

UNIVERSIDADE
AUTÓNOMA
DE LISBOA



**DEPARTAMENTO CIÊNCIAS E TECNOLOGIAS
LICENCIATURA INFORMÁTICA DE GESTÃO
2018/2019**

Laboratório de Projeto
3ºAno – 2ºSemestre

Docentes:

**Prof. Dr. Daniel Silvestre
Prof. Dr. Mário Marques da Silva
Prof. Dr. Raúl Dionísio
Profa. Dra. Valéria Pequeno**

**Trabalho sobre:
Gestão de Frotas – Vehicle Diagnostic**

**Professor Orientador:
Prof. Dr. Raul Dionísio**

Carlos Manuel Pancadas, aluno 30000569 PL (LIG)

Duarte Filipe Aguiar, aluno 20160427 PL (LEI)

Rui Guilherme Moinhos e Silva, aluno 19901535 PL (LIG)

Agradecimentos

Ao professor orientador do Laboratório de Projeto, Prof. Dr. Raúl Dionísio por submeter para aprovação um projeto totalmente novo. Pelo apoio na estruturação inicial e fase de desenvolvimento do projeto, orientação pragmática na definição do caminho a escolher para atingir os objetivos a que nos propusemos, escolha da melhor forma de responder aos desafios que surgiram durante a elaboração do projeto e manutenção do foco do projeto na aplicabilidade prática do mesmo.

Aos professores dos cursos de Engenharia Informática e de Gestão pelo papel fundamental na aprendizagem das matérias e competências adquiridas, pelo entusiasmo e sentido de compromisso, nomeadamente:

Prof. Dr. João Vela Bastos, Prof. Dr. António Cabeças, Profa. Dra. Ana Quaresma, Profa. Dra. Isabel Alvarez, Prof. Dr. Arlindo Donário e Prof. Dr. Ricardo Borges dos Santos, Prof. Dr. Gonçalo Valadão, Prof. Dr. Nuno Brás, Prof. Dr. João Guerreiro, Prof. Dr. Rui Neves, Profa. Dra. Patrícia Ferreira, Prof. Dr. Laércio Cruvinel Junior, Prof. Dr. Enes da Silveira e Prof. Dr. Marco Costa, Prof. Dr. António Caldeira, Profa. Dra. Valéria Pequeno e Prof. Dr. Mário Marques da Silva.

Aos colegas de curso de Engenharia Informática e de Gestão, aos parceiros de trabalho pelo entusiasmo, transmissão de conhecimentos, partilha de experiências que nos proporcionaram culminando neste projeto final de curso.

*“Don't be encumbered by history.
Go off and do something wonderful.”
Robert Noyce*

Resumo

O presente documento descreve o trabalho de grupo desenvolvido no âmbito da disciplina Laboratório de Projeto, estando organizado em 14 capítulos.

Começa com uma introdução para explicar o âmbito, motivação e objetivos deste projeto.

Em seguida seguem-se os capítulos que percorrem a metodologia escolhida que tipicamente se usa em projetos de implementação de sistemas de informação.

No capítulo 2 ilustramos o uso do Microsoft Project para a gestão deste projeto, que serviu de base para todas as atividades planeadas.

No capítulo 3 aborda a fase de levantamento de requisitos funcionais e não-funcionais, seguindo-se no capítulo 4 os detalhes sobre a fase de análise e desenho técnico da solução.

Neste capítulo está descrita a solução end-to-end desde a recolha de dados ao tratamento da informação na consola aplicacional de apoio à decisão, que passaremos a designar por dashboard DSS (Decision Support System). Aqui descrevemos também o sistema interno, incluindo a infraestrutura de suporte, aos servidores de base de dados e serviço FTP.

No capítulo 5 descrevemos a fases de desenvolvimento e implementação da solução proposta pelo nosso projeto onde descrevemos as configurações necessárias para ativar o tratamento de dados na base de dados e subsequente utilização por parte do gestor através do dashboard DSS. Neste capítulo ilustramos os diversos painéis de utilização da aplicação disponíveis para o gestor de frotas autorizado, que assumem um papel muito importante neste projeto.

No capítulo 6 percorre a fase de testes funcionais, integração e de aceitação que permitiram colocar a solução operacional para a fase seguinte descrita no capítulo 7, que contempla a infraestrutura de produção.

No capítulo 8 identificamos as principais ferramentas e tecnologias usadas no decorrer deste projeto, no capítulo 9 colocamos os custos do projeto e no capítulo 10 ditamos a metodologia de trabalho

Finalmente no capítulo 11 apresentamos as nossas conclusões sobre o projeto e onde descrevemos as linhas gerais para desenvolvimento futuro tendo em conta o grande potencial desta solução.

Palavras-chave

Dashboard, DSS, Gestão, ELM327, Viatura, Gestor, Solução, Diagnostico

Conteúdo

<i>Agradecimentos</i>	<i>2</i>
<i>Resumo.....</i>	<i>3</i>
<i>Palavras-chave.....</i>	<i>5</i>
<i>Lista de Figuras/Imagens</i>	<i>8</i>
<i>1. Introdução</i>	<i>9</i>
<i>1.1. Motivação</i>	<i>9</i>
<i>1.2. Objetivos</i>	<i>11</i>
<i>2. Metodologia de Gestão do Projeto.....</i>	<i>12</i>
<i>3. Requisitos</i>	<i>14</i>
<i>3.1. Requisitos Não-Funcionais</i>	<i>14</i>
<i>3.2. Requisitos Funcionais.....</i>	<i>16</i>
<i>4. Análise Funcional e Desenho Técnico</i>	<i>18</i>
<i>4.1. Diagrama de Fluxo de Dados.....</i>	<i>18</i>
<i>4.2. Modelação e Base de Dados</i>	<i>20</i>
<i>4.3. Identificação de pressupostos</i>	<i>20</i>
<i>4.4. Modelo Entidade Relacionamento</i>	<i>21</i>
<i>4.5. Conversão do modelo lógico em modelo físico</i>	<i>22</i>
<i>4.6. Modelo Relacional</i>	<i>24</i>
<i>5. Desenvolvimento e Implementação</i>	<i>26</i>
<i>5.1. Tratamento de Dados.....</i>	<i>26</i>
<i>5.2. Consola Aplicacional – Dashboard DSS.....</i>	<i>31</i>
<i>6. Testes e Resultados.....</i>	<i>40</i>
<i>6.1. Testes funcionais, integração e de aceitação</i>	<i>40</i>
<i>6.2. Testes Aplicacionais no dashboard DSS</i>	<i>43</i>
<i>7. Infraestrutura de Produção</i>	<i>44</i>

7.1. Características do Servidor com SQL Server.....	44
7.2. Características do Servidor com FTP e DSS	45
8. Tecnologias e Ferramentas	47
9. Custos do Projeto	48
9.1. Custos gerais	48
10. Metodologia do Trabalho.....	49
10.1. Características da Equipa de Trabalho.....	49
11. Conclusões e Trabalho Futuro.....	50
11.1. Conclusões.....	50
11.2. Trabalho Futuro	51
Glossário de Termos.....	53
Referências Bibliográficas	55
Apêndices	57
Apêndice A – Lista de Indicadores de Leitura de Sensores da Viatura	57
Apêndice B – Scripts SQL	58
Apêndice C - Dispositivo ELM327.....	61

Lista de Figuras/Imagens

- Figura 2.1 – Metodologia de Gestão do Projeto
- Figura 2.2 – Tarefas do Projeto
- Figura 4.1 – Diagrama Principal do Fluxo de Dados
- Figura 4.2 – Data Center
- Figura 4.3 – Modelo Entidade Relacional
- Figura 4.4 – Modelo Entidade Relacional
- Figura 4.5 – Principais Indicadores da Viatura
- Figura 5.1 – Execução Scripts Automáticos
- Figura 5.2 – Eliminação Tabelas Temporárias
- Figura 5.3 – Processo Importação dos Dados
- Figura 5.4 – Carregamento dos Dados Finais
- Figura 5.5 – Diagrama de Manutenção da Base de Dados
- Figura 5.6 – Estrutura Funcional dashboard DSS
- Figura 5.7 – Login na Aplicação DSS
- Figura 5.8 – Menu Principal
- Figura 5.9 – Opções Menu Principal
- Figura 5.10 – Opções Perfil
- Figura 5.11 – Alertas Veículo
- Figura 5.12 – Criação Nova Viatura
- Figura 5.13 – Criação Novo Utilizador
- Figura 6.1 – Dispositivo ELM327
- Figura 6.2 – Dispositivo ELM327 Instalado na Viatura
- Figura 6.3 – Registos Dados das Viaturas
- Figura 7.1 – Características Servidor SQL Server
- Figura 7.2 - Características Servidor FTP e DSS
- Figura 7.3 – Paineil Filezilla Server
- Figura 7.4 – Versão FileZilla Server

1. Introdução

Neste capítulo são apresentadas motivações e os objetivos que estão na génese do desenvolvimento deste trabalho para a disciplina Laboratório de Projeto.

1.1. Motivação

Contendo a lista apresentada pelos Professores, que iriam orientar diretamente os diversos trabalhos a realizar para a disciplina de Laboratório de Projeto, vários projetos interessantes e que versam algumas das áreas que suportam a licenciatura de Engenharia Informática e a licenciatura de Informática de Gestão, optámos por uma abordagem diferente da maioria dos nossos colegas e levámos o nosso desafio mais além com a escolha de um projeto novo, assente numa base prática e com eventual aplicação comercial futura.

Este desafio na escolha levou-nos numa discussão de ideias que culminou com a proposta para a criação de um sistema integrado de gestão de frotas automóveis, que permite não só responder às questões básicas associadas às aplicações de gestão de frota já existentes no mercado, mas também acrescentando alguns fatores diferenciadores destas, que explorando a possibilidade de recolha de dados remota, permitem a monitorização de diversos indicadores de funcionamento das viaturas e posterior tratamento analítico recorrendo modelos de inteligência artificial.

Neste sentido fizemos um breve estudo de mercado, procurando soluções já existentes nesta área e avaliando os seus principais benefícios, mas sobretudo identificando as suas eventuais lacunas para que de acordo com a análises dos dados recolhidos nos permitissem desenvolver uma solução chave na mão, que fossem possível implementar em qualquer entidade que necessite de informação para tornar mais eficiente e económica e a gestão da sua frota automóvel.

Esta abordagem que foi efetuada no desenvolvimento desta solução tem em consideração os custos duma empresa, de média ou grande dimensão, associados à gestão da sua frota automóveis, no que respeita à utilização, indicadores de controlo e manutenção, que afetam e condicionam de forma significativa o desempenho das empresa, absolvendo recursos humanos e financeiros que, de outra forma poderiam ser canalizados para o desenvolvimento e criação de riqueza da própria empresa .

Assim esta solução permite às empresas gerir as suas frotas automóveis com eficiência, sendo para isso apenas necessário proceder à instalação de pequenos dispositivos nas viaturas que possibilitam a recolha de alertas e de outros dados relevantes para apoio à decisão, otimização da utilização e manutenção preventiva e corretiva.

1.2. Objetivos

Este projeto tem por âmbito a criação de um sistema integrado de gestão e diagnóstico de viaturas automóveis, através de recolha de diversos dados estáticos e dinâmicos das viaturas, utilizando para o efeito um dispositivo ELM327® ligado na tomada de diagnóstico OBD2 e um dispositivo móvel com acesso Bluetooth, GPS e rede móvel para partilha de dados.

Associando a recolha de dados estáticos e dinâmicos das viaturas, provenientes dos diversos sensores existentes (motor, caixa, acelerador, travão, etc) e do histórico de dados armazenados em sistema, a aplicação permite gerar indicadores para a gestão e tratamento de frota de automóveis, auxiliando os decisores e os responsáveis pela gestão das frota de automóveis das empresas, na melhoria da tomada de decisão relativa às necessidades de adequação da manutenção preventiva, corretiva das viaturas da sua frota, mas também na determinação de perfis de condução dos vários utilizadores das mesmas.

Os objetivos principais deste projeto são dos seguintes:

- 1) Criação de histórico para cada viatura da frota incluindo perfil de utilização;
- 2) Apoiar na definição de critérios de manutenção preventiva e corretiva;
- 3) Mapa de Gestão -Dashboard para DSS - Decision Support System

2. Metodologia de Gestão do Projeto

Na gestão deste projeto de Laboratório recorremos à aprendizagem obtida na disciplina de Gestão de Projetos, disciplina lecionada pelo Prof Dr. António Cabeças no primeiro semestre do 3ºAno e na respetiva documentação, que detalhamos no índice deste relatório, de acordo com o definido no PMBoK® (Project Management Body of Knowledge) que nos permitiu sistematizar regras e recomendações a seguir no decorrer da execução do projeto, contribuindo a para obtenção de melhores resultados e também para o sucesso do mesmo.

Recorremos aos princípios elementares da metodologia, tipicamente usada para implementação de sistemas de informação e que compõe as seguintes etapas:

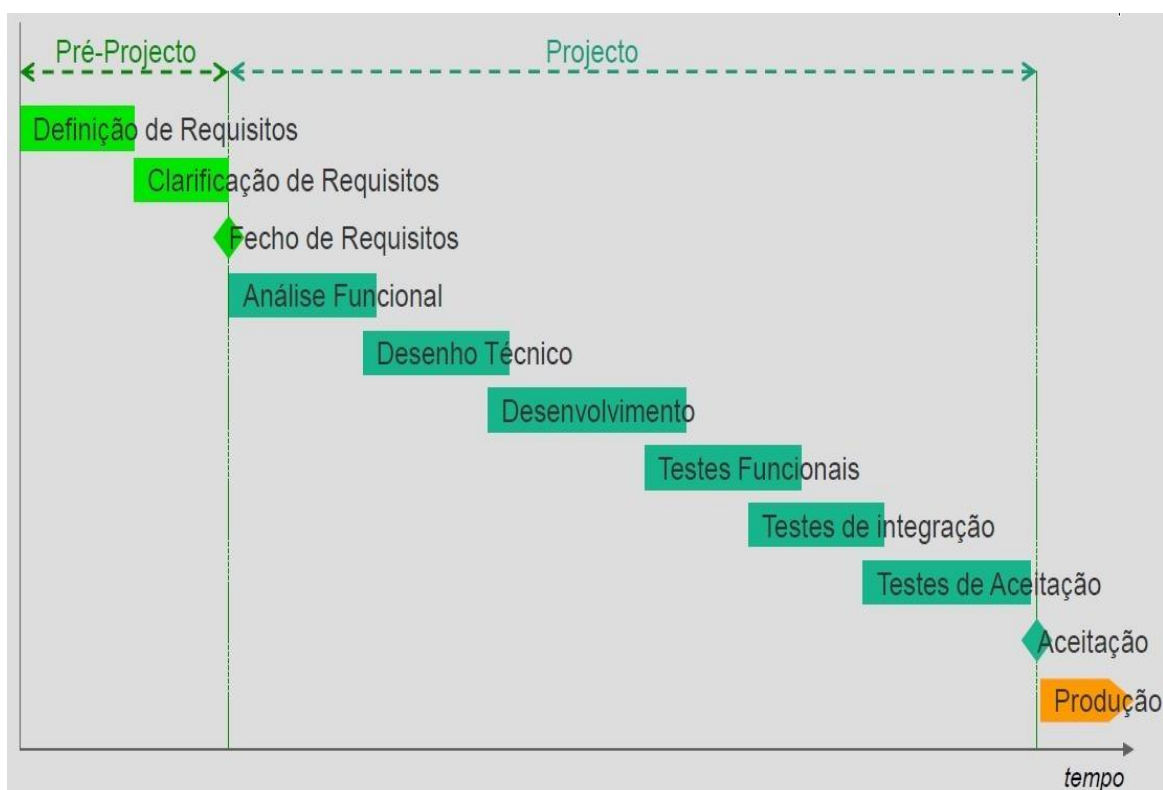


Figura 2.1 – Metodologia de Gestão do Projeto

Para gerir o desenvolvimento deste projeto recorreremos ao programa Microsoft Project® que se tornou o fio condutor da identificação das principais tarefas e da atribuição temporal deste projeto.

Assim as principais atividades estão definidas com as seguintes etapas (incluindo respetivos períodos temporais) que na sua maioria foram cumpridos:
















	Modo de	Nome da Tarefa	Duração	Início	Conclusão
✓		Âmbito do projeto	6 dias	Qua 06-03-19	Qua 13-03-19
✓		▸ Definição de requisitos macro	14 dias	Qua 13-03-19	Seg 01-04-19
✓		Clarificação de requisitos	5 dias	Seg 01-04-19	Sex 05-04-19
✓		Definição de requisitos funcionais	6 dias	Sex 05-04-19	Sex 12-04-19
✓		Fecho de requisitos	2 dias	Sex 12-04-19	Seg 15-04-19
✓		Ánalise funcional	2 dias	Qui 18-04-19	Sex 19-04-19
✓		Desenho técnico	6 dias	Sex 19-04-19	Sex 26-04-19
✓		Desenvolvimento	31 dias	Sex 26-04-19	Sex 07-06-19
✓		Testes funcionais	6 dias	Sex 07-06-19	Sex 14-06-19
✓		Testes de integração	6 dias	Sex 14-06-19	Sex 21-06-19
✓		Testes de aceitação	7 dias	Sex 21-06-19	Dom 30-06-19
		Fecho de projeto e Relatório	2 dias?	Dom 30-06-19	Seg 01-07-19
		Produção		Seg 01-07-19	
					

Figura 2.2 – Tarefas do Projeto

3. Requisitos

Definido o âmbito e os objetivos do projeto, tornou-se necessário efetuar o levantamento de requisitos, tendo nós optado por separar o levantamento de requisitos funcionais e não-funcionais. Tivemos em conta na definição dos requisitos macro, que seria imperioso ir de encontro à satisfação de necessidades definidas para este projeto, nomeadamente na satisfação dos objetivos estabelecidos no Capítulo 1, secção 1.2.

- Os requisitos funcionais compreendem uma ligação entre o sistema implementado e os elementos que o rodeiam.
- Os requisitos não funcionais procuram identificar as respetivas restrições e filtros no sistema implementado

3.1. Requisitos Não-Funcionais

Por se tratar de uma aplicação desenvolvida de raiz que irá integrar dados de várias viaturas, obtidos através de um serviço FTP externo (internet). Os dados são, recolhidos, tratados e armazenados num servidor de base de dados SQL e posteriormente utilizados no dashboard front-end da aplicação de gestão de frota automóvel.

Procurámos responder às questões fundamentais para avaliação dos componentes de software:

1. Quais as oportunidades de colaboração para uma solução desta natureza (parcerias, integração);
2. Que tipos de soluções software estão disponíveis; (Base de Dados, Servidores, Programação);
3. Analisar os custos e benefícios dos produtos disponíveis (Back End e Front-End);
4. Avaliar os riscos de segurança (proteção de dados);
5. Consultar potenciais utilizadores sobre métodos de implementação e utilização;
6. Avaliar fornecedores de produtos;
7. Que componentes estão acessíveis no mercado e quais nos permitem uma implementação com o menor custo possível para;
8. O tempo disponível para o desenvolvimento e implementação do projeto;

As respostas a estas questões que se traduzem, em última análise, nas escolhas efetuadas. Basearam-se nas necessidades decorrentes deste projeto, nomeadamente o objetivo de implementar uma solução para comportar a estrutura de fluxo de dados, desde a sua recolha (na viatura automóvel), à apresentação em modo de gestão (aplicação de software) que identificaram, em cada instância, os respetivos elementos de integração, bem como a respetiva arquitetura. Assim as escolhas foram as seguintes:

- Dispositivo para recolha de dados instalado no veículo via tomada OBDII®
- Software para recolha e envio de dados do veículo – Torque®
- Dispositivo para serviço web/FTP do veículo via rede móvel – S.O. Android
- Software para Criação da Base de Dados – SQL®
- Software para desenvolvimento da aplicação de gestão da frota – Java®, Netbeans®

3.2. Requisitos Funcionais

Era intenção do grupo de trabalho centrar o âmbito do projeto na recolha, tratamento e armazenamento dos dados das viaturas, através da criação de um dispositivo para colocação nas viaturas, que permitiria o acesso à informação disponibilizada pela tomada OBD2, transmissão e armazenamento automatizado dessa informação, devidamente tratada e organizada, numa base de dados que posteriormente pudesse ser acedida por uma das várias aplicações de gestão de frota existentes atualmente no mercado.

Decorrente da análise de mercado não foi determinada uma solução existente, disponível para a nossa utilização, que servisse de base para o desenvolvimento da componente de gestão de frotas, com os objetivos descritos no Capítulo 1.2.

Apesar do número considerável de aplicações genéricas para gestão de frotas automóveis existentes à data da definição dos requisitos do projeto, na análise efetuada não nos foi dado a conhecer uma aplicação que integrasse os componentes de perfil de condutor, manutenção preventiva, manutenção corretiva e nível de personalização de indicadores decorrentes da leitura dos sensores existentes nos novos automóveis e cujo acesso é efetuado através da tomada OBD2. Nos Estados Unidos da América todas as viaturas posteriores a 1996 têm obrigatoriamente que disponibilizar este tipo de tomada standard, para acesso diagnóstico da viatura. Na Europa esta obrigação apenas foi implementada em 2001, para todas as viaturas a gasolina e em 2003 para todas as viaturas a gasóleo.

A solução deverá ter capacidades de integração de dados numa consola de gestão, que passaremos a designar de dashboard DSS, que irá permitir ao gestor da frota automóvel identificar perfis de utilizadores, viaturas, níveis de alerta, definir medidas de manutenção preventiva e corretiva.

Aliás este fator foi determinante e permite resolver algumas das lacunas identificadas e que se traduzem numa oportunidade de negócio que prontamente identificaremos Capítulo 12 no nosso relatório.

Assim ajustamos o âmbito o projeto, para que o mesmo tivesse uma abrangência que permitisse a automação da criação e gestão de um fluxo de informação End-to-End, entre as viaturas com o dispositivo de recolha de dados de diagnóstico instalado e os gestores de frota que remotamente controlam o funcionamento dos vários sistemas da viatura, nomeadamente os que poderão se não forem adotadas mediadas imediatas por parte da gestão de frota, levar a eventuais, paragens, funcionamentos erráticos e, em casos extremos, danos permanentes nas viaturas, sobre sua responsabilidade.

Enquadramos estes requisitos funcionais numa ótica de Sistemas de Apoio à Decisão (DSS- Decision Support Systems) que tipicamente podem servir uma área de gestão departamental com a relevância de dados para os devidos efeitos.

4. Análise Funcional e Desenho Técnico

4.1. Diagrama de Fluxo de Dados

Nesta fase definimos como diagrama principal do fluxo de dados representado na seguinte figura:

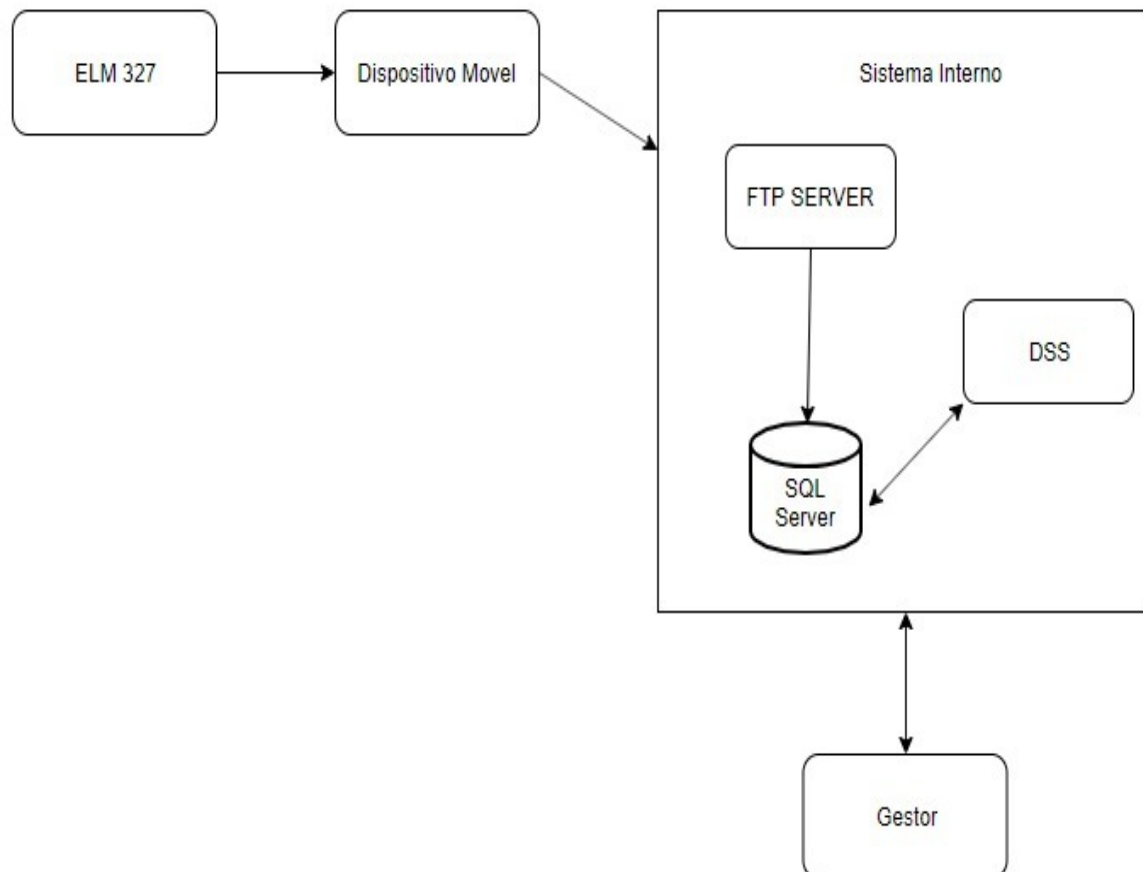


Figura 4.1 – Diagrama Principal do Fluxo de Dados

Descrição de cada um dos elementos, do diagrama de fluxos de dados da imagem 4.1 que integram a solução:

1. **Dispositivo ELM327®** - ligado na tomada universal **OBDII®** para diagnósticos do veículo automóvel que permite a recolha de dados de uma vasta gama de sensores presentes. Este dispositivo é emparelhado via Bluetooth® com um Dispositivo Móvel.
2. **Dispositivo Móvel** – Telemóvel com sistema Operativo Android® usado, recorrendo a uma aplicação de software móvel (Torque®), para recolher os dados provenientes do dispositivo ELM327®, em formato *.csv que são enviados, via GPRS, para um Servidor FTP.
3. **Sistema Interno** – Composto por 3 elementos: um servidor FTP, um servidor SQL e um servidor Aplicacional:
 - a. Servidor FTP: Tem como função principal receber a informação proveniente dos dispositivos externos, nomeadamente do dispositivo móvel;
 - b. Servidor SQL: Vai buscar os dados à diretoria FTP, para alimentar a sua Base de Dados que tem definidas rotinas de processamento da informação que serão posteriormente acedidas pelo Servidor Aplicacional;
 - c. Servidor Aplicacional: Acede aos dados processados pela Base de Dados apresentando-os através de écrans interativos e listas de relatórios, compostos num dashboard DSS aplicacional que será acedido por um gestor de frotas com acesso autorizado;



Figura 4.2 – Data Center

4.2. Modelação e Base de Dados

Tal como definido na fase de levantamento de requisitos foi necessário recorrer à implementação de um Servidor de Base de Dados, para o qual utilizamos os conhecimentos adquiridos nas disciplinas de Modelação de Base de Dados e Aplicação de Base de Dados, lecionadas pelo Prof. Dr. Marco Costa e Profa. Dra. Valéria Pequeno.

4.3. Identificação de pressupostos

Com base na análise das necessidades identificadas e considerando os pressupostos a seguir descritos, foi elaborado um modelo Entidade Relacionamento, a partir de agora designado como Modelo ER, com uma possível solução para implementação de uma base de dados

Pressupostos

1. Consideramos que, para uma aplicação de gestão de frotas automóveis de uma empresa apenas um utilizador pode ser assignado a uma viatura específica.
2. Pode haver veículos sem utilizador assignado, sendo o campo “ativo” colocado a zero.
3. O número mínimo de veículos automóveis para constituição de uma frota é 2.
4. Um veículo pode mudar de utilizador, sempre que a empresa o entenda.
5. Para o registo de uma nova viatura, assignada a um utilizador, é necessário registar o dispositivo com o novo identificador na Base de Dados.
6. Deve existir a opção de criar, alterar e apagar utilizadores.
7. Deve existir a opção de mudar utilizador de uma viatura.
8. Deve existir a opção de criar e apagar viaturas.
9. Os dados recolhidos pelo dispositivo ELM232® via OBDII® devem ter tratados recorrendo à criação de tabelas temporárias, uma para cada viatura, devido à inexistência de um identificador que diferencie o ficheiro *.csv de cada veículo correspondente.
10. A tabela Viatura deve conter os campos IDVeículo, matrícula, marca, modelo, cilindrada, combustível, ativo, e obd2_ID.
11. A tabela Utilizador deve conter os campos nome, data _nascimento, género e doc_identificativo.
12. Deverá criada uma tabelas TORQUE que resultada da obtenção dos dados recolhidos pelo ficheiro *.csv que contém até 3500 indicadores, provenientes dos dados emitidos por cada um dos sensores da viatura.

4.4. Modelo Entidade Relacionamento

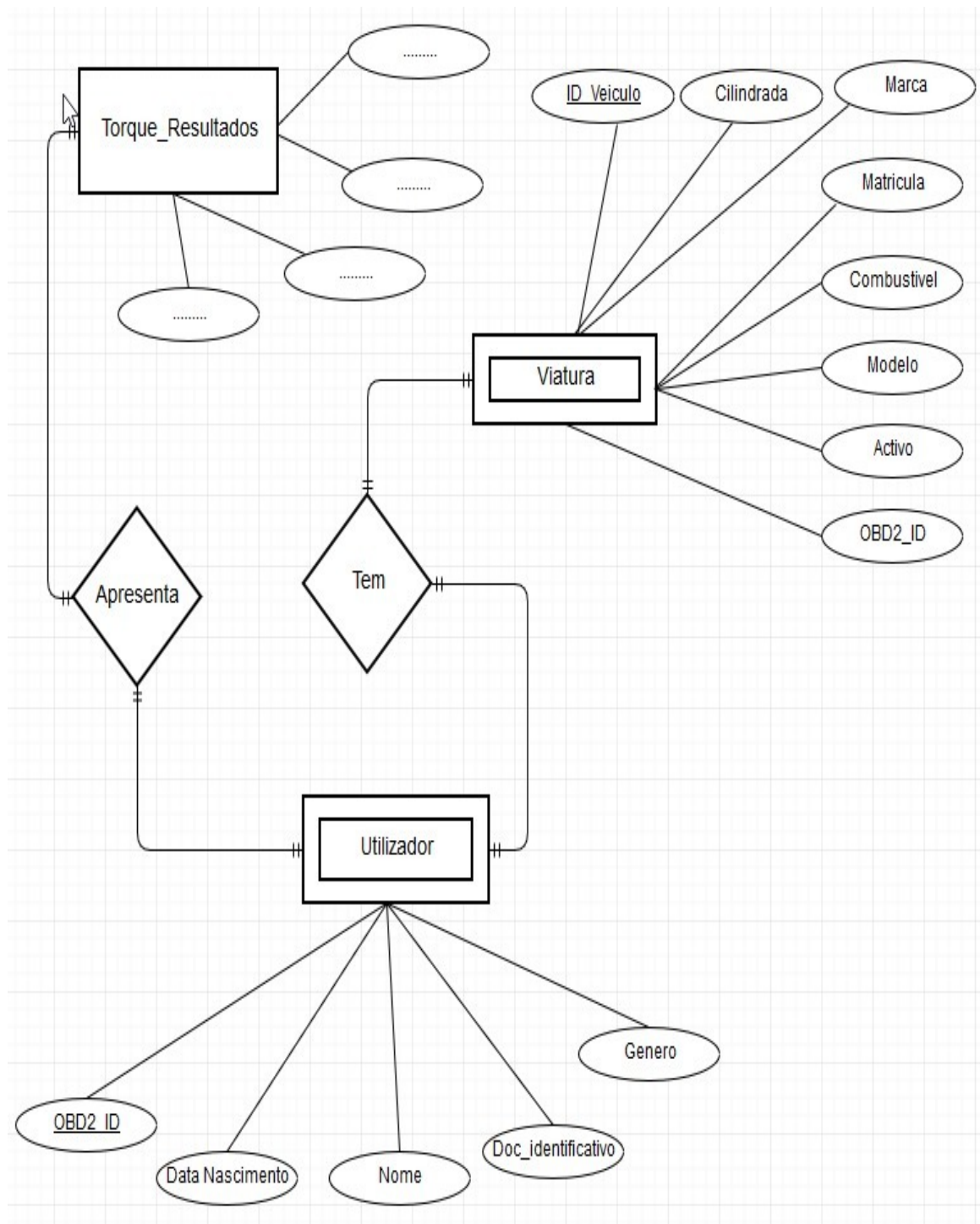


Figura 4.3 – Modelo Entidade Relacional

4.5. Conversão do modelo lógico em modelo físico

Para efetuarmos a passagem do Modelo ER para o modelo relacional, exemplificado na imagem do ponto anterior, foram tomados os seguintes passos:

Passo 1

Para cada entidade forte criar uma tabela, convertendo os atributos identificadores em chave primária e os restantes atributos em campos, excetuando os atributos multivalor (se existirem – sendo que no nosso desenho não foi necessário)

Utilizador (obd2_id, nome, data_nascimento, genero, doc_identificador)

Viatura (IDVeiculo, matricula, marca, modelo, cilindrada, combustível, activo, obd2_id)

Passo 2

Para cada entidade fraca criar uma tabela convertendo os atributos identificadores em chave-primária, em conjunto com a chave-primária da tabela relacionada

Passo 3

Para os relacionamentos 1:1, o atributo identificador de uma das entidades, bem como os atributos do relacionamento, é levado para a outra entidade. Considerámos a boa prática de levar os atributos para a entidade que tem a obrigatoriedade da relação, caso haja.

No nosso modelo temos vários relacionamentos 1:1, nomeadamente na tabela Viatura com Utilizador e Viatura com a tabela TORQUE_Resultados.

Passo 4

Para os relacionamentos 1:N, o atributo identificador de uma das entidades, à tabela do lado N é acrescentada a chave primária da outra tabela relacionada, bem como os atributos de relacionamento. No nosso modelo não contém este tipo de relacionamento.

Passo 5

Para os relacionamentos N:M, é criada uma tabela composta pelas chaves primárias, e pelos atributos de relacionamento. No nosso modelo não contém este tipo de relacionamento.

Passo 6

Para os atributos multivalor, é criada uma tabela com o atributo multivalor e acrescentada a chave primária da entidade que lhe deu origem, sendo a chave primária desta tabela composta por todos os campos. No nosso modelo não contém este tipo de relacionamento.

4.6. Modelo Relacional

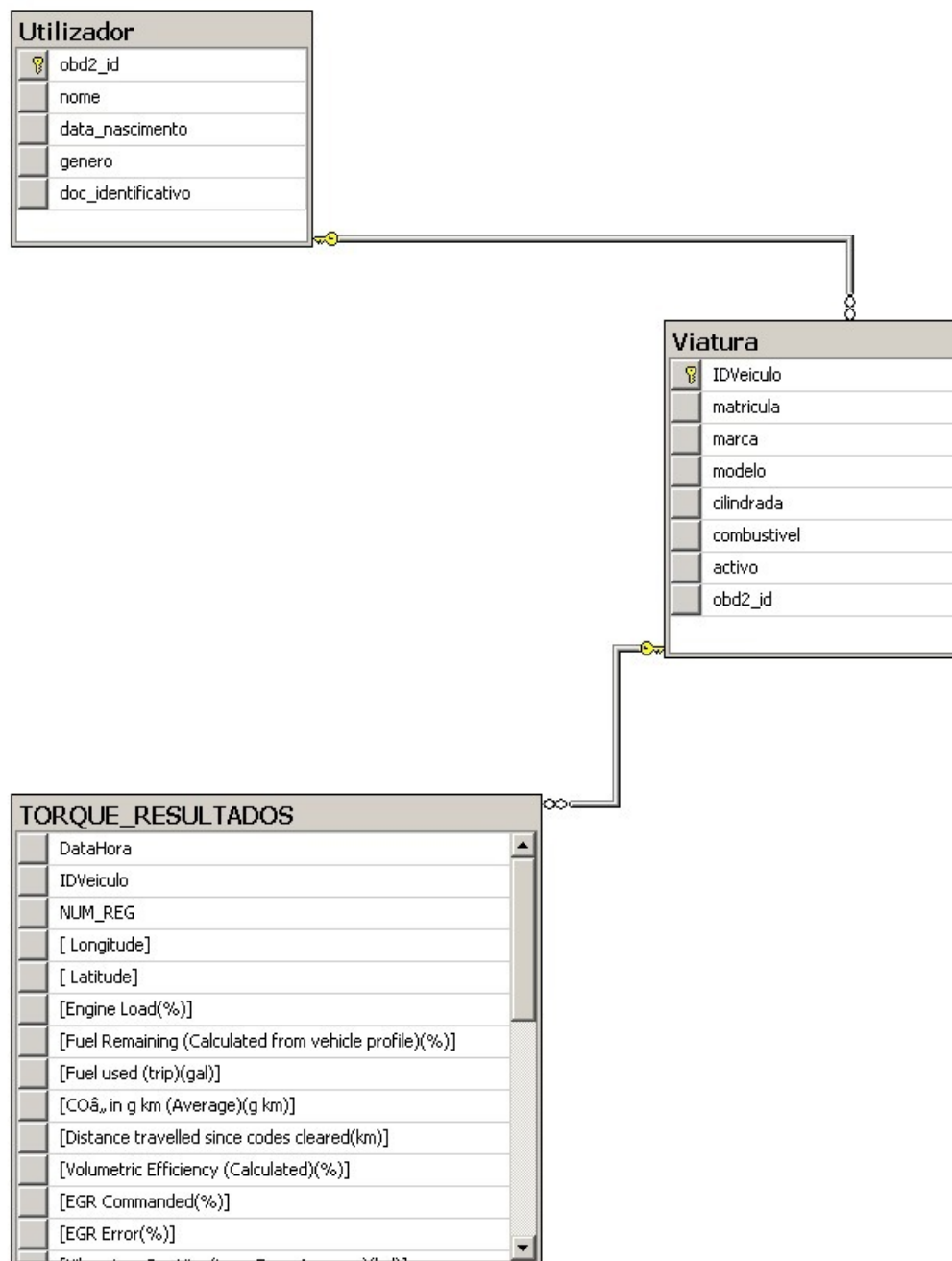


Figura 4.4 – Modelo Entidade Relacional

4.6.1. Tabela principal dos indicadores da viatura

Esta tabela resulta do trabalho de seleção exaustivo, para determinar de entre os cerca de 3500 indicadores captados pelo dispositivo, quais seriam os 50 indicadores que consideramos ter maior relevância e que em fase posterior poderão ser úteis para os gestores tomarem decisões sobre a sua frota automóvel.

	Column Name	Data Type	Allow Nulls
▶	DataHora	datetime	<input checked="" type="checkbox"/>
	IDVeiculo	varchar(1)	<input checked="" type="checkbox"/>
	NUM_REG	varchar(50)	<input checked="" type="checkbox"/>
	[Longitude]	varchar(50)	<input checked="" type="checkbox"/>
	[Latitude]	varchar(50)	<input checked="" type="checkbox"/>
	[Engine Load(%)]	varchar(50)	<input checked="" type="checkbox"/>
	[Fuel Remaining (Calc...	varchar(50)	<input checked="" type="checkbox"/>
	[Fuel used (trip)(gal)]	varchar(50)	<input checked="" type="checkbox"/>
	[COâ„in g km (Averag...	varchar(50)	<input checked="" type="checkbox"/>
	[Distance travelled sin...	varchar(50)	<input checked="" type="checkbox"/>
	[Volumetric Efficiency ...	varchar(50)	<input checked="" type="checkbox"/>
	[EGR Commanded(%)]	varchar(50)	<input checked="" type="checkbox"/>
	[EGR Error(%)]	varchar(50)	<input checked="" type="checkbox"/>
	[Kilometers Per Litre(L...	varchar(50)	<input checked="" type="checkbox"/>
	[Litres Per 100 Kilomet...	varchar(50)	<input checked="" type="checkbox"/>
	[Trip average Litres 1...	varchar(50)	<input checked="" type="checkbox"/>
	[Percentage of City dr...	varchar(50)	<input checked="" type="checkbox"/>
	[Percentage of Highw...	varchar(50)	<input checked="" type="checkbox"/>
	[Percentage of Idle dr...	varchar(50)	<input checked="" type="checkbox"/>
	[Throttle Position(Man...	varchar(50)	<input checked="" type="checkbox"/>
	[Barometric pressure (...]	varchar(50)	<input checked="" type="checkbox"/>
	[Intake Manifold Press...	varchar(50)	<input checked="" type="checkbox"/>
	[Fuel Rail Pressure(psi)]	varchar(50)	<input checked="" type="checkbox"/>
	[Engine RPM(rpm)]	varchar(50)	<input checked="" type="checkbox"/>
	[Acceleration Sensor(...]	varchar(50)	<input checked="" type="checkbox"/>
	[Acceleration Sensor(...]	varchar(50)	<input checked="" type="checkbox"/>
	[Acceleration Sensor(...]	varchar(50)	<input checked="" type="checkbox"/>
	[Air Fuel Ratio(Measur...	varchar(50)	<input checked="" type="checkbox"/>
	[Engine Coolant Temp...	varchar(50)	<input checked="" type="checkbox"/>
	[Turbo Boost & Vacuu...	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 4.5 – Principais Indicadores da Viatura

5. Desenvolvimento e Implementação

Esta fase de desenvolvimento teve a maior alocação tempo no projeto, cerca de um mês, dado que foi necessário programar os diversos componentes, desde a recolha dos dados ao seu tratamento e entrada no sistema interno, como designado no capítulo anterior e as respetivas configurações necessárias nos servidores de FTP, Base de Dados e Aplicaacional. Nesta fase foi igualmente necessário iniciar os testes funcionais de integração que serão posteriormente detalhados no capítulo 6.

5.1. Tratamento de Dados

5.1.1. Importação de registos – Sistema Interno

A importação de registos é um processo automático, via script (código SQL) que vai buscar os ficheiros *.csv ao Servidor FTP, renomeando os ficheiros, mantendo os dados no seu estado original (em bruto).

Para a execução deste processo são criadas tabelas temporárias para permitir o tratamento dos dados importados. Assim que os dados são importados os ficheiros originais (*.csv) são automaticamente eliminados.

Após a conclusão deste processo estamos em condições de alimentar a tabela principal TORQUE_RESULTADOS que suporta a aplicação principal. Esta tabela contém os cerca de 50 indicadores referentes aos dados lidos pelos respetivos sensores do veículo. Relembramos que estes 50 indicadores foram selecionados pela sua relevância dentro dos cerca de 3500 campos do ficheiro original *.csv.

De referir que os scripts automáticos estão agendados para serem executados com uma periodicidade de 10 minutos, sendo este valor parametrizável de acordo com as necessidades.

5.1.2. Rotinas Automáticas – Scripts SQL

A seguinte tarefa, agendada para ser executada a cada 10 minutos, é composta pela execução dos seguintes scripts:

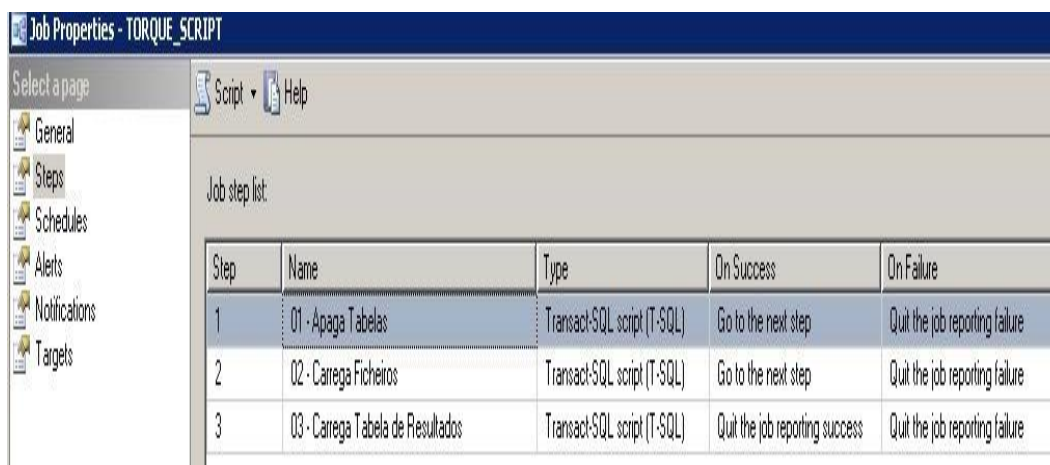


Figura 5.1 – Execução Scripts Automáticos

5.1.2.1. Detalhe do Script 01:

```

/*****
** Apaga Tabelas temporárias ( TORQUE, VEICULO_A, VEICULO_B )
*****/

USE TORQUE

DELETE FROM dbo.TORQUE
DELETE FROM dbo.VEICULO_A
DELETE FROM dbo.VEICULO_B

```

Figura 5.2 – Eliminação Tabelas Temporárias

5.1.2.2. Detalhe do Script 02:

```

/*****
/* VEICULO |
*****/

Declare @myCmd_A_01 as varchar(60)
Declare @myCmd_A_02 as varchar(60)
DECLARE @IsFileExists_A INT

/* Definir comandos a utilizar (Rename File e Delete File) *****/
set @myCmd_A_01 = 'REN C:\Torque\Veiculo_A\trackLog*.csv Veiculo_A.csv'
set @myCmd_A_02 = 'DEL C:\Torque\Veiculo_A\Veiculo_A.csv'

/* 1º PASSO *****/
/* Executar comando de rename do ficheiro .CSV *****/
EXEC master..xp_cmdshell @myCmd_A_01

/* Executar comando de verificação de existência do ficheiro .CSV *****/
EXEC Master..xp_fileexist 'C:\Torque\Veiculo_A\Veiculo_A.csv', @IsFileExists_A OUT

/* Testar existência de ficheiro .CSV *****/
IF @IsFileExists_A = 1
BEGIN
    /* 2º PASSO *****/
    /* Importar dados do ficheiro .CSV referentes ao Veiculo A, para a Tabela temporária VEICULO_A, usando o comando BULK INSERT *****/
    BULK INSERT dbo.VEICULO_A
    FROM 'C:\Torque\Veiculo_A\Veiculo_A.csv'
    WITH (MAXERRORS = 0, FIRSTROW = 2, RowTerminator = '0x0a', FieldTerminator = ',')
]

    /* 3º PASSO *****/
    /* Executar comando de delete do ficheiro .CSV *****/
    EXEC master..xp_cmdshell @myCmd_A_02

    /* 4º PASSO *****/
    /* Executar Transacção para Alimentar a Tabela principal 'TORQUE' *****/
    BEGIN TRANSACTION myTransA;
        INSERT INTO dbo.TORQUE SELECT VEICULO_A.*, 'A' FROM dbo.VEICULO_A
    COMMIT;
END

```

Figura 5.3 – Processo Importação dos Dados

5.1.2.3. Detalhe do Script 03:

```

/*****
/* Query que alimenta a tabela final de resultados por veiculo - TORQUE_RESULTADOS
/* Resultados dos registos dos ultimos 10 minutos - Alterável - com histórico ( DataHora )
*****/

USE TORQUE

INSERT INTO dbo.TORQUE_RESULTADOS
SELECT
    GETDATE() as DataHora,
    TORQUE.IDVeiculo,
    SUM(1) AS NUM_REG,
    MIN([ Longitude]),
    MIN([ Latitude]),
    MIN([Engine Load(%)]),
    MIN([Fuel Remaining (Calculated from vehicle profile)(%)]),
    MIN([Fuel used (trip)(gal)]),
    MIN([COâ, in g km (Average)(g km)]),
    MIN([Distance travelled since codes cleared(km)]),
    MIN([Volumetric Efficiency (Calculated)(%)]),
    MIN([EGR Commanded(%)]),
    MIN([EGR Error(%)]),
    MIN([Kilometers Per Litre(Long Term Average)(kpl)]),
    MIN([Litres Per 100 Kilometer(Long Term Average)(l 100km)]),
    MIN([Trip average Litres 100 KM(l 100km)]),
    MIN([Percentage of City driving(%)]),
    MIN([Percentage of Highway driving(%)]),
    MIN([Percentage of Idle driving(%)]),
    MIN([Throttle Position(Manifold)(%)]),
    MIN([Barometric pressure (from vehicle)(psi)]),
    MIN([Intake Manifold Pressure(psi)]),
    MIN([Fuel Rail Pressure(psi)]),
    AVG(CAST([Engine RPM(rpm)] AS DECIMAL(8, 2) )),
    MIN([Acceleration Sensor(X axis)(g)]),
    MIN([Acceleration Sensor(Y axis)(g)]),
    MIN([Acceleration Sensor(Z axis)(g)]),
    MIN([Air Fuel Ratio(Measured)( 1)]),
    MAX([Engine Coolant Temperature(Â°C)]),
    MIN([Turbo Boost & Vacuum Gauge(psi)])

FROM
    dbo.TORQUE AS TORQUE,
    ( SELECT IDVeiculo, DateAdd(mi, -10, MAX([ Device Time])) AS TIME_FROM, DateAdd(mi, 0, MAX([ Device Time])) AS TIME_TO FROM dbo.TORQUE GROUP BY IDVeiculo ) AS DATAS

WHERE
    TORQUE.IDVeiculo = DATAS.IDVeiculo AND ( TORQUE.[ Device Time] BETWEEN DATAS.TIME_FROM AND DATAS.TIME_TO )

GROUP BY
    TORQUE.IDVeiculo

```

Figura 5.4 – Carregamento dos Dados Finais

5.1.3. Diagrama da Manutenção da Base de Dados

Um fator determinante para o desempenho de uma base de dados e para sua integridade é a implementação de um plano de manutenção diária. Isto significa que devido ao elevado número de registos era de extrema importância para a viabilidade desta solução estar integrada com este plano de manutenção.

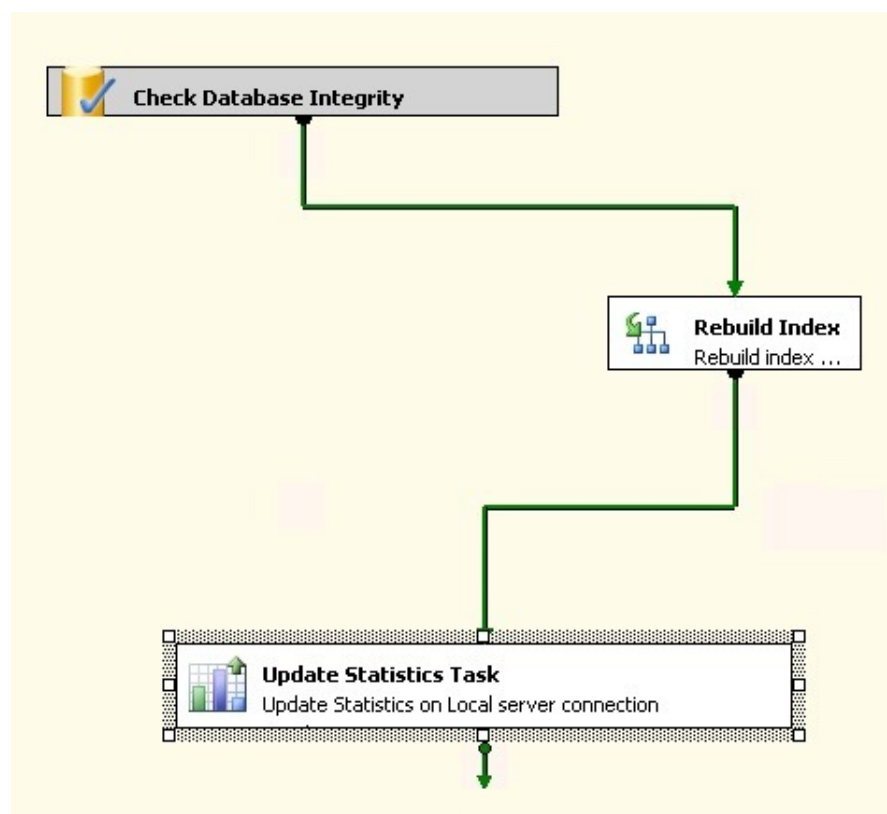


Figura 5.5 – Diagrama de Manutenção da Base de Dados

5.2. Consola Aplicacional – Dashboard DSS

Nesta fase de desenvolvimento chegamos ao componente mais importante deste projeto no que diz respeito a vertente do sistema de gestão.

Procurámos identificar os elementos essenciais que permitissem ao gestor de frota, i.e., ao utilizador(s) do programa que gere as viaturas, utilizadores, perfis e alertas.

É este programa que designamos como Dashboard DSS e que a seguir descrevemos na sua estrutura conceptual.

5.2.1. Estrutura Conceptual do Dashboard DSS

5.2.1.1. Definição de Decision Support System (Sistema de Apoio à Decisão)

Sistemas de apoio à decisão é uma classe de Sistemas de Informação, ou sistemas baseados em conhecimento. Refere-se simplesmente a um modelo genérico de tomada de decisão que analisa um grande número de variáveis para que seja possível o posicionamento a uma determinada questão.

No nosso projeto esta vertente assume a sua expressão com a implementação de uma consola, dashboard DSS, que iremos descrever no ponto 5.2.2

5.2.2. Estrutura conceptual

Ao definirmos os objetivos deste projeto tivemos em consideração um dos elementos mais importantes e diferenciadores da solução que passaria por desenvolver uma consola aplicacional de gestão (Dashboard DSS) para albergar as funcionalidades que descrevemos em seguida:

1. Resumo diagnóstico do estado da viatura;
2. Matriz de histórico de alertas;
3. Sistema de alarme para alertas com elevada prioridade
4. Monitorização de trajetos das viaturas (GPS)
5. Perfil de condução
6. Gestão de viaturas e utilizadores
7. Relatórios de dados analíticos
8. Relatórios de histórico de alertas

O nosso projeto procurou desenvolver algumas destas funcionalidades com prova de conceito (Proof of Concept) para a qual implementámos a consola aplicacional DSS

5.2.3. Estrutura funcional do Dashboard DSS

A nossa consola aplicacional – Dashboard DSS tem a seguinte estrutura funcional:

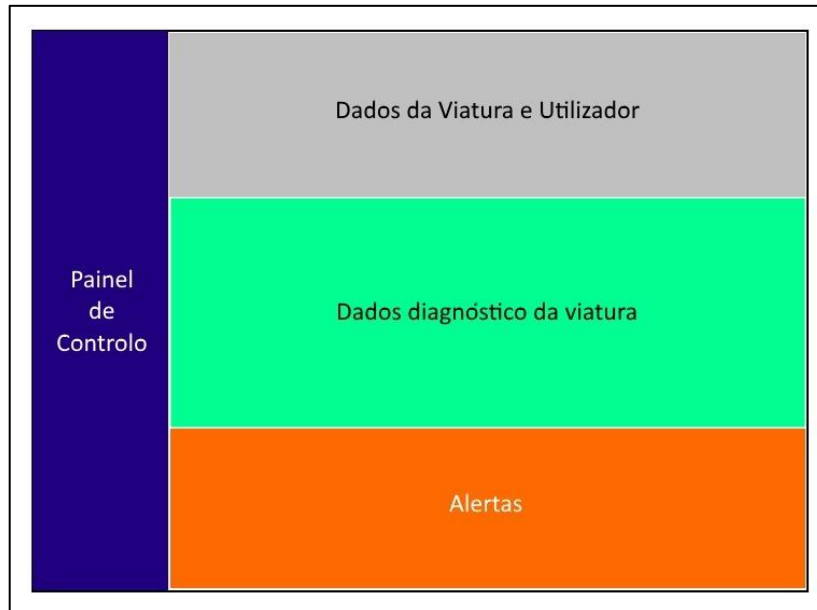


Figura 5.6 – Estrutura Funcional dashboard DSS

O dashboard DSS encontra-se dividido em 4 zonas, tal como descrito na figura 5.6.

- **Painel de Controlo (interativo):** Área da aplicação onde se encontram as diversas teclas para forçar a atualização dos dados de perfil, alerta e viatura/utilizador e também as teclas para abrir os ecrãs de criação de viaturas e utilizadores.
- **Dados da viatura e utilizador (Informativo):** Área da aplicação onde se encontram os dados da viatura e do utilizador que se está em contexto.
- **Dados diagnóstico da viatura (Informativo):** Área da aplicação onde se encontram os dados da viatura em contexto, nomeadamente, período a que respeitam os dados, nº atual de kms, indicadores do tipo de condução e indicadores relativos a velocidade e consumo da viatura. Os dados indicados poderão ser alterados, retirados ou complementados em colaboração e de acordo com as necessidades dos gestores de frota automóvel.
- **Alertas:** Área da aplicação onde se encontram os alertas de captados e tratados pela aplicação de acordo com os parâmetros de cada modelo de viatura.

5.2.4. Painel inicial de acesso ao Dashboard DSS

O acesso à aplicação é permitido apenas a utilizadores autorizados, sendo as suas credenciais validadas através do nome de utilizador e password.



Figura 5.7 – Login na Aplicação DSS

5.2.5. Painel do Menu Principal

Quando o gestor autorizado é validado entra no seguinte painel principal:

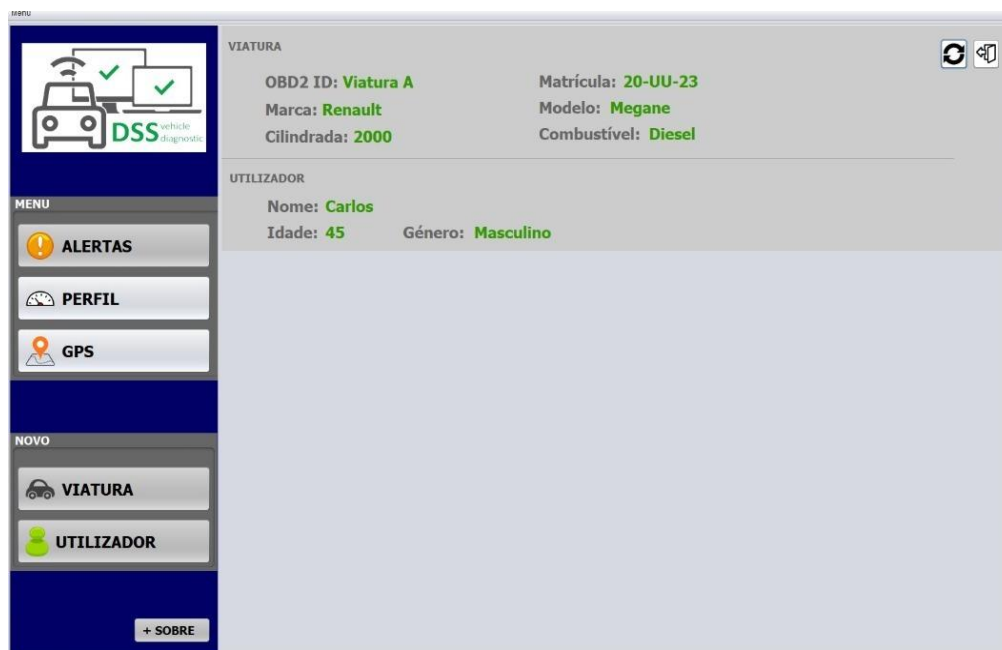


Figura 5.8 – Menu Principal

5.2.6. Painel de opções do MENU Principal

No lado esquerdo o gestor pode selecionar as opções de ALERTAS, PERFIL, GPS, para obter uma atualização dos dados sobre o veículo que está em contexto.

- Vista geral do painel principal



Figura 5.9 – Opções Menu Principal

- PERFIL – permite atualizar os principais indicadores de utilização do veículo

Menu

VIATURA

OB2 ID: ID Matrícula: Matrícula
Marca: Marca Modelo: Modelo
Cilindrada: Cilindrada Combustível: Combustível

UTILIZADOR

Nome: Nome
Idade: Idade Género: Genero

DADOS DE UTILIZAÇÃO

PERÍODO:
DE 0 A 0 **Nº ATUAL DE KMS: 0**

TIPO DE CONDUÇÃO*

CIDADE: 0%
AUTOESTRADA: 0%
PARADO: 0%

VELOCIDADE E CONSUMO*

VELOCIDADE MÉDIA: 0 KMS/HORA
VELOCIDADE MÁXIMA: 0 KMS/HORA
CONSUMO MÉDIO: 0 LITROS/100

*Os valores apresentados têm por base a utilização da viatura com sistema ativo


ALERTAS

TIPO DE ALERTA	DATA E HORA	OBSERVAÇÕES
----------------	-------------	-------------

Figura 5.10 – Opções Perfil

- Zona de ALERTAS – permite atualizar à listagem de alertas do veículo

ALERTAS

 **Temperatura do óleo elevada**

TIPO DE ALERTA	DATA E HORA	OBSERVAÇÕES
Temperatura do óleo elevada	06/07/2019 - 18:59	
Temperatura do água elevada	07/07/2019 - 18:33	
Temperatura do óleo elevada	05/07/2019 - 18:19	
Temperatura do óleo elevada	07/07/2019 - 15:59	Alerta recorrente (3)
Voltagem reduzida	04/07/2019 - 10:49	

Figura 5.11 – Alertas Veículo

5.2.7. Painel de opções do NOVO

Neste ecrã ao podem ser criados novas viaturas e novos utilizadores

- VIATURA – acede ao painel de criação de uma nova viatura



The screenshot shows a software window titled 'CRIAR VIATURA' (Create Vehicle). In the top left corner, there is a logo featuring a car with a signal antenna and a laptop with a green checkmark, with the text 'DSS vehicle diagnostic' below it. The main area of the window contains several input fields for vehicle data: 'OBD2_ID:' with a two-part text box, 'MATRÍCULA:' with a single text box, 'MARCA:' with a wide text box, 'MODELO:' with a wide text box, 'CILINDRADA:' with a text box followed by 'CM3', and 'COMBUSTÍVEL:' with a dropdown menu. At the bottom left is a 'Sair' (Exit) button, and at the bottom right is a 'Gravar' (Save) button.

Figura 5.12 – Criação Nova Viatura

- Preenchimento do quadro de criação de viatura:
 - Ecrã utilizado para a inserção dos dados respeitantes a novas viaturas;
 - Após preenchimento dos dados da viatura, o utilizador deverá gravar os dados, voltando ao ecrã principal de aplicação;
 - A tecla sair permite sair do ecrã sem salvar os dados.

- UTILIZADOR – acede ao painel de criação de um novo utilizador

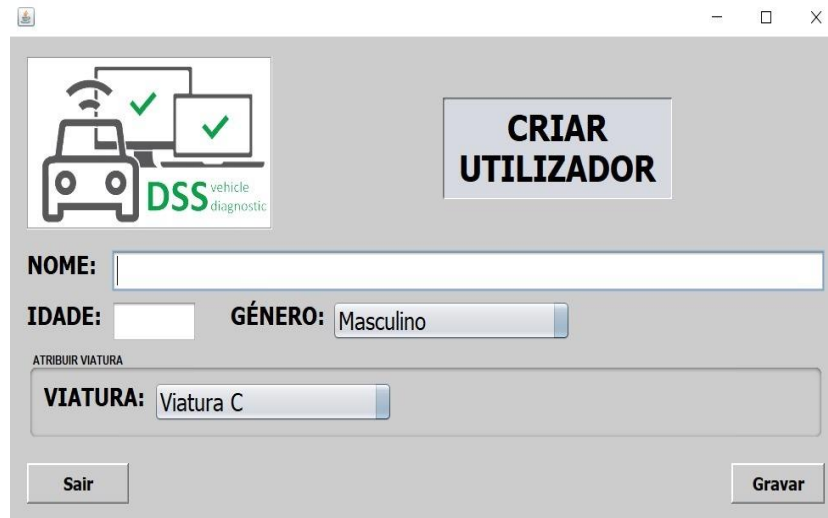


Figura 5.13 – Criação Novo Utilizador

- Preenchimento do quadro de criação de utilizador:
 - Ecrã utilizado para a inserção dos dados respeitantes a novos utilizadores;
 - Após preenchimento dos dados do utilizador da viatura, o utilizador deverá gravar os dados, voltando ao ecrã principal de aplicação;
 - A tecla sair permite sair do ecrã sem salvar os dados.

6. Testes e Resultados

Durante a fase de desenvolvimento iniciaram-se os primeiros testes sendo que para uma avaliação mais adequada e estruturada optámos por seguir a metodologia base deste projeto, segmentando esta fase em 3 seções que a seguir detalhamos.

6.1. Testes funcionais, integração e de aceitação

6.1.1. Dispositivo ELM327 para recolha de dados instalado no veículo via tomada OBDII

Depois de escolhido qual o dispositivo de recolha de dados da viatura, optando pela sua versão Bluetooth, foi necessário testá-lo no que respeita à instalação, configuração (ligação a um dispositivo móvel) e eventuais questões que teríamos de contemplar para o conectar com sucesso.



Figura 6.1 – Dispositivo ELM327

Após consulta das características e instruções da ficha ELM327, procedemos à instalação da mesma numa viatura, que passaremos a designar por Veículo A (V.A). A instalação da ficha ELM327 é extremamente fácil e rápida, bastando para tal ligar a ficha na tomada universal OBD2 da viatura. A tomada OBD2 encontra-se geralmente localizada debaixo da consola/volante das viaturas automóveis.



Figura 6.2 – Dispositivo ELM327 Instalado na Viatura

6.1.2. Recolha e envio de dados do veículo – Torque® e Fillezilla (FTP)

Aqui foi necessário configurar a app Torque disponível para versão Android, que instalámos num dispositivo móvel. Esta aplicação quando estabelecida a ligação Bluetooth com a ficha ELM327 e estando a viatura em funcionamento, permite a criação de um ficheiro que fica armazenado na memória do equipamento móvel.

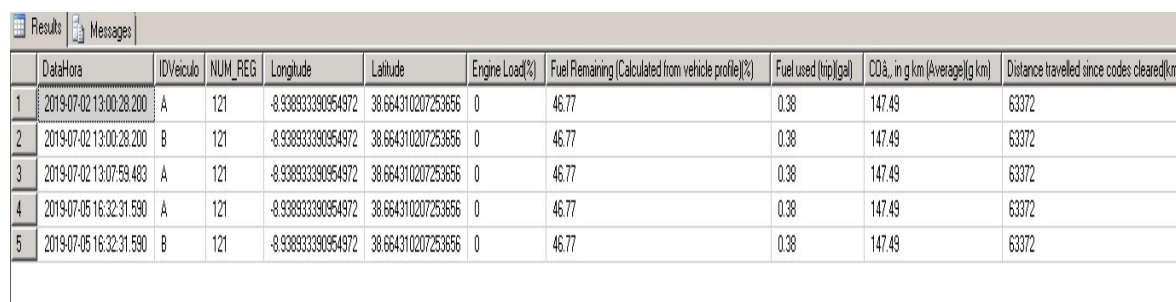
O ficheiro anterior é enviado a cada 10 minutos, em resultado da parametrização do serviço FTP. Assim foi possível identificar as características do ficheiro para o tratamento dos dados na fase seguinte.

6.1.3. Montagem da Base de Dados – SQL

Atualmente a solução encontra-se a operar numa cloud privada, que designámos anteriormente como sistema interno e onde foram desenvolvidas várias rotinas para tratamento do ficheiro recebido. O ficheiro recebido contém nesta fase do fluxo de dados, cerca de 3500 indicadores resultantes das leituras efetuadas pelos sensores da viatura. Foi necessário configurar as rotinas em linguagem SQL, para importar o ficheiro e transformá-lo num formato e dimensão adequada para usufruto da aplicação Dashboard DSS.

Importa referir que para testarmos este modelo de recolha de dados e posterior tratamento foram utilizadas 2 viaturas em simultâneo, para verificar o cumprimento das necessidades de funcionamento de uma gestão de frotas de automóveis.

Exemplo de uma tabela com dados tratados provenientes da leitura de 2 veículos, que estão tipicamente identificados no nosso programa como Veículo A e Veículo B.



	DataHora	IDVeiculo	NUM_REG	Longitude	Latitude	Engine Load(%)	Fuel Remaining (Calculated from vehicle profile)(%)	Fuel used (trip)(gal)	CO2, in g/km (Average)(g/km)	Distance travelled since codes cleared(km)
1	2019-07-02 13:00:28.200	A	121	-8.9389333390954972	38.664310207253656	0	46.77	0.38	147.49	63372
2	2019-07-02 13:00:28.200	B	121	-8.9389333390954972	38.664310207253656	0	46.77	0.38	147.49	63372
3	2019-07-02 13:07:59.483	A	121	-8.9389333390954972	38.664310207253656	0	46.77	0.38	147.49	63372
4	2019-07-05 16:32:31.590	A	121	-8.9389333390954972	38.664310207253656	0	46.77	0.38	147.49	63372
5	2019-07-05 16:32:31.590	B	121	-8.9389333390954972	38.664310207253656	0	46.77	0.38	147.49	63372

Figura 6.3 – Registos Dados das Viaturas

Os dados em cima exemplificados, resultam dos exaustivos testes realizados e podem desta forma ser disponibilizados para popular o dashboard DSS, permitindo assim o acesso à informação pretendida pelo gestor de frota automóvel.

6.2. Testes Aplicacionais no dashboard DSS

Na última fase de testes de integração foi testada a aplicação desenvolvida e que designámos DSS.

Conseguimos disponibilizar os dados para o gestor de frota automóvel, poder visualizar a informação que considere relevante para a sua atividade e para as decisões de que tem de tomar.

Os menus e opções foram testados em ambiente real, considerando que tivemos durante esta fase duas viaturas a transmitir dados de forma exaustiva para podermos fazer uma avaliação de desempenho a diversos níveis, velocidade de transferência de dados, conversação das tabelas SQL, permitindo a qualificação final, que normalmente faz parte da fase de testes de aceitação por um cliente.

7. Infraestrutura de Produção

Depois de concluída a fase de testes foi importante, partindo da avaliação de desempenho detalhada no capítulo anterior, considerámos a que infraestrutura adequada em termos gerais passa pela solução que a seguir exemplificamos.

7.1. Características do Servidor com SQL Server

Neste servidor optámos por ter apenas o serviço de base de dados, usando para o efeito o software SQL Server 2016

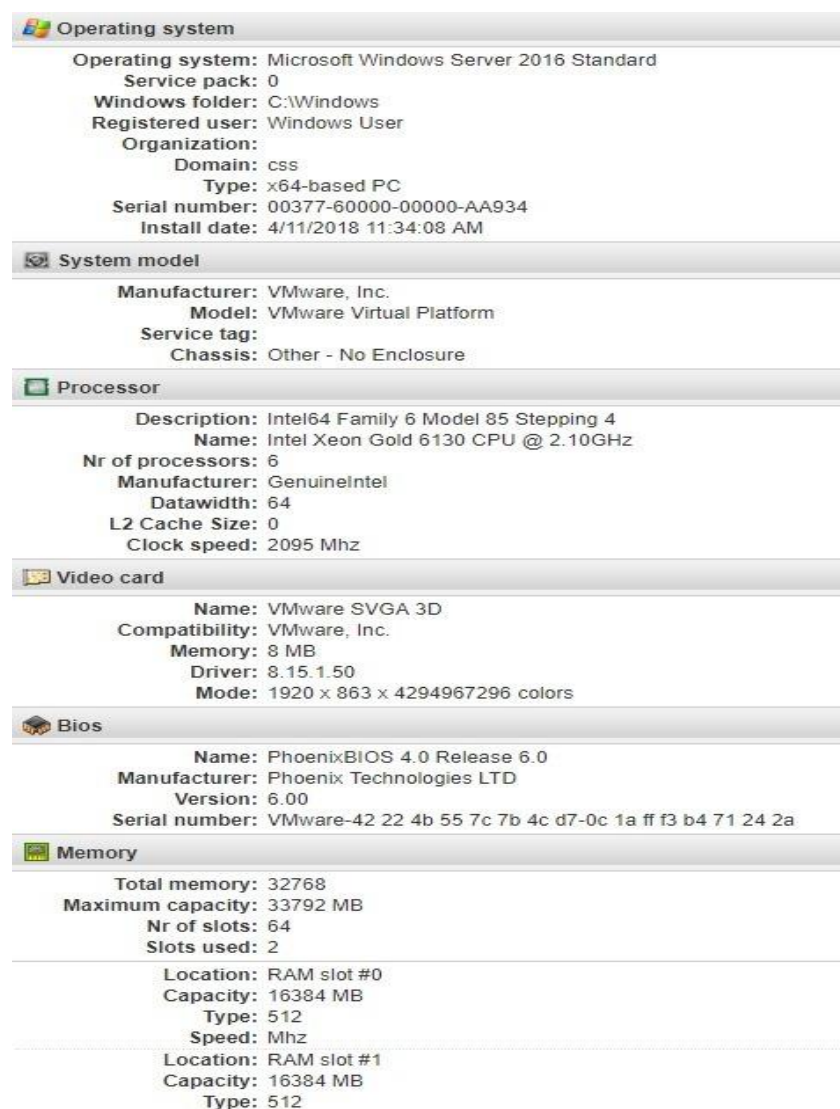


Figura 7.1 – Características Servidor SQL Server

7.2. Características do Servidor com FTP e DSS

Neste servidor considerámos a instalação do serviço FTP e da Framework aplicacional.

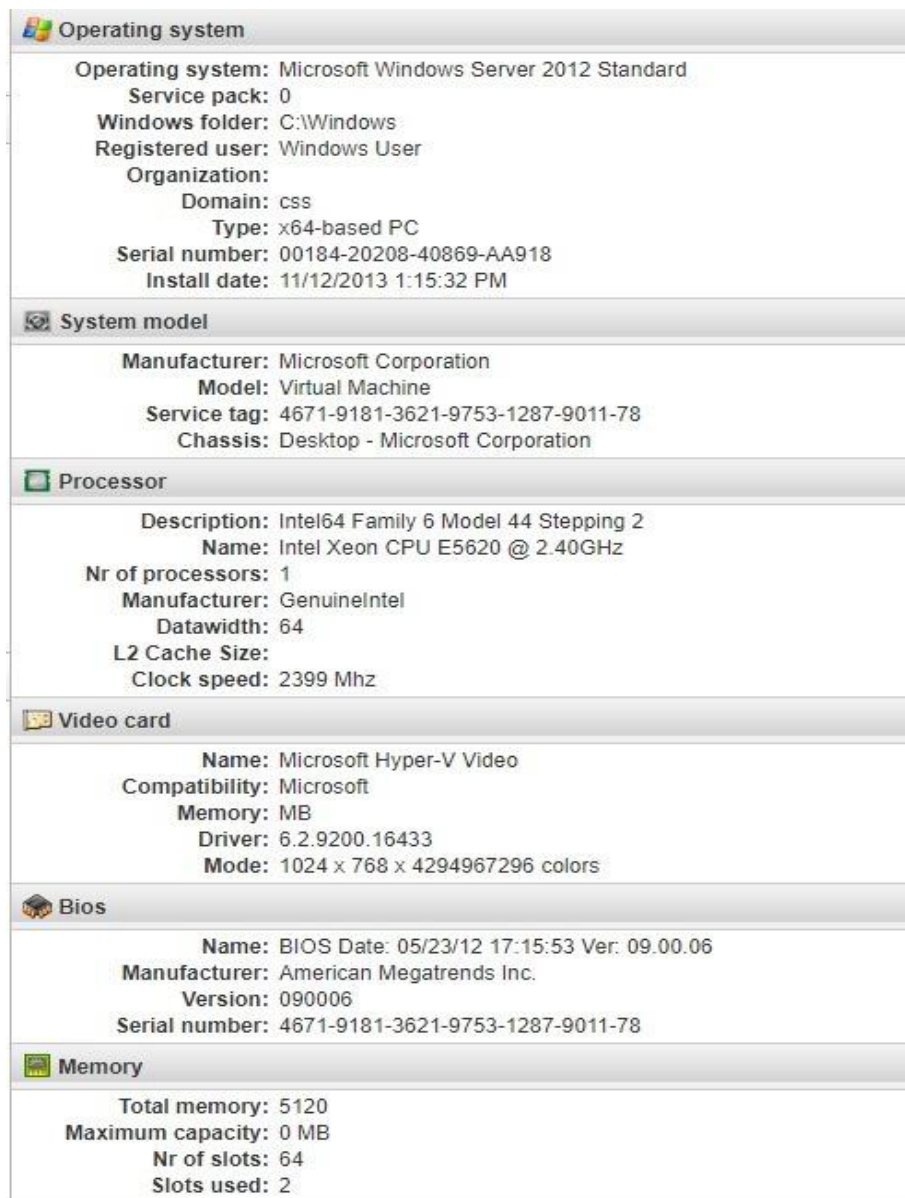


Figura 7.2 - Características Servidor FTP e DSS

Escolhemos software o FileZilla para o serviço FTP, por ser intuitivo, de fácil manuseamento e opensource.

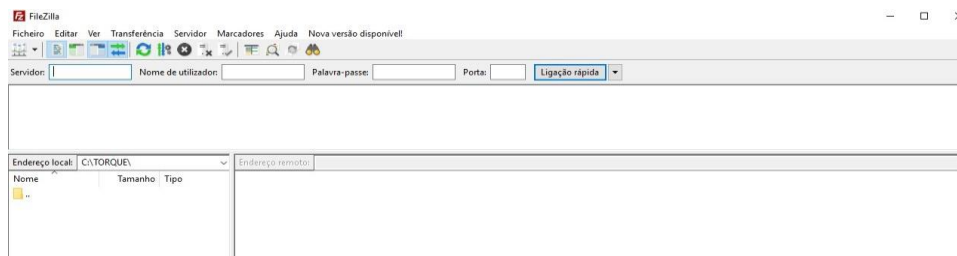


Figura 7.3 – Painel Filezilla Server

Installed software				
	Software	Version	Publisher	Install Date
G	Data Loader	41.0.0	salesforce.com	
G	FileZilla Client 3.33.0	3.33.0	Tim Kosse	

Figura 7.4 – Versão FileZilla Server

8. Tecnologias e Ferramentas

As tecnologias escolhidas tiveram por base as necessidades espelhadas nos requisitos. Um dos membros do grupo providenciou acesso à infraestrutura de servidores de uma empresa privada, pelo que tivemos a possibilidade de, com estes serviços, simular um ambiente de produção.

As principais tecnologias que serviram para implementação foram:

- Plataformas de Servidor: Microsoft Windows Server
- Plataforma telemóvel: Sistema Operativo Android
- App Móvel: Torque®
- IDE de Programação: Netbeans
- Linguagem de Programação: JAVA
- Base Dados: *SQL*
- Serviço FTP: Fillezilla

9. Custos do Projeto

Logo após definirmos o âmbito do projeto procuramos na fase de levantamento de requisitos identificar os elementos principais que seriam necessários para realizarmos este projeto.

O custo de realização deste projeto, no que diz respeito a componentes que foram adquiridos, foi relativamente acessível, mas temos que também ter em consideração que o facto de podermos dispor de acesso a uma infraestrutura real de servidores (empresa privada) facilitou e reduziu substancialmente o custo global do projeto.

No capítulo 12 tecemos algumas considerações sobre futuros desenvolvimentos deste projeto, com o objetivo de tornar a solução DSS disponível via Cloud Service. Igualmente procuramos enquadrar a solução numa perspectiva de modelo de negócio.

De seguida identificamos os principais elementos e os seus respetivos custos.

9.1. Custos gerais

Dispositivo ELM327	€ 12,50
Dispositivo Móvel (Android).....	€ 110,00
Dispositivo Móvel (Android) custo de dados.....	€ 7,50
Viagens de Automóvel para recolha dados, aproximadamente 250Km /Gásóleo	€ 25,00
Aluguer de serviços Server Windows c/ SQL, FTP, c/ débito 100Mb.....	€ 380,00 /mês
Serviços Open Sourc Server MySQL, FTP, c/ débito 100Mb.....	€ 150,00

10. Metodologia do Trabalho

Ao longo do desenvolvimento do projeto foram utilizadas várias metodologias, que foram sendo identificadas nos respetivos capítulos, como por exemplo na gestão do projeto, na modelação da base de dados, no desenho técnico, na implementação e testes que e nos auxiliaram na execução deste trabalho de fim de curso.

A maioria dessas ferramentas foram-nos introduzidas nas diversas disciplinas que foram lecionadas ao longo de ambas as licenciaturas de Engenharia Informática e a de Informática de Gestão.

10.1. Características da Equipa de Trabalho

O grupo de trabalho surgiu na sequência de anteriores atividades de grupo dado que os três colegas partilharam nos últimos anos algumas das disciplinas das licenciaturas mencionadas o que facilitou a dinâmica de trabalho desde o planeamento, à definição de objetivos, à atribuição de tarefas, à alocação temporal e principalmente às características técnicas e conceptuais de cada elemento.

De acordo com a natureza do projeto temos 3 áreas de desenvolvimento que foram trabalhadas tanto remotamente como conjuntamente, nomeadamente a vertente de programação, a de integração do sistema interno (servidores FTP, Base de Dados, Aplicacional) e a elaboração do relatório final.

De salientar que a condição de alunos de pós-laboral com responsabilidades profissionais, condicionou pontualmente o desenvolvimento do projeto, mas também acabou por nos auxiliar na elaboração deste projeto, pois a alargada experiência profissional de todos contribui de forma decisiva na integração da engenharia informática com os componentes de gestão e também para uma abordagem mais focada dos problemas que surgiram durante o período em que efetuamos o trabalho.

11. Conclusões e Trabalho Futuro

11.1. Conclusões

Este projeto teve como âmbito a criação de um sistema integrado de diagnóstico e gestão de viaturas automóveis, através da recolha e tratamento de diversos dados estáticos e dinâmicos dessas viaturas, utilizando para o efeito um dispositivo ELM327® ligado na tomada de diagnóstico OBD2, um dispositivo móvel com acesso Bluetooth, GPS e GPRS para partilha de dados.

Associando os dados estáticos e dinâmicos das viaturas, provenientes dos diversos sensores existentes (motor, caixa, acelerador, travão, etc) e o histórico de dados armazenados em sistema, a aplicação permite gerar indicadores para a apoio ao tratamento e gestão de uma frota de automóveis, auxiliando os decisores e os responsáveis pela gestão das frota de automóveis das empresas, na melhoria da tomada de decisão relativa às necessidades de adequação da manutenção preventiva, corretiva das viaturas da sua frota, mas também na determinação de perfis de condução dos vários utilizadores das mesmas.

Podemos concluir que os principais objetivos foram cumpridos, sendo que para o efeito temos como prova de conceito (Proof of Concept) uma solução operacional, End-to-End que poderá ser alvo para futuro desenvolvimento com vista a enquadrar um modelo de negócio flexível e adaptável aos diferentes clientes alvo.

Os objetivos principais deste projeto foram:

- 1) Criação de histórico para cada viatura da frota incluindo perfil de utilização;
- 2) Apoiar na definição de critérios de manutenção preventiva e corretiva;
- 3) Mapa de Gestão -Dashboard para DSS - Decision Support System

11.2. Trabalho Futuro

Ao definirmos os objetivos deste projeto tivemos em consideração um dos elementos mais importantes e diferenciadores da solução que passaria por desenvolver uma consola aplicacional de gestão (Dashboard DSS) para albergar as funcionalidades que mencionamos neste relatório:

- a) Resumo diagnóstico do estado da viatura;
- b) Matriz de histórico de alertas;
- c) Sistema de alarme para alertas com elevada prioridade
- d) Monitorização de trajetos das viaturas (GPS)
- e) Perfil de condução
- f) Gestão de viaturas e utilizadores
- g) Relatórios de dados analíticos
- h) Relatórios de histórico de alertas

A visão que suporta o futuro desenvolvimento, com base na prova de conceito que demonstrámos neste projeto, assenta na implementação das funcionalidades que irão permitir a integração de algoritmos, baseados em inteligência artificial, que servirão de motor para a produção de relatórios de dados analíticos, como por exemplo prever quando uma viatura poderá vir a necessitar de manutenção preventiva, antes de determinados alertas serem ativados (que atual sistema já prevê).

Atendendo à dimensão do projeto e ao tempo limitado para a conclusão do mesmo, houve necessidade de priorizar as tarefas inerentes ao trabalho, não nos sendo possível implementar a funcionalidade que permitia obter mapas de trajeto das viaturas, apesar de que a nossa base de dados estar a recolher as informações das coordenadas GPS de cada viatura registada.

Por último consideramos que este projeto poderá servir de base para um desenvolvimento futuro, numa tese de Mestrado, com o objetivo de expandir as seguintes vertentes:

- Modelo de negócio – Solução chave na mão com dispositivo all-in-one com mini box para instalação em viatura
- Acesso aos registos de dados das viaturas em Real Time por WebServices
- Solução Multi-Plataforma - dashboard DSS em tablets e outras plataformas móveis para gestão remota
- Infra-structure as a Service (IaaS) - Solução Cloud para o sistema de serviços e base de dados

Glossário de Termos

Aplicação WEB – aplicação adequada para funcionar via Internet.

Base de dados – coleção de elementos de dados organizados segundo um conjunto de tabelas formalmente descritas.

Beans – Um objeto que respeita um padrão de desenho e nomeação bem definido. Um bean encapsula as suas variáveis e suporta métodos.

Browser – Programa utilizado para visualizar documentos em HTML.

CVS – *Concurrent Versions System, software* que controla versões de uma aplicação durante o seu desenvolvimento.

Framework – moldura de objetos, ferramenta.

FTP – protocolo de transferência de ficheiros, File Transfer Protocol

HTML – Abreviatura de *HyperText Markup Language*. Linguagem de programação utilizada para a construção de páginas a serem disponibilizadas na Internet.

HTTP – protocolo de Internet que permite comunicação entre cliente e servidor.

Internet – o maior conjunto de redes de dados do mundo, tendo em comum apenas a utilização do protocolo TCP/IP, também designada por rede de redes.

JAVA – linguagem de programação orientada a objetos.

JSP – (*Java Server Pages*), extensão natural de servlets e que permite intercalar HTML e Java.

Open Source – Filosofia de distribuição e licenciamento de *software* na qual o código fonte está disponível gratuitamente, permitindo assim que esse mesmo código possa ser melhorado por voluntários

Oracle – empresa que comercializa produtos de base de dados, ferramentas e aplicações juntamente com serviços de consultoria.

Servlet – Componente WEB que gera conteúdo dinâmico em JAVA e que interage com cliente Web por um paradigma pedido/resposta (baseado em HTTP).

S.O. – Sistema Operativo, como Windows, Linux, MacOS X, Unix, UBUNTU.

SQL – *Structured Query Language*, é a linguagem padrão para realizar operações de leitura, escrita e modificação de informação guardada em bases de dados relacionais.

Referências Bibliográficas

- [1] – Tecnologia Java <http://java.sun.com/>
- [2] – MySQL <http://www.mysql.com/>
- [3] – Ambiente de Desenvolvimento NetBeans <http://www.netbeans.org>
- [4] – Ambiente de Desenvolvimento Eclipse <http://www.eclipse.org>
- [5] – Contentor de Servlet e JSP <http://jakarta.apache.org/tomcat/>
- [6] – Controlo de Versões <http://www.cvshome.org/>
- [7] – Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, Design Patterns - Elements of Reusable Object-Oriented Software, Addison Wesley Longman, Inc, 1994.
- [8] – Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. Middleware. In Web Services: Concepts, Architectures and Applications, Data-Centric Systems and Applications, pages 29–66. Springer Berlin Heidelberg, 2004. ISBN 978-3-642-07888-0.
- [9] – S. Krakowiak. Middleware architecture with patterns and frameworks. 2009.56
- Referências bibliográficas 57
- [10] – Kurt Geihs. Middleware challenges ahead. Computer, 34(6):24–31, 2001. ISSN 0018- 9162.
- [11] – OBJECT MANAGEMENT GROUP et al. The common object request broker: Architecture and specification (corba), 1998. pages 123–149. Springer Berlin Heidelberg, 2004. ISBN 978-3-642-07888-0.
- [12] – S. K Lippert and C. Govindarajulu. Technological, organizational, and environmental antecedents to web services adoption. Communications of the IIMA, 6(1):147–160, 2006.

[13] – Xuan Shi. Sharing service semantics using soap-based and rest web services. IT Professional, 8(2):18–24, March 2006. ISSN 1520-9202.

[14] – Martin Fowler. UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional, 2004.

[15] – DSDM Consortium et al. Dsdm agile project framework-handbook, 2014.

Apêndices

Apêndice A – Lista de Indicadores de Leitura de Sensores da Viatura

GPS Time (varchar(50), null)
Device Time (varchar(50), null)
Longitude (varchar(50), null)
Latitude (varchar(50), null)
GPS Speed (Meters second) (varchar(50), null)
Horizontal Dilution of Precision (varchar(50), null)
Altitude (varchar(50), null)
Bearing (varchar(50), null)
G(x) (varchar(50), null)
G(y) (varchar(50), null)
G(z) (varchar(50), null)
G(calibrated) (varchar(50), null)
0-100mph Time(s) (varchar(50), null)
100-200mph Time(s) (varchar(50), null)
Actual engine % torque(%) (varchar(50), null)
GPS Altitude(m) (varchar(50), null)
Android device Battery Level(%) (varchar(50), null)
GPS bearing(°) (varchar(50), null)
Barometer (on Android device)(mb) (varchar(50), null)
Torque(m) (varchar(50), null)
Engine Load(Absolute)(%) (varchar(50), null)
Engine Load(%) (varchar(50), null)
Charge air cooler temperature (CACT)(°C) (varchar(50), null)
Fuel trim bank 1 Long Term(%) (varchar(50), null)
Fuel trim bank 1 sensor 1(%) (varchar(50), null)
Fuel trim bank 1 sensor 2(%) (varchar(50), null)
Fuel trim bank 1 sensor 3(%) (varchar(50), null)
Fuel trim bank 1 sensor 4(%) (varchar(50), null)
Fuel trim bank 2 Long Term(%) (varchar(50), null)
Fuel trim bank 2 sensor 1(%) (varchar(50), null)
Fuel trim bank 2 sensor 2(%) (varchar(50), null)
Fuel trim bank 2 sensor 3(%) (varchar(50), null)
Fuel trim bank 2 sensor 4(%) (varchar(50), null)
Ethanol Fuel % (varchar(50), null)
Fuel Remaining (Calculated from vehicle profile)(%) (varchar(50), null)
Fuel Used (Trip)(gall) (varchar(50), null)
CO ₂ in g /km (Instantaneous)(g/km) (varchar(50), null)
CO ₂ in g /km (Average)(g/km) (varchar(50), null)
Fuel cost (Trip)(cost) (varchar(50), null)
Cost per mile km (Trip)(£, \$, km) (varchar(50), null)
GPS vs OBD Speed difference(km/h) (varchar(50), null)
Distance to empty (Estimated)(km) (varchar(50), null)
Trip distance (stored in vehicle profile)(km) (varchar(50), null)
Distance travelled with Hb, O2, IRI(m) (varchar(50), null)
Distance travelled since codes cleared(%) (varchar(50), null)
Trip Distance(km) (varchar(50), null)
DPP Pressure(psi) (varchar(50), null)
DPP Temperature(°C) (varchar(50), null)
Drivers demand engine % torque(%) (varchar(50), null)
Volumetric Efficiency (Calculated)(%) (varchar(50), null)
ECR Commanded(%) (varchar(50), null)
Engine reference torque(km) (varchar(50), null)
ECR Error(%) (varchar(50), null)
GPS Accuracy(m) (varchar(50), null)
Exhaust gas temp Bank 1 Sensor 1(°C) (varchar(50), null)
Exhaust gas temp Bank 1 Sensor 2(°C) (varchar(50), null)
Exhaust gas temp Bank 1 Sensor 3(°C) (varchar(50), null)
Exhaust gas temp Bank 1 Sensor 4(°C) (varchar(50), null)
Exhaust gas temp Bank 2 Sensor 1(°C) (varchar(50), null)
Exhaust gas temp Bank 2 Sensor 2(°C) (varchar(50), null)
Exhaust gas temp Bank 2 Sensor 3(°C) (varchar(50), null)
Exhaust gas temp Bank 2 Sensor 4(°C) (varchar(50), null)
Exhaust Pressure(psi) (varchar(50), null)
Fuel Rate (direct from ECU)(l/h) (varchar(50), null)
Fuel Trim Bank 1 Short Term(%) (varchar(50), null)
Fuel Trim Bank 2 Short Term(%) (varchar(50), null)
Hybrid Battery Charge (%) (varchar(50), null)
Kilometers Per Litre(Instant)(gall) (varchar(50), null)
Kilometers Per Litre(Long Term Average)(gall) (varchar(50), null)
Engine kW (At the wheels)(kW) (varchar(50), null)
GPS Latitude(°) (varchar(50), null)
GPS Longitude(°) (varchar(50), null)
Litres Per 100 Kilometers(Instant)(100km) (varchar(50), null)
Litres Per 100 Kilometers(Long Term Average)(100km) (varchar(50), null)
Miles Per Gallon(Instant)(mpg) (varchar(50), null)
Miles Per Gallon(Long Term Average)(mpg) (varchar(50), null)
Average trip speed(whilst moving only)(km/h) (varchar(50), null)
Average trip speed(whilst stopped or moving)(km/h) (varchar(50), null)
Trip average Litres 100 KM(100km) (varchar(50), null)
Trip average MPG(mpg) (varchar(50), null)
Trip average KPL(kpl) (varchar(50), null)
NOx Post SCR(ppm) (varchar(50), null)
NOx Pre SCR(ppm) (varchar(50), null)
Fuel Level (From Engine ECU)(%) (varchar(50), null)
O2 Sensor1 Equivalence Ratio (varchar(50), null)
O2 Sensor2 Equivalence Ratio (varchar(50), null)
O2 Volts Bank 1 sensor 1(V) (varchar(50), null)
O2 Volts Bank 1 sensor 2(V) (varchar(50), null)
O2 Volts Bank 1 sensor 3(V) (varchar(50), null)
O2 Volts Bank 1 sensor 4(V) (varchar(50), null)
O2 Volts Bank 2 sensor 1(V) (varchar(50), null)
O2 Volts Bank 2 sensor 2(V) (varchar(50), null)
O2 Volts Bank 2 sensor 3(V) (varchar(50), null)
O2 Volts Bank 2 sensor 4(V) (varchar(50), null)
Percentage of City driving(%) (varchar(50), null)
Percentage of Highway driving(%) (varchar(50), null)
Percentage of Idle driving(%) (varchar(50), null)
Throttle Position(Manifold)(%) (varchar(50), null)
Absolute Throttle Position(%) (varchar(50), null)
Accelerator PedalPosition(%) (varchar(50), null)
Accelerator PedalPosition(°) (varchar(50), null)
Relative Accelerator Pedal Position(%) (varchar(50), null)
Relative Throttle Position(%) (varchar(50), null)
Horsepower (At the wheels)(hp) (varchar(50), null)
Barometric pressure (from vehicle)(psi) (varchar(50), null)
Fuel pressure(psi) (varchar(50), null)
Intake Manifold Pressure(psi) (varchar(50), null)
Fuel Rail Pressure(psi) (varchar(50), null)
Fuel Rail Pressure (relative to manifold vacuum)(psi) (varchar(50), null)
Evap System Vapour Pressure(Pa) (varchar(50), null)
Engine RPM(rpm) (varchar(50), null)
GPS Satellites (varchar(50), null)
Acceleration Sensor(X axis)(g) (varchar(50), null)
Acceleration Sensor(Y axis)(g) (varchar(50), null)
Acceleration Sensor(Z axis)(g) (varchar(50), null)
Acceleration Sensor(Total)(g) (varchar(50), null)
O2 Sensor2 Equivalence Ratio (varchar(50), null)
O2 Sensor7 Equivalence Ratio (varchar(50), null)
O2 Sensor1 Equivalence Ratio (varchar(50), null)
O2 Sensor1 Equivalence Ratio(alternate) (varchar(50), null)
O2 Sensor3 Equivalence Ratio (varchar(50), null)
O2 Sensor4 Equivalence Ratio (varchar(50), null)
Tuning Advance(°) (varchar(50), null)
Air Fuel Ratio(HeatandX 1) (varchar(50), null)
Air Fuel Ratio(Commanded)(X 1) (varchar(50), null)
Commanded Equivalence Ratio(lambda) (varchar(50), null)
Fuel flow rate minute(gal min) (varchar(50), null)
Fuel flow rate hour(gal hr) (varchar(50), null)
Mass Air Flow Rate(g/s) (varchar(50), null)
Ambient air temp(°C) (varchar(50), null)
Intake Air Temperature(°C) (varchar(50), null)
Catalyst Temperature (Bank 1 Sensor 1)(°C) (varchar(50), null)
Catalyst Temperature (Bank 1 Sensor 2)(°C) (varchar(50), null)
Catalyst Temperature (Bank 2 Sensor 1)(°C) (varchar(50), null)
Catalyst Temperature (Bank 2 Sensor 2)(°C) (varchar(50), null)
Engine Coolant Temperature(°C) (varchar(50), null)
Transmission Temperature(Method 1)(°C) (varchar(50), null)
Transmission Temperature(Method 2)(°C) (varchar(50), null)
Engine Oil Temperature(°C) (varchar(50), null)
0-100mph Time(s) (varchar(50), null)
0-200mph Time(s) (varchar(50), null)
0-30mph Time(s) (varchar(50), null)
0-60mph Time(s) (varchar(50), null)
1-4 mile time(s) (varchar(50), null)
1-0 mile time(s) (varchar(50), null)
100-0mph Time(s) (varchar(50), null)
40-60mph Time(s) (varchar(50), null)
60-120mph Time(s) (varchar(50), null)
60-130mph Time(s) (varchar(50), null)
60-80mph Time(s) (varchar(50), null)
80-100mph Time(s) (varchar(50), null)
80-120mph Time(s) (varchar(50), null)
Trip Time(since journey start(s) (varchar(50), null)
Trip time(whilst stationary)(s) (varchar(50), null)
Run time since engine start(s) (varchar(50), null)
Trip time(whilst moving)(s) (varchar(50), null)
Turbo Boost & Vacuum Gauge(psi) (varchar(50), null)
Turbo Pressure Control(psi) (varchar(50), null)

Apêndice B – Scripts SQL

```

/*SELECT *
FROM [Torque].[dbo].[UAL] */

/*select * from [Torque].[dbo].[UAL] where [ Device Time] not in
(select top ((select COUNT(*) from [Torque].[dbo].[UAL] ) -5 ) [ Device Time]
from [Torque].[dbo].[UAL]) */

/*select top 5 * from [Torque].[dbo].[UAL]
order by [GPS Time] desc */

/*SELECT * FROM [Torque].[dbo].[UAL]
WHERE convert(datetime, ([ Device Time])) >= DateADD(mi, -30,
Current_TimeStamp) */

/*SELECT * FROM [Torque].[dbo].[UAL]
WHERE (select max(convert(datetime, [ Device Time])) from
[Torque].[dbo].[UAL]) <= DateADD(mi, -5, Current_TimeStamp)
order by [ Device Time] */

/*Select MAX([ Device Time]) from [Torque].[dbo].[UAL]
Select DateAdd(mi, -10, MAX([ Device Time])) from [Torque].[dbo].[UAL] */

```

Exemplo: Devolve o valor **MINIMO** de "Fuel" dos registos dos **ultimos 10 MINUTOS** dos registos inseridos na tabela (tem

```

Select IDVeiculo, MIN([Fuel Remaining (Calculated from vehicle profile) (%)])
from [Torque].[dbo].[UAL]
WHERE [ Device Time] between
(Select IDVeiculo, DateAdd(mi, -10, MAX([ Device Time])) from
[Torque].[dbo].[UAL] ) and (Select IDVeiculo, MAX([ Device Time]) from
[Torque].[dbo].[UAL])

```

/*

Query (no Script) que alimenta a tabela final de resultados por veículo - TORQUE_RESULTADOS

Resultados dos registos dos últimos 10 minutos - Alterável - com histórico (Data Hora)

Informação de dados seleccionados (por veiculo no período de tempo considerado):

- N° de Registos
- Valor Minimo de "Fuel Remaining"
- Valor Médio "Engine RPM"
- Valor Miáximo "Engine Coolant Temperature"
- Valor Minimo "Engine Coolant Temperature"

*/

```

INSERT INTO Torque.dbo.TORQUE_RESULTADOS
SELECT

```

```

    GETDATE() as DataHora,
    TORQUE.IDVeiculo,
    SUM(1) AS NUM_REG,
    MIN(TORQUE.[ Longitude]),
    MIN(TORQUE.[ Latitude]),
    MIN(TORQUE.[Engine Load(%)]),

```

```

    MIN(TORQUE.[Fuel Remaining (Calculated from vehicle profile)(%)]),
    MIN(TORQUE.[Fuel used (trip)(gal)]),
    MIN(TORQUE.[COâ,, in g km (Average)(g km)]),
    MIN(TORQUE.[Distance travelled since codes cleared(km)]),
    MIN(TORQUE.[Volumetric Efficiency (Calculated)(%)]),
    MIN(TORQUE.[EGR Commanded(%)]),
    MIN(TORQUE.[EGR Error(%)]),
    MIN(TORQUE.[Kilometers Per Litre(Long Term Average)(kpl)]),
    MIN(TORQUE.[Litres Per 100 Kilometer(Long Term Average)(l 100km)]),
    MIN(TORQUE.[Trip average Litres 100 KM(l 100km)]),
    MIN(TORQUE.[Percentage of City driving(%)]),
    MIN(TORQUE.[Percentage of Highway driving(%)]),
    MIN(TORQUE.[Percentage of Idle driving(%)]),
    MIN(TORQUE.[Throttle Position(Manifold)(%)]),
    MIN(TORQUE.[Barometric pressure (from vehicle)(psi)]),
    MIN(TORQUE.[Intake Manifold Pressure(psi)]),
    MIN(TORQUE.[Fuel Rail Pressure(psi)]),
    AVG(CAST(TORQUE.[Engine RPM(rpm)] AS DECIMAL(8, 2) )),
    MIN(TORQUE.[Acceleration Sensor(X axis)(g)]),
    MIN(TORQUE.[Acceleration Sensor(Y axis)(g)]),
    MIN(TORQUE.[Acceleration Sensor(Z axis)(g)]),
    MIN(TORQUE.[Air Fuel Ratio(Measured)( 1)]),
    MAX(TORQUE.[Engine Coolant Temperature(Â°C)]),
    MIN(TORQUE.[Turbo Boost & Vacuum Gauge(psi)])
/* Acrescentar aqui novos campos */
FROM
    Torque.dbo.TORQUE AS TORQUE,
    ( SELECT IDVeiculo, DateAdd(mi, -10, MAX([ Device Time])) AS TIME_FROM,
      DateAdd(mi, 0, MAX([ Device Time])) AS TIME_TO
    FROM Torque.dbo.TORQUE GROUP BY IDVeiculo ) AS DATAS
WHERE
    TORQUE.IDVeiculo = DATAS.IDVeiculo AND
    ( TORQUE.[ Device Time] BETWEEN DATAS.TIME_FROM AND DATAS.TIME_TO )
GROUP BY
    TORQUE.IDVeiculo

```

Queries para colocar na Aplicação:

```

/* Devolve-nos o ultimo registo (mais actual) por veiculo, da tabela
TORQUE_RESULTADOS */
SELECT [DataHora]
      , [IDVeiculo]
      , [NUM_REG]
      , [Longitude]
      , [Latitude]
      , [Engine Load(%)]
      , [Fuel Remaining (Calculated from vehicle profile)(%)]
      , [Fuel used (trip)(gal)]
      , [COâ,, in g km (Average)(g km)]
      , [Distance travelled since codes cleared(km)]
      , [Volumetric Efficiency (Calculated)(%)]
      , [EGR Commanded(%)]
      , [EGR Error(%)]
      , [Kilometers Per Litre(Long Term Average)(kpl)]
      , [Litres Per 100 Kilometer(Long Term Average)(l 100km)]
      , [Trip average Litres 100 KM(l 100km)]
      , [Percentage of City driving(%)]
      , [Percentage of Highway driving(%)]
      , [Percentage of Idle driving(%)]
      , [Throttle Position(Manifold)(%)]

```

```

, [Barometric pressure (from vehicle) (psi)]
, [Intake Manifold Pressure (psi)]
, [Fuel Rail Pressure (psi)]
, [Engine RPM (rpm)]
, [Acceleration Sensor (X axis) (g)]
, [Acceleration Sensor (Y axis) (g)]
, [Acceleration Sensor (Z axis) (g)]
, [Air Fuel Ratio (Measured) ( 1)]
, [Engine Coolant Temperature (°C)]
, [Turbo Boost & Vacuum Gauge (psi)]
FROM [Torque].[dbo].[TORQUE_RESULTADOS]
WHERE [DataHora] in (select MAX(DataHora) from
[Torque].[dbo].[TORQUE_RESULTADOS])

/* Devolve o Valor Mensal do Histórico (TORQUE_Resultados) */
/* Não esquecer de acrescentar campos se forem acrescentados novos parametros
na tabela TORQUE_RESULTADOS */
SELECT
TORQUE.IDVeiculo,
DATAS.TIME_FROM, DATAS.TIME_TO,
SUM(CAST(TORQUE.[NUM_REG] AS DECIMAL)) AS NUM_REG,
MIN(TORQUE.[ Longitude]),
MIN(TORQUE.[ Latitude]),
MIN(TORQUE.[Engine Load(%)]),
MIN(TORQUE.[Fuel Remaining (Calculated from vehicle profile) (%)]),
MIN(TORQUE.[Fuel used (trip) (gal)]),
MIN(TORQUE.[COâ,, in g km (Average) (g km)]),
MIN(TORQUE.[Distance travelled since codes cleared(km)]),
MIN(TORQUE.[Volumetric Efficiency (Calculated) (%)]),
MIN(TORQUE.[EGR Commanded(%)]),
MIN(TORQUE.[EGR Error(%)]),
MIN(TORQUE.[Kilometers Per Litre(Long Term Average) (kpl)]),
MIN(TORQUE.[Litres Per 100 Kilometer(Long Term Average) (l 100km)]),
MIN(TORQUE.[Trip average Litres 100 KM(l 100km)]),
MIN(TORQUE.[Percentage of City driving(%)]),
MIN(TORQUE.[Percentage of Highway driving(%)]),
MIN(TORQUE.[Percentage of Idle driving(%)]),
MIN(TORQUE.[Throttle Position (Manifold) (%)]),
MIN(TORQUE.[Barometric pressure (from vehicle) (psi)]),
MIN(TORQUE.[Intake Manifold Pressure (psi)]),
MIN(TORQUE.[Fuel Rail Pressure (psi)]),
AVG(CAST(TORQUE.[Engine RPM (rpm)] AS DECIMAL(8, 2) )),
MIN(TORQUE.[Acceleration Sensor (X axis) (g)]),
MIN(TORQUE.[Acceleration Sensor (Y axis) (g)]),
MIN(TORQUE.[Acceleration Sensor (Z axis) (g)]),
MIN(TORQUE.[Air Fuel Ratio (Measured) ( 1)]),
MAX(TORQUE.[Engine Coolant Temperature (°C)]),
MIN(TORQUE.[Turbo Boost & Vacuum Gauge (psi)])
FROM
Torque.dbo.TORQUE_RESULTADOS AS TORQUE, (
SELECT IDVeiculo, DateAdd(m, -1, MAX(DataHora)) AS TIME_FROM,
DateAdd(m, 0, MAX(DataHora)) AS TIME_TO FROM
Torque.dbo.[TORQUE_RESULTADOS] GROUP BY IDVeiculo
) AS DATAS
WHERE
TORQUE.IDVeiculo = DATAS.IDVeiculo AND ( TORQUE.[DataHora] BETWEEN
DATAS.TIME_FROM AND DATAS.TIME_TO )
GROUP BY
TORQUE.IDVeiculo, DATAS.TIME_FROM, DATAS.TIME_TO

```

Apêndice C - Dispositivo ELM327

O ELM327 é um [microcontrolador](#) programado produzido pela ELM Electronics para traduzir a interface de [diagnóstico on-board](#) (OBD) encontrada na maioria dos carros modernos. O protocolo de comando ELM327 é um dos mais populares padrões de interface PC-to-OBD e também é implementado por outros fornecedores.

O ELM327 original é implementado no microcontrolador [PIC18F2480](#) da [Microchip Technology](#).

O ELM327 faz parte de uma família de tradutores de OBD da ELM Electronics. Outras variantes implementam apenas um subconjunto dos protocolos OBD.



O ELM327 abstrai o protocolo de baixo nível e apresenta uma interface simples que pode ser chamada por meio de um [UART](#), normalmente por uma ferramenta de diagnóstico portátil ou por um programa de computador conectado por [USB](#), [RS-232](#), [Bluetooth](#) ou [Wi-Fi](#). Novos aplicativos incluem smartphones.

Há um grande número de programas disponíveis que se conectam ao ELM327. [\[1\]](#)

A função desse software pode incluir:

- Relatório de códigos de erro
- Limpar códigos de erro

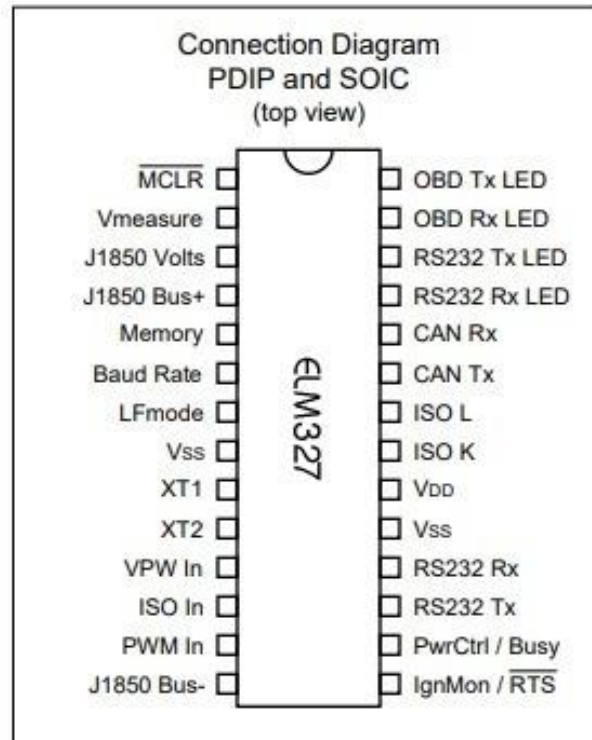
Funções ELM327

- Ler os códigos de problemas de diagnóstico, genéricos e específicos do fabricante, e exibir seu significado (mais de 3.000 definições de códigos genéricos no banco de dados).
- Limpar os códigos de problemas e desligar o MIL ("Luz indicadora de mau funcionamento", mais comumente conhecida como "Check Engine Light")
- Exibir dados atuais do sensor
- RPM do motor
- Valor da carga calculada
- Temperatura do Líquido Refrigerante
- Status do sistema de combustível
- Velocidade do veículo
- Guarnição de Combustível de Curto Prazo
- Guarnição de combustível a longo prazo
- Pressão do coletor de admissão
- Avanço do tempo
- Temperatura do ar de admissão
- Taxa de fluxo de ar
- Posição Absoluta do Acelerador
- Tensões do sensor de oxigênio / guarnições de combustível de curto prazo associadas
- Status do sistema de combustível
- Pressão de Combustível

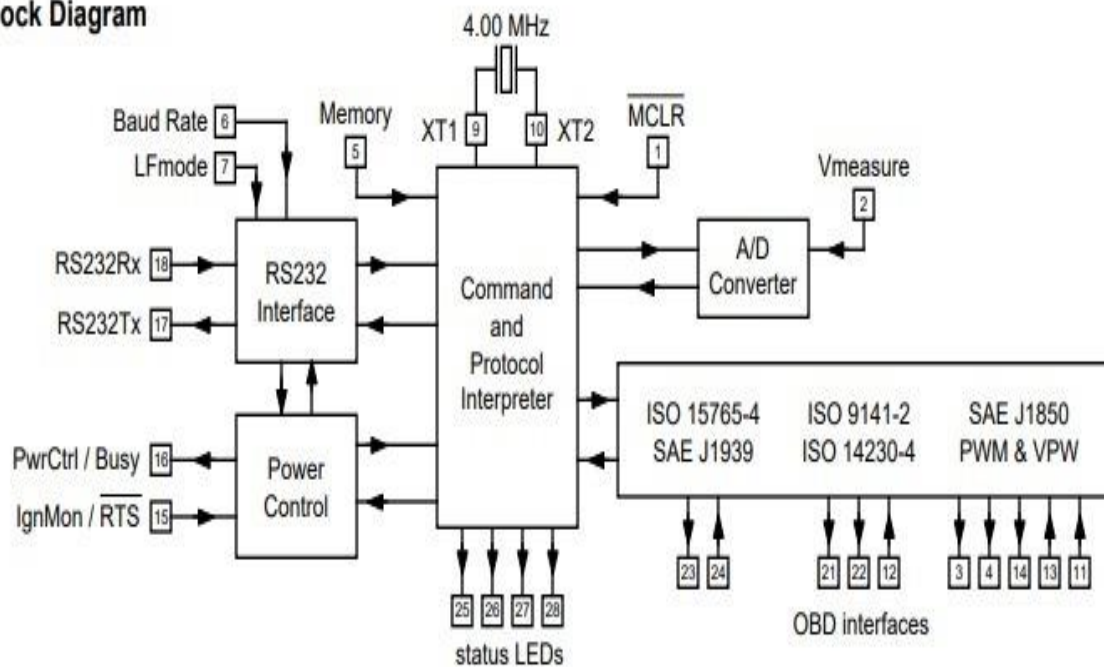
Protocolos suportados pelo ELM327

Os protocolos suportados pelo ELM327 são:

- [SAE J1850](#) PWM (41,6 [kbit / s](#))
- [SAE J1850](#) VPW (10,4 kbit / s)
- ISO 9141-2 (5 baud init, 10,4 kbit / s)
- ISO 14230-4 KWP (5 baud init, 10,4 kbit / s)
- ISO 14230-4 KWP (inicialização rápida, 10,4 kbit / s)
- ISO 15765-4 [CAN](#) (ID de 11 bits, 500 kbit / s)
- ISO 15765-4 CAN (ID de 29 bits, 500 kbit / s)
- [ISO 15765-4 CAN \(ID de 11 bits, 250 kbit / s\)](#)
- ISO 15765-4 CAN (ID de 29 bits, 250 kbit / s)
- SAE J1939 (250kbit / s)
- SAE J1939 (500kbit / s)



Block Diagram



Página intencionalmente em branco