquitetura do Protótipo**



Fonte: Elaborada pelo Autor

## 5.2 Lista de Requisitos

Os requisitos funcionais descrevem as operações que o sistema irá realizar, como ele deve se comportar em algumas situações e, também podem descrever suas entradas e saídas. Já os requisitos não funcionais descrevem as restrições sobre os serviços limitações do sistema, portabilidade, interface com o usuário entre outros (DA SILVA, 2015).

A Tabela 2 apresenta os requisitos funcionais para o protótipo Nastabil obtidos com base no que os sistemas de coleta de dados em veículos atualmente apresentam.

Tabela 2 – Requisitos Funcionais do Aplicativo Nastabil

Requisito	Descrição
RF01 – Identificar Veículo	Permitir a alteração da identificação do veículo. Através deste campo é possível diferenciar o veículo no aplicativo <i>web</i> .
RF02 – Configurar Dispositivo <i>Bluetooth</i>	Neste Campo o usuário deverá digitar o nome do dispositivo que enviará as informações coletadas por <i>Bluetooth</i> .
RF03 – Indicar conexão <i>Bluetooth</i>	Através de um ícone, informar o usuário que está conectado com o dispositivo <i>Bluetooth</i> configurado.
RF04 – Configurar Servidor FTP	O usuário poderá alterar todas as configurações do servidor FTP, nome do servidor, porta, usuário e senha.
RF05 – Nota instantânea da Dirigibilidade	Mostrar na tela principal um gráfico representando a agressividade da direção, afim de incentivar o condutor a fazer manobras suaves
RF06 – Nota instantânea para consumo de combustível	Quando disponível, mostrar na tela principal um gráfico representando o consumo de combustível instantâneo, afim de incentivar o condutor a economizar combustível.
RF07 – Mostrar dados coletados	Preencher os demais espaços da interface com informações relevantes coletadas via <i>Bluetooth</i>

Fonte: Elaborada pelo Autor

A tabela 3 apresenta os requisitos não funcionais.

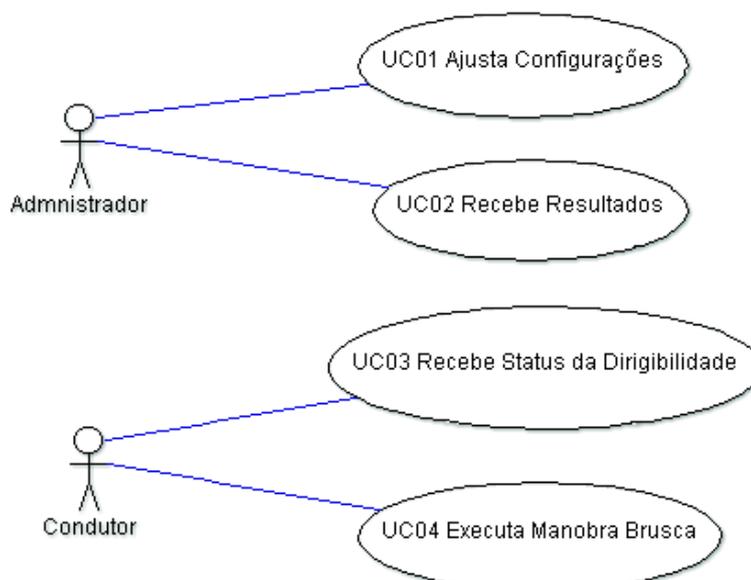
**Tabela 3 – Requisitos Não Funcionais do Aplicativo Nastabil**

Requisito	Descrição
RNF01 – Compatibilidade com versão do sistema operacional	O aplicativo deve ser compatível com o sistema <i>Android</i> 2.3 ou superior.
RNF02 – Leiaute de tela	O sistema deve ser executado com a tela em modo retrato.
RNF03 – Intervenção do usuário	O sistema deve rodar com o menor número possível de intervenção do usuário.
RNF04 – Bloqueio de tela do dispositivo	Sempre que o dispositivo móvel estiver conectado a uma fonte de energia, a tela do dispositivo deverá permanecer ligada.

Fonte: Elaborada pelo Autor

### 5.3 Casos de Uso

A figura 9 apresenta o diagrama de caso de uso do Nastabil, o qual é uma representação gráfica da interação dos atores com o sistema. Os atores são as pessoas que utilizam o sistema, os casos, os quais são representados por elipses, são as ações que os atores podem executar e por fim as linhas representam as iterações entre o ator e o caso (PRESSMAN, 2016).

**Figura 9 – Diagrama de Caso de Uso**

Fonte: Elaborada pelo Autor

Os quadros a seguir são especificações dos casos de usos.

#### Quadro 1 – UC01 – Ajusta as Configurações

<b>Descrição</b>	As configurações devem ser ajustadas por algum administrador. As configurações são: identificação do veículo, nome do dispositivo <i>Bluetooth</i> e configurações de FTP.
<b>Atores Envolvidos</b>	Administrador da Frota
<b>Pré-condições</b>	Aplicativo sendo usado pela primeira vez, quando o dispositivo for utilizado em outro veículo e quando é necessário mudar alguma configuração do FTP.
<b>Pós-condições</b>	O aplicativo fica pronto para ser utilizado no veículo sem nenhuma intervenção do condutor.

Fonte: Elaborado pelo autor

#### Quadro 2 – UC02 – Recebe Resultados

<b>Descrição</b>	Quando o dispositivo móvel estabelecer uma conexão com a internet, os dados coletados são enviados ao servidor FTP para posterior análise.
<b>Atores Envolvidos</b>	Administrador da Frota
<b>Pré-condições</b>	O dispositivo deve estar devidamente configurado.  O dispositivo deve ter alguma forma de conexão com a internet, por 3G/4G ou wifi.
<b>Pós-condições</b>	Os dados persistidos no servidor.

Fonte: Elaborado pelo autor

**Quadro 3 – UC03 – Recebe Status de Dirigibilidade**

<b>Descrição</b>	São barras gráficas que indicam a maneira que o condutor está dirigindo, a primeira barra gráfica alerta manobras bruscas, e a segunda barra gráfica indica consumo instantâneo.
<b>Atores Envolvidos</b>	Condutor do veículo
<b>Pré-condições</b>	A barra gráfica que indica manobras bruscas necessita que o dispositivo móvel possua acelerômetro.  A barra gráfica que indica consumo necessita que o dispositivo esteja conectado por <i>Bluetooth</i> com o dispositivo microcontrolado que coleta os dados de combustível e consumo via CAN.
<b>Pós-condições</b>	Uma maneira do condutor ser influenciado a conduzir de uma forma mais econômica e suave.

Fonte: Elaborado pelo autor

**Quadro 4 – UC04 – Executa Manobra Brusca**

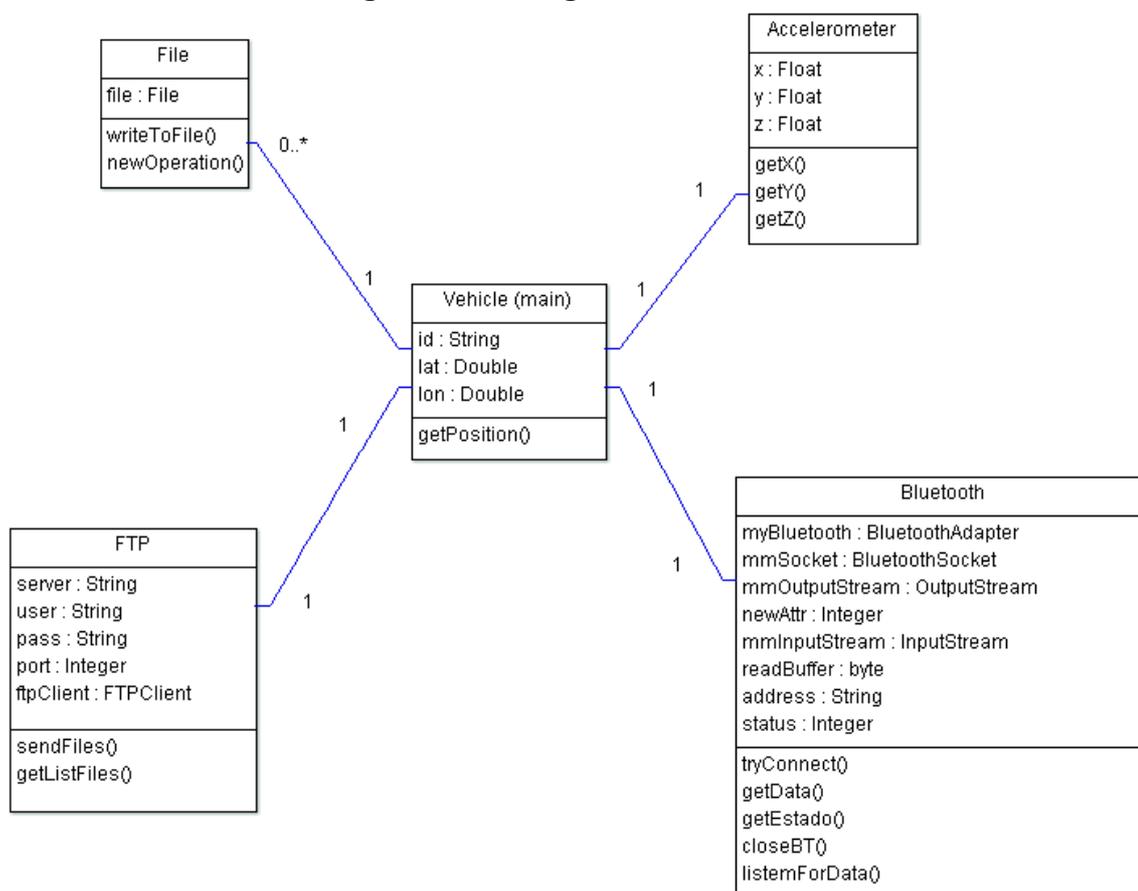
<b>Descrição</b>	Quando o condutor executa uma manobra brusca, o aplicativo registra um evento.
<b>Atores Envolvidos</b>	Condutor do veículo
<b>Pré-condições</b>	Se o dispositivo possuir o recurso do acelerômetro, é possível identificar uma manobra brusca por uma variação grande na soma dos valores dos eixos x, y e z coletados pelo acelerômetro.
<b>Pós-condições</b>	Indica na barra gráfica que ocorreu uma manobra brusca e cria um arquivo JSON com o evento ocorrido.

Fonte: Elaborado pelo autor

## 5.4 Construção do Aplicativo *Android*

Preliminarmente, foi implementado alguns aplicativos a fim de testar algumas funcionalidades separadamente, como por exemplo, enviar arquivo via FTP, comunicação *Bluetooth*, criação de arquivos, aquisição da posição e leitura dos valores do acelerômetro. Ao constatar que o teste foi bem-sucedido, foram criadas classes para serem utilizadas no aplicativo principal. A Figura 10 apresenta uma visão das principais classes do protótipo, e seus respectivos atributos e métodos, afim de demonstrar sucintamente como o sistema funcionará (PRESSMANN, 2016).

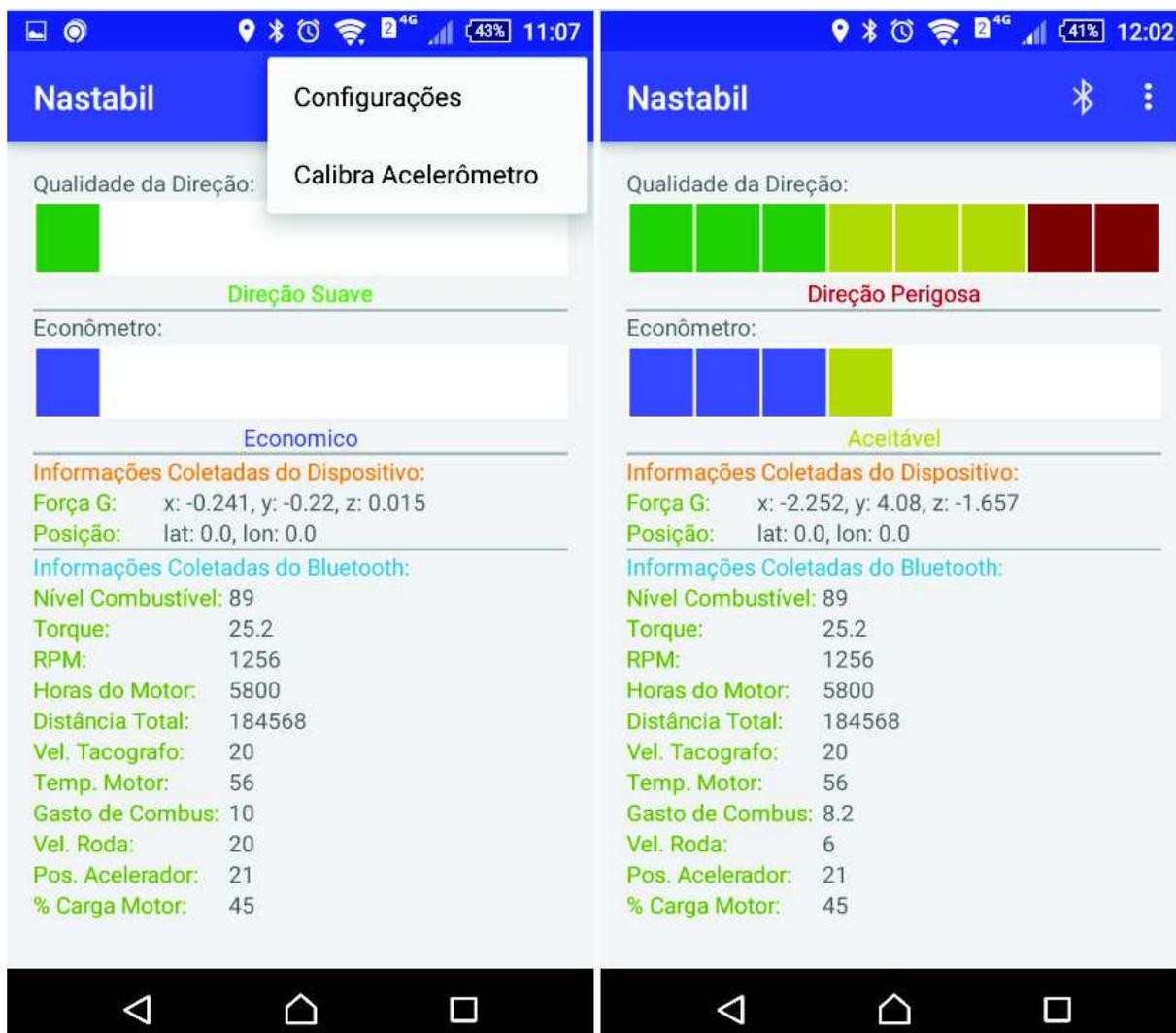
**Figura 10 – Diagrama de Classes**



Fonte: Elaborado pelo autor

O sistema foi construído para ter o menor número de intervenções do usuário possível, por esta razão a tela principal é intencionalmente muito simples, esta tela tem o objetivo de mostrar graficamente para ao motorista a maneira que ele está conduzindo. Para teste, a tela principal também mostra alguns valores coletados pela comunicação *Bluetooth* como é apresentado na figura 11.

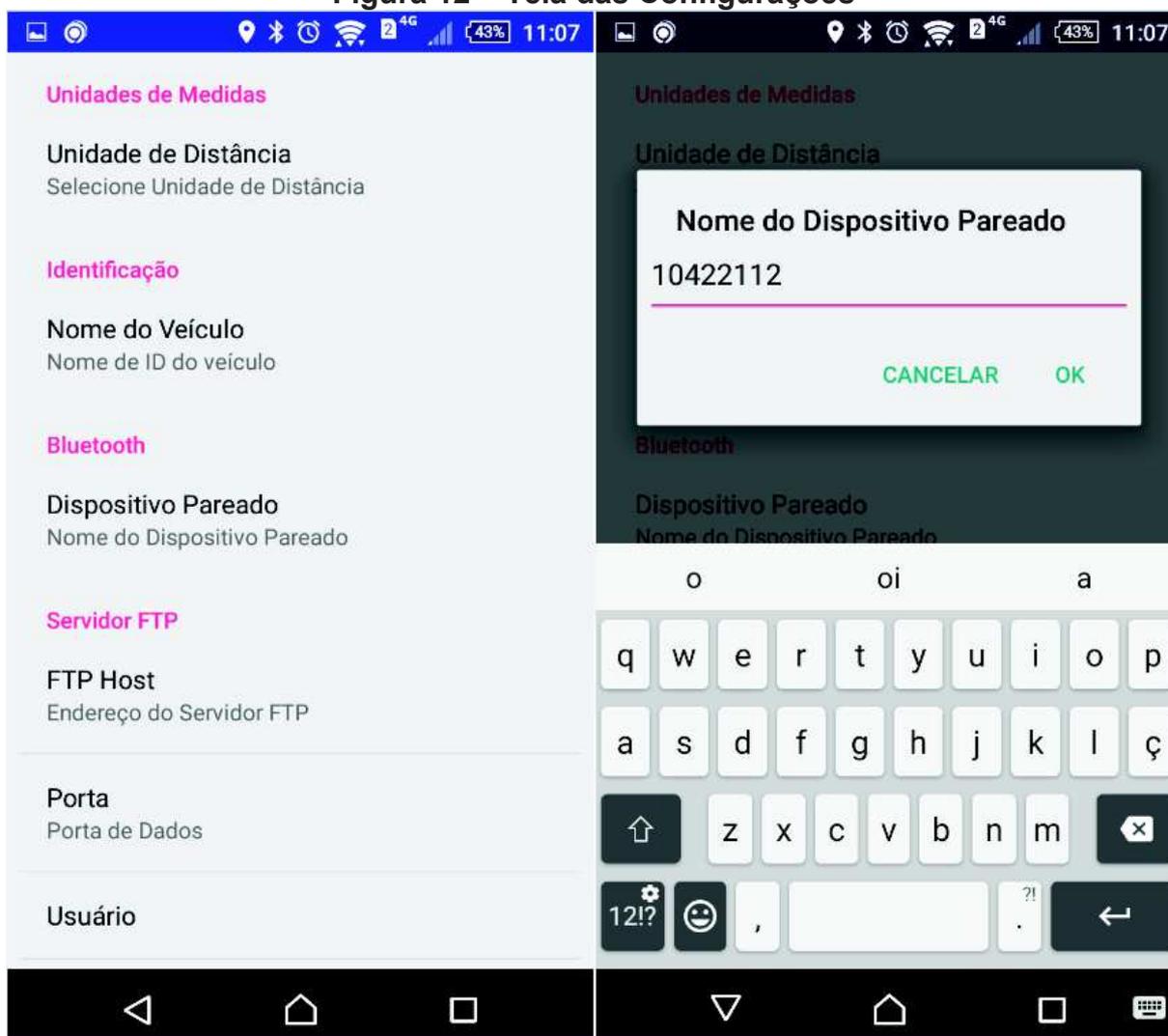
Figura 11 – Tela Principal



Fonte: Elaborado pelo autor

A configuração é uma parte do aplicativo destinada aos administradores da frota, nesta tela podemos alterar a identificação do veículo, o sistema de medidas, o nome do dispositivo *Bluetooth* que o aplicativo deverá coletar os dados e as configurações do servidor FTP. A figura 12 mostra a tela das configurações.

Figura 12 – Tela das Configurações



Fonte: Elaborado pelo autor

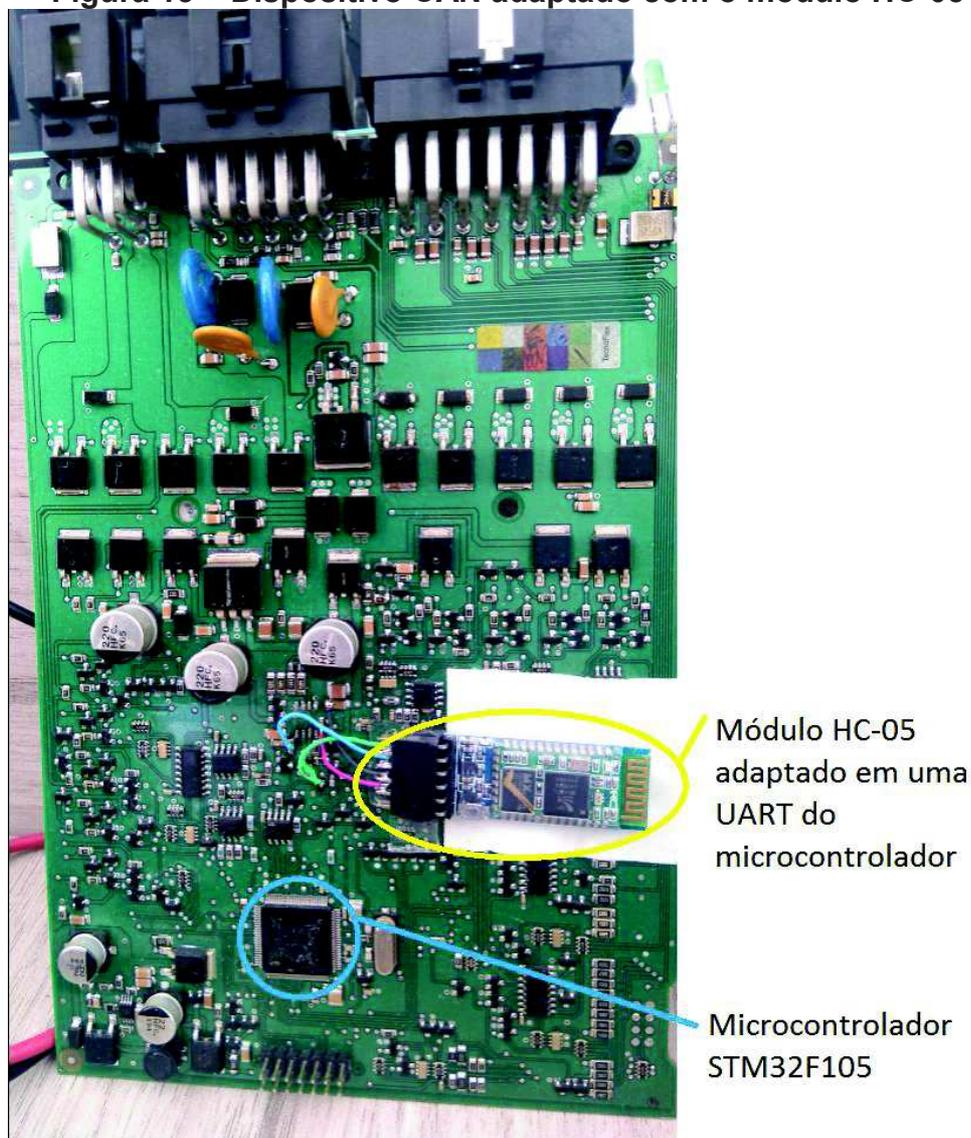
### 5.5 Dispositivo Microcontrolado para Coleta de Dados

Para interface entre o veículo e o dispositivo móvel foi necessário o uso de um dispositivo que utiliza como núcleo o microcontrolador STM32F105, com o periférico CAN e com um módulo *Bluetooth* HC-05 adaptado em uma porta UART do microcontrolador. Este dispositivo chamamos de módulo CAN, que possui a função de ler os pacotes CAN no formato J1939 provenientes do veículo e a interpretação dos dados relevantes para o sistema.

Para a seleção dos dados relevantes foi utilizado um *software* que utiliza diagramas de blocos, que pode ser visualizado no APÊNDICE A e no APÊNDICE B. No APÊNDICE C podemos visualizar a forma que os dados serão empacotados para

serem enviados através do *Bluetooth*. A Figura 13 apresenta o módulo CAN, adaptado com o módulo HC-05.

**Figura 13 – Dispositivo CAN adaptado com o módulo HC-05**



Fonte: Elaborado pelo autor

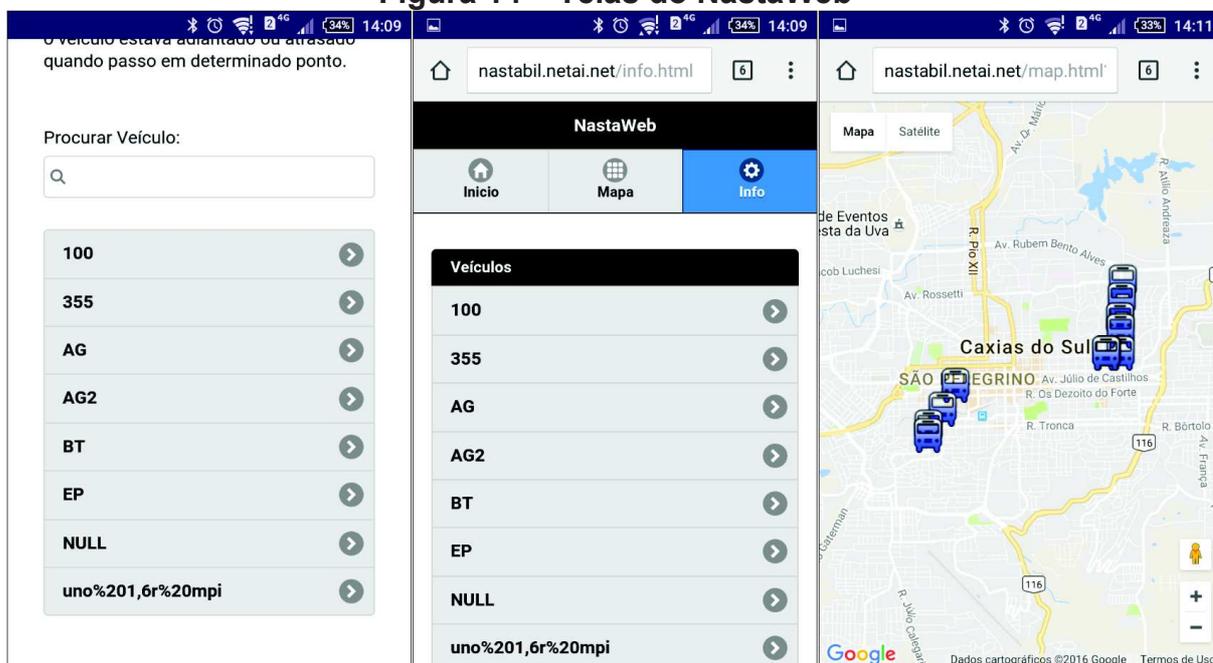
## 5.6 Aplicativo *Web* para Visualizar os Resultados

O NastaWeb é um aplicativo *web* criado para visualizar os resultados obtidos dos veículos, ele mostra de uma maneira simples uma lista dos dados coletados por veículo ou estas mesmas informações disponibilizadas em um mapa. A interface do aplicativo é responsiva, para que seja utilizável em computadores ou dispositivos móveis. Para criação deste aplicativo foi utilizado as linguagens HTML e *JavaScript*.

Um recurso interessante que este aplicativo disponibiliza é a automatização da leitura das informações, ou seja, não é necessária nenhuma intervenção para adicionar o veículo na lista do aplicativo, o NastaWeb lista as pastas como veículos e os arquivos das pastas como os dados deste veículo. O APÊNDICE D e APÊNDICE E mostram a estrutura das pastas no servidor.

A figura 14 mostra as principais telas do aplicativo criado para visualizar os dados coletados. O APÊNDICE F e o APÊNDICE G mostram o NastaWeb em outro formato de tela.

**Figura 14 – Telas do NastaWeb**



Fonte: Elaborado pelo autor

## 6 AVALIAÇÃO

Esta seção apresenta a avaliação da aplicação, a estratégia utilizada é estudo de cenário.

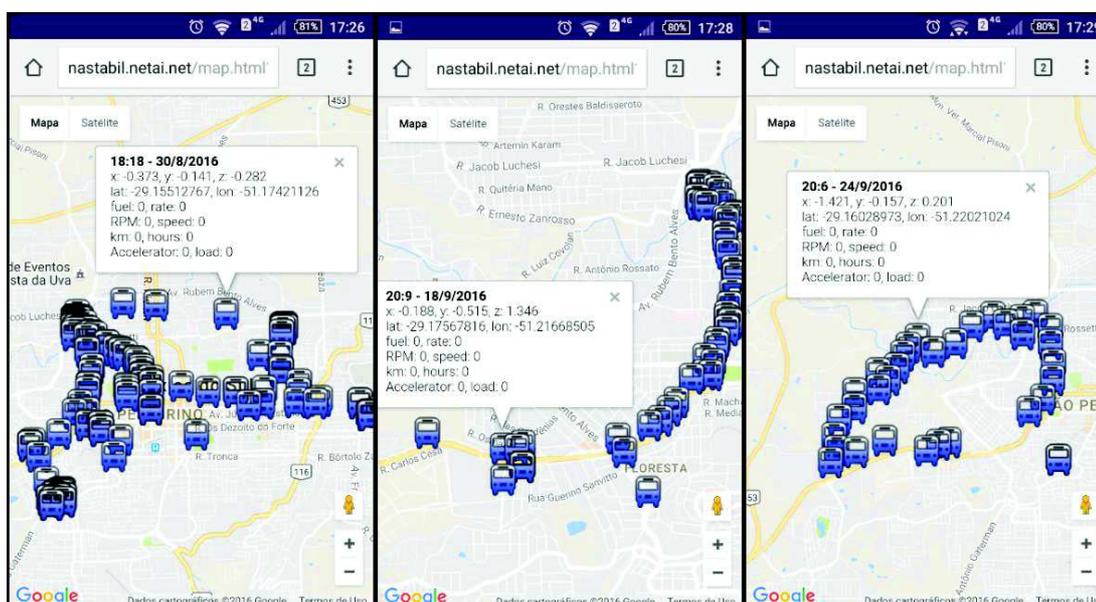
Cenário – Administrando uma Pequena Frota.

Cenário demonstra a utilização do sistema para a administração de uma pequena frota de veículos, utilizando um *smartphone* em cada veículo.

*Antônio Gustavo é proprietário de uma empresa de transporte, a AG Transportes, a empresa conta com três ônibus que são utilizados para transporte de funcionários. Ele adquiriu três smartphones Android QbeX QX A18 por R\$350 cada, para equipar cada ônibus, este equipamento possui acelerômetro, GPS, Bluetooth e processador e memória RAM suficientes para utilizar o aplicativo Nastabil, com um custo realmente baixo. Por ser uma empresa pequena, Antônio Gustavo utilizou o servidor FTP padrão, os dados coletados podem ser visualizados na página [nastabil.netai.net](http://nastabil.netai.net), ele pode acompanhar o trajeto que seus veículos fizeram, e verificar se seus funcionários realizaram manobras bruscas.*

No cenário o administrador configurou três smartphones com os nomes de veículos AG, AG2 e AG3, a Figura 15 mostra o trajeto dos três veículos, e a Figura 16 apresenta parte da lista dos dados coletados.

**Figura 15 – Os Trajeto dos veículos AG, AG2 e AG3 respectivamente.**



Fonte: Elaborado pelo autor

**Figura 16 – Lista dos dados coletados dos veículos AG, AG2 e AG3.**

Veículo	Tempo	Data	x	y	z	lat	lon	fuel	speed	rate	accelerator
AG	13:0:24	27/8/2016	8.589	2.31	-5.041	0	0	0l	0km/h	0km/l	0%
	13:0:26	27/8/2016	-1.194	2.163	1.157	0	0	0l	0km/h	0km/l	0%
	13:1:17	27/8/2016	-1.039	5.349	-1.933	0	0	0l	0km/h	0km/l	0%
AG2	16:1:6	17/9/2016	0.762	3.662	-0.937	-29.16939545	-51.19879658	0l	0km/h	0km/l	0%
	11:6:37	21/9/2016	-0.306	5.503	-1.933	0	0	89l	20km/h	10km/l	21%
	8:49:35	8/9/2016	0.161	-2.344	2.247	-29.16736697	-51.1492354	0l	0km/h	0km/l	0%
AG3	20:17:23	24/9/2016	-1.421	-0.157	0.201	-29.17864109	-51.23497481	0l	0km/h	0km/l	0%
	20:2:41	24/9/2016	-1.421	-0.157	0.201	-29.15744007	-51.20244841	0l	0km/h	0km/l	0%
	20:20:26	24/9/2016	-1.421	-0.157	0.201	-29.17597455	-51.21661249	0l	0km/h	0km/l	0%

Fonte: Elaborado pelo autor

## 7 CONCLUSÃO

Este artigo apresentou uma alternativa de baixo custo para coleta de dados em veículos e envio para um servidor para posterior análise com o objetivo de possibilitar a administração de uma frota de veículos. A questão de pesquisa inicial que guiou o trabalho, pode ser respondida de forma positiva tendo em vista que os objetivos foram alcançados.

Inicialmente, foram pesquisados sistemas semelhantes, os quais influenciaram algumas escolhas do desenvolvimento deste trabalho, posteriormente, foi elaborado um estudo mais aprofundado dos conceitos utilizados para este trabalho descrito na seção Referencial Teórico e por fim o desenvolvimento de um protótipo para provar o conceito.

Os dados coletados e persistidos em um servidor foi o principal objetivo deste projeto, esta funcionalidade foi provada, e os resultados indicaram a viabilidade de um projeto no âmbito comercial utilizando os conceitos deste trabalho. O aplicativo *Android* além de ser a interface entre o módulo CAN e o servidor, serviu como um mecanismo para informar instantaneamente ao motorista a forma da condução, tornando o sistema mais completo e aproveitando melhor os recursos do dispositivo móvel.

O NastaWeb, aplicativo *web* para visualizar os resultados, teve um papel de demonstrar uma parte das possibilidades de utilização dos dados dos veículos persistidos em um servidor, apesar de requerer diversos aperfeiçoamentos para ser utilizado em uma aplicação real, o NastaWeb se mostrou promissor.

### 7.1 Trabalhos Futuros

No decorrer do desenvolvimento, algumas possíveis melhorias foram notadas e estão listadas a seguir:

Divisão do pacote de dados proveniente da comunicação *Bluetooth*. Os dados com variação mais frequente como velocidade do veículo, RPM, consumo instantâneo seriam enviados através de um pacote com frequência igual ou menor a um segundo, já os dados com variação menos frequente como distância percorrida, horas de funcionamento do motor, seriam enviados por um pacote com frequência de um minuto ou mais.

Utilização de outro sistema de envio de informação ao servidor. O aplicativo NastaWeb algumas vezes apresentou baixa performance devido à grande quantidade de arquivos JSON lidos. Ainda utilizando o protocolo FTP poderia utilizar o recurso de acrescentar ao arquivo (comando *append*), com isso o aplicativo NastaWeb precisará abrir menos arquivos para mostrar os mesmos dados. Outra possibilidade seria utilizar um sistema com banco de dados.

Simplificar ainda mais a tela principal. Para o propósito de teste, as variáveis originadas do *Bluetooth* foram mostradas na tela principal, mas para um uso comercial estas variáveis deverão ser mostradas em uma tela separada, assim, a tela principal ficará mais limpa, apenas com as barras gráficas indicando ao motorista a maneira que o veículo está sendo conduzido.

## REFERÊNCIAS

ATZORI, Luigi; IERA, Antonio; MORABITO, Giacomo. The internet of things: A survey. **Computer networks**, v. 54, n. 15, p. 2787-2805, 2010.

BARBOSA, Jorge et al. Computação Móvel e Ubíqua no Contexto de uma graduação de Referência. **Revista Brasileira de Informática na Educação**, v. 15, n. 3, 2007.

BAXEVANIS, Yannis. NMEA 2000, 2012. Disponível em: <<http://www.nuovamarea.net/blog/category/nmea%202000/>>. Acesso em 17 de julho. 2016.

BHASKAR, Ashish; QU, Ming; CHUNG, Edward. Bluetooth Vehicle Trajectory by Fusing Bluetooth and Loops: Motorway Travel Time Statistics. **IEEE Transactions on Intelligent Transportation Systems**, v. 16, n. 1, p. 113-122, 2015. COOK, J. A.; FREUDENBERG, J. S. Controller Area Network (CAN). 2007.

DA SILVA, Mirian Camila; DA SILVA, Mislene Dalila. Desenvolvimento de um aplicativo de auxílio para localização no Centro Universitário de Patos de Minas–UNIPAM. **Revista Perquirere**, v. 12, n. 2, p. 99-112, 2015.

DAWSON, Matt. **Social Theory for Alternative Societies**. Palgrave Macmillan, 2016.

DE LA CRUZ, B. et al. Transmission of ECG signals with android mobile system via Bluetooth. In: **Health Care Exchanges (PAHCE), 2015 Pan American**. IEEE, 2015. p. 1-6.

DE OLIVEIRA, Luid Pereira; FERRAZ, Aline Almeida. Integração de Ferramentas de Telemetria e Sistemas de TMS para Gestão Operacional do Transporte Rodoviário de Cargas (TRC). **Ingeniería de Transporte**, v. 18, n. 2, 2016.

DI NATALE, Marco et al. **Understanding and using the controller area network communication protocol: theory and practice**. Springer Science & Business Media, 2012.

DOS REIS, GUSTAVO GARCIA. **APLICABILIDADE DE MICROCONTROLADORES DA FAMÍLIA STM32 NO CÁLCULO DE PARÂMETROS DE SINAIS ANALÓGICOS**. 2011

EPIFANIO, Gustavo Pacheco. **MONITORAMENTO DE VARIÁVEIS ELÉTRICAS DE UM SISTEMA FOTOVOLTAICO COM ARDUINO E ANDROID**. 2015. Tese de Doutorado. Universidade Federal do Rio de Janeiro.

FRIESEN, M. R.; MCLEOD, R. D. Bluetooth in intelligent transportation systems: a survey. **International Journal of Intelligent Transportation Systems Research**, v. 13, n. 3, p. 143-153, 2015.

JUNG, Cláudio Rosito et al. Computação embarcada: Projeto e implementação de veículos autônomos inteligentes. **Anais do CSBC**, v. 5, p. 1358-1406, 2005.

LECHETA, Ricardo R. **Google Android 4ª edição**. Novatec Editora, 2015.

MARQUES, Gonçalo. **Sistema inteligente de monitorização e controlo de energia-IOT ao serviço da eficiência energética**. 2015.

OTSUKA, Gilberto Sadao; ZANELATO, Ana Paula Ambrósio. O SISTEMA ANDROID NO UNIVERSO DOS DISPOSITIVOS MÓVEIS. **ETIC-ENCONTRO DE INICIAÇÃO CIENTÍFICA-ISSN 21-76-8498**, v. 8, n. 8, 2015.

PEREIRA, Lucio Camilo Oliva; DA SILVA, Michel Lourenço. **Android para desenvolvedores**. Brasport, 2009.

PIETERSE, Heloise; OLIVIER, Martin S. Bluetooth command and control channel. **Computers & Security**, v. 45, p. 75-83, 2014.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software-8ª Edição**. McGraw Hill Brasil, 2016.

PROSTT, Marcela Eloisa. **Interface Web utilizando design responsivo: um estudo de caso aplicado a smartphones, tablets, computadores e televisores**. 2013.

RODRIGUES, Johnson Andrade. **Implementação da Comunicação Sem Fio de um Módulo Estimador da Frequência Cardíaca Fetal Baseado em FPGA**. 2016

SOUZA, Bruno Jose de. **Sistemas inteligentes de transporte aplicados ao controle de tráfego**. 2015.

VASCONCELOS, Wanderson Douglas Souza; JUNIOR, Ivanilson França Vieira; SOARES, Allyson Amilcar Angelus Freire. ESTUDO E DESENVOLVIMENTO DE UMA REDE SEM FIO BASEADA EM BLUETOOTH SOBRE A FERRAMENTA DE PROTOTIPAGEM ARDUINO. In: **IX Congresso de Iniciação Científica do IFRN**. 2013.

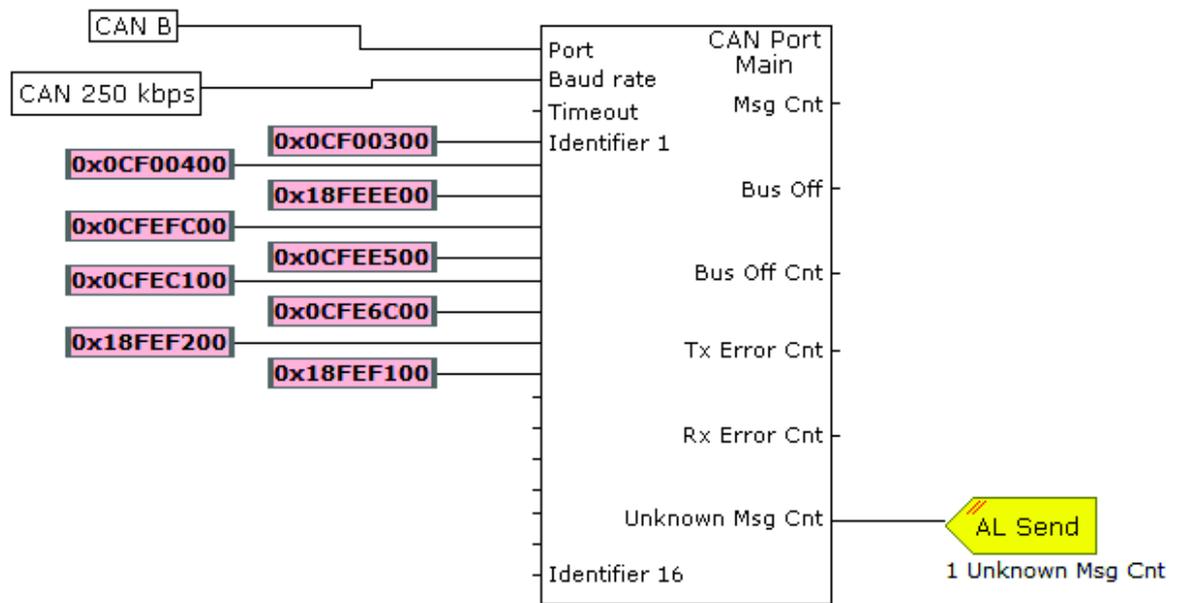
YANG, Yalian et al. Research and development of hybrid electric vehicles can-bus data monitor and diagnostic system through obd-ii and android-based smartphones. **Advances in Mechanical Engineering**, v. 5, p. 741240, 2013.

YIN, Robert K. **Estudo de Caso-: Planejamento e Métodos**. Bookman editora, 2015.

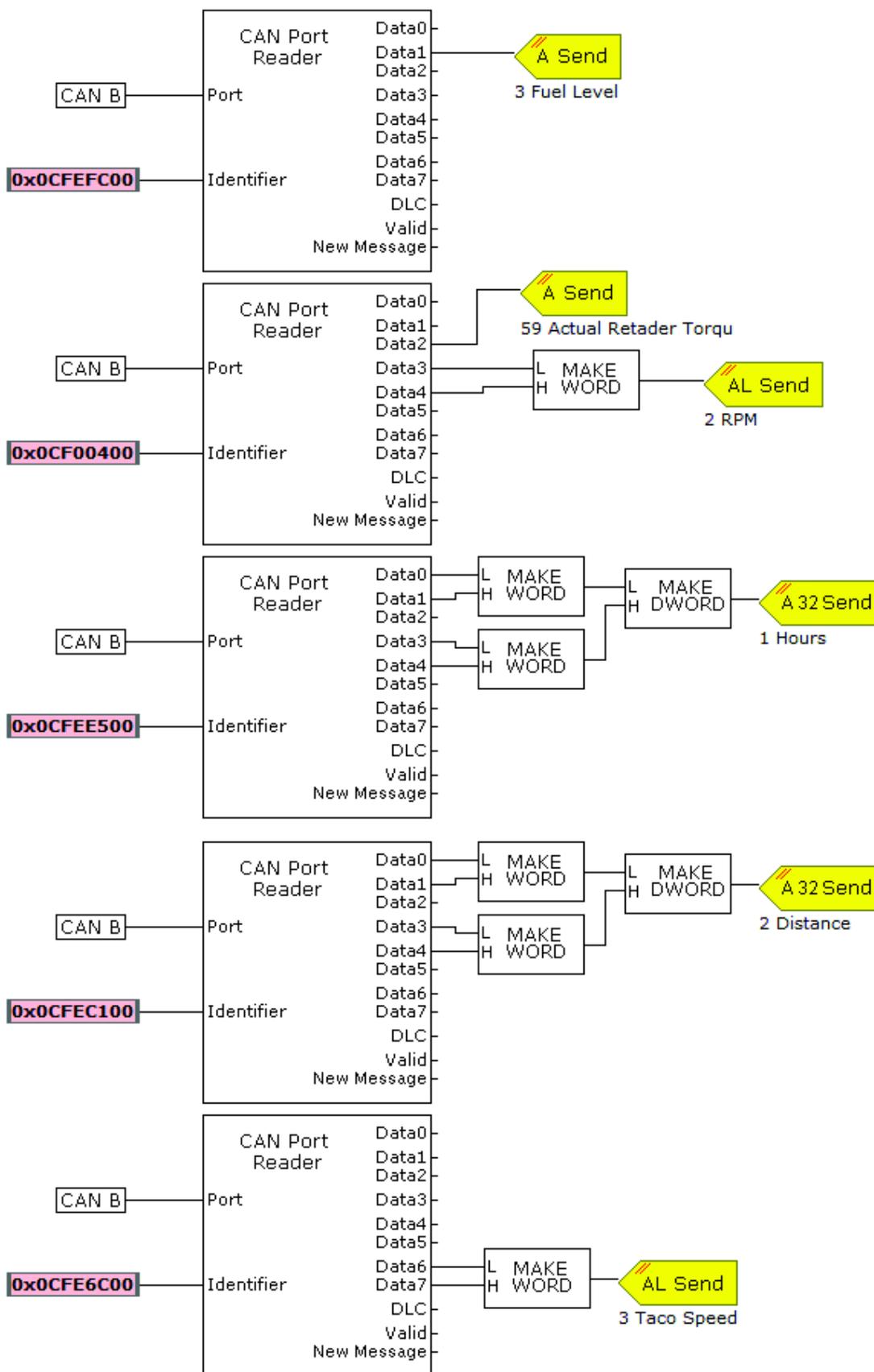
YUN, Doo Seop et al. Development of the eco-driving and safe-driving components using vehicle information. In: **2012 International Conference on ICT Convergence (ICTC)**. IEEE, 2012. p. 561-562.

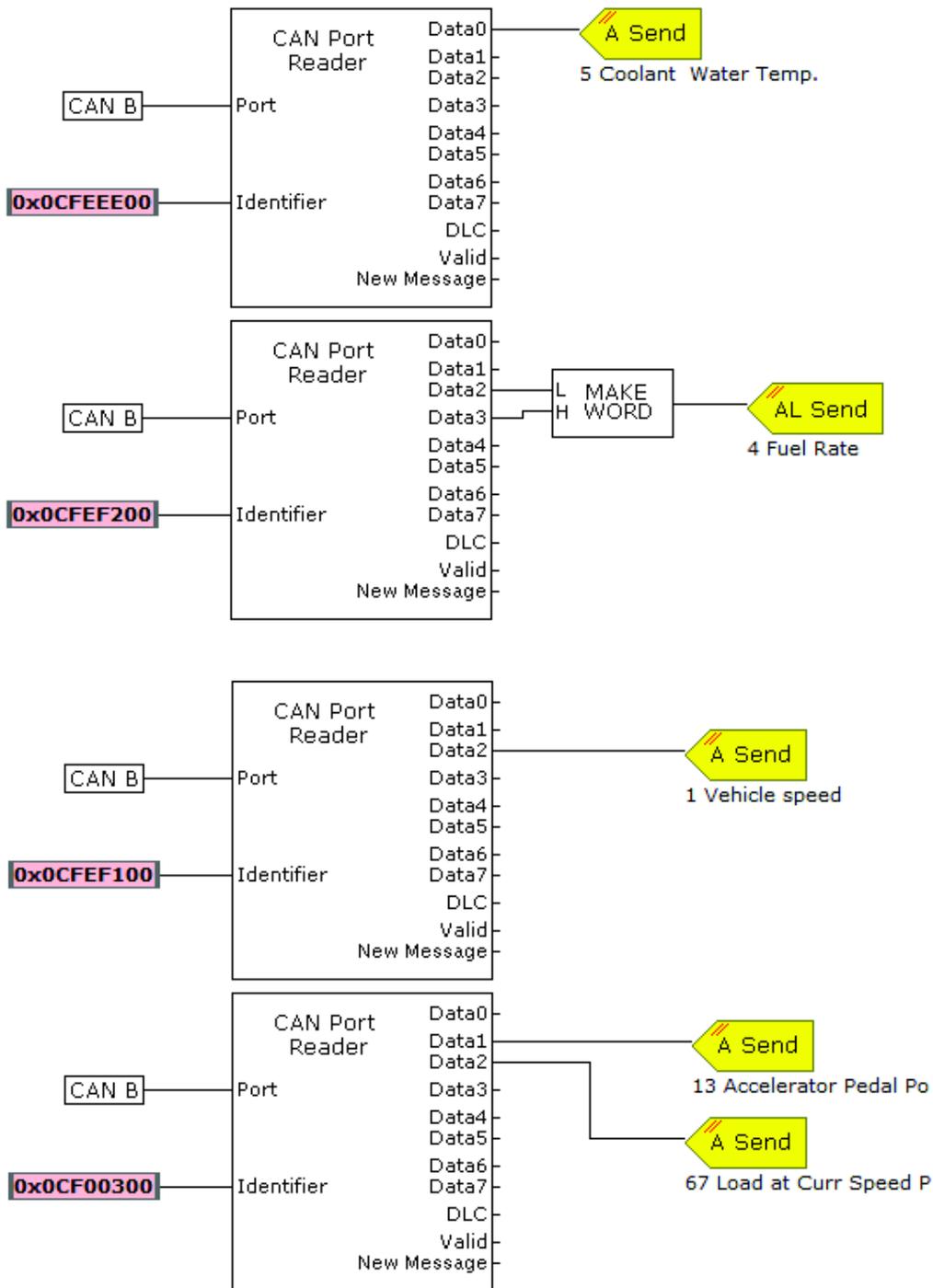
## APÊNDICES

APÊNDICE A – Diagramas de blocos para ler os pacotes CAN relevantes (inicialização).



## APÊNDICE B – Diagramas de blocos para ler os pacotes CAN relevantes (parte 2)







APÊNDICE E – Arquivos JSON em uma pasta de um veículo, neste exemplo na pasta do veículo AG2.

Nome	Tamanho	Tipo	Modificado	Permissões	Proprietári...
..					
20160907_151100.json	290	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_151104.json	282	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_151129.json	308	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_151155.json	309	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_151156.json	308	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_151928.json	310	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_151929.json	309	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_152221.json	308	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_152251.json	311	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...
20160907_152251.json	307	Arquivo JS...	07/09/2016 15:...	0644	9253178 92...

148 arquivos. Tamanho: 45.422 bytes

APÊNDICE F – Tela do NastaWeb em um computador (Tela inicial)



## Monitoramento de Frota

O NastaWeb é um app Web para monitorar uma frota de veículo.

A idéia é que o veículo transmita um arquivo json com seu trajeto e eventos cada vez que chegar no pátio, e este app use estes arquivos json para mostrar ao administrador da frota as informações coletadas.

Com isso o administrador poderá verificar se em algum ponto teve excesso de velocidade, alguma freada brusca, se o veículo estava adiantado ou atrasado quando passo em determinado ponto.

Procurar Veículo:



100



355



AG



APÊNDICE G – Tela do NastaWeb em um computador (Mapa)

