

OBD-JRP

Monitoramento Veicular com Java e Raspberry Pi

Aluno: Ricardo Artur Staroski

Orientador: Miguel Alexandre Wisintainer

Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Requisitos funcionais e não funcionais
- Especificação
- Implementação
- Operacionalidade da implementação
- Resultados e discussões
- Conclusões
- Vantagens
- Limitações
- Sugestões

Introdução

- TCC1: O que fazer? O que não fazer?
 - Ferramenta didática composta por máquina virtual e ambiente de desenvolvimento?
 - Simulador de tráfego com Google Maps?
 - Display veicular para notificar condutores imprudentes?
 - Que tal alguma coisa com OBD?

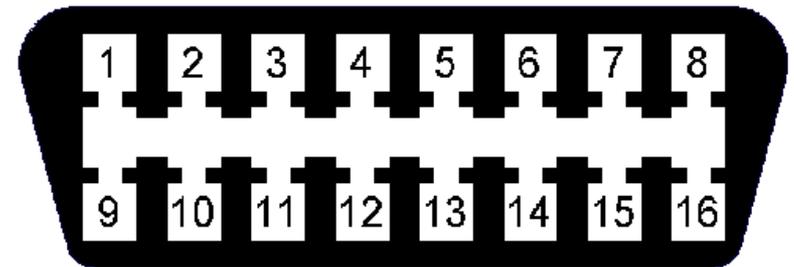
Objetivos

- Firmware
 - Monitorar porta OBD2 de um veículo, ler dados e enviá-los a um servidor
- Servidor
 - Java Servlet que recebe os dados e os armazena
- Páginas web
 - Visualizar dos dados e histórico

Fundamentação Teórica

- OBD

- California Air Resource Board (CARB)
- Society of Automotive Engineers (SAE)
 - Conector SAE J1962
 - Protocolos ECU



- Resolução CONAMA nº 357
 - 100% dos carros, até Janeiro 2011

Fundamentação Teórica

- Interface ELM327
 - Troca de mensagens com a ECU
 - Conexão Serial, USB, WiFi e Bluetooth
 - Interface texto
 - Comandos AT
 - PIDs OBD2



Fundamentação Teórica

- Raspberry Pi 3 Model B
 - PC do tamanho de um cartão de crédito;
 - processador ARM quad core de 1,2GHz;
 - 1GB de RAM;
 - 1 entrada micro SD (S.O. e armazenamento);
 - Bluetooth e WiFi;
 - 4 portas USB;
 - 1 porta micro USB (alimentação);
 - saída HDMI.

Fundamentação Teórica

Raspberry Pi 3
Model B

Dimensions
85.6mm x 56mm x 21mm

element14

4 x USB 2
Ports

40 Pin
Extended GPIO

Broadcom
BCM2837 64bit
Quad Core CPU
at 1.2GHz,
1GB RAM

10/100
LAN Port

On Board
Bluetooth 4.1
Wi-Fi

3.5mm 4-pole
Composite Video
and Audio
Output Jack

MicroSD
Card Slot

CSI Camera Port

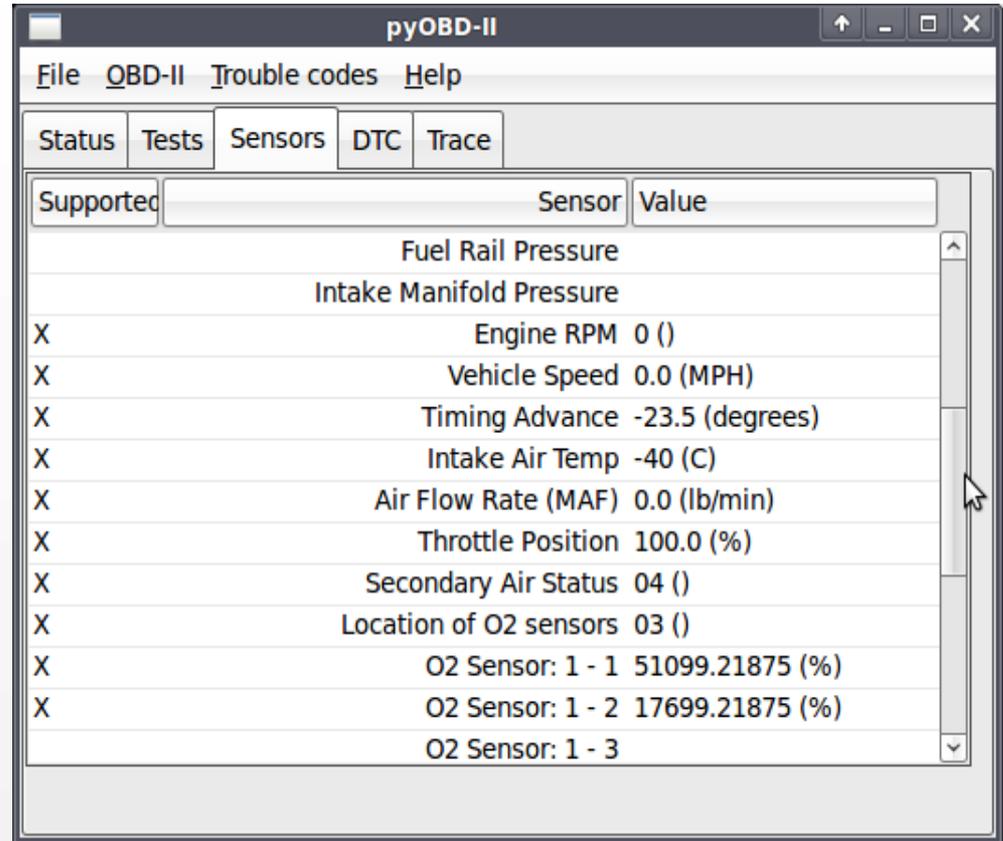
DSI Display Port

Micro USB Power Input.
Upgraded switched
power source that can
handle up to 2.5 Amps

Full Size HDMI
Video Output

Trabalhos Correlatos

- PyOBD
 - PC e notebooks
 - Interface USB
 - Executar testes
 - Leituras

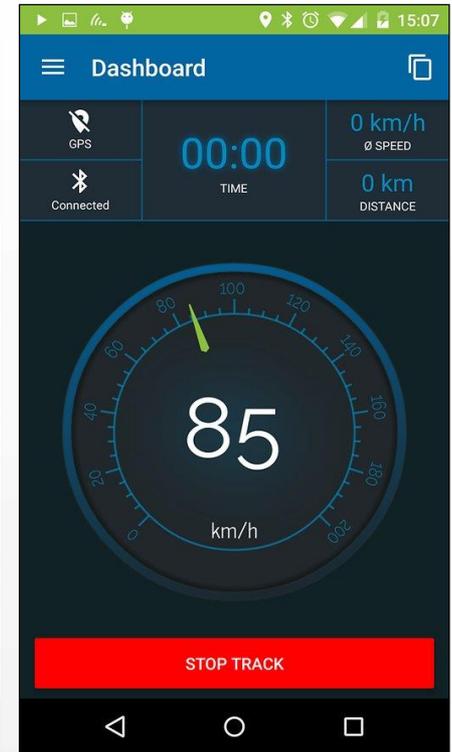
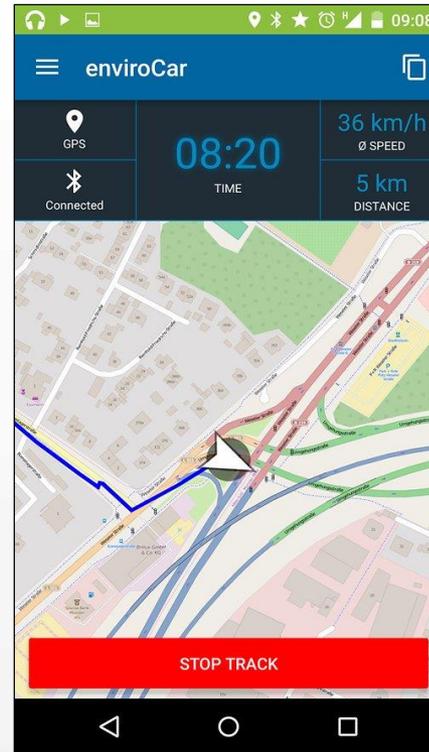


The screenshot shows the pyOBD-II application window. The title bar reads 'pyOBD-II'. The menu bar includes 'File', 'OBD-II', 'Trouble codes', and 'Help'. Below the menu bar are tabs for 'Status', 'Tests', 'Sensors', 'DTC', and 'Trace'. The 'Sensors' tab is active, displaying a table with columns 'Supported', 'Sensor', and 'Value'. The table lists various sensors and their current values.

Supported	Sensor	Value
	Fuel Rail Pressure	
	Intake Manifold Pressure	
X	Engine RPM	0 ()
X	Vehicle Speed	0.0 (MPH)
X	Timing Advance	-23.5 (degrees)
X	Intake Air Temp	-40 (C)
X	Air Flow Rate (MAF)	0.0 (lb/min)
X	Throttle Position	100.0 (%)
X	Secondary Air Status	04 ()
X	Location of O2 sensors	03 ()
X	O2 Sensor: 1 - 1	51099.21875 (%)
X	O2 Sensor: 1 - 2	17699.21875 (%)
	O2 Sensor: 1 - 3	

Trabalhos Correlatos

- EnviroCar
 - Android
 - Interface Bluetooth
 - Upload servidor
 - Leituras
 - Trajeto
 - Compartilhar



RF Firmware

- inicializar automaticamente ao ligar;
- conectar-se ao ELM327 via Bluetooth;
- armazenar dados localmente até enviar ao servidor;
- tentar conectar-se com servidor a cada 5 minutos;
- enviar o número do chassi e dados armazenados localmente.

RNF Firmware

- desenvolvimento em Java SE;
- executar em sistema Raspbian.

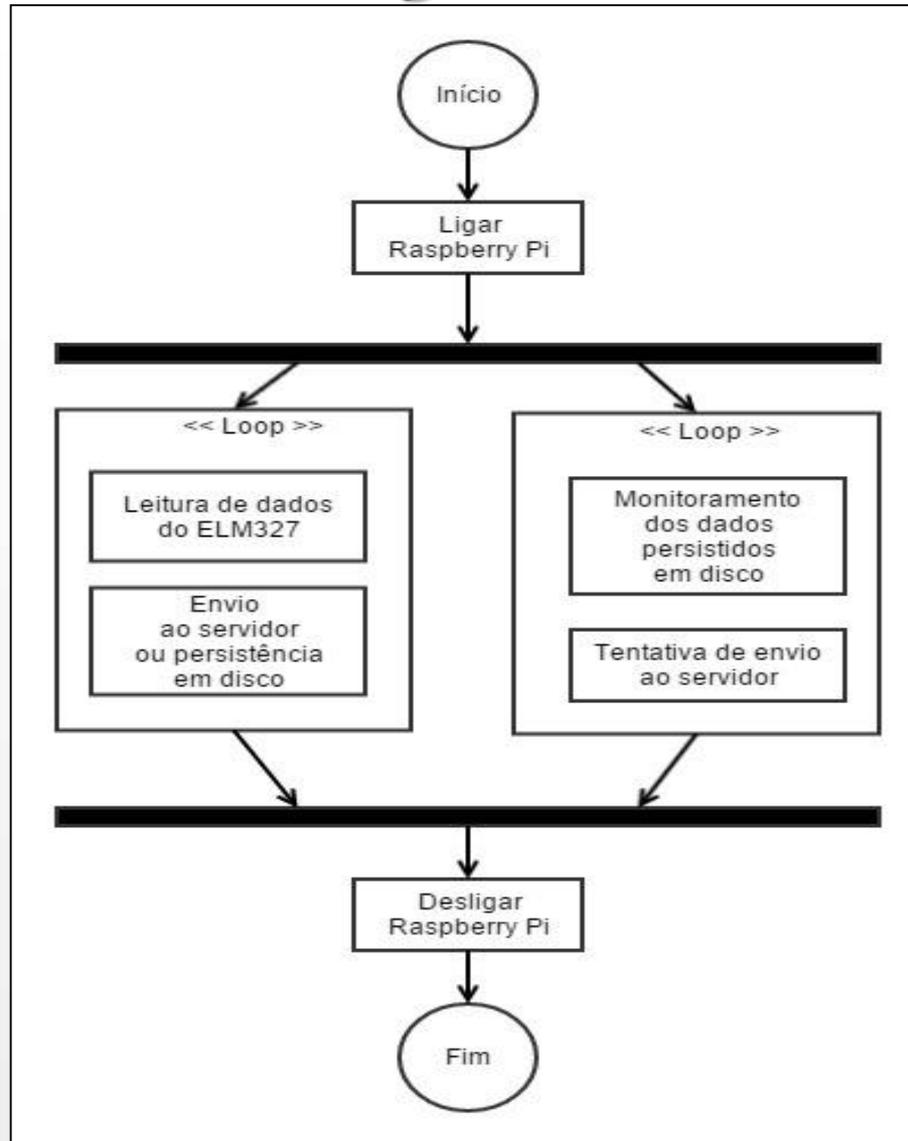
RF Servidor

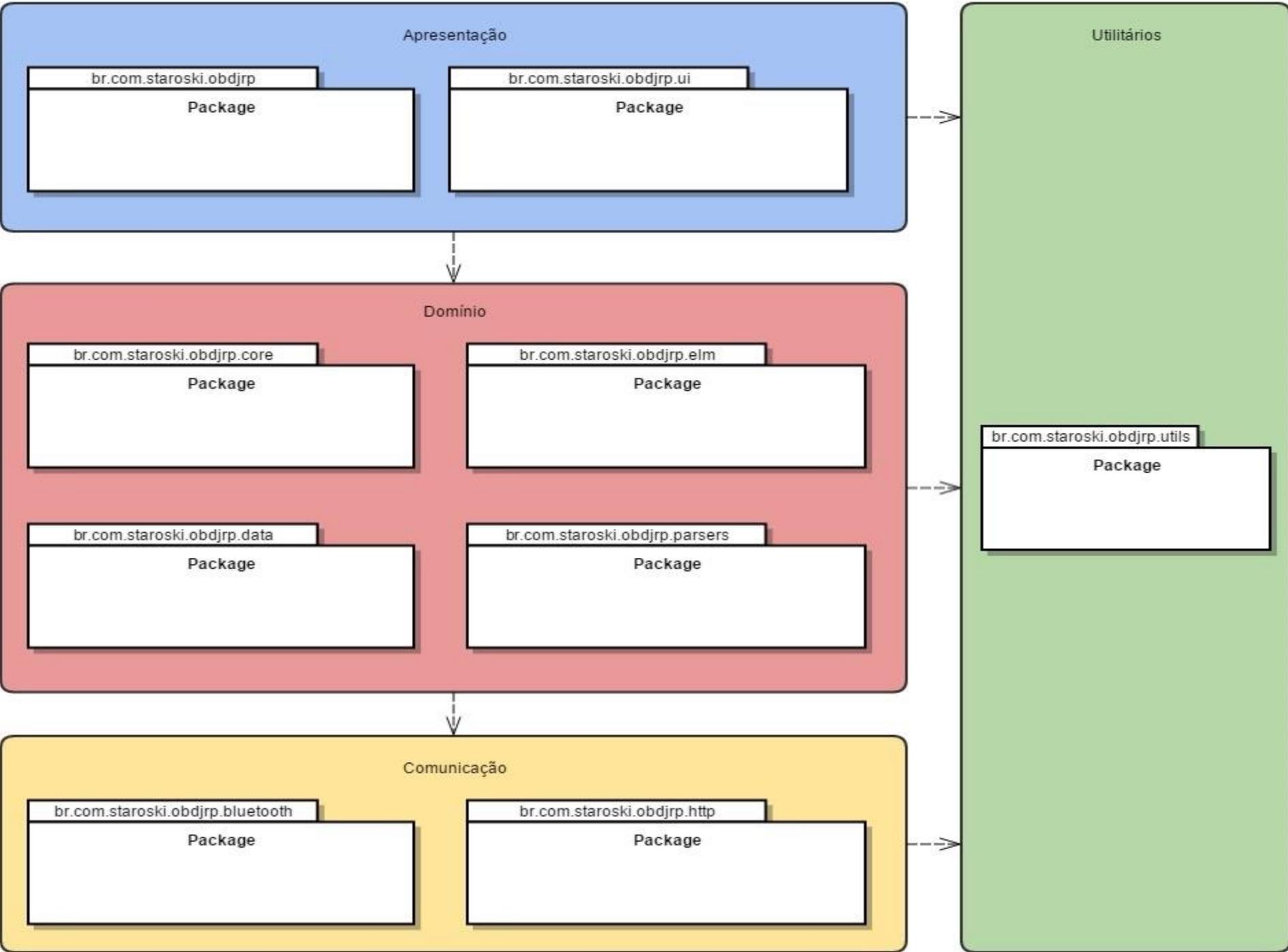
- responder à requisições GET e POST;
- persistir os dados enviados pelo firmware;
- disponibilizar consulta a partir do chassi;
- apresentar gráficos com dados coletados;
- apresentar tabela com dados coletados.

RNF Servidor

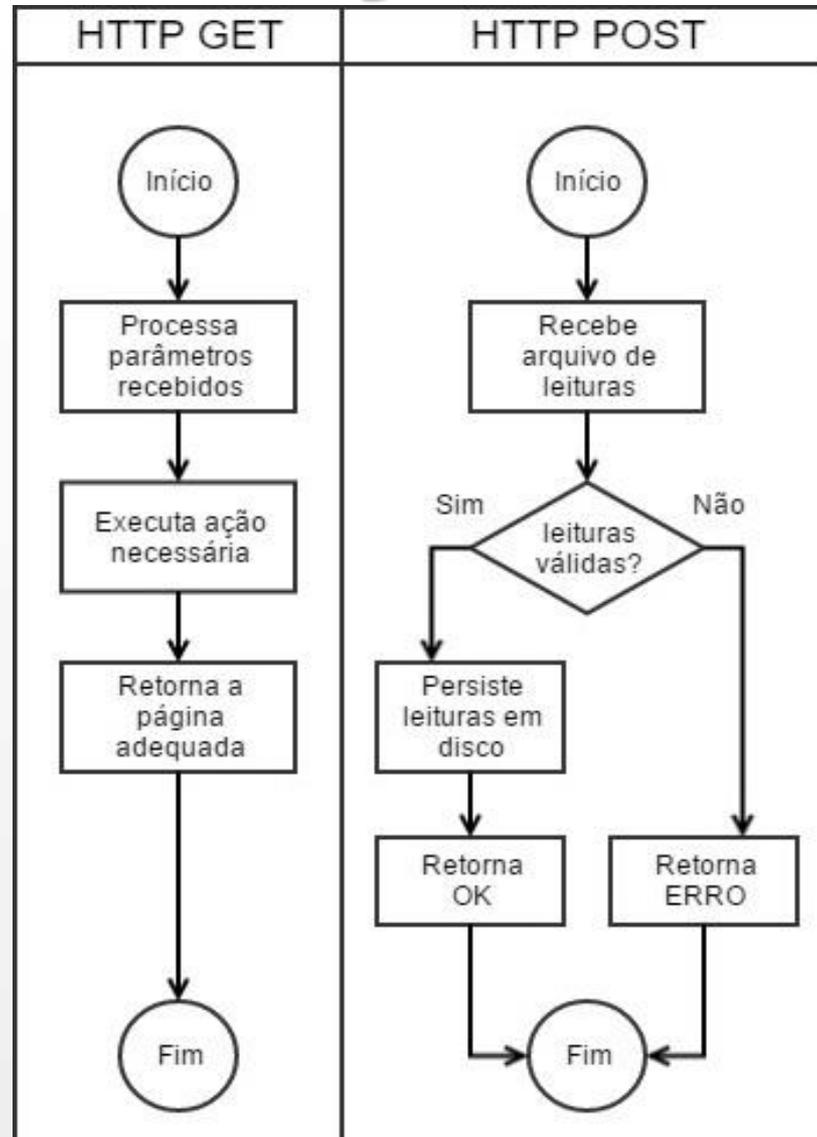
- desenvolvimento em Java EE;
- persistir dados em XML;
- executar no Apache TomCat;
- páginas responsivas;
- páginas em HTML, CSS e JavaScript.

Especificação Firmware



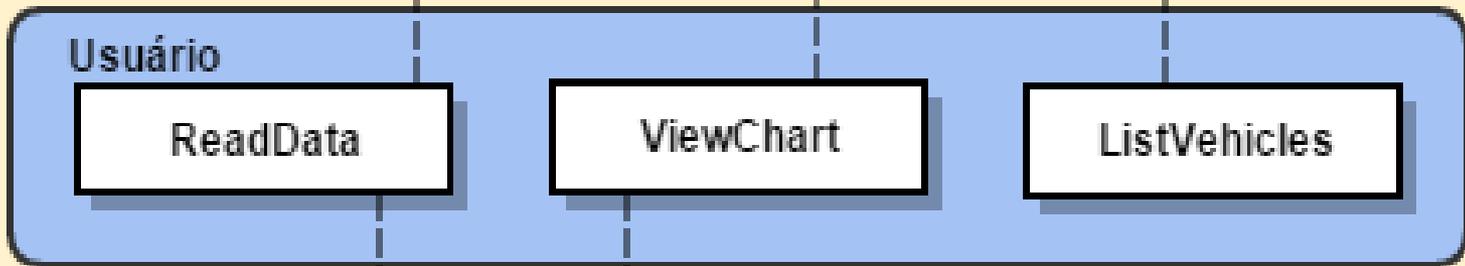
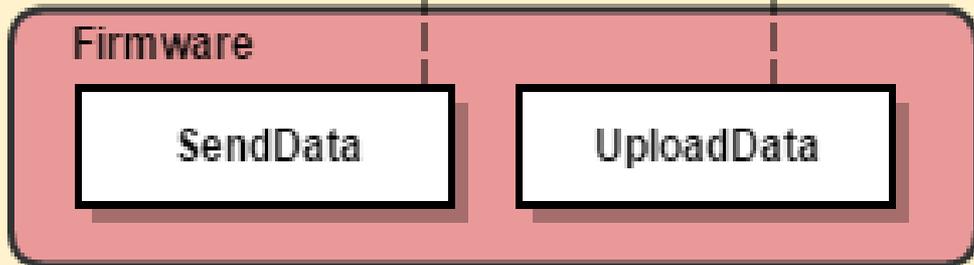


Especificação Servidor



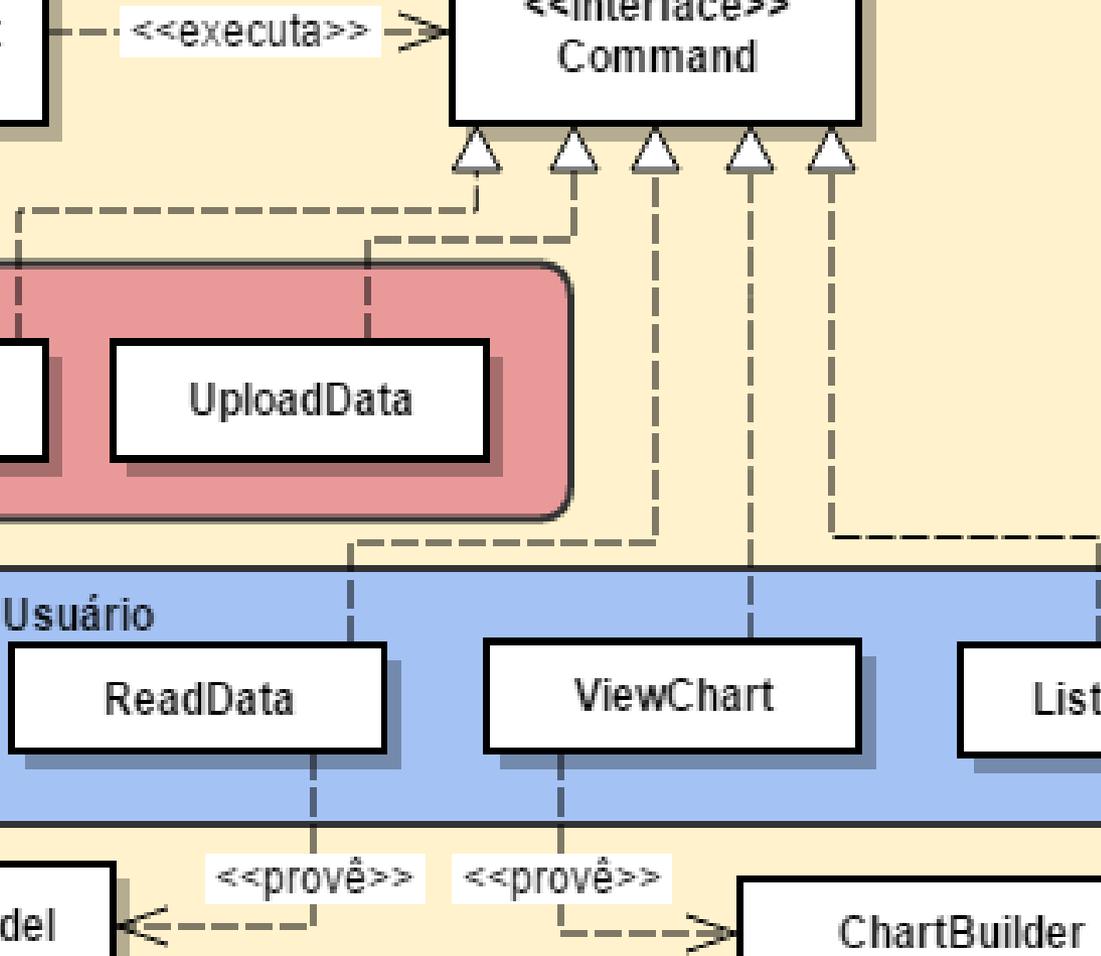


<<executa>>



<<provê>>

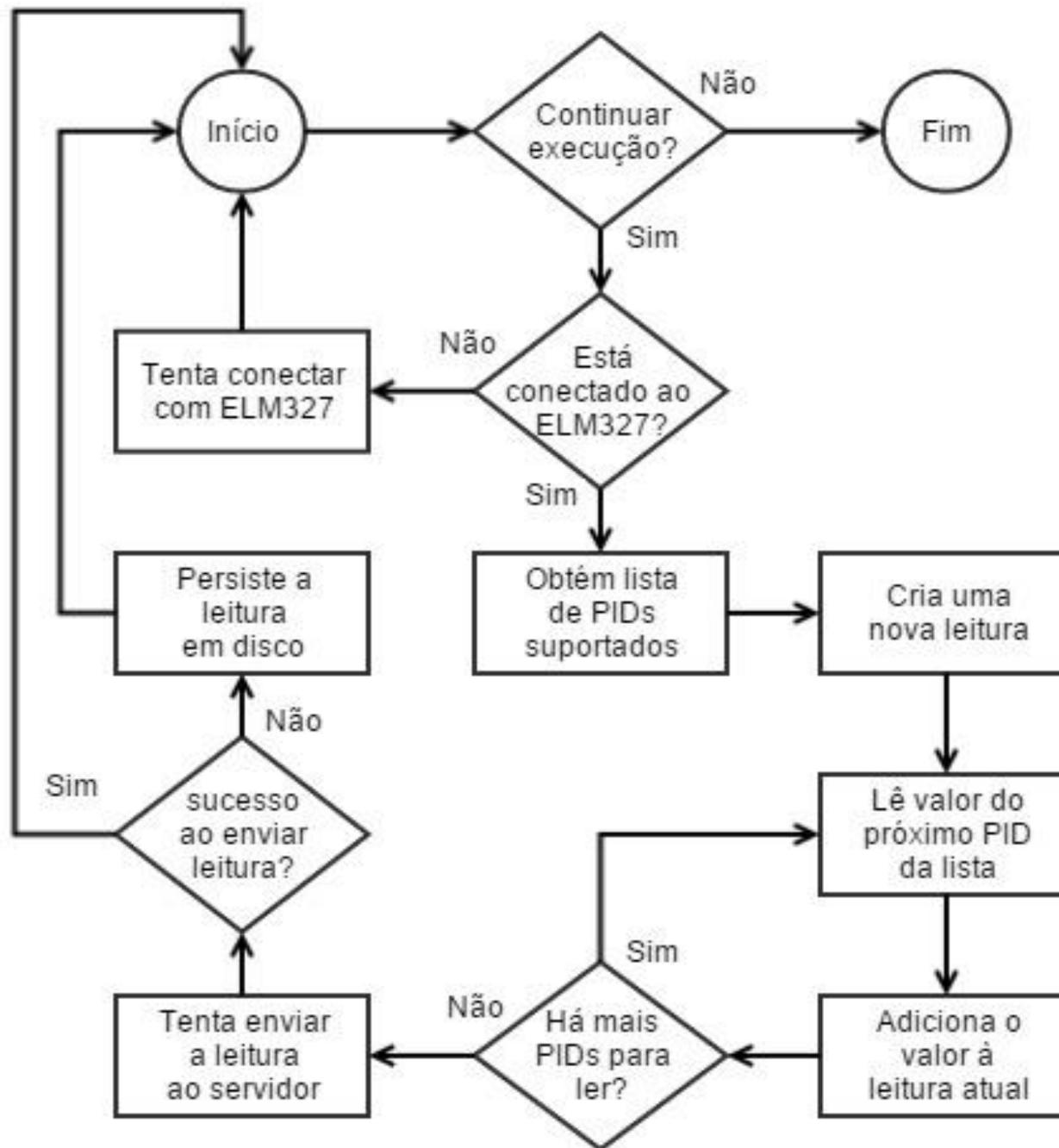
<<provê>>



Implementação Firmware

Implementação Firmware

- Leitura da interface ELM327



Implementação Firmware

- Inicializar ELM327

```
ATZ // reset
```

```
ATE0 // desativar eco
```

```
ATH0 // desativar cabeçalhos
```

```
ATS0 // desativar espaços
```

```
ATSP0 // protocolo automático
```

Implementação Firmware

- PIDs reservados obtém PIDs suportados

01**00** → [01 .. 20]

01**20** → [21 .. 40]

01**40** → [41 .. 60]

01**60** → [61 .. 80]

01**80** → [81 .. A0]

01**A0** → [A1 .. C0]

01**C0** → [C1 .. E0]

01**E0** → [E1 .. FF]

Implementação Firmware

- Exemplo de máscara de bits

0100 → BEBACAFE // 4 bytes

00 = offset

BE	BA	CA	FE
10111110	10111010	11001010	11111110
^	^ ^	^ ^	^ ^
1	8 9	16 17	24 25 32

Implementação Firmware

BE: 10111110 → 1 3 4 5 6 7

BA: 10111010 → 9 11 12 13 15

CA: 11001010 → 17 18 21 23

FE: 11111110 → 25 26 27 28 29 30 31

- $\text{hexa}(\text{dec}(\text{offset}) + \text{índice}) = \text{PID suportado}$

BE: 01 03 04 05 06 07

BA: 09 0B 0C 0D 0F

CA: 11 12 15 17

FE: 19 1A 1B 1C 1D 1E 1F

Implementação Firmware

- Ler RPM (PID 0C)

010C → 2BC0

Table B.13 — PID \$0C definition

PID (hex)	Description	Data byte	Min. value	Max. Value	Scaling/bit	External test equipment SI (Metric) / English display
0C	Engine RPM	A, B	0 min ⁻¹	16383.75 min ⁻¹	1/4 rpm per bit	RPM: xxxxx min ⁻¹

Engine RPM shall display revolutions per minute of the engine crankshaft.

$$\text{dec}(2BC0) = 11200$$

$$11200 / 4 = 2800 \text{ RPM}$$

Implementação Firmware

- Ler velocidade (PID 0D)

010D → 32

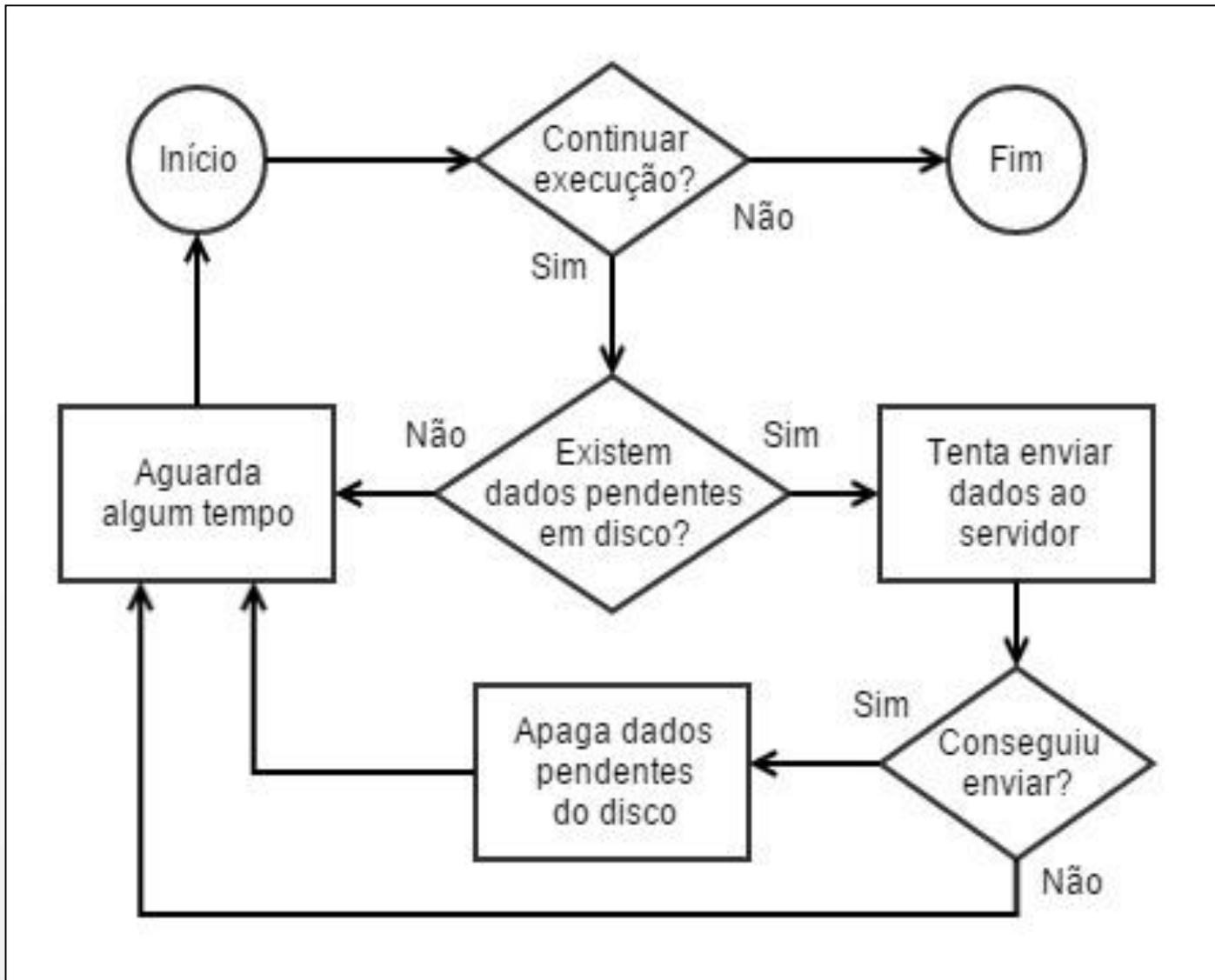
Table B.14 — PID \$0D definition

PID (hex)	Description	Data byte	Min. value	Max. Value	Scaling/bit	External test equipment SI (Metric) / English display
0D	Vehicle Speed Sensor	A	0 km/h	255 km/h	1 km/h per bit	VSS: xxx km/h (xxx mph)
VSS shall display vehicle road speed, if utilized by the control module strategy. Vehicle speed may be derived from a vehicle speed sensor, calculated by the PCM using other speed sensors, or obtained from the vehicle serial data communication bus.						

dec (32) = 50Km/h

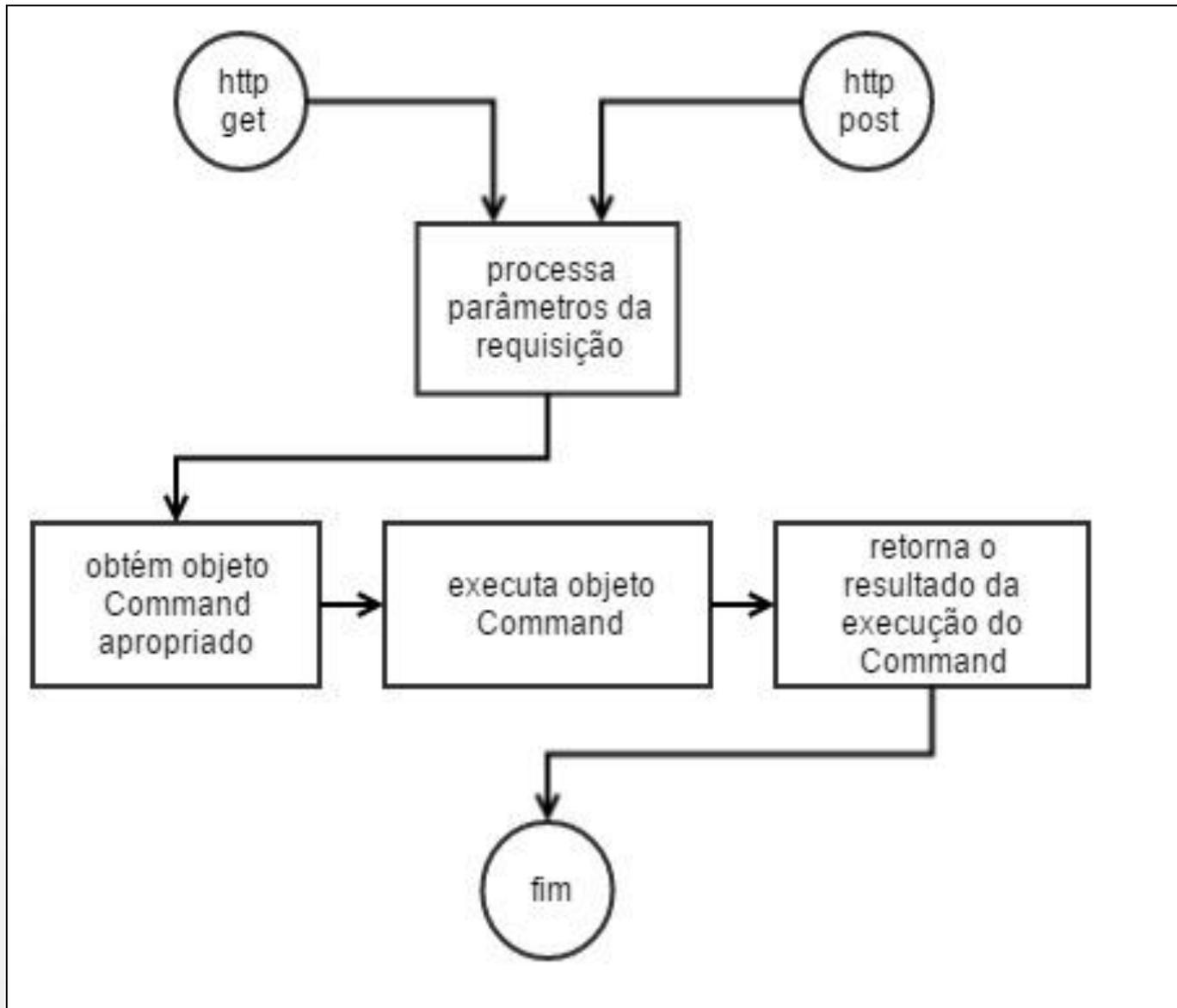
Implementação Firmware

- Envio dos dados pendentes



Implementação Servidor

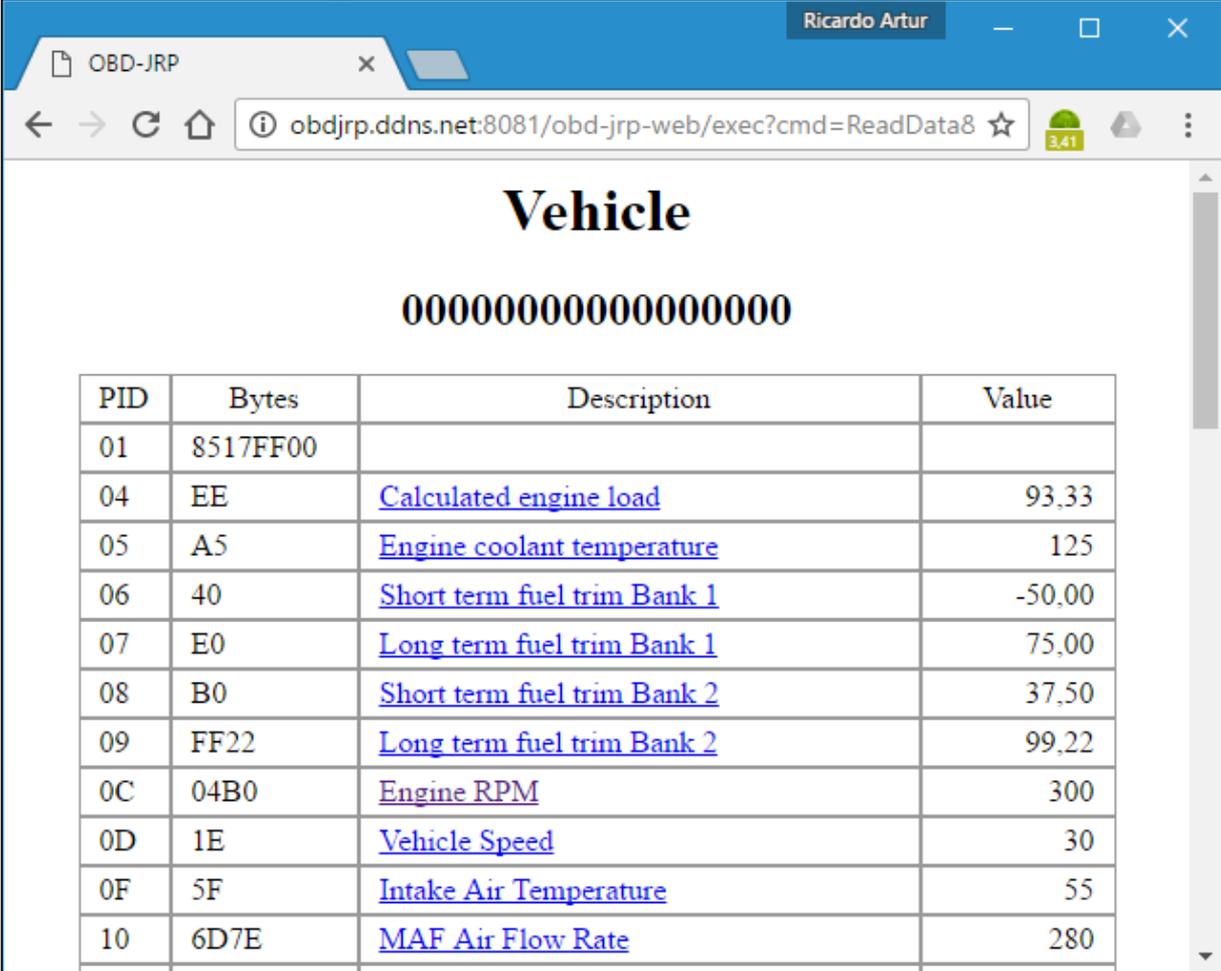
- Processamento das requisições HTTP



Operacionalidade da Implementação



Operacionalidade da Implementação

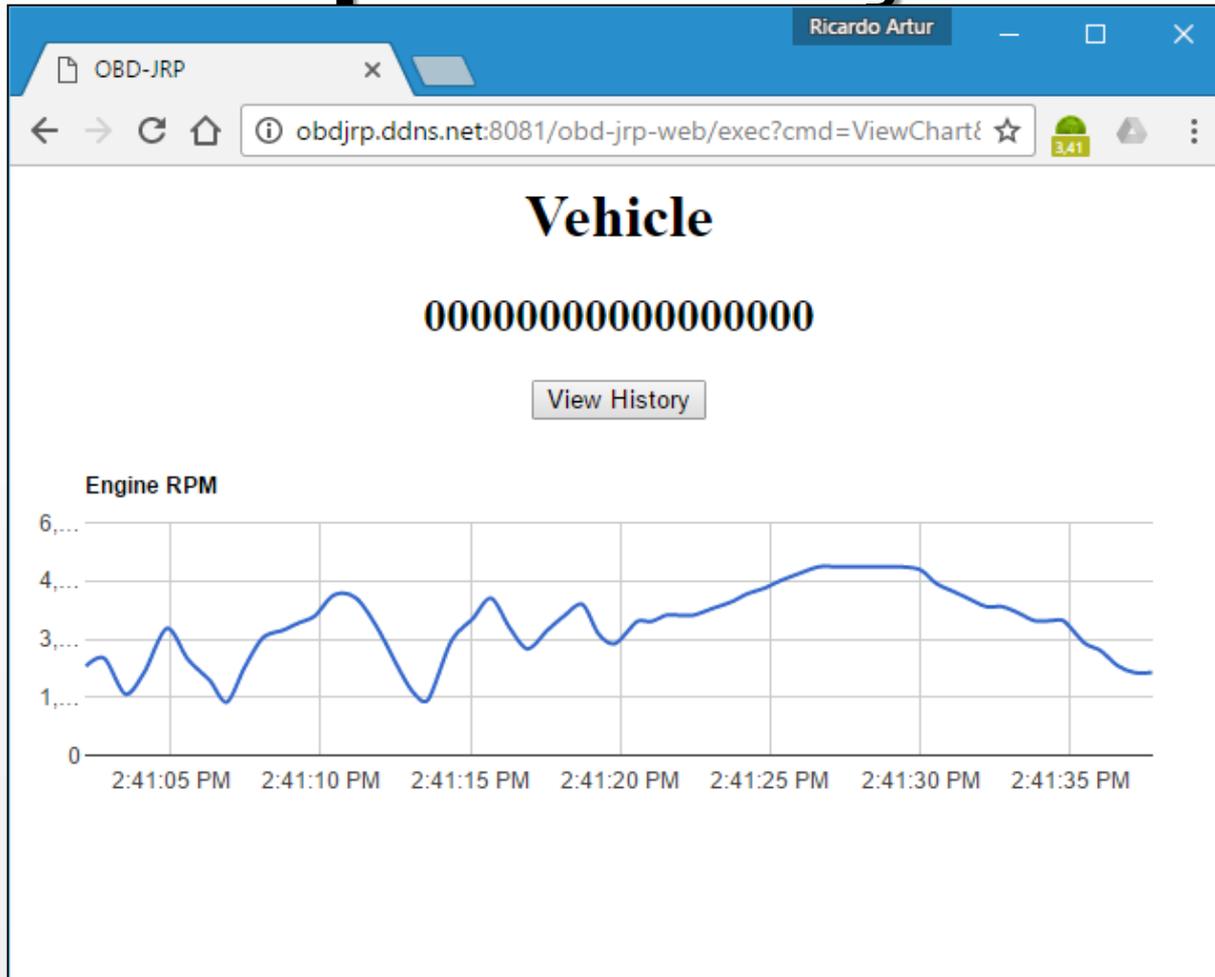


Vehicle

000000000000000000

PID	Bytes	Description	Value
01	8517FF00		
04	EE	Calculated engine load	93,33
05	A5	Engine coolant temperature	125
06	40	Short term fuel trim Bank 1	-50,00
07	E0	Long term fuel trim Bank 1	75,00
08	B0	Short term fuel trim Bank 2	37,50
09	FF22	Long term fuel trim Bank 2	99,22
0C	04B0	Engine RPM	300
0D	1E	Vehicle Speed	30
0F	5F	Intake Air Temperature	55
10	6D7E	MAF Air Flow Rate	280

Operacionalidade da Implementação



Operacionalidade da Implementação

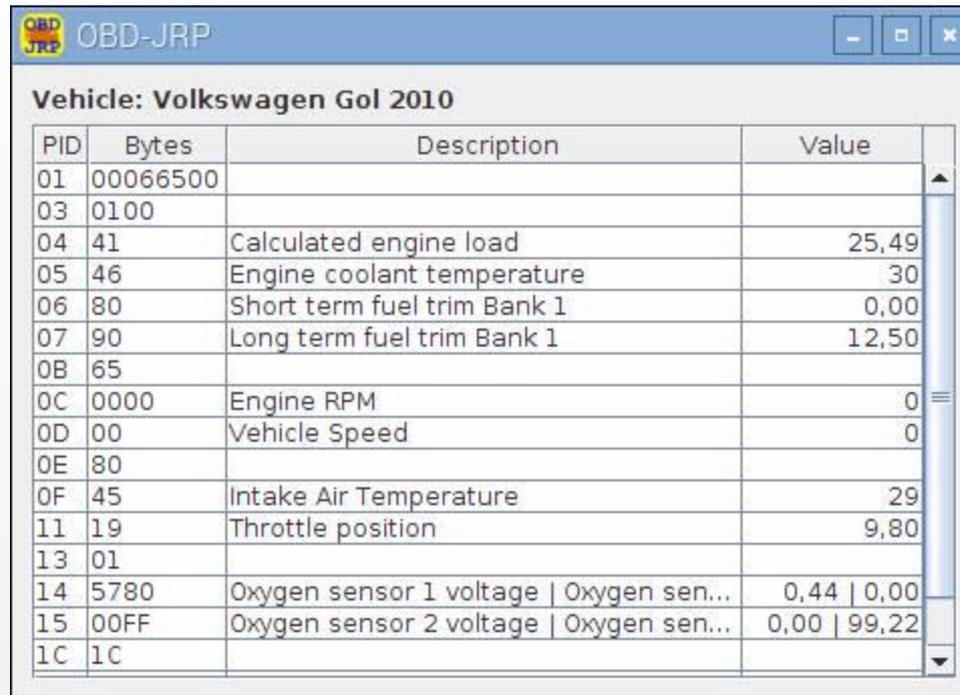
Vehicle

000000000000000000

PID 0C History

Time	Bytes	Value
2016-11-25-14-43-54-046	2FE4	3065
2016-11-25-14-43-53-280	177C	1503
2016-11-25-14-43-52-504	365C	3479
2016-11-25-14-43-51-757	2D20	2888
2016-11-25-14-43-51-153	1344	1233
2016-11-25-14-43-50-372	2B68	2778
2016-11-25-14-43-49-620	317C	3167
2016-11-25-14-43-48-908	0BBC	751
2016-11-25-14-43-48-257	1D28	1866
2016-11-25-14-43-47-576	2114	2260

Operacionalidade da Implementação



The screenshot shows a software window titled "OBD-JRP" with a standard Windows-style title bar. Below the title bar, the text "Vehicle: Volkswagen Gol 2010" is displayed. The main content is a table with four columns: "PID", "Bytes", "Description", and "Value". The table lists various engine and vehicle parameters with their corresponding values.

PID	Bytes	Description	Value
01	00066500		
03	0100		
04	41	Calculated engine load	25,49
05	46	Engine coolant temperature	30
06	80	Short term fuel trim Bank 1	0,00
07	90	Long term fuel trim Bank 1	12,50
0B	65		
0C	0000	Engine RPM	0
0D	00	Vehicle Speed	0
0E	80		
0F	45	Intake Air Temperature	29
11	19	Throttle position	9,80
13	01		
14	5780	Oxygen sensor 1 voltage Oxygen sen...	0,44 0,00
15	00FF	Oxygen sensor 2 voltage Oxygen sen...	0,00 99,22
1C	1C		

Resultados e Discussões

PID	Gol	SpaceFox
01	X	X
03	X	X
04	X	X
05	X	X
06	X	
07	X	
0B	X	X
0C	X	X
0D	X	X
0E	X	X
0F	X	X
11	X	X
13	X	X
14	X	
15	X	
1C	X	X
21	X	X
51	X	X

Resultados e Discussões

- Veículos
 - nem todos implementam OBD2
 - montadoras com “jeitinho” brasileiro
- EnviroCar
 - OBD-JRP → monitoramento
 - EnviroCar → rede social
- PyOBD
 - OBD-JRP → embarcado e autônomo
 - PyOBD → Scanner automotivo no notebook

Conclusões

- Objetivos propostos foram atendidos.
 - Firmware para monitorar veículo;
 - Servidor para recebimento dos dados;
 - Páginas web para consultas.
- Bluetooth
 - Integrado ao Raspberry Pi 3
 - Java: quase uma década sem atualização
 - Python: hobbystas de IoT atualizando

Conclusões

- O Raspberry Pi
 - PC em miniatura;
 - preços de 120 até mais de 500 reais.
- Interface ELM327 Bluetooth
 - instalação simples
 - interface texto
 - preços em torno de 30 reais.

Vantagens

- dimensões reduzidas;
- instalação simples em qualquer veículo que implemente OBD2;
- não depende de interação humana após instalado;
- monitoramento a partir de qualquer computador ou dispositivo móvel com navegador e acesso à internet.

Limitações

- pré-configuração de acesso à internet;
- não possui botão para desligar;
- invólucro indiscreto no veículo;
- não utiliza banco de dados.

Sugestões

- banco de dados;
- sistema cognitivo;
- hardware para desligar;
- computador de bordo;
- integração com GPS.