

SISTEMA DE GESTÃO DE TRANSPORTE DE ENVIOS PARCELARES

Francisco Alves Pacheco



Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Sistemas e Planeamento Industrial

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2013

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: Francisco Alves Pacheco, N° 1070224, 1070224@isep.ipp.pt

Orientação científica: Professor António José Galvão Ramos, agr@isep.ipp.pt

Co-orientação científica: Professor Manuel Joaquim Pereira Lopes, mpl@isep.ipp.pt

Empresa: Caetano Parts – Grupo Salvador Caetano

Orientação Empresa: Eng.º Ricardo Manuel Oliveira Soares,
ricardo.soares@caetanoretail.pt



Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Sistemas e Planeamento Industrial

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

18 de novembro de 2013

Agradecimentos

A realização deste trabalho envolveu a colaboração de algumas pessoas, que directa ou indirectamente, estiveram envolvidas durante este processo de aprendizagem e maturação, a todos eles um sentido Obrigado. Com especial agradecimento:

Ao Professor António Galvão Ramos pela orientação e disponibilidade demonstrada ao longo da realização de todo o trabalho.

Ao Eng. Ricardo Soares pela oportunidade de realização do trabalho integrado numa empresa de referência, como a Caetano Parts e pela confiança depositada durante todo o processo.

Aos colaboradores da empresa, em especial ao António Mota, à Angelina Morais, à Sónia Fernandes e à Balbina Cunha pela forma como me receberam e me integraram no dia-a-dia da Caetano Parts.

Ao Eng. Tiago Teixeira pela completa disponibilidade e apoio incondicional no esclarecimento de dúvidas tecnológicas que surgiram.

Por último mas não menos importante aos Meus Pais, à Minha Irmã e à Ana pelo incansável apoio durante todo o meu percurso académico, e em especial pela motivação e confiança na realização desta etapa.

Resumo

No ambiente empresarial actual, cada vez mais competitivo e exigente, é um factor fundamental para o sucesso das empresas a sua capacidade de atingir e melhorar os níveis de satisfação exigidos pelos clientes. Para identificar as melhorias a implementar, as empresas devem ser capazes de monitorizar e controlar todas as suas actividades e processos. O acompanhamento realizado às actividades delegadas a empresas externas, como por exemplo o transporte de mercadorias, é dificultado quando os prestadores destes serviços não possuem ferramentas de apoio que disponibilizem informação necessária para o efeito.

A necessidade de colmatar esta dificuldade na recolha da informação durante a distribuição de uma encomenda na empresa Caetano Parts, uma empresa de revenda de peças de substituição automóvel, levou ao desenvolvimento de uma ferramenta que permite fazer o seguimento de uma encomenda em todas as suas fases, permitindo ao responsável pelas operações acompanhar o estado da encomenda desde o instante em que a encomenda é colocada, passando pelo seu processamento dentro das instalações, até à sua entrega ao cliente.

O sistema desenvolvido é composto por dois componentes, o *front-end* e o *back-end*. O *front-end* é composto por uma aplicação *web*, e por uma aplicação *Android* para dispositivos móveis. A aplicação *web* disponibiliza a gestão da base de dados, o acompanhamento do estado da encomenda e a análise das operações. A aplicação *Android* é disponibilizada às empresas responsáveis pelo transporte das encomendas e possibilita a actualização *online* da informação acerca do processo de entrega. O *back-end* é composto pela unidade de armazenamento e processamento da informação e encontra-se alojado num servidor com ligação à *internet*, disponibilizando uma interface com o serviço móvel do tipo serviço *web*.

A concepção, desenvolvimento e descrição das funcionalidades desta ferramenta são abordadas ao longo do trabalho. Os testes realizados ao longo do desenvolvimento validaram o correcto funcionamento da ferramenta, estando pronta para a realização de um teste piloto.

Palavras-Chave

Nível serviço, transportadora, serviço *web*, *android*, *outsourcing*.

Abstract

Today's business environment, increasingly more competitive and challenging, the achievement and improvement of customer satisfaction is the key factor in one's company success. These improving companies must be able to keep up and control all activities and processes of their operations. But some outsourced services that usually are part of the business like carriers make this control difficult only because they don't own proper communication tools.

At Caetano Parts, that is an aftermarket company of automotive replacement parts, it is faced this monitoring problem when transporting daily company goods. So it is found the necessity to overcome it just by gathering information during all the transportation processes from the very moment the order is placed until it is delivered to the customer.

All these issues make the company to develop a communication system that's divided in two parts, front-end and back-end. The front-end is basically a web application that provides database management like tracking the order and analysing entirely the operation and an Android application that carriers will own and allow them to update all the related information during the service. The back-end part aims to be the storage and information processing unit that is hosted by an internet server. This provides all the background support of the entire system.

The system's design, development and demonstration is discussed throughout this paper. All the tests performed during the development of the project confirmed that all modules worked correctly. However it is necessary to execute one more extensive test to do a real validation.

Keywords

Service level, carrier, web service, android, outsourcing.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XV
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. PROBLEMA.....	1
1.3. MOTIVAÇÃO	2
1.4. OBJECTIVOS.....	3
1.5. ORGANIZAÇÃO DO RELATÓRIO	4
2. REVISÃO BIBLIOGRÁFICA	5
2.1. O MERCADO DE PEÇAS DE SUBSTITUIÇÃO AUTOMÓVEL.....	5
2.1.1. <i>O Mercado de Peças de substituição Automóvel em Portugal</i>	8
2.2. SERVIÇO AO CLIENTE	14
2.2.1. <i>Medir o Nível de Serviço</i>	18
2.3. O OUTSOURCING DO TRANSPORTE.....	19
2.3.1. <i>O que é o Outsourcing?</i>	20
2.3.2. <i>Motivações e riscos do Outsourcing</i>	21
2.3.3. <i>Critérios de Selecção de serviços de transporte</i>	24
2.3.4. <i>Monitorização da Performance</i>	26
2.4. SOLUÇÕES PARA SISTEMAS TRANSPORTE DE ENVIOS PARCELARES	30
2.4.1. <i>Cloud Logistics</i>	31
2.4.2. <i>Smart Freight Ware</i>	32
2.4.3. <i>Precision</i>	33
3. DESCRIÇÃO DO PROBLEMA - CASO CAETANO PARTS	35
3.1. ENQUADRAMENTO.....	35
3.2. PROCESSO ACTUAL – TRATAMENTO DE ENCOMENDA.....	37
3.3. PROCESSO PROPOSTO – TRATAMENTO DE ENCOMENDA	40
4. DESENVOLVIMENTO DA FERRAMENTA	43
4.1. ARQUITECTURA	43
4.2. FUNCIONALIDADES DA FERRAMENTA	45
4.2.1. <i>Gestão de Dados</i>	45
4.2.2. <i>Recolha de Dados da Factura</i>	51

4.2.3.	<i>Aplicação Android</i>	54
4.2.4.	<i>Aplicação Web</i>	62
4.3.	DEMONSTRAÇÃO DAS FUNCIONALIDADES	70
4.3.1.	<i>Gestão da base de dados</i>	71
4.3.2.	<i>Recolha de dados das encomendas</i>	78
4.3.3.	<i>Lista de picking</i>	80
4.3.4.	<i>Aplicação móvel</i>	81
4.3.5.	<i>Dashboard</i>	83
5.	CONCLUSÕES	89
5.1.	INTRODUÇÃO.....	89
5.2.	LIMITAÇÕES E DESENVOLVIMENTOS FUTUROS	90
	REFERÊNCIAS DOCUMENTAIS	93
	ANEXO A. TECNOLOGIAS DESENVOLVIMENTO DO <i>BACK-END</i>	101
	ANEXO B. TECNOLOGIAS DE DESENVOLVIMENTO DO <i>FRONT-END</i>	115
	ANEXO C. AMBIENTE DE DESENVOLVIMENTO INTEGRADO	125

Índice de Figuras

Figura 1 Registo de veículos ligeiros a nível mundial, entre 2003 e 2015.....	6
Figura 2 Rentabilidade Pós- Venda.....	7
Figura 3 Idade Média do Parque Automóvel em 2000, 2008 e 2011	10
Figura 4 Países onde opera o Grupo Salvador Caetano	14
Figura 5 Uso dos serviços para o aumento do valor do produto	16
Figura 6 Plano cíclico de planeamento e monitorização	17
Figura 7 Rácios dos custos das actividades logísticas.....	20
Figura 8 Factores de escolha transportadora	25
Figura 9 Layout plataforma da Cloud Logistics.....	31
Figura 10 Layout plataforma SmartTMS™	32
Figura 11 Layout Plataforma Precision.....	33
Figura 12 Mapa de actuação, por zona, das transportadoras contratadas pela Caetano Parts	36
Figura 13 Processo Actual de Tratamento de uma Encomenda	38
Figura 14 Quadro definição rotas (esquerda) e local de recolha dos pacotes por rota (direita)	39
Figura 15 Proposta de Alteração do Processo de Tratamento de uma Encomenda	42
Figura 16 Arquitectura do Sistema.....	44
Figura 17 Exemplo id de identificação de transportadora.....	48
Figura 18 Relação entre tabelas da base de dados.....	52
Figura 19 Processo de recolha de dados a partir da factura	53
Figura 20 Extracto de código para criação base de dados SQLite	56
Figura 21 Pedido <i>Http Post</i> em <i>Android</i>	57
Figura 22 Resposta dados formato JSON	59
Figura 23 Código XML do <i>layout</i> de uma <i>list view</i>	59
Figura 24 Código XML do <i>layout</i> de uma linha de <i>list view</i>	60
Figura 25 Janela principal aplicação móvel (a) e <i>dialog box</i> da interacção com linha da lista (b) ..	61
Figura 26 Extracto de código para obter posição geográfica	62
Figura 27 Janela inicial aplicação <i>web</i>	63
Figura 28 Janela “Localizar Envio” aplicação <i>web</i>	64
Figura 29 Janela “Localizar Envio Avançado” aplicação <i>web</i>	64
Figura 30 Janela “Imprimir Etiqueta” aplicação <i>web</i>	65
Figura 31 Exemplo de etiqueta de despacho	65
Figura 32 Página <i>web</i> da lista de <i>picking</i>	66
Figura 33 <i>Layout Dashboard</i>	67
Figura 34 Tabela detalhes globais das operações.....	67

Figura 35 Gráficos da distribuição das encomendas pelas fases e pelas transportadoras	68
Figura 36 Tabela da distribuição das encomendas pelas diferentes fases agrupadas por transportadora	68
Figura 37 Gráfico evolução do nível serviço da transportadora nos últimos 6 meses	69
Figura 38 Detalhes Encomendas <i>Dashboard</i>	69
Figura 39 Janela gestão dados dos motoristas	70
Figura 40 Criação utilizador aplicação <i>web</i>	72
Figura 41 Resultado tabela <i>utilizadores</i> após criação dos utilizadores	72
Figura 42 Alteração dados utilizador aplicação <i>web</i>	73
Figura 43 Resultado tabela <i>utilizadores</i> após edição dos dados dos utilizadores.....	73
Figura 44 Página inicial e formulário de <i>login</i>	74
Figura 45 Criação transportadoras aplicação <i>web</i>	74
Figura 46 Resultado tabela <i>transportadoras</i> após criação	75
Figura 47 Criação motoristas aplicação <i>web</i>	76
Figura 48 Resultado tabela <i>motoristas</i> após criação	77
Figura 49 Criação clientes aplicação <i>web</i>	77
Figura 50 Resultado tabela <i>clientes</i> após criação	78
Figura 51 Vista pormenorizada dos dados da factura	79
Figura 52 Resultado recolha dados da encomenda da Figura 51	79
Figura 53 Resultado tabela <i>envios</i> após processamento das facturas.....	80
Figura 54 Lista de <i>picking</i> após criação do registo das encomendas	80
Figura 55 Etiqueta gerada com os dados da encomenda	81
Figura 56 Actualização tabela <i>envios</i> após confirmação de encomenda preparada	81
Figura 57 Obtenção lista aplicação móvel.....	82
Figura 58 Tabela <i>envios</i> após de actualização da lista de encomendas na aplicação móvel	83
Figura 59 Janela de confirmação de entrega da encomenda através da aplicação móvel	84
Figura 60 Actualização tabela <i>envios</i> após <i>interacção</i> da aplicação móvel.....	85
Figura 61 <i>Dashboard</i> com a informação dos dados da demonstração	85
Figura 62 Vista detalhada das encomendas da <i>dashboard</i>	85
Figura 63 <i>Dashboard</i> após adição das encomendas.....	87
Figura 64 Vista pormenorizada gráfico evolução nível serviço das transportadoras nos últimos seis meses	87
Figura 65 Arquitectura da Stack dos serviços <i>web</i>	107
Figura 66 Exemplo <i>RESTful Web Service</i>	108
Figura 67 Exemplo estrutura de dados JSON.....	112
Figura 68 Código de Barras 1D vs 2D	113
Figura 69 Exemplos Códigos de Barras 2D	113
Figura 70 Exemplo de código Java	116
Figura 71 Exemplo de código HTML	117

Figura 72 Exemplo de código PHP	117
Figura 73 Exemplo de código JavaScript.....	118
Figura 74 Evolução da distribuição dos principais SO móveis na Europa (Setembro 2012 – Agosto 2013)	119
Figura 75 Evolução da distribuição de SO móveis em Portugal (Setembro 2012 – Agosto 2013)	119
Figura 76 Distribuição do SDK <i>Android</i>	120
Figura 78 Exemplo de criação Actividade Android.....	123
Figura 79 Exemplo código XML para <i>layout</i> Android.....	123
Figura 80 Exemplo <i>layout</i> de uma actividade Android.....	124
Figura 81 Adição de <i>plug-in</i> no Eclipse.....	125
Figura 82 <i>Android</i> SDK Tools	126
Figura 83 Criação projecto <i>Android</i>	127

Índice de Tabelas

Tabela 1 Empregos directos da indústria automóvel na EU em 2010.....	5
Tabela 2 Mercado Veículos Particulares Novos	9
Tabela 3 Representação das Marcas Automóveis	11
Tabela 4 Razões para <i>Outsourcing</i>	22
Tabela 5 Benefícios e problemas do <i>outsourcing</i> de actividades.....	23
Tabela 6 Elementos de análise mais utilizados para avaliar performance da transportadora.....	29
Tabela 7 Elementos de análise para avaliar performance	30
Tabela 8 Razões para não renovar contractos com serviços logísticos	30
Tabela 9 Estrutura de dados da tabela <i>utilizadores</i>	46
Tabela 10 Estrutura de dados da tabela <i>clientes</i>	47
Tabela 11 Estrutura de dados da tabela <i>transportadoras</i>	48
Tabela 12 Estrutura de dados da tabela <i>motoristas</i>	49
Tabela 13 Estrutura de dados da tabela <i>envios</i>	50
Tabela 14 Estrutura de dados da classe <i>Delivery</i>	56
Tabela 15 <i>Status</i> respostas à aplicação móvel.....	58
Tabela 16 Informação utilizadores criados	71
Tabela 17 Informação utilizadores após alteração	73
Tabela 18 Informação transportadoras criadas.....	75
Tabela 19 Informação motoristas criados	76
Tabela 20 Informação clientes criados.....	78
Tabela 21 Dados encomendas demonstração <i>dashboard</i>	86
Tabela 22 Limites PostgreSQL	105
Tabela 23 Prós e Contras das Base de Dados PostgreSQL e MySQL	106
Tabela 24 Tipos MIME mais utilizados nos serviços <i>web REST</i>	111
Tabela 25 Características dos principais Códigos de Barras 2D.....	113
Tabela 26 Versões dos SDK <i>Android</i>	120

Acrónimos

ACEA	–	European Automobile Manufacturers Association
ACID	–	Atomicidade, Consistência, Isolamento, Durabilidade
API	–	Application Programming Interface
BD	–	Base de Dados
CPU	–	Central Processing Unit
CRUD	–	Create, Read, Update e Delete
CSS	–	Cascading Style Sheets
GIS	–	Geographic Information System
GPS	–	Global System Position
HTML	–	HyperText Markup Language
HTTP	–	HyperText Transfer Protocol
JIT	–	Just-In-Time
JSON	–	JavaScript Object Notation
JVM	–	Java Virtual Machine
MIME	–	Multipurpose Internet Mail Extensions
MVCC	–	Multi-Version Concurrency Control
OEM	–	Original Equipment Manufacturer
REST	–	Representational State Transfer
S.A	–	Sociedade Anónima
S.G.P.S	–	Sociedade Gestora de Participações Sociais
SDK	–	Software Development Kit
SGML	–	Standard Generalized Markup Language
SHA	–	Secure Hash Algorithm
SMTP	–	Simple Mail Transfer Protocol
SOAP	–	Simple Object Access Protocol
TMS	–	Transportation Management System
UDDI	–	Universal Description, Discovery and Integration

1. INTRODUÇÃO

Neste capítulo é feita a contextualização, descrito o problema, a motivação e o objectivo para a realização do trabalho. Por último é descrita a estrutura deste documento.

1.1. CONTEXTUALIZAÇÃO

A Caetano Parts é uma empresa que integra o Grupo Salvador Caetano, com sede no Porto, e é uma das principais empresas no mercado da venda de peças de substituição para automóveis em Portugal. A empresa centraliza a gestão de peças de substituição automóvel das marcas comercializadas pelo Grupo Salvador Caetano.

A Caetano Parts está inserida num ambiente empresarial extremamente competitivo, no qual atingir um nível de serviço ao cliente apropriado e efectivo é cada vez mais um factor fundamental para o sucesso das empresas (Rushton et al. 2010). A Caetano Parts, no âmbito do seu processo de melhoria contínua, identificou a necessidade de compreender e quantificar o nível de serviço que presta aos seus clientes e que este objectivo é dificultado pelo facto dos serviços de transporte das encomendas para os clientes, serem efectuados por empresas externas sem capacidade de fornecerem a informação necessária para quantificar o nível de serviço da Caetano Parts.

1.2. PROBLEMA

A satisfação do cliente é um factor de extrema importância nas actividades empresariais. Sendo a Caetano Parts uma empresa dedicada à venda de produtos, existe uma preocupação em compreender e quantificar qual o nível de serviço obtido na sua actividade, uma vez que

o aumento do nível de serviço conduz a um aumento de satisfação por parte do cliente (Shahin 2011).

O ciclo de vida de uma encomenda na Caetano Parts pode dividir-se em duas grandes etapas, a fase do processamento da encomenda nas instalações (armazém) e o transporte da encomenda até ao cliente. O acompanhamento da encomenda enquanto esta se encontra no armazém é realizado tendo por base os rigorosos sistemas de gestão implementados na empresa. Em contrapartida, após a saída da encomenda das instalações esta tarefa está bastante dificultada.

A Caetano Parts delega o serviço de transporte a empresas externas, maioritariamente a empresas de pequena/média dimensão e que actuam a nível regional. Devido à sua organização simples e estando inseridas num mercado pouco maduro, estas não apresentam sistemas de gestão e de informação desenvolvidos. De facto foi possível confirmar que estas se restringem a executar o serviço de transporte das encomendas, não fornecendo nenhum *feedback* acerca das operações que realizam. A recolha limitada das informações e dados sobre a distribuição das encomendas só é possível através de documentos como a guia de transporte. Contudo essa fonte verificou-se não ser fiável, devido ao não preenchimento dos documentos pelos motoristas e ao facto de quando são devolvidos os documentos, estes não serem devolvidos em tempo oportuno para se proceder ao tratamento dos dados.

Como forma de ultrapassar a dificuldade de acompanhar a encomenda após a sua recolha pela transportadora e da recolha da informação necessária para avaliar o serviço prestado, surgiu a necessidade de criar uma solução, de baixo custo, capaz de possibilitar ao expedidor acompanhar as operações e em simultâneo construir uma fonte de informação que permita entender e avaliar o serviço prestado.

1.3. MOTIVAÇÃO

A principal motivação no desenvolvimento deste projecto é criação de uma ferramenta inovadora que permita controlar e monitorizar o nível de serviço ao cliente, uma vez que não foram encontradas no mercado ferramentas que possibilitem o acompanhamento de uma encomenda ao longo de todo o seu processo na óptica da entidade expedidora, quando esta recorre a diferentes transportadoras.

Para a Caetano Parts, o desenvolvimento desta ferramenta é vista como uma vantagem competitiva face à concorrência, permitindo reforçar o conhecimento acerca da actuação da empresa no serviço prestado aos clientes. Este conhecimento possibilitará, por um lado, perceber em que nível se encontra e por outro auxiliar na identificação dos pontos de melhoria a implementar.

Ao autor, a realização deste projecto, sendo ele multidisciplinar, levará ao enriquecimento significativo de conhecimentos e competências académicas e profissionais.

1.4. OBJECTIVOS

O objectivo principal deste trabalho é dotar a empresa de uma ferramenta necessária para a avaliação do serviço prestado ao cliente. A solução apresentada passa pelo desenvolvimento de uma ferramenta de recolha de dados e sistematização da informação do processamento de encomendas, constituída por duas componentes, *front-end* e *back-end*. O *front-end* é constituído pelas aplicações de interface com o utilizador, como é o caso da aplicação *web* e da aplicação móvel fornecida às empresas de transporte. O *back-end* será constituído pela unidade de armazenamento e processamento dos dados.

Uma vez que a aplicação para dispositivos móveis será fornecida às empresas de transporte, pretende-se que esta seja desenvolvida com recurso a soluções *open-source*, tentando diminuir os custos associados com a implementação e manutenção do sistema. A unidade de armazenamento e processamento dos dados será alojada num servidor existente na empresa com ligação à internet e será dotado de uma interface do tipo serviço *web* para a comunicação com a aplicação móvel, assegurando um elevado grau de interoperabilidade, disponibilidade e padronização.

A solução deve apresentar as seguintes funcionalidades:

1. Aplicação móvel:

- Listagem das entregas a serem realizadas pelo motorista;
- Registo das coordenadas geográficas no momento da submissão dos dados.

2. Aplicação *web*:

- Consulta do estado da encomenda;

- Gestão da base de dados;
- Listagem de encomendas em processo de *picking*;
- *Dashboards* com dados diários, semanais, mensais e anuais.

1.5. ORGANIZAÇÃO DO RELATÓRIO

Esta dissertação está estruturada em cinco grandes capítulos.

O capítulo 1 introduz o tema da dissertação, define o problema, a motivação e os objectivos para a realização do projecto.

No capítulo 2 é caracterizado o mercado de peças de substituição automóvel na Europa e em Portugal, fazendo referência aos principais intervenientes do mercado em Portugal. É abordado o tema do serviço ao cliente, a importância do transporte nessa área, a forma como pode ser monitorizado e quais os sistemas de gestão existentes no mercado para transporte de envios parcelares.

No capítulo 3 é realizado um enquadramento do problema da Caetano Parts, é descrito o processo actual de processamento de uma encomenda e é apresentada uma proposta de alteração.

O capítulo 4 descreve as tecnologias de desenvolvimento utilizadas, o desenvolvimento da solução, abordando a arquitectura e as funcionalidades e é apresentada uma demonstração da ferramenta.

Por fim no capítulo 5 é feita uma síntese de resultados do projecto, são apresentadas as suas limitações e são sugeridos possíveis desenvolvimentos futuros.

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo é apresentado o mercado de peças automóvel na Europa e em Portugal, incluindo uma referência aos principais intervenientes deste mercado em Portugal. É dada atenção à importância do nível de serviço ao cliente e como esse pode ser avaliado. Dentro deste ponto, é analisada a influência do serviço do transporte, a forma como este é abordado através do outsourcing, as razões que levam a esta decisão e a forma de selecção e avaliação/monitorização do mesmo. Por fim, são apresentadas soluções existentes no mercado, para a monitorização de serviços de transporte de envios parcelares.

2.1. O MERCADO DE PEÇAS DE SUBSTITUIÇÃO AUTOMÓVEL

No ano de 2010, o volume de negócios da indústria automóvel europeia foi de cerca de 536 mil milhões de euros, sendo uma das indústrias com maior relevo no contexto económico europeu, não só no volume de negócios, mas sobretudo no nível de emprego associado (Tabela 1). Segundo dados da *European Automobile Manufacturers Association* (ACEA), a indústria automóvel europeia representa 27% da produção mundial de viaturas e 24% das vendas mundiais de viaturas particulares, com 13,4 milhões de registos. No mercado de

Tabela 1 Empregos directos da indústria automóvel na EU em 2010 (Adaptado de Basílio 2013)

<i>Empresas > 20 assalariados</i>	<i>EU 27</i>	<i>DE</i>	<i>BE</i>	<i>ES</i>	<i>FR</i>	<i>IT</i>	<i>PT</i>	<i>UK</i>
<i>Empregos directos (milhares)</i>	2420	798	43	164	260	184	23	177

trabalho a contribuição desta indústria é de 2,4 milhões de empregos directos e 10,3 milhões de empregos indirectos, valores que representam cerca de 5,6% dos postos de trabalho de toda a Europa (ACEA 2011).

Tendo em conta os estudos efectuados pela empresa de consultoria BIPE¹, o mercado automóvel europeu finalizou o ano de 2012 com 13,7 milhões de novos registos de veículos automóveis ligeiros, uma diminuição de 23%, em relação a 2007, ano que antecede a chegada da crise económica e financeira. Com o mercado europeu em contracção, prevê-se que o crescimento no mercado automóvel mundial seja influenciado essencialmente pelos países emergentes, principalmente os “BRIC” (Brasil, Rússia, Índia e China) (Basílio 2013). A Figura 1 apresenta a variação do número de registos automóvel a nível mundial desde 2003 e as previsões até 2015.

Neste contexto, o mercado das peças de substituição para automóvel contribui significativamente para os proveitos provenientes da cadeia de valor automóvel. Contudo, comparando com a venda de automóveis novos, este constitui uma percentagem menor de receitas (Fernihough & Gyimesi 2008).

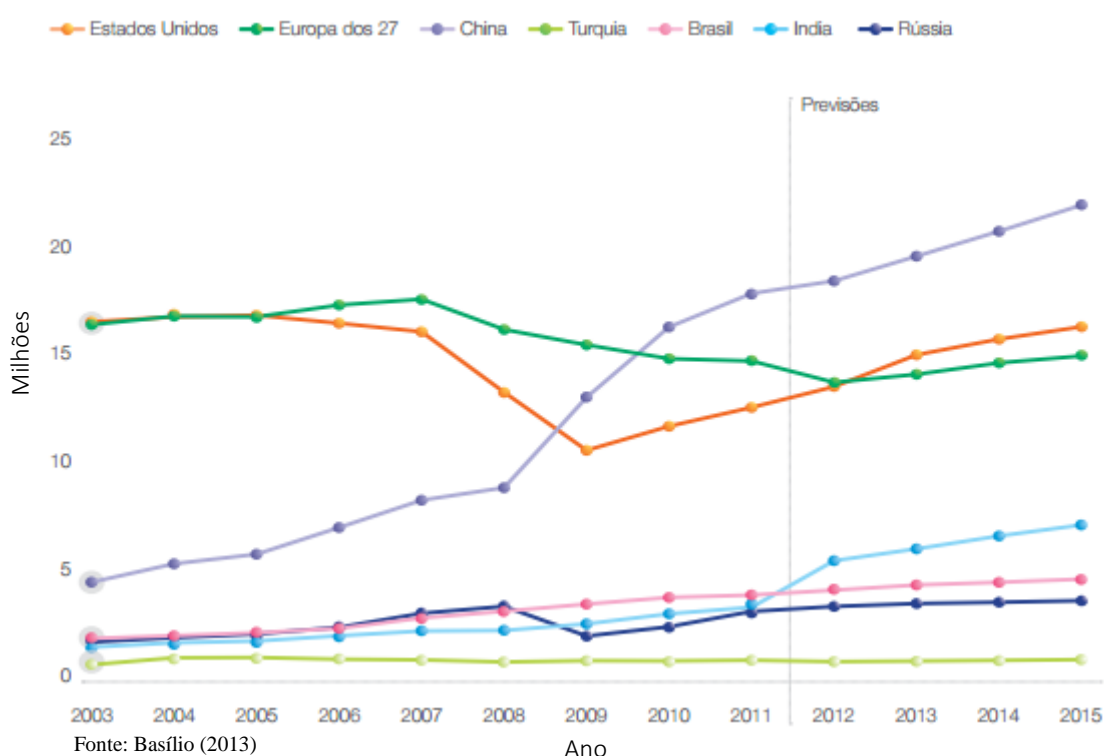


Figura 1 Registo de veículos ligeiros a nível mundial, entre 2003 e 2015

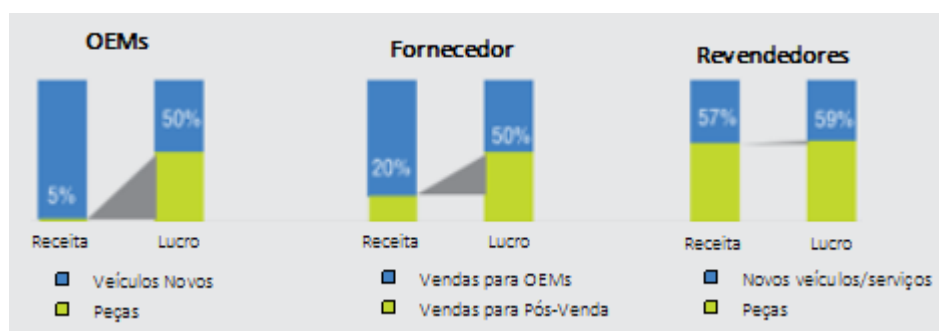
¹ Empresa de estudos e consultoria, pode ser consultado em www.bipe.com

Apesar do mercado pós-venda, onde a venda de peças de substituição para automóveis se insere, representar uma pequena percentagem das receitas dos fabricantes (*OEM - Original Equipment Manufacturer*), o lucro conseguido pode ir até aos 50% (Ver Figura 2). Este mercado também proporciona bons resultados para os fornecedores e revendedores de peças de substituição automóvel. É importante realçar que a diferença entre a receita e lucro do pós-venda é menos expressiva para os revendedores.

Quanto mais desenvolvido o mercado, maior a rentabilidade obtida a partir dos serviços pós-venda, tais como, peças de substituição e seguros (Fernihough & Gyimesi 2008). No início do desenvolvimento dos mercados, o principal foco é a construção do automóvel sendo os principais lucros provenientes da venda dos automóveis novos. Consoante o mercado se vai desenvolvendo, os lucros das actividades a jusante da venda começam a ultrapassar os das vendas dos veículos. Num mercado maduro, mais de 70% do lucro é obtido através das actividades a jusante da venda, sendo a venda de peças de substituição para automóveis o principal factor (Ho & Lo 2005).

O desempenho das empresas nos serviços de pós-venda é um factor importante na escolha do consumidor e de um condutor na fidelidade à marca. A qualidade, disponibilidade e o custo das peças de substituição são componentes críticos para o consumidor. Segundo Fernihough & Gyimesi (2008), o mercado de retalho de peças de substituição é um mercado atractivo devido ao seu potencial de crescimento. As tendências registadas na utilização dos automóveis pelos seus proprietários mostram que haverá uma crescente necessidade de peças e serviços de substituição. Dois factores que contribuem para este crescimento são:

- O aumento da idade média de veículos em circulação - conduz a um aumento da procura de peças mais propensas a desgaste;



Fonte: Fernihough & Gyimesi (2008)

Figura 2 Rentabilidade Pós- Venda

- O crescimento significativo nos mercados emergentes de veículos *low-cost* - necessitam de ser suportados por uma infra-estrutura de peças de substituição.

Sendo o mercado das peças de substituição automóvel, um mercado que evidencia boas oportunidades, é natural que a concorrência seja intensa e vá aumentando ao longo do tempo. A concorrência permite que os consumidores tenham vários canais de distribuição disponíveis para a compra das peças automóveis, como, oficinas de reparação automóvel, fabricantes, retalhistas e lojas de *internet*.

Para além da concorrência resultante dos múltiplos canais de distribuição, tem sido crescente por parte dos revendedores de peças de substituição automóvel a aposta nas “peças de concorrência” de qualidade equivalente às peças OEM, com preços mais reduzidos.

Apesar disso, uma das maiores pressões no sector tem surgido do mercado de peças contrafeitas, entendendo-se por peça contrafeita uma peça que é produzida em imitação de uma peça OEM. Estima-se que 80% das peças contrafeitas sejam provenientes da China e que sejam vendidas entre 50 a 85% abaixo do preço das peças OEM (Commerce 2009). Estima-se ainda, que o mercado de peças contrafeitas representou em 2011, 45 mil milhões de dólares.

Os intervenientes do mercado das peças têm também de combater o crescente “mercado paralelo”. As peças provenientes deste mercado são peças genuínas mas produzidas e comercializadas através de canais ilegais. Por exemplo, uma empresa que fabrica uma determinada peça durante o horário laboral e continua a produzi-la “fora de horas”. Assim, as peças são iguais mas produzidas de forma ilegal, pelo que terão de seguir caminhos ilegais para a sua comercialização. Pelo facto de este processo se realizar “fora de horas”, torna-se complicado quantificar a dimensão e o seu impacto neste mercado.

2.1.1. O MERCADO DE PEÇAS DE SUBSTITUIÇÃO AUTOMÓVEL EM PORTUGAL

Em 2011 a actividade de comércio, manutenção e reparação automóvel em Portugal representou 11,9% de toda a actividade de comércio. Dentro desta, a venda de peças e acessórios para veículos automóveis é a segunda actividade com maior expressão, logo a seguir à venda de veículos automóveis, com um volume de negócios de 1,602 milhões de euros (14,2% do total). Se a análise for feita apenas aos serviços de manutenção e reparação

automóvel, a venda de peças e acessórios assumem uns expressivos 43,8% do volume de negócios total. (INE 2011).

Apesar da queda na venda de viaturas novas (Ver Tabela 2), na maioria dos países da Europa em 2012 e com a tendência a manter-se em 2013 no caso de Portugal, o mercado das peças e acessórios perspectiva um futuro com algumas possibilidades de melhoria.

Segundo dados apresentados por Llera, na 2ª Convenção Ibérica de Reparação Automóvel “Situação do Após Venda” de 2009 promovido pela Associação Nacional das Empresas do Comércio e da Reparação Automóvel (ANECRA), tem-se assistido a um aumento moderado mas contínuo dos quilómetros realizados por ano num automóvel. Este dado, mostra a necessidade de uma maior frequência na visita dos automóveis para revisão programada, o que conseqüentemente leva a um aumento de necessidade de peças de substituição automóvel. O estudo do Cetelem sobre o mercado automóvel na Europa em 2013, apresenta um outro dado que poderá ser um bom indicativo neste mercado, o aumento da vida útil do automóvel, que se tem assistido ao longo da última década, não só em Portugal mas também nos restantes países da Europa. A Figura 3 apresenta a evolução da idade média do parque automóvel em alguns países europeus. Esta tendência do adiamento na troca de automóvel surge como medida do consumidor para contornar a subida de custos de aquisição de

Tabela 2 Mercado Veículos Particulares Novos (Fonte: Basílio 2013)

	2010 (milhares)	2011 (milhares)	2012 (milhares)	Varição 2011/2012	2013 (milhares)	Varição 2012/2013
<i>França</i>	2 252	2 207	1 920	-12,94%	1 975	2,86%
<i>Itália</i>	1 960	1 744	1 430	-15,42%	1 530	3,73%
<i>Espanha</i>	982	810	750	-7,41%	785	4,67%
<i>Portugal</i>	223	153	105	-31,37%	100	-4,78%
<i>Bélgica- Luxemburgo</i>	597	622	535	-13,99%	540	0,93%
<i>Reino Unido</i>	2 031	1 941	1 985	2,27%	2 060	3,78%
<i>Alemanha</i>	2 916	3 174	3 140	-0,44%	3 200	1,27%
<i>Países- Baixos</i>	483	556	530	-10,07%	495	1,00%
<i>Polónia</i>	334	320	294	-8,13%	305	3,74%
<i>República Checa</i>	169	171	174	1,75%	180	3,45%
<i>Eslováquia</i>	64	68	60	-11,76%	62	3,33%
<i>Hungria</i>	45	72	43	-40,69%	50	17,10%
<i>Total 13 Países</i>	12 056	11 838	11 031	-6,82%	11 297	2,41%
<i>Europa 27</i>	13 783	73 638	12 680	-7,01%	12 950	2,13%

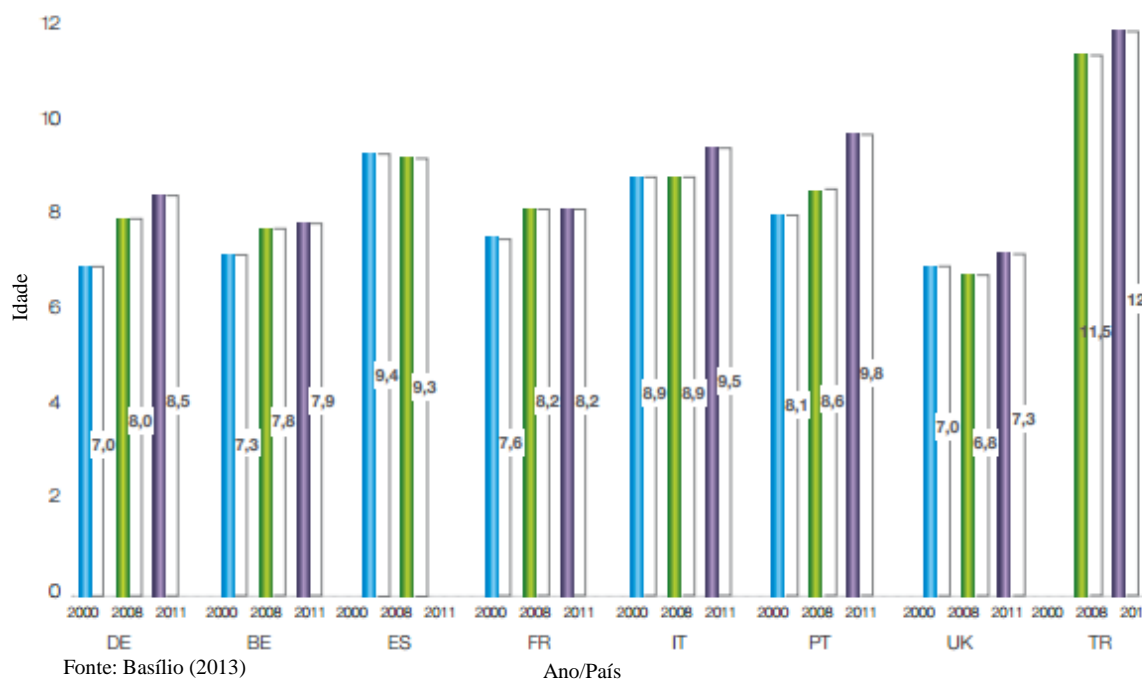


Figura 3 Idade Média do Parque Automóvel em 2000, 2008 e 2011

automóvel. Esta medida é mal vista no mercado de venda de automóveis novos, mas pode ser como um bom indicador no mercado de peças de substituição, uma vez que, automóveis mais antigos necessitam de uma maior frequência de serviços de manutenção.

Outra tendência que tem vindo a crescer, e que deverá ser um nicho de mercado a explorar, é a manutenção “Faça Você Mesmo”. Este modelo, segundo o estudo da Cetelem (2013), representa em Portugal 12% do total de manutenções feitas, devendo ser um mercado à qual as empresas de venda de peças devem direccionar atenções.

Em sentido contrário e acompanhando a situação económica geral, 6,3% dos condutores inquiridos¹ pela ANECRA declararam que estão a tentar poupar em manutenções.

O mercado nacional é formado por cerca de 40 empresas distribuidoras de peças de substituição automóvel, podendo ser destacadas sete que formam o “grupo forte”. Elas são a Gamobar Peças, Auto-Industrial Peças, MCoutinho Peças, Santogal Peças, JAP, Sofrapa e a Caetano Parts.

¹ Inquérito a 1982 condutores em Fevereiro de 2009

A Tabela 3 apresenta as representações das marcas de veículos ligeiros mais comercializados em Portugal em Janeiro/Fevereiro de 2013, pelas principais empresas de retalho de peças de substituição automóvel em Portugal.






2.1.1.1. *GAMOBAR PEÇAS*

A Gamobar foi fundada em 1963 e começou a sua actividade como importadora da marca automóvel Peugeot. Ao longo do tempo foram adicionando novas representações ao seu *portfolio* (Ver Tabela 3) e deixaram de ser retalhistas internos e passaram a fornecer oficinas que não pertenciam ao grupo. Actualmente a actividade de venda de peças de substituição automóvel é suportada por um armazém com sede no Porto (Gamobar Peças Online 2013).

2.1.1.2. *AUTO-INDUSTRIAL PEÇAS*

A Auto-Industrial Peças é uma das empresas pioneiras no sector automóvel em Portugal tendo iniciado a sua actividade em 1920. Actualmente é representante de dezasseis marcas automóveis e reparador autorizado de duas (Ver Tabela 3), a venda de peças de substituição automóvel das marcas que comercializam é feito através de três armazéns localizados no Porto, Coimbra e Leiria (Auto-Industrial Online 2013).

Tabela 3 Representação das Marcas Automóveis (ANECRA 2013)

	Gamobar	Auto-Industrial	MCoutinho	Santogal	JAP	Sofrapa	Caetano Parts
	●	●	●		●	●	●
	●		●	●			●
			●	●	●	●	●
			●	●		●	●
	●	●	●	●	●		●
		●	●	●			●
	●	●	●	●			●
			●	●		●	●
	●	●	●	●		●	●
		●	●	●		●	●
Outras	●	●	●	●	●	●	●

2.1.1.3. *MCOUTINHO PEÇAS*

Iniciou a actividade em 1956, como agente da marca automóvel Ford. Ao longo da sua história o grupo foi crescendo com a agregação de várias concessões de diferentes marcas e com a criação de novas empresas, nomeadamente, centros de colisão, seguradora, centro de peças e *rent-a-car* (MCoutinho 2013).

Actualmente o grupo representa vinte seis marcas automóveis (Ver Tabela 3), com oito concessionárias divididas pelo Norte Interior e Norte Litoral.

A MCoutinho peças surgiu em 1999 e trouxe uma gestão centralizada ao negócio das peças de substituição, que até aí eram geridas pelas concessões. Actualmente, a MCoutinho Peças tem 6400m² de armazém, distribuídos pelo Porto e Lisboa e que contem cerca de 51000 referências em *stock*. A empresa apresenta um volume de negócios de 40 milhões de euros/ano (MCoutinho 2013).

2.1.1.4. *SANTOGAL PEÇAS*

A Santogal surge inicialmente, em 1946, com o nome de Mocar, sendo o importador oficial da marca automóvel Alfa Romeo em Portugal. Apenas em 1991, já com representação de mais marcas é que foi criada a Santogal S.G.P.S. Até 2007 foram sendo agrupadas ao grupo novas marcas de automóveis e motos, sendo actualmente vinte seis as marcas representadas (Ver Tabela 3). Em 2003 surgiu a Santogal Peças, que comercializa peças e acessórios de todas as marcas representadas (Santogal Peças Online 2013).

A Santogal tem a sua actividade concentrada essencialmente na Grande Lisboa, tendo contudo em 2007 alargado o seu território para Madrid.

2.1.1.5. *JAP*

O Grupo JAP é um grupo centenário, que iniciou a sua actividade em 1904 no Marco de Canaveses. Actualmente o grupo actua principalmente na zona Norte, abrangendo nove áreas de negócio, entre elas uma central de peças de origem e multimarca (Ver Tabela 3). (JAP Online 2013).

2.1.1.6. SOFRAPA

Fundada em 1958 sob a sigla R.O.M.A a Sofrapa dedicava-se inicialmente de forma exclusiva à representação de componentes e acessórios dos automóveis franceses. Com a evolução do sector, a Sofrapa diversificou a sua oferta, criando unidades de negócio diferentes. Actualmente, a empresa apresenta uma gama de produtos que engloba todo o tipo de componentes auto (Ver Tabela 3) e conta com uma frota de 17 ligeiros de mercadorias e parceria com várias transportadoras, garantindo a entrega dos produtos em todos os pontos do país, a partir do armazém com sede em Odivelas (Sofrapa Online 2013).

2.1.1.7. CAETANO PARTS

A Caetano Parts é uma empresa que se integra no Grupo Salvador Caetano, S.G.P.S, S.A, responsável por mais de 6500 postos de trabalho e que teve como ponto de partida a criação de empresa Martins, Caetano e Irmão em 1946. (Grupo Salvador Caetano Online 2013)

O Grupo é constituído por três unidades de negócio principais, organizadas por três *sub-holdings*¹(Grupo Salvador Caetano Online 2013):

- Toyota Caetano Portugal S.A, que agrega o negócio industrial e da representação automóvel Toyota;
- Salvador Caetano Auto (SGPS) S.A, referente ao retalho automóvel multimarca no mercado ibérico.
- Salvador Caetano.Com (SGPS) S.A, que constitui o negócio na área das tecnologias de informação.

Para além destas o grupo detém outras empresas na indústria de autocarros, componentes automóveis, entre outras áreas de actuação. Ao longo dos anos o Grupo assistiu a uma expansão constante ao nível de território de actuação, começando por se expandir no território nacional, e avançando posteriormente para mercados internacionais, a Figura 4 apresenta os países onde o Grupo opera.

¹ Em português Sociedade Gestora de Participações Sociais (SGPS) é uma forma de sociedade criada com o objectivo de administrar um grupo de empresas.

Para Gibson-odgers et al. (2008) o serviço ao cliente consiste no processo de satisfação do cliente, relativamente a um produto ou serviço, de forma a que o cliente considere que as suas necessidades foram atendidas, com eficiência e compreensão. Quando as expectativas não são atendidas, o serviço é considerado inaceitável e provoca uma insatisfação no cliente. Caso contrário, quando as expectativas são preenchidas, o serviço torna-se satisfatório (Fitzsimmons & Fitzsimmons 2006).

Na perspectiva de Christopher (2005) os produtos só tem valor até estarem nas mãos dos clientes dentro do tempo e no local pretendido. Assim, o serviço ao cliente pode ser definido como uma provisão consistente no âmbito do tempo e do espaço.

As empresas não se devem limitar apenas ao produto e às suas características, devem também oferecer toda uma gama de serviços que acrescentem valor ao produto. Christopher (2005) cita Theodore Levitt¹ ao afirmar que “as pessoas não compram produtos, elas compram benefícios”, com esta afirmação Levitt quer passar a mensagem que a totalidade da oferta da empresa leva ao aumento da percepção de valor que o cliente tem sobre o produto. Um exemplo que suporta esta afirmação é a de um produto acabado no armazém e o mesmo produto na posse do cliente. O produto é o mesmo em relação às suas características, contudo o produto nas mãos do cliente tem mais valor, pois neste caso o serviço de distribuição já acrescentou valor ao produto (Christopher 2005).

A Figura 5 **Erro! A origem da referência não foi encontrada.** demonstra o acréscimo de valor do produto, através do incremento de serviços. O círculo central representa o valor do produto tal como está no armazém, o anel exterior representa o valor que pode ser acrescentado com o incremento de alguns serviços.

As componentes logísticas do nível de serviço podem ser classificadas de diferentes formas. Segundo Rushton et al. (2010) e Christopher (2005), estes podem ser vistas como “elementos transaccionais” e divididas em 3 categorias, que reflectem a natureza e o tempo do serviço. Podem ser vistos como elementos:

- (i) pré-transaccionais, que antecedem a transacção entre vendedor-cliente, como a definição da política serviço ao cliente e organização da estrutura;

¹ Economista Alemão e um “pensador” do Marketing.

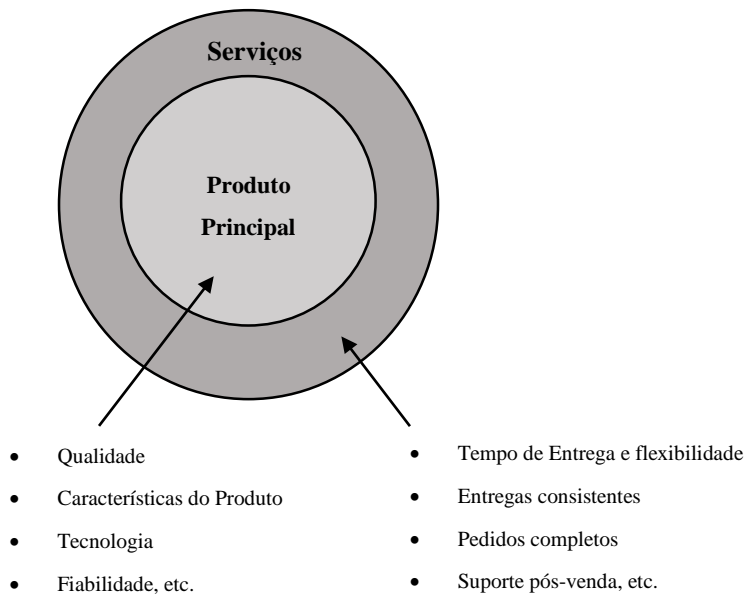


Figura 5 Uso dos serviços para o aumento do valor do produto (Adaptado de Christopher 2005)

(ii) elementos pós-transaccionados, posteriores à transacção, como o pós-venda, a disponibilidade de peças, a garantia e os procedimentos de reclamações;

(iii) elementos em transacção, que estão directamente relacionados com o momento da transacção, como disponibilidade de *stock*, tempo de entrega e condição do bem.

Os elementos de serviço podem também ser classificados por dimensões multifuncionais. A intenção é avaliar os diferentes parâmetros do serviço ao cliente, em todas as actividades da empresa. As quatro principais dimensões multifuncionais são:

(i) tempo – normalmente o tempo de ciclo de uma ordem;

(ii) confiança – garantia de entregas a tempo e sem danos;

(iii) comunicação – facilidade de fazer pedidos;

(iv) flexibilidade – capacidade de reconhecer e responder a novas necessidades dos clientes.

Existem diferentes elementos para a avaliação do serviço ao cliente, e a sua relevância e importância varia com o produto, empresa e mercado em que se está inserido.

Alguns estudos demonstram que níveis de serviço baixos, levam os clientes a mudar facilmente de fornecedor para um que preencha os seus requisitos. Devido a esse facto, é

necessário estabelecer um plano que determine uma política de serviço que preencha os requisitos do cliente (Rushton et al. 2010). Por outro lado, existe consenso de que um alto nível de serviço leva ao aumento da satisfação do cliente (Shahin 2011). Esta relação pode levar a relações mais longas entre o cliente e o prestador do serviço (Etgar & Fuchs 2009).

Gibson-odgers et al. (2008) refere que as empresas devem planear as suas acções e comportamentos tendo em vista a satisfação do cliente, e que devem ser capazes de se adaptar às alterações das necessidades, desejos e expectativas dos clientes. Nesse sentido Fernihough & Gyimesi (2008) afirmam que, as empresas do mercado automóvel pós-venda devem implementar uma estratégia que satisfaça os seguintes objectivos: melhor compreensão dos clientes, optimização da cadeia de aprovisionamento e relações com parceiros mais fortes. A ameaça contínua da concorrência sobre os lucros, leva a que estas empresas necessitem de adoptar medidas para gerarem valor.

As contínuas mudanças na procura e exigência dos clientes leva a que seja necessária uma aproximação sistemática e uma revisão contínua das várias operações logísticas, pois estas requerem um planeamento dinâmico (Rushton et al. 2010).

A Figura 6 representa um exemplo de um plano cíclico de planeamento e controlo.

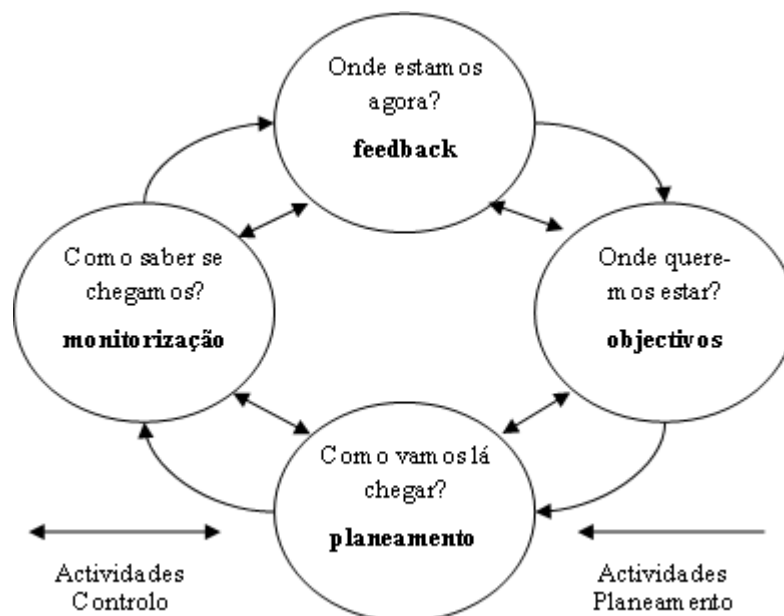


Figura 6 Plano cíclico de planeamento e monitorização (Adaptado de Rushton et al. 2010)

O ciclo inicia-se com o “*Onde estamos agora?*”, em que o objectivo é obter um plano geral da situação actual. A segunda etapa determina quais as metas a atingir. Nesta fase deve-se ter em conta aspectos como a satisfação do cliente. O terceiro passo do ciclo corresponde ao momento do planeamento e da definição da estratégia, para atingir os objectivos definidos anteriormente. Finalmente, é necessário uma monitorização e um controlo dos procedimentos para medir a eficácia¹ das operações em comparação com o plano traçado. O ciclo é fechado após esta etapa e o processo recomeça novamente. Isto permite manter a natureza dinâmica e indicia a contínua revisão do planeamento.

2.2.1. MEDIR O NÍVEL DE SERVIÇO

Existem diferentes formas de medir o nível de serviço prestado, e devem ser adaptadas não só ao sector da empresa, mas também aos requisitos dos seus clientes. O mais importante, independentemente da medida que é usada, é que mostre o fundamental do requerido pelo cliente. Esta situação pode nem sempre ser óbvia. Tendo como um exemplo o cumprimento de uma encomenda. É possível medir o seu cumprimento de diferentes formas, como por exemplo (Rushton et al. 2010):

- O número de ordens completamente satisfeitas num determinado período, e.g. 18 de 20 (90%);
- O número de itens enviados de uma determinada encomenda, e.g. 75 de 80 requisitados (94%);
- Valor de uma encomenda satisfeito, e.g. 750€ de 900€ total (83%).

Qualquer uma das medidas listadas acima podem ser usadas, não existe nenhuma certa, nem nenhuma errada. Deve ser usada a medida que melhor se adequa à análise a ser feita. Uma boa prática poderá ser usar uma combinação das várias medidas, contudo, o problema alcança uma maior complexidade à medida que mais itens de análise vão sendo acrescentados.

Outras análises podem ser feitas, por exemplo, uma prática recorrente das transportadoras parcelares é realizar a análise de performance baseada no tempo de operação (Bhatnagar &

¹ Do dicionário: “capacidade de cumprir os objectivos pretendidos”

Viswanathan 2000). Uma das medidas efectuadas é o tempo decorrido entre a recolha da encomenda até à sua entrega final.

Christopher (2005) e Rushton et al. (2010) defendem que uma definição apropriada dos objectivos de serviço é mais fácil com a adopção do conceito “ordens perfeitas”. Estas medidas englobam todos os atributos que completam uma ordem e satisfazem plenamente o cliente. Um conceito como este remete uma definição de ordem perfeita específica para cada cliente, contudo é possível agrupar clientes em segmentos, e identificar os requisitos a preencher para este segmento. Os principais factores que contribuem para as “ordens perfeitas” são:

- Encomendas enviadas com as quantidades certas;
- Encomendas entregues exactamente na data e hora pretendida pelo cliente;
- Encomendas sem problemas no envio (extravios, recusas, danos);
- Encomendas com documentação de envio completa e correcta.

A percentagem de “ordens perfeitas” pode ser calculada através da expressão 1:

$$\text{Ordens Perfeitas} = \frac{\text{nr}^\circ \text{ ordens perfeitas}}{\text{total de ordens}} \times 100\% \quad (1)$$

2.3. O OUTSOURCING DO TRANSPORTE

A logística tem grande importância nas empresas para assegurarem vantagens competitivas, como a redução do custo das operações e a possibilidade de incrementar valor ao produto final (Sanchez 2013). Assim, a logística requer serviços especializados, pelo que os prestadores de serviços logísticos são muitas vezes contractados, sendo na maioria dos casos contratados os serviços de transporte e de armazenamento. (Lieb & Bentz 2004; Lieb 2008; Hannon 2008).

O transporte pode ser considerado como uma actividade que não acrescenta valor ao produto, mas que é indispensável para corresponder às expectativas do cliente (Czerska 2009). É também a actividade económica mais importante no contexto da actividade logística. Sem sistemas de transporte bem desenvolvidos, a logística pode não ser uma actividade com vantagens no panorama da operação. Pelo contrário, um bom sistema de transporte em actividades logísticas pode proporcionar mais eficiência logística, redução dos custos de operação e aumento da qualidade do serviço (Tseng 2005). Tseng (2005) baseia-se na estimativa efectuada por Chang (1998) na *Air Transportation Association* para apresentar o

custo das principais actividades logísticas. Analisando a Figura 7 é possível comprovar que a maior fatia diz respeito aos gastos com transporte (29%), o equivalente a quase um terço dos custos logísticos totais. Assim sendo, o transporte tem um papel importante em toda a actividade logística pelo que deve ser analisado continuamente para futura melhoria e redução dos custos de operação.

Tanto na Europa como na América do Norte, o recurso a transportadoras tem sofrido um aumento nos últimos anos, em detrimento do transporte providenciado pela própria empresa (Rushton et al. 2010). Langley et al. (2003) afirma que as actividades com maior *outsourcing* por parte das empresas são o armazenamento e transporte. Com a globalização dos negócios e conseqüente concorrência, tem aumentado a necessidade das empresas fazerem a entrega dos seus produtos adaptadas às necessidades dos clientes.

2.3.1. O QUE É O OUTSOURCING?

O *outsourcing* é um acordo na qual uma empresa contrata outra empresa, para realizar tarefas que até aí eram realizadas internamente. O *outsourcing*, não só adquire produtos e serviços a fontes externas à empresa, como também, tem como objectivo partilhar responsabilidades e riscos (Mccarthy & Anagnostou 2004; Zwolinska 2012). A partir dos anos 90 a importância económica e estratégica do *outsourcing* nas empresas aumentou, evoluindo da utilização limitada a tarefas que não acrescentavam valor ao produto, como a segurança e a limpeza, para apresentar grande impacto em tarefas chave e que acrescentam valor, como as

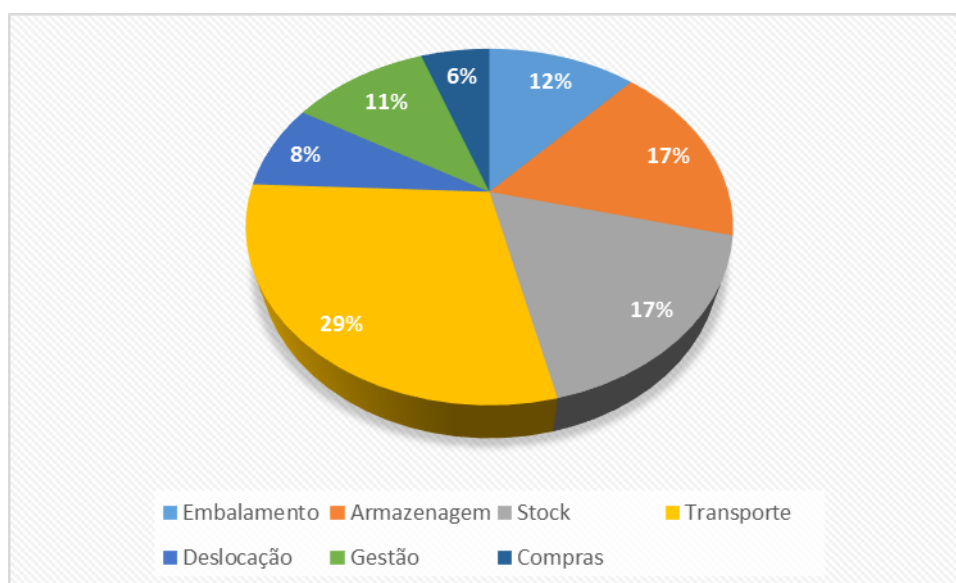


Figura 7 Rácios dos custos das actividades logísticas (Adaptado de Tseng 2005)

tecnologias de informação e a logística, sendo também utilizada já em actividades principais de produção, como o *design* ou processamento do produto (Mccarthy & Anagnostou 2004).

Goles & Chin (2002) definem o *outsourcing* como “uma ligação em curso entre empresa subcontratada e empresa que contrata, orientada para a parceria de longo prazo e com reconhecimento mútuo de que os benefícios alcançados por uma empresa são, em parte, dependentes da outra empresa”. Já Gay & Essinger (2002) consideram que esta prática é uma estratégia de gestão na qual a empresa celebra parcerias, para as actividades secundárias, com empresas especializadas e que prestem um serviço eficiente.

Como as definições anteriores indicam, as relações de *outsourcing*, embora diferindo na forma como são caracterizadas, são sempre relações de troca (Grover et al. 1996; Lacity & Willcocks 1998).

2.3.2. MOTIVAÇÕES E RISCOS DO OUTSOURCING

Uma das razões mais importantes para *outsourcing* do transporte é a capacidade que os fornecedores de serviços têm de fornecer aos clientes um serviço especializado e experiente, que sendo facultado pela própria empresa seria dispendioso ou difícil de atingir (Razzaque & Sheng 1998; Jharkharia & Shankar 2007).

Existem alguns estudos sobre as motivações e benefícios de optar pelo *outsourcing* de actividades. Abraham & Taylor (1993) identificam 3 razões chave: (i) redução de custos, (ii) transferência da incerteza na procura para a empresa contractada e (iii) acesso a serviços especializados, que podem não ser possíveis de obter dentro da empresa. Para N. Kakabadse & A. Kakabadse (2000) as principais razões são: (i) económica, (ii) qualidade e (iii) inovação.

Nos estudos realizados ao longo do tempo sobre as motivações para optar pelo *outsourcing*, (Kenyon & Meixell 2011) acabam por ter conclusões semelhantes, que incluem a redução de custos (Aimi 2007; Jiang et al. 2006; Lau & Zhang 2006), o melhoramento da qualidade do produto (Bardhan et al. 2007), o melhoramento da flexibilidade (Lau & Zhang 2006), o aumento da área de actuação (Skjoett-Larsen 2000) ou para obter um rápido aumento de capacidade (Linder 2004; Mason et al. 2002).

Wilding (2004) compilou as razões para o *outsourcing* dos serviços logísticos na indústria de bens de consumo europeia, a partir de outros estudos (Szymankiewicz 1994; Boyson et al.

Tabela 4 Razões para Outsourcing (Adaptado de Wilding 2004)

<i>Razões</i>	<i>P-E Internatioal (1994): Industria Bens Consumo</i>	<i>Boyson (1999): Indústria em geral</i>	<i>Fernie (1999): retalhistas</i>	<i>Laarhoven (2000): maioria das indústrias</i>	<i>Penske Logistics (1999): algumas</i>	<i>Ranking Global</i>
<i>Custo ou receita</i>	3.Redução custos	1.Redução custos	5.Tendência para eficiência	1.Redução custos	1.Redução custos	40
<i>Serviço</i>	2.Melhoramento do serviço		4.Serviço Especializado	2.Melhoria serviço	3.Melhoria nível serviço	27
<i>Flexibilidade de Operações</i>	1.Flexível		1.Sistema mais flexível	3.Flexibilidade estratégica		26
<i>Foco Negócio</i>	5.Não é a actividade principal	2.Outsourcing de actividade não principal		4.Foco na actividade principal		17
<i>Utilização de activos e eficiência</i>			2.Concentração de recursos financeiros no negócio principal		2.Aumento eficiência	16
<i>Mudança de gestão</i>		4.Re-design da cadeia de abastecimento		5.Mudança na implementação	4.Melhoria na distribuição	16
<i>Especialização Transportadora</i>			3.Explorar especialização do contractado			6
<i>Constrangimento</i>		3.Área de outsourcing				6
<i>Investimento</i>	4.Evitar custos					5

1999; Fernie 1999; Van Laarhoven et al. 2000; Logistics 1999), tal como é possível observar na Tabela 4. Em cada coluna é apresentado o *ranking* atribuído pelo autor do estudo das razões para se optar pelo *outsourcing*, ou seja, quando está presente o número 1 antes da razão significa que aquela é a razão principal para a *outsourcing* naquele estudo, quando tem o número 2 significa que é a segunda mais influente, etc. No final são contabilizadas as cotações e obtido um *ranking* total que engloba todas as avaliações, este é feito atribuindo 10 pontos quando a razão é dada como a principal, 8 pontos quando é a segunda, 6 para a

terceira, 5 para a quarta e 4 para a quinta. Estes pontos são somados em linha originando então o *ranking* global. A pontuação total possível na Tabela 4 são 50 pontos, no caso dos 5 estudos terem encontrado a mesma razão principal para o *outsourcing*. Neste caso, conclui-se que a redução de custos (40 pontos), melhoria do nível serviço (27), aumento da flexibilidade (26), foco no negócio principal (17), melhoria da utilização de recursos (16) e mudança de gestão (16) são as razões mais comuns para proceder ao *outsourcing* de serviços logísticos. Estes resultados são coerentes, com outros autores (Harry L. Sink & C. John Langley 1997; Razzaque & Sheng 1998; Gay & Essinger 2002; Jharkharia & Shankar 2007), que apresentam as mesmas razões como os principais factores que levam a externalizar actividades.

Existem também problemas, identificados pelos autores, associados ao *outsourcing*, como por exemplo a falta de controlo na actividade externalizada, redução da flexibilidade e aumento dos custos. A perda de flexibilidade está relacionada com os acordos contratuais de longa duração. O *outsourcing* é realizado por muitas organizações com o intuito de controlar ou reduzir custos, no entanto, há alguma evidência de que isso pode não acontecer, sendo que em alguns casos os custos acabam por aumentar (Mccarthy & Anagnostou 2004).

As razões pelo qual uma empresa opta pelo *outsourcing* de algumas actividades, nem sempre levam aos resultados pretendidos. Os vários autores chegaram em geral às mesmas conclusões, que estão sintetizadas na Tabela 5, e que apresenta os benefícios encontrados do *outsourcing*, mas também os problemas encontrados em algumas parcerias.

Tabela 5 Benefícios e problemas do *outsourcing* de actividades

Benefícios:
Redução dos custos com as actividades externalizadas
Aumento de flexibilidade
Aumento do nível de serviço nas actividades externalizadas
Diminuição do número de trabalhadores
Maior disponibilidade para se focar na actividade principal
Maior especialização nas actividades logísticas
Aumento das tecnologias de informação
Problemas:
Diminuição no controlo das actividades externalizadas
O tempo e esforço disponibilizado nestas actividades não diminuí
Redução de custos não notada
Qualidade pretendida do serviço não atingida
Nível de serviço não corresponde ao pretendido
Aumento de reclamações dos clientes

A existência de sistemas de avaliação de performance bem definidas permitem ao expedidor avaliar as diferenças entre o que está a comprar e o seu valor relativo, avaliar o fornecedor de transporte em relação às expectativas criadas e como ferramenta para estabelecer metas de serviço em futuras utilizações (Brooks 2000). Existem trabalhos sobre avaliação da performance a partir da perspectiva do cliente (e.g. Dumond 1991), no entanto, estes trabalhos não se focam na avaliação de performance de serviços de transporte, e tão pouco na análise prática da implementação da monitorização das actividades.

2.3.3. CRITÉRIOS DE SELECÇÃO DE SERVIÇOS DE TRANSPORTE

No caso do sector das peças automóveis, o transporte tende a ser expresso e parcelar, com possibilidade de entrega mais de uma vez ao dia ou até ao dia útil seguinte e à porta do destinatário. As características do transporte expresso são (Tseng 2005): (i) serviço porta-a-porta; (ii) eficiência; (iii) rastreabilidade; (iv) JIT; (v) tipos de envios diversificados.

Com a competição baseada no tempo e os resultados provenientes da implementação de estratégias *Juste-in-time (JIT)*, têm surgido novas parcerias “expedidor-transportadora” (Brooks 2000).

O JIT surgiu no Japão no início dos anos 50, como uma nova abordagem na indústria e tem sido aplicado com sucesso, nomeadamente na indústria automóvel. Esta abordagem tem implicações significativas para a distribuição e logística. Para Rushton et al. (2010) o conceito geral de JIT é proporcionar um sistema produtivo que elimine todas as actividades que não agreguem valor ao produto final, nem permitam o fluxo contínuo de material. Sucintamente, o objectivo do JIT é eliminar os elementos dispendiosos e que geram desperdício num processo produtivo.

Vários autores (Crum & Allen 1991; Kleinsorge et al. 1991; Gibson et al. 1993; Brooks 1995) referem que tem havido uma mudança na selecção de transportadoras, do critério preço de envio de uma parcela, para alianças e serviços de transporte negociados e que se adequem ao pretendido. Lieb & Millen (1988,1990) observaram que a implementação de estratégias JIT leva à mudança tanto em critérios pelos quais as transportadoras são seleccionadas, como também ao número de transportadoras usadas. Tendo um leque diversificado de prestadoras de serviço de transporte, os expedidores podem optar pela transportadora que apresenta melhor serviço.

Várias pesquisas (Razzaque & Sheng 1998; Lynch 2000; Andersson & Norrman 2002) apontam para uma tarefa difícil, no momento de perceber quais os critérios mais apropriados para a escolha do fornecedor de serviços. A complexidade desta tarefa aumenta, consoante for maior o número de critérios a avaliar (Meade & Sarkis 1998). Rushton et al. (2010) referencia um estudo realizado por Szymankiewicz (1996) em Inglaterra, para identificar os factores que são tidos em conta na selecção de uma transportadora (Ver Figura 8), sendo os três mais importantes o serviço prestado, os recursos humanos e o custo.

A opção, por uma transportadora, nem sempre é uma tarefa simples, até porque existem vários tipos de serviços de transporte, tais como (Rushton et al. 2010):

- *Distribuição dedicada*: é um tipo de serviço subcontratado e em que as empresas facultam ao cliente todas as suas necessidades de distribuição de forma exclusiva;
- *Distribuição multi-utilizador*: este tipo de distribuição é similar à distribuição dedicada, em que a principal diferença é que o serviço é facultado a um pequeno grupo de clientes. Uma das características deste serviço é que, idealmente, os clientes são todos os expedidores para destinos comuns;
- *Distribuição especializada*: esta distribuição é utilizada para o armazenamento e movimentação de produtos que requerem cuidados especiais no seu manuseamento;

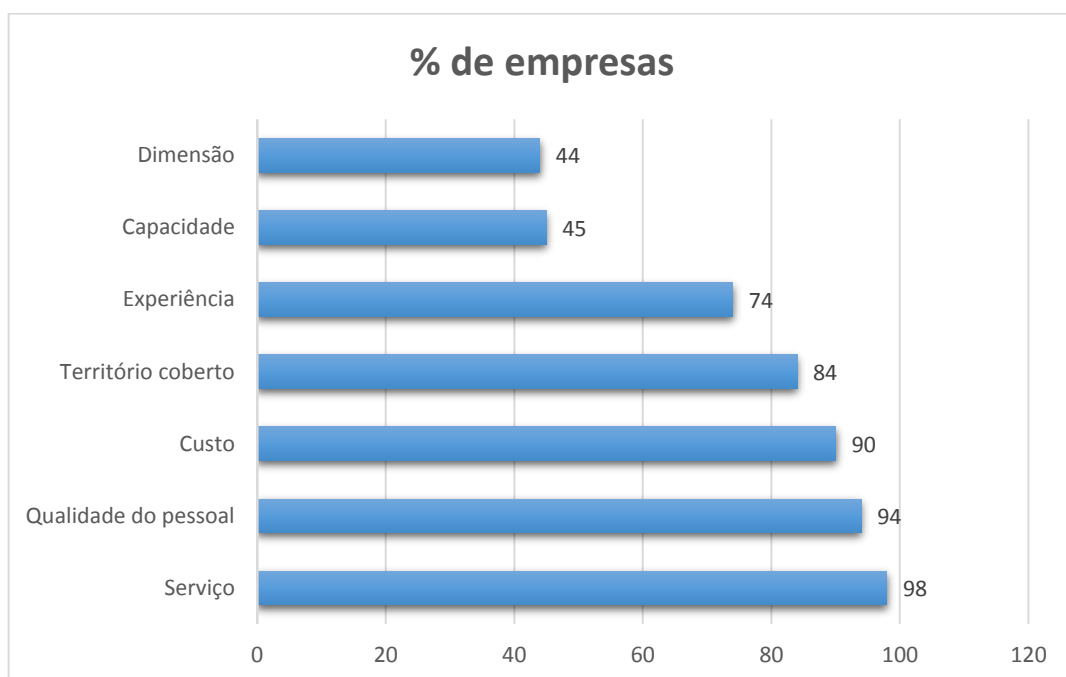


Figura 8 Factores de escolha transportadora (Adaptado de Rushton et al. 2010)

- *Distribuição regional multi-cliente*: esta distribuição chega até todos os clientes e é adaptada à generalidade dos produtos. Normalmente é fornecida por uma transportadora “generalista” e que concentra a sua actividade numa área regional específica;
- *Distribuição nacional multi-cliente*: esta é muito semelhante à distribuição multi-cliente regional, mas em que a área de actuação é mais ampla (nível nacional) e inclui operações de transbordo;
- *Distribuição em trânsito*: nesta distribuição não estão envolvidas operações de armazenagem, a transportadora apenas recolhe o pacote e efectua a sua entrega;
- *Distribuição ocasional*: várias empresas usam este tipo de serviço para ocasiões excepcionais ou como apoio ao próprio serviço de entrega. Alguns dos motivos para esta utilização são: apoio aos picos de procura sazonal, apoiar picos de procura semanais, para produtos fora do padrão, para envios fora da área de actuação, etc.

Os critérios pelo qual o expedidor escolhe determinada transportadora têm sido estudados por vários autores. A maioria dos estudos demonstrou a dificuldade de ligar os atributos de escolha da transportadora, com a percepção tida pelo expedidor da performance dessa mesma transportadora (Brooks 2000). De acordo com o modelo Aaker & Day (1980) uma transportadora deve ser vista como tendo os atributos considerados importantes pelo expedidor e como sendo a que melhor performance apresenta para esses atributos.

2.3.4. MONITORIZAÇÃO DA PERFORMANCE

No estabelecimento de uma parceria entre fornecedor de serviços e empresa cliente devem ser tidos em consideração os seguintes factores: (Gay & Essinger 2002):

- Clarificar e acordar objectivos/metapas;
- Medir a performance;
- Acordar penalizações por não obter a performance mínima;
- Monitorização frequente;
- Partilha de *feedback* e implementação de melhorias.

Nos contactos com terceiros, a introdução de práticas inovadoras e de monitorização, podem ser benéficas para melhorar a eficiência e comparar a performance de serviços contratados (Ferne 1999). A monitorização da performance é essencial para um controlo eficiente em

qualquer empresa e estão dependentes de sistemas de informação eficientes (Rushton et al. 2010).

Segundo Brooks (2000) a vontade de monitorizar e acompanhar a performance de uma transportadora, surge a partir da necessidade de melhorar os resultados na decisão de qual o serviço de transporte mais adequado a contratar, sendo este um pequeno mas vital procedimento a tomar na implementação do modelo logístico (Fawcett & Clinton 1996). Chow et al. (1994) considera que a avaliação da performance da transportadora é uma ferramenta usada pelas transportadoras e pelos responsáveis logísticos para adquirir, assegurar e manter a vantagem competitiva.

A performance pode ser definida segundo Kearney (1985), como o rácio do *output* actual em relação ao *output* definido como *standard*. Este *output standard* pode ser uma referência do sector ou uma meta que a empresa defina. Mentzer & Konrad (1991) refere-se à medição da performance como uma análise da eficácia e eficiência na realização de uma determinada tarefa.

A satisfação do cliente é referenciada como sendo uma medida de performance importante (Brooks 2000). Contudo do ponto de vista de Rhea & Shrock (1987) a avaliação da performance abrange mais do que avaliar apenas a satisfação do cliente.

Assim sendo, uma avaliação global da performance deve incluir tanto uma avaliação externa da eficácia (alcançar ou exceder as expectativas dos clientes), como uma avaliação interna da eficiência (utilização recursos) para alcançar uma rentabilidade a longo prazo.

Mentzer & Konrad (1991) explicam que o paradigma eficiência-eficácia, equilibra a relação entre o cliente (expedidor) e o fornecedor (transportadora), fornecendo a transportadora com as indicações necessárias para o estabelecimento das metas de prestação do serviço e evidenciando a necessidade de melhorias contínuas no serviço.

No estudo realizado por Brooks (2000), em 1999, à performance de 455 transportadoras Canadianas e Americanas, este concluiu que 38% das mesmas considera que a monitorização deve ser feita por parte do expedidor. Conclui também que o número de empresas que utilizam medidas quantificáveis como, desvio em minutos, horas ou dias do tempo de entrega, percentagem de rejeitados, taxa de erro de facturação ou número de reclamações, foi menor do que o previsto.

Os parâmetros de avaliação da performance de transportadoras mais frequentemente utilizados são (Zwolinska 2012):

- Escala de tempo;
- Preço do transporte em relação aos concorrentes;
- Experiência e profissionalismo;
- Condições dos veículos;
- Danos durante carga e descarga;
- Documentação correcta;
- Nível global dos serviços prestados.

Deve-se ter em consideração que os parâmetros apresentados não estão incluídos num grupo fechado. Devem ser acrescentados ou limitados em coerência com a avaliação que se pretende efectuar. O parâmetro mais comum e que mais vezes é referenciado, quando se trata da avaliação de performance, é o baseado no tempo, por exemplo, tempo de carga, tempo em trânsito, etc. (Michlowicz 2012).

Os parâmetros listados acima podem ser divididos em dois grupos. O primeiro refere-se aos parâmetros directamente relacionados com quem fornece o serviço de transporte, neste grupo estão incluídos os parâmetros: velocidade e eficiência, experiência e profissionalismo e um nível de serviço global que corresponda ao pretendido. O segundo grupo diz respeito aos parâmetros que estão directamente relacionados com quem organiza o transporte, do qual se pode referenciar: o correcto preenchimento dos documentos, o cuidado com a carga a transportar, evitando danos e defeitos e a manutenção que assegure o correcto funcionamento do veículo.

A avaliação da qualidade deve ser feita regularmente, com vista a fazer uma monitorização efectiva e obtendo a possibilidade de implementar melhorias. A avaliação é feita pela soma dos parâmetros que a influenciam e que estão sujeitos à quantidade de ocorrências referentes a um determinado espaço temporal, nomeadamente:

- Quantidade de cargas e descargas efectuadas dentro do tempo (ou o número de atrasos);
- O número de transportes efectuados correctamente;
- Número de ocorrência de estragos ou extravios;
- Número de documentos preenchidos em desacordo com o pretendido.

Quando se pretende obter o mais alto nível de serviço, deve-se ter em conta o grau de satisfação do cliente (Zwolinska 2012).

A Tabela 6 apresenta os resultados obtidos por Brooks (2000) em que a medida mais utilizada para avaliação da performance é a recolha e/ou entrega no momento certo, o que vai de encontro ao referenciado anteriormente. Danos, documentação correcta, equipamento correcto e em boa condição são também dos aspectos mais mencionados. Já a Tabela 7, apresenta os resultados obtidos no estudo efectuado por Wilding (2004) a 330 transportadoras de bens de consumo de toda a Europa, para avaliar os elementos de análise mais utilizados na avaliação da performance de uma transportadora. O mesmo referencia que 78% das empresas considera ter avaliações formais da performance.

Wilding (2004) constatou que na maioria das parcerias era utilizado algum método de análise da performance e que um grande número dos contractos (74%) não foram renovados. Isto mostra o descontentamento existente em relação às empresas de serviços logísticos. Este descontentamento é significativo e relevante, pois apenas 10% das parcerias estudadas renovaram os contractos sem problemas.

A Tabela 8 evidencia as principais razões para a não renovação da parceria, sendo que as principais razões estão relacionadas com o serviço (68%) e o custo (52%). Todas as empresas pretendem atingir um nível de serviço de 100%, contudo é aceitável sempre que os valores se situam entre os 95% e os 99%. Normalmente, nas parcerias estabelecidas entre empresas é salvaguardada a hipótese do limite inferior ser ultrapassado, com o pagamento de penalizações (Zwolinska 2012).

**Tabela 6 Elementos de análise mais utilizados para avaliar performance da transportadora
(Adaptado de Brooks 2000)**

<i>Elemento de Análise</i>	<i>Expedidor</i>	<i>Transportadora</i>
<i>Recolha momento certo</i>	80%	60%
<i>Entrega momento certo</i>	80%	49%
<i>Facturação correcta</i>	43%	54%
<i>Perda e danos</i>	67%	48%
<i>Documentação correcta</i>	-	41%
<i>Equipamento apropriado</i>	-	35%
<i>Equipamento boas condições</i>	47%	31%

Tabela 7 Elementos de análise para avaliar performance (Adaptado de Wilding 2004)

<i>Grupo medidas de Performance</i>	<i>Medidas de Performance</i>	<i>%utilização</i>
<i>Entrega</i>	Tempo de entrega, pontualidade, falhas	46
<i>Custo</i>	Custo por envio, controlo do custo, bónus por falha, visibilidade total dos custos	32
<i>Qualidade</i>	Nível serviço, qualidade das ordens de envio, qualidade geral	22
<i>Inventario</i>	Exactidão no <i>stock</i> disponível, quantidade de stock, rapidez de escoamento	20
<i>Exactidão de picking</i>	Qualidade de <i>picking</i> , acerto <i>picking</i>	18
<i>Responsabilidade e flexibilidade</i>	Resposta a questões, administração, serviço cliente	14
<i>Erros e controlo de estragos</i>	Percentagem de perdas, rácios de erros	14
<i>Tempo de entrega</i>	Velocidade de entrega	6
<i>Carga e descarga</i>	Pontualidade e tempo de carga/descarga	4
<i>Documentação</i>	Documentação correcta	4
<i>Variação entre performance actual e expectativa</i>	Comparação da performance e expectativas	4
<i>Outros</i>	Temperatura dos produtos, utilização veículo, recursos humanos e satisfação cliente	10

Tabela 8 Razões para não renovar contractos com serviços logísticos (Adaptado de Wilding 2004)

<i>Razões</i>	<i>Frequência (%)</i>
<i>Serviço e qualidade: fraca performance, fraco serviço, problemas serviço cliente</i>	68
<i>Custos</i>	52
<i>Problemas comunicação: fraca fluidez de informação, dificuldade em chegar a acordo, cultura, falta de ajuda em momentos de crise</i>	10
<i>Mudança de operação: dificuldade de adaptação à mudança, falta de inovação</i>	8
<i>Decisões estratégicas: centralização da distribuição, decisão de actuação própria</i>	4
<i>Fraca gestão</i>	4
<i>Oferta da concorrência</i>	2
<i>Instabilidade financeira</i>	2
<i>Sem adição de valor</i>	2

2.4. SOLUÇÕES PARA SISTEMAS TRANSPORTE DE ENVIOS PARCELARES

Neste subcapítulo são apresentados três exemplos de soluções de gestão de transporte de envios parcelares existentes no mercado. As soluções encontradas são destinadas ao mercado norte-americano e funcionam como agregadoras de serviços de informação disponibilizadas pelas próprias transportadoras. As soluções existentes realçam, a falta de ferramentas no mercado que possibilitem a interacção com prestadoras de serviços de transporte que não apresentem serviços de informação desenvolvidos.

2.4.1. CLOUD LOGISTICS

A Cloud Logistics (C. L. Online 2013) apresenta uma solução de *Transportation Management System* (TMS) (Ver Figura 9) dedicada ao expedidor. Esta ferramenta baseada na *web* permite ser moldada consoante as necessidades e objectivos do utilizador. Carregando a sua lista de destinatários, alternativas de transporte e outras informações, permite fazer o acompanhamento e verificação de alertas de todos os despachos, tanto através do serviço *web*, como também através de uma aplicação móvel.

Este serviço tem uma curiosidade inerente, pelo facto de ter sido desenvolvido como uma “rede social”, pois é possível criar grupos de trabalho, entre expedidor, transportador, receptor ou outro interveniente, onde podem ser deixadas mensagens, documentos, vídeos ou imagens.

Sendo este um serviço *web*, pode ser acedido a partir de qualquer lugar tendo uma ligação à internet e um *browser* para acesso. Contudo, este serviço apresenta alguma rigidez em alguns pontos, pois só é compatível com transportadoras que estejam munidas de sistemas de informação e que disponibilizem elas mesmo todas as informações sobre as operações de distribuição. Não foi possível apurar que tipo de análise a plataforma executa e se sugere

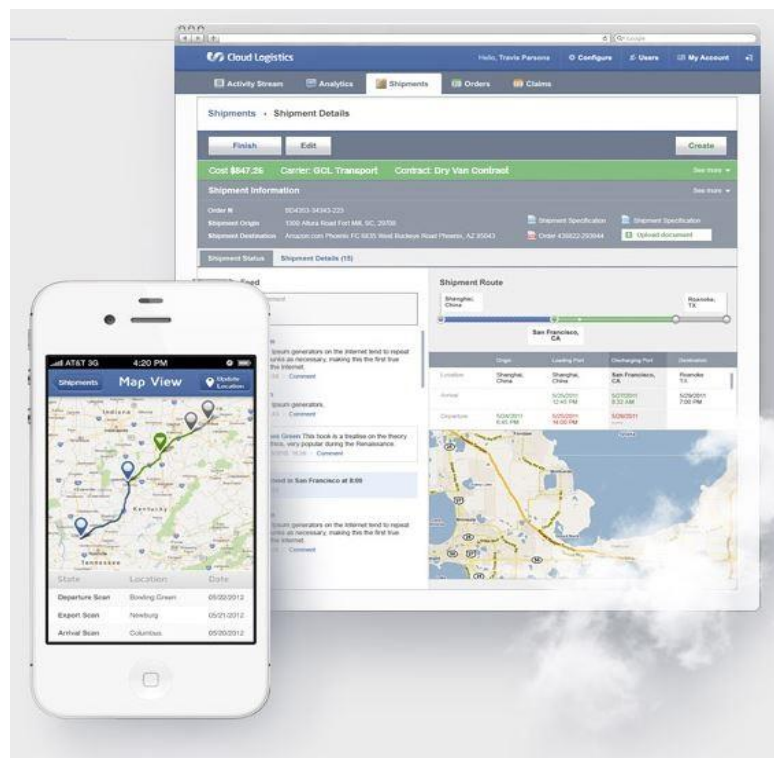


Figura 9 Layout plataforma da Cloud Logistics

algum tipo de proposta sobre a transportadora mais adequada para determinado despacho. Este serviço só está disponível nos Estados Unidos da América.

2.4.2. SMART FREIGHT WARE

SmartTMS™ (S. F. W. Online 2013) é um TMS flexível e baseado em sistemas *cloud*, construído para expedidores gerirem e controlar os seus envios, com uma interface *web* (Ver Figura 10) simples e amigável para utilizador.

A plataforma permite:

- Planear despachos – permite otimizar os envios por custo ou tempo de entrega;
- Processamento de despacho – com possibilidade de gerar etiquetas de envio;
- Rastreamento do despacho;
- Base de dados de histórico.

A plataforma tem algumas limitações, por um lado só consegue apresentar informação das transportadoras que disponibilizem essa informação via *web*, por outro lado não permite a criação de etiquetas *standard* para todos os despachos, apenas para as transportadoras que já disponibilizam esse serviço.

Em suma, esta plataforma funciona como um agregador de informação e de tratamento de dados que facilita a visualização de informação de diferentes fontes num só local. Tal como o serviço *Cloud Logistics*, apenas está disponível nos Estados Unidos da América.

Shipment No.	Shipper, City, State Zip, Shipper Ref #	Shipped-to, City, State Zip, Ship-to Ref #	Pieces & Weight Total Charges	Pickup Carrier & Pro# Date Picked Up	# Labels
SM00035445 NEW	Joe Shipper Lawrenceville, GA, 30043	Chicago Warehouse Bensenville, IL, 60106	3 pcs. 1500 lbs. \$145.45	USF Holland 79708080	<input type="text"/>
SM00035231 NEW	Georgia Shipper Lawrenceville, GA, 30043 09808089	Joe Warehouse Itasca, IL, 60143	1 pcs. 1500 lbs. \$208.11	Milan Express 435345435	<input type="text"/>
SM00034957 DELIVERED	Text ship Lawrenceville, GA, 30043	Consignee Itasca, IL, 60143	40 pcs. 45000 lbs. \$450.00	PF Truck Co 12/7/2010	<input type="text"/>
SM00034951 DELIVERED	jo Lawrenceville, GA, 30043	robert Itasca, IL, 60143	24 pcs. 30000 lbs. \$425.00	PF Truck Co 12/6/2010	<input type="text"/>
SM00034940 NEW	Joe Shipper Lawrenceville, GA, 30043 34566	Joe Warehouse Itasca, IL, 60143	1 pcs. 1500 lbs. \$208.11	Milan Express 3452345435	<input type="text"/>
SM00034685 NEW	Joe Shipper Lawrenceville, GA, 30043	Joe Warehouse Itasca, IL, 60143	2 pcs. 1000 lbs. \$234.41	USF Holland	<input type="text"/>
SM00034550	Joe Shipper	Joe Warehouse	1 pcs. 1500 lbs.	Milan Express	<input type="text"/>

Figura 10 Layout plataforma SmartTMS™

2.4.3. *PRECISION*

A Precision (Precision Online 2013) apresenta um vasto leque de soluções que visam reduzir os custos e simplificar as actividades logísticas. A solução apresentada é dedicada ao expedidor e integra várias componentes que podem ou não fazer parte da plataforma. O TMS permite aos utilizadores ver tarifas, rotas, criar etiquetas, monitorizar níveis de serviço das várias transportadoras e para diferentes tipos de transporte. Este serviço permite obter em tempo real os custos e os tempos em trânsito das várias transportadoras, possibilitando também a consolidação de encomendas, quando for mais vantajoso.

Para além do TMS a Precision complementa a plataforma (Ver Figura 11) com uma aplicação de tratamento e gestão de dados, apresentando gráficos, relatórios e diversas informações acerca dos transportes efectuados e com uma unidade para toda a gestão de transportes internacionais.

Actualmente o TMS da Precision é dedicado ao mercado dos Estados Unidos da América, contudo já contem na sua base de dados algumas transportadoras Europeias. Tal como as soluções anteriores todos os dados recolhidos são obtidos directamente da transportadora, não sendo disponibilizado nenhuma alternativa para transportadoras que não tenham tecnologias de informação implementadas.

The screenshot shows a web-based form for shipping management. At the top, it indicates the user's location: 'You are here: Home - Desktop Shipping - Origin/Destination'. Below this is a breadcrumb trail and a 'Shipment Wizard' dropdown menu currently set to 'Shipment'. The form is organized into several columns and sections:

- Ship To:** Includes checkboxes for 'Mailing List' and 'New Address'. Below this is a 'Deliver To' section with a 'Show Address Details' link and input fields for Name, Contact, Address, City, State, Postal Code, Country (pre-filled with 'US'), Telephone, and Email.
- Shipment Details:** Contains input fields for Promised Date, Promised Time, Business Unit, Cost Center, and Project Code.
- Ship From:** Features a 'Deliver From' dropdown menu (pre-filled with 'A9901') and an 'Overwrite Address' checkbox.
- Shipper:** Includes a 'Same as Ship From' checkbox and a 'Shipper' dropdown menu (pre-filled with 'A9901') with an 'Overwrite Address' checkbox.

At the bottom left, there is a small icon, and at the bottom right, there is a 'Hex' link.

Figura 11 Layout Plataforma Precision

3. *DESCRIÇÃO DO PROBLEMA* *- CASO CAETANO PARTS*

Neste capítulo é feito um enquadramento do problema na Caetano Parts, abordado o processo actual de tratamento de uma encomenda e explanada a sugestão de alteração no processo.

3.1. ENQUADRAMENTO

A Caetano Parts é uma empresa do mercado automóvel que centra as suas actividades na venda e distribuição de peças de substituição automóvel. As operações são suportadas por dois armazéns localizados no Porto e Setúbal. O processo de venda de uma encomenda pode ser dividida em duas fases. A primeira fase engloba as operações realizadas dentro das instalações da empresa como o processamento e preparação da encomenda e a segunda fase refere-se à distribuição e entrega da encomenda pela transportadora ao cliente final.

Ao contrário das etapas realizadas nas instalações da empresa, a distribuição das encomendas é externalizada. O compromisso estabelecido com a maioria das transportadoras é a da encomenda ser entregue nas instalações do cliente no dia seguinte à recolha, independentemente da região de entrega. A principal transportadora (identificada como Transportadora 1 na Figura 12), é responsável pela distribuição do maior volume de encomendas, e efectua recolhas várias vezes ao dia e faz a entrega nesse mesmo dia. Esta

transportadora apresenta uma distribuição dedicada e surgiu de um *spin-off*¹ da Caetano Parts, como tal o serviço é efectuada tem por base uma avença mensal, independentemente do número de encomendas transportadas. As restantes transportadoras baseiam a sua actividade num valor unitário por parcela, o valor cobrado por cada uma respeita um mapa de escalões definidos pela dimensão da parcela.

A Figura 12 apresenta o mapa de actuação das diferentes transportadoras (limites aproximados).

Do armazém do Porto, a região do distrito do Porto e a zona litoral até Aveiro é coberta pela transportadora 1. As regiões de Braga, Viana do Castelo, Bragança e Vila Real são cobertas pela transportadora 2. A transportadora 4 faz uma distribuição para os concelhos de Viseu, Leiria e Coimbra. Sendo o restante país coberto pela transportadora 3.

No que diz respeito aos envios efectuados a partir do armazém de Setúbal, as transportadoras 1 e 3 são as mesmas que actuam a partir do Porto, sendo que a transportadora 1, faz a

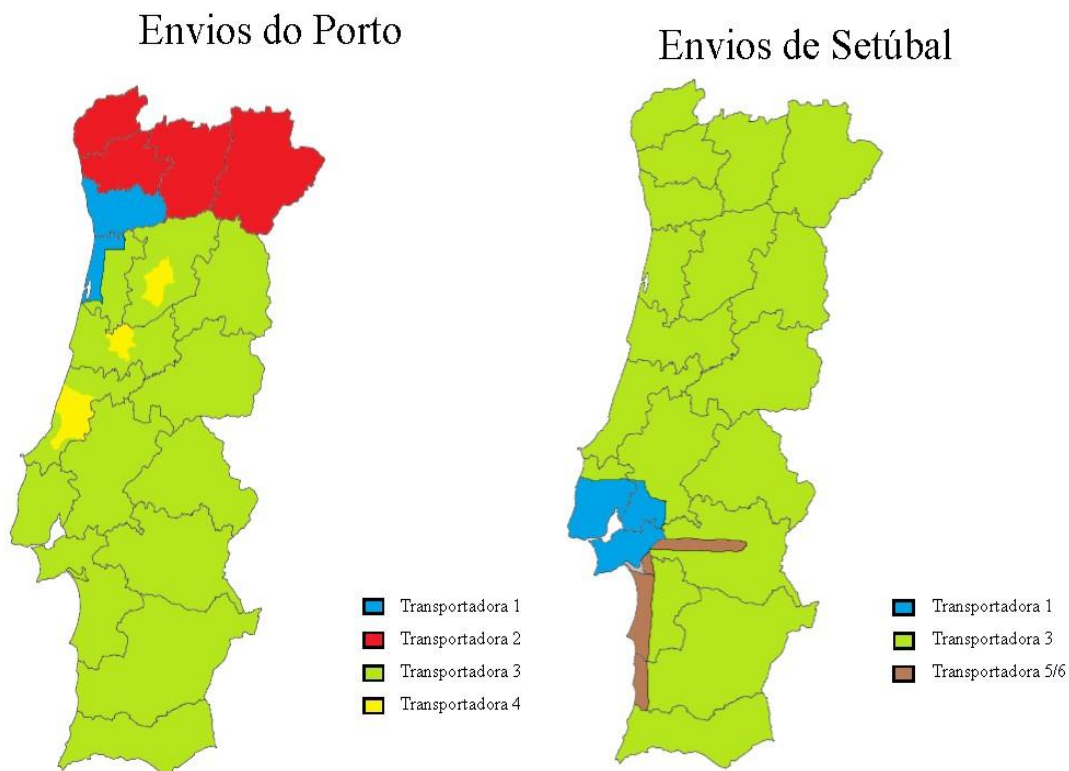


Figura 12 Mapa de actuação, por zona, das transportadoras contratadas pela Caetano Parts

¹ Divisão de uma actividade da empresa, formando uma empresa independente

cobertura do distrito de Lisboa até Torres Vedras, um pouco do distrito de Santarém e a zona norte do distrito de Setúbal. A transportadora 5/6 efectua entregas desde Setúbal até Évora e a zona litoral desde o sul do Rio Sado até Odemira (distrito de Beja). Para o resto do país as entregas são feitas pela transportadora 3.

As diferentes transportadoras apresentam sistemas de informação distintos, na maioria dos casos pouco desenvolvidos. Exceptuando a transportadora com cobertura nacional, que apresenta um portal *web* que é actualizado sempre que exista uma alteração no estado do envio, as restantes não permitem obter um acompanhamento “em directo” da encomenda após a sua saída do armazém, nem tão pouco disponibilizam informações necessárias para avaliar o seu nível de serviço e conseqüentemente o nível de serviço que a Caetano Parts pratica com os seus clientes. Estas transportadoras apresentam um sistema de operação básico, fazendo a recolha dos dados manualmente, com suporte a documentos de papel. Nestes casos, a recolha dos dados, evidencia-se problemática, por diferentes motivos. Os problemas mais comuns são o não preenchimento dos campos necessários no momento da recolha e da entrega da encomenda e a não devolução dos documentos ou a entrega fora do tempo útil para processamento dos dados.

De forma a mitigar estes problemas surgiu a necessidade de desenvolver uma solução capaz de monitorizar os envios nas diferentes transportadoras e posterior tratamento dos dados para quantificação do nível de serviço da Caetano Parts.

3.2. PROCESSO ACTUAL – TRATAMENTO DE ENCOMENDA

A Figura 13 apresenta as etapas do processo de tratamento de encomenda desde o momento de colocação de uma nova encomenda, até ao seu despacho.

O processo inicia-se com a nota de encomenda realizada pelo cliente, que pode ser efectuada através de telefone, correio electrónico, *fax* ou portal *online*. O operador abre um processo *work in progress (wip)* e verifica a disponibilidade das peças, se alguma ou todas as peças estiverem em falta é feito um pedido ao fornecedor de peças automóveis. Após o pedido das peças em falta ser realizado, as mesmas são recepcionadas no armazém, no prazo de um a dois dias úteis.

No momento da recepção da encomenda é feita uma verificação do estado das peças, por um operador responsável, que caso exista algum defeito reporta a situação ao fornecedor de

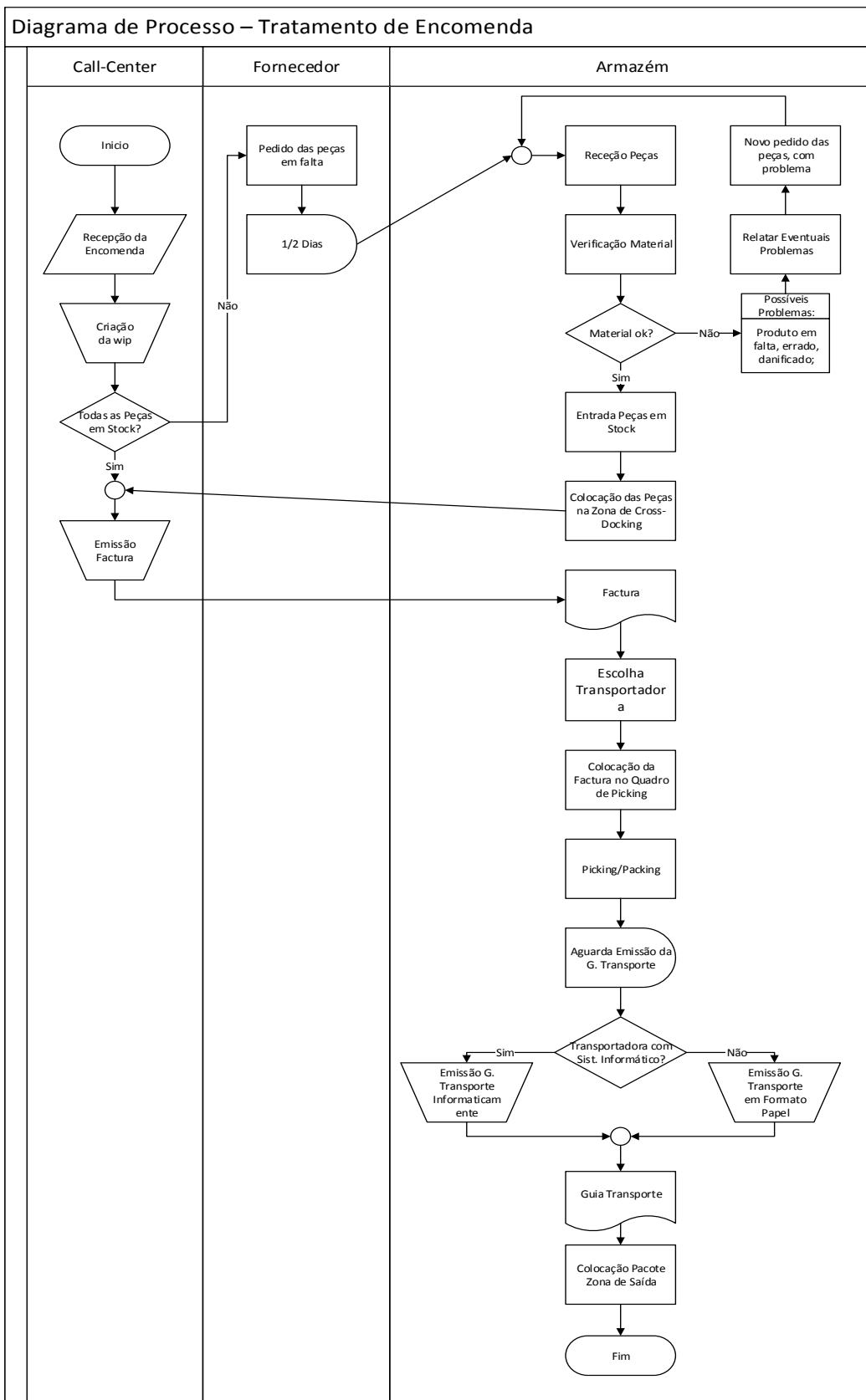


Figura 13 Processo Actual de Tratamento de uma Encomenda

peças automóveis e o processo de encomenda reinicia-se. Se as peças se encontrarem em

boas condições e nas quantidades correctas, são encaminhadas para a zona de *cross-docking*¹.

Na abertura da *wip*, caso os itens pedidos estejam todos disponíveis ou no momento em que são recepcionadas os itens em falta, o operador do *call-center* responsável pela encomenda, dá ordem de facturação, terminando aqui as operações realizadas no *call-center*.

No momento da emissão da factura, as etapas que restam são todas realizadas pelos operadores do armazém. Servindo as facturas como documento informativo dos dados do cliente, o primeiro processo é proceder à selecção da transportadora/rota, mediante o destino da encomenda. Este processo é rígido, uma vez que está estabelecida a alocação das transportadoras para todas as regiões. Assim, o operador verifica o destino da encomenda e coloca a factura no quadro de *picking* (Ver Figura 14) na posição correspondente à transportadora e à hora de recolha.

O *picking* é realizado por ordem da hora de recolha, uma vez que as diferentes transportadoras fazem recolhas a horas diferentes, assim, são processados primeiro os pedidos recolhidos mais cedo. Segue-se o *packing* e os pacotes são colocados numa zona em espera, até que outro operador proceda ao preenchimento das guias de transporte. Neste



Figura 14 Quadro definição rotas (esquerda) e local de recolha dos pacotes por rota (direita)

¹ Estratégia de distribuição que envolve o movimento de material directamente do local de recepção para o local de distribuição, com o mínimo de tempo decorrido entre eles (Apte & Viswanathan 2000).

ponto, o preenchimento pode ser realizado através de métodos distintos. Caso a transportadora que vai realizar a distribuição possua *software* próprio para o efeito, o operador realiza o seu preenchimento informaticamente, é impressa uma etiqueta com as informações do envio e é colada no exterior da embalagem. Nas restantes transportadoras, as guias de transporte são preenchidas em formato papel, e a factura serve como informação do destino da encomenda. A encomenda pronta a despachar é colocada na zona de saída (Ver Figura 14) correspondente e a partir daí só é possível acompanhar o pacote enviado contactando directamente com o motorista responsável.

Actualmente, os acordos estabelecidos entre a Caetano Parts e as transportadoras definem as seguintes cláusulas:

- Região de actuação da transportadora;
- Prazo previsto para entrega em cada zona;
- Valor a pagar por cada despacho, sendo normalmente taxado mediante o volume da parcela;
- Restrições nos componentes a transportar (*eg*: produtos explosivos, vidros, etc.);
- Penalizações no caso de perda ou danos de algum componente, no caso de culpa da transportadora são restituídos os valores dos componentes implicados.

Nem sempre é realizado um acompanhamento efectivo da actuação das transportadoras, salvo quando existe uma queixa por parte do cliente.

3.3. PROCESSO PROPOSTO – TRATAMENTO DE ENCOMENDA

A proposta de melhoria (Ver Figura 15) apenas altera os procedimentos a partir da emissão da factura. Todo o processo anterior a esta etapa, mantem-se inalterada.

No instante em que é dada ordem de emissão da factura, são desencadeadas diferentes acções:

- Introdução de um novo registo na base de dados com informação da encomenda, permitindo a consulta do estado e o seu acompanhamento;
- Introdução da encomenda na lista de *picking*;
- Selecção da transportadora;
- Criação da etiqueta de transporte e impressão da factura.

A partir deste momento o operador só necessita distribuir os documentos pelo quadro de *picking* e proceder ao *picking* e *packing* pela ordem da hora de recolha de cada transportadora e colocar o pacote pronto para entrega na zona de saída.

É disponibilizado um portal para o responsável de operações que permite consultar, criar e alterar os registos que constam na base de dados, localizar encomendas e consultar dados sobre as operações diárias, semanais, mensais e anuais.

As informações relativas às encomendas associadas a cada par transportadora/motorista é mostrada numa aplicação *Android* disponível para o dispositivo móvel do motorista e onde são listadas todas as entregas ainda por realizar.

Esta implementação pretende obter ganhos em vários pontos:

- (i) redução das operações realizadas pelos operários no preenchimento dos documentos necessários, tal como a possível redução de erros no preenchimento dos mesmos;
- (ii) promover a constante melhoria por parte das transportadoras;
- (iii) possibilidade de acompanhar a encomenda, mesmo depois da saída da mesma do armazém;
- (iv) acesso a estatísticas e dados que proporcionem um melhor entendimento do estado actual e de possíveis melhorias, mas também que aumente o poder negocial perante as transportadoras.

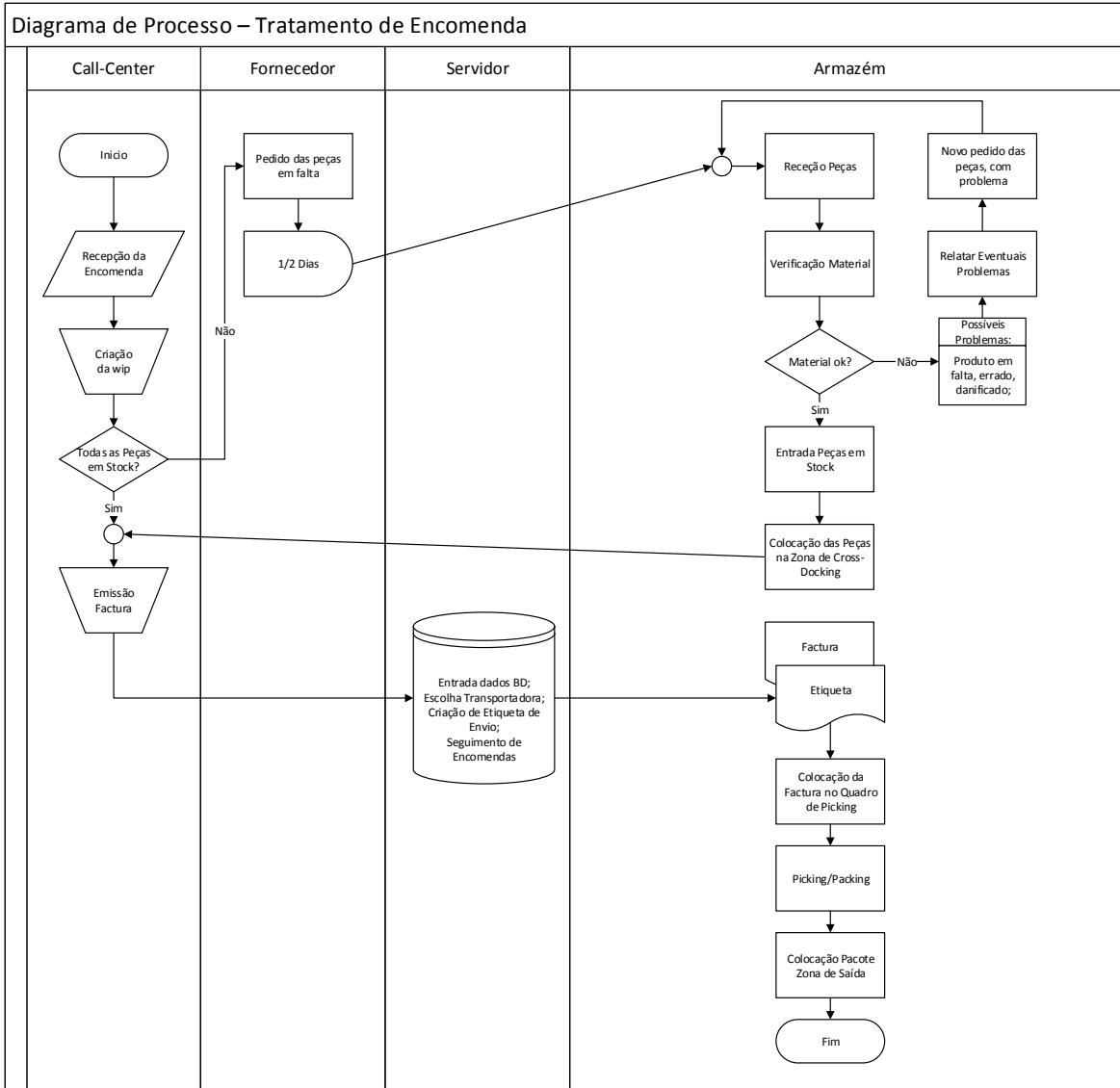


Figura 15 Proposta de Alteração do Processo de Tratamento de uma Encomenda

4. *DESENVOLVIMENTO DA FERRAMENTA*

Neste capítulo descreve-se o desenvolvimento do sistema apresenta-se a sua arquitectura, os desenvolvimentos realizados ao nível de front-end, como a aplicação Android e o portal web, e ao sistema de back-end. É também demonstrado o funcionamento do sistema.

4.1. ARQUITECTURA

O sistema engloba diversos processos que acompanham a evolução de uma encomenda, desde que é realizada até à sua entrega. Envolvendo no processo, a recolha, armazenamento, processamento e apresentação da informação ao utilizador. A arquitectura está dividida em dois grandes blocos: *front-end* e *back-end*. No *front-end* estão incluídas as aplicações *Android* e *web* de interface com o utilizador, que permitem também recolha, armazenamento local e apresentação dos dados. O *back-end* é constituído pelo serviço de recolha, armazenamento e processamento dos dados.

O armazenamento e processamento será alocado num servidor na rede, permitindo o acesso fácil à informação a partir de qualquer lugar e também assegurando a independência das características dos equipamentos que efectuem esse acesso.

Os dados que compõem a base de dados são recolhidos, tanto de forma automática como de forma manual, consoante o momento da actualização da tabela. Numa primeira fase, a recolha dos dados é realizada em *background* e corresponde à introdução de uma nova entrada na base de dados, relativo a uma nova encomenda. Posteriormente, as actualizações feitas aos registos presentes na base de dados são realizados através da aplicação *web* ou do dispositivo móvel, mediante a operação a realizar.

A Figura 16 apresenta a arquitectura do sistema. O *back-end* é composto pelo módulo de armazenamento, que agrega a informação sobre os clientes, encomendas, transportadoras, motoristas e utilizadores. No espaço de *back-end* são efectuados os processamentos da recolha de dados iniciais, criação das páginas *web* e também operações secundárias, como a ordem de impressão das facturas. A troca de dados entre o dispositivo móvel e a base de dados é assegurado através de um serviço *web* do tipo *Restful* (Ver Anexo C), esta interacção baseia-se num sistema de pedido/resposta.

O *front-end* engloba duas aplicações: aplicação móvel e a aplicação *web*.

A aplicação *web* permite aos utilizadores, aceder às funções do sistema a partir de um terminal com acesso à internet. Este aplicativo engloba funcionalidades de gestão de dados, consulta do estado das encomendas, listagem de *picking* em andamento e *dashboard* de análise de dados.

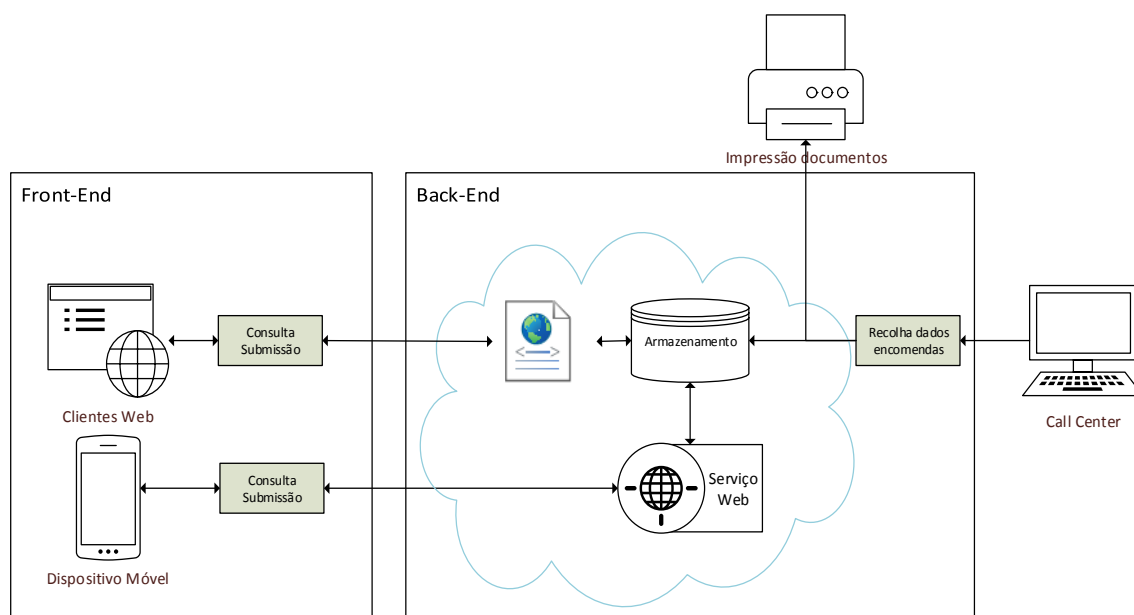


Figura 16 Arquitectura do Sistema

A aplicação móvel fornece ao sistema a capacidade de actualização de dados via internet, em tempo real e em qualquer lugar. Permitindo ao utilizador (motorista) visualizar as encomendas a seu cargo, tal como dar ordem de sucesso/insucesso no momento da entrega. Em simultâneo serão recolhidas e armazenadas as *Geographic Information System (GIS)*, que podem ser utilizadas posteriormente para validação da entrega no local correcto. As informações relativas às entregas serão armazenadas localmente, no dispositivo móvel, minimizando o volume de dados trocados com o servidor. Este aplicativo será utilizado pelos motoristas, que terão de estar equipados com um dispositivo com *software Android*, ligação à internet e sensor GPS (*Global Positioning System*).

4.2. FUNCIONALIDADES DA FERRAMENTA

O sistema apresenta diferentes funcionalidades: (i) gestão da informação armazenada na base de dados dos utilizadores, clientes, envios, transportadoras e motoristas; (ii) visualização do estado de uma operação (despacho); (iii) recolha de dados das encomendas em *background* e posterior actualização através da aplicação móvel; (iv) processamento da informação, incluindo a geração da lista de *picking* e a *dashboard* de análise de operações.

4.2.1. GESTÃO DE DADOS

O armazenamento dos dados é realizado com recurso a um sistema de armazenamento *open-source*, como forma de mitigar os custos associados à manutenção. As duas principais soluções existentes no mercado, *MySQL* e *PostgreSQL* foram comparadas tendo em conta características como, o armazenamento dos dados, a integridade dos dados, o tipo de dados suportados, a interface, a possibilidade de *backups* e a importação de dados. A avaliação pode ser consultada no Anexo C. Das avaliações realizadas conclui-se que o *PostgreSQL* comparativamente ao *MySQL*, apresenta características equiparáveis ou superiores, especialmente no campo da fiabilidade e da recuperação de erros, preenchendo os requisitos Atomicidade, Consistência, Isolamento, Durabilidade (ACID) e tendo métodos de recuperação de erros bem desenvolvidos. Acrescem a estes argumentos, dois factores importantes como, a capacidade de a base de dados (BD) suportar a maioria dos dados e estar preparada com interfaces que permitem a comunicação através de várias linguagens.

Neste subcapítulo, são apresentadas as estruturas criadas na base de dados, como é o caso dos objectos: utilizadores, clientes, transportadoras, motoristas e envio e também as interacções possíveis entre o utilizador e a base de dados usando a aplicação *web*.

4.2.1.1. UTILIZADORES

Os utilizadores com acesso das funcionalidades da aplicação *web* são caracterizados na base de dados na tabela *utilizadores*, que se encontra estruturada com o mapa de dados apresentado na Tabela 9.

Os utilizadores são criados e geridos pelo administrador que por sua vez é também um utilizador. Os utilizadores são diferenciados entre si por níveis.

As funções possíveis na gestão dos dados dos utilizadores são:

- **Criação:** a criação de um novo registo de utilizador é realizada com sucesso com o preenchimento correcto da estrutura da Tabela 9. Ao submeter a operação, o sistema verifica se o *username* e nome estão dentro dos limites de caracteres permitidos e a senha é encriptada em SHA-1 (*Secure Hash Algorithm*). Se os requisitos anteriores forem cumpridos, o serviço verifica se o *username* já está em uso, e conclui a operação ou não, mediante a resposta. A coluna nível define as permissões de acesso às diferentes páginas do portal, sendo que é possível optar por: (i) permissões de administrador, em que o utilizador tem permissões para percorrer todas as páginas do portal e efectuar qualquer operação; (ii) permissões intermédias, mais direccionadas aos operadores de *picking*, permitindo o acesso à lista de *picking* e à reimpressão de etiquetas; (iii) permissões básicas, em que os utilizadores apenas têm acesso à zona de localização de encomendas, sendo equiparado a um visitante.
- **Edição:** em relação a um utilizador pode ser alterada a *password* e o nível de acesso.
- **Eliminação:** esta função elimina toda a informação armazenada na base de dados sobre o utilizador.
- **Gestão de Sessões:** o acesso às diferentes funcionalidades requer uma autenticação por parte do utilizador, originando uma sessão. A autenticação do utilizador é realizada através de um formulário de *login* e na submissão é verificado se o utilizador existe e

Tabela 9 Estrutura de dados da tabela *utilizadores*

<i>Coluna</i>	<i>Tipo</i>
<i>username</i>	character
<i>password</i>	character
<i>nome</i>	character
<i>nivel</i>	integer

se a *password* está correcta, em caso afirmativo são armazenados como dados da sessão o nome de utilizador e o respectivo nível de acesso.

4.2.1.2. *CLIENTES*

Os dados relativos aos clientes são representados na base de dados na tabela *clientes*, sendo a sua gestão realizada pelo administrador. A estrutura dos dados armazenados está presente na Tabela 10.

As funções possíveis na gestão dos dados dos clientes são:

- **Criação:** a criação de um novo registo de cliente é realizada com sucesso com o preenchimento dos campos da Tabela 10, exceptuando o campo *contacto* que não é de preenchimento obrigatório. O sistema no momento da criação verifica se o cliente já existe através da duplicação do número contribuinte.
- **Edição:** na tabela *clientes* apenas as colunas *id* e *contribuinte* não são editáveis.
- **Eliminação:** esta função marca o registo como inactivo, de forma a manter a integridade dos dados.

4.2.1.3. *TRANSPORTADORAS*

As transportadoras são caracterizadas na base de dados pela tabela *transportadoras* e apresentam a estrutura presente na Tabela 11.

As funções possíveis na gestão dos dados das transportadoras são:

- **Criação:** a criação de um novo registo de uma transportadora é realizado com o preenchimento obrigatório das colunas *id*, *nome* e *nacional*. A coluna *id* identifica a transportadora e a rota, sendo representadas por um número superior a três algarismos, em que os dois últimos algarismos identificam a rota (Ver Figura 17). O *id* é o

Tabela 10 Estrutura de dados da tabela *clientes*

<i>Coluna</i>	<i>Tipo</i>
<i>id</i>	character
<i>nome</i>	character
<i>contribuinte</i>	character
<i>morada</i>	character
<i>cpostal</i>	integer
<i>cpostal2</i>	integer
<i>localidade</i>	character
<i>contacto</i>	character

Tabela 11 Estrutura de dados da tabela *transportadoras*

<i>Coluna</i>	<i>Tipo</i>
<i>id</i>	integer
<i>nome</i>	character
<i>ldmotorista</i>	integer
<i>regiaook</i>	array
<i>nacional</i>	boolean
<i>escalao1</i>	numeric
<i>escalao2</i>	numeric

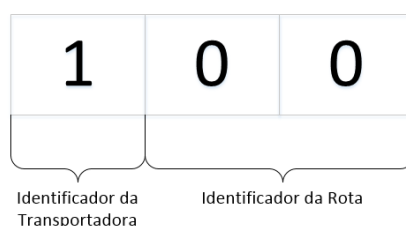


Figura 17 Exemplo id de identificação de transportadora

identificador a ser inserido na aplicação móvel para identificação. No caso de o *id* terminar em 00, a transportadora não apresenta rotas específicas. Assim, sendo o *id* é único para cada transportadora. A coluna *ldmotorista* apenas deve ser preenchida quando o motorista é conhecido e será relacionada com a tabela *motoristas* para identificação do mesmo. A coluna *nacional* é um booleano que identifica se a transportadora faz entregas em todo o país. Caso a cobertura não seja nacional, deve ser preenchido o *array regiaook* com os códigos postais de actuação. As colunas *escalao1* e *escalao2* deverão ser preenchidas com os valores dos custos do transporte corresponde a cada escalão de produtos.

- **Edição:** é possível editar as colunas *ldmotorista*, *regiaook*, *nacional*, *escalao1* e *escalao2* na tabela *transportadoras*.
- **Eliminação:** esta função elimina toda a informação armazenada na base de dados sobre a transportadora.

4.2.1.4. *MOTORISTAS*

Os motoristas são identificados na base de dados na tabela *motoristas*, os dados associados apenas podem ser alteradas pelo administrador e baseiam-se na estrutura presente na Tabela 12.

Tabela 12 Estrutura de dados da tabela *motoristas*

<i>Coluna</i>	<i>Tipo</i>
<i>id</i>	character
<i>nome</i>	character
<i>contacto</i>	character

As funções possíveis na gestão dos dados dos motoristas são:

- **Criação:** a criação de um novo registo de motorista é realizada com o preenchimento obrigatório dos campos *id*, *nome*. O campo *id* identifica o motorista podendo ser identificado através de um nome, número ou combinação de ambos, com um máximo de 10 caracteres. A criação de um novo motorista pressupõe a introdução de um *id* único, assim sendo o sistema verifica se o *id* já se encontra em uso, antes da introdução dos novos dados na tabela.
- **Edição:** apenas é possível editar a coluna *contacto*, sendo necessário criar um novo registo caso se queira alterar algum dos outros campos.
- **Eliminação:** esta função marca o registo como inactivo, de forma a manter a integridade dos dados.

4.2.1.5. ENVIOS

A tabela *envios* armazena os dados relativos às encomendas submetidas, esta é a tabela central do sistema, sendo que as restantes tabelas surgem como suporte às informações presentes nesta. Os dados (Ver Tabela 13) que constam desta tabela são inseridos de forma indirecta, sem necessidade de introdução manual do utilizador. As funções possíveis na gestão dos envios são:

Criação: a criação de uma nova entrada na base de dados tende a ser feita de forma automática, através da recolha dos dados da factura. Sendo de qualquer forma possível realizar a operação através de um formulário de preenchimento manual. Inicialmente e obrigatoriamente são preenchidos as colunas *ano*, *factura*, *cbarras*, *dcriacao*, *contribuinte*, *cobrar*, *statusid*, *completo*. A coluna *cbarras* foi criado para atribuir à encomenda um número identificativo único, pois o número da factura pode-se repetir com a mudança de ano, assim este campo corresponde à união do ano com o número da factura e é também o campo utilizado como informação para a criação do código de barras. As colunas *contribuinte*, *cobrar*, *vcobrar*, *transportadora* e *ficheiro* são preenchidas com os dados

Tabela 13 Estrutura de dados da tabela *envios*

<i>Coluna</i>	<i>Tipo</i>
<i>ano</i>	integer
<i>factura</i>	character
<i>cbarras</i>	character
<i>dcriacao</i>	timestamp
<i>contribuinte</i>	character
<i>cobrar</i>	boolean
<i>vcobrar</i>	character
<i>drecolha</i>	timestamp
<i>idtransportadora</i>	integer
<i>idmotorista</i>	character
<i>dentrega</i>	timestamp
<i>tentrega</i>	timestamp
<i>dlimite</i>	timestamp
<i>atraso</i>	boolean
<i>tatraso</i>	timestamp
<i>peso</i>	character
<i>volume</i>	character
<i>statusid</i>	integer
<i>completo</i>	boolean
<i>ficheiro</i>	character
<i>comentario</i>	text
<i>coordenadas</i>	character

retirados da factura, a coluna *cobrar* é um booleano que indica se a encomenda é à cobrança ou não, a coluna *vcobrar* é preenchida com o valor total da factura. A coluna *transportadora* é preenchida mediante o destino da encomenda, o sistema recolhe o código postal e verifica quais as transportadoras que actuam nessa área, fazendo a atribuição mediante o resultado. O campo *ficheiro* corresponde ao nome do ficheiro *Portable Document Format (PDF)* gerado na emissão da factura. As colunas *dcriação*, *drecolha*, *dentrega*, *tentrega*, *tatraso* correspondem à data de criação da factura, data de recolha, data de entrega, tempo de entrega e tempo de atraso na entrega da encomenda ao cliente, todas as datas e tempos são guardadas em *timestamp*¹ sem fuso horário, tornando assim este campo padronizado e facilitando a comparação entre duas marcas temporais. A coluna *atraso* funciona como campo de controlo que indica se a encomenda foi entregue atrasada em relação à data que consta na coluna *dlimite* e que é definida no momento de atribuição da transportadora. A coluna *statusid* indica o estado actual da encomenda podendo ser “Em Picking”, “Pronto para Entrega”, “Em

¹ *Timestamp* (marca temporal) é uma identificação codificada da data e hora de uma ocorrência.

Distribuição”, “Entregue” e “Falha Entrega”, assumindo valores entre 0 e 4, respectivamente. A coluna *coordenadas* guarda o registo das coordenadas do local de entrega da encomenda. O booleano *completo* funciona como campo de controlo que indica se a encomenda já foi entregue ou não. De entre todos, as colunas que podem sofrer alterações através da utilização da aplicação móvel são *drecolha*, *dentrega*, *tentrega*, *atraso*, *tatraso*, *statusid*, *completo*, *comentário* e *coordenadas*.

- **Edição:** na tabela *envios* é possível alterar as colunas correspondentes ao contribuinte, à transportadora e ao valor a cobrar.
- **Eliminação:** esta função elimina toda a informação armazenada na base de dados sobre o envio.

A Figura 18 apresenta a relação entre as tabelas que constituem a base de dados. A tabela principal é a tabela *envios* sendo as restantes utilizados como suporte de informação. A tabela *utilizadores* não tem nenhuma relação com as restantes, pois só é utilizada para identificar os utilizadores da aplicação *web*.

4.2.2. RECOLHA DE DADOS DA FACTURA

Os dados que constituem o *core* central do sistema, são recolhidos em diferentes fases da operação, que vai desde a submissão de uma encomenda até à sua entrega ao cliente, e são recolhidos através de métodos distintos.

Inicialmente são recolhidas as informações necessárias para a criação do registo da encomenda na base de dados. Estas informações são recolhidas em *background*, de forma automática e transparente. Contudo, existe possibilidade desses dados serem inseridos manualmente através da aplicação *web*, caso haja essa necessidade.

Posteriormente e já na fase do transporte da encomenda, são realizadas as recolhas das restantes informações utilizando a aplicação móvel, desenvolvida para esse efeito. Nesta situação as primeiras recolhas são realizadas no momento do levantamento da encomenda por parte do motorista e a recolha de informações final fica destinada a ser realizada no momento da entrega da encomenda ao cliente.

As informações iniciais sobre as encomendas são recolhidas no momento da emissão da factura. Nesta fase o sistema ideal era realizar uma recolha dos dados directamente do *software* de facturação utilizado na empresa, contudo essa solução não foi possível devido à

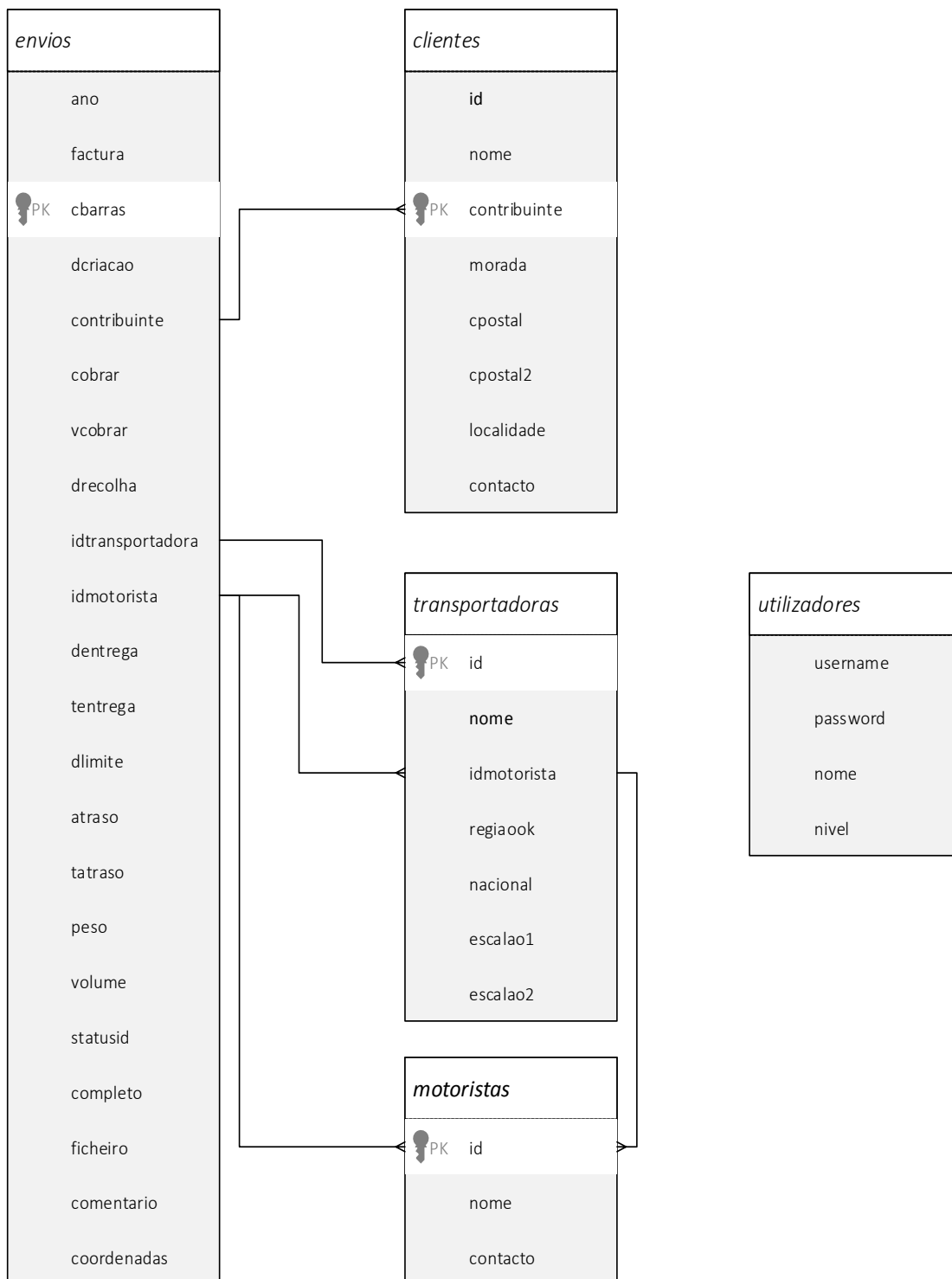


Figura 18 Relação entre tabelas da base de dados

indisponibilidade por parte departamento informático para realizar a integração dos sistemas. Assim sendo, optou-se pela recolha desses dados a partir do ficheiro da factura em formato *PDF*. A Figura 19 apresenta o diagrama de processo de recolha desses dados. Os ficheiros em formato *PDF* são emitidos pelos operadores do *call-center*, que ao invés de submeterem a factura para impressão habitual fazem uma impressão em formato *PDF*, para um local

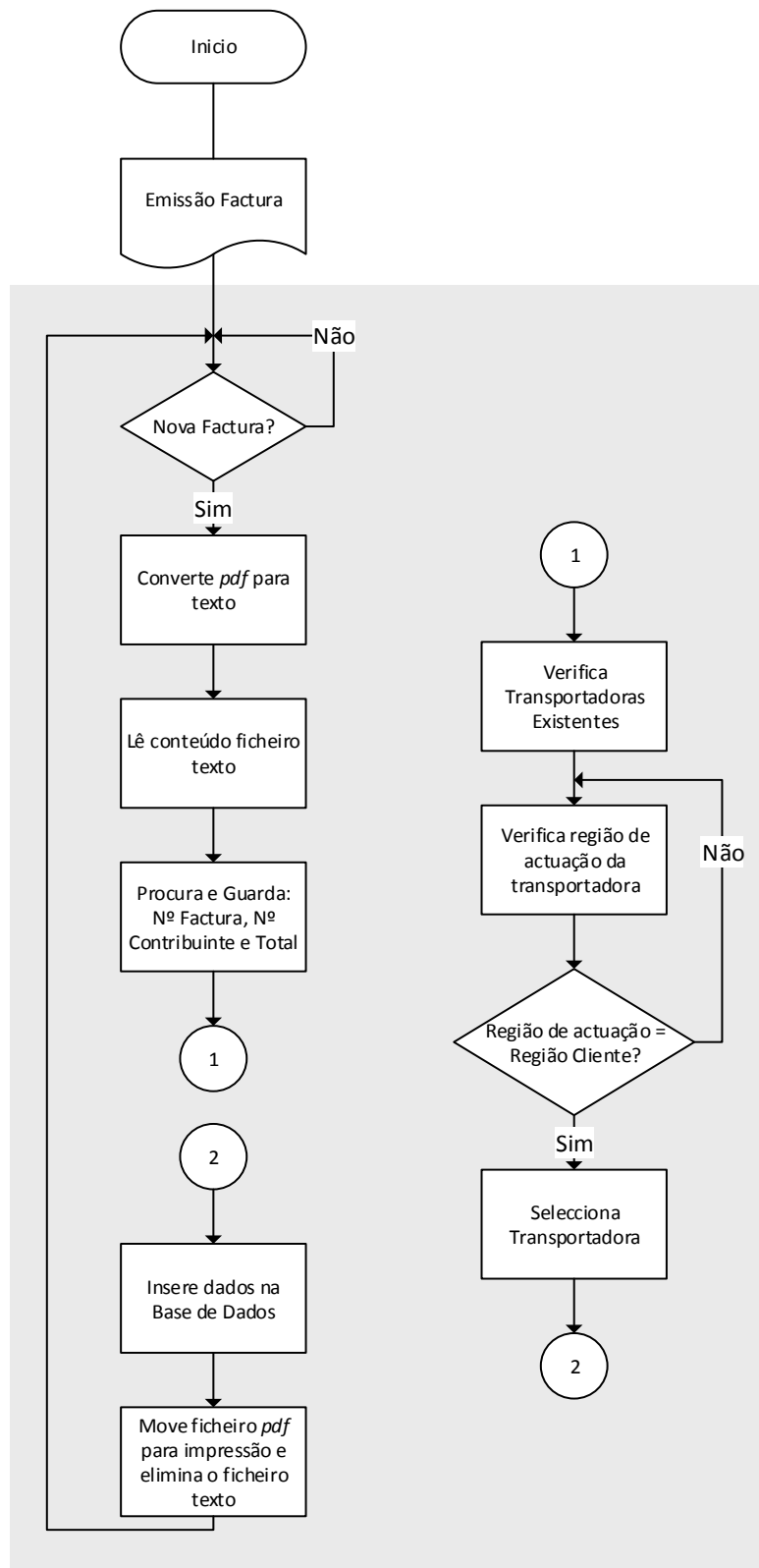


Figura 19 Processo de recolha de dados a partir da factura

específico do servidor, onde são depois manipulados e é dada a ordem de impressão em papel.

Todo o processo de manipulação dos documentos *PDF* foi desenvolvido em linguagem PHP, para uniformização das linguagens utilizadas. O processo inicia-se com a emissão da factura, que é guardada em formato *PDF* num directório do servidor previamente estabelecido. Esse directório está constantemente a ser monitorizado, com a intenção de se verificar se existem novos ficheiros *PDF* para serem processados. A monitorização é realizada com recurso a um ficheiro *batch* que tem como objectivo executar ciclicamente os comandos do código *php* encarregue por recolher os dados do ficheiro. Caso existam, novos ficheiros em formato *PDF* (só estes são considerados), é utilizada o *script pdftotext*, desenvolvido pela *Xpdf*, que efectua a conversão do ficheiro para o formato texto, tornando a leitura dos dados, mais simples.

Após originar o ficheiro texto, todo o seu conteúdo é armazenado numa *string*, essa *string* é lida em busca das palavras-chave que indicam as posições em que se encontram os dados pretendidos. A partir destas pesquisas é possível recolher todos os dados necessários na factura e que se consideram necessários, como o número da factura, o número de contribuinte do cliente e o valor total facturado. No momento, em que se tem acesso ao número de contribuinte, obtém-se em simultâneo acesso aos dados do cliente, fazendo a correspondência com o mesmo número na tabela *clientes*, da base de dados. Assim sendo, é possível verificar qual a transportadora, das disponíveis, que pode ficar encarregue da entrega da encomenda. Este procedimento é realizado comparando o código postal da morada do cliente, com os códigos postais que são abrangidos pelas operações das transportadoras. Após a selecção da transportadora responsável pela entrega da encomenda, é realizada a introdução de um novo registo na base de dados na tabela *envios* com todos estes dados. Complementarmente é guardada a data e hora da criação deste registo.

No final, o ficheiro texto já não é mais necessário e é eliminado. O ficheiro *PDF* original é movido para outra pasta para ser impresso. A ordem de impressão dos documentos é realizada em *background* utilizando um *script* que monitoriza esta pasta e sempre que um novo documento dá entrada, envia-o para o *spooler* de impressão.

4.2.3. APLICAÇÃO ANDROID

A aplicação móvel foi desenvolvida para ambientes *Android* por ser o sistema operativo mais utilizado e em expansão no número de utilizadores na Europa e em Portugal. O desenvolvimento em *Android* é realizado em Java com recurso a um *kit* de desenvolvimento,

que disponibiliza as ferramentas e bibliotecas necessárias para a criação das aplicações. O Anexo B apresenta uma descrição sobre linguagem Java e apresenta as características do sistema operativo *Android*, assim como uma breve introdução ao desenvolvimento deste tipo de aplicativos.

A aplicação móvel foi desenvolvida utilizando ambientes de desenvolvimento integrado (*IDE – Integrated Development Environment*). Os principais *IDE open-source* para programação em Java são o Eclipse e o NetBeans. Para o desenvolvimento da solução foi utilizado o Eclipse, pois são disponibilizados pela *Google plug-ins* para este *IDE*. A configuração deste *software* para desenvolvimento de aplicativos *Android* pode ser consultada no Anexo C.

A aplicação *Android* não terá apenas a função de actualizar os registos da base de dados, mas actuará também como auxiliar do motorista, com informações acerca das encomendas que estão em curso de entrega.

4.2.3.1. ARMAZENAMENTO DE DADOS

A aplicação móvel tem o seu próprio espaço de armazenamento de dados, que foi utilizado com o objectivo de diminuir a troca de dados entre a aplicação e o servidor e ao mesmo tempo dar uma resposta mais rápida ao utilizador. As bases de dados criadas são acessíveis através de qualquer classe da aplicação, mas são inacessíveis através do exterior da aplicação. Para a criação da base de dados *SQLite* é recomendado criar uma subclasse *SQLiteOpenHelper*, onde são executados os comandos para as operações na base de dados. A Figura 20 mostra um exemplo de criação de uma base de dados *SQLite* em *Android*.

Os dados das encomendas são representados na memória através da classe *Delivery* e apresenta uma estrutura de dados como a figurada na Tabela 14.

São possíveis as seguintes funções de gestão:

- **Criação:** A criação de um registo de encomenda leva ao preenchimento obrigatório da maioria dos campos da Tabela 14. O campo *order_id* corresponde ao número identificativo da encomenda, o *receivedDate* é a data da recolha por parte da transportadora, no caso de existir um valor a cobrar este é preenchido no campo *shipping* e por fim são preenchidos os campos relacionados com os dados do cliente, entre eles, *clientId*, *clientName*, *clientAdress* e *clientContact*. Estes registos são

```

public class DictionaryOpenHelper extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 2;
    private static final String DICTIONARY_TABLE_NAME =
"dictionary";
    private static final String DICTIONARY_TABLE_CREATE =
+
        "CREATE TABLE " + DICTIONARY_TABLE_NAME + " ("
        KEY_WORD + " TEXT, " +
        KEY_DEFINITION + " TEXT);";

    DictionaryOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(DICTIONARY_TABLE_CREATE);
    }
}

```

Figura 20 Extracto de código para criação base de dados SQLite

Tabela 14 Estrutura de dados da classe *Delivery*

<i>Coluna</i>	<i>Tipo</i>
<i>order_id</i>	String
<i>receivedDate</i>	Date
<i>shipping</i>	Float
<i>clientId</i>	Integer
<i>clientName</i>	String
<i>clientAdress</i>	String
<i>clientContact</i>	String
<i>deliver</i>	Boolean
<i>deliverDate</i>	Date
<i>dateDiference</i>	Float
<i>comment</i>	String
<i>coordinate</i>	String

criados no momento em que o utilizador faz um pedido ao servidor da lista de encomendas para entrega.

- **Edição:** os registos são editados, de forma transparente ao utilizador e apenas podem ser editados os campos *deliver*, *deliverDate*, *dateDiference*, *comment* e *coordinate*. Estes campos são actualizados no momento em que o utilizador dá uma ordem de sucesso ou insucesso na entrega. O campo *deliver* é um booleano que indica se a

encomenda foi entregue ou não, o campo *deliverDate* regista a data e hora da entrega ao cliente e é calculado recorrendo ao *dateDifference* que indica o tempo decorrido entre o momento da recolha e o momento da entrega da encomenda. No momento em que é dada ordem de sucesso ou falha na entrega são registadas as coordenadas, e armazenadas no campo *coordinate*, opcionalmente o utilizador pode também deixar um comentário que ficará guardado no campo *comment*.

- **Eliminação:** a exclusão de um registo de encomenda implica a eliminação total dos dados armazenados na memória local.

4.2.3.2. TROCAS DE INFORMAÇÃO

Para estabelecer a troca de dados entre a aplicação móvel e a base de dados do servidor foi necessário estabelecer um protocolo de comunicação. As comunicações efectuadas entre a aplicação móvel e o *back-end* são realizadas através de um serviço *web Restful*, usando mensagens HTTP. As respostas por parte do *back-end* para a aplicação móvel são estruturadas em JavaScript Object Notation (*JSON*), tornando-se mais coerentes e aumentando a interoperabilidade entre sistemas, mantendo a simplicidade e o rápido processamento. No Anexo C é apresentado mais detalhadamente os serviços *web*, como se processa o seu funcionamento e a formatação de dados *JSON*.

A Figura 21 exemplifica como é realizado um pedido *Http Post*, neste caso são passados dois pares de chave/valor. Na implementação do sistema, os pares de chave/valor, variam mediante a operação a realizar. As operações podem ser de leitura dos registos da base de

```
public void postData() {
    // Create a new HttpClient and Post Header
    HttpClient httpClient = new DefaultHttpClient();
    HttpPost httppost = new HttpPost("http://www.yoursite.com");

    // Add your data
    List<NameValuePair> pairs = new ArrayList<NameValuePair>();
    pairs.add(new NameValuePair("key1", "value1"));
    pairs.add(new NameValuePair("key2", "value2"));
    httppost.setEntity(new UrlEncodedFormEntity(pairs));

    // Execute HTTP Post Request
    HttpResponse response = httpClient.execute(httppost);
}
```

Figura 21 Pedido *Http Post* em *Android*

Tabela 15 Status respostas à aplicação móvel

<i>Status</i>	<i>Mensagem</i>
0	Lista Entregas Actualizada
1	Informação Actualizada
2	Erro no acesso à BD
3	Identificação Transportador Inválido
4	Encomenda Inexistente
5	Operação não Conhecida

dados em relação às encomendas alocados ao transportador e neste caso são enviados dois pares chave/valor com a informação da operação a realizar e a identificação do transportador. Quando o transportador tem acesso à lista, pode submeter o sucesso ou insucesso da entrega das encomendas e neste caso os pares chave/valor enviados identificam a operação, o número da encomenda, a data da entrega, o resultado, um comentário e as coordenadas.

Em resposta ao pedido, realizado pela aplicação móvel, é retornada uma mensagem codificada em *JSON*. Este tipo de codificação é de fácil interpretação, não necessita de muita memória de processamento e pode ser interpretada em qualquer tipo de linguagem, adequando-se perfeitamente ao pretendido.

Aos pedidos da lista de encomendas ou de submissão de dados com o servidor, podem ser retornadas respostas com diferentes resultados. Identificadas pelo campo *status* da formatação *JSON* (Ver Tabela 15). As considerações a ter na formatação de dados *JSON*, são apresentadas no Anexo C. As respostas do lado do servidor seguem sempre um formato bem definido e distinto quando é realizado um pedido ou uma submissão. A Figura 22 apresenta a resposta dada pelo servidor a um pedido da aplicação móvel e mostra também como é entendida a formatação do objecto *JSON*.

No caso de ser feito um pedido da lista de encomendas a entregar, a resposta refere sempre o campo *status*, que indica o resultado obtido, o campo *responseType*, que identifica esta resposta como uma resposta a um pedido de lista, o campo *length*, que refere a quantidade de encomendas pendentes e o *array deliveries*, com a informação relativa a cada encomenda.

No caso de ser realizado um envio de dados para o servidor, este responde com o *status* da operação e a que encomenda esse *status* corresponde.

Pedido de Lista

Submissão de dados

<pre>\$json=array('response'=>array('status'=>0,'responseType'=>0,'length'=>\$length,'deliveries'=>\$results)); echo ''.json_encode(\$json).'</pre>	<pre>\$json=json_encode(array('response'=>array('status'=>0,'orderId'=>\$orderId))); echo \$json;</pre>
<pre>{ "response":{ "status": , "responseType": , "length": , "deliveries":[{ "orderId": , "shipping": , "clientId": , "clientName": , "clientAddress": , "contact": }] } }</pre>	<pre>{ "response":{ "status": , "orderId": } }</pre>

Figura 22 Resposta dados formato JSON

4.2.3.3. *LAYOUT*

O *layout* da aplicação é construído com recurso à classe *ListView*, que é capaz de mostrar itens em forma de lista, a Figura 23 mostra a declaração *Extensible Markup Language (XML)* de um *layout* do tipo lista.

Para gerir os dados mostrados na lista é utilizado um *adapter*. O *adapter* gere o modelo de dados e adapta-os a uma linha individual numa *list view*. Cada linha numa *list view* pode adquirir o aspecto pretendido pelo programador, a Figura 24 mostra um exemplo de uma linha da lista com dois campos de texto. O *adapter* vai preencher os campos editáveis do *layout* da linha através do método *getView()*.

```
<ListView
xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/listview"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content" />
```

Figura 23 Código XML do *layout* de uma *list view*

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:minHeight="50dp"
    android:orientation="horizontal"
    android:padding="5dip" >
    <TextView
        android:id="@+id/firstline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textStyle="bold"
        android:typeface="sans"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/secondline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/firstline"
        android:layout_marginTop="5dip"
        android:text=""
        android:textColor="#343434"
        android:textSize="15sp" />
</RelativeLayout>

```

Figura 24 Código XML do *layout* de uma linha de *list view*

A Figura 25 apresenta o *layout* da janela principal da aplicação, com alguns exemplos de encomendas na lista e da *dialog box* que surge quando o utilizador interage (clica) com uma linha da lista. Cada linha da lista corresponde a uma encomenda diferente e que ainda não foi dada como entregue, as informações apresentadas correspondem ao nome do cliente, ao código único da encomenda, igual ao código lido no código de barras da etiqueta, a morada do cliente e o valor a cobrar caso este seja diferente de zero. Clicando em qualquer linha da lista surge uma *dialog box* que permite indicar se a encomenda foi entregue com sucesso ou se houve alguma falha na entrega e opcionalmente um comentário.

4.2.3.4. RECOLHA COORDENADAS GEOGRÁFICAS

Como forma de validação e controlo do local de entrega da encomenda a aplicação móvel recolhe as coordenadas geográficas, que podem posteriormente ser utilizadas pelo supervisor para confirmar o local onde foi realizada a submissão.



Figura 25 Janela principal aplicação móvel (a) e *dialog box* da interação com linha da lista (b)

A localização geográfica pode ser obtida através do GPS do equipamento ou através da ligação à rede. A classe *LocationManager* dá acesso aos serviços de localização do *Android*. O pedido da localização do dispositivo deve ser feito chamando o método *requestLocationUpdates()* da classe *LocationManager* e passando-o para o *LocationListener*. O *LocationListener* é notificado aquando da alteração da localização do dispositivo. A Figura 26 mostra como definir o *LocationListener* e pedir actualizações da localização.

O método *requestLocationUpdates()* permite definir os parâmetros a utilizar para obter a posição geográfica. O primeiro parâmetro define o método a utilizar para triangular a posição, podendo ser através de GPS (*GPS_PROVIDER*) ou através da rede (*NETWORK_PROVIDER*). O segundo e terceiro parâmetro definem a frequência com que a localização é actualizada, o segundo parâmetro define o intervalo de tempo entre actualizações e o terceiro a distância mínima entre actualizações. O último parâmetro é o *LocationListener* que recebe as notificações das novas localizações.

```

// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Called when a new location is found by the network
location provider.
        makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status,
Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};

// Register the listener with the Location Manager to receive
location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK
_PROVIDER, 0, 0, locationListener);

```

Figura 26 Extracto de código para obter posição geográfica

4.2.4. APLICAÇÃO WEB

A aplicação *web* funciona como a interface entre a unidade de armazenamento, processamento de dados e o utilizador, toda a aplicação foi desenvolvida em PHP com extractos em *JavaScript* (consultar Anexo B).

A aplicação *web* disponibiliza as seguintes funcionalidades:

- Localizar encomendas;
- Gerar etiquetas para identificação da encomenda;
- Lista de *picking*, para apoio à operação de *picking* da encomenda;
- *Dashboard* com informações das operações diárias, semanais, mensais e anuais;
- Gestão da base de dados.

A Figura 27 apresenta a janela inicial da aplicação *web*, como é possível verificar na imagem a aplicação *web* é dividida nas cinco funcionalidades descritas, que apenas podem ser acedidas após login do utilizador e mediante as permissões que este possuir. A cada utilizador pode ser atribuída permissão de administrador, tendo assim acesso a todas as funcionalidades da aplicação *web*, permissões intermédias, ficando neste caso sem acesso à *dashboard* e à gestão da base de dados, e permissões básicas sendo atribuído apenas acesso à funcionalidade de localizar envios.

4.2.4.1. LOCALIZAR ENVIO

A secção para a localização de envio (Ver Figura 28), permite que o utilizador verifique o estado de uma encomenda inserindo o número da factura correspondente. É disponibilizado ao utilizador a possibilidade de efectuar uma pesquisa avançada (Ver Figura 29) onde é possível realizar a pesquisa das encomendas por data, transportadora, cliente, valor a cobrar, por envio à cobrança e por sinalizados como entregues.

4.2.4.2. IMPRIMIR ETIQUETA

A funcionalidade de imprimir etiqueta, permite reimprimir a etiqueta que contém as principais informações da encomenda, inserindo para isso o ano e número da factura no



Figura 27 Janela inicial aplicação *web*

Portal de Envios

Localizar Envio Imprimir Etiqueta Picking Administração DB Dashboard

Olá, Administrador! Log Out

Introduza o número do documento:

Ano Factura Nº Factura Pesquisar

Código	Data & Hora Criação	Cliente	Data & Hora Recolha	Transportadora	Motorista	Data & Hora Entrega	V. Cobrar	Status	Comentário	Coordenadas
201398586768	18/10/2013 15:24:46	Auto Reparadora	18/10/2013 18:00:00	Transportador2	Motorista 2	19/10/2013 13:40:25	100.68	Entregue	OK	41.152470,-8.630427

Figura 28 Janela “Localizar Envio” aplicação web

Olá, Administrador! Log Out

Data: 27-10-2013 Dia-Mês-Ano

Cliente: Nº Contribuinte

Transportadora: [dropdown]

À Cobrança:

Valor a Cobrar: [input] €

Listar Entregues:

Pesquisar

Código	Data & Hora Criação	Cliente	Data & Hora Recolha	Transportadora	Motorista	Data & Hora Entrega	V. Cobrar	Status	Comentário	Coordenadas
201398137974	27/10/2013 00:58:14	AutoRui	27/10/2013 12:27:34	Transportador3	Manuel	27/10/2013 14:06:10	84.18	Entregue	entregue	38.790507,-9.124629
201398533053	27/10/2013 00:38:04	Moises	27/10/2013 12:27:34	Transportador3	Manuel	27/10/2013 14:07:40		Falha na Entrega	cliente ausente	41.152749,-8.622188

Figura 29 Janela “Localizar Envio Avançado” aplicação web

formulário disponibilizado (Ver Figura 30). A Figura 31 apresenta o exemplo de uma etiqueta utilizada para identificação de uma encomenda. Na etiqueta constam os dados:

- Do cliente (destinatário);
- Da Caetano Parts (remetente);
- Da transportadora responsável pela distribuição da encomenda;
- Do valor a cobrar, caso este exista;
- Do número de unidades e peso, quando disponibilizado;
- Do código de barras para identificação da encomenda através de um leitor de código de barras.

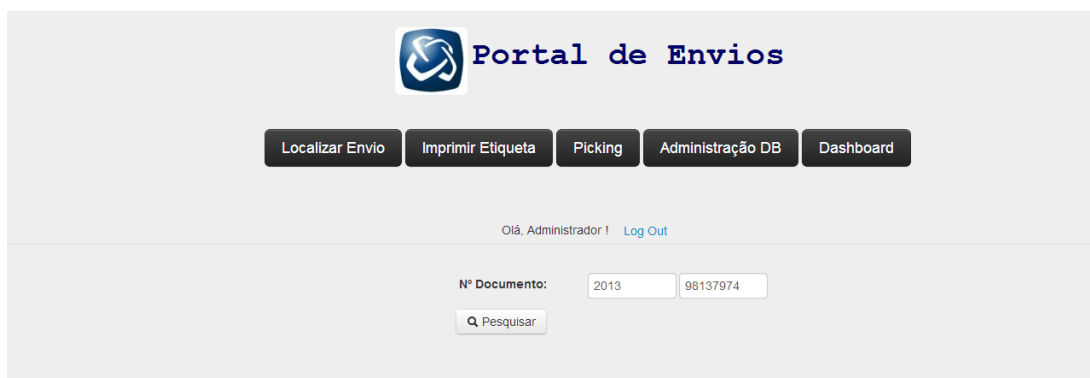


Figura 30 Janela “Imprimir Etiqueta” aplicação *web*

No Anexo B são apresentadas algumas noções básicas sobre códigos de barras. A etiqueta quando impressa deve ser colada no exterior da embalagem da encomenda correspondente.

4.2.4.3. *LISTA PICKING*

Os dados recolhidos ao longo do processo, têm como principal função fornecer à empresa a capacidade de acompanhar as encomendas ao longo da sua fase de vida. Contudo, esses dados são importantes para avaliar o serviço prestado e para análise das operações decorrentes, para além de poderem ser aplicados como facilitadores do processo. Neste âmbito, surge a lista de *picking* que tem como principal objectivo organizar a informação de todas as encomendas pendentes e servir de suporte ao operador de *picking*. A *dashboard*,



Figura 31 Exemplo de etiqueta de despacho

por sua vez, organiza os dados e apresenta-os de forma gráfica e sumarizada, para uma avaliação geral da situação diária, semanal, mensal ou anual.

A lista de *picking* funciona como um intermediário entre o lançamento da encomenda e a passagem da informação para a lista do transportador. Isto é, quando a encomenda é criada e passa pelo processo de recolha de dados e atribuição da transportadora é inserida na lista de *picking* (Ver Figura 32), mostrando ao operador quais as encomendas que se encontram ainda em processo de finalização para despacho. Após o operador proceder à recolha do material que consta na factura da encomenda, deve imprimir a etiqueta com as informações do despacho e dar ordem de encomenda preparada.

Neste momento o estado da encomenda é actualizado na base de dados e deixa de estar no estado “Em *Picking*”, para ficar no estado “Pronto Para Entrega”, significa que a encomenda deixa de estar disponível nesta lista e passa a constar da lista da aplicação móvel.

4.2.4.4. DASHBOARD

A *dashboard* é uma secção da aplicação *web*, que permite ao utilizador ter uma perspectiva geral das operações. Os dados recolhidos ao longo de todo o processo são organizados e manipulados nesta secção, para mostrar os resultados obtidos.

A Figura 33 mostra o *layout* da página inicial do *dashboard*. Desta página fazem parte duas tabelas sumárias das actividades e três gráficos de informação complementar. A tabela



The screenshot shows the 'Portal de Envios' dashboard. At the top, there is a logo and the title 'Portal de Envios'. Below the title are four buttons: 'Localizar Envio', 'Imprimir Etiqueta', 'Picking', and 'Administração DB'. Underneath the buttons, it says 'Olá, Administrador!' and 'Log Out'. The main part of the dashboard is a table with the following data:

Documento	Data & Hora Criação	Cliente	Transportadora	V. Cobrar	Status	Imprimir Etiqueta	Preparado
98137974	15/10/2013 11:48:58	Automeclia	Transportador 1	84.18	Em Picking		<input checked="" type="checkbox"/>
98586768	15/10/2013 11:48:58	Auto Reparadora	Transportador 2		Em Picking		<input checked="" type="checkbox"/>
98586777	15/10/2013 11:48:58	Barbosa	Transportador 2		Em Picking		<input checked="" type="checkbox"/>
98586851	15/10/2013 11:48:59	Tiago Unipessoal	Transportador 1		Em Picking		<input checked="" type="checkbox"/>
98134540	15/10/2013 11:48:57	3 Pintos	Transportador 1	102.77	Em Picking		<input checked="" type="checkbox"/>

Figura 32 Página *web* da lista de *picking*

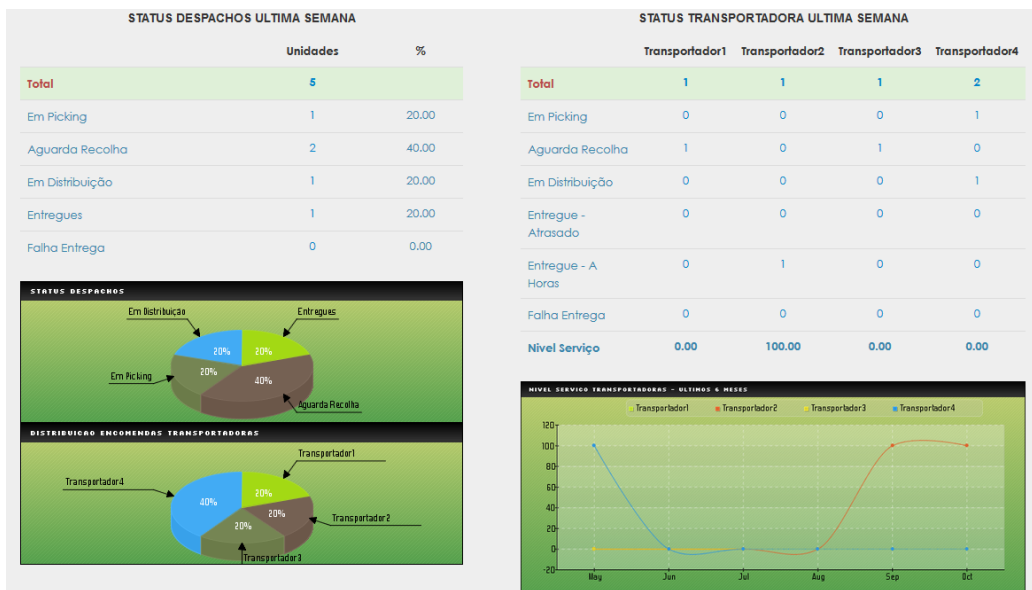


Figura 33 Layout Dashboard

apresentada na Figura 34 apresenta de forma sumária quantas encomendas estão em cada fase do processo. A Figura 35 apresenta dois gráficos que sintetizam e mostram a distribuição das encomendas pelas diferentes fases do processo e a distribuição das encomendas pelas diferentes transportadoras, respectivamente. A Figura 36 apresenta a tabela presente na *dashboard* que organiza o número de encomendas em cada fase do processo agrupadas pelas respectivas transportadoras. Os dados apresentados nas duas tabelas e nos dois gráficos mencionados, podem corresponder à actividade do dia, da última semana, do último mês e do último ano, consoante a escolha do utilizador. O gráfico presente

STATUS DESPACHOS ULTIMA SEMANA

	Unidades	%
Total	5	
Em Picking	1	20.00
Aguarda Recolha	2	40.00
Em Distribuição	1	20.00
Entregues	1	20.00
Falha Entrega	0	0.00

Figura 34 Tabela detalhes globais das operações

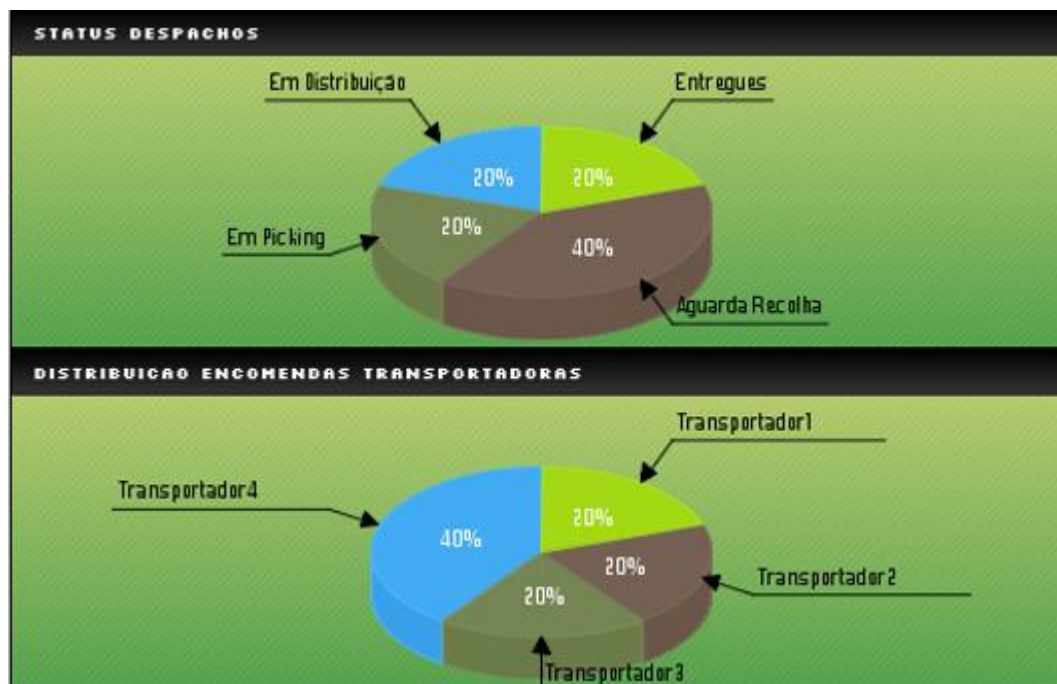


Figura 35 Gráficos da distribuição das encomendas pelas fases e pelas transportadoras

STATUS TRANSPORTADORA ULTIMA SEMANA				
	Transportador1	Transportador2	Transportador3	Transportador4
Total	1	1	1	2
Em Picking	0	0	0	1
Aguarda Recolha	1	0	1	0
Em Distribuição	0	0	0	1
Entregue - Atrasado	0	0	0	0
Entregue - A Horas	0	1	0	0
Falha Entrega	0	0	0	0
Nível Serviço	0.00	100.00	0.00	0.00

Figura 36 Tabela da distribuição das encomendas pelas diferentes fases agrupadas por transportadora na Figura 37 mostra a evolução do nível de serviço de cada transportadora ao longo dos últimos 6 meses. Este nível de serviço é obtido através do rácio do número de encomendas

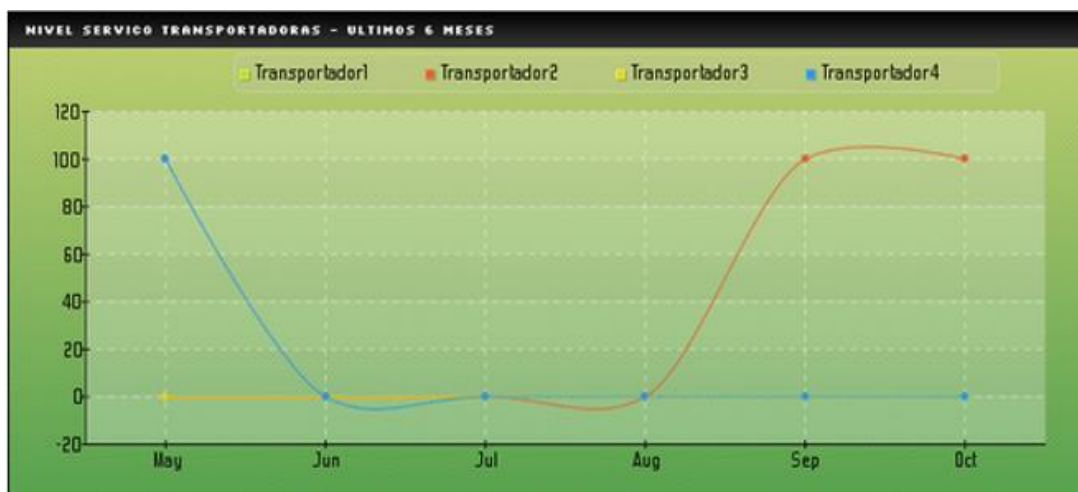



Figura 37 Gráfico evolução do nível serviço da transportadora nos últimos 6 meses

entregue até à hora, previamente definida como limite, em relação ao total de encomendas entregues.

Os itens das tabelas são clicáveis e permitem ao utilizador consultar com detalhe as encomendas referentes à selecção. A Figura 38 mostra a página apresentada quando o utilizador clica no primeiro campo (“5”) da tabela da Figura 34.

Por sua vez, na lista de detalhes apresentada, sempre que uma encomenda tenha um registo de coordenadas geográficas, este campo é clicável e direcciona o utilizador para a página


Dashboard

DETALHES DE TODAS AS ENCOMEDAS ENTRE 14/10/2013 E 22/10/2013

Código	Data & Hora Criação	Cliente	Data & Hora Recolha	Transportadora	Motorista	Data & Hora Entrega	V. Cobrar	Status	Comentário	Coordenadas
201398134540	18/10/2013 14:24:46	3 Pintos	18/10/2013 16:11:00	Transportador1	Motorista1		102.77	Pronto Para Entrega		
201398533053	18/10/2013 14:24:45	Cliente		Transportador3	Motorista 3			Pronto Para Entrega		
201398586768	18/10/2013 14:24:46	Auto Reparadora	18/10/2013 17:00:00	Transportador2	Motorista 2	19/10/2013 13:25:33		Entregue		41.152470,
201398586796	18/10/2013 14:24:47	Viaudy		Transportador4	Motorista 4			Em Picking		
201398586851	18/10/2013 14:24:47	Tiago Unipessoal	18/10/2013 17:43:04	Transportador4	Motorista 4			Em Distribuicao		

Figura 38 Detalhes Encomendas *Dashboard*

web da *Google Maps*. Esta opção está disponível em todas as páginas da aplicação *web*, em que sejam apresentadas coordenadas.

4.2.4.5. **GESTÃO DA BASE DE DADOS**

A aplicação *web* disponibiliza ao administrador das operações um interface com a base de dados que se encontra instalada no servidor. Esta interface dá a possibilidade de o utilizador efectuar algumas operações de gestão. A Figura 39 apresenta o exemplo de um menu de gestão da base de dados, nesta imagem é apresentada a janela para a criação de um novo registo de motorista. As opções disponibilizadas são semelhantes para todas as tabelas da base de dados e permitem criar, modificar e eliminar as informações das encomendas, clientes, transportadoras, motoristas e utilizadores.

4.3. **DEMONSTRAÇÃO DAS FUNCIONALIDADES**

Para assegurar o correcto funcionamento da ferramenta, foram realizados testes e depurações ao longo do desenvolvimento da mesma. Neste subcapítulo é apresentada uma demonstração das funcionalidades do sistema e os resultados obtidos com as acções realizadas. A demonstração é realizada num servidor local, com a aplicação *web* e a unidade de armazenamento instalada no servidor. A aplicação móvel é simulada através do *Android Virtual Device Manager* disponibilizado pelo Eclipse com equivalência a um dispositivo com sistema operativo *Android 2.2*.



The screenshot displays the 'Portal de Envios' web application interface. At the top, there is a logo and the title 'Portal de Envios'. Below this, a navigation bar contains buttons for 'Localizar Envio', 'Imprimir Etiqueta', 'Picking', 'Administração DB', and 'Dashboard'. A secondary navigation bar includes links for 'Home', 'Gerir Envios', 'Gerir Clientes', 'Gerir Transportadoras', 'Gerir Motoristas' (which is highlighted), and 'Gerir Utilizadores'. On the right side of this bar, it says 'Olá, Administrador!' and 'Log Out'. The main content area features three buttons: 'Criar Motorista', 'Eliminar Motorista', and 'Modificar Informações'. Below these buttons is a form with three input fields labeled 'Id:*', 'Nome:*', and 'Contacto:'. A 'Criar' button is positioned at the bottom of the form.

Figura 39 Janela gestão dados dos motoristas

4.3.1. GESTÃO DA BASE DE DADOS

Para o correcto funcionamento da ferramenta é necessário que todas as tabelas da base de dados estejam preenchidas com as informações dos utilizadores, transportadoras, motoristas e clientes. O preenchimento das tabelas pode ser realizado através da importação de ficheiros em formato *Comma-Separated Values (CSV)* e *XML*, com a utilização de ferramentas administrativas baseadas em páginas *web*, como é o caso do *phpPgAdmin* para o *PostgreSQL*.

Para os testes o preenchimento será realizado através da aplicação *web*. Toda a gestão da base de dados será realizada com uma conta de administrador previamente criada.

4.3.1.1. UTILIZADORES

Na aplicação *web* acedendo à página “Administração DB” na aba “Gerir Utilizadores”, é possível criar, modificar e eliminar utilizadores. Foram criados dois utilizadores com os dados da Tabela 16. A Figura 40 mostra o formulário para a criação do utilizador, em que é necessário preencher o *username*, a *password*, o nome e definir o nível de permissões (0 – permissão básica; 1 – permissão intermédia; 2 – permissão de administrador). No momento da submissão do formulário, o sistema verifica se o *username* já está em uso, caso esteja disponível é mostrada a mensagem da Figura 40-1, caso o *username* já não esteja disponível é mostrada a mensagem da Figura 40-2. Na introdução dos dados foram provocados deliberadamente os erros da Figura 40-3. Os utilizadores foram correctamente armazenados na base de dados como mostra a Figura 41.

Tabela 16 Informação utilizadores criados

<i>Username</i>	<i>Password</i>	<i>Nome</i>	<i>Nível</i>
francisco	12345	Francisco	2
pacheco	54321	Pacheco	1



Figura 40 Criação utilizador aplicação *web*

username	password	nome	nível
francisco	8cb2237d0679ca88db6464eac60da96345513964	Francisco	2
pacheco	348162101fc6f7e624681b7400b085eeac6df7bd	Pacheco	1

Figura 41 Resultado tabela *utilizadores* após criação dos utilizadores

No objecto *utilizadores*, apenas pode ser alterado o nível da permissão e a *password*. Assim procedeu-se às alterações que constam na Tabela 17 através do formulário da Figura 42. O resultado final na tabela da base de dados é a presente na Figura 43.

A criação de utilizadores, tem como objectivo controlar o acesso às páginas da aplicação *web*, ou seja, utilizadores com permissões de administrador têm acesso a todas as secções da

Tabela 17 Informação utilizadores após alteração

<i>Username</i>	<i>Password</i>	<i>Nome</i>	<i>Nível</i>
francisco	12345678	Francisco	2
pacheco	54321	Pacheco	0

Figura 42 Alteração dados utilizador aplicação web

username	password	nome	nivel
francisco	7c222fb2927d828af22f592134e8932480637c0d	Francisco	2
pacheco	348162101fc6f7e624681b7400b085eeac6df7bd	Pacheco	0

Figura 43 Resultado tabela *utilizadores* após edição dos dados dos utilizadores

aplicação, utilizadores com permissões intermédias não têm acesso ao controlo da BD nem à *dashboard* e utilizadores com permissões básicas só podem visualizar o estado das encomendas na secção “Localizar Envio”. Esse controlo é realizado através da página inicial da aplicação que apresenta um formulário de *login*, como mostra a Figura 44-1. Caso o utilizador tente aceder às secções da aplicação sem efectuar o *login* ou com um *login* sem as permissões necessárias é mostrada a mensagem da Figura 44-3 e o utilizador é

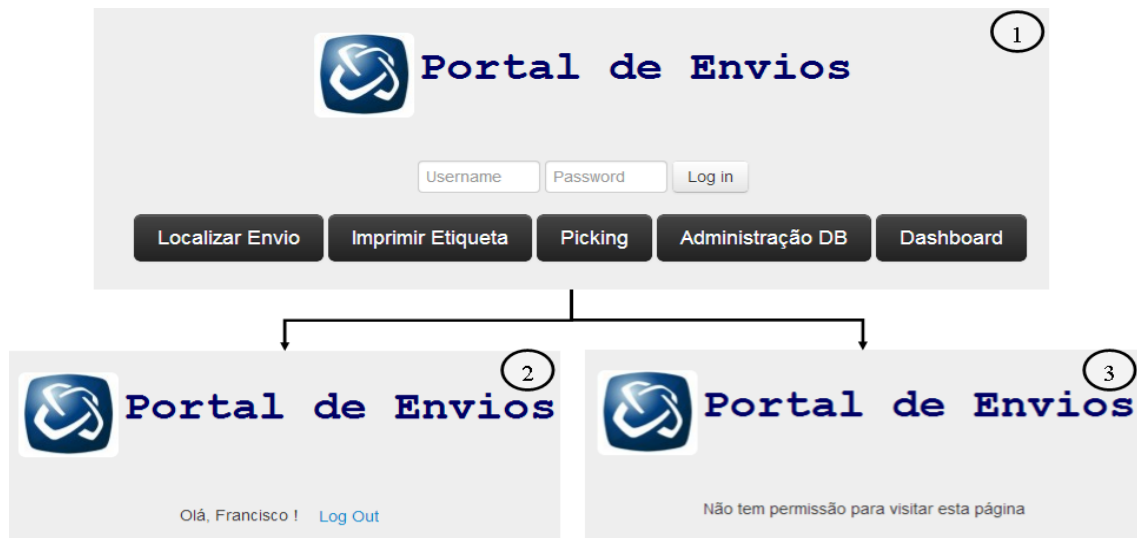


Figura 44 Página inicial e formulário de *login*

reencaminhado para a página inicial. Quando o *login* é efectuado com sucesso, o formulário de login é substituído pela saudação e nome do utilizador, como mostra a Figura 44-2.

4.3.1.2. *TRANSPORTADORAS*

Os processos de criação e modificação dos dados das tabelas da base de dados são idênticos para todos os objectos. Foram criadas três transportadoras, uma com cobertura nacional e duas com cobertura regional e com códigos postais de cobertura distintos. A Figura 45

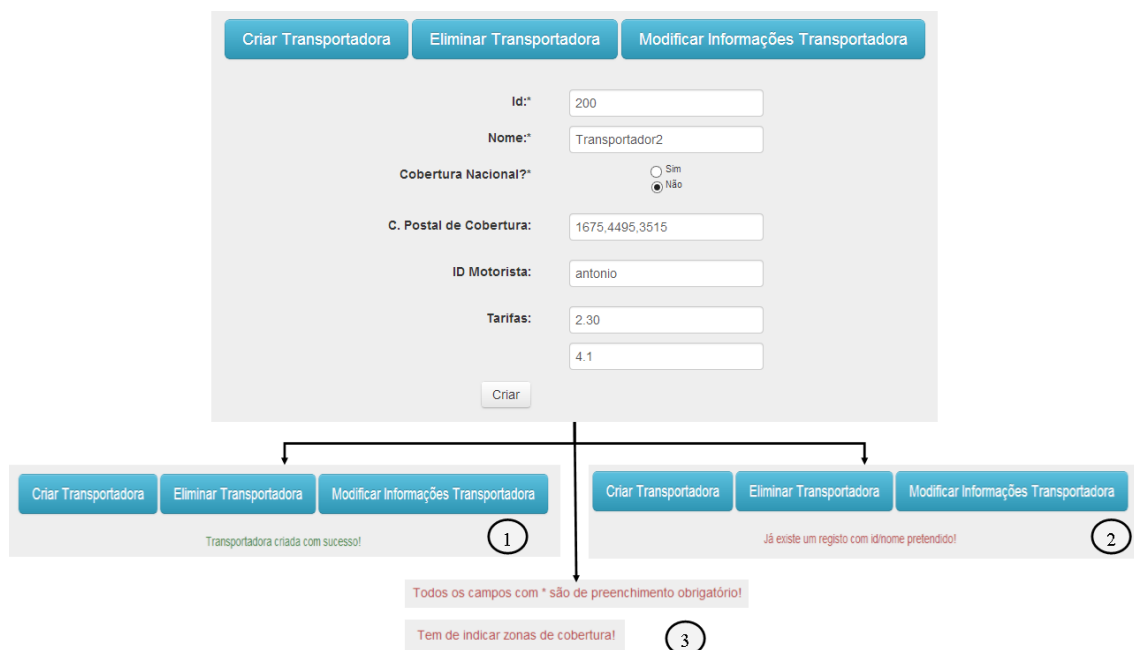


Figura 45 Criação transportadoras aplicação *web*

apresenta o formulário de criação do registo de uma transportadora, através do qual foram criadas as transportadoras com os dados da Tabela 18 e sendo o resultado final na base de dados o apresentado na Figura 46. No momento da criação do novo registo, o sistema verifica se o *Id* introduzido está livre e mostra as mensagens da Figura 45-1 e Figura 45-2, mediante o resultado. A Figura 45-3 mostra as mensagens de erro que podem surgir, caso os campos obrigatórios não sejam preenchidos e caso se indique que a transportadora não é de cobertura nacional e não se preencha o campo relativo aos códigos postais. Outros possíveis erros no preenchimento, como a colocação de “,” no lugar de “.” e vice-versa, são corrigidos no momento da recepção dos dados no servidor.

4.3.1.3. MOTORISTAS

A criação dos registos de motoristas requer o preenchimento e submissão do formulário da aplicação *web*, como mostra a Figura 47. Foram criados três motoristas, que correspondem aos motoristas escalados para cada transportadora. Os registos foram criados com os dados da Tabela 19. O *Id* do motorista deve ser único assim sendo, no momento da criação, o sistema verifica se o *Id* está disponível e mediante a resposta concretiza ou não a inserção

Tabela 18 Informação transportadoras criadas

<i>id</i>	<i>nome</i>	<i>nacional</i>	<i>regiaook</i>	<i>idmotorista</i>	<i>escalao1</i>	<i>escalao2</i>
100	Transportador1	TRUE	null	jose	2.45	4
200	Transportador2	FALSE	1675,4495,3515	antonio	2.30	4
300	Transportador3	FALSE	1000,1700,3000,4460,4580	manuel	3	4.1

<i>id</i>	<i>nome</i>	<i>nacional</i>	<i>regiaook</i>	<i>idmotorista</i>	<i>escalao1</i>	<i>escalao2</i>
100	Transportador1	TRUE	NULL	jose	2.45	4
200	Transportador2	FALSE	{1675,4495,3515}	antonio	2.30	4
300	Transportador3	FALSE	{1000,1700,3000,4460,4580}	manuel	3	4.1

Figura 46 Resultado tabela *transportadoras* após criação

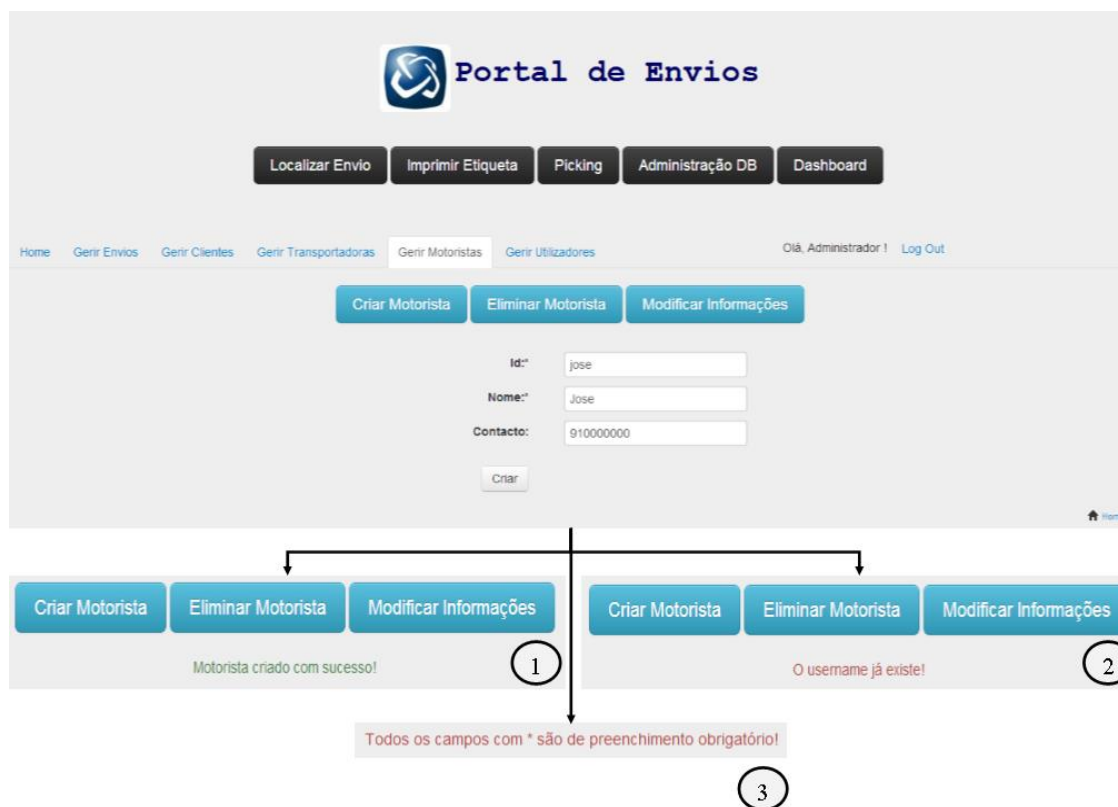


Figura 47 Criação motoristas aplicação web

Tabela 19 Informação motoristas criados

<i>id</i>	<i>nome</i>	<i>contacto</i>
jose	Jose	910000000
antonio	Antonio	920000000
manuel	Manuel	930000000

dos dados do motorista, como mostra a Figura 47-1 e Figura 47-2. Caso não sejam preenchidos os campos obrigatórios é mostrada a mensagem da Figura 47-3. A Figura 48 apresenta os dados dos motoristas na tabela da base de dados após a criação.

4.3.1.4. CLIENTES

Por fim tem de ser criados os registos dos clientes. A Figura 49 apresenta o formulário a ser preenchido para a correcta criação dos registos. No objecto *clientes* o identificador único é

id	nome	contacto
jose	Jose	910000000
antonio	Antonio	920000000
manuel	Manuel	930000000

Figura 48 Resultado tabela *motoristas* após criação

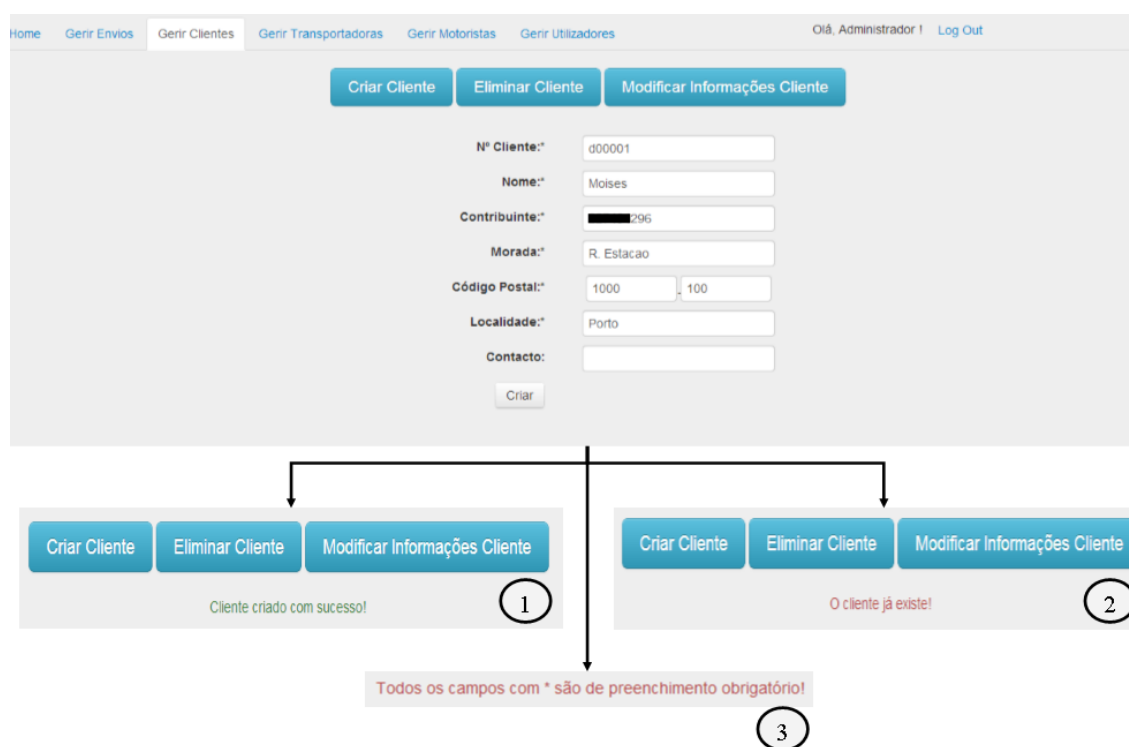


Figura 49 Criação clientes aplicação *web*

o número de contribuinte¹, pois o número de cliente interno pode ser repetido, por exemplo nos clientes sem crédito. Assim, no momento da criação o sistema verifica se o número de contribuinte inserido já se encontra atribuído na base de dados, e consoante a resposta é finalizada ou não a criação do registo e mostra as mensagens da Figura 49-1 e Figura 49-2. Esta tabela em semelhança com as restantes requer que determinados campos sejam preenchidos obrigatoriamente, nesse sentido o sistema, antes de prosseguir, verifica se os

¹ Os números de contribuintes utilizados são reais. Nas imagens apenas são exibidos os últimos 3 algarismos, para salvaguardar a confidencialidade dos dados.

campos obrigatórios foram preenchidos, sendo exibido em caso de erro no preenchimento a mensagem da Figura 49-3.

Foram criados dois registos de clientes com os dados da Tabela 20. A Figura 50 mostra o resultado da tabela *clientes* na base de dados após a criação dos clientes através do formulário. Numa implementação real ao tratar-se de centenas ou milhares de registos é recomendado fazer a criação de um ficheiro CSV com os campos ordenados pela tabela da base de dados e proceder à importação da mesma através de uma ferramenta administrativa.

4.3.2. RECOLHA DE DADOS DAS ENCOMENDAS

Com os dados que asseguram o correcto funcionamento do sistema e que permitem associar as informações retiradas das facturas das encomendas é possível testar e demonstrar o correcto funcionamento das restantes funcionalidades. Como explanado no subcapítulo 4.2.2, o processo inicia-se com a emissão da factura para uma localização predefinida no servidor e a recolha da informação sobre a encomenda pelo algoritmo estabelecido desenvolvido para esse efeito. A Figura 51 apresenta um exemplo da factura em formato *PDF*, com as informações a serem recolhidas. Quando uma nova factura é criada o sistema procede à extracção dos dados e insere um novo registo na base de dados, uma vez que o directório é constantemente monitorizado. Os resultados dos dados extraídos são apresentados na Figura 52-1 caso o algoritmo seja corrido num navegador internet, a Figura 52-2 mostra o que realmente acontece no servidor com a execução do *batch* que corre o código *php* ciclicamente. A Figura 52-3 apresenta o resultado final inserido na base de dados,

Tabela 20 Informação clientes criados

<i>id</i>	<i>nome</i>	<i>morada</i>	<i>c. postal</i>	<i>contacto</i>	<i>contribuinte</i>	<i>localidade</i>
d00001	Moises	R. Estacao	1000-100		*****296	Porto
C00001	AutoRui	R. Casal	1700-471		*****519	Lisboa

<i>id</i>	<i>nome</i>	<i>morada</i>	<i>cpostal</i>	<i>cpostal2</i>	<i>contacto</i>	<i>contribuinte</i>	<i>localidad</i>
d00001	Moises	R. Estacao	...	1000	100	██████296	Porto
C00001	AutoRui	R. Casal	...	1700	471	██████519	Lisboa

Figura 50 Resultado tabela *clientes* após criação

Figura 51 Vista pormenorizada dos dados da factura

1

Nº Factura: 98533053
 Nº Cliente: ████████296
 Dcriacao: 1382830684
 Dlimite: 1382958000
 V. Factura: 15.95
 Cobrar: FALSE
 Transportadora: 300
 Motorista: manuel

```

2
C:\WINDOWS\System32\cmd.exe
N| Factura: 98533053<br>N| Cliente: ████████296<br>Dcriacao: 1382830684<br>Dlimite: 1382958000<br>V. Factura: 15.95<br>Cobrar: FALSE<br>Transportadora: 300 <br>Motorista: manuel
    
```

ano	factura	cbarras	dcriacao	contribuinte	cobrar	vcobrar	drecolha	idtransportadora	idmotorista	dentrega	peso	volume	statusid	completo	ficheiro	comentario	dlimite
2013	98533053	201398533053	1382813801	██████296	FALSE	15.95		300	manuel		NULL	NULL	0	FALSE	IN_01170170	NULL	1382871600

3

Figura 52 Resultado recolha dados da encomenda da Figura 51

o que confirma a correcta execução do algoritmo. As colunas *peso* e *volume* não são preenchidas pois estes dados não são fornecidos pela factura ou outro meio. Foram processadas duas facturas, sendo os resultados finais os apresentados na Figura 53.

ano	factura	cbarras	dcriacao	contribuinte	cobrar	vcobrar	drecolha	idtransportadora	idmotorista	dentrega	peso	volume	statusid	completo	ficheiro	comentario	dlimite
2013	98533053	201398533053	1382830684	██████296	FALSE	15.95		300	manuel		NULL	NULL	0	FALSE	IN_01170170	NULL	1382958000
2013	98137974	201398137974	1382831894	██████519	TRUE	84.18		300	manuel		NULL	NULL	0	FALSE	IN_01310896	NULL	1382958000

Figura 53 Resultado tabela *envios* após processamento das facturadas

Caso seja processada uma factura em que algum dos dados não tenha correspondência com as outras tabelas da base de dados, como por exemplo a não existência dos dados do cliente da factura na tabela *clientes*, esta será lida correctamente e o registo inserido na base de dados, contudo nas tarefas seguintes o registo da encomenda não será apresentado ou poderá ser apresentado com falhas.

4.3.3. LISTA DE PICKING

Com a introdução do registo de uma encomenda na base de dados está é considerada inicialmente como estando em processo de *picking*. Nesse sentido a lista de *picking* funciona como guia de apoio ao trabalho do operador responsável pelo *picking*. Para além de funcionar como estado intermédio entre a colocação da encomenda e a indicação de estar “Pronto Para Entrega”.

A Figura 54 apresenta a lista de picking com as duas encomendas processadas anteriormente. Nesta página da aplicação *web* o utilizador tem ainda a possibilidade de criar e imprimir uma etiqueta carregando no botão “Imprimir Etiqueta”, esta etiqueta após impressão deve ser colocada na parte exterior do pacote a ser transportado. A Figura 55 apresenta a etiqueta gerada para a primeira encomenda da lista, sendo este o modelo de etiqueta para todas as

Documento	Data & Hora Criação	Cliente	Transportadora	V. Cobrar	Status	Imprimir Etiqueta	Preparado
98533053	27/10/2013 00:38:04	Moises	Transportador3		Em Picking		<input checked="" type="checkbox"/>
98137974	27/10/2013 00:58:14	AutoRui	Transportador3	84,18	Em Picking		<input checked="" type="checkbox"/>

Figura 54 Lista de *picking* após criação do registo das encomendas



Figura 55 Etiqueta gerada com os dados da encomenda

encomendas. O código de barras QR quando lido apresenta o número identificativo e único da encomenda na base de dados, o mesmo número apresentado entre “*”.

Quando a encomenda está pronta para ser entregue, o operador de *picking* deve carregar no botão “Preparado”, para o registo da encomenda ser retirada da lista de *picking* e simultaneamente alterar o estado na base de dados para “Pronto Para Entrega”, identificado pelo número 1 na coluna *statusid* (Ver Figura 56-2).

4.3.4. APLICAÇÃO MÓVEL

Quando os registos das encomendas estão com o *statusid* da tabela *envios* a 1, significa que esta está pronta a ser recolhida pelo motorista, ou seja, os registos ficam disponíveis para ser mostrados na aplicação móvel.

Documento	Data & Hora Criação	Cliente	Transportadora	V. Cobrar	Status	Imprimir Etiqueta	Preparado
98533053	27/10/2013 00:38:04	Moises	Transportador3		Em Picking		<input type="checkbox"/>
98137974	27/10/2013 00:58:14	AutoRui	Transportador3	84.18	Em Picking		<input checked="" type="checkbox"/> 1

ano	factura	cbarras	dcriacao	contribuinte	cobrar	vcobrar	drecolha	idtransportadora	idmotorista	dentrega	peso	volume	statusid	completo	ficheiro	comentario	cli
2013	98137974	201398137974	1382831894	503213519	TRUE	84.18		300	manuel		NULL	NULL	0	FALSE	IN_01310896	NULL	1382958000
2013	98533053	201398533053	1382830684	227494296	FALSE	15.95		300	manuel		NULL	NULL	1	FALSE	IN_01170170	NULL	1382958000

Figura 56 Actualização tabela *envios* após confirmação de encomenda preparada

A Figura 57 mostra como obter a lista das encomendas para entrega de um motorista. Na primeira utilização é necessário navegar para o menu e seleccionar “Preferências” (Ver Figura 57-1), no menu preferências introduz-se o *id* do motorista (Ver Figura 57-2), tal como consta na identificação da base de dados, e a lista é actualizada com as encomendas pendentes de entrega assignadas ao *id* do motorista introduzido (Ver Figura 57-3). Caso o *id* introduzido não tenha correspondência com nenhum *id* da base de dados a lista mantém-se em branco e é mostrada a mensagem da Figura 57-4. No momento em que é criada a lista na aplicação móvel é assumido que o motorista vai proceder à recolha das encomendas, assim sendo, a tabela *envios* da base de dados é actualizada com a data e hora da recolha (em *timestamp*) e é alterado o *statusid* para o nível 2 (Em Distribuição), como mostra a Figura 58.

Com a lista das entregas actualizada, o responsável pela entrega tem tudo preparado para a distribuição das encomendas. No momento da entrega ou tentativa de entrega de uma determinada encomenda o utilizador apenas tem de seleccionar a linha correspondente à encomenda e surgirá uma janela de interacção com o utilizador (Ver Figura 59-1). Nesta janela é possível confirmar o sucesso ou a falha na entrega da encomenda e deixar um comentário opcional. A Figura 59-2 mostra a confirmação de uma encomenda entregue com o aparecimento da mensagem “Informação Actualizada!” e a linha é removida da lista. Quando a encomenda é dada como não entregue (Ver Figura 59-3), essa informação é actualizada na base de dados e a linha correspondente continua na lista da aplicação como



Figura 57 Obtenção lista aplicação móvel

ano	factura	cbarras	dcriacao	contribuinte	cobrar	vcobrar	drecoilha	idtransportadora	idmotorista	dentrega	peso	volume	statusid	completo	ficheiro	comentario	dlim
2013	98137974	201398137974	1382831894	503213519	TRUE	84.18	1382876854	300	manuel		NULL	NULL	2	FALSE	IN_01310896	NULL	138295801
2013	98533053	201398533053	1382830684	227494296	FALSE	15.95	1382876854	300	manuel		NULL	NULL	2	FALSE	IN_01170170	NULL	138295801

Figura 58 Tabela *envios* após de actualização da lista de encomendas na aplicação móvel

uma encomenda pendente. Caso, o registo da encomenda não se encontre na base de dados é mostrada a mensagem da Figura 59-4.

Qualquer uma das operações descrita desencadeia a actualização da base de dados. A Figura 60 mostra as alterações realizadas na base de dados durante esta demonstração, realçando a alteração do *statusid*, o registo da data e hora de entrega, do tempo de entrega, do comentário e das coordenadas. Em simultâneo o sistema verifica se a encomenda foi entregue em atraso, comparando a data de entrega com a data limite. Caso a encomenda seja entregue atrasada é calculado o tempo do atraso.

4.3.5. DASHBOARD

O *dashboard* é direccionada ao supervisor das operações e tem como objectivo proporcionar uma visão geral das operações.

Durante a demonstração e testes, apenas foram processadas duas encomendas, uma com entrega conseguida e outra com falha na entrega. Ambas distribuídas pela mesma transportadora. A Figura 61 apresenta o relatório diário das operações. É possível ver na tabela da direita que a Transportadora3 tem um total de 2 encomendas atribuídas e a distribuição das mesmas pelas diferentes etapas. Na tabela da esquerda é possível visualizar o acumulado das encomendas em cada etapa. Os gráficos da esquerda confirmam o já referido e presente nas tabelas, que 50% das encomendas se encontram entregues e os restantes 50% tiveram falha na entrega e para além disso que 100% das encomendas foram distribuídas pela Transportadora 3. O gráfico da direita apresenta o nível de serviço ao longo dos últimos 6 meses para todas as transportadoras. Nesta fase ainda não existe histórico de encomendas, justificando assim o nível de serviço ser de 0% até ao mês de Outubro. Como

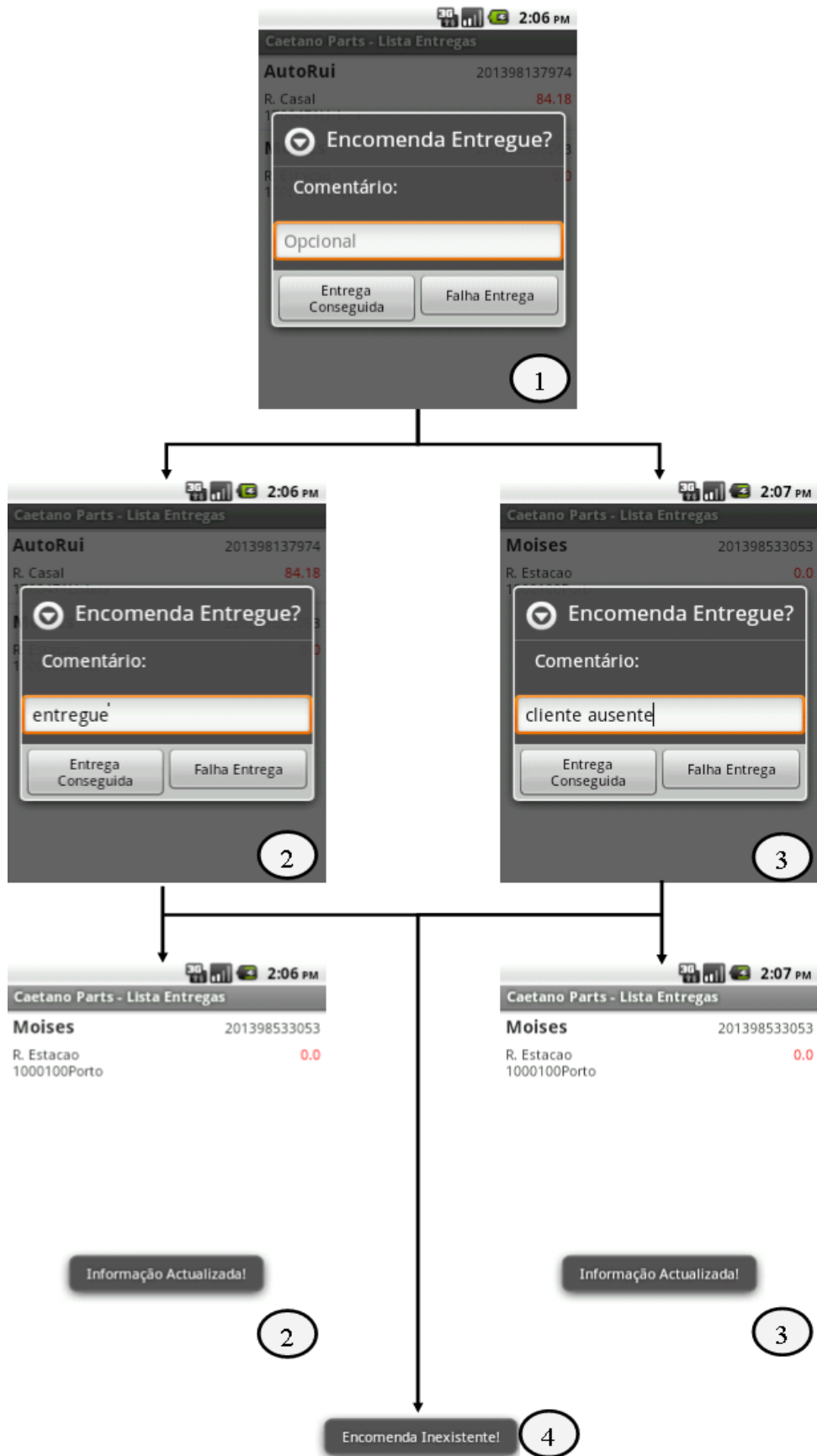


Figura 59 Janela de confirmação de entrega da encomenda através da aplicação móvel

contribuinte	cobrar	vcobrar	drecolha	idtransportadora	idmotorista	entrega	peso	volume	statusid	completo	ficheiro	comentario	dlimite	atraso	tatraso	coordenadas	entrega
503213519	TRUE	84.18	1382876854	300	manuel	1382882770	NULL	NULL	3	TRUE	IN_01310896	entregue	1382958000	FALSE	NULL	38.790507,-9.124629	5916.0
227494296	FALSE	15.95	1382876854	300	manuel	1382882860	NULL	NULL	4	FALSE	IN_01170170	cliente ausente	1382958000	NULL	NULL	41.152749,-8.622188	6006.0

Figura 60 Actualização tabela envios após interação da aplicação móvel

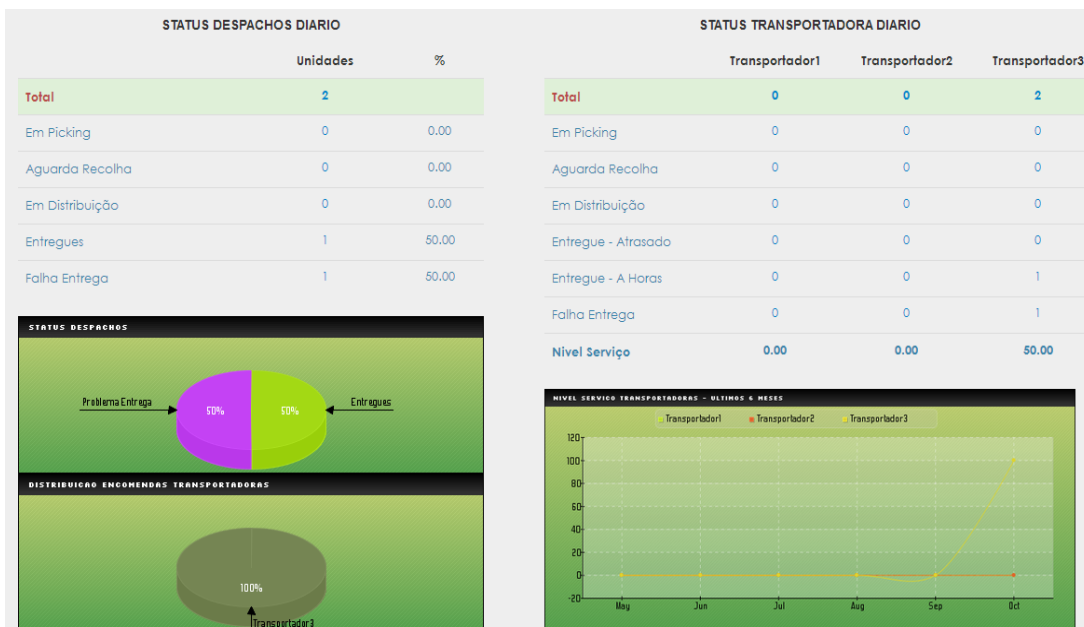


Figura 61 Dashboard com a informação dos dados da demonstração

o nível de serviço é obtido pela relação das encomendas entregues em atraso com as encomendas entregues totais, é possível verificar que o nível de serviço da Transportadora 3 no mês de Outubro é de 100%, pois pela simulação efectuada realizou uma entrega e sem atrasos.

É possível consultar em detalhe a que encomendas correspondem cada valor presente nas tabelas. A Figura 62 mostra a que encomendas se refere o valor 2 do total de encomendas da Transportadora3.

Código	Data & Hora Criação	Cliente	Data & Hora Recolha	Transportadora	Motorista	Data & Hora Entrega	V. Cobrar	Status	Comentário	Coordenadas
201398137974	27/10/2013 00:58:14	AutoRui	27/10/2013 12:27:34	Transportador3	Manuel	27/10/2013 14:06:10	84.18	Entregue	entregue	38.790507,-9.124629
201398533053	27/10/2013 00:38:04	Moises	27/10/2013 12:27:34	Transportador3	Manuel	27/10/2013 14:07:40		Falha na Entrega	cliente ausente	41.152749,-8.622188

Figura 62 Vista detalhada das encomendas da dashboard

Para possibilitar a visualização das variações que podem ocorrer nas tabelas e gráficos presentes na *dashboard* foram adicionados mais registros de encomendas ao sistema, distribuídas como apresentado na Tabela 21. A Figura 63 apresenta o resultado da *dashboard* com a adição das novas encomendas. A distribuição das encomendas introduzidas foi aleatória, tendo o cuidado que o nível de serviço de cada uma seja diferente das restantes a cada mês, possibilitando a identificação da variação do nível de serviço ao longo do tempo, como é possível ver na Figura 64. O nível de serviço apresentado, na tabela da direita, refere-se ao nível de serviço do espaço temporal analisado, neste caso um ano. Que seria o mesmo que visualizar para apenas seis meses pois é até onde se estende o espaço temporal das encomendas registadas.

Tabela 21 Dados encomendas demonstração *dashboard*

		TRANSPORTADOR1	TRANSPORTADOR2	TRANSPORTADOR3
MAIO	Entrega a Horas	1	1	2
	Entrega Atrasada	1	2	0
JUNHO	Entrega a Horas	2	1	3
	Entrega Atrasada	1	1	1
JULHO	Entrega a Horas	1	2	2
	Entrega Atrasada	2	1	0
AGOSTO	Entrega a Horas	1	3	2
	Entrega Atrasada	1	1	1
SETEMBRO	Entrega a Horas	2	3	3
	Entrega Atrasada	1	0	1
OUTUBRO	Em Picking	2	0	2
	Aguarda Recolha	1	1	1
	Em Distribuição	0	1	0
	Entrega a Horas	0	3	5
	Entrega Atrasada	4	2	0
	Falha Entrega	2	0	0

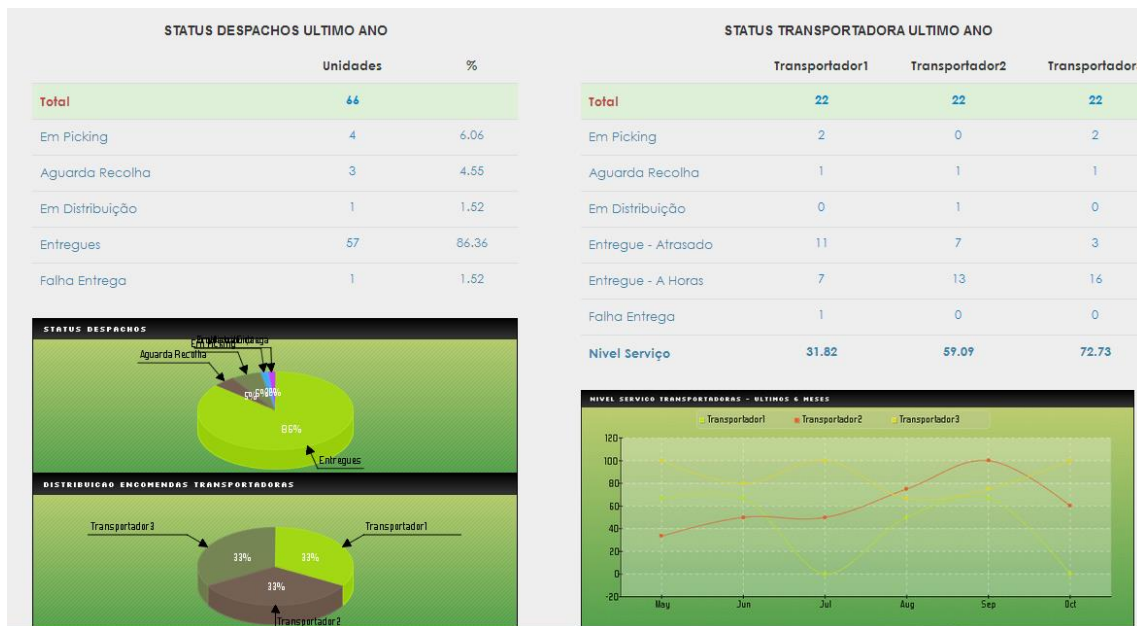


Figura 63 Dashboard após adição das encomendas

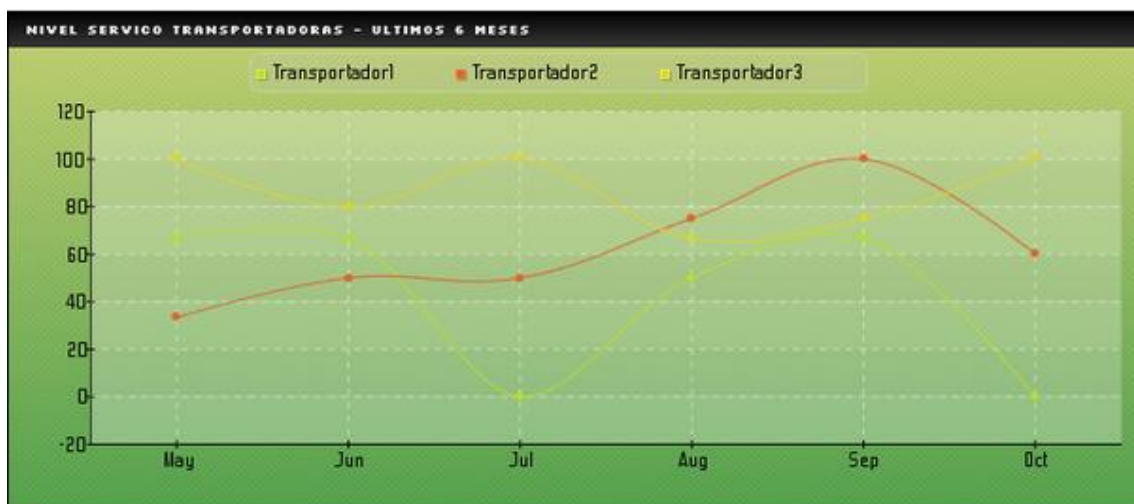


Figura 64 Vista pormenorizada gráfico evolução nível serviço das transportadoras nos últimos seis meses

5. CONCLUSÕES

Neste capítulo são avaliados os resultados alcançados em relação aos objectivos propostos e são identificadas limitações e apontadas sugestões de desenvolvimento futuro.

5.1. INTRODUÇÃO

O principal objectivo passava pelo desenvolvimento de uma ferramenta, utilizando recursos *open-source*, que permitisse acompanhar todas as etapas de uma encomenda e que servisse de base para avaliar as operações decorrentes. O protótipo gerado permite ao utilizador a consulta do estado das encomendas em qualquer altura, desde a colocação por parte do cliente até à entrega do mesmo, tal como a consulta dos níveis de serviço das operações de distribuição das mesmas, com recurso a uma aplicação *web*, a uma ferramenta para dispositivo móvel *Android* e a uma unidade de processamento em execução em *background*.

Trata-se de um sistema distribuído repartido em *front-end* e *back-end*. O *front-end* inclui as aplicações de interface com o utilizador, a aplicação móvel para dispositivos *Android* e a aplicação *web*. O *back-end* é constituído pela unidade de armazenamento e pelo serviço *web* de processamento de dados, garantindo interoperabilidade, disponibilidade e padronização.

A ferramenta *Android* mostrou ser adequada no processamento dos dados e na interface que oferece ao utilizador. Esta aplicação necessita de uma ligação à internet e recorre à informação proveniente do sensor de GPS para determinar a sua posição. A aplicação *web* mostrou-se capaz de fazer a ponte entre o processamento e armazenamento dos dados que é realizado em *background* e o utilizador.

Os testes e depurações efectuadas ao longo do desenvolvimento do sistema às funcionalidades tanto do *back-end* como do *front-end* possibilitou confirmar o correcto funcionamento dos diferentes módulos, tal como demonstrado pelo que se pode considerar que os objectivos propostos, no início do projecto, foram atingidos.

5.2. LIMITAÇÕES E DESENVOLVIMENTOS FUTUROS

Sendo o sistema desenvolvido um protótipo que prevê colmatar uma necessidade sem soluções no mercado é necessário realizar testes piloto mais exigentes que permitam identificar limitações da ferramenta. No entanto, são propostos desenvolvimentos futuros com vista a complementar as funcionalidades implementadas e a ultrapassar as limitações:

1. Aplicação móvel:

- Funcionalidade para assegurar a correcta transacção de dados, quando o acesso à internet está indisponível.
- Funcionalidade para colmatar a falha na recolha das coordenadas geográficas, quando não há sinal GPS.
- Inclusão de um leitor de código de barras, para a leitura do código de barras da etiqueta. Funcionando por um lado como garantia de que o responsável da entrega está na presença da encomenda no momento de submissão dos dados e como alternativa à identificação da encomenda sem necessidade de pesquisar na lista.
- Criação de filtros que permitam visualizar encomendas já entregues.
- Criação de alarmes de encomendas em atraso e encomendas prioritárias.

2. Aplicação *web*:

- Desenvolvimento da *dashboard*, assegurando a apresentação de dados das operações mais detalhados, ao nível da região e do cliente.
- Criação de alertas para tentativas de manipulação de resultados.

3. *Layout*:

- Melhorias ao nível do aspecto visual e da navegação da aplicação *web* e aplicação móvel e uniformização do *layout* das duas aplicações.

4. Gestão da informação:

- Criação do histórico de estados da encomenda.
- Modelo para selecção da transportadora mais adequada à distribuição de cada encomenda.

Referências Documentais

- Aaker, D.A. & Day, G.S., 1980. *Marketing Research: Private and Public Sector Decisions* (Toronto: Wiley).
- Abraham, K.G. & Taylor, S.K., 1993. *Firms' use of outside contractors: Theory and evidence*,
- ACEA, 2011. *The Automobile Industry Pocket Guide 2011.* , p.74.
- Aimi, G., 2007. Logistics Outsourcing: What it Takes to Succeed. *Supply Chain Management Review*, 11(8), p.13.
- Anchor, 2010. MySQL vs. PostgreSQL comparison. Available at: http://www.anchor.com.au/hosting/dedicated/mysql_vs_postgres [Accessed July 24, 2013].
- Andersson, D. & Norrman, A., 2002. Procurement of logistics services—a minutes work or a multi-year project? *European Journal of Purchasing and Supply Management*, (8), pp.3–14.
- Android, O., 2013. Android Activities. Available at: <http://developer.android.com/guide/components/activities.html> [Accessed September 24, 2013].
- ANECRA, 2013. Quotas de Mercado – Top Ten. Available at: <http://www.anecra.pt/gabecono/pdf/quotasTT.pdf>.
- Apte, U.M. & Viswanathan, S., 2000. Effective cross docking for improving distribution efficiencies. *International Journal of Logistics*, 3(3), pp.291–302.
- Arnold, K., Gosling, J. & Holmes, D., 2000. *The Java programming language*, Addison-wesley Reading, MA.
- Badros, G.J., 2013. JavaML: A Markup Language for Java Source Code. Available at: <ftp://ftp.cs.washington.edu/homes/gjb/papers/javaml/javaml.html#REF-Java>.
- Bardhan, I., Mithas, S. & Lin, S., 2007. Performance Impacts of Strategy , Information Technology Applications , and Business Manufacturing Plants. , 16(6), pp.747–762.
- Basílio, D., 2013. O Observador Cetelem 2013. Available at: www.oobservador.pt.
- Bhatnagar, R. & Viswanathan, S., 2000. Re-engineering Global Supply Chains - Alliances between Manufacturing Firms and Global Logistics Services Providers. *International Journal of Physical Distribution & Logistics Management*, 30(1), pp.13–34.

- Booth, D. et al., 2004. Web Services Architecture. W3C. Available at:
<http://www.w3.org/TR/ws-arch/#introduction>.
- Boyson, S. et al., 1999. Managing effective third party logistics relationships: what does it take? *Journal of Business Logistics*, 20, pp.73–100.
- Brooks, M.R., 2000. Performance evaluation of carriers by North American companies. *Transport Reviews: A Transnational Transdisciplinary Journal*, (January 2013).
- Brooks, M.R., 1995. Understanding the ocean container market—a seven country study. *Maritime Policy & Management*, 22(1), pp.39–49.
- Chang, Y.H., 1998. Logistical Management. *Hwa-Tai Bookstore Ltd., Taiwan*.
- Chow, G., Heaver, T.D. & Henriksson, L.E., 1994. Logistics Performance: Definition and Measurement. *International Journal of Physical Distribution & Logistics Management*, 24(1), pp.17–28.
- Christopher, M., 2005. *Logistics and Supply Chain management* 3^o ed., FT Prentice Hall.
- Commerce, U.S.D. of, 2009. *U . S . Automotive Parts Industry Annual Assessment*, Available at:
http://www.trade.gov/mas/manufacturing/oaai/build/groups/public/@tg_oaai/documents/webcontent/tg_oaai_003759.pdf.
- Conrad, T., 2004. PostgreSQL vs. MySQL vs. Commercial Databases: It's All About What You Need. *Jupitermedia Corporation*, pp.1–5. Available at:
<http://www.devx.com/dbzone/Article/20743>.
- Crockford, D., 2006. RFC 4627 - The application/json Media Type for JavaScript Object Notation. *JSON, IETF, Request for Comments*.
- Crum, M.R. & Allen, B.J., 1991. The changing Nature of the Motor Carriershipper Relationship: Implications for the Trucking Industry. *Transportation Journal*, 31, pp.41–54.
- Czerska, J., 2009. *Improving the stream of the value*,
- Deitel, H.M., 2001. *Xml Como Programar*, BOOKMAN COMPANHIA ED. Available at:
<http://www.google.pt/books?id=Q2VnzLtwR60C>.
- Dumond, E.J., 1991. Performance Measurement and Decision-making in a Purchasing Environment. *International Journal of Purchasing and Materials Management*, 27, pp.21–31.
- Etgar, M. & Fuchs, G., 2009. Why and how service quality perceptions impact consumer responses. *Managing Service Quality*, 19(4), pp.474–485.

- Fawcett, S. & Clinton, S.R., 1996. Enhancing logistics performance to improve the competitiveness of manufacturing organizations. *Production and Inventory Management Journal*, 37, pp.40–46.
- Fernie, J., 1999. Outsourcing Distribution in UK Retailing. , p.21.
- Fernihough, A. & Gyimesi, K., 2008. Performance in reserve. , p.20.
- Fielding, R. et al., 1999. RFC 2616. *Hypertext Transfer Protocol--HTTP/1.1*, 2(1), p.2.
- Fitzsimmons, J.A. & Fitzsimmons, M.J., 2006. *Service management: Operations, strategy, and information technology*, McGraw-Hill New York.
- Freeman, E. & Robson, E., 2011. *Head First HTML5 Programming: Building Web Apps with JavaScript*, O'Reilly Media, Incorporated.
- Gamobar Peças Online, 2013. Gamobar Peças Online. Available at: <http://www.grupogamobar.pt/pecas.asp> [Accessed February 14, 2013].
- Gao, J. et al., 2009. A 2D barcode-based mobile payment system. In *Multimedia and Ubiquitous Engineering, 2009. MUE'09. Third International Conference on*. pp. 320–329.
- Gay, C.L. & Essinger, J., 2002. Inside Outsourcing.
- Gibson, B.J., Sink, H.L. & Mundy, R.A., 1993. Shipper-Carrier Relationships and Carrier Selection Criteria. *Logistics and Transportation Review*, 29(4), pp.371–382.
- Gibson-odgers, P. et al., 2008. The World of Customer Service. , p.16.
- Goles, T. & Chin, W.W., 2002. Relational exchange theory and IS outsourcing: developing a scale to measure relationship factors. *Information systems outsourcing*, pp.221–250.
- Group, P., 2013. What can PHP do? Available at: <http://pt1.php.net/manual/en/intro-whatcando.php> [Accessed September 20, 2013].
- Grover, V., Cheon, M.J. & Teng, J.T.C., 1996. The effect of service quality and partnership on the outsourcing of information systems functions. *Journal of Management Information Systems*, pp.89–116.
- Hannon, D., 2008. Vesuvius makes the most of its 3PL implementation.
- Harry L. Sink & C. John Langley, J., 1997. A Managerial Framework for the Acquisition of Third-Party Logistics Services. *Journal of Business Logistics*, 18(2), pp.163–190.
- Ho, G. & Lo, V., 2005. China Auto Conference Call “Value Migration in China’s Car Sector: Winning in the After-market.” In Bear Stearns Equity Research, p. 8. Available at: http://www.oliverwyman.com/pdf_files/Auto_China_Winning_Aftermarket.pdf.

- Hors, A. Le & Jacobs, I., 1999. *HTML 4 . 01 Specification*,
- INE, 2011. *Estatísticas do Comércio 2011*,
- Jharkharia, S. & Shankar, R., 2007. Selection of logistics service provider: An analytic network process (ANP) approach. *Omega*, 35(3), pp.274–289. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S0305048305000897> [Accessed December 3, 2012].
- Jiang, B., Frazier, G. V & Prater, E.L., 2006. Outsourcing effects on firms' operational performance: An empirical study. *International Journal of Operations & Production Management*, 26(12), pp.1280–1300.
- Kakabadse, N. & Kakabadse, A., 2000. Critical review--Outsourcing: a paradigm shift. *Journal of Management Development*, 19(8), pp.670–728.
- Kearney, A.T.I., 1985. *Measuring and Improving Productivity in Physical Distribution* (Chicago: Council of Logistics Management).
- Kenyon, G.N. & Meixell, M.J., 2011. Success factors and cost management strategies for logistics outsourcing. *Journal of Management and Marketing Research*, pp.1–17.
- Kleinsorge, I.K., Chary, P.B. & Tanner, R.D., 1991. The Shipper-Carrier Partnership: A New Tool for Performance Evaluation. *Journal of Business Logistics*, 12.
- Van Laarhoven, P., Berglund, M. & Peters, M., 2000. Third-party logistics in Europe--five years later. *International Journal of Physical Distribution & Logistics Management*, 30(5), pp.425–442.
- Lacity, M.C. & Willcocks, L.P., 1998. An empirical investigation of information technology sourcing practices: Lessons from experience. *MIS quarterly*, pp.363–408.
- Langley, C., Allen, G. & Tyndall, G., 2003. Third-party logistics study 2003: results and findings of the eighth annual study.
- Lau, K.H. & Zhang, J., 2006. Drivers and obstacles of outsourcing practices in China. *International Journal of Physical Distribution & Logistics Management*, 36(10), pp.776–792. Available at: <http://www.emeraldinsight.com/10.1108/09600030610714599>.
- Lieb, R., 2008. The North American third-party logistics industry in 2007: the provider CEO perspective. *Transportation Journal*, 47(2), pp.39–53.
- Lieb, R. & Bentz, B.A., 2004. *The Use of Third-Party Logistics Services by Large American Manufacturers : The 2004 Survey*.
- Lieb, R.C. & Millen, R.A., 1988. JIT and corporate transportation requirements. *Transportation Journal*, 27, pp.5–10.

- Lieb, R.C. & Millen, R.A., 1990. The Responses of General Commodity Motor Carriers to Just-In-Time Manufacturing Programs. *Transportation Journal*, 30, pp.5–11.
- Linder, J.C., 2004. Transformational outsourcing. *MIT Sloan Management Review*, 45(2), pp.52–58.
- Llera, G. de, 2009. Situação do Após Venda.
- Logistics, P., 1999. Survey Finds Bottom Line Drives Outsourcing. *Logistics Management & Distribution Report*, May, p.24.
- Lynch, C., 2000. Logistics outsourcing: a management guide. *Council of Logistics Management Publications*.
- Mason, S.J. et al., 2002. Improving electronics manufacturing supply chain agility through outsourcing. *International Journal of Physical Distribution & Logistics Management*, 32(7), pp.610–620. Available at: <http://www.emeraldinsight.com/10.1108/09600030210442612> [Accessed March 14, 2013].
- Mccarthy, I. & Anagnostou, A., 2004. The impact of outsourcing on the transaction costs and boundaries of manufacturing. , 88, pp.61–71.
- MCoutinho, G., 2013. MCoutinho Peças. Available at: http://www.mcoutinhopecas.pt/getfile.php?xp=2&src=file16_pt&ext=pdf.
- Meade, L. & Sarkis, J., 1998. Strategic analysis of logistics and supply chain management systems using the analytical network process. *Transportation Research Part E: Logistics and Transportation Review*, 34(3), pp.201–215.
- Mentzer, J.T. & Konrad, B.P., 1991. AN EFFICIENCY EFFECTIVENESS APPROACH TO LOGISTICS PERFORMANCE ANALYSIS. *Journal of Business Logistics*, 12, pp.33–62.
- Michlowicz, E., 2012. Zarys logistyki przedsiębiorstwa. *Wydawnictwa AGH*.
- Monson-Haefel, R., 2004. *J2EE Web services*, Addison-Wesley Professional.
- Ohbuchi, E., 2004. Barcode Readers using the Camera Device in Mobile Phones.
- Online, Auto-Industrial, 2013. Auto-Industrial Online. Available at: <http://www.auto-industrial.pt/> [Accessed February 14, 2013].
- Online, Cloud Logistics, 2013. Cloud Logistics Online. Available at: <http://www.gocloudlogistics.com/> [Accessed March 20, 2013].
- Online, Caetano Parts, 2013. Caetano Parts. Available at: <http://www.caetanoparts.pt/> [Accessed January 23, 2013].

- Online, Grupo Salvador Caetano, 2013. Grupo Salvador Caetano. Available at: <http://www.gruposalvadorcaetano.pt/> [Accessed January 23, 2013].
- Online, JSON, 2013. Introducing JSON. Available at: <http://www.json.org/> [Accessed October 12, 2013].
- Online, JAP, 2013. JAP Online. Available at: <http://www.grupojap.pt> [Accessed February 14, 2013].
- Online, PostgreSQL, 2013. PostgreSQL About. Available at: <http://www.postgresql.org/about/> [Accessed July 22, 2013].
- Online, Precision, 2013. Precision Online. Available at: <http://www.precisionsoftware.com> [Accessed March 26, 2013].
- Online, Smart Freight Ware, 2013. Smart Freight Ware Online. Available at: <http://www.smartfreightware.com/> [Accessed March 20, 2013].
- Online, Santogal Peças, 2013. Santogal Peças Online. Available at: <http://www.santogal.pt/Pecas.aspx> [Accessed February 14, 2013].
- Pautasso, C., 2008. REST vs. SOAP: Making the Right Architectural Decision. , p.43. Available at: <http://www.jopera.org/files/soa-amsterdam-restws-pautasso-talk.pdf>.
- Pawlan, M., 2000. Essentials of the Java™ Programming Language: A Hands-On Guide. Available at: <http://itvd.files.wordpress.com/2009/04/ebook-pdf-java-programming-language-basics.pdf>.
- PHP Group, 2013. What is PHP? Available at: <http://pt1.php.net/manual/en/intro-what-is.php> [Accessed September 20, 2013].
- Razzaque, M.A. & Sheng, C.C., 1998. Outsourcing of logistics functions : a literature survey. , 28(2), pp.89–107.
- Rhea, M.J. & Shrock, D.L., 1987. Measuring the effectiveness of physical distribution customer service programs. *Journal of Business Logistics*, 8(1).
- Rodriguez, A., 2008. RESTful Web services : The basics. , pp.1–11. Available at: <http://www.ibm.com/developerworks/webservices/library/ws-restful/ws-restful-pdf.pdf>.
- Rushton, A., Oxley, J. & Croucher, P., 2010. *Handbook of Logistics and Distribution Management* 2º ed., Kogan Page.
- Sanchez, I., 2013. Future Scenario Planning in Brazilian Logistics and Transportation Sector. , pp.1–105.
- Schottmuller, A., 2011. Top 14 Things Marketers Need to Know About QR Codes. Available at: <http://searchenginewatch.com/article/2066777/Top-14-Things-Marketers-Need-to-Know-About-QR-Codes> [Accessed August 7, 2013].

- Shahin, A., 2011. Estimation of Customer Dissatisfaction Based on Service Quality Gaps by Correlation and Regression Analysis in a Travel Agency. , 6(3), pp.99–108.
- Skjoett-Larsen, T., 2000. Third party logistics--from an interorganizational point of view. *International Journal of Physical Distribution & Logistics Management*, 30(2), pp.112–127.
- Sofrapa Online, 2013. Sofrapa Online. Available at: <http://www.sofrapa.pt> [Accessed February 14, 2013].
- Somera, G., 2006. *Treinamento Profissional em Java*, Universo dos Livros Editora LTDA.
- SQLite, O., 2013. About SQLite. Available at: <http://www.sqlite.org/about.html> [Accessed September 30, 2013].
- Stats, G., 2013. Stat Counter - Global Stats. Available at: <http://gs.statcounter.com/> [Accessed September 20, 2013].
- Szymankiewicz, J., 1994. Contracting Out or Selling Out?: Survey into the Current Issues Concerning the Outsourcing of Distribution. *Logistics Information Management*, 7(1), pp.28–35.
- Szymankiewicz, J., 1996. The changing role of third party logistics.
- Tseng, Y., 2005. The role of transportation in logistics chain. , 5, pp.1657–1672.
- W3Schools, 2013. JSON. Available at: <http://www.w3schools.com/json/> [Accessed October 12, 2013].
- Wilding, R., 2004. CUSTOMER PERCEPTIONS ON LOGISTICS OUTSOURCING IN THE EUROPEAN Authors : Faculty of Economics and Business Administration , CUSTOMER PERCEPTIONS ON LOGISTICS OUTSOURCING IN THE EUROPEAN. , 34(8).
- Zwolinska, B., 2012. Key Performance Indicators (KPI) of Transport Services Assessment. *Carpathian Logistics Congress*, pp.7–12.

Anexo A. Tecnologias desenvolvimento do *back-end*

O *back-end* pode ser visto como o lado do servidor, ou seja, o processamento remoto de uma aplicação. Neste projecto podemos considerar o *back-end* constituído pela base de dados, pelo serviço *web*, pela unidade de recolha e processamento dos dados e pelas aplicações que são executadas sem interacção com o utilizador, como a geração do código de barras e a recolha dos dados das facturas

1. *Base de Dados*

As bases de dados (BD) são uma característica presente na maioria das arquitecturas de *software*. Algumas BD como *Oracle*, *Microsoft SQL Server*, *Microsoft Access* e a *IBM DB2* são muito populares mas fazem parte do grupo de bases de dados comerciais. Nas últimas duas décadas a comunidade *open-source* tem aumentado a qualidade dos suas ferramentas computacionais, fazendo dele mais apetecível para os empreendedores. Nesse sentido assistiu-se a uma vontade de migrar do *software* comercial para o *software open-source*. Aparecendo na liderança deste segmento as BD *PostgreSQL* e a *MySQL*.

A comparação entre as BD tem de ser feita através das características que cada uma apresenta (Conrad 2004). As principais características a considerar na comparação das BD são apresentadas a seguir:

1. Armazenamento de dados

MySQL tem vários mecanismos de armazenamento de dados disponíveis. Originalmente era usado o *ISAM/MyISAM*, um mecanismo leve e rápido mas com falta de alguns recursos e sem transacções seguras de dados, este mecanismo foi depois substituído por um mecanismo mais avançado, o *InnoDB*. O *InnoDB* diz-se ser um mecanismo com certificado ACID e que oferece recursos como o bloqueio ao nível da linha e recuperação após falhas. Devido a ter um número superior de recursos o *InnoDB* é mais lento e pesado que o *MyISAM*, mas essa característica é perfeitamente aceite pelos utilizadores. O *MySQL* disponibiliza outros mecanismos de armazenamento, mas o *InnoDB* é o principal. Como os mecanismos podem apresentar recursos diferentes entre si, no momento da escolha deve-se ter em consideração os recursos necessários.

O *PostgreSQL* utiliza apenas um mecanismo de armazenamento, denominado por *Postgres Storage System*.

2. Integridade dos dados

Uma das características chave de qualquer DB é a integridade dos dados (Conrad 2004). O certificado ACID (Atomicidade, Consistência, Isolamento e Durabilidade) é a qualificação utilizada para assegurar a integridade dos dados durante as transacções.

Atomicidade significa que todas as tarefas numa transacção (uma única instrução ou um conjunto de instruções) devem ser executadas em caso de sucesso ou nenhuma deve ser executada em caso de falha. Ou seja, após o final da transacção a base de dados não deve mostrar resultados parciais da transacção (Anchor 2010). Um exemplo, pode ser uma transferência de dinheiro entre dois bancos. Se um banco é creditado, é fundamental que o outro seja debitado.

Consistência diz respeito às regras de integridade dos dados, como o tipo de dados de uma coluna, a indicação de um intervalo de valores da coluna ou a singularidade de chaves (Anchor 2010). Por exemplo, quando numa BD existe uma chave primária, então qualquer inserção ou alteração de dados não deve duplicar essa coluna.

Isolamento significa que os eventos decorrentes de uma transacção são isolados de outras transacções que não estejam concluídas (Anchor 2010). Ou seja, várias transacções podem tentar aceder ao mesmo registo da BD em certo momento. O isolamento evita que transacções paralelas interfiram umas nas outras, obtendo-se assim em transacções paralelas o mesmo resultado que se obtinha se as transacções ocorressem em simultâneo.

Durabilidade garante que quando uma transacção é concluída os resultados não são perdidos (Anchor 2010) mesmo em presença de falhas.

Ambas suportam recuperações após transacções falhadas, isto é são capazes de retornar a uma versão da base de dados estável. *MySQL* utiliza o sistema de bloqueio da linha durante as transacções. *PostgreSQL* utiliza o método *Multi-Version Concurrency Control (MVCC)*, com este método quando uma transacção é efectuada

é tirada uma “foto” a todos os dados. Versões mais recentes do *PostgreSQL* suportam o bloqueio da linha como opção mas o *MVCC* continua a ser o método mais utilizado.

3. Tipos de dados

Tanto o *MySQL* e o *PostgreSQL* suportam os tipos de dados mais comuns. *PostgreSQL* suporta tipos de dados definidos pelo utilizador. Ambos suportam dados do tipo geográfico, conhecido como *GIS*, para além disso o *PostgreSQL* suporta dados de reconhecimento de redes, reconhecendo Ipv4 e Ipv6.

4. Replicação

A replicação é uma das características mais importantes de uma BD. Isto consiste na partilha da informação para sistemas redundantes, para melhorar a fiabilidade e acessibilidade à informação. Tanto o *MySQL* como o *PostgreSQL* tem suporte para vários tipos de cenário de replicação.

5. Plataforma de execução

O *MySQL* e o *PostgreSQL* são compatíveis com diferentes plataformas, incluindo *Windows*, *Linux*, *FreeBSD* e *MacOSX*. O *MySQL* usa um modelo de *threads*, onde todos os utilizadores se ligam a um único *daemon* para acesso à base de dados. O *PostgreSQL* não utiliza um modelo de *threads* e por cada nova ligação é criado um novo processo.

6. Interface

PostgreSQL e o *MySQL* suportam *ODBC* e *JDBC* para ligação à rede e também os métodos nativos para estabelecer ligações. Outra característica importante da interface das BD é o método de autenticação.

O *MySQL* utiliza um método simples, todas as informações de autenticação são guardadas dentro de uma tabela. Quando o utilizador tenta aceder à BD, as credenciais digitadas são comparadas com as que constam na tabela, verificando de onde o utilizador está a aceder e a que recursos têm acesso.

O *PostgreSQL* pode utilizar um método semelhante, mas também incorpora outros métodos de autenticação distintos. Por exemplo, pode ser utilizado um ficheiro externo ao qual a BD acede e define que utilizador tem acesso a que base de dados e recursos. Pode ser também utilizada a autenticação local, por exemplo quando um

utilizador entra no sistema utilizando uma palavra-chave, esse *login* dá acesso também à BD.

Também se pode aceder através de métodos de programação. Ambas as BD suportam o acesso através de *C/C++*, *Java*, *Perl*, *Python* e *PHP*.

7. Backups

Em relação aos *backups* as BD *open-source* não apresentam as melhores soluções. Ambas têm *scripts* que facilitam a criação de um *backup* dos dados da base de dados e do seu esquema. Existem agentes externos que providenciam soluções para a mesma função mas os resultados nem sempre são satisfatórios.

8. Importação de Dados

Ambas as BD têm ferramentas que auxiliam na importação dos dados provenientes de BD comerciais. Estas ferramentas são disponibilizadas por desenvolvedores externos e podem ser comerciais ou *open-source*. Para além disso, o *PostgreSQL* incorpora um recurso que permite a importação dos dados da *Oracle* e da *MySQL*. Notoriamente, quanto mais complexo o esquema da base de dados, mais difícil é a conversão para o modelo da *PostgreSQL*, pode até nem ser tudo importado na perfeição.

POSTGRESQL

PostgreSQL é uma BD *open-source* objecto-relacional. Com mais de 15 anos de desenvolvimento esta arquitectura ganhou grande relevo devido à fiabilidade e correcta integridade dos dados. É compatível com os principais sistemas operativos e é totalmente aprovada pelo certificado ACID. A BD inclui a maioria dos tipos de dados SQL:2008, incluindo *integer*, *numeric*, *boolean*, *char*, *varchar*, *date*, *interval* e *timestamp*, para além disto suporta dados de imagem, som e vídeo. A BD apresenta interfaces nativas com as linguagens *C/C++*, *Java*, *.Net*, *Perl*, *Python*, *Ruby*, *Tcl*, *ODBC*, entre outros (PostgreSQL Online 2013).

PostgreSQL apresenta bons processos de recuperação e eliminação de falhas, como o *MVCC*, pontos de restauro, replicação assíncrona e optimizador de *query*. É altamente

escalável, tanto na quantidade de dados que pode gerir e o número de usuários que pode conter. Alguns limites do PostgreSQL estão apresentados na Tabela 22.

MYSQL COMMUNITY

MySQL Community é a versão grátis da mais popular base de dados *open-source*, sob a licença GPL e que conta com uma comunidade grande de desenvolvedores e utilizadores. Conta com diferentes estruturas de dados como *InnoDB*, *MyISAM*, *CSV* e outros e é compatível com os principais sistemas operativos. Oferece interface para comunicação através de diferentes linguagens (*ODBC*, *JDBC*, *.NET*, etc.). Apresenta elevada performance no seu desempenho e tira partido do sistema de replicação e de partição implementados.

Para além disso tem uma interface visual própria, para desenvolvimento e administração bastante consistente e intuitiva.

POSTGRESQL VS MYSQL

A Tabela 23 sintetiza algumas das vantagens e desvantagens das duas principais soluções *open-source*.

Para o presente desenvolvimento optou-se pela BD *PostgreSQL* pois apresenta limites de armazenamento suficientes, pode ser executado nos sistemas operativos principais e pode ser utilizada livremente, independente do fim a que se destina. O *MySQL* é mais limitado no que à utilização livre diz respeito, definindo-a através da máxima “*Free use for those who never copy, modify or distribute*” (Em Português: Uso livre para aqueles que nunca copiem, modifiquem ou distribuam).

Em relação às características discutidas acima, o *PostgreSQL* comparativamente ao *MySQL*, apresentou argumentos equiparáveis ou superiores. Especialmente no campo da fiabilidade

Tabela 22 Limites PostgreSQL (Adaptado de PostgreSQL Online 2013)

<i>Limite</i>	<i>Valor</i>
<i>Dimensão máxima da base de dados</i>	Ilimitado
<i>Dimensão máxima da tabela</i>	32 TB
<i>Dimensão máxima da linha</i>	1,6 TB
<i>Dimensão máxima do campo</i>	1 GB
<i>Número máximo de linhas por tabela</i>	Ilimitado
<i>Número máximo de colunas por tabela</i>	250-1600, dependendo do tipo de dados da coluna
<i>Número máximo de índices por tabela</i>	Ilimitado

Tabela 23 Prós e Contras das Base de Dados PostgreSQL e MySQL (Adaptado de Anchor 2010)

	PostgreSQL	MySQL
Prós	Rico em ferramentas: <ul style="list-style-type: none"> • Capacidade de utilizar dados GIS; • Pesquisa flexível em <i>full-text</i>; • Diversos métodos de replicação; • Linguagem <i>server-side</i> poderosa (PL/pgSQL); • Definição de dados pelo utilizador; Método MVCC; Certificado ACID; Versão comercial disponibilizada por terceiros; Bem documentado; Acesso por interfaces gráficas consistente;	Mais leve (menos recursos necessários); Versão comercial disponível; Muito utilizada, o que leva a haver muitos desenvolvimentos disponíveis; Performance elevada, em <i>queries</i> SELECT e especialmente em tabelas MyISAM;
Contras	Método de replicação menos desenvolvido;	MyISAM não suporta transacções e não preenche completamente o certificado ACID (Possibilidade de utilizar tabelas InnoDB que permitem transacções e já são mais consistentes); Por defeito as tabelas MyISAM estão seleccionadas; Tabelas MyISAM apenas suportam bloqueio à tabela; Má performance quando utilizadas subqueries;

e da recuperação de erros, preenchendo os requisitos ACID e tendo métodos de recuperação de erros bem desenvolvidos. Acrescentando a estes argumentos, dois factores importantes são a capacidade de a BD suportar a maioria dos dados e estar nutrida de interfaces que permitem a comunicação através de várias linguagens.

2. *Serviços Web*

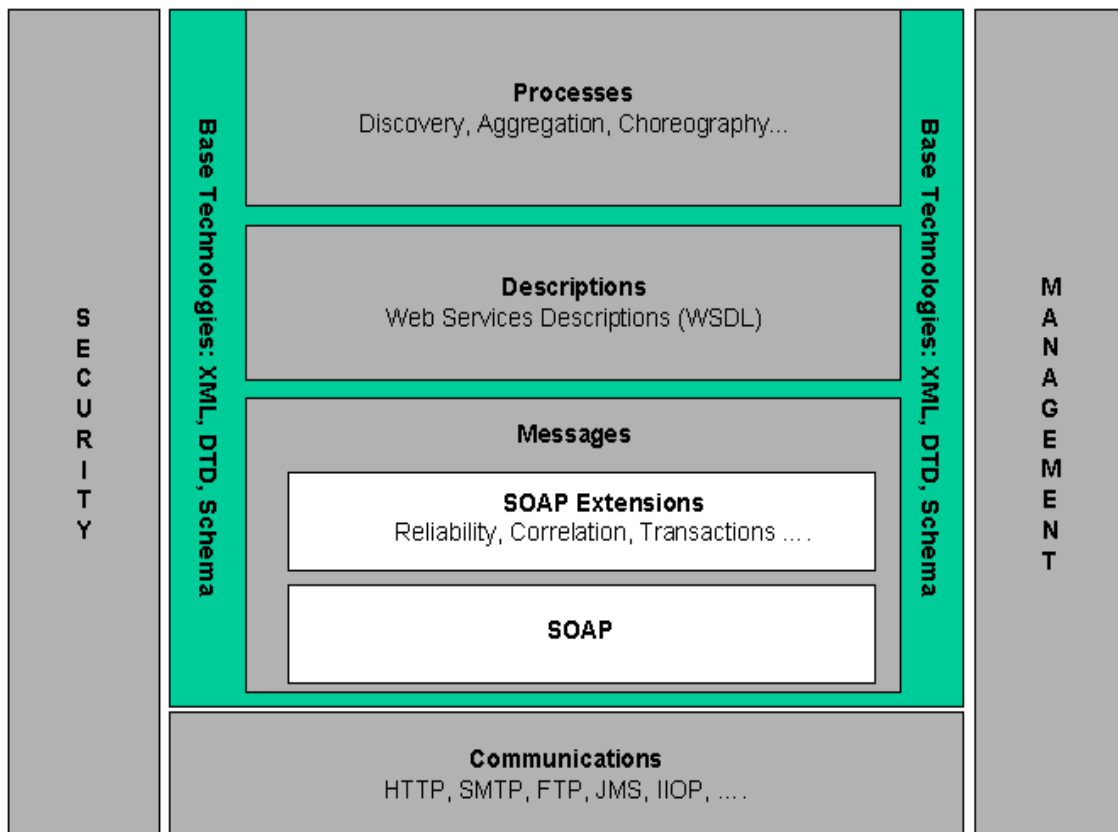
Os serviços *web* proporcionam a interoperabilidade entre aplicações diferentes, que são executadas em várias plataformas e sistemas (Monson-Haefel 2004; Booth et al. 2004). Esta propriedade provém do facto dos serviços *web* serem independentes de qualquer linguagem de programação, sistema operativo e plataforma (Monson-Haefel 2004).

Os serviços *web* são identificados por um *Uniform Resource Identifier* (URI), descritos usando *XML*. A arquitectura destes serviços envolve diferentes tecnologias em camadas

inter-relacionadas, sendo a linguagem *XML* a base para essas tecnologias: (i) protocolo de troca de mensagens *Simple Object Access Protocol (SOAP)*; (ii) descrição dos serviços através de *Web Services Description Language (WSDL)*; (iii) publicação e pesquisa de serviços *web* através do protocolo *Universal Description, Discovery and Integration (UDDI)* (Booth et al. 2004). A Figura 65 apresenta a arquitetura de um serviço *web*, de uma forma simples. A linguagem *XML* é utilizada para anotação dos dados, o protocolo *SOAP* para a transferência de dados, o *WSDL* é usado para descrever os serviços e o *UDDI* para “listar” os serviços disponíveis.

REPRESENTATIONAL STATE TRANSFER WEB SERVICE

Representational State Transfer (REST) web service é um simples serviço *web* implementado através de *HTTP* e os princípios do *REST*. Este serviço ganhou aceitação por ser uma alternativa mais simples que o *SOAP* e a *Web Services Description Language (WSDL)*. Uma das forças desta mudança foi provedores como a *Yahoo*, *Google* e *Facebook* terem também adotado este serviço, em detrimento do *SOAP* e *WSDL*.



Fonte: Booth et al. (2004)

Figura 65 Arquitetura da Stack dos serviços *web*

REST foi introduzido em 2000 por Roy Fielding na Universidade da Califórnia, mas com pouco impacto inicial. Se for feita uma análise actual ao número de serviços *web* que utilizam estes princípios, verifica-se de facto que estes têm vindo a aumentar e apresentam já um grande impacto no plano geral (Rodriguez 2008).

REST define um conjunto de princípios de arquitectura, pela qual é possível estruturar serviços *web* que se focam nos recursos do sistema, englobando, como as informações são endereçadas e transferidas através de *HTTP* por diferentes clientes e com linguagens diferentes. Em suma, uma arquitectura do estilo *REST* consiste numa comunicação entre clientes e servidores. No qual o cliente inicia a comunicação com pedidos ao servidor e este processa os pedidos e retorna as respostas, a Figura 66 mostra como ocorre a troca de dados entre cliente e servidor.

Rodriguez (2008) afirma que os serviços *web REST* assentam em quatro princípios fundamentais:

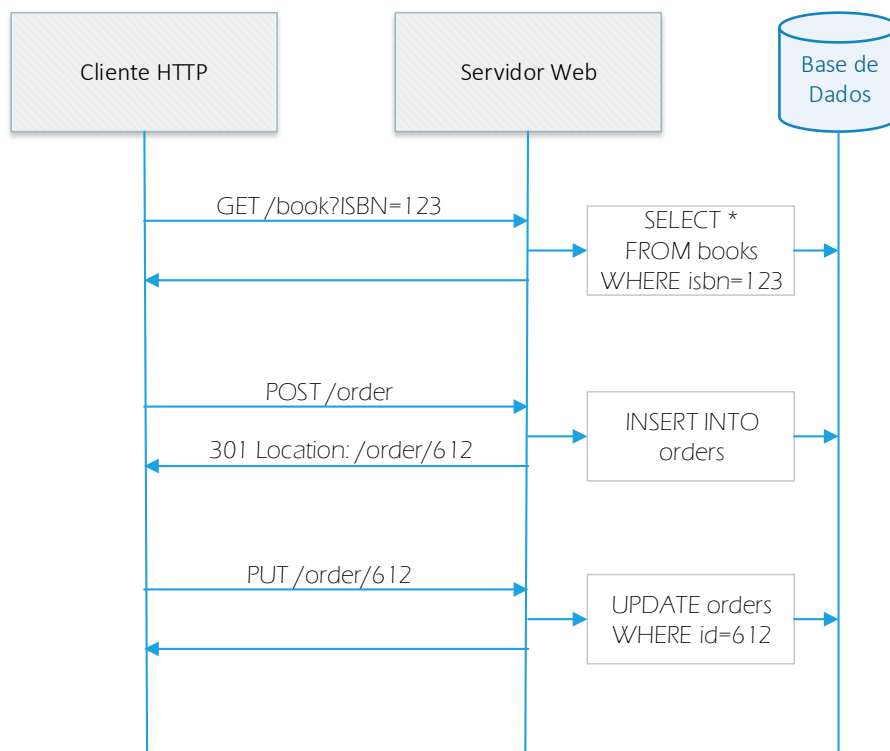


Figura 66 Exemplo *RESTful* Web Service (Adaptado de Pautasso 2008)

1. Uso de métodos HTTP

Uma das principais características deste serviço *web* é o uso dos métodos *HTTP* (*Post*, *Get*, *Put* e *Delete*) tal como definidos pelo protocolo RFC 2616. Ao fazer uso destas operações bem definidas, *REST* estabeleceu um princípio combinatório entre as operações *Create*, *Read*, *Update* e *Delete* (*CRUD*¹) e os métodos *HTTP*. Essa correspondência é realizada através dos seguintes princípios:

- Para Criar (*Create*) um recurso no servidor, utiliza-se o método *Post*.
- Para Ler (*Read*) um recurso, utiliza-se o método *Get*.
- Para Actualizar (*Update*) um recurso, utiliza-se o método *Put*.
- Para Eliminar (*Delete*) um recurso, utiliza-se o método *Delete*.

2. Protocolo Cliente/Servidor sem Estado

Um protocolo sem estado é um protocolo que trata cada pedido de forma independente, sendo que cada par de pedido/resposta não é influenciado por transacções anteriores.

Este tipo de protocolo simplifica o desenvolvimento do servidor, pois não necessita por exemplo de ser alocada memória dinamicamente para armazenar dados das contínuas conversações. Em contrapartida, é necessário incluir um maior número de informação em cada pedido, de forma ao servidor poder gerar uma resposta correcta.

Um serviço sem estado, não só demonstra um melhor desempenho, como transfere a responsabilidade de manter o estado para o lado do cliente. Num serviço *web REST*, o servidor tem como função gerar respostas e providenciar uma interface que possibilite o cliente de manter o estado de uma comunicação.

3. Endereço estruturado – como URIs

Do ponto de vista do cliente, as *URIs* vão definir o quão intuitivo um serviço *web REST* vai ser. As *URIs* dos serviços *REST* devem ser de tal forma intuitivas, que deve ser possível prevêê-las. As *URIs* devem ser pensadas para que não seja necessário dar

¹ *CRUD* (*Create*, *Read*, *Update* e *Delete*) define as quatro operações básicas utilizadas em bases de dados ou na interface com os utilizadores.

nenhuma explicação para se tornarem perceptíveis. Assim sendo, a estrutura de um *URI* deve ser simples, previsível e facilmente compreendida.

Uma boa forma de atingir esta estrutura é fazer-lo da mesma forma que se estrutura a localização de um ficheiro dentro de pastas. Este tipo de *URI* é hierárquica, com raiz num único caminho e com ramificação a partir dele. Assim sendo, esta organização pode ser vista de forma figurativa, como uma árvore à qual estão ligados os troncos e da qual se ramificam outros troncos. Tomando como exemplo, um serviço *web* de uma empresa na qual é possível consultar documentos, a *URI* para esse pedido deveria ser algo parecido com:

```
http://www.mycompany.org/documents/invoice/{number_invoice}
```

Neste exemplo, a raiz `/documents`, tem o caminho `/invoice` abaixo dela, que por sua vez tem os documentos que estão disponíveis para consulta.

Existem algumas directrizes, na construção da estrutura das *URIs* para um serviço *web REST*:

- Esconder a extensão da linguagem (*.php*, *.asp*, etc) utilizada do lado do servidor, para não ser necessário alterar a *URI*, caso se altere de linguagem;
- Manter todos os nomes em minúsculas;
- Substituir espaços por hífens e *underscores*;
- Evitar, sempre que possível, o uso de *query strings*;

As *URI* devem ser estáticas para que quando um recurso ou a implementação do serviço altere, o caminho continue a ser o mesmo.

4. Transferência XML, JavaScript Object Notation (JSON) ou ambos

A última consideração no desenvolvimento de um serviço *web* do tipo *REST* é ter em conta o formato da estrutura de dados durante a troca de informação de pedido/resposta ou do corpo *HTTP*. É nesta restrição que tudo se deve manter simples, legível e interligado.

Deve ser dado ao cliente a possibilidade de requerer os dados no formato que mais lhe convém, construindo o serviço de forma a que este incorpore um cabeçalho *HTTP Accept*, onde o cabeçalho é do tipo Multipurpose Internet Mail Extensions (*MIME*). Os tipos *MIME* mais comuns nestes serviços *web* são referidos na Tabela 24.

Tabela 24 Tipos MIME mais utilizados nos serviços *web* REST

<i>Tipos MIME</i>	<i>Conteúdo</i>
<i>JSON</i>	application/json
<i>XML</i>	application/xml
<i>XHTML</i>	application/xhtml+xml

Isto permite que o serviço possa ser utilizado por uma grande variedade de clientes escritos em linguagens diferentes e executados em diferentes plataformas. Dando uso aos tipo de estrutura de dados *MIME* e aos cabeçalhos *HTTP Accept*, é possível que os clientes possam escolher a estrutura de dados que melhor se adequa e aumentando assim a interoperabilidade entre aplicativos.

JSON

No desenvolvimento deste trabalho optou-se pela estrutura de dados *JavaScript Object Notation (JSON)* para troca de informação entre aplicação móvel e o *back-end*. *JSON* é um formato leve para troca de dados e de implementação e compreensão simples, tanto para a máquina como para o humano. Em relação ao seu “rival” *XML*, este tem como vantagens ser mais leve, rápido e mais fácil de analisar (W3Schools 2013). *JSON* é completamente independente de qualquer linguagem de programação, pois usa convenções de várias, sendo a ideal para a troca de dados. Os padrões para a estruturação do formato *JSON* constam no documento RFC 4627. Existem duas estruturas que o compõem:

- O conjunto de *name/value*. Denominado nas diferentes linguagens por *object*, *record*, *struct*, *dictionary*, *hash table*, *keyed list* ou *associative array*.
- Uma lista ordenada de valores. Conhecida nas linguagens como *array*, *vector*, *list* ou *sequence*.

Estas estruturas são comuns a todas as linguagens, tornando assim o *JSON* uma estrutura universal, capaz de ser lida e interpretada pelas diferentes linguagens.

Em *JSON* a apresentação de dados rege-se por uma estrutura bem definida e que segue as seguintes regras (JSON Online 2013):

- Um objecto é um conjunto de pares *name/value*. O objecto começa com “{” e termina com “}”. Cada *name* é seguido por “:” e os pares *name/value* são divididos por “,”;
- Um *array* é um conjunto de valores ordenados. O *array* começa com “[” e termina com “]”, os valores do *array* são separados por “,”;

- Uma string pode ser composta por nenhum ou um conjunto de caracteres entre aspas duplas.

A Figura 67 apresenta um exemplo de dados com formatação *JSON*.

3. *Código de Barras - Noções Básicas*

Um código de barras é uma representação gráfica de dados sobre o objecto à qual está anexado. Os códigos de barras tradicionais armazenam os dados a forma de linhas paralelas com diferentes larguras e são conhecidos como códigos de barra 1D ou lineares (Ver Figura 68). Os códigos de barras lineares codificam uma sequência de números e letras, fazendo variar as espessuras das barras e dos espaços entre elas. O leitor pode desta forma medir a intensidade da luz reflectida a partir das barras pretas e dos espaços em branco, assim é possível ler, recuperar, processar e validar os dados. Tem aumentado a utilização dos códigos de barras lineares, desde a sua primeira aparição. Hoje em dia são visíveis por todo lado e associados a todo o tipo de produtos e identificações, uma vez que este sistema é fácil de armazenar, transferir, processar e validar de forma digital, assim este tipo de identificação demonstram-se como uma forma simples, prática e económica de codificar informação.

Códigos de barras lineares desde cedo que se mostraram limitados na quantidade e tipo de dados que continham. Apareceram então no final dos anos 80 os códigos de barras de duas dimensões (2D), com o objectivo de suprimir essas necessidades. Existem dois tipos de códigos de barras 2D, os códigos de barras *Stacked*, como o PDF417 e os do tipo *Matrix*, como os *Data Matrix* e os *QR Code* (Ver Figura 69).

Alguns dos códigos de barras 2D mais comuns estão presentes na Tabela 25, tal como as suas principais características.

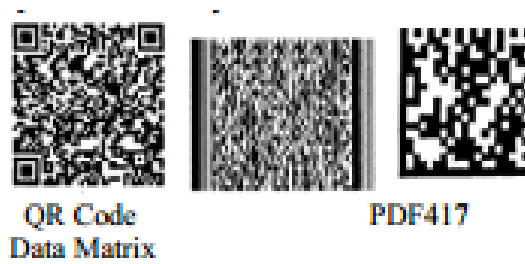
```
{
  "employees": [
    { "firstName":"John" , "lastName":"Doe" },
    { "firstName":"Anna" , "lastName":"Smith" },
    { "firstName":"Peter" , "lastName":"Jones" }
  ]
}
```

Figura 67 Exemplo estrutura de dados JSON



Fonte: Schottmuller (2011)




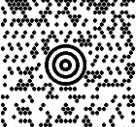
Figura 68 Código de Barras 1D vs 2D



Fonte: (Gao et al. 2009)

Figura 69 Exemplos Códigos de Barras 2D

Tabela 25 Características dos principais Códigos de Barras 2D (Adaptado de Gao et al. 2009; Ohbuchi 2004)

	<i>QR Code</i>	<i>PDF417</i>	<i>DataMatrix</i>	<i>Maxi Code</i>
<i>Criador (País)</i>	DENSO (Japão)	Symbol Technologi (USA)	RVSI Acuity CiMatrix (USA)	UPS (USA)
<i>Dados Numéricos</i>	7089	2710	3116	138
<i>Dados alfanuméricos</i>	4296	1850	2355	93
<i>Dados Binários</i>	2953	1018	1556	
<i>Dados Kanji</i>	1817	554	778	
<i>Principais Características</i>	Grande capacidade de armazenamento; Tamanho de impressão reduzido; Scan rápido;	Grande capacidade de armazenamento;	Tamanho de impressão reduzido;	Scan rápido;
<i>Normas</i>	AIM International; JEIDA; JIS; ISSO; GS1	AIM International; ISO;	AIM International; ISO;	AIM International; ISO;
<i>Exemplo</i>				

Comparado com os códigos de barras 1D, que armazenam uma quantidade de dados muito limitados, os códigos de barras 2D apresentam muito mais capacidade. Como demonstrado na Tabela 25 os códigos de barras QR podem armazenar informação até 7089 dígitos, 4296 letras e 2953 dados binários. A popularidade dos códigos de barras 2D advém da capacidade armazenar informação num espaço de impressão reduzido. Factores como o fim a que se destinam, as normas, a informação a conter e formato de impressão, devem ser tomados em conta no momento da selecção do formato a utilizar. Em contrapartida estes códigos necessitam de equipamentos mais avançados para a sua descodificação, mas que se tornou facilitado com a capacidade de processamento de imagem dos equipamentos portáteis (ex.: telemóvel) e que podem assim ser usados como leitores destes códigos (Gao et al. 2009).

Anexo B. Tecnologias de desenvolvimento do *front-end*

O *front-end* pode ser encarado como a interface entre o utilizador e o *back-end*. Neste âmbito, englobam-se as tecnologias de suporte ao desenvolvimento da aplicação móvel e da aplicação *web*.

1. *Linguagem de Programação - Java*

Java é uma popular linguagem de programação orientada a objectos desenvolvida pela *Sun Microsystems* em meados dos anos 90 (Arnold et al. 2000) e deve a sua rápida aceitação ao facto de ser usada como uma linguagem de programação para aplicações *World Wide Web* (WWW) (Badros 2013). As aplicações desenvolvidas são independentes da plataforma. Esta característica provem do facto do compilador de Java, em vez de gerar código binário, cria um código intermédio designado *bytecodes*. Os *bytecodes* são instruções semelhantes ao código máquina, mas que não se encontram relacionados com nenhum processador (Somera 2006). Os programas Java são compilados em *bytecodes* e interpretados pela máquina virtual Java - *Java Virtual Machine (JVM)*. Isto permite que qualquer sistema de computador com a *JVM* instalada seja capaz de executar programas em Java.

A plataforma Java é constituída pela *JVM* e pelas interfaces de *aplicativos Java - Application Programming Interface (APIs)*. *APIs Java* são bibliotecas de código compilado que podem ser utilizados nos programas desenvolvidos (Pawlan 2000). Estas permitem adicionar funcionalidades prontas e que permitem personalização, de forma a economizar tempo de programação. Existem disponíveis *APIs de Java* oficiais e inúmeras *APIs* desenvolvidas por terceiros.

A Figura 70 mostra um exemplo simples de uma *API Java*, esta *API* imprime uma linha de texto para a consola, com a mensagem “*Hello World!*”.

No desenvolvimento do projecto é feito uso da linguagem de programação Java no desenvolvimento da aplicação móvel em *Android*.

```
//Simple Example
class ExampleProgram {
public static void main(String[] args){
System.out.println("Hello World!");
}}
```

Figura 70 Exemplo de código Java

2. *Linguagem de Anotação – HTML*

Na *web* são usadas linguagens de anotação para descrever propriedades de informação. O intuito desta linguagem é criar um mecanismo que torne a informação compreensível para as aplicações e para os utilizadores. As anotações de informações tem um objectivo diferente das linguagens de programação tradicionais, que é executarem acções por uma determinada ordem (Deitel 2001).

A *HyperText Markup Language (HTML)* é uma linguagem *web*, considerada como universalmente entendida, destinada a apresentar dados, esta linguagem é interpretada pelo navegador *web* (Hors & Jacobs 1999). Esta interoperabilidade reduz os custos dos fornecedores de conteúdos, uma vez que apenas precisam desenvolver uma versão do documentos e esta será compatível entre todos os *browsers* e plataformas. O *HTML* provém do *Standard Generalized Markup Language (SGML)* que se destina a estruturar texto.

Segundo Hors & Jacobs (1999), o *HTML* dá aos programadores a possibilidade de:

- Publicação *online* de documentos com cabeçalhos, texto, tabelas, listas, fotos, etc;
- Recuperar informação através de *links* hipertexto¹;
- Criação de formulários para a realização de transacções;
- Possibilita inclusão de folhas de cálculo, vídeos e outros aplicativos;

Actualmente o *HTML* encontra-se na versão *HTML5* que integra funcionalidades como armazenamento local ou multiprocessamento (Freeman & Robson 2011). A Figura 71

¹ Do dicionário: “Sequência de texto que permite a remissão para outra localização (documento, ficheiro, página da Internet, etc.)”

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
<TITLE>My first HTML document</TITLE>
</HEAD>
<BODY>
<P>Hello world!</P>
</BODY>
</HTML>

```

Figura 71 Exemplo de código HTML

apresenta um simples exemplo de código *HTML* onde apenas é apresentado um parágrafo com a mensagem “*Hello world!*”.

3. ***PHP: HyterText Preprocessor***

PHP é uma linguagem de *script open-source*, muito utilizada para desenvolvimento de conteúdos dinâmicos no desenvolvimento de páginas *web*. Sendo esta uma das primeiras linguagens a privilegiar o processamento do lado do servidor, utilizando *scripts* directamente nas páginas *HTML*, em vez de chamar ficheiros externos (PHP Group 2013).

O código *PHP* é delimitado num ficheiro *HTML* para *tags* `<?php` e `?>`, que correspondem respectivamente ao início e ao final do *script*. A Figura 72 mostra um extracto de código simples e que mostra ao utilizador a mensagem “*Hello World!*”.

```

<!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>My first PHP</title>
</head>
<body>
<?php
    echo "Hello World!";
?>
</body>
</html>

```

Figura 72 Exemplo de código PHP

O *PHP* tornou-se popular pelo facto de ser uma linguagem de compreensão simples para iniciantes e oferecer recursos mais avançados para programadores mais experientes. O *PHP* pode ser utilizado na maioria dos sistemas operacionais e é suportado pelos principais servidores *web*. O *PHP* não está limitada apenas a gerar páginas *HTML*, sendo possível também gerar imagens, arquivos PDF e animações Flash. Contudo um dos principais pontos fortes do PHP é ser suportado pela vasta gama de bases de dados existente (Group 2013).

4. *JavaScript*

O *JavaScript* é uma linguagem de *script*, baseada em Java, que pode ser incluída num documento *HTML* ou em ficheiros externos. Esta linguagem permite tal como o *PHP* a introdução de conteúdos dinâmicos e confere maior interactividade à pagina *web*. É actualmente a principal linguagem para programação do lado do cliente, tornando as páginas *web* mais interactivas para o cliente, tendo sido criada nesse sentido para que os scripts pudessem ser processados do lado do cliente, libertando o servidor dessas funções. Figura 73 apresenta um extracto de código simples de código *JavaScript*.

5. *Android*

O *Android* é uma plataforma de desenvolvimento para dispositivos móveis com SO baseado em Linux. Os dispositivos que incorporam Linux tem vindo a expandir-se tanto na Europa como em Portugal, sendo mesmo a plataforma mais utilizada. As Figura 74 e Figura 75 mostram a evolução da distribuição dos principais SO tanto na Europa como em Portugal, segundo o *Stat Counter* (Stats 2013).

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Web Page</h1>
<script>
document.write("<p>Hello World!</p>");
</script>
</body>
</html>
```

Figura 73 Exemplo de código JavaScript

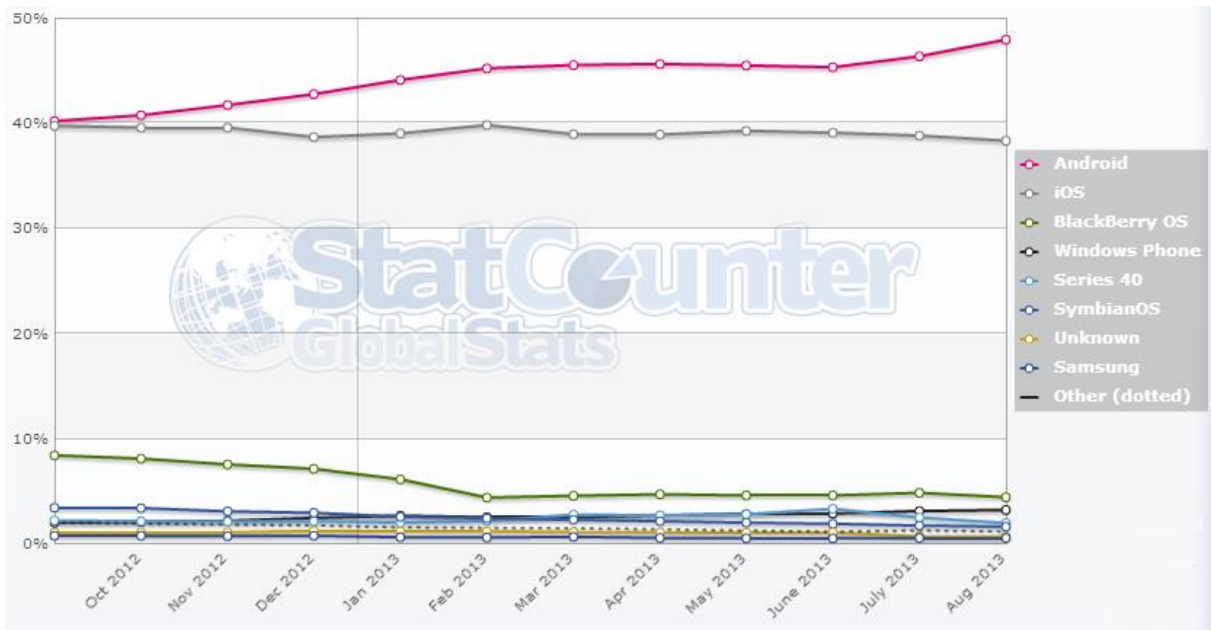


Figura 74 Evolução da distribuição dos principais SO móveis na Europa (Setembro 2012 – Agosto 2013)

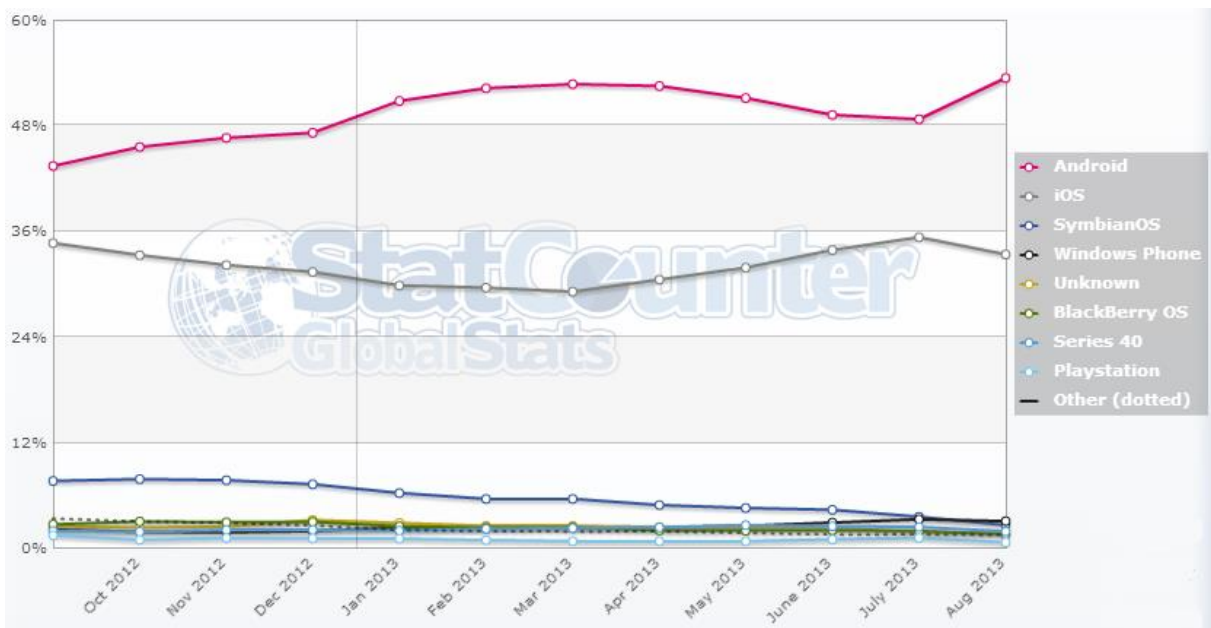


Figura 75 Evolução da distribuição de SO móveis em Portugal (Setembro 2012 – Agosto 2013)

O desenvolvimento em *Android* é feito em Java e necessita de um *kit* de desenvolvimento, *Software Development Kit* (SDK), próprio que disponibiliza as ferramentas e bibliotecas essenciais à criação de aplicações. Assim sendo, é importante no momento do desenvolvimento da aplicação seleccionar a versão SDK sobre a qual se vai trabalhar. Para isso é necessário conhecer alguns pressupostos, entre eles, as versões existentes, a distribuição das versões pelos dispositivos e é necessário ter em conta que aplicações criadas

para versões de SDK inferiores são compatíveis com versões mais recentes. A própria página do *Android* disponibiliza essa informação, na Tabela 26 e Figura 76 são mostradas as versões do SDK e a sua distribuição. O desenvolvimento da aplicação *Android* foi realizada com o SDK 2.2, pois é a versão mais baixa a ser considerada no estudo de distribuição e garantindo assim que a aplicação é compatível para todas as versões superiores.

ACTIVIDADES

Uma actividade é um componente da aplicação que proporciona uma janela de interface com o utilizador e que lhe permite interagir com a aplicação. Tipicamente uma aplicação é constituída por várias actividades que estão ligadas entre si. A actividade principal (*main*) é apresentada ao utilizador no arranque da aplicação *Android*. Cada actividade pode dar início a outra possibilitando a execução de diferentes acções. Num determinado instante apenas está visível uma actividade. Cada actividade pode existir essencialmente em três estados:

Tabela 26 Versões dos SDK *Android*

Nome	Versão	API	Distribuição
<i>Froyo</i>	2.2	8	2.4%
<i>Gingerbread</i>	2.3.3 – 2.3.7	10	30.7%
<i>Honeycomb</i>	3.2	13	0.1%
<i>Ice Cream Sandwich</i>	4.0.3 – 4.0.4	15	21.7%
<i>Jelly Bean</i>	4.1.x	16	36.6%
<i>Jelly Bean</i>	4.2.x	17	8.5%

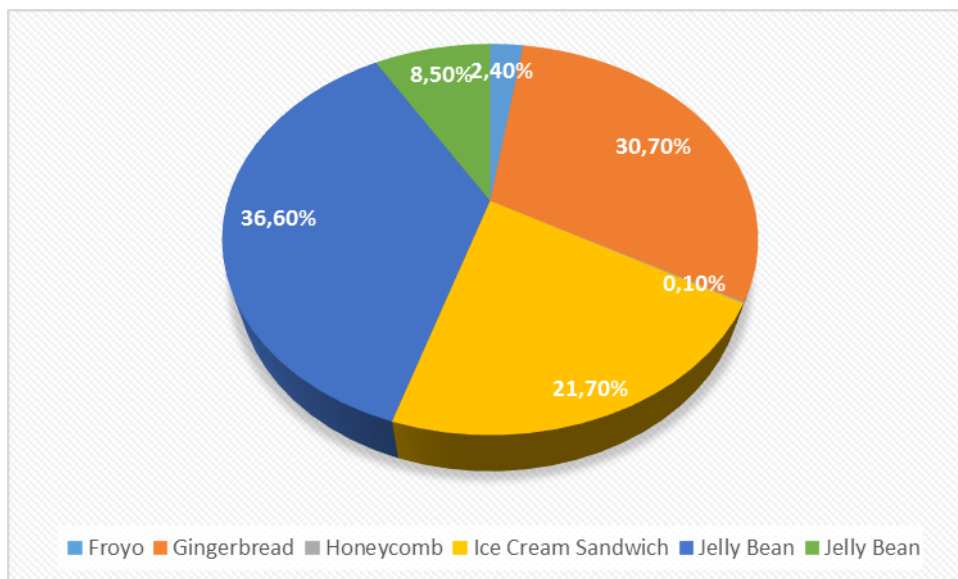


Figura 76 Distribuição do SDK *Android*

- **Em execução**: a actividade está visível no ecrã do dispositivo e é nesta que recai a atenção do utilizador.
- **Em pausa**: a actividade está em execução mas não é naquele momento a actividade principal. Esta actividade pode estar visível de uma forma mais transparente ou não ocupando todo o ecrã, pois a atenção do utilizador recai sobre outra actividade, por exemplo, quando se recebe uma chamada. Neste estado o sistema armazena todas as características da actividade, apenas libertando esses recursos caso sistema necessite de mais memória.
- **Parada**: a actividade está totalmente tapada por outra actividade, encontrando-se em *background*. O sistema armazena todas as características de uma actividade, mesmo quando ela está no estado parada. No entanto, a actividade não está visível para o utilizador e pode ser destruída a qualquer momento que o sistema necessite de memória.

A passagem entre os estados do ciclo de vida de uma actividade (Ver Figura 77) pode ser controlado através de métodos apropriados:

- **onCreate()**: método chamado quando a actividade está a iniciar. Após este método a actividade encontra-se em execução.
- **onStart()**: método de restauro da aplicação. Quando esta se encontra em estado parado e necessita de voltar a ser visível para o utilizador.
- **onResume()**: método de preparação de uma actividade em pausa. Método utilizado para a actividade voltar a interagir com o utilizador. Nesta actividade devem ser iniciados os recursos de acesso exclusivo, como por exemplo a camera.
- **onPause()**: método chamado quando a actividade vai ser colocada em *background* mas não vai ser destruída. Este método deve ser usado para parar processos que consomem *Central Processing Unit* (CPU) e memória.
- **onStop()**: método de paragem da actividade quando está não está visível ao utilizador.

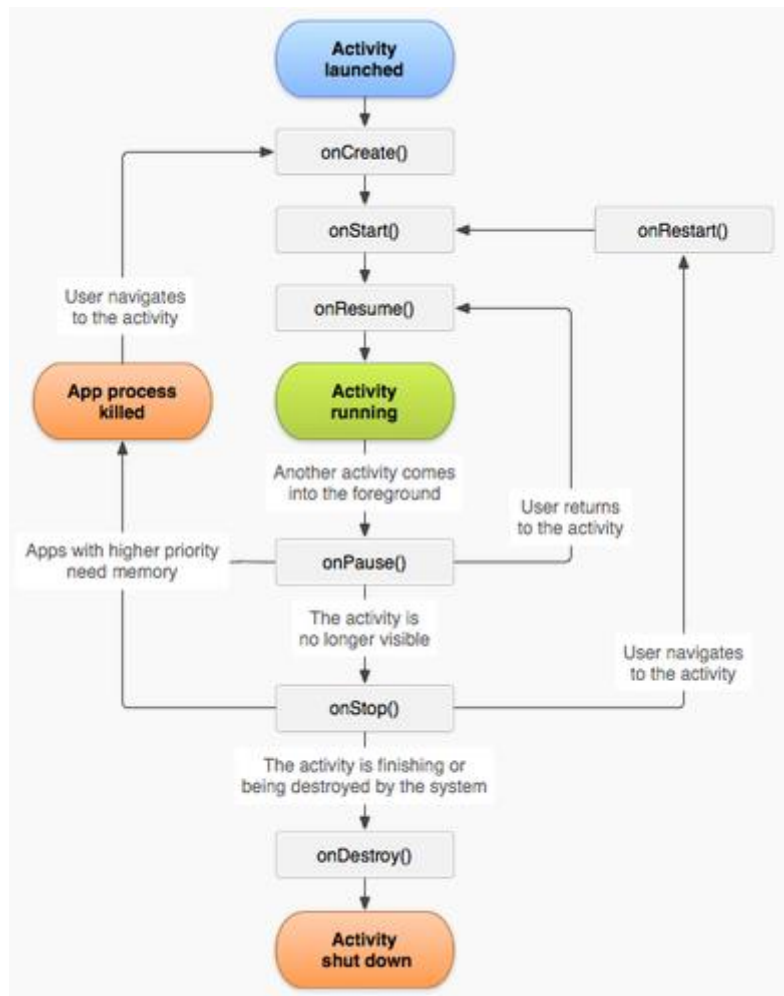


Figura 77 Ciclo de vida de uma actividade (Android 2013)

- **onDestroy()**; método de encerramento da actividade. Pode ser utilizado quando o sistema destrói a actividade ou quando esta é terminada pela aplicação, libertando todos os recursos.

A Figura 78 exemplifica a criação de uma actividade *Android*. As actividades devem ser declaradas no manifesto. O manifesto é um documento *XML*, que contém toda a informação da aplicação, inclusive as actividades para estas estarem acessíveis pelo sistema.

LAYOUT

O *layout* define a estrutura visual da interface com o utilizador. O *layout* pode ser declarado de duas formas: (i) através de um documento *XML*, que define o aspecto da actividade ou (ii) através da adição de elementos durante a execução da actividade, permitindo alterar a

```

public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
}

```

Figura 78 Exemplo de criação Actividade Android

interface a partir de interações com o utilizador. Estes dois métodos de definição de *layout* podem ser usados, tanto separados como em conjunto.

A utilização de um ficheiro *XML* permite a separação do código da aparência da aplicação do código que determina o seu comportamento. Utilizando o vocabulário *Android* para *XML*, facilmente se consegue projectar um interface, com uma estrutura semelhante à de um ficheiro *HTML*. A Figura 79 apresenta a definição do *layout* ilustrado na Figura 80, os componentes visuais que constam da actividade são identificados por um valor representado no ficheiro *XML* como “*android:id*”, no exemplo é visível a existência de uma zona de texto e de um botão.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>

```

Figura 79 Exemplo código XML para *layout* Android



Figura 80 Exemplo *layout* de uma actividade Android

SQLite

SQLite é uma biblioteca que implementa um sistema de base de dados. Com certificação ACID, implementa a maioria da sintaxe e tipo de dados das base de dados comuns. *SQLite* não se trata de um servidor, como a normal base de dados *SQL*, mas sim de uma aplicação local que lê e escreve dados em ficheiros. Sendo assim, uma escolha comum em sistemas simples e com poucos acessos (SQLite 2013).

Anexo C. Ambiente de desenvolvimento integrado

Neste anexo são descritos alguns dos passos para configurar o *software* Eclipse, para o desenvolvimento da aplicação móvel *Android*. O *software* Eclipse pode ser obtido a partir do endereço <http://www.eclipse.org/downloads>.

Adição de *Android Plug-in*

O Eclipse permite a adição de *plug-ins* para o desenvolvimento de aplicações para determinados ambientes. Um *plug-in* é um módulo de software capaz de adicionar funcionalidades a programas de maior dimensão.

Para a integração de *plug-ins* no Eclipse, deve-se seguir os seguintes passos:

1. Na aba *Help* seleccionar opção “*Install New Software*”;
2. Na janela que é apresentada clicar em “*Add*” e inserir o URL do *plug-in* a instalar e carregar “*Ok*” (Ver Figura 81);
3. Os *plug-ins* disponíveis serão apresentados, seleccionar os pretendidos e concluir a instalação.

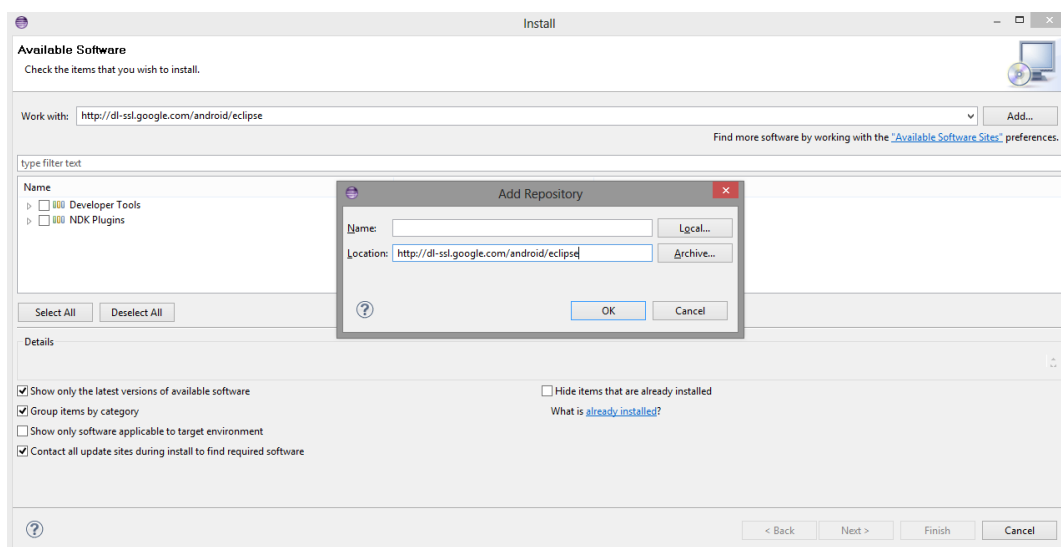


Figura 81 Adição de *plug-in* no Eclipse

ANDROID SDK

O desenvolvimento para a plataforma *Android* requer a instalação do *SDK* do *Android*. Este pacote inclui o *Android SDK Manager*, que permite seleccionar as API e *SDK* específicos que se pretende instalar para o desenvolvimento da aplicação (Ver Figura 82).

O *Android* disponibiliza um *plug-in* para integração com o Eclipse, *Android Development Tools (ADT)*. Este *plug-in* oferece um ambiente integrado para o desenvolvimento de aplicações *Android*. Este *plug-in* capacita o Eclipse para a configuração de projectos *Android*, desenvolver interface com o utilizador e exportar o pacote da aplicação para distribuição. O *ADT plug-in* está disponível através do endereço: <https://dl-ssl.google.com/android/eclipse/>.

A criação de um novo projecto *Android* no Eclipse segue os seguintes passos:

1. Na barra de ferramentas seleccionar *File -> New -> Android Application Project*;

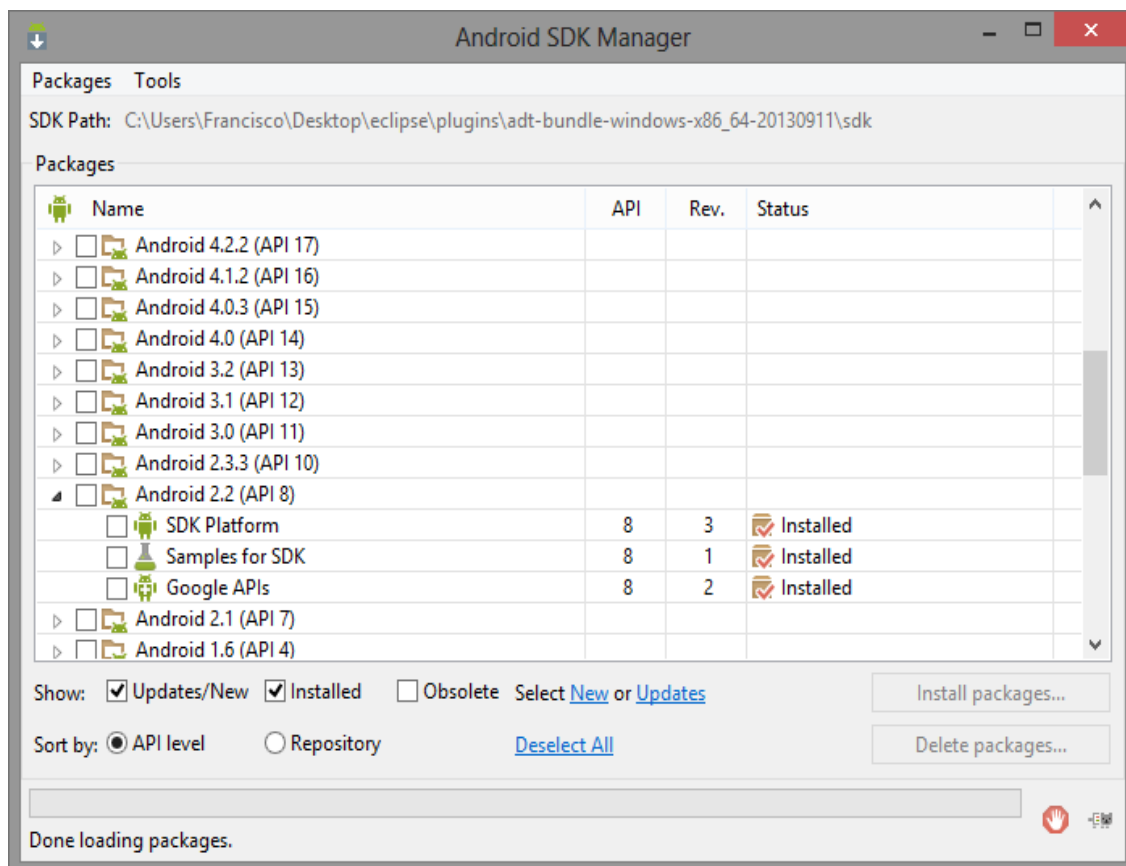


Figura 82 *Android* SDK Tools

2. Na nova janela deve-se introduzir o nome do projecto e seleccionar o SDK seleccionado para a aplicação (Ver Figura 83);
3. Concluir passos para criar pacote do projecto.

As aplicações criadas podem ser testadas/executadas com recurso a um dispositivo físico, configurando o equipamento para o modo de depuração USB. Ou através do emulador disponibilizado no Eclipse.

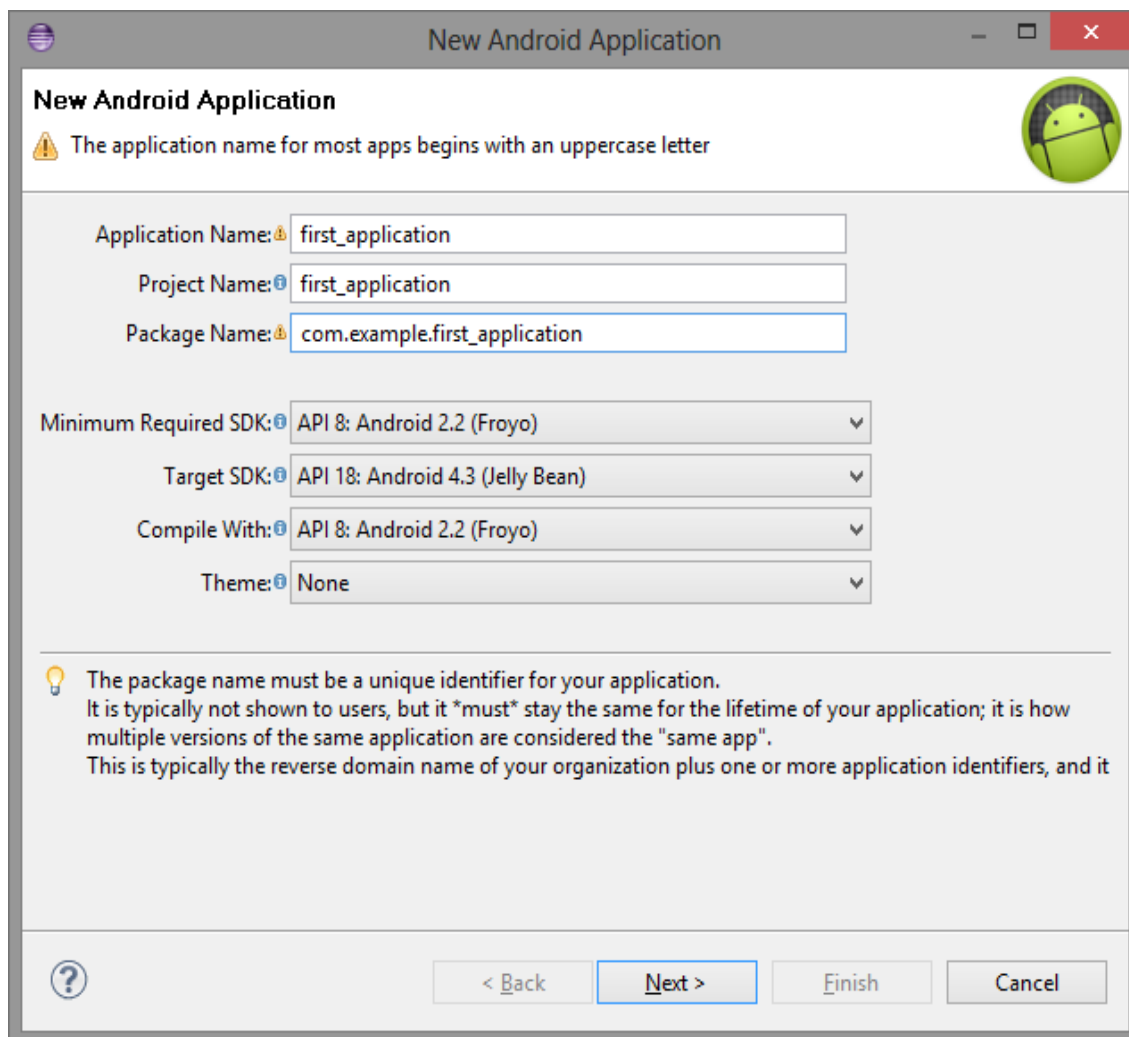


Figura 83 Criação projecto *Android*