



Universidade do Minho  
Escola de Engenharia

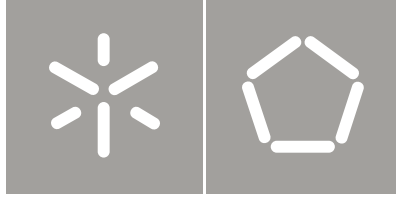
Freddy Dinis Rodrigues Gonçalves

Um Sistema de  
Informação Espaço-Temporal  
para Objectos Móveis

Freddy Dinis Rodrigues Gonçalves  
Um Sistema de  
Informação Espaço-Temporal para Objectos Móveis

UMinho | 2012

janeiro de 2012



Universidade do Minho  
Escola de Engenharia

Freddy Dinis Rodrigues Gonçalves

Um Sistema de  
Informação Espaço-Temporal  
para Objectos Móveis

Tese de Mestrado  
Ciclo de Estudos Integrados Conducentes ao  
Grau de Mestre em Engenharia de Comunicações

Trabalho efetuado sob a orientação do  
Professora Doutora Maribel Yasmina Santos

## **DEDICATÓRIA**

Desejo dedicar esta dissertação aos meus pais e aos meus irmãos. Pela confiança que sempre depositaram em mim, pelo incondicional amor e apoio, sem o qual nunca teria chegado onde me encontro hoje.



## **AGRADECIMENTOS**

A concretização deste trabalho só foi possível devido a um conjunto de condições e meios postos à disposição, bem como à dedicação, empenho e vontade de vários intervenientes, a quem gostaria de expressar os meus mais profundos e sinceros agradecimentos.

À Universidade do Minho, pelos meios e condições postos à disposição.

À minha orientadora Doutora Maribel Yasmina Santos, quero expressar o meu enorme reconhecimento pela ajuda, apoio e estímulo manifestados desde sempre, bem como pelos conhecimentos e orientações incondicionalmente transmitidos.

À equipa de investigadores liderada pela Prof. Monica Wachowicz que disponibilizou os ficheiros de dados sobre o movimento utilizados neste trabalho.

A todos os meus amigos, pelo incansável apoio, que mais não fosse, por vezes é o mais importante, apoio moral.

Estou muito grato pelo apoio prestado por todos os meus amigos, conhecidos ao longo da minha formação, Diogo Mendes, Hélder Ribeiro, César Alexandre, Laurent, Diogo Gomes, Francisco Silva, Huguinho, Luís Gonçalves, Rui Rodrigues entre outros, pelo encorajamento e apoio ao longo de toda a minha formação, sem os quais não teria sido possível realizar este percurso da mesma forma.

Um agradecimento especial para a irmandade e meus irmãos de coração Melo, César, João Luís, Pedro, Filipe, Lip, Miguel Correia, Sousa e Michael pela motivação e apoio nos momentos mais difíceis.

Um agradecimento especial ao presidente, João Melo, quero expressar o meu enorme reconhecimento pelo constante apoio, encorajamento e irmandade revelada em toda a minha vida, bem como pelos conhecimentos e experiências transmitidas.

À Liliana porque sim.

Finalmente, gostaria de agradecer aos meus pais e aos meus irmãos pelo seu constante apoio, compreensão, sacrifício e incentivo durante o decorrer deste estudo e, sobretudo, pelo seu amor e confiança em mim.

## RESUMO

A grande evolução tecnológica tanto dos Sistemas de Informação Geográfica (SIG) bem como das tecnologias de localização (GPS, WI-FI, RFID) contribuíram para um significativo aumento da recolha de dados espaciais. Devido a esta proliferação e às grandes quantidades de dados que são recolhidos, são necessárias bases de dados e mecanismos apropriados para o seu armazenamento e sua respectiva análise.

No passado, o desenho de sistemas de informação para objectos móveis basearam-se muitas vezes em abordagens (aplicações) restritas a tecnologias de localização específicas, levando a um vasto leque de modelos de dados, modelos de base de dados e respectivas funcionalidades. Para superar esta proliferação de modelos, este projecto propõe um sistema de informação espaço-temporal independente do domínio de aplicação, e com o intuito de se abstrair da tecnologia de posicionamento utilizada para a recolha de dados.

Este sistema de informação espaço-temporal visa o desenvolvimento de um sistema de armazenamento, análise e visualização de dados sobre objectos móveis capazes de representar e armazenar dados com características espaciais e de realizar análises sobre os mesmos. Este sistema é intitulado STAR e integra três componentes principais, as bases de dados espaço-temporais, os SIG e os dados espaço-temporais. A base de dados espaço-temporal desenhada permite o armazenamento dos dados espaço-temporais bem como o armazenamento da geometria do espaço no qual ocorre o movimento. Esta geometria é representada por pontos, linhas e/ou polígonos. Os SIG auxiliam os processos de análise e visualizações de dados. Os dados espaço-temporais são intitulados neste projecto de objectos móveis, podendo o movimento dos mesmos ser analisado no STAR.

Os resultados alcançados são promissores na demonstração de como o sistema é capaz de armazenar dados de posicionamento com formatos diferentes, e na aplicação de análises, visualizações e representações sobre os mesmos.

**PALAVRAS-CHAVE:** Dados espaciais; Objectos Móveis; Base de Dados Espaço-Temporal; SIG





**ABSTRACT**

The huge technological evolution of both the Geographic Information Systems (GIS) as well as location technologies (GPS, Wi-Fi, RFID) contributes to a large increase in spatial data collection. Because of this proliferation and to the large amounts of data that are collected, databases and appropriate mechanisms for their storage and their analysis are necessary.

In the past, the design of information systems for moving objects was often based on approaches (applications) restricted to specific location technologies, leading to a wide range of data models, database models and functionalities. To overcome this proliferation of models, this project proposes a spatio-temporal information system that is independent of the application domain and that abstract the positioning technology used for data collection.

This spatio-temporal information system aims the development of a storage system with data analysis and visualization mechanism for moving objects that can represent and store data with spatial characteristics and perform spatial analysis on them. This system is named STAR and incorporates three main components, spatio-temporal databases, GIS and spatio-temporal data. The designed spatio-temporal database enables the storage of the spatio-temporal data and the geometry of the space in which the movement occurs. This geometry is represented by points, lines and/or polygons. The GIS help the processes of analysis and visualization of data. The spatio-temporal data in this project are entitled to moving objects, their movement can be analysed in STAR.

The results obtained so far are promising in demonstrating how the system is capable of storing positioning data with different formats, and in applying analysis, visualizations and different representations of the analysed data.

**KEYWORDS:** Spatial Data; Moving Objects; Spatio-Temporal Databases; GIS

**ACRÓNIMOS**

Neste documento são utilizados vários acrónimos que representam abreviações de designações utilizadas. Os acrónimos adoptados são:

API	<i>Application Programming Interface</i>
BD	Base de Dados
BDE	Base de Dados Espacial
BDT	Base de Dados Temporal
DML	<i>Data Manipulation Language</i>
GPS	Sistema de Posicionamento Global ( <i>Global Positioning System</i> )
LBS	<i>Location Based Services</i>
MOD	Base de Dados de Objectos Móveis ( <i>Moving Object Database</i> )
NPGSQL	<i>.Net Data Provider for PostgreSQL</i>
ODBC	<i>Open Database Connectivity</i>
ORDBMS	<i>Object Relational Database Management System</i>
RFID	<i>Radio Frequency Identification</i>
SIG	Sistemas de Informação Geográfica
SGBD	Sistema de Gestão de Base de Dados
SGBDG	Sistema de Gestão de Base de Dados Geográfica
SQL	<i>Structured Query Language</i>
TD	<i>Temporal Database</i>
UML	<i>Unified Modeling Language</i>
WI-FI	<i>Wireless Fidelity</i>

## ÍNDICE

DEDICATÓRIA.....	III
AGRADECIMENTOS .....	V
RESUMO .....	VII
ABSTRACT .....	IX
ACRÓNIMOS .....	X
ÍNDICE .....	XI
ÍNDICE DE FIGURAS.....	XV
ÍNDICE DE TABELAS.....	XVII
1. INTRODUÇÃO .....	1
1.1. <i>Enquadramento e Motivação</i> .....	1
1.2. <i>Finalidade e Objectivos do Trabalho</i> .....	3
1.3. <i>Organização do Documento</i> .....	4
2. REVISÃO DO ESTADO DA ARTE .....	7
2.1. <i>Armazenamento e Análise de Dados sobre Objectos Móveis</i> .....	7
2.1.1. Sistemas de Informação Geográfica .....	7
2.1.2. Modelos de base de Dados para objectos móveis.....	12
2.2. <i>Representação de dados espaciais e objectos móveis</i> .....	18
2.2.1. Representação de tipos de dados espaciais .....	19
2.2.2. Representação de tipos dados Espaço-temporais.....	22
2.3. <i>Sistemas de Armazenamento e Análise de Dados sobre Objectos Móveis</i> .....	24
2.3.1. <i>Hermes: Aggregative LBS via a Trajectory DB Engine</i> .....	24
2.3.2. <i>Design and Implementation of Moving Object Database for Truck Information Systems - DMOTIS</i> .....	28
2.3.3. <i>SECONDO: An Extensible DBMS Architecture and Prototype</i> .....	32
3. STAR: UM SISTEMA DE INFORMAÇÃO ESPAÇO-TEMPORAL PARA OBJECTOS MÓVEIS.....	39
3.1. <i>Especificação de Requisitos</i> .....	39
3.2. <i>Contextualização das opções de desenvolvimento</i> .....	40
3.2.1. Sistema de Gestão de Base de Dados (SGBD).....	40
3.2.2. Base de Dados Centralizada.....	42
3.2.3. Arquitectura em Camadas .....	44
3.2.4. Representação e Análise de Objectos Móveis .....	46
3.3. <i>Arquitectura do Sistema</i> .....	48
3.3.1. Camada de Apresentação ( <i>Presentation Layer</i> ).....	50
3.3.2. Camada de Negócio ( <i>Business Layer</i> ) .....	51
3.3.3. Camada de Dados ( <i>Data Layer</i> ) .....	51

3.3.4.	Camada de Apoio ( <i>Helper Classes Layer</i> ) .....	52
3.3.5.	Modelo de Dados ( <i>Database</i> ) .....	53
3.4.	<b>Funcionalidades do Sistema</b> .....	56
3.4.1.	Actores do Sistema.....	57
3.4.2.	Diagramas de Casos de Uso .....	57
3.5.	<b>Síntese comparativa das especificações STAR v.1.0 - STAR v.2.0</b> .....	66
4.	<b>IMPLEMENTAÇÃO DO PROTÓTIPO PARA O STAR v2.0</b> .....	71
4.1.	<b>Enquadramento Tecnológico</b> .....	71
4.1.1.	<i>C# e Visual Studio</i> .....	72
4.1.2.	PostgreSQL .....	75
4.1.3.	PostGIS .....	77
4.1.4.	ZedGraph.....	78
4.1.5.	Sharpmap .....	79
4.2.	<b>Implementação Realizada</b> .....	79
4.2.1.	Conexão à Base de Dados - NPGSQL .....	79
4.2.2.	Importar Ficheiro .....	80
4.2.3.	Visualizar Dados .....	88
4.2.4.	Analisar Dados.....	92
4.2.5.	Correcção de Dados .....	93
5.	<b>APRESENTAÇÃO E AVALIAÇÃO DO PROTÓTIPO IMPLEMENTADO</b> .....	95
5.1.	<b>Apresentação do STAR v2.0</b> .....	95
5.1.1.	Instalação do STAR v2.0 .....	95
5.1.2.	Apresentação da Interface do STAR v2.0 .....	101
5.2.	<b>Apresentação das Funcionalidades do STAR v. 2.0</b> .....	107
5.2.1.	Autenticar.....	108
5.2.2.	Gerir Arquivos e Gerir Ficheiros .....	110
5.2.3.	Gerir Dados .....	115
5.3.	<b>Caracterização das Análises e Visualização dos Dados</b> .....	125
5.3.1.	Descrição dos conjuntos de dados .....	125
5.3.2.	Análise e Visualizações de Dados .....	126
5.4.	<b>Avaliação do Trabalho Implementado</b> .....	137
6.	<b>CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS</b> .....	141
6.1.	<b>Trabalho Realizado</b> .....	141
6.2.	<b>Trabalhos Futuros</b> .....	144
6.3.	<b>Considerações Finais</b> .....	145
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	147
	<b>ANEXOS</b> .....	153
	<b>ANEXO 1 – SCRIPT DE CRIAÇÃO DA BD</b> .....	154

ANEXO 2 – FIGURAS DO SISTEMA STAR V. 2.0 .....	157
ANEXO 3 – FIGURAS DO SISTEMA STAR V. 2.0 .....	163
ANEXO 4 – FIGURAS DO SISTEMA STAR V. 2.0 .....	166



## ÍNDICE DE FIGURAS

<i>Figura 1 - Componentes de um SIG (Adaptado de: (Câmara e Freitas, 1995) p.2)</i> .....	11
<i>Figura 2 - Relação entre Bases de dados Temporais, Espaciais e de Objectos Móveis (Retirado de: (Abdul-Kader, 2009) p. 65)</i> .....	13
<i>Figura 3 - Relação entre conceitos Espaço-Temporais, Temporais e Espaciais (Adaptado de: (Zhang, 2005))</i> ....	16
<i>Figura 4 - Tipos de Dados Espaciais (Adaptado de: (Gutting, 2000))</i> .....	20
<i>Figura 5 - Coleções de Objectos Espaciais (Retirado de: (Gutting, 1994) p.3)</i> .....	20
<i>Figura 6 - Representação por células (Adaptado de: (Gatrel, 1991) p.125)</i> .....	21
<i>Figura 7 - Representação Vectorial (Retirado de: (Santos, 2001) p.18)</i> .....	22
<i>Figura 8 - Tipos de dados Espaço - Temporais (Retirado de: (Erwig et al., 1999))</i> .....	22
<i>Figura 9 - Representação de Tipos de dados Espaço-Temporais: mpoint e mregion (Retirado de: (Gutting, 2007) p. 3)</i> .....	23
<i>Figura 10 - Consulta a Base de Dados de Trajectória (Retirado de: (Pelekis et al., 2008) p. 1)</i> .....	25
<i>Figura 11 - Arquitectura HERMES (Retirado de: (Pelekis et al., 2008) p. 2)</i> .....	26
<i>Figura 12 - Imagens da Interface HERMES: desktop e mobile (Retirado de: (Pelekis et al., 2008) p. 3)</i> .....	28
<i>Figura 13 - Rede de Estradas do Irão (Retirado de: (Qajary et al., 2008) p. 3)</i> .....	29
<i>Figura 14 - Modelo de Dados de Trajectória num MOD (Retirado de: (Qajary et al., 2008) p. 4)</i> .....	30
<i>Figura 15 - Arquitectura DMOTIS (Adaptado de: (Qajary et al., 2008) p. 5)</i> .....	31
<i>Figura 16 - Componentes SECONDO (Retirado de: (Almeida et al., 2006) p. 3)</i> .....	33
<i>Figura 17 - Arquitectura do Kernel SECONDO (Retirado de: (Almeida et al., 2006) p. 3)</i> .....	35
<i>Figura 18 - Protocolo do Optimizador SECONDO (Retirado de: (Almeida et al., 2006) p. 4)</i> .....	36
<i>Figura 19 - Interface Gráfica SECONDO (Retirado de: (Gutting et al., 2007) p. 5)</i> .....	37
<i>Figura 20 - Sistema de Gestão de Base de Dados (Adaptado de: (Schunemann, 2010) p. 2)</i> .....	41
<i>Figura 21 - Base de Dados Centralizada</i> .....	43
<i>Figura 22 - Arquitectura em Camadas (Retirado de: (Silva, 2007) p. 18)</i> .....	45
<i>Figura 23 - Representação de trajectórias. Em (a) direcção das trajectórias e (b) a largura é associada à velocidade (Retirado de: (Orellana et al., 2009))</i> .....	46
<i>Figura 24 - - Representação de Vectores (Retirado de: (Orellana et al., 2009))</i> .....	47
<i>Figura 25 - Representação de Redes. Em (a) nós e arcos (b) Raio está associado ao número de nós de conexão e a largura da linha está associada ao custo (Retirado de: (Orellana et al., 2009))</i> .....	47
<i>Figura 26 - Representação de Fluxos. Em (a) fluxo discreto (b) fluxo contínuo (Retirado de: (Orellana et al., 2009))</i> .....	48
<i>Figura 27 - Arquitectura do Sistema</i> .....	49
<i>Figura 28 - STAR v2.0 - Camada de Apresentação</i> .....	50
<i>Figura 29 - STAR v2.0 - Camada de Negócio</i> .....	51
<i>Figura 30 - STAR v2.0 - Camada de Dados</i> .....	52
<i>Figura 31 - Exemplo do ficheiro Log gerado pelo sistema</i> .....	53

<i>Figura 32 - Modelo de Dados Implementado</i> .....	54
<i>Figura 33 - Diagrama de Classes</i> .....	55
<i>Figura 34 – Diagrama de Casos de Uso: Interface do Sistema</i> .....	58
<i>Figura 35 – Diagrama de Casos de uso: Autenticar</i> .....	59
<i>Figura 36 – Diagrama de Casos de Uso: Gerir Arquivos</i> .....	59
<i>Figura 37 – Diagrama de Casos de Uso: Ficheiros</i> .....	60
<i>Figura 38 – Diagrama de Casos de Uso: Importar Ficheiro</i> .....	62
<i>Figura 39 - Caso de Uso: Gerir Dados</i> .....	62
<i>Figura 40 - Caso de Uso: Visualizar Dados</i> .....	63
<i>Figura 41 – Diagrama de Casos de Uso: Analisar Dados</i> .....	65
<i>Figura 42 – Diagrama de Casos de Uso: Corrigir Dados</i> .....	66
<i>Figura 43 - Conexão</i> .....	80
<i>Figura 44 - Funcionalidade Importe</i> .....	81
<i>Figura 45 - Funcionamento do Importe</i> .....	82
<i>Figura 46 - Infer</i> .....	84
<i>Figura 47 – Formato da Expressão Regular</i> .....	85
<i>Figura 48 - Parser</i> .....	86
<i>Figura 49 - Ingester</i> .....	87
<i>Figura 50 - Funcionamento da Visualização de Dados</i> .....	88
<i>Figura 51 –Visualizar Dados</i> .....	90
<i>Figura 52 - Algoritmo para Cálculo de Vectores Direcctionais</i> .....	91
<i>Figura 53 - Funcionamento da Análise de Dados</i> .....	92
<i>Figura 54 – Análise de Dados</i> .....	93
<i>Figura 55 - Instalação do STAR v. 2.0</i> .....	96
<i>Figura 56 - Instalação do STAR v. 2.0</i> .....	96
<i>Figura 57 - Instalação do STAR v. 2.0</i> .....	97
<i>Figura 58 - Instalação do STAR v. 2.0</i> .....	97
<i>Figura 59 - Instalação PostgreSQL</i> .....	98
<i>Figura 60 - StackBuilder</i> .....	99
<i>Figura 61 - Aplicações Adicionais: StackBuilder</i> .....	99
<i>Figura 62 – PgAdminIII: Nova Base de Dados</i> .....	100
<i>Figura 63 – PgAdminIII: Importar Backup</i> .....	101
<i>Figura 64 - STAR v 2.0: Interface Utilizador</i> .....	102
<i>Figura 65 – STAR v. 2.0: Componente Buttons &amp; Menu</i> .....	103
<i>Figura 66 - STAR v. 2.0: Componente Tree</i> .....	104
<i>Figura 67 - STAR v. 2.0: User Account Details Control</i> .....	105
<i>Figura 68 - STAR v. 2.0: Folder Control</i> .....	105
<i>Figura 69 - STAR v2.0: File Control</i> .....	106



<i>Figura 70 - STAR v. 2.0: Device Control</i> .....	106
<i>Figura 71 - STAR v. 2.0: Current Viewer</i> .....	107
<i>Figura 72 - STAR v. 2.0: Autenticação</i> .....	108
<i>Figura 73 - STAR v. 2.0 – Erro Autenticação</i> .....	109
<i>Figura 74 - STAR v. 2.0: Novo utilizador</i> .....	109
<i>Figura 75 - STAR v. 2.0: Alteração da conexão à BD</i> .....	110
<i>Figura 76 - STAR v. 2.0: Erro Configuração Conexão</i> .....	110
<i>Figura 77 - STAR v. 2.0: Criar novo Arquivo</i> .....	111
<i>Figura 78 - STAR v. 2.0: Pesquisar Arquivos</i> .....	112
<i>Figura 79 - STAR v. 2.0: Listar Arquivos</i> .....	112
<i>Figura 80 - STAR v. 2.0: Import File</i> .....	114
<i>Figura 81 - STAR v. 2.0: Import File</i> .....	115
<i>Figura 82 - STAR v. 2.0: Draw Trajectory</i> .....	116
<i>Figura 83 - STAR v. 2.0: Draw Vector</i> .....	117
<i>Figura 84 - STAR v. 2.0: Draw Vector</i> .....	118
<i>Figura 85 - STAR v. 2.0: Draw Points</i> .....	118
<i>Figura 86 –Show Data: R274, R709 e R483</i> .....	119
<i>Figura 87 - STAR v. 2.0: Represent Data</i> .....	120
<i>Figura 88 - STAR v. 2.0: Represent Data</i> .....	121
<i>Figura 89 - STAR v. 2.0: Scarter Plot</i> .....	122
<i>Figura 90 - STAR v. 2.0: Histogram</i> .....	123
<i>Figura 91 - STAR v. 2.0: Show Data</i> .....	124
<i>Figura 92 - STAR v. 2.0: Data Cleaning</i> .....	125
<i>Figura 93 – Desenho dos pontos do pedestre R263</i> .....	127
<i>Figura 94 – Desenho dos pontos do pedestre R660</i> .....	128
<i>Figura 95 – Desenho dos pontos do pedestre R656</i> .....	128
<i>Figura 96 - Desenho dos pontos dos pedestres R660 e R656</i> .....	129
<i>Figura 97 –Trajectória do pedestre R263</i> .....	130
<i>Figura 98 - Trajectória do pedestre R656</i> .....	130
<i>Figura 99 - Trajectória do pedestre R660</i> .....	131
<i>Figura 100 –Vectores do pedestre R656</i> .....	131
<i>Figura 101 - Vectores do pedestre R660</i> .....	132
<i>Figura 102 - Vectores do pedestre R263</i> .....	132
<i>Figura 103 - Tabela Data: conjunto de dados Amsterdam</i> .....	133
<i>Figura 104 - Desenho de Histograma e do Gráfico de Dispersão: Amsterdam</i> .....	134
<i>Figura 105 - Correção de Dados: Amsterdam</i> .....	134
<i>Figura 106 – Histograma: Amsterdam</i> .....	135
<i>Figura 107 – Gráfico de dispersão: Amsterdam</i> .....	136

*Figura 108 - Data: conjunto de dados Amsterdam* ..... 137

---

## ÍNDICE DE TABELAS

<i>Tabela 1 - Resumo dos Requisitos (Adaptado de : (Shekhar et al., 1997) p.104).....</i>	<i>10</i>
<i>Tabela 2 - Operações Espaço – Temporais (Retirado de: (Gutting, 2007) p. 3) .....</i>	<i>23</i>
<i>Tabela 3 – Construtores para Objectos Móveis (Retirado de: (Silva e Santos, 2010) p. 10) .....</i>	<i>23</i>
<i>Tabela 4 – Requisitos STAR v2.0 .....</i>	<i>40</i>
<i>Tabela 5 - Actores do Sistema.....</i>	<i>57</i>
<i>Tabela 6 - Comparação das funcionalidades entre as duas versões.....</i>	<i>67</i>
<i>Tabela 7 - Limites do PosgreSQL.....</i>	<i>76</i>



# 1. INTRODUÇÃO

## 1.1. Enquadramento e Motivação

Ao longo de vários anos foram desenvolvidos sistemas de recolha de dados com características espaciais. A grande evolução tecnológica tanto dos Sistemas de Informação Geográfica (SIG) bem como das tecnologias de localização (GPS, WI-FI, RFID) contribuíram para um grande aumento da recolha de dados espaciais. Estes dados sobre movimento, associados a entidades ou objectos, estão neste trabalho associados a objectos móveis.

Segundo Liang et al. (2005), os objectos em movimento são um subconjunto dos objectos espaço-temporais. Estes apresentam características espaciais e temporais, sendo que os seus atributos espaciais estão em constante alteração com o decorrer do tempo.

Devido ao grande volume de dados recolhidos e para que sua respectiva análise se tornasse possível, verificou-se a necessidade de conceber e implementar repositórios capazes de armazenar e processar este tipo de dados. Através da análise aos repositórios já existentes, verificou-se que todos estes apresentam algumas limitações, sendo que em todos os repositórios analisados destaca-se a limitação ao nível do tipo de dados que armazenam, pois apesar de existirem várias tecnologias de recolha de dados sobre o movimento (GPS, redes móveis, redes Wi-Fi, Bluetooth, RFID), os sistemas disponíveis só armazenam dados provenientes de uma destas tecnologias (dados sobre o movimento de automóveis, animais ou pedestres recolhidos recorrendo à tecnologia GPS, dados sobre a utilização de redes Wi-Fi, entre muitos outros exemplos), o que limita claramente o armazenamento e análise dos mesmos.

Também decorrente da análise bibliográfica realizada, constatou-se que estes sistemas, além da limitação referida anteriormente, apresentam ainda limitações associadas ao tipo de objectos que são armazenados e ao tipo de elementos que são utilizados para representar o movimento. Os trabalhos revistos habitualmente só analisam dados sobre um tipo de objectos, veículos. (e.g. Nóbrega, et al., 2004; Qajary, 2008; Pelekis et al., 2008), sendo que a representação do movimento dos objectos recorre à utilização de trajectórias (e.g. Liang et al., 2005; Gütting e Ding, 2006), existindo outros elementos de representação que podem ser usados, entre eles, vectores, redes e os fluxos.

Todas as limitações apresentadas anteriormente restringem a análise de vários tipos de dados, sobre outro tipo de objectos, bem como impedem a análise dos mesmos recorrendo a diversos

tipos de representação do movimento.

Identificadas as principais restrições existentes nos vários trabalhos analisados, é fulcral referir que nesta dissertação pretende-se dar continuidade ao trabalho já iniciado por José Mendes (Mendes, 2010) na sua dissertação de mestrado (STAR: Um Sistema de Armazenamento, Análise e Visualização de Dados sobre o Movimento Humano) detalhando os mecanismos de análise de dados sobre objectos móveis e procedendo à sua respectiva implementação. Tal exige que toda a estrutura da base de dados espacial de suporte ao sistema STAR seja revista de forma a suportar os mecanismos de análise espacial requeridos pelo STAR. Além da base de dados espacial, a especificação dos requisitos do sistema tem também de ser revista de forma a incorporar as novas funcionalidades que serão propostas no âmbito desta dissertação de mestrado.

Define-se assim como finalidade deste trabalho, a especificação e a implementação da segunda versão do sistema STAR. Nesta segunda versão é dada ênfase à identificação e implementação de novas funcionalidades para o sistema, tais como:

- Análise e Visualização de dados com recurso a mapas: pontos, trajectórias e vectores direccionais de cada ponto;
- Criação independente de gráficos de análise sobre vários atributos (escolha das variáveis e suas posições, como da frequência de amostragem dos mesmos): Histogramas e Gráficos de Dispersão;
- Importação total de dados relativos a um dado ficheiro para o sistema (independentemente do seu tamanho e do número de atributos);
- Possibilidade de mudança de conexão de base de dados: Base de Dados Local, Base de Dados Remota;
- Adição de várias funcionalidades sobre as representações resultantes das análises e visualizações de dados, bem como dos gráficos: exportar imagens, *zoom in*, *zoom out*, *set default scale*, copiar, formato de exportação, guardar imagem como, mostrar informação do valor dos pontos, etc.
- Correção de Dados de um Ficheiro: possibilidade de eliminar dados importados para o sistema, de forma a reduzir erros existentes nas análises e visualizações dos dados (e.g. velocidades nulas).

## 1.2. Finalidade e Objectivos do Trabalho

De acordo com a finalidade identificada no Enquadramento e Motivação, é possível definir um conjunto de objectivos a satisfazer ao longo da realização deste trabalho, bem como os principais resultados esperados.

Para melhorar a percepção e compreensão de quais os requisitos necessários para atingir os objectivos desta dissertação de mestrado, identifica-se como finalidade deste projecto, formulando-se como sua principal tese, a necessidade de desenvolver um sistema de armazenamento, análise e visualização de dados sobre objectos móveis capaz de representar e armazenar dados com características espaciais e de realizar análises sobre os mesmos. Descrita a principal finalidade desta dissertação de mestrado, destaca-se que a mesma tem como requisito o desenvolvimento da segunda versão do STAR recorrendo unicamente a *Software* aberto (*Open Source*), requisito que não foi possível satisfazer na implementação da primeira versão. De acordo com a finalidade definida é possível identificar um conjunto de objectivos que serão de seguida descritos, bem como os principais resultados e contributos associados à sua concretização.

O primeiro objectivo está associado à pesquisa bibliográfica que é necessário realizar sobre a área de estudo, com o objectivo de conhecer os fundamentos teóricos e a literatura associada aos Sistemas de Informação Geográfica (SIG), aos modelos de bases de dados para objectos móveis (BDE, BDT e Bases de Dados Espaço-Temporais) e, por último, à análise de sistemas similares ao STAR.

Após a pesquisa bibliográfica, e como segundo objectivo do trabalho, o estudo avançará com base no trabalho desenvolvido na primeira versão do STAR. Esta primeira versão será analisada em detalhe, sendo exploradas todas as suas funcionalidades, bem como todas as opções tomadas no seu desenvolvimento. Em particular, será realizada a análise da arquitectura do sistema, os diagramas de casos de uso e o diagrama de classes proposto. É importante referir que todas as opções tomadas na elaboração da versão 1 do STAR serão analisadas e, se necessário, reformuladas, de forma a tornar o sistema uma referência no armazenamento e análise de dados sobre o movimento de objectos móveis. Pretende-se identificar lacunas na especificação, assim como problemas de implementação.

Completa a análise da primeira versão do STAR, o terceiro objectivo a atingir é definir os requisitos da versão 2 do STAR. No âmbito deste objectivo serão identificadas melhorias a implementar, novas funcionalidades e funcionalidades já especificadas mas não implementadas. Pretende-se nesta nova versão melhorar aspectos de desempenho da versão

anterior e dar prioridade à identificação e implementação de novas funcionalidades, principalmente ao nível da análise de dados.

Como quarto e último objectivo, a implementação de todas as funcionalidades identificadas anteriormente e a melhoria dos aspectos menos bem conseguidos na primeira versão do STAR. Esta fase compreenderá uma fase de testes onde será verificado o desempenho do sistema na análise de dados. Na implementação da segunda versão do STAR usar-se-á o *PostgreSQL* com a extensão *PostGIS* para o desenvolvimento/armazenamento da base de dados de objectos móveis. Para a visualização dos dados ao nível geográfico, recorrer-se-á à biblioteca *SharpMap*, sendo que para os restantes gráficos recorrer-se-á à biblioteca *ZedGraph*. Tal permitirá dar cumprimento a uma das principais tarefas desta segunda versão, a utilização de *Software* aberto.

De seguida é descrita toda a estrutura e organização do documento, sendo cada capítulo descrito e contextualizado.

### **1.3. Organização do Documento**

Esta secção apresenta uma breve descrição de cada um dos capítulos que compõem esta dissertação, de forma a facilitar a sua compreensão. No Capítulo 1 (Introdução) é feito o enquadramento do tema em estudo e faz-se uma abordagem genérica à importância e necessidade do desenvolvimento de um sistema de armazenamento, análise e visualização de dados sobre objectos móveis capaz de representar e de armazenar dados com características espaço-temporais e de realizar análises sobre os mesmos. Em seguida são apresentados os objectivos do trabalho, associando a cada objectivo os resultados e contribuições esperadas. Este capítulo culmina com uma breve descrição do conteúdo da dissertação.

O Capítulo 2 (Estado da Arte) começa com a definição e apresentação de conceitos associados aos Sistemas de Informação Geográfica (SIG), sendo detalhados e descritos os seus componentes. De seguida são introduzidos os modelos de bases de dados para objectos móveis, sendo identificados e descritos os modelos existentes, dando particular ênfase às bases de dados espaço-temporais, pois são as que se enquadram neste trabalho. O capítulo prossegue com a descrição e identificação dos elementos de representação de dados espaciais e de objectos móveis. Por fim termina com a apresentação e descrição de aplicações similares ao STAR.



O Capítulo 3 (STAR: Um Sistema de Informação Espaço-Temporal para Objectos Móveis) traduz a componente central realizada no âmbito deste trabalho. Este capítulo começa com uma breve apresentação dos requisitos da segunda versão do STAR, resultado da exploração dos casos de uso e diagrama de classes definidos na versão anterior, identificando aspectos que podem ser melhorados em termos de especificação, bem como problemas de implementação. Posteriormente, é apresentada a arquitectura do sistema e detalhados os componentes que a constituem. Na última secção deste capítulo são descritas as funcionalidades do sistema projectado nesta dissertação, sendo descritas novas funcionalidades, bem como referidos aspectos que melhoram o desempenho em relação à versão anterior.

No capítulo 4 (Implementação do Protótipo do STAR) começa-se por descrever as tecnologias adoptadas para o desenvolvimento do protótipo, bem como outras tecnologias existentes. São destacadas quais as suas principais vantagens bem como suas limitações, justificando o porquê da escolha das mesmas.

De seguida são descritas as funcionalidades desenvolvidas, apresentando excertos da implementação realizada. Todas as funcionalidades serão analisadas justificando as opções tomadas e descrevendo como foram realizadas.

O capítulo 5 (Apresentação e Avaliação do Protótipo Implementado) inicia-se com a apresentação da segunda versão do STAR, descrevendo o processo de instalação e a constituição da sua interface. De seguida são apresentadas e explicadas todas as funcionalidades do protótipo, realizando demonstrações do funcionamento do mesmo (*print screens*). Posteriormente são aprofundadas as análises e visualizações de dados possíveis de realizar sendo novamente descritas e apresentadas as imagens da execução. Nesta secção serão usados dois ficheiros disponibilizados para análise, sendo descritos estes dois conjuntos de dados. Este capítulo termina com a avaliação do trabalho implementado e dos resultados obtidos.

O último capítulo (Capítulo 6 – Considerações finais e perspectivas futuras) elabora uma síntese do projecto de investigação, sendo expostas as principais conclusões deste estudo, bem como perspectivas futuras e aspectos que poderão ser investigados posteriormente e que são decorrentes das principais conclusões deste trabalho.



## **2. REVISÃO DO ESTADO DA ARTE**

O capítulo de revisão do estado da arte pretende enquadrar conceitos teóricos relevantes associados ao tema em desenvolvimento nesta dissertação. Pretende-se enquadrar um conjunto de fundamentos teóricos associado às bases de dados de objectos móveis, sistemas de informação geográfica, características relacionadas com o movimento de objectos móveis e análise de sistemas similares ao STAR.

### **2.1. Armazenamento e Análise de Dados sobre Objectos Móveis**

Os Sistemas de Informação Geográfica, bem como os dispositivos móveis, que utilizam tecnologias Sem Fios ou Sistemas de Posicionamento Global, proliferaram na última década, o que “conduziu à recolha de grandes quantidades de dados espaciais” (Silva e Santos, 2010).

Devido a esta proliferação e às grandes quantidades de dados que são recolhidos, são necessárias bases de dados apropriadas para o seu armazenamento e mecanismos apropriados para a sua análise.

Esta secção está organizada da seguinte forma. Primeiro, a secção 2.1.1 descreve os Sistemas de Informação Geográfica (SIG), sua definição, contextualização de conceitos, definição de componentes e respectiva capacidade de análise espacial.

Esta secção termina com a apresentação dos modelos de bases de dados para objectos móveis, descrevendo e contextualizando os vários tipos existentes, privilegiando a definição e contextualização dos tipos que se enquadram no objecto de estudo em desenvolvimento nesta dissertação, bases de dados espaciais, temporais e espaço-temporais.

#### **2.1.1. Sistemas de Informação Geográfica**

O termo Sistemas de Informação Geográfica (SIG) baseia-se e aplica-se ao trabalho desenvolvido por sistemas que realizam tratamento computacional de dados geográficos devolvendo informações com base nas suas características alfanuméricas bem como na sua localização espacial (Câmara e Davis, 2001).

O estudo específico deste sistema inicia-se através da sua definição e contextualização. É difícil enunciar uma única definição de SIG, pois estes sistemas abrangem conhecimentos multidisciplinares.

Apesar de não existir uma definição *standard* para os SIG, são de seguida apresentadas algumas definições de SIG usualmente utilizadas na comunidade associada a esta área:

“Um conjunto poderoso de ferramentas para recolher, armazenar, aceder, transformar e visualizar dados espaciais no mundo real”, (Burrough, 1986);

“Um sistema de suporte à decisão que integra dados referenciados espacialmente num ambiente de respostas a problemas”, (Cowen, 1988);

“Qualquer conjunto manual ou computacional de procedimentos utilizados para armazenamento e análise de dados geo-referenciados”, (Aronoff, 1989).

Estas definições de SIG demonstram e reflectem a multiplicidade de visões possíveis para estes sistemas. Inicialmente, os SIG foram concebidos como ferramentas de armazenamento, pesquisa e visualização de dados geográficos (Fotheringham e Rogerson, 1994), sendo que estes evoluíram no sentido de se tornarem ferramentas de uso mais vasto (Davis, 2007).

A partir das definições apresentadas é possível indicar as principais características dos SIG que, segundo Davis (2007, referenciado em (Mendes 2010)), qualquer SIG deverá ser capaz de realizar:

- Representar graficamente informações espaciais, associando aos gráficos informações alfanuméricas tradicionais;
- Representar informações gráficas utilizando o modelo vectorial (pontos, linhas e polígonos) e/ou *raster* (matrizes de pixéis);
- Recuperar informações baseadas em critérios alfanuméricos e em relações espaciais topológicas, tais como cruzamento e adjacência;
- Realizar operações de aritmética de áreas, tais como união, intersecção e diferença;
- Limitar acessos e controlar a entrada de dados através da prévia construção de um modelo de dados;
- Recuperar dados geográficas, usando algoritmos de indexação espacial;
- Oferecer recursos para a saída de resultados sob a forma de mapas, gráficos e tabelas, utilizando impressoras e *plotters*;
- Oferecer recursos para o desenvolvimento de aplicações específicas, utilizando linguagens de programação.

Depois de descrito e contextualizado um SIG, segue-se a apresentação da estrutura geral de um SIG. São apresentadas duas propostas que caracterizam diferentes visões de como este sistema é composto, sendo apresentados os vários componentes que determinam as funcionalidades existentes.

Primeiramente apresenta-se a visão de Shekhar *et al.* (1997). De acordo com os autores, os SIG são usados para a recolha, análise e apresentação de informação, descrevendo as propriedades físicas e lógicas do mundo geográfico. Os dados geo-referenciados são dados espaciais que dizem respeito a um local na superfície terrestre. Segundo os mesmos autores, num SIG existem 4 unidades funcionais: entrada de dados (*Data Input*), modelo de dados (*Data Model*), manipulação de dados (*Data Manipulation*) e apresentação de resultados (*Result Presentation*). A partir desta definição é possível indicar as principais funções de cada unidade funcional, sendo apresentado como síntese um resumo dos requisitos e exemplos de uso (Tabela 1) de cada unidade descrita:

- Entrada de Dados – Os dados espaciais ou medidas espaciais são captados por sensores ou sistema de posicionamento global, sendo posteriormente introduzidos no sistema. Este bloco pode realizar procedimentos de validação dos dados, por forma a aumentar a precisão dos mesmos. Esta validação reduz a possibilidade de erros provenientes da captação dos dados, pois aumenta a precisão dos valores introduzidos.
- Modelo de Dados – Um modelo de dados conceptual é um tipo de abstracção de dados que esconde os detalhes de armazenamento dos dados, usando conceitos lógicos (mais fáceis de entender). Este suporta dados de entrada, manipulação e apresentação de resultados. Muitos SIG são organizados como uma colecção de temas. Cada tema normalmente apresenta valores de um atributo do espaço geográfico, podendo no entanto serem combinados dois ou mais atributos no mesmo tema. Um tema deve particionar, decompor, e fragmentar o espaço contínuo para um determinado valor (ou faixa de valores) do atributo. As partições ou fragmentos do espaço dentro de cada tema são frequentemente armazenados na base de dados e podem ser tratados como objectos e entidades.
- Manipulação de Dados – Os dados geográficos são sujeitos a consultas e análises através de várias operações, incluindo pesquisas espaciais e

sobreposições. Estas operações incluem operações geométricas (áreas e intersecções), operações topológicas (existem ou não conexões) e operações métricas (distâncias).

- Apresentação de Resultados – O SIG apresenta os resultados visualmente na forma de mapas, que consistem em imagens gráficas com dados vectoriais ou *raster*, exibidas sobre uma respectiva área, visualização 3D, animação e produção cartográfica.

Unidade Funcional	Requisitos Modelo de Dados	Exemplos
Entrada de Dados	Espaço Contínuo; Consistência e Qualidade dos dados;	Restrições; Modelos de interpolação;
Modelo de Dados	Geometria; Topologia;	Pontos; Curvas; Polígonos; Vectores;
Manipulação de Dados	Conjunto de Operações; Relações Espaciais; Análise de rede;	Topologia; Métrica; Direcção;
Apresentação de Resultados	Representação Visual;	Mapas; Restrições de Visualização;

**Tabela 1 - Resumo dos Requisitos (Adaptado de : (Shekhar et al., 1997) p.104)**

Uma outra visão acerca da estrutura e componentes de um SIG é apresentada por Câmara e Medeiros (2003). De acordo com estes autores um SIG apresenta cinco componentes (Figura 1), estruturados hierarquicamente formando três níveis distintos:

- Nível Interno do Sistema – Encontra-se a componente Sistema de Gestão de Bases de Dados Geográficos (SGBDG), responsável pelo armazenamento e recuperação de dados espaciais.
- Nível Intermédio – É constituído por três componentes: Entrada e integração de dados, Consulta e Análise Espacial, Visualização e *plotters*. Este deve possuir mecanismos de consulta e processamento de dados espaciais. A primeira componente é responsável pela entrada e integração de dados espaciais. Geralmente esses dados são provenientes de dados de campo, fotografias aéreas, digitalização e dados de sensores remotos.

A componente de consulta e análise espacial realiza consultas e análises espaciais sobre os dados introduzidos no sistema (e.g. manutenção e análise de atributos descritivos, análise integrada de dados espaciais e descritivos e formatação de saída, manutenção de dados espaciais). A última componente realiza a visualização das análises e consultas realizadas na componente descrita anteriormente. A visualização consiste em apresentações gráficas, controlo da visualização de mapas, resultados analíticos e *plotters*.

- Nível do Utilizador: Contém a componente Interface Utilizador que permite visualizar mapas e atributos, consultar, manipular e actualizar análise espaciais. Componente mais próximo do utilizador (Nível Utilizador).

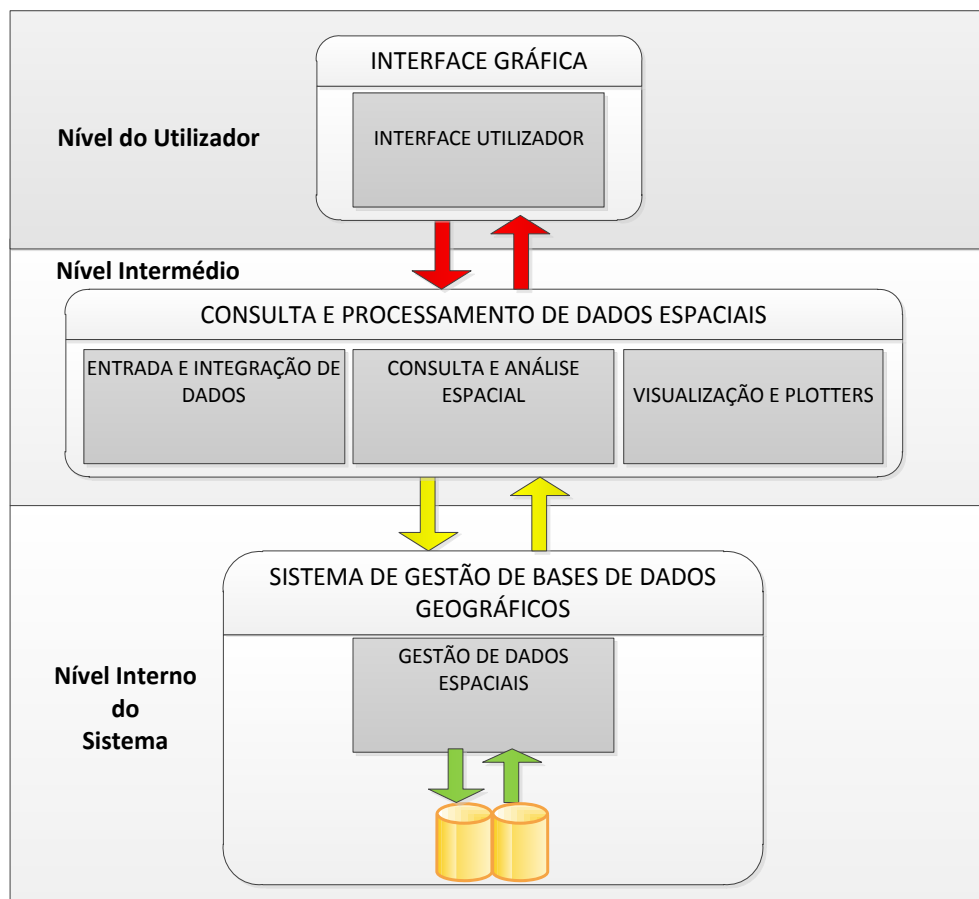


Figura 1 - Componentes de um SIG (Adaptado de: (Câmara e Freitas, 1995) p.2)

### 2.1.2. Modelos de base de Dados para objectos móveis

Os recentes avanços nos sistemas de comunicação conduziram ao importante aparecimento de aplicações de bases de dados sobre objectos móveis. Hoje em dia, há um crescente aumento de dados de mobilidade sobre pessoas, animais e objectos, devido ao avanço tecnológico nos sistemas de captura de movimento. Aplicações de dados de objectos móveis estão a proliferar em várias áreas, entre elas:

- Procura de padrões de mobilidade humana;
- Lazer;
- Optimizações de trajectórias de veículos;
- Implementação de serviços baseados em localização;
- Serviços meteorológicos;
- Serviços turísticos;
- Computação móvel.

Com a evolução das tecnologias de captura e sistemas de comunicação, os sistemas convencionais de gestão de bases de dados são responsáveis por guardar e processar um grande volume de informação.

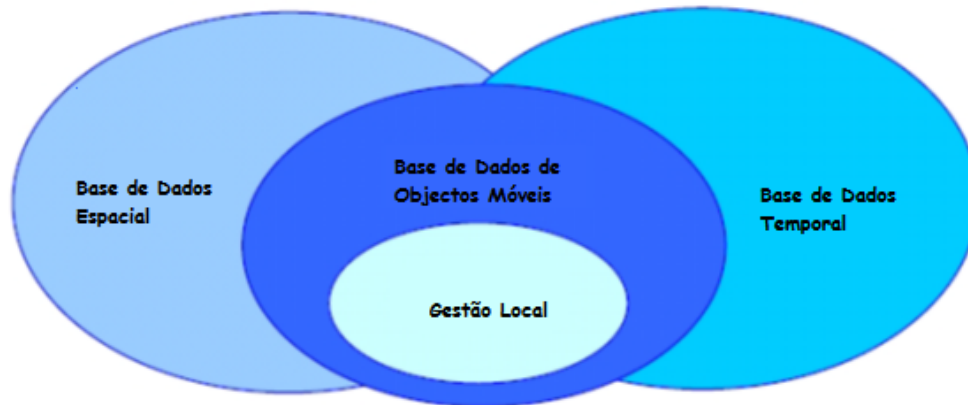
De acordo com Abdul-Kader (2009) os SGBD não estão preparados para lidar com dados em contínua mudança (objectos em movimento), trabalhando com dados estáticos num determinado intervalo de tempo. Esta tecnologia fornece uma base para responder de forma eficiente a consultas sobre objectos em movimento. No entanto, há um conjunto de capacidades que deve ser integrado, adaptado e construído em cima deste para suportar bases de dados de objectos móveis (MOD).

Os sistemas existentes não estão preparados para lidar com a constante alteração dos dados, sendo que a chave para resolver este problema está associada à modelação da localização de objectos em movimento de forma a permitir aos SGBD prever a futura localização destes, usando índices eficientes e formas de consulta aos mesmos (Dodge e Alesheikh, 2005).

Na Figura 2 encontram-se retratadas as relações existentes entre bases de dados de objectos móveis, as bases de dados espaciais e as bases de dados temporais.

De seguida serão descritos os três tipos de bases de dados anteriormente referidos: bases de dados espaciais, bases de dados temporais e bases de dados espaço-temporais. As bases de dados espaço-temporais surgem para dar suporte ao armazenamento e análise de dados sobre objectos móveis.





**Figura 2 - Relação entre Bases de dados Temporais, Espaciais e de Objectos Móveis (Retirado de: (Abdul-Kader, 2009) p. 65)**

### **Bases de Dados Espaciais**

O espaço pode ser definido como um quadro que formaliza relações específicas entre um conjunto de objectos. Dependendo das relações, diferentes modelos de espaço podem ser usados: *set-based space*, *topological space*, *Euclidean space*, *metric space* e *network space* (Worboys, 1995), referenciado em Shekhar et al., (1999).

As bases de dados espaciais são diferentes das bases de dados convencionais em dois aspectos. Primeiramente, é um requisito que estas armazenem tipos de dados complexos, como pontos, linhas e polígonos. O segundo aspecto está relacionado com as funcionalidades necessárias para processar estes tipos de dados. Para tal é necessário o uso de operadores espaciais que são consideravelmente mais sofisticados que os operadores das bases de dados convencionais que realizam o processamento de tipos dados alfanuméricos (Yeung e Hall, 2007).

Segundo Gutting (1994), um sistema de base de dados espaciais não tem uma definição globalmente aceite, sendo que de acordo com a sua visão pessoal este apresenta as seguintes características:

1. Um sistema de base de dados espaciais é um sistema de base de dados.
2. Este possui tipos de dados espaciais, no seu modelo de dados, e consultas aos mesmos.
3. Suporta tipos de dados espaciais na sua implementação, fornecendo, pelo menos, indexação espacial e algoritmos eficientes para *Spatial Joins*.

Um sistema de base de dados espaciais é um sistema de base de dados completo, com recursos adicionais para a manipulação de dados espaciais. Esta habilidade de lidar e relacionar dados espaciais e não espaciais simultaneamente distingue os sistemas de bases de dados espaciais dos outros tipos de sistemas de bases de dados.

O segundo requisito, a introdução de tipos de dados espaciais (e.g. ponto, linha, região), fornece uma abstracção para a modelação da estrutura de entidades geométricas no espaço, bem como das suas relações, propriedades e operações. O sistema de base de dados espaciais é definido na primeira característica apresentada, sendo que sem os tipos de dados espaciais, este não é capaz de realizar a sua principal função.

No terceiro requisito o autor define que a indexação espacial é obrigatória, ou seja, o sistema deve ser capaz de recuperar de uma larga colecção de objectos, em determinado espaço, objectos que estão numa determinada área.

Este deve também suportar relacionamentos de objectos de diferentes classes, através de relações espaciais. Ao ser obrigatória a indexação espacial, esta permite que os dados espaciais presentes em uma BDE sejam acedidos e analisados pela sua localização (coordenadas geográficas) e relações topológicas, que definem a posição das características no espaço relativamente a outros objectos (dados espaciais). Juntos, localização e topologia, permitem a utilização de vários métodos de sobreposição e combinação de características espaciais em mapas, denominado de *spatial join*, para fins de análise espacial.

### **Bases de dados Temporais**

O tempo é essencial para modelar a constante mudança no mundo. Contudo, os sistemas de bases de dados convencionais representam o mundo real como uma fotografia do estado actual (visão estática), sendo inadequado para muitas aplicações onde factos e dados precisam de ser interpretados no contexto de tempo. Este tipo de representações tornam-se desapropriadas para aplicações onde o tempo é uma parte indispensável do sistema (e.g. sistemas de informação médica) (Qiu et al., 1992).

O aspecto mais relevante da interacção entre o tempo e a informação são as múltiplas relações existentes entre os dois conceitos, onde é possível distinguir claramente três relações (Sarace e Theodoulidis, 1995):

- Hora do evento (*Event Time*): é o momento em que ocorre um evento no mundo real, por outras palavras, o momento em que as informações sobre o objecto modelado são válidas.
- Tempo de Transacção (*Transaction Time*): é o momento em que a informação sobre um determinado objecto ou evento é armazenado na base de dados.
- Tempo definido pelo utilizador (*User defined Time*): é o tempo definido e ligado com a percepção de cada utilizador do mundo real.

De acordo com a capacidade de lidar com *event time* e/ou *transaction time*, Snoodgrass e Ahn (1986) identificam quatro tipos de bases de dados:

- *Snapshot databases* – são bases de dados convencionais sem recurso ao tempo. Os estados anteriores da base de dados e os do mundo real são completamente ignorados.
- *Rollback databases* – Suportam unicamente os tempos de transacção. Esta preserva unicamente os estados anteriores da base de dados não suportando os do mundo real, não podendo suportar mudanças nos dados (e.g. corrigir erros em dados passados).
- *Historical databases* – As bases de dados históricas gravam um único estado histórico por relação. Suportam *valid time*, o momento em que a relação era válida. A semântica do tempo válido está ligada à realidade, sendo que assim estas bases de dados representam o conhecimento actual existente sobre o passado.
- *Temporal databases* – Suportam tanto a hora do evento como tempos de transacção. Este tipo de base de dados pode ver informação válida em qualquer momento, propriedade que nenhuma das bases de dados anteriormente apresentadas pode realizar.

As bases de dados temporais foram projectadas justamente para lidar com informações dinâmicas envolvendo o presente, o passado e o futuro. Segundo Pattel (2003) estas “oferecem” tipos de dados temporais e armazenam informações relacionadas com o passado, presente e futuro (e.g. histórico do mercado de acções ou a movimentação de empregados

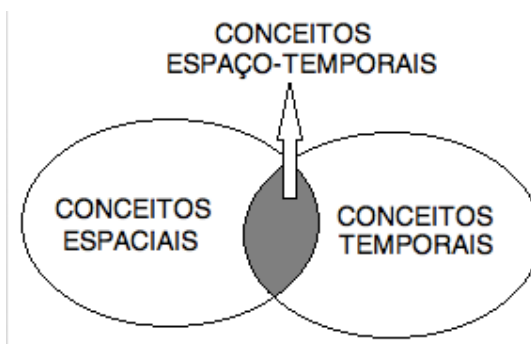
dentro de uma organização). Assim, uma base de dados temporal armazena uma colecção de dados relacionados no tempo.

De acordo com a visão do autor, as bases de dados temporais podem ser definidas como uma base de dados formada pela compilação e armazenamento de dados temporais. A diferença entre dados temporais e não temporais consiste no período de tempo anexado aos dados. Este período expressa quando é que os dados foram considerados válidos ou armazenados no sistema. Os dados armazenados nos sistemas convencionais observam os dados válidos no presente (e.g. “tempo = agora”), sendo que quando os dados são modificados, removidos ou inseridos, o estado da base de dados é alterado formando um novo estado, não sendo possível aceder ao estado anterior pois este não está mais disponível.

Essencialmente, e como conclusão, os dados temporais são formados pela “etiquetagem” de tempo em dados comuns.

### ***Bases de Dados Espaço-Temporais***

As bases de dados espaço-temporais são bases de dados que incorporam conceitos espaciais, temporais e espaço-temporais (Figura 3), captando os aspectos espaciais e temporais dos dados (Zhang, 2005).



**Figura 3 - Relação entre conceitos Espaço-Temporais, Temporais e Espaciais (Adaptado de: (Zhang, 2005))**

De acordo com Dodge e Alesheikh (2005) a importância da incorporação da variável tempo na modelação de dados é amplamente reconhecida por muitas pesquisas, sendo que a maioria das intensões mencionadas estão centradas em torno de domínios não espaciais (e.g. Banca). Embora o conceito de gestão do tempo seja o mesmo, a grande diferença entre as bases de dados espaciais e as bases de dados espaço-temporais é ao que estas se referem. Como os

dados espaciais são espacialmente referenciados, a topologia espacial tem de ser mantida em todos os processos de actualização. Esta manutenção da topologia torna-se crítica quando diferentes versões de um objecto têm que ser acedidas para cada intervalo de tempo correspondente (necessidade de correcção da topologia). Assim, a validade dos dados é um dos principais conceitos, pois se apenas os dados mais recentes são mantidos, excluindo os restantes, a base de dados pode tornar-se estática (e.g. como tirar uma fotografia de um objecto dinâmico). Sendo que muitas aplicações de análise espacial apresentam como requisito viajar através de um histórico dos dados, ou até mesmo apresentar estimativas futuras sobre os mesmos, a exclusão dos dados “não recentes” revela-se problemática.

Conclui-se, portanto, que o tempo como uma dimensão independente é necessária para este tipo de aplicações. Manter um domínio de tempo activo numa base de dados espacial dá origem ao termo base de dados espaço-temporal.

Uma base de dados espaço-temporal ideal, para além de ter as funcionalidades associadas a todas as bases de dados espaciais, apresenta também a capacidade para manter o controlo dos dados alterados. Nestas bases de dados, o mecanismo de actualização das topologias espaciais e temporais mais recentes encontra-se implementado. Além do processo de actualização dos objectos geográficos, mantendo a validade das relações topológicas (tanto temporais como espaciais), as relações entre os atributos são também operacionais (Roshannezhad e Aliasghar, 1996).

Os objectos espaço-temporais estão limitados a uma certa validade temporal. Quando uma nova informação é recolhida, a base de dados tem que ser alterada a fim de acrescentar os novos dados, sendo que por vezes é necessário recriar completamente a base de dados para armazenar o novo estado, o que pode provocar redundância. Como alternativa apresentada, a base de dados pode recorrer à marcação de tempos (*time stamped*) indicando a validade temporal dos objectos (Dodge e Alesheikh, 2005).

A capacidade de realizar versões de bases de dados é essencial para todas as aplicações espaço-temporais, onde o conhecimento do mundo real é incerto e em constante mudança ao longo do tempo (Ott e Swisaczny, 2001).

Há um conjunto de capacidades fundamentais que são necessárias nas bases de dados de objectos móveis e aplicações de bases de dados espaço-temporais. Como referido anteriormente, os SGBD existentes não estão preparados para lidar com dados em constante mudança. Referem-se, de seguida, algumas características de bases de dados que lidam com dados espaço-temporais (Dodge e Alesheikh, 2005):

- Características elementares (tipos de dados básicos, características da linguagem SQL, restrições de integridade declarativa, abstrações de programação, geração automática de indicadores);
- Acesso multi-utilizador e transacções;
- Programação na base de dados (procedimentos armazenados, *triggers*);
- Elementos de administração de base de dados (migração de dados, cópias de controlo, *backup*);
- Portabilidade e escalabilidade;
- Desempenho e questões VLDB (*very large database*): optimização de consultas, estruturas de suporte par optimização de consultas, apoio para processamento analítico, a alocação de espaço em disco, limites de tamanho de dados, conhecimentos de implementações VLDB;
- Bases de dados distribuídas (acesso a múltiplas bases de dados de suporte);
- Tipos de dados complexos (armazenamento de objectos de grande dimensão, extensões e suporte para vários tipos de dados complexos);
- Desenvolvimento de aplicações e interfaces;
- Confiabilidade (recuperação de falhas);
- Questões comerciais (disponibilidade de suporte técnico);
- Visão Espacial.

## 2.2. Representação de dados espaciais e objectos móveis

Os dados espaciais são um termo usado para descrever dados que pertencem ao espaço. Estes dados são geométricos e variáveis consistindo em pontos, linhas, rectângulos, áreas, polígonos, volumes bem como o tempo. Os dados espaciais incluem normalmente “atributos” ou “dados não espaciais” (e.g. o nome de um rio, o tipo de solo encontrado numa região, a velocidade actual durante um determinado intervalo de tempo) (Samet e Araneff, 1994).

De acordo com o mesmo estudo, os dados espaciais podem ser discretos ou contínuos. Enquanto discretos (e.g. pontos num espaço multidimensional), estes podem ser modelados usando técnicas tradicionais de SGBD relacionais. Dados associados a linhas, regiões ou intervalos de tempo são contínuos, significando que estes estendem-se a regiões no espaço ou no tempo.

Nos SIG as primitivas geométricas, ponto, linha e polígono, foram amplamente aceites como elementos de representação baseado em objectos, bem como as *grids* ou *rasters* para a representação do espaço (Orellana et al., 2009). No caso de movimento, não está estabelecida nenhuma caracterização universal dos elementos de representação, sendo que vários investigadores utilizam “metáforas” para analisar o movimento (e.q. “movimento como trajectória”, “movimento como actividade” ou “movimento como equilíbrio”) (Wachowicz et al., 2008).

Esta secção começa pela definição e apresentação dos tipos de dados espaciais existentes, descrevendo cada tipo detalhadamente. São também enumeradas as diferentes formas de representação da informação. De seguida são apresentados os elementos de representação de movimento mais comuns.

Este capítulo termina com a apresentação e representação dos tipos de dados espaço-temporais existentes.

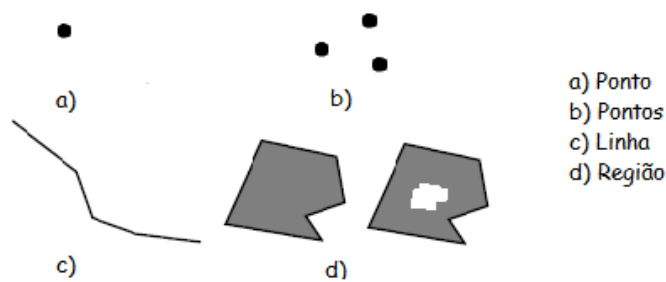
### **2.2.1. Representação de tipos de dados espaciais**

A introdução e definição dos tipos de dados espaciais proporcionam uma abstracção essencial na modelação dos objectos espaciais. Diferentes tipos de dados espaciais podem ser definidos dependendo do modelo escolhido (discreto ou contínuo).

Segundo Gutting (1994), existem duas importantes visões alternativas do que precisa de ser representado numa aplicação de base de dados espaciais:

1. Objectos no espaço: Distinção de entidades distribuídas no espaço, sendo que cada uma tem a sua própria “descrição” geométrica.
2. Espaço: Descrição do próprio espaço. Descrição de todos os pontos do mesmo.

A primeira alternativa permite modelar cidades, rios, florestas, sendo que a segunda descreve a temática dos mapas, por exemplo a partição de um país em distritos. No entanto, é possível conciliar as duas visões, oferecendo tanto o conceito de modelar objectos únicos ou espacialmente relacionados com colecções de objectos. Para modelar objectos únicos (singulares) a abstracção fundamental são o ponto, a linha e a região ou polígono (Figura 4). Para modelar colecções de objectos relacionados espacialmente existem as partições e as redes.



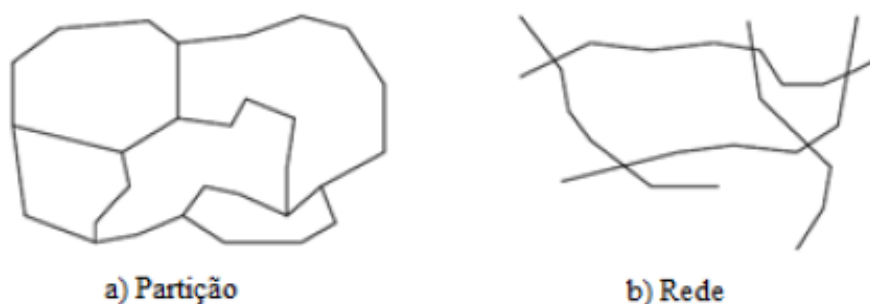
**Figura 4 - Tipos de Dados Espaciais (Adaptado de: (Gutting, 2000))**

Gutting et al. (2000) definem cada um dos tipos representados da seguinte forma:

- Ponto - Cada ponto representa uma posição no plano euclidiano;
- Pontos - Corresponde a um conjunto de pontos;
- Linha - É um conjunto finito de curvas contínuas;
- Região - É um conjunto contínuo de partes disjuntas, denominadas *faces*. Estas podem conter buracos, como demonstra a Figura.

No mesmo sentido, Gutting (1994, referenciado em (Santos e Amaral, 2002), define que a modelação de objectos espaciais passa pela sua abstracção recorrendo a pontos, linhas ou polígonos. Um ponto representa o aspecto geométrico de um objecto para o qual apenas a sua localização no espaço é relevante. Uma linha é constituída por um conjunto de segmentos de recta que permitem a representação de objectos através dos quais a movimentação no espaço é possível. O polígono é uma abstracção de um objecto que tem como aspectos essenciais a sua localização e a sua extensão.

As partições e as redes são as duas principais colecções de objectos referenciados espacialmente (Figura 5).



**Figura 5 - Colecções de Objectos Espaciais (Retirado de: (Gutting, 1994) p.3)**



Uma rede representa um grafo embebido num plano, sendo constituído por um conjunto de pontos e linhas, que formam os nodos da rede e representam as arestas da mesma, respectivamente. Uma partição pode ser considerada como um conjunto de regiões disjuntas que partilham limites, originando relações espaciais entre as mesmas (adjacência).

Os modelos de representação das propriedades geométricas diferem nas capacidades de garantir a formalização de conceitos espaciais (Egenhofer e Herring, 1991).

Existem dois métodos para a representação de dados espaciais: representação baseada em células ou representação baseada em objectos, sendo que a partir de cada um destes métodos obtêm-se modelos com estruturas de dados distintos (Santos, 2001):

- Modelo matricial de resolução fixa (representação por células) – Este representa os dados numa matriz, onde cada elemento espacial possui um valor único, fornecendo informação sobre a sua localização (Davis e Simonett, 1991). Segundo o estudo de Adam e Gangopadhyay (1997), referenciado em (Santos e Amaral, 2002), para além de fornecer a posição geográfica, este também possui um valor referente à área geográfica que representa.
- Modelo vectorial (representação por objectos) – Este representa os dados geográficos por uma série de coordenadas (x,y) ou (x,y,z) (Maguire e Dangermond, 1991).

A representação baseada em células (Figura 6) consiste na divisão do espaço num conjunto de células independentes, que abrangem toda uma zona em análise, sendo que os objectos geográficos que se encontram embebidos no espaço são representados através do conteúdo das mesmas.

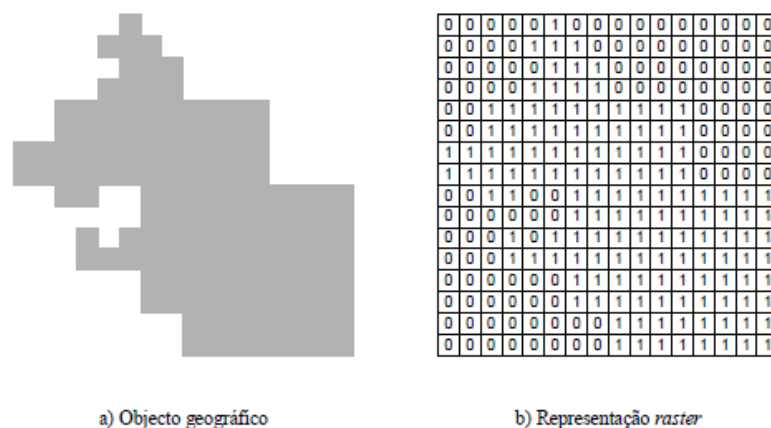
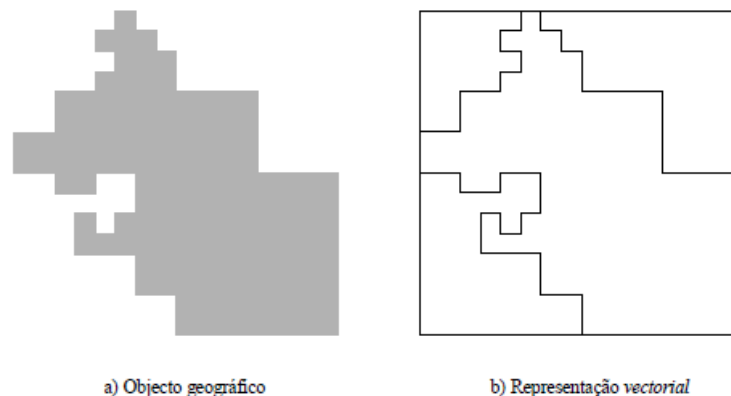


Figura 6 - Representação por células (Adaptado de: (Gatrell, 1991) p.125)

No modelo vectorial (Figura 7) os objectos são representados pelas entidades geográficas (pontos, linhas e polígonos) que os constituem. Estas entidades podem ser combinadas de maneira a representarem objectos compostos (várias unidade geométricas do mesmo tipo) ou objectos complexos (várias unidades geométricas de diferentes tipos) (Adam e Gangopadhyay, 1997).



**Figura 7 - Representação Vectorial (Retirado de: (Santos, 2001) p.18)**

Comparando os dois modelos, o modelo vectorial traduz a realidade com mais exactidão, devido a uma estrutura de armazenamento de informação que representa as coordenadas, que definem acontecimentos. Este modelo apresenta maior complexidade e requer maior capacidade de processamento que o modelo matricial de resolução fixa. (Santos, 2001).

### 2.2.2. Representação de tipos dados Espaço-temporais

Erwig et al. (1999), com o intuito de ampliar o modelo de dados espacial para “captar” o tempo, definem que os tipos de dados espaço-temporais encapsulam geometrias dependentes do tempo com operações apropriadas. Para além dos objectos (dados), os atributos que descrevem mudanças nas geometrias ao longo do tempo são de bastante interesse.

Para alargar o modelo espacial para que este concilie o espaço e o tempo, o estudo apresenta dois novos tipos de dados (Figura 8): *moving point* (mpoint) e *moving region* (mregion). Estes tipos são definidos como mapeamentos do tempo para o espaço Erwig et al. (1999), referenciado em (Silva e Santos, 2010):

$$\begin{aligned} \text{mpoint} &= \text{time} \rightarrow \text{point} \\ \text{mregion} &= \text{time} \rightarrow \text{region} \end{aligned}$$

**Figura 8 - Tipos de dados Espaço - Temporais (Retirado de: (Erwig et al ., 1999))**

Estes novos tipos de dados requerem operações apropriadas com se pode visualizar na Tabela 2.

<b>intersection:</b>	$mpoint \times mregion$	$\rightarrow mpoint$
<b>distance:</b>	$mpoint \times mpoint$	$\rightarrow mreal$
<b>trajectory:</b>	$mpoint$	$\rightarrow line$
<b>deftime:</b>	$mpoint$	$\rightarrow periods$
<b>length:</b>	$line$	$\rightarrow real$
<b>min:</b>	$mreal$	$\rightarrow real$

Tabela 2 - Operações Espaço – Temporais (Retirado de: (Gutting, 2007) p. 3)

Gutting et al. (2000) definem um sistema fechado de operações e um conjunto de tipos móveis utilizando o prefixo “m” para o as designar: mpoint, mpoints, mregion, mint, mstring, mbool. Para definir este conjunto de tipos de dados móveis os autores recorrem a construtores de diversos tipos (Tabela 3).

Type constructor	Signature
int, real, string, bool	$\rightarrow$ Base
point, point, line, region	$\rightarrow$ Spatial
instant	$\rightarrow$ Time
moving, intime	Base $\cup$ Spatial $\rightarrow$ Temporal
range	Base $\cup$ Spatial $\rightarrow$ Range

Tabela 3 – Construtores para Objectos Móveis (Retirado de: (Silva e Santos, 2010) p. 10)

O tipo *moving point* (mpoint) pode representar entidades, tais como veículos, pessoas ou animais que se deslocam em determinada área, enquanto o *moving region* (mregion) pode representar furacões, incêndios florestais, exércitos, etc. Geometricamente os valores dos tipos de dados espaço-temporais estão incorporados num espaço 3D (2D + tempo) (Figura 9) (Gutting, 2007).

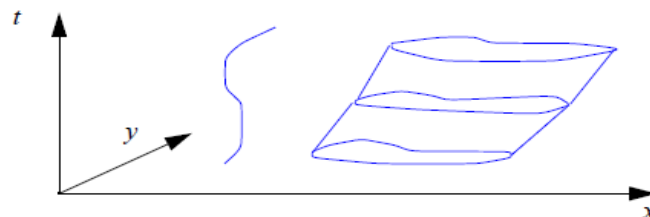


Figura 9 - Representação de Tipos de dados Espaço-Temporais: mpoint e mregion (Retirado de: (Gutting, 2007) p. 3)

### 2.3. Sistemas de Armazenamento e Análise de Dados sobre Objectos Móveis

Os dados sobre objectos móveis podem ser obtidos através de várias tecnologias de localização (como *add-ons* para GSM e infra-estruturas UMTS), tecnologias de posição por satélite (*GPS* e *GLONASS*) e sistemas de posicionamento *indoor* (*Wi-Fi*, *Bluetooth*, entre outros) (Santos et al., 2011). Existindo esta variedade de tecnologias para recolha de dados associados a objectos em movimento, foi desenvolvida uma ampla gama de aplicações para analisar estes dados (Nóbrega et al., 2004), (Wachowicz et al., 2008), (Lee et al., 2005), (Weng et al., 2005), sendo que diferentes modelos de base de dados foram também implementados para suportar estas aplicações (Nóbrega et al., 2004), (Wolfson et al., 1999) - (Praing et al., 2007).

Este capítulo relata um conjunto de aplicações com algumas funcionalidades similares às que se pretendem desenvolver neste projecto, descrevendo as suas componentes, principais opções e motivações.

#### 2.3.1. *Hermes: Aggregative LBS via a Trajectory DB Engine*

Hermes é um sistema proposto por Pelekis et al. (2008), baseado numa linguagem de consulta para uma base de dados de trajectórias, que permite um suporte agregador de *Location based services* (LBS). Segundo estes autores, o que motivou o desenvolvimento deste protótipo foi a noção de que “Quanto maior o conhecimento sobre a trajectória de um utilizador melhor será o aproveitamento dos avanços nos processos de análise espaço-temporal, contribuindo para LBS mais capazes”.

Devido ao rápido crescimento da utilização dos dispositivos móveis e tecnologias de posicionamento, emergiram as bases de dados de objectos móveis, núcleo das pesquisas de bases de dados espaço-temporais (Gutting e Schneider, 2005). Estas representam a componente básica de qualquer aplicação orientada a serviços de localização (LBS).

Segundo Pelekis et al. (2008) os modelos de LBS, apesar de existirem há alguns anos, apresentam serviços limitados, não explorando as capacidades de *software* e os avanços nas pesquisas acerca das bases de dados de objectos móveis. Estes fornecem serviços aos utilizadores móveis tendo apenas em conta o tempo de localização actual e a respectiva velocidade, chegando ao servidor (MOD) como uma sequência de actualizações (Sistla et al., 1997). Pelo facto das aplicações baseadas nos serviços de localização (LBS) lidarem com enormes volumes de dados, o que aumenta o processamento, o desempenho torna-se um

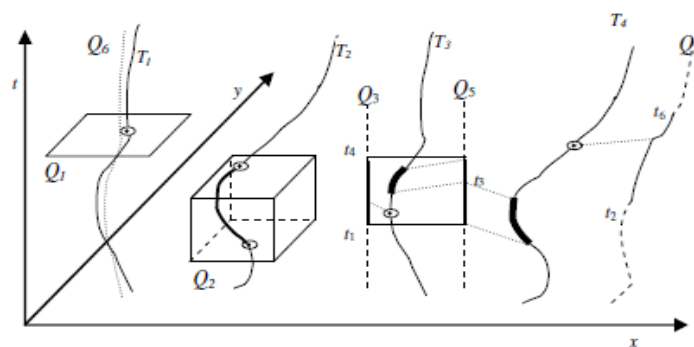
problema significativo. Surge assim a necessidade de aplicar processamentos de consultas eficientes e técnicas de indexação. Para além disto, o modelo descrito tem aplicabilidade limitada em aplicações do mundo real, já que estimativas seguras sobre posições futuras devem também envolver posições passadas.

Devido a isto, Pelekis et al. (2008) adoptaram a ideia de manter uma base de dados de objectos móveis que consiste em trajectórias, isto é, uma sequência de pontos 3D para cada utilizador móvel ( $\langle(x_{i-1}, y_{i-1}, t_{i-1}), \dots, (x_{i-n}, y_{i-n}, t_{i-n})\rangle$ ). A construção da trajectória futura de um utilizador pode ser conseguida através da distribuição de padrões de tráfego ao longo de uma rede de estradas: origem do utilizador (localização do pedido LBS) e destino do mesmo (local de destino) (Wolfson et al, 2002). Posteriormente a trajectória pode ser deduzida usando uma extensão de tempo dependente do algoritmo de *Dijkstra's*.

Desta forma o HERMES, em vez de manter unicamente uma base de dados de objectos móveis, ou seja, armazenar simplesmente dados actuais, mantém uma base de dados de trajectórias, que permite apoiar de forma mais eficiente e inteligente os LBS (Frentzos et al., 2007).

A manutenção de uma base de dados de trajectórias (TD) apresenta várias vantagens. Permite utilizar métodos de acesso especializados para trajectórias (Pfoser et al., 2000), obtendo menores custos de actualização, e fornecendo uma base sólida de processamento de consultas (e.g. suporte de consultas a trajectórias relacionadas) (Figura 10). Para além disto uma TD suporta consultas avançadas sobre trajectórias, incluindo análises baseadas em coordenadas (e.g. distâncias), em trajectórias (e.g. topológicas, baseadas em similaridade) ou a combinação das mesmas (Theodoridis e Benchmark, 2003), (Pfoser et al., 2000).

O sistema HERMES adopta as soluções propostas e demonstra a sua implementação, aproveitando a extensão ORDBMS.

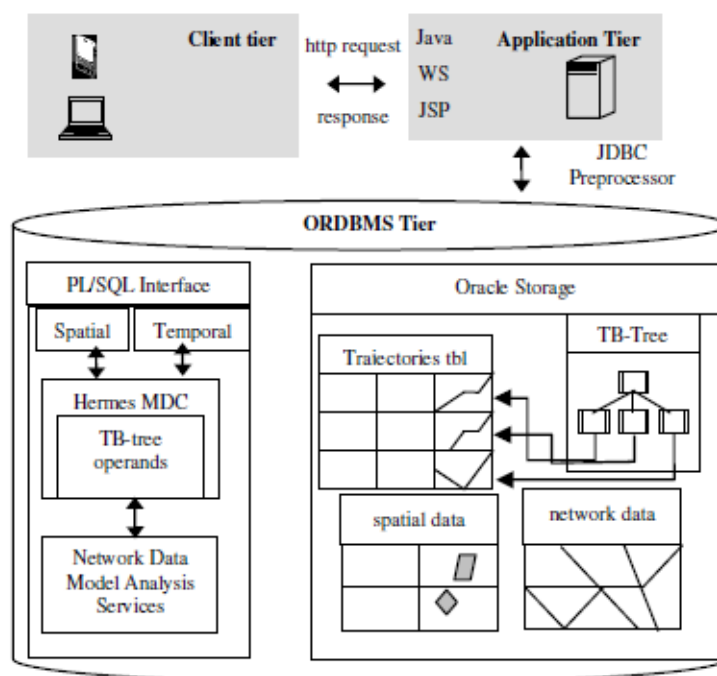


**Figura 10 - Consulta a Base de Dados de Trajectória (Retirado de: (Pelekis et al., 2008) p. 1)**

### Arquitectura HERMES

O principal objectivo do HERMES é apoiar a modelação e a consulta de objectos em constante movimento. O mecanismo da base de dados HERMES é desenvolvido como uma extensão do sistema que fornece funcionalidades de trajectória para o Oracle ORDBMS (Pelekis et al., 2006), (Pelekis et al., 2006). De uma forma mais específica, o sistema define um tipo de dados trajectória e um conjunto de operações<sup>1</sup> como um conjunto de dados Oracle, que é reforçado por uma trajectória especial de preservação de acesso, nomeada *TB-tree* (Pfoser et al., 2000).

A arquitectura HERMES divide-se em três camadas (Figura 11): Camada ORDBMS; Camada Cliente; Camada Aplicação.



**Figura 11 - Arquitectura HERMES (Retirado de: (Pelekis et al., 2008) p. 2)**

De seguida uma das camadas que compõem a arquitectura do HERMES são explicadas separadamente:

<sup>1</sup> Mais informações acerca do tipo de dados de trajectória e do conjunto de operações definidos pelo sistema encontra-se nos seguintes artigos: (Frentzos et al. 2007), (Frentzos et al.,2007), (Pelekis et al.,2007).

- Camada ORDBMS: O *Oracle ORDBMS* Servidor aperfeiçoado com a capacidade de armazenamento e realização de consultas a trajetórias, serve como infra-estrutura de apoio ao LBS. A principal componente desta camada é o *Hermes MDC (moving data cartridge)*<sup>2</sup>.
- Camada Aplicação: Esta é uma camada intermédia desenvolvida com o intuito de tirar partido do Hermes-MDC e do eficiente apoio prestado ao LBS. Para esta camada foi adoptado o ORACLE AS, que pode servir de servidor *web* ou plataforma *wireless*, permitindo além disso a visualização de dados.
- Camada Cliente: Esta camada envolve entre outros, *browsers* para *desktop* ou telemóvel e aplicações *java* que permitem o envio e recepção de solicitações (*http*).

### **Especificações HERMES**

A Figura 12 apresenta duas imagens da interface HERMES (área de trabalho e móvel respectivamente), que ilustram o processo de consulta.

O HERMES consiste em três etapas:

- Especificação da consulta: Definição da consulta, podendo escolher entre várias consultas predefinidas ou formação de uma consulta própria.
- Apresentação de resultados: O sistema processa a consulta e apresenta os resultados no mapa.
- Refinamento da consulta: Para facilitar a interacção, a consulta pode ser aperfeiçoada ou serem explorados novos campos da mesma.

---

<sup>2</sup> Consultar Pelekis et al., (2006) e Oracle 10g (<http://otn.oracle.com/pls/db10g>) para obter informação mais detalhada sobre o funcionamento da componente MDC.

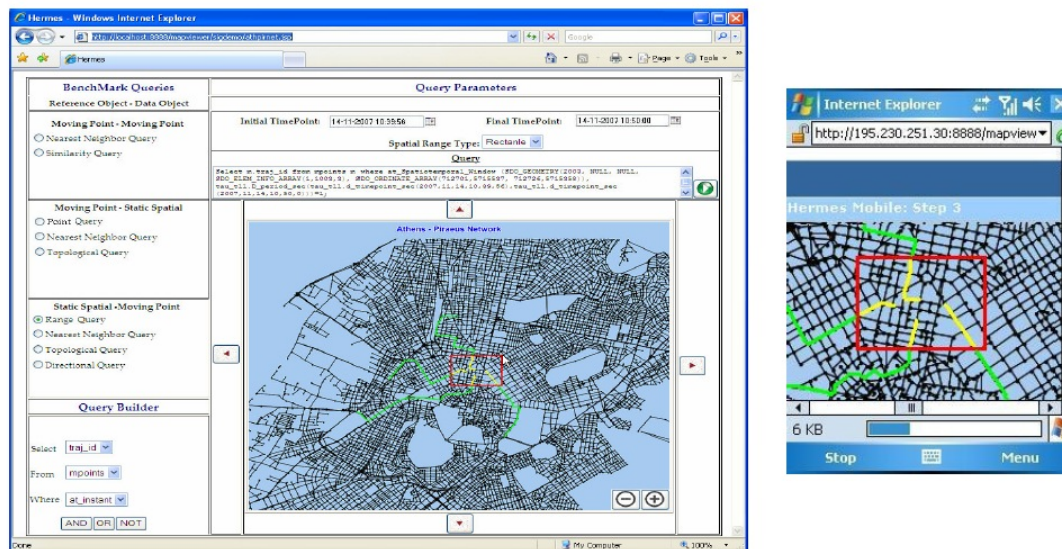


Figura 12 - Imagens da Interface HERMES: desktop e mobile (Retirado de: (Pelekis et al., 2008) p. 3)

### 2.3.2. Design and Implementation of Moving Object Database for Truck Information Systems - DMOTIS

O DMOTIS é uma ferramenta que suporta uma base de dados espaço-temporal para realização de análises, consultas e visualizações a objectos dinâmicos, que mudam a sua localização de uma forma discreta ou contínua no tempo. A capacidade de realizar a gestão e visualização do movimento de camiões e de responder a algumas consultas espaço-temporais sobre o movimento dos mesmos, faz com que esta seja considerada uma ferramenta bastante útil para grandes empresas da área (Qajary e Alesheikh, 2008).

Segundo Qajary e Alesheikh (2008), as base de dados de objectos móveis são um tipo específico de base de dados espaço-temporais. As MOD são o núcleo de sistemas de informação de camiões, sendo que para organizar e elaborar este sistema de uma forma correcta é necessário o estudo dos objectos móveis, suas propriedade e relações. Os



principais componentes de um MOD são os dados de trajectórias, os dados espaciais e não espaciais. Os dados de trajectórias são obtidos através de amostras de movimento<sup>3</sup> captados por GPS. Esta medição não é precisa, tendo um erro associado. Devido a este erro, o mapeamento da posição de um veículo para uma rede de estradas não é uma tarefa trivial, especialmente se a trajectória é imprecisa e as medições são esparsas.

Esta ferramenta utiliza um método discreto para descrever as informações de localização de objectos móveis. Toda a trajectória de um objecto móvel é representada por um conjunto de segmentos de linha no espaço espaço-temporal, sendo que em cada segmento de linha, o movimento de um objecto móvel possui as seguintes propriedades:

- I. Espacialmente o objecto move-se ao longo de uma linha recta.
- II. A velocidade do mesmo é constante.

As coordenadas do ponto inicial e final de cada segmento, juntamente com uma componente temporal são armazenadas numa base de dados. O algoritmo de correspondência do mapa permite mapear as trajectórias em uma rede de estradas (Figura 13).



**Figura 13 - Rede de Estradas do Irão (Retirado de: (Qajary et al., 2008) p. 3)**

---

<sup>3</sup> Amostras de movimento consiste em capturar medidas de posição do movimento em pontos discretos no tempo.

Essencialmente a rede rodoviária capta o aspecto espacial das trajectórias. O modelo de dados conceptual ilustra a abordagem da captura dos dados bem como o armazenamento de trajectória destes por meio de uma rede de movimento, adoptado.

O modelo de dados para trajectória desenvolvido neste estudo apresenta como entidades principais a estrada, segmento-estrada, trajectória, trajectória-segmento, camião e por último o nó (Figura 14).

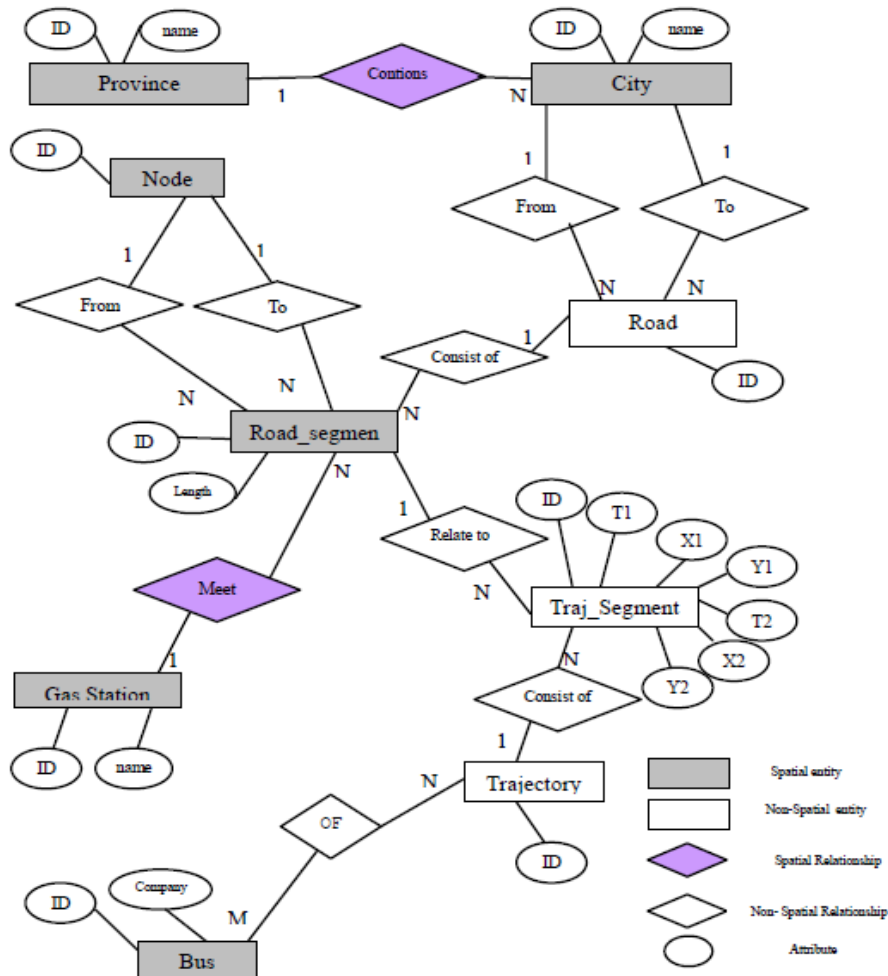


Figura 14 - Modelo de Dados de Trajectória num MOD (Retirado de: (Qajary et al., 2008) p. 4)

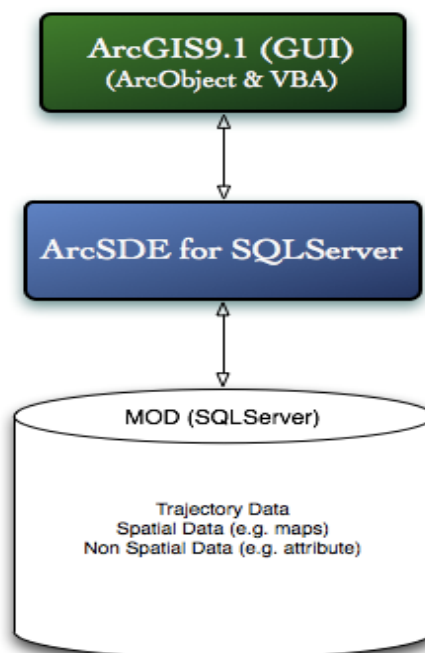
Neste modelo cada estrada (*Road*) consiste em vários segmentos de estrada (*Road\_segmen*). Os segmentos de estrada são entidades espaciais de estradas, entre duas cidades por exemplo e que formam em conjunto uma entidade estrada. O nó (*Node*) é a intersecção entre dois segmentos de estrada que têm duas relações possíveis com cardinalidade de 1 para N: “*To*” e “*From*” (“para” e “de”). Assim as entidades nó e segmento de estrada formam a rede de estradas. As trajectórias estão relacionadas com os segmentos de estrada, e consistem em

segmentos de trajectórias (*Traj\_Segment*) que formam uma relação de 1 para N com a rede de estrada (*Road\_Segment*). O segmento de rede de estradas em conjunto com os nós define a extensão espacial da rede rodoviária, definindo o aspecto espacial das trajectórias.

Outra relação importante existente no modelo apresentado é a relação que existe entre a trajectória e o camião (1 para N). A relação é de 1 para N pois o corte do sinal *GPS*, que localizou o veículo pode não estar disponível durante um período de tempo indeterminado, sendo que quando este é recuperado não existe trajectória anterior e a trajectória foi modificada. Para concluir a análise ao modelo, o aspecto temporal da trajectória é capturado através da atribuição de um tempo de saída e um tempo de unidade para a entidade segmento, sendo que cada segmento de trajectória (*Traj\_Segment*) armazenado possui as coordenadas do ponto inicial e final e o respectivo tempo.

### Arquitectura DMOTIS

Depois de apresentado o modelo de dados da base de dados de objectos móveis e como é captada a trajectória no mesmo é apresentada a arquitectura do DMOTIS (Figura 15). A principal componente da arquitectura DMOTIS consiste numa base de dados para objectos móveis (MOD) que contém dados de trajectória e outros dados espaciais (e.g. estradas, cidades, mapas).



**Figura 15 - Arquitectura DMOTIS (Adaptado de: (Qajary et al., 2008) p. 5)**

O DMOTIS é constituído por três componentes:

- SQLServer™
- ArcSDE™
- ArcGIS™

A primeira componente é o SGBD, que neste sistema é o *SQLServer*, a segunda componente é responsável pela conexão à base de dados de um ambiente *SIG* e por adicionar funcionalidades espaciais ao DBMS adoptado. Por último a terceira componente o *ArcGIS* é a interface com o utilizador adoptada, que permite trabalhar com camadas geográficas e visualizar e trabalhar com os mesmos dados.

### **2.3.3. *SECONDO: An Extensible DBMS Architecture and Prototype***

O SECONDO é uma plataforma extensível de um SGBD, adequado para a construção de protótipos de pesquisa e aprendizagem de arquitecturas/implementações de sistemas de base de dados. Este não tem um modelo de dados fixo, estando acessível para a implementação de novos modelos (Gutting et., 2004).

De acordo com o mesmo artigo, o objectivo deste trabalho é fornecer uma estrutura de sistemas de base de dados genérico, que pode ser preenchida com implementações de diferentes modelos de sistemas de gestão de bases de dados. Deve ser possível implementar modelos relacionais, modelos orientados a objectos, temporais ou *XML*, e acomodar tipos de dados espaciais, objectos móveis, fórmulas químicas, etc. A estratégia usada para este trabalho atingir o objectivo mencionado consistiu nos seguintes pontos:

- Separar as componentes independentes do modelo de dados e os mecanismos do sistema de base de dados das partes de dados dependentes do modelo.
- Apresentar um formalismo para descrever o modelo de dados implementado, com o intuito de ser capaz de fornecer interfaces simples, entre o sistema e o conteúdo.
- Estruturar a implementação de um modelo de dados em uma colecção de módulos de álgebra, sendo que cada um fornece operações e estruturas de dados específicas.

Os trabalhos em torno do sistema decorreram ao longo de vários anos. Foram realizadas experiências e tentativas de implementação de sistemas extensíveis o que ajudou muito a projectar o sistema actual.

O sistema Gral (Gutting, 1989) retratava um SGBD extensível por tipos de dados atómicos e qualquer tipo de operações (incluindo operações de relação). No entanto o modelo de dados central era fixo não coincidindo com a ideia original do SECONDO.

A ideia original do SECONCO<sup>4</sup> era implementar uma componente de *software* para análise e optimização baseada em regras, podendo ser usada com vários modelos de dados e linguagens de consulta, bem como respectiva representação (Gutting, 1993). De acordo com Almeida et al. (2006), o principal objectivo do SECONDO é fornecer uma estrutura para sistemas de bases de dados genérica que pode ser usada com implementações de dados de vários SGBD. Este sistema (Figura 16) baseia-se em três grandes componentes:

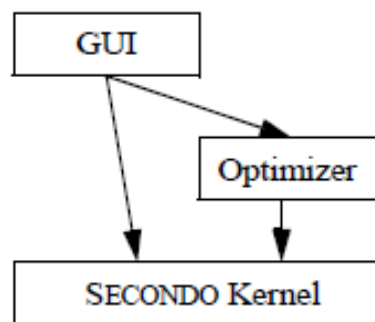


Figura 16 - Componentes SECONDO (Retirado de: (Almeida et al., 2006) p. 3)

- O *Kernel* SECONDO: implementa modelos de dados específicos, sendo extensível por módulos de álgebra. Este fornece processamento de consultas e análises sobre as álgebras implementadas, estando desenvolvido em cima da base de dados BerkleyDB em C++.

---

<sup>4</sup> Para obter informação mais detalhada sobre a ideia original do SECONDO e a projecção do mesmo consultar: Gutting, 1993. *Second-Order Signature: A Tool for Specifying Data Models, Query Processing, and Optimization*,

- *Optimizer*: determina e proporciona a optimização de consultas, implementando a parte essencial do SQL (como por exemplo: línguas), estando desenvolvido em Prolog.
- *GUI*: é uma interface extensível para um sistema de gestão de base de dados extensível, onde novos tipos de dados ou modelos disponibilizam as próprias visualizações ou métodos de visualização. Disponibiliza ainda uma interface especializada para tipos de dados espaciais e objectos móveis, fornecendo uma interface de base de dados espacial bastante sofisticada, incluído animação de objectos em movimento. Está desenvolvido em Java (Gutting,2007).

Os três componentes podem ser usados de diferentes formas, juntos (Figura 16) ou de forma independente. O *Kernel* SECONDO pode ser usado como um sistema de utilizador único ou em modo cliente-servidor. Como um sistema único, este pode ser interligado com uma interface simples (*shell*), ou com o optimizador. Em modo cliente-servidor, o *kernel* pode servir clientes que executam tanto em interface de linha de comando, como interface de utilizador.

No caso do optimizador este pode ser usado separadamente para transformar consultas SQL em planos de consulta executados no SECONDO.

Por último, a componente GUI pode ser usada de forma independente para procurar dados espaciais ou espaço-temporais que residam em arquivos (Gutting, 2007).

### **Kernel SECONDO**

A Figura 17 ilustra os componentes que constituem o Kernel do SECONDO, apresentando a sua arquitectura.

De acordo com o estudo de Almeida et al. (2006), um modelo de dados é implementado como um conjunto de tipos de dados e operações. Estes são agrupados em álgebras.

Um módulo de álgebra fornece um conjunto de construtores tipo, que implementam uma estrutura de dados para cada um destes. Cada registo de um construtor dentro de uma álgebra necessita de um conjunto de funções de apoio.

A componente Kernel actua como gestor de base de dados (Gutting et al., 2004). Este pode avaliar um plano de consulta também denominado por consulta executável. O processamento

das consultas (*Query*) é realizado da seguinte forma: o *Command Manager* recebe uma consulta executável, analisa-a passando o resultado para o processador de consultas (*Query Processor & Catalog*). Este avalia a consulta através da construção de uma árvore de operação, percorre-a invocando operadores provenientes das álgebras existentes (Dicker e Gutting, 2000). Os objectos SECONDO são armazenados e recuperados pelo *Storage Manager* na base de dados, sendo esta componente gerida pelo *Catalog*.

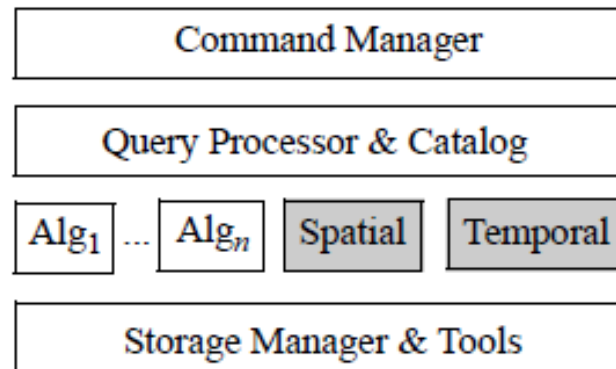


Figura 17 - Arquitectura do Kernel SECONDO (Retirado de: (Almeida et al., 2006) p. 3)

### Optimizer

Segundo Gutting et al. (2004), o Optimizador (*Optimizer*) apresenta vários requisitos:

- Como qualquer optimizador, este precisa de encontrar rapidamente bons planos de consulta.
- Deve funcionar bem para tipos de dados não definidos (*non-standard data types*) com conjuntos de operações.
- O esforço de implementação para apoiar a optimização dos dados não padronizados não deve ser esmagadora, sendo idealmente reduzida.
- Os mecanismos de extensão para o optimizador devem ser simples e compreensíveis.
- Para o ensino, é uma vantagem se o algoritmo de optimização for simples e claro e o código acessível e bem documentado.

Esta componente decompõe o problema geral em três partes, com o código completamente separado: optimização de consultas conjuntas; estimativa de selectividade e implementação de

uma linguagem de consulta. Na Figura 18 é ilustrado o funcionamento do otimizador do SECONDO para uma consulta específica.

```

opt-server >
optimization-input : select [id, line, val(trip atinstant sixfifty) as
pos] from [trains, stations] where [trip passes loc, sname contains
"Mehringdamm", trip present sixfifty]

Computing best Plan ...

Elapsed Time: 1218 ms
Predicate Cost: 0.0994104 ms
Selectivity : 0.119884

Elapsed Time: 62 ms
Predicate Cost: 0.001 ms
Selectivity : 0.00578035
Destination node 7 reached at iteration 5
Height of search tree for boundary is 2

optimization-result : Trains feed project[Id, Line, Trip] Stations
feed project[Loc, SName] filter[(.SName contains
"Mehringdamm")] symmjoin[(.Trip passes ..Loc)] filter[(.Trip
present sixfifty)] extend[Pos: val((.Trip atinstant sixfifty))]
project[Id, Line, Pos] consume

opt-server >

```

Figura 18 - Protocolo do Otimizador SECONDO (Retirado de: (Almeida et al., 2006) p. 4)

## GUI

Por fim a visualização dos dados da consulta é possível na interface gráfica do utilizador (*JavaGUI*). Esta componente comunica com o *kernel* do sistema e o otimizador via TCP/IP, sendo que esta pode ser utilizada por vários utilizadores, exibindo conjuntos de dados diferentes.

A interface do utilizador está dividida em quatro elementos (Figura 19): a área de comando (superior esquerdo), o gestor de objectos (superior direito), a área que contém o utilizador actual (baixo). Existe ainda a barra de progresso (meio) que permite visualizar objectos em movimento, sendo a animação controlada pelos botões que permitem realizar várias funções (aumentar velocidade, parar, etc.). Na área de comando, o utilizador pode introduzir consultas e comandos de controlo *Javagui*. Este interpreta se a consulta é executável ou se esta apresenta uma sintaxe do otimizador. Se estiver na sintaxe do otimizador, este envia a consulta para o mesmo e recebe o plano numa forma executável, sendo este enviado para o Kernel do sistema. O resultado da consulta é posteriormente entregue num formato genérico,



ao *Object Manager* que o armazena, seleccionando um visualizador capaz de o exibir para processamento posterior.

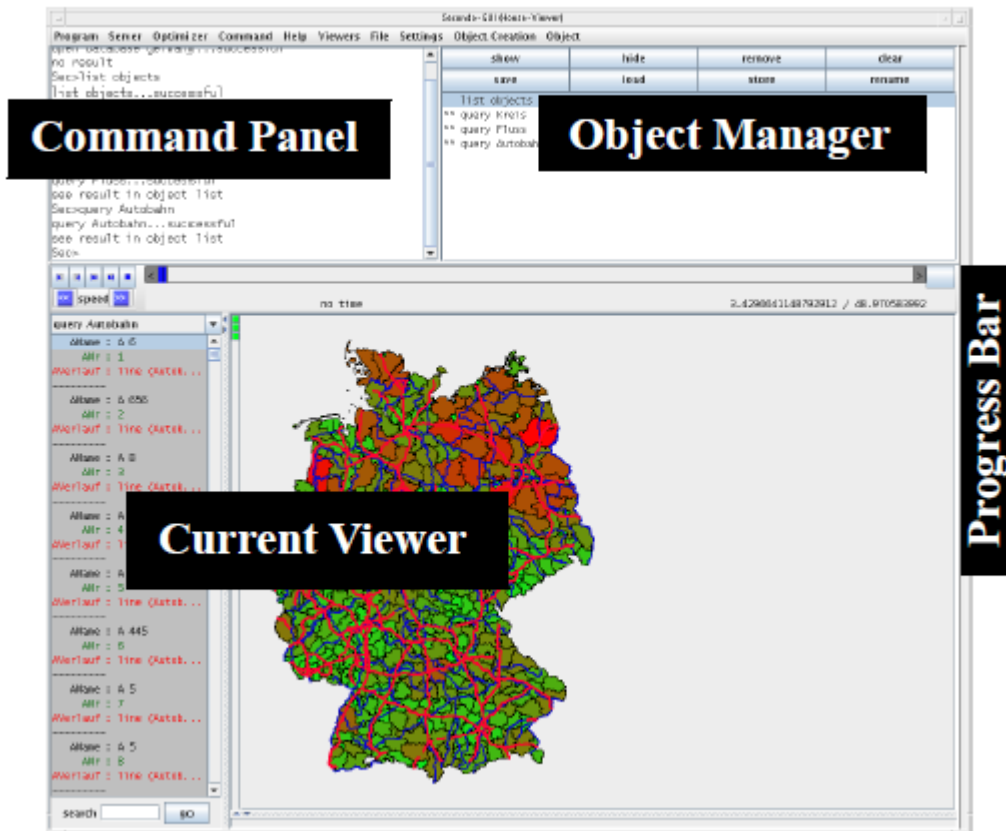


Figura 19 - Interface Gráfica SECONDO (Retirado de: (Gutting et al., 2007) p. 5)



### **3. STAR: UM SISTEMA DE INFORMAÇÃO ESPAÇO-TEMPORAL PARA OBJECTOS MÓVEIS**

O trabalho descrito nesta dissertação pretende dar continuidade ao trabalho iniciado por José Mendes (Mendes, 2010) na sua dissertação de mestrado (STAR v1.0), detalhando os mecanismos de análise de dados sobre objectos móveis e procedendo à sua respectiva implementação. Toda a estrutura de base de dados espaço-temporal de suporte ao sistema teve de ser revista de forma a ser capaz de suportar mecanismos de análise requeridos pelo STAR, bem como, a especificação dos requisitos do mesmo, de forma a incorporar novas funcionalidades.

Neste capítulo apresenta-se a descrição da arquitectura conceptual do sistema e as suas funcionalidades que servem de base à implementação do protótipo da segunda versão do STAR. Todas as opções tomadas no seu desenvolvimento serão documentadas, descrevendo as suas características e principais razões que levaram à sua escolha.

Este capítulo inicia-se com a especificação dos requisitos do novo sistema, seguindo-se a contextualização e introdução de conceitos relativos às opções de arquitectura e de implementação tomadas no desenvolvimento da segunda versão do STAR. Posteriormente é apresentada e descrita a arquitectura do sistema realçando as suas componentes.

Por último são apresentadas as funcionalidades do STAR através dos diagramas de caso de uso, que permitem caracterizar e definir o modo de funcionamento do sistema e é realizada uma comparação entre as funcionalidades implementadas na primeira e segunda versão do STAR, evidenciando quais os avanços entre as duas versões.

#### **3.1. Especificação de Requisitos**

A definição de requisitos é um passo obrigatório no processo de desenvolvimento de um protótipo. Da análise efectuada à primeira versão do STAR, bem como ao estudo de outros sistemas, obtiveram-se 5 requisitos gerais para a implementação do sistema, que se encontram descritos na tabela 4.

Estes requisitos apenas identificam as orientações fundamentais para o bom e eficaz desenvolvimento da segunda versão do STAR, não representando os requisitos do sistema, do ponto de vista das funcionalidades a implementar.

Requisito	Descrição
Arquitectura	De forma a facilitar a escalabilidade e actualização do sistema este deve possuir uma arquitectura modular por camadas.
Estrutura	O sistema no seu todo forma uma estrutura organizada de camadas interligadas entre si.
Independência	Todas as camadas que formam o sistema devem ser independentes e adaptáveis a novas camadas ou estruturas.
Implementação	A implementação de todo o sistema e de todas as camadas que formam a sua arquitectura deve ser realizada recorrendo unicamente a código aberto ( <i>open source</i> ).
Apresentação	Interface intuitiva ( <i>user-friendly</i> ).

Tabela 4 – Requisitos STAR v2.0

### 3.2. Contextualização das opções de desenvolvimento

Para o desenvolvimento do sistema proposto e com o objectivo de cumprir todos os requisitos identificados na secção anterior, várias opções de implementação foram identificadas. Nesta secção são documentadas e contextualizadas algumas dessas opções, descrevendo as suas principais características que levaram à razão da sua escolha.

Esta secção começa com a descrição de sistemas de gestão de base de dados. Posteriormente são detalhadas as bases de dados centralizadas e modelos de arquitectura em camada, definindo quais as vantagens e desvantagens do uso dos mesmos. Por último são identificadas e descritas as representações e análises a objectos móveis existentes.

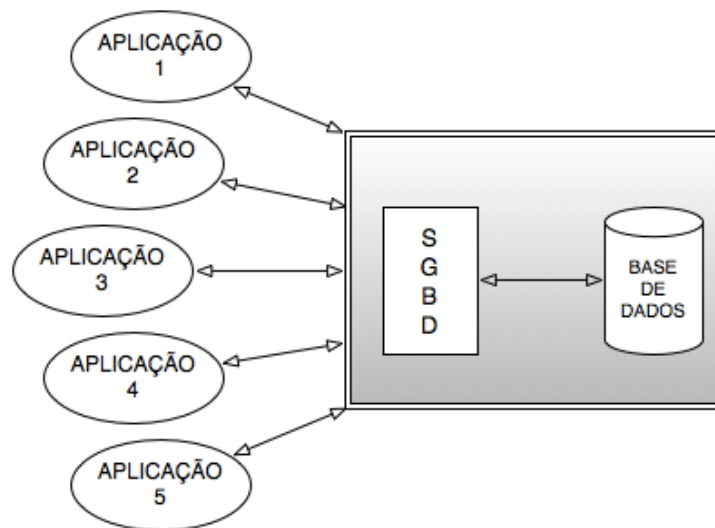
#### 3.2.1. Sistema de Gestão de Base de Dados (SGBD)

Um SGBD pode ser caracterizado como um recurso de *software* composto por programas e utilitários destinados às tarefas voltadas para a completa gestão de uma base de dados. Este representa a principal componente de uma base de dados, pois concentra todos os recursos que definem o que um sistema computacional deve possuir para gerir informações e atender às necessidades de integração das novas tecnologias. A utilização deste sistema tornou a

administração das bases de dados mais segura, fazendo com que as aplicações não tenham acesso directo aos dados armazenados (Figura 20), pois todos os pedidos realizados pelas aplicações passam a ser analisados e processados pelo SGBD (Pessoa, 2001).

De acordo com Schmitz (2010) que cita Lisboa (2002), as principais características de um SGBD incluem:

- Definição da Base de Dados (especificação e descrição dos tipos, estruturas e restrição dos dados);
- Construção da Base de Dados (carga inicial dos dados);
- Manipulação da Base de Dados (operações de inclusão e exclusão de dados);
- Consulta aos Dados (extração de informações armazenadas na Base de Dados).



**Figura 20 - Sistema de Gestão de Base de Dados (Adaptado de: (Schunemann, 2010) p. 2).**

O funcionamento de um SGBD pode ser especificado da seguinte forma (Schunemann, 2010):

- O utilizador emite uma solicitação de acesso;
- O SGBD recebe a solicitação e analisa-a;
- O SGBD inspecciona os esquemas externos (ou sub-esquemas) relacionados com o respectivo utilizador, os mapeamentos entre os três níveis e a definição da estrutura de armazenamento;
- O SGBD realiza as operações solicitadas na base de dados.

De acordo com Shunemann (2010), as principais funções de um SGBD são:

- Cumprimento de integridade;
- Cumprimento de segurança;
- Cópias de segurança (*Backup*) e recuperação;
- Controle de concorrência;
- Dicionário de dados;
- Desempenho;
- Optimização e execução de comandos DML;
- Interação com o sistema de arquivos do sistema operacional;
- Consulta aos Dados (extração de informações armazenadas na Base de Dados).

O SGBD adoptado para o desenvolvimento do protótipo da segunda versão do STAR é o *PostgreSQL* (<http://www.postgresql.org/>) que é descrito de uma forma específica no próximo capítulo desta dissertação.

### **3.2.2. Base de Dados Centralizada**

No desenvolvimento da segunda versão do STAR optou-se por um modelo de arquitectura baseado numa base de dados centralizada.

Numa arquitectura com base de dados centralizada existe uma única base de dados que suporta todo o sistema operacional<sup>5</sup>.

Segundo Ferros (2009), uma base de dados centralizada tem como principal propósito centralizar todo o armazenamento, actualizações, pesquisas e outras alterações num único local tendo em vista vantagens como:

- Padronização na inserção de dados;
- Maior facilidade de manutenção;

---

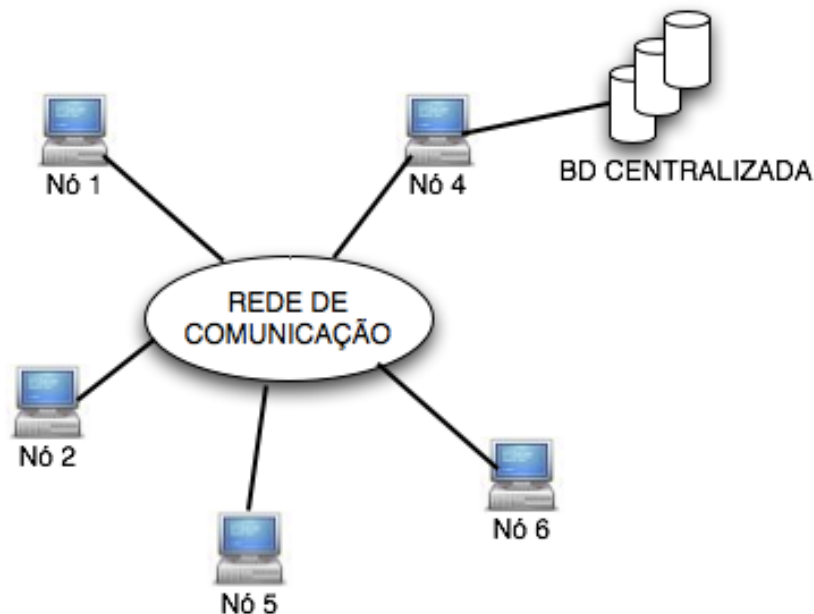
<sup>5</sup> Por sistema operacional entende-se como todas as transacções de actualização (operações de inserção, alteração e eliminação) e consultas de dados (operações de selecção).

- Redução de custos;
- Agilidade na pesquisa de informações.

Contudo, esta também apresenta algumas desvantagens, pois podem surgir problemas que tornam inviável a adoção desta solução (dependendo do cenário onde o sistema se encontra implementado):

- Baixa disponibilidade;
- Comprometimento do desempenho;
- Interdição de acesso caso ocorra falha na ligação ou no sistema central.

As bases de dados centralizadas são geridas por um SGBD e encontram-se apenas em um local da rede (um nó da rede) (Figura 21). No STAR independentemente do nó em que esteja a base de dados, todos os outros nós podem aceder à mesma, bastando alterar a conexão à base de dados.



**Figura 21 - Base de Dados Centralizada**

As principais características apresentadas por este tipo de arquitectura são:

- Dados integrados (mesmo local);
- Redução de dados duplicados (reduz o espaço dos arquivos);
- Facilita a implementação de mecanismos de concorrência entre clientes;

- Não existe inconsistência entre cópias (só 1 cópia);
- Redundância é indesejável;
- DBA (Administração da base de dados) garante a segurança de todo o sistema;
- Responsabilidade centralizada;
- Maior vulnerabilidade a falhas.

### 3.2.3. Arquitectura em Camadas

Nos últimos anos várias pesquisas ocorreram no sentido do aperfeiçoamento de padrões e técnicas para o desenvolvimento de sistemas de maior qualidade e manutenção simplificada (Silva, 2007). As técnicas inicialmente criadas, durante o período em que se utilizavam sistemas estruturados, são a coesão e o acoplamento. Segundo Calçado (2006), a coesão agrupa elementos de acordo com a sua similaridade, o acoplamento caracteriza-se na independência de componentes, onde se pode alterar determinado componente, sem que existam estragos noutros. O uso de camadas (*Layers*) é um padrão arquitectural que ajuda na tarefa de separar responsabilidades, promovendo baixo acoplamento e alta coesão em um sistema. A aplicação destas métricas não está restrita a objectos, trechos de código ou atributos, são também aplicadas à arquitectura de uma aplicação.

Cada camada é responsável por agrupar componentes, pacotes e classes com características comuns, promovendo a coesão e evitando o acoplamento, pois a forma como as camadas comunicam é controlada.

O uso de camadas permite uma série de vantagens e desvantagens (Calçado 2006, referenciado em Silva 2007):

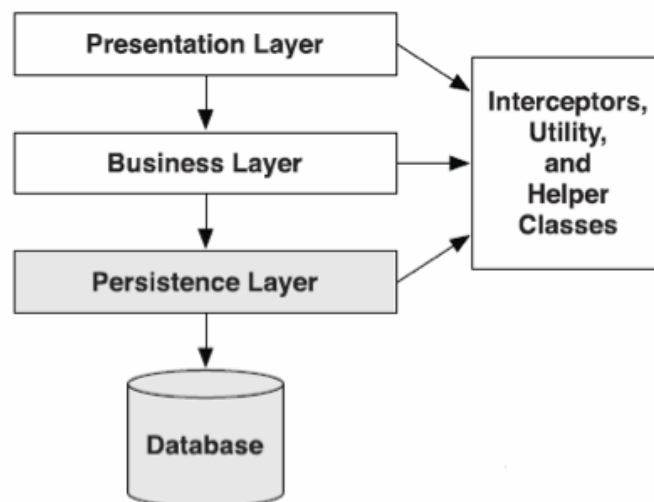
- Redução da complexidade: agrupam componentes e simplificam a comunicação entre os mesmos;
- Favorecem a coesão: agrupam componentes de responsabilidades relacionadas;
- Promovem a reutilização, pois as camadas podem ser utilizadas em outros sistemas, ou podem ser substituídas;
- Reduzem dependência/acoplamento: a regra de comunicação evita dependências directas entre componentes de camadas diferentes;



- Podem complicar sistemas simples, pois nem todos os sistemas exigem o uso de camadas;
- Excesso de camadas tornam a aplicação bastante complexa;
- Limitações tecnológicas.

Uma arquitectura de camadas típica usa três camadas (Bauer e King, 2005): apresentação, lógica de negocio e persistência (Figura 22), sendo que cada camada apresenta as seguintes funcionalidades:

- Camada de Apresentação (*Presentation Layer*) – Esta representa o ponto mais alto e a lógica da interface com o utilizador. O código responsável pela apresentação e controlo do interface forma esta camada;
- Camada de negócio (*Business Layer*) – Geralmente é responsável pela implementação de qualquer regra de negócio ou requisito do sistema, que seria considerado pelo utilizador parte de domínio do problema;
- Camada de persistência (*Persistence Layer*) – É caracterizada por um grupo de classes e componentes responsáveis pela memória de dados e sua recuperação. Esta inclui um modelo das entidades de domínio de negócio;
- Classes de Apoio (*Utility and Helper Classes*) – Todas as aplicações têm um conjunto de assistentes de infra-estrutura ou classes de utilidade, sendo usado por todo o aplicativo. Estes não formam uma camada, são usados pelas mesmas;
- Base de Dados (*Database*) - A bases de dados encontra-se fora do aplicativo. Representa o estado do sistema.



**Figura 22 - Arquitectura em Camadas (Retirado de: (Silva, 2007) p. 18).**

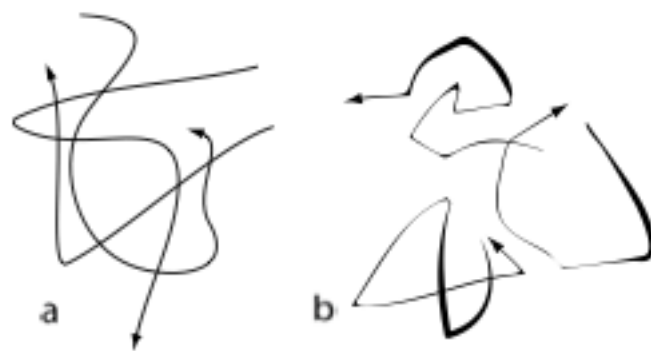
### 3.2.4. Representação e Análise de Objectos Móveis

De acordo com Andrienko et al. (2008), a análise de objectos móveis apresenta alguns problemas:

- Falta de procedimentos que permitem generalizar os resultados;
- Os dados estão normalmente associados a trajectórias;
- A falta de um modelo que caracterize o movimento humano no espaço.

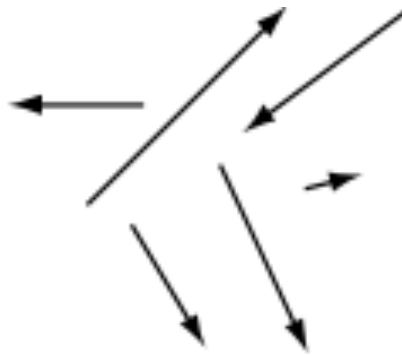
Como referido no capítulo anterior (secção 2.2), não existe nenhuma caracterização globalmente aceite para a representação do movimento, sendo que vários estudos utilizam metáforas para analisar o movimento (Wachowicz et al., 2008). As metáforas usadas incorporam alguns elementos de representação, estando entre os mais mencionados na literatura as trajectórias, os vectores, as redes e os fluxos:

1. Trajectórias: Estas representam o “caminho” espacial de um objecto em movimento num intervalo de tempo fixo (Andrienko et al., 2008). Vários autores usam o termo “rota espaço-temporal” (Hagerstrand, 2002) ou “linha geo-espacial” (Hornsby e Egenhofer, 2002), dando ênfase à componente temporal. Assim a trajectória refere-se a todas as posições ocupadas pelo objecto entre o início e o fim do movimento, sendo que na prática nem todas as posições são conhecidas, podendo ser interpoladas a partir de medições. Este é o elemento mais utilizado nas análises ao movimento, sendo que no entanto apresenta algumas limitações que não devem ser negligenciadas (Figura 23). Por exemplo, a especificação do início e o fim da trajectória é sempre requerida, dificultando análises a movimentos colectivos.



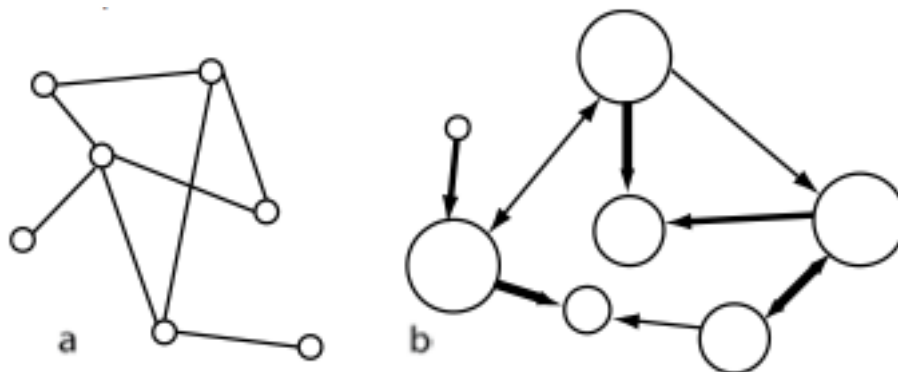
**Figura 23 - Representação de trajectórias. Em (a) direcção das trajectórias e (b) a largura é associada à velocidade (Retirado de: (Orellana et al., 2009)).**

2. **Vectores:** São entidades não dimensionais das quais a velocidade e a posição espaço-temporal são conhecidas, podendo ainda apresentar outros atributos como aceleração ou ângulo de rotação. Vectores de movimento diferem de trajectórias pois estes são uma representação granular de movimento, não estando associados a toda a trajectória do movimento do objecto, portanto as suas posições anteriores e posteriores não são conhecidas (Figura 24).



**Figura 24 - - Representação de Vectores (Retirado de: (Orellana et al., 2009)).**

3. **Redes:** Representam a topologia e geometria do movimento restrito a um sistema interligado linear. Os nós representam origens, destinos ou “passando por”, enquanto os arcos representam os canais físicos do movimento com uma capacidade finita (Figura 25). Os arcos podem ser “directos” ou “indirectos” de acordo com a especificação ou não da direcção do movimento. No primeiro caso, estes podem ser unidireccionais ou bidireccionais, sendo que a largura associada a cada arco representa o custo implícito de um elemento em movimento (Miller e Shaw, 2001).



**Figura 25 - Representação de Redes. Em (a) nós e arcos (b) Raio está associado ao número de nós de conexão e a largura da linha está associada ao custo (Retirado de: (Orellana et al., 2009)).**

4. Fluxos: Representam a quantidade de movimento no espaço. Embora esteja muito relacionado com as redes, os fluxos também têm sido utilizados para representar propriedades de movimento generalizado. O estudo de Tobler (2003) apresenta uma distinção clara entre dois tipos de representações de fluxo: (a) "fluxo discreto" representado por vectores origem e destino e (b) "fluxo contínuo", no qual todo o espaço é ocupado por um campo vectorial que representa a direcção e a velocidade do mesmo. Estes dois tipos correspondem com as representações dos elementos (objectos) e do espaço respectivamente (Figura 26).

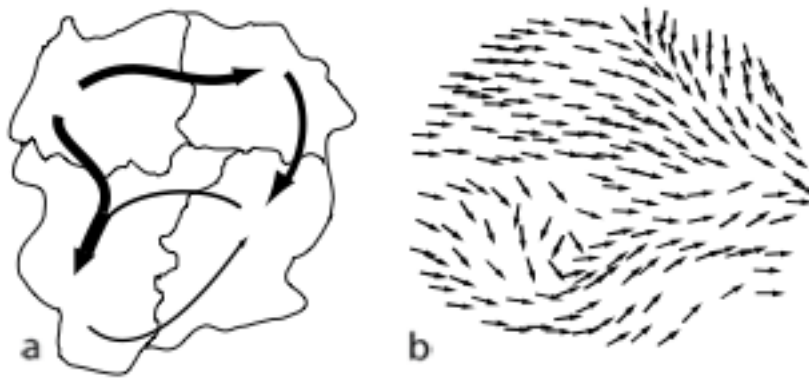


Figura 26 - Representação de Fluxos. Em (a) fluxo discreto (b) fluxo contínuo (Retirado de: (Orellana et al., 2009)).

### 3.3. Arquitectura do Sistema

O desenvolvimento da segunda versão do STAR foi baseado numa arquitectura em três camadas, em que cada camada resolve e cuida de problemas específicos. Existe ainda uma quarta camada (*Helper Classes Layer*) paralela a toda a arquitectura (Figura 27). A organização de um projecto em camadas tem como objectivo permitir que as aplicações sejam desenvolvidas de forma produtiva e com facilidade de manutenção.

Entre outros benefícios, a divisão em camadas independentes permite a substituição da interface gráfica ou do meio de armazenamento dos dados sem afectar as regras de negócio da aplicação. Isto facilita o reuso das classes, maior flexibilidade na escolha de tecnologias, extensibilidade, pois novas funcionalidades podem ser adicionadas sem grande impacto nas já existentes, manutenção e realização de testes às mesmas, pois cada camada pode ser testada separadamente e camadas inteiras podem ser modificadas sem afectar as outras.

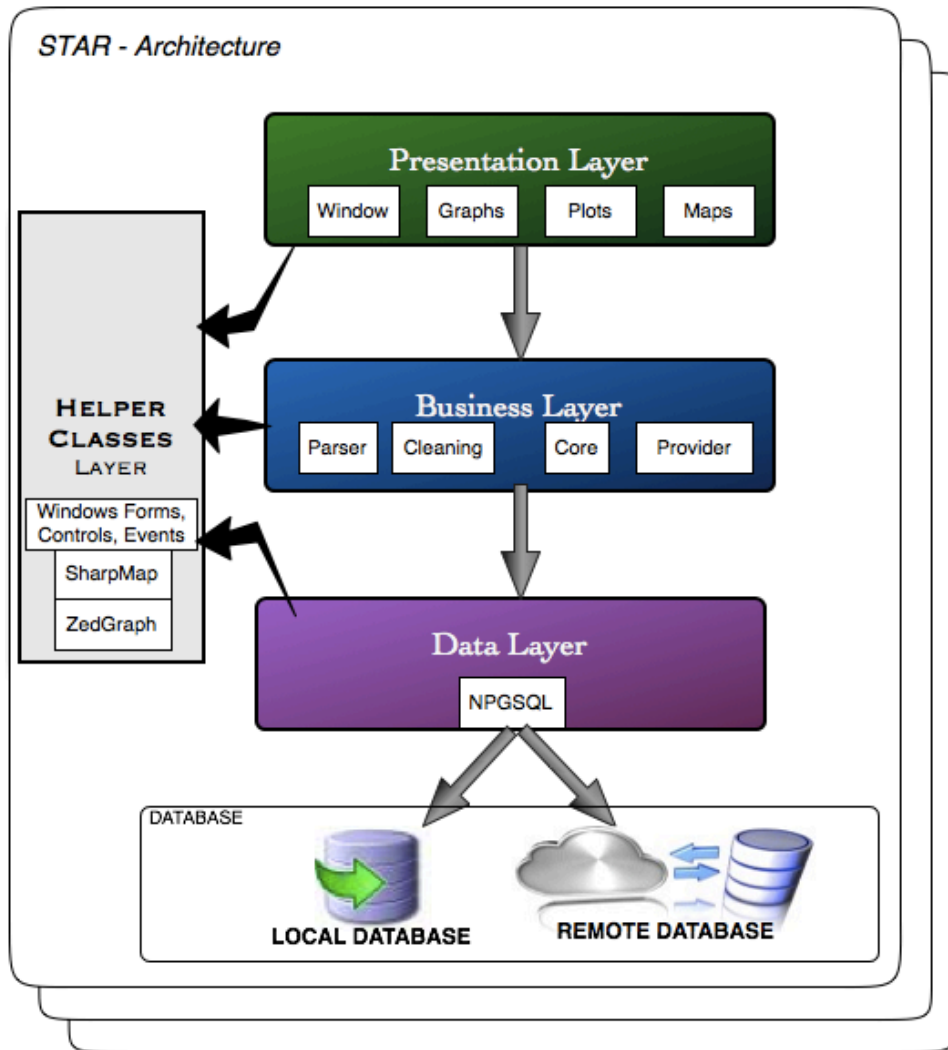


Figura 27 - Arquitectura do Sistema

O sistema encontra-se dividido em camadas separadas prestando diferentes serviços:

- Serviços de Apresentação: Fornecem a interação entre o utilizador e a aplicação.
- Serviços de Negócio: Representam o núcleo da aplicação em termos de processamento.
- Serviços de Dados: Fornecem serviços de persistência.
- Serviços de Apoio: Fornecem uma infra-estrutura de apoio ou classes de utilidade, sendo usado por todo o aplicativo.

A camada de Apoio (*Helper Classes Layer*) é usada por todas as camadas acima apresentadas, representando um auxílio de controlo e diagnóstico do sistema.

A arquitectura apresentada permite aos utilizadores usarem o sistema de formas distintas. Estes podem conectar-se a uma base de dados local, ou seja, no seu próprio computador, ou então conectar-se a uma base de dados remota, de acordo com as suas necessidades.

Para o utilizador este modelo de funcionamento é invisível, pois para este é tudo uma aplicação. De seguida são explicadas cada uma das camadas de uma forma mais detalhada, recorrendo ao uso de imagens para uma melhor percepção do sistema.

### 3.3.1. Camada de Apresentação (*Presentation Layer*)

A camada de apresentação é geralmente chamada de *GUI (Graphical User Interface)*. Esta é implementada sobre a plataforma *.NET*. De uma forma mais específica, a segunda versão do STAR recorre à tecnologia de apresentação *Windows Forms*.

Esta camada tem a função de implementar uma interface de entrada e saída para a interacção da aplicação com o utilizador. Como papel principal destaca-se a validação das informações fornecidas pelo utilizador, sendo responsável pela interacção com os serviços fornecidos pela camada de negócio (*Business Layer*).

Esta camada possui toda a lógica de apresentação, desde a entrada de dados à visualização de resultados. A Figura 28 ilustra a implementação da segunda versão do STAR, identificando todas as camadas desenvolvidas, detalhando de uma forma particular, a camada de apresentação.

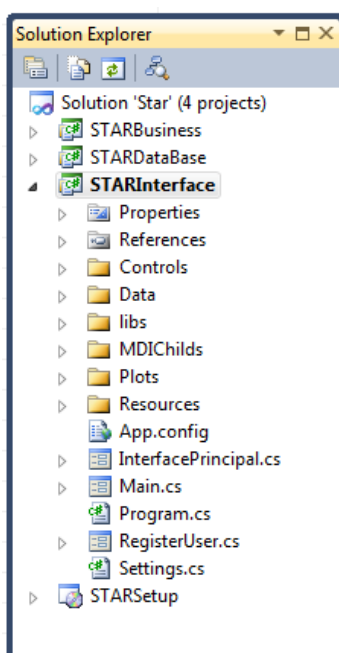


Figura 28 - STAR v2.0 - Camada de Apresentação

### 3.3.2. Camada de Negócio (*Business Layer*)

A segunda camada implementa a lógica de negócio da aplicação, sendo que fornece a *API* para todas as funcionalidades do sistema. Esta foi projectada com o intuito de ser completamente independente das camadas superiores, permitindo a reutilização das suas funcionalidades numa qualquer camada de apresentação.

Operações como o importe, correcção e normalização de dados fornecidos, geração de mapas, histogramas, gráficos de dispersão, cálculo de vectores, trajectórias são realizadas nesta camada.

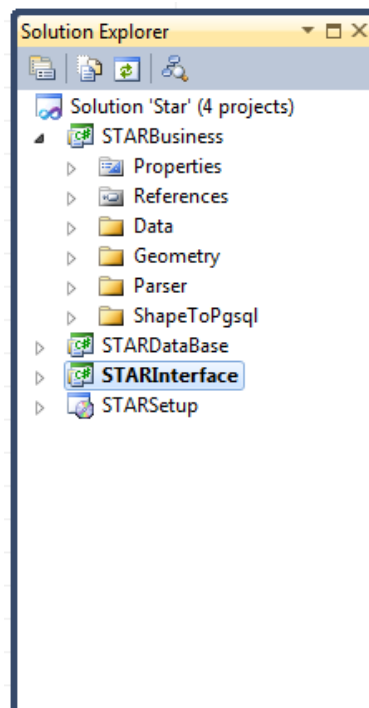


Figura 29 - STAR v2.0 - Camada de Negócio

A Figura 29 ilustra a camada de negócio, representando todos os seus componentes.

### 3.3.3. Camada de Dados (*Data Layer*)

A camada de dados mapeia os dados existentes na base de dados em entidades persistentes em memória.

Tal como a camada de negócio, esta foi projectada para ser independente das camadas superiores. Internamente é responsável por toda a gestão da persistência dos dados em

memória, funcionando como o único elo de ligação com a base de dados, sendo de importância maior para garantir a consistência dos mesmos.

Todas as interações com a base de dados estão assim contidas nesta camada, não devendo existir em nenhuma outra camada da arquitectura apresentada.

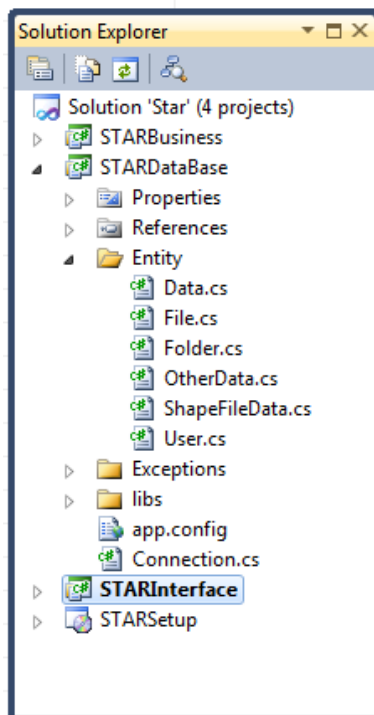


Figura 30 - STAR v2.0 - Camada de Dados

Esta camada é caracterizada por um grupo de classes (entidades) que representam uma abstracção sobre a base de dados (Figura 30).

### 3.3.4. Camada de Apoio (*Helper Classes Layer*)

Todas as aplicações têm um conjunto de assistentes de infra-estrutura ou classes de utilidade, sendo usado por todo o aplicativo. Esta camada apresenta todas as classes e bibliotecas necessárias para a realização de todas as tarefas implementadas na segunda versão do STAR.

Para além disto, a camada de apoio disponibiliza uma infra-estrutura de registo e controlo de todas as opções tomadas pelo utilizador, tendo como resultado um ficheiro de *log* que representa um rasto da execução da aplicação.

Este registo pode ser usado em várias vertentes, sendo que na implementação da segunda versão do STAR, este é implementado com o intuito de conhecer, analisar e validar o comportamento da aplicação aquando da sua execução.



Um ficheiro de log pode ser utilizado para auditoria e diagnóstico de problemas na aplicação. Este ficheiro (Figura 31) é construído com o intuito de ver mais facilmente, ou de uma forma exacta e simples, o que está a acontecer ou já aconteceu na aplicação quando esta foi executada, sem necessidade de recorrer a um *debugger*.

Ficheiro	Editar	Formatar	Ver	Ajuda	
2011-09-22	18:20:34,159	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordy = 5858001,22
2011-09-22	18:20:34,160	DEBUG	STARBusiness.Data.FileItemData	-	Inserindo na base de dados a linha: coordx = 731718,
2011-09-22	18:20:34,160	DEBUG	STARBusiness.Data.FileItemData	-	Type DateTime = 17-08-2006 13:45
2011-09-22	18:20:34,160	DEBUG	STARBusiness.Data.FileItemData	-	Type Speed = 3,57
2011-09-22	18:20:34,160	DEBUG	STARBusiness.Data.FileItemData	-	Type Bearing = 157,64
2011-09-22	18:20:34,160	DEBUG	STARBusiness.Data.FileItemData	-	Type DeviceID = R901
2011-09-22	18:20:34,160	DEBUG	STARDataBase.Entity.Data	-	POINT (731718.71 5858001.22)
2011-09-22	18:20:34,160	DEBUG	STARDataBase.Entity.Data	-	SQL Query: Npgsql.NpgsqlCommand
2011-09-22	18:20:34,162	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordx = 731721,76
2011-09-22	18:20:34,162	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordy = 5857948,5
2011-09-22	18:20:34,162	DEBUG	STARBusiness.Data.FileItemData	-	Inserindo na base de dados a linha: coordx = 731721,
2011-09-22	18:20:34,162	DEBUG	STARBusiness.Data.FileItemData	-	Type DateTime = 17-08-2006 13:46
2011-09-22	18:20:34,162	DEBUG	STARBusiness.Data.FileItemData	-	Type Speed = 5,01
2011-09-22	18:20:34,162	DEBUG	STARBusiness.Data.FileItemData	-	Type Bearing = 176,68
2011-09-22	18:20:34,162	DEBUG	STARBusiness.Data.FileItemData	-	Type DeviceID = R901
2011-09-22	18:20:34,162	DEBUG	STARDataBase.Entity.Data	-	POINT (731721.76 5857948.5)
2011-09-22	18:20:34,162	DEBUG	STARDataBase.Entity.Data	-	SQL Query: Npgsql.NpgsqlCommand
2011-09-22	18:20:34,164	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordx = 731739,84
2011-09-22	18:20:34,164	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordy = 5857937,42
2011-09-22	18:20:34,164	DEBUG	STARBusiness.Data.FileItemData	-	Inserindo na base de dados a linha: coordx = 731739,
2011-09-22	18:20:34,164	DEBUG	STARBusiness.Data.FileItemData	-	Type DateTime = 17-08-2006 13:46
2011-09-22	18:20:34,164	DEBUG	STARBusiness.Data.FileItemData	-	Type Speed = 12,72
2011-09-22	18:20:34,164	DEBUG	STARBusiness.Data.FileItemData	-	Type Bearing = 121,5
2011-09-22	18:20:34,164	DEBUG	STARBusiness.Data.FileItemData	-	Type DeviceID = R901
2011-09-22	18:20:34,164	DEBUG	STARDataBase.Entity.Data	-	POINT (731739.84 5857937.42)
2011-09-22	18:20:34,164	DEBUG	STARDataBase.Entity.Data	-	SQL Query: Npgsql.NpgsqlCommand
2011-09-22	18:20:34,166	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordx = 731756,45
2011-09-22	18:20:34,167	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordy = 5857943,64
2011-09-22	18:20:34,167	DEBUG	STARBusiness.Data.FileItemData	-	Inserindo na base de dados a linha: coordx = 731756,
2011-09-22	18:20:34,167	DEBUG	STARBusiness.Data.FileItemData	-	Type DateTime = 17-08-2006 13:46
2011-09-22	18:20:34,167	DEBUG	STARBusiness.Data.FileItemData	-	Type Speed = 15,88
2011-09-22	18:20:34,167	DEBUG	STARBusiness.Data.FileItemData	-	Type Bearing = 69,47
2011-09-22	18:20:34,167	DEBUG	STARBusiness.Data.FileItemData	-	Type DeviceID = R901
2011-09-22	18:20:34,167	DEBUG	STARDataBase.Entity.Data	-	POINT (731756.45 5857943.64)
2011-09-22	18:20:34,167	DEBUG	STARDataBase.Entity.Data	-	SQL Query: Npgsql.NpgsqlCommand
2011-09-22	18:20:34,169	DEBUG	STARBusiness.Data.FileItemData	-	Type Coordx = 731747,29

Figura 31 - Exemplo do ficheiro Log gerado pelo sistema

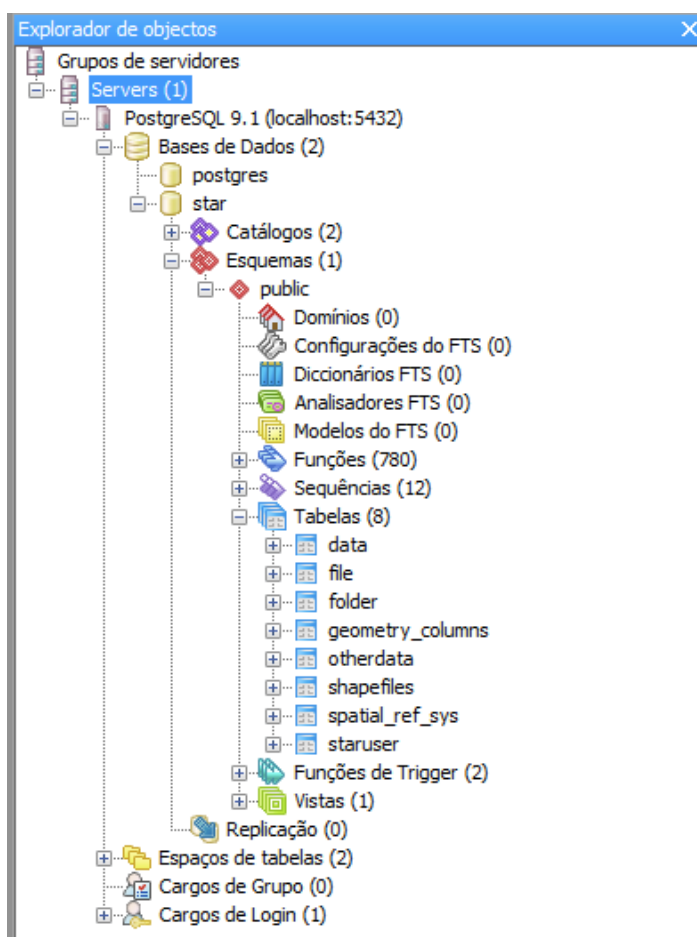
### 3.3.5. Modelo de Dados (*Database*)

O uso de uma modelação de dados orientada a objectos foram propostos no estudo realizado por Worboys et al., (1990), com o objectivo de ultrapassar as limitações dos modelos E-R e as restrições impostas pelos SGBD aquando da utilização de dados espaço-temporais. O modelo de dados da segunda versão do STAR é apresentado com recurso ao diagrama de classes<sup>6</sup>

<sup>6</sup> Diagramas de classes apresentam um conjunto de classes, interfaces e relações. Estes diagramas não são só importantes para visualizar, especificar e documentar modelos estruturais, como também o são para a construção de sistemas executáveis, isto é, que executem determinadas tarefas (Booch et al., 1999).

(Figura 33). Estes diagramas fazem parte das componentes UML, sendo dos mais utilizados de entre os diagramas disponíveis (Dobing e Parsons, 2006).

O modelo de dados da segunda versão do STAR é composto por 8 tabelas (Figura 32). No diagrama de classes (Figura 33) só se encontram apresentadas seis tabelas pois as duas tabelas restantes são automaticamente criadas pelo sistema de base de dados aquando da adição dos componentes espaciais (funções, tipos, entre outros) ao modelo elaborado. Estas duas tabelas (tabelas de metadados), *Geometry\_Columns* e *Spatial\_Ref\_SYS*, são criadas para assegurar a consistência dos dados. Na primeira tabela (*Geometry\_Columns*) são armazenadas as informações das tabelas espaciais e na tabela *Spatial\_Ref\_SYS* são carregadas as informações dos sistemas de coordenadas utilizados pela base de dados.



**Figura 32 - Modelo de Dados Implementado**

De seguida apresenta-se o diagrama de classes da segunda versão do STAR, prosseguindo-se uma breve descrição das classes e relações ilustradas na Figura 33.

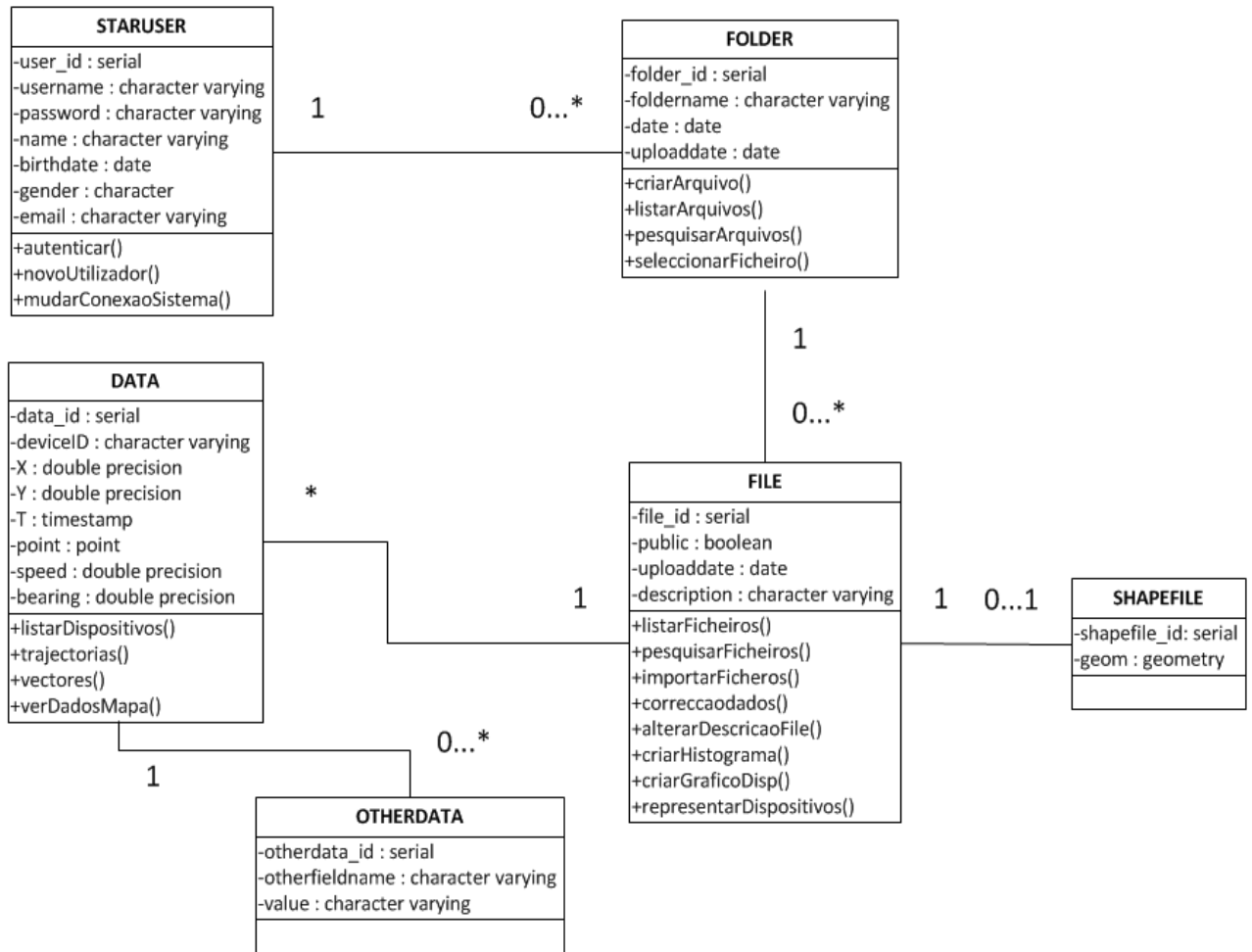


Figura 33 - Diagrama de Classes

A classe *staruser* armazena todos os utilizadores registados no sistema. Todos os utilizadores possuem um *user\_id*, *username*, *password*, *name*, *birthdate*, *gender* e *email*. O *username* e a *password* permitem ao sistema identificar cada utilizador, não permitindo que dois utilizadores tenham os mesmos dados de acesso ao sistema. Como métodos do sistema, a classe *Staruser* possui o registo do novo utilizador, a autenticação no sistema e a alteração da conexão a uma base de dados. O método *autenticar* permite ao utilizador aceder a todas as funcionalidades do STAR, usando o *username* e a *password* para dar entrada no mesmo.

A classe *folder* é constituída pelo atributo identificador *folder\_id* e pelo *foldername*, *date* e *uploaddate*. Esta possui métodos que permitem ao utilizador criar, listar, pesquisar arquivos, e seleccionar ficheiros que pertencem ao arquivo escolhido.

A classe *file* é formada pelo atributo de identificação *file\_id* e pelos atributos *public* (privado ou publico), *uploaddate* e *description*. Como métodos de classe, esta é constituída pelos métodos que permitem ao utilizador listar, pesquisar e importar ficheiros, apresentando ainda

a correcção de dados, alteração da descrição do ficheiro, criação de histogramas e gráficos de dispersão e representação dos dados num mapa.

A classe *data* possui como atributo identificador o *data\_id* e pelo *divideID*, *x*, *y*, *t*, *point*, *speed* e *bearing*. Os atributos *x* e *y* representam as coordenadas dos dados, o *t* representa o tempo, o *speed* e o *bearing* a velocidade e o ângulo, respectivamente. Os métodos desta classe são listar dispositivos e mostrar trajectórias, vectores e representação dos dispositivos no mapa.

A classe *otherdata* possui como atributo de identificação o *otherfile\_id*, apresentando como restantes atributos o *otherfieldname* e o *value*. O *otherfieldname* e o *value* representam o nome do atributo e o seu valor, respectivamente. Assim esta classe armazena todos os atributos de um ficheiro importado que não correspondem a nenhum atributo pertencente à classe *data*, mas que fazem parte do ficheiro importado.

Por fim a classe *shapefile* apresenta o atributo identificador *shapefile\_id*, tendo ainda o atributo *geom*. Esta classe armazena os *shapefiles* que são depois carregados para a criação de mapas.

### 3.4. Funcionalidades do Sistema

Com o objectivo de descrever as funcionalidades presentes no sistema, recorreu-se ao uso de uma componente da linguagem *Unified Modeling Language*<sup>7</sup>, os diagramas de caso de uso<sup>8</sup>, que permitem capturar e documentar os requisitos funcionais de um sistema (Anda e Sjoberg, 2005).

O diagrama de casos de uso descreve a funcionalidade proposta para um novo sistema, onde cada caso de uso representa uma unidade discreta de interacção entre um actor (humano ou

---

<sup>7</sup> UML é uma linguagem normalizada para a especificação, visualização, construção e documentação de artefactos de sistemas de *software* (Rumbaugh et al, 1998).

<sup>8</sup> O diagrama de caso de uso especifica o comportamento do sistema ou parte deste, descrevendo a sequência de acções que o sistema executa para produzir um resultado observável para o actor. Os actores são utilizadores, dispositivos de hardware ou sistemas que podem interagir com o sistema de várias formas.

máquina) e o sistema. De seguida apresenta-se a definição dos actores e casos de uso relevantes para a especificação das funcionalidades do sistema STAR.

### 3.4.1. Actores do Sistema

Neste sistema só existe um tipo de actor, o utilizador, que consoante o seu registo no sistema pode aceder a todas as funcionalidades presentes no sistema, como se pode ver no diagrama de casos de uso interface do sistema (Figura 34). O utilizador quando registado, e depois de efectivada a respectiva autenticação no sistema, tem acesso às seguintes funcionalidades: alteração de dados pessoais, importação de ficheiros, correcção de dados, e um conjunto de funcionalidades referentes a ficheiros, arquivos e dispositivos. Por outro lado, quando não está registado no sistema, pode aceder às secções disponíveis para o seu nível de acesso, que incluem unicamente a secção de registo no sistema.

A informação relativa aos utilizadores é armazenada numa base de dados, referida na secção 3.3.5. Os dados armazenados no momento do registo do utilizador são: o nome do utilizador, o *username*, a senha de acesso, o email, a data de nascimento, e o género. As senhas de acesso são armazenadas utilizando o algoritmo “*Message-Digest 5*” ou MD5<sup>9</sup>, que confere e garante aos utilizadores segurança em relação à sua senha de acesso.

Na tabela 5 é apresentado o único utilizador do sistema e a sua respectiva descrição.

Nome	Descrição
Utilizador	Representa qualquer utilizador do sistema.

Tabela 5 - Actores do Sistema

### 3.4.2. Diagramas de Casos de Uso

Com o objectivo de simplificar a leitura dos diagramas de casos de uso, algumas funcionalidades são agrupadas em casos de uso de alto nível (e.g. Gerir Ficheiros, Gerir

<sup>9</sup> O MD5 é um algoritmo de *hash* de 128 bits unidirecional desenvolvido pela *RSA Data Security, Inc* descrito na RFC 1321 (<http://www.ietf.org/rfc/rfc1321.txt>).

Arquivos, Gerir Dados), sendo os mesmos posteriormente detalhados. De seguida apresentam-se os diagramas de caso de uso que representam as funcionalidades do STAR.

### Interface do Sistema

No caso de uso interface do sistema (Figura 34) estão representadas as principais funcionalidades do STAR.

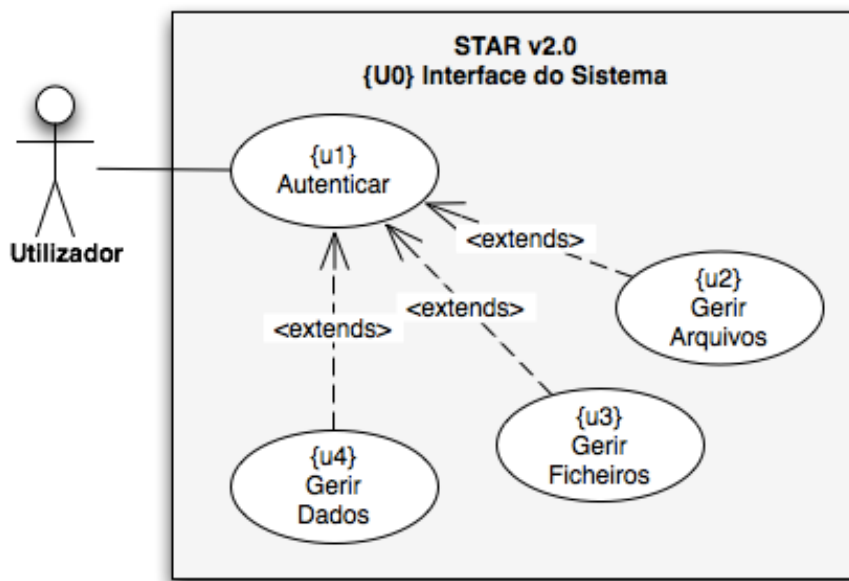


Figura 34 – Diagrama de Casos de Uso: Interface do Sistema

Através da Figura 34 podemos verificar que o sistema interage apenas com um actor, o Utilizador. Este, depois de se autenticar no sistema, pode gerir arquivos, ficheiros e dados. Estes três casos de uso são de alto nível agrupando vários casos de uso referentes a dados, arquivos e ficheiros.

### Autenticar

O caso de uso autenticar (Figura 35) representa a entrada no sistema por parte do Utilizador. Assim este poderá registar-se, se ainda não for um Utilizador do sistema, ou efectuar o *Login* no mesmo se já estiver registado.

Por último, para além das opções acima descritas, o utilizador poderá mudar a conexão à base de dados, sendo que para isso tem de especificar os seguintes campos: *host*, *portnumber*, *password*, *user* e *database name*. Ao fim de especificar a nova conexão, o utilizador pode

gravar a nova ligação sendo armazenada para a próxima entrada no sistema. Sempre que pretender mudar a ligação o utilizador terá de voltar a modificar a conexão à base de dados.

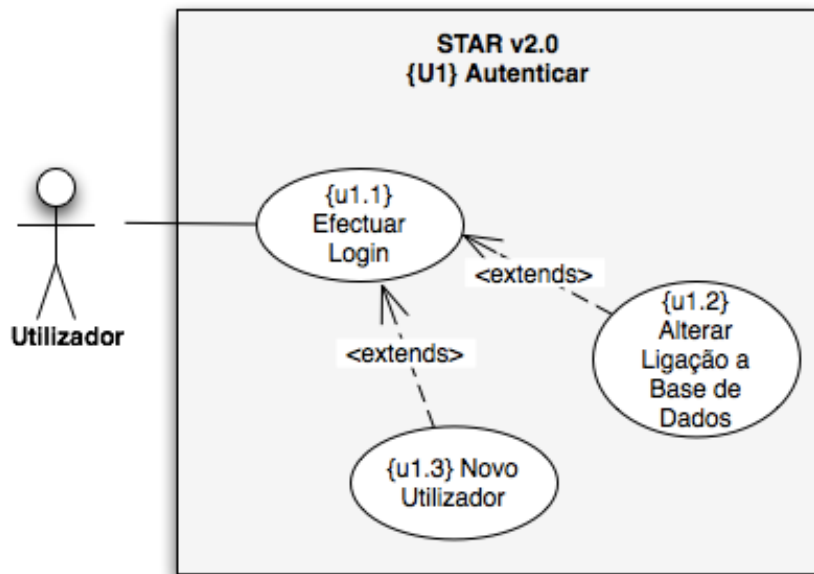


Figura 35 – Diagrama de Casos de uso: Autenticar

### Gerir Arquivos

O caso de uso gerir arquivos (Figura 36) representa um conjunto de funcionalidades sobre a gestão de arquivos efectuada pelo Utilizador.

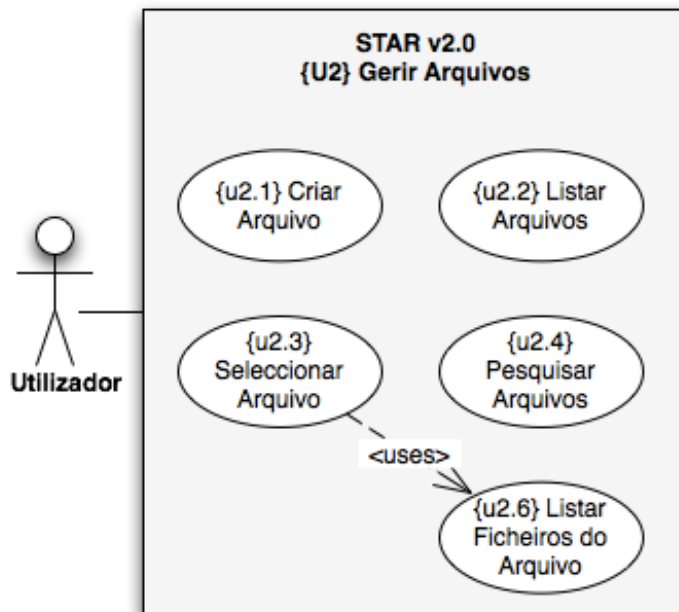


Figura 36 – Diagrama de Casos de Uso: Gerir Arquivos

O Utilizador poderá criar, pesquisar, listar e seleccionar arquivos. Ao usar a funcionalidade Seleccionar arquivo, este lista todos os ficheiros que são visíveis (explicado no caso de uso gerir ficheiros) ao utilizador autenticado e que estão contidos no arquivo seleccionado.

### **Gerir Ficheiros**

O caso de uso gerir ficheiros (Figura 37), representa um conjunto de funcionalidades sobre a gestão dos ficheiros que pode ser efectuada pelo Utilizador.

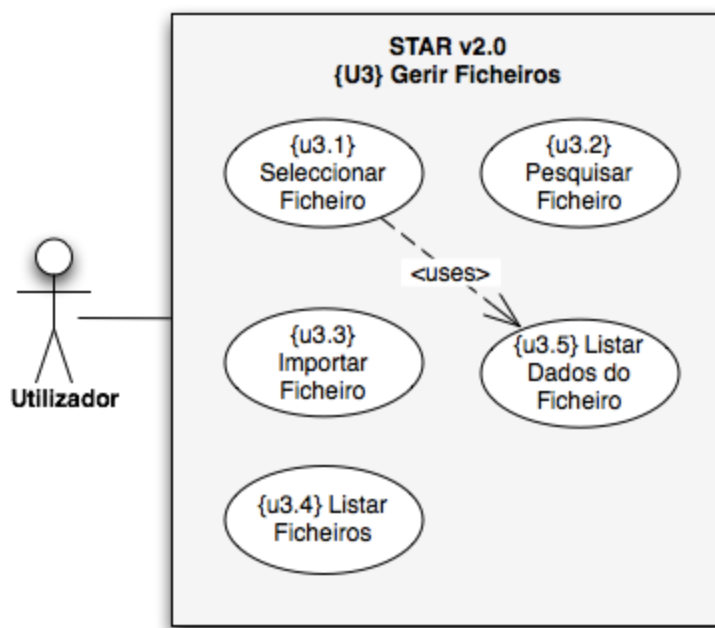


Figura 37 – Diagrama de Casos de Uso: Ficheiros

O Utilizador poderá listar, pesquisar, seleccionar e importar ficheiros. Quando utiliza as funcionalidades de listagem, pesquisa e selecção de ficheiros, o utilizador só terá acesso a ficheiros que foram classificados como públicos por outros utilizadores, ou ficheiros importados para o sistema pelo próprio Utilizador. Todos os outros ficheiros (privados) não são visíveis ao Utilizador.

Todos os ficheiros importados para o sistema estão classificados como: privados ou públicos. Sendo que cada uma das classificações possibilita, ou não, o uso dos ficheiros por todos os utilizadores, ou somente pelo utilizador que realizou a importação do mesmo.

Ao usar a funcionalidade seleccionar ficheiro, este listará todos os dados (entidades móveis) que fazem parte deste, podendo, se assim entender, realizar análises a estes dados, sendo esta outra funcionalidade do sistema abaixo descrita (Caso de uso: Gerir Dados).



### **Importar Ficheiro**

A funcionalidade importar ficheiro (Figura 38) representa para além da criação do ficheiro no sistema o importe de dados para o mesmo.

Para a criação de um novo ficheiro no sistema, o Utilizador terá de seleccionar a directoria para a qual pretende importar o ficheiro e qual o arquivo a que pretende adicioná-lo. De seguida, é obrigatório realizar a classificação do mesmo: privado se pretende ser o único a visualizar e aceder ao ficheiro, ou público se todos os utilizadores do sistema podem ter acesso ao ficheiro e seus dados. Esta funcionalidade não é iniciada enquanto o utilizador não efectivar os três passos anteriormente descritos: seleccionar a directoria do ficheiro (3.3.1), seleccionar o arquivo (3.3.3) e classificar o ficheiro a importar (3.3.2).

Posteriormente, o Utilizador poderá indicar (seleccionar) um mapa (3.3.5) que pretende que esteja “associado” ao ficheiro a importar. Esta selecção não é obrigatória para o funcionamento da funcionalidade, pois só altera a forma como são realizadas as análises e visualizações dos dados (Caso de uso: Gerir Dados), com ou sem recurso a mapa.

Por último, o Utilizador introduz as posições dos atributos dos dados do respectivo ficheiro (introduzir posições ficheiro, 3.3.4). Para esta introdução o Utilizador poderá recorrer à funcionalidade inferir posições ficheiro (3.3.7), que automaticamente infere as posições dos atributos do ficheiro indicado. Se por alguma razão o Utilizador pretender mudar alguma posição resultante desta funcionalidade, pode fazê-lo manualmente corrigindo posições do ficheiro (3.3.6). Este pode alterar as posições e os nomes dos atributos, se assim o entender.

A introdução das posições dos atributos no ficheiro é obrigatória, uma vez que a mesma garante o bom funcionamento desta funcionalidade, tanto na inserção dos dados no sistema, como nas restantes funcionalidades relativas aos dados.

Para indicar as posições dos atributos no ficheiro, o utilizador tem um conjunto de atributos pré-definidos que foram identificados como mais comuns (ou mais utilizados) em ficheiros com dados de objectos móveis. O utilizador poderá acrescentar novos atributos e suas respectivas posições ou retirar atributos. Este procedimento será posteriormente mais detalhado na explicação da implementação realizada (Secção 4.2.2) e na apresentação das funcionalidades (Secção 5.2.3).

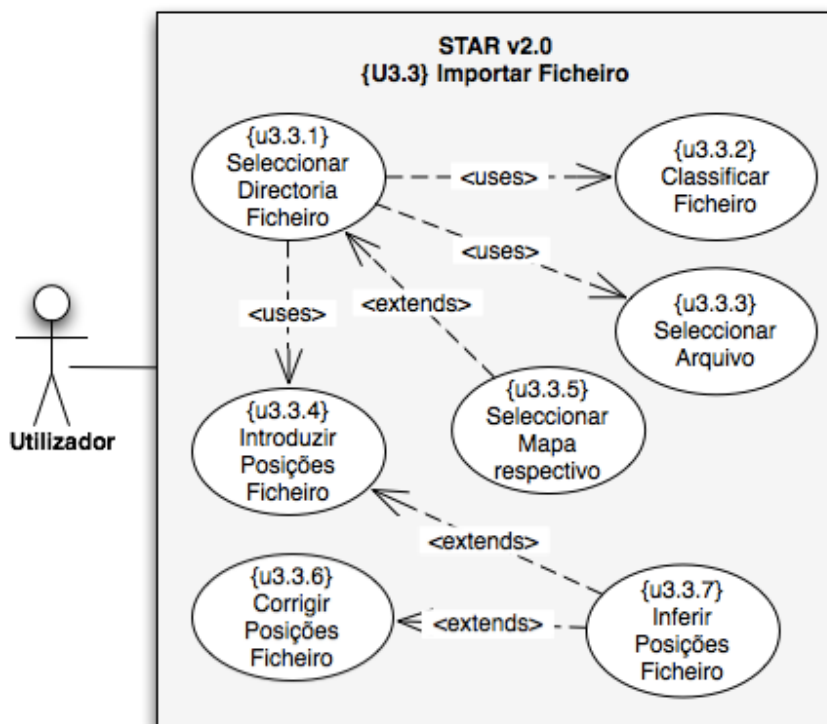


Figura 38 – Diagrama de Casos de Uso: Importar Ficheiro

**Gerir Dados**

O caso de uso gerir dados (Figura 39) integra o conjunto de funcionalidades que podem ser realizadas sobre os dados relativos a cada entidade móvel e que se encontram associados a determinado ficheiro.

O Utilizador poderá listar e seleccionar dados relativos a um determinado ficheiro podendo realizar análises, correcções e visualizações dos mesmos.

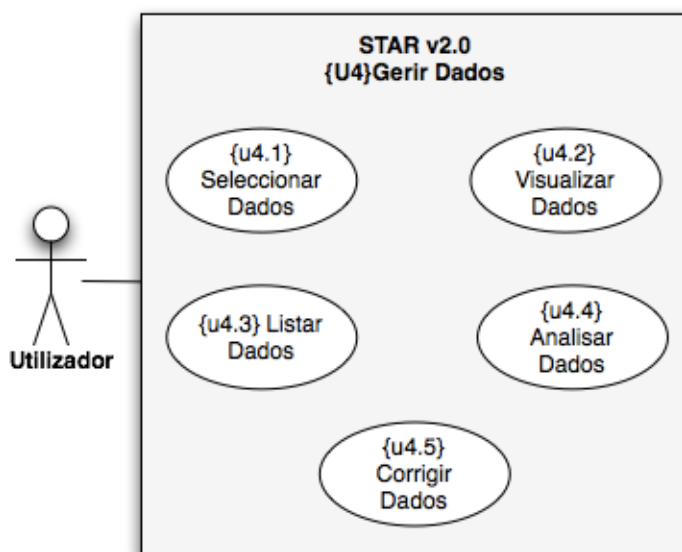


Figura 39 - Caso de Uso: Gerir Dados

### ***Visualizar Dados***

O caso de uso visualizar dados (Figura 40) representa um conjunto de funcionalidades sobre a visualização de dados que podem ser efectuadas pelo Utilizador.

O Utilizador poderá visualizar trajectórias, vectores, os dados no mapa e os dados em uma tabela. Cada uma destas funcionalidades, excepto a visualização dos dados numa tabela, recorre à criação de um mapa (se este for introduzido ao importar ficheiros) para a respectiva representação. Caso não tenha sido importado nenhum mapa, a representação decorre em tela branca.

A funcionalidade visualizar trajectórias (4.2.1) apresenta as trajectórias de todos os registos seleccionados pelo Utilizador, na forma de uma ou várias linhas (trajectória), dependendo do número de registos seleccionados. A funcionalidade visualizar dados no mapa (4.2.2) consiste na representação dos dados seleccionados pelo Utilizador na forma de pontos. A funcionalidade visualizar vectores (4.2.3) necessita da realização de cálculos das direcções dos dados seleccionados (4.2.4), para representar os vectores direccionais de cada registo (posição) seleccionado. Este é representado sob a forma de um ponto e uma seta, sendo que o ponto refere a posição do objecto móvel e a seta indica a direcção seguida pelo objecto. Na secção 4.2.3 será detalhado como é realizado este cálculo da direcção.

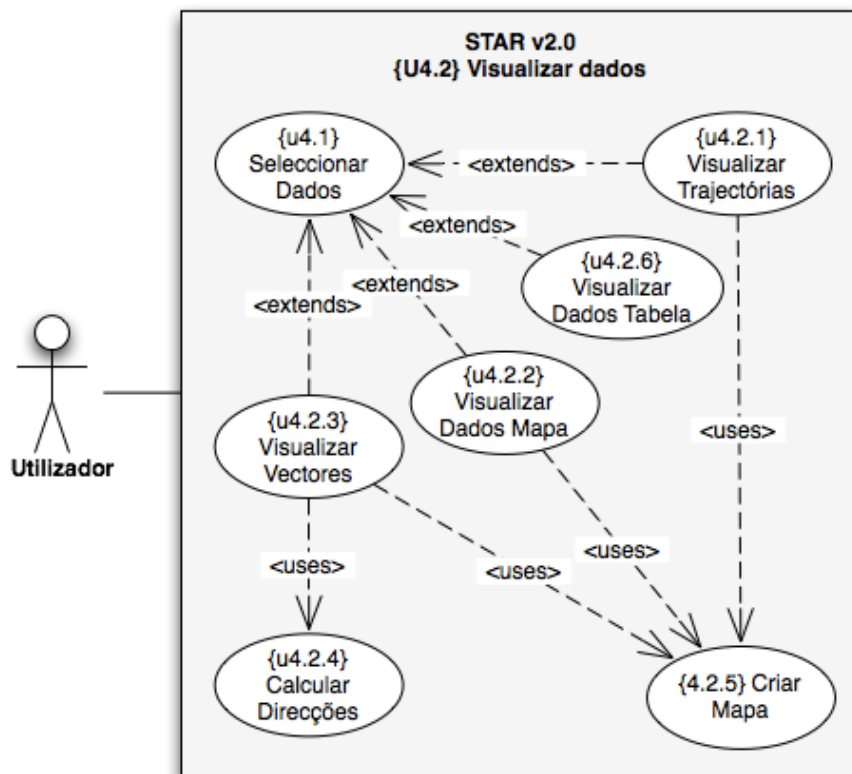


Figura 40 - Caso de Uso: Visualizar Dados

As funcionalidades referidas anteriormente utilizam a funcionalidade seleccionar dados para trabalhar sobre os dados seleccionados, e no seu resultado final é possível efectuar *Zoom* e *Pan* a todas as representações obtidas. Todos os pontos representados bem como o mapa podem ser ampliados ou reduzidos de acordo com as pretensões do utilizador, podendo este mover o mapa para melhor percepção de todas as representações. São também apresentadas informações acerca do valor das coordenadas de cada ponto ou localização no mapa.

Por último a funcionalidade visualizar dados num tabela (4.2.6), apresenta os dados de todos os registos seleccionados pelo Utilizador, na forma de uma tabela. Esta tabela ilustra os dados (entidades móveis seleccionadas) contidos na Tabela *data* da BD desenvolvida (*deviceID*, *X*, *Y*, *DateTime*, *Speed*, *Bearing*), podendo o Utilizador ordenar por ordem crescente ou decrescente qualquer coluna desta tabela.

### **Analisar Dados**

Esta funcionalidade (Figura 41 – Caso de Uso: Analisar Dados) representa um conjunto de funcionalidades sobre a análise de dados que poderá ser efectuada pelo utilizador. O utilizador ao seleccionar um ficheiro poderá realizar qualquer uma das funcionalidades identificadas no caso de uso analisar dados (4.4). Todas as funcionalidades operam sobre a totalidade dos dados de cada ficheiro.

O Utilizador poderá criar histogramas, gráficos de dispersão, representar dados do ficheiro e visualizar dados numa tabela. A funcionalidade representar dados do ficheiro recorre à criação de um mapa (se este foi introduzido na funcionalidade de importação de ficheiros) para a respectiva representação. Caso não tenha sido importado nenhum mapa, a representação decorre em tela branca. Todos os dados relativos ao ficheiro serão representados sob a forma de pontos. Todas as opções identificadas na visualização de dados, estão também presentes nesta representação: *Zoom*, *Pan* e Informação coordenadas.

A criação dos gráficos, tanto de dispersão como dos histogramas, decorre em forma de janela sendo representado sob a forma de pontos (4.4.2) e de barras (4.4.1). O Utilizador opta por quais os atributos que pretende analisar, e no caso dos histogramas qual o número de intervalos que pretende para a representação. No caso da criação dos histogramas este escolhe em qual dos eixos pretende visualizar cada um dos atributos.

Esta representação proporciona ainda um conjunto alargado de opções ao utilizador acerca dos gráficos gerados: imprimir, copiar, formatar folha, ampliar, reduzir, guardar imagem como, *pan* e mostrar valores dos pontos representados.

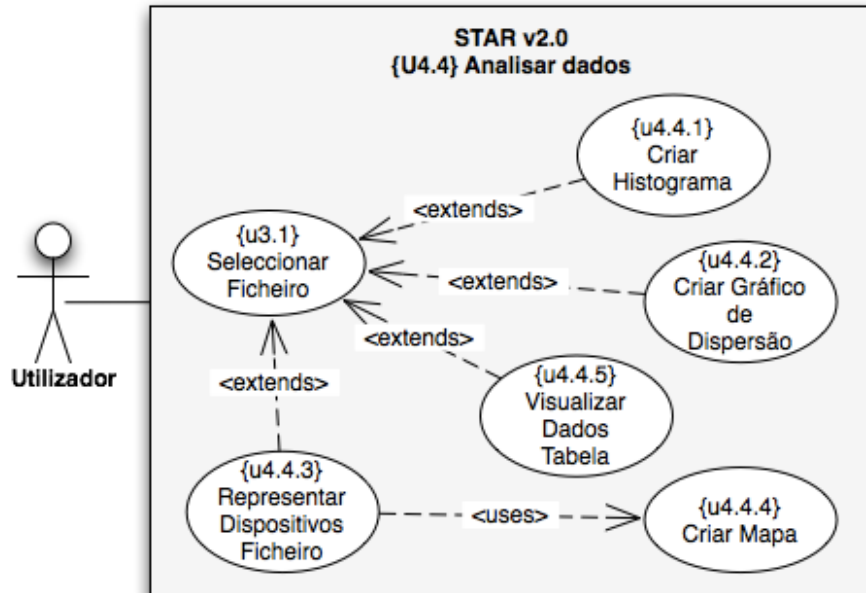


Figura 41 – Diagrama de Casos de Uso: Analisar Dados

Por último, a visualização de dados numa tabela (4.4.5), apresenta todos os dados de um ficheiro, seleccionado pelo Utilizador, sobre a forma de tabela. Como na funcionalidade, visualizar dados tabela, caso de uso visualizar dados (4.2), esta tabela ilustra a totalidade dos dados contidos na Tabela *data* da BD desenvolvida, referente ao ficheiro especificado. O Utilizador pode ordenar por ordem crescente ou decrescente qualquer coluna desta tabela.

### Corrigir Dados

O caso de uso corrigir dados (Figura 42) integra um conjunto de funcionalidades sobre os dados disponíveis no sistema (entidades móveis) que poderão ser utilizados pelo Utilizador de formas distintas. A correcção de dados apresenta como principal vantagem a limpeza de possíveis falhas de dados nos ficheiros importados, tornando os resultados de análises e visualizações mais coerentes. A detecção de falhas pode ser feita através dos gráficos de dispersão, histogramas ou visualização de dados numa tabela. Assim, o Utilizador poderá optar entre analisar 4 componentes essenciais nos dados armazenados: a velocidade, ângulo e as duas coordenadas associadas à localização. Estes atributos foram identificados como os

mais comuns entre os ficheiros sendo que, por não existir uma padronização dos ficheiros, são os passíveis de corrigir na aplicação.

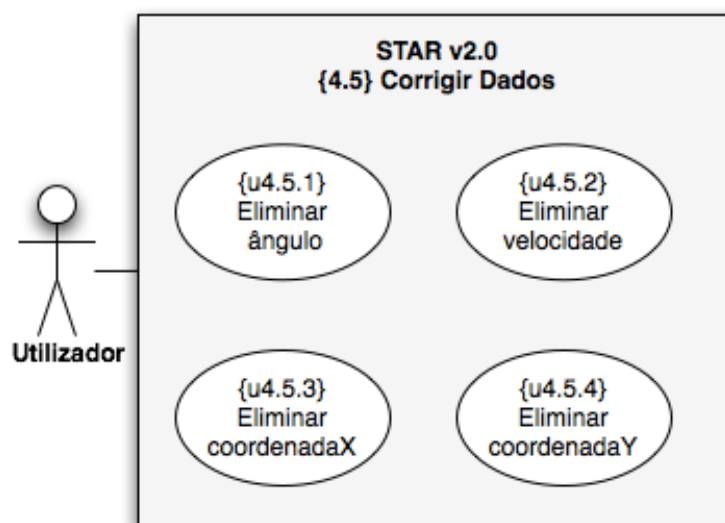


Figura 42 – Diagrama de Casos de Uso: Corrigir Dados

O Utilizador ao seleccionar o ficheiro poderá fazer vários tipos de correcções aos dados integrados no mesmo. Poderá eliminar todos os dados com velocidades a *null*, eliminar dados com velocidades superiores ou inferiores a um valor especificado pelo utilizador (e.g. 10m/s), eliminar dados com ângulos superiores ou inferiores a um valor definido pelo mesmo (e.g. 360°) e, por último, eliminar dados com coordenadas superiores ou inferiores a um valor pretendido.

### 3.5. Síntese comparativa das especificações STAR v.1.0 - STAR v.2.0

Nesta secção é realizada uma comparação entre a primeira e a segunda versão do STAR, sendo identificados os melhoramentos conseguidos no desenvolvimento desta versão. Como referido no capítulo introdutório, a primeira versão do STAR foi analisada em detalhe sendo exploradas todas as suas funcionalidades e avaliados tanto a arquitectura do sistema como o diagrama de classes. Esta análise permitiu identificar aspectos que podem ser melhorados, ao nível do desenvolvimento, o que levou à proposta de uma nova arquitectura do sistema bem como de um novo diagrama de classes. A arquitectura do sistema da primeira versão não está estruturada em camadas e o diagrama de classes apresenta uma complexidade que não é necessária dadas as funcionalidades implementadas.. Do ponto de vista das funcionalidades, a tabela 6 resume todas as funcionalidades implementadas nas duas versões do STAR.

STAR Versão 1	STAR Versão 2
<ul style="list-style-type: none"> <li>• Autenticar</li> <li>• Registar novo Utilizador</li> <li>• Gerir Arquivos <ul style="list-style-type: none"> <li>➤ Criar arquivo</li> <li>➤ <del>Eliminar arquivo</del></li> <li>➤ Listar arquivos</li> <li>➤ Pesquisar arquivos</li> <li>➤ <del>Classificar Arquivos</del></li> <li>➤ <del>Alterar Detalhes Arquivo</del></li> </ul> </li> <li>• Gerir Ficheiros <ul style="list-style-type: none"> <li>➤ <del>Criar ficheiro</del></li> <li>➤ <del>Abrir ficheiro</del></li> <li>➤ Listar ficheiros</li> <li>➤ Pesquisar ficheiros</li> <li>➤ <del>Alterar Detalhes Ficheiro</del></li> <li>➤ <del>Eliminar ficheiros</del></li> <li>➤ Classificar ficheiros</li> <li>➤ <del>Carregar/Descarregar ficheiros</del></li> </ul> </li> <li>• Analisar/Visualizar Dados <ul style="list-style-type: none"> <li>➤ Criar Histograma</li> <li>➤ Criar Gráfico Dispersão</li> <li>➤ <del>Mostrar Estatísticas</del></li> <li>➤ Mostrar Vectores</li> <li>➤ Mostrar Trajectórias</li> <li>➤ <del>Criar Tabelas</del></li> <li>➤ Representar dados</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Autenticar</li> <li>• Registar novo Utilizador</li> <li>• <b>Alterar Conexão BD</b></li> <li>• Gerir Arquivos <ul style="list-style-type: none"> <li>➤ Criar arquivo</li> <li>➤ Listar arquivos</li> <li>➤ Pesquisar arquivos</li> <li>➤ <b>Seleccionar arquivo</b> <ul style="list-style-type: none"> <li>▪ <b>Lista Ficheiros do Arquivo</b></li> </ul> </li> </ul> </li> <li>• Gerir Ficheiros <ul style="list-style-type: none"> <li>➤ Listar ficheiros</li> <li>➤ Pesquisar ficheiros</li> <li>➤ <b>Importar Ficheiros</b> <ul style="list-style-type: none"> <li>▪ <b>Inferir posição ficheiro</b></li> <li>▪ <b>Classificar ficheiro</b></li> <li>▪ <b>Seleccionar mapa</b></li> </ul> </li> </ul> </li> <li>• Gerir Dados <ul style="list-style-type: none"> <li>➤ <b>Seleccionar dados</b></li> <li>➤ <b>Listar dados</b></li> <li>➤ Visualizar dados <ul style="list-style-type: none"> <li>▪ Visualizar Dados Mapa</li> <li>▪ Visualizar Trajectórias</li> <li>▪ Visualizar Vectores</li> <li>▪ <b>Visualizar Dados Tabela</b></li> </ul> </li> <li>➤ Analisar dados <ul style="list-style-type: none"> <li>▪ <b>Representar Dados Mapa</b></li> <li>▪ Criar Histograma</li> <li>▪ Criar Gráfico de Dispersão</li> <li>▪ <b>Visualizar Dados Tabela</b></li> </ul> </li> <li>➤ <b>Correcção dados</b></li> </ul> </li> </ul>

Tabela 6 - Comparação das funcionalidades entre as duas versões

Na primeira versão, as funcionalidades que se encontram rasuradas não foram implementadas, e para as destacadas a cinzento foram realizados melhoramentos no seu funcionamento

durante o desenvolvimento da segunda versão. As funcionalidades apresentadas a negrito, na segunda versão, representam novas funcionalidades do sistema. Todas as funcionalidades apresentadas nesta segunda versão foram implementadas de raiz com o intuito de cumprir o requisito de recorrer unicamente a *software open source*.

Todas as funcionalidades de eliminação e alteração de detalhes de ficheiros, arquivos ou dados, não foram consideradas na segunda versão do STAR. Esta opção deveu-se à preferência de tornar o sistema num sistema unicamente de entrada, onde vários utilizadores podem aceder aos mesmos ficheiros, arquivos e dados, ao mesmo tempo, e em diferentes computadores.

A opção de classificar arquivos, criar ficheiro e abrir ficheiro, foram também eliminadas pois tornavam-se redundantes na forma como foi estruturada e desenvolvida a segunda versão do STAR. A classificação de arquivos e ficheiros não confere nenhuma vantagem, tornando-se desnecessária, pois basta, classificar como privados ou públicos, um dos dois conjuntos. As opções de criar e abrir ficheiros, tornam-se desadequadas pois ao ser importado um ficheiro para o sistema é automaticamente criado um novo ficheiro no mesmo, e todos os seus dados passam a estar disponíveis para visualização, análise e estudo, não sendo necessário aceder novamente ao ficheiro.

As funcionalidades inexistentes na primeira versão do STAR e implementadas nesta versão são: alterar conexão à base de dados, correcção de dados e importação de ficheiros. Ao ser permitido alterar a conexão à BD, o sistema torna-se ubíquo, pois este deixa de estar restringido a uma base de dados local (no computador pessoal), passando a aceder a base de dados remotas (preparadas para tal), onde vários utilizadores acedem e usam todas as potencialidades do protótipo. A importação de ficheiros e correcção de dados já importados torna o sistema mais eficaz, permitindo análises mais rigorosas a conjuntos de dados, sendo possível eliminar dados de acordo com o que cada utilizador pretende.

As operações de análise e visualização de dados foram reagrupadas nesta segunda versão do STAR, formando dois conjuntos diferentes, pois umas operam sobre a totalidade dos dados de um ficheiro (Análise de Dados) e as outras sobre um conjunto de dados definido pelo utilizador do sistema (Visualização de Dados). Na análise dos dados, a criação de histogramas e gráficos de dispersão tornou-se mais simples e com mais funcionalidades. Foram adicionados um conjunto alargado de opções acerca dos gráficos gerados (funcionalidade já referidas anteriormente). Também a representação dos dados foi melhorada, sendo esta representado sobre um mapa, podendo este ser ampliado, reduzido e movido de acordo com o



que o utilizador pretender. Por último, foi acrescentada a funcionalidade de visualização dos dados numa tabela, importados para a Tabela *data* na BD desenvolvida. Esta apresenta sobre a forma de tabela, todos os dados (entidades móveis) de um ficheiro.

Nas visualizações dos dados, todas as representações e visualizações de dados numa tabela, utilizam as mesmas funcionalidades descritas na representação de dados e visualização de dados numa tabela, o que confere maior independência e possibilidade de análise das entidades móveis.

Por último, a criação de tabelas e estatísticas não foram desenvolvidas na nova versão, pois não foram consideradas funcionalidades cruciais nesta nova versão do sistema.



## 4. IMPLEMENTAÇÃO DO PROTÓTIPO para o STAR versão 2.0

Este capítulo inicia-se com a descrição das tecnologias escolhidas para o desenvolvimento da segunda versão do STAR, apresentando as razões das suas escolhas. Ao fim de realizado todo o enquadramento das tecnologias e opções tomadas para a implementação do sistema, este prossegue com a análise, apresentação e descrição das implementações realizadas no desenvolvimento da ferramenta. Recorrer-se-á a apresentação de Figuras sempre que o processo de explicação necessitar de tal detalhe.

### 4.1. Enquadramento Tecnológico

A segunda versão do STAR foi desenvolvida utilizando a linguagem C#, recorrendo para tal ao *Microsoft Visual Studio 2010* (<http://www.microsoft.com/visualstudio/pt-br/>).

Descrita a principal finalidade desta dissertação de mestrado, no capítulo 1, destaca-se que a mesma tem como requisito o desenvolvimento da segunda versão do STAR recorrendo unicamente a *Software* aberto, requisito que não foi possível satisfazer na implementação da primeira versão. Com este intuito, realizou-se um estudo de todas as possibilidades existentes para o desenvolvimento deste sistema, analisando quais as mais adequadas para o desenvolvimento de todas as funcionalidades pretendidas.

Existem vários SGBD's existentes no mercado: *PostgreSql*, *MySQL*, *Oracle XE*, *SQL Server 2005*. Na primeira versão do STAR foi usado o *SQL Server 2005* para aceder à BD, sendo utilizadas interrogações SQL e ligações ODBC. Nesta segunda versão do sistema, esta opção não seria viável pois não cumpria o requisito anteriormente identificado sendo portanto o grupo de sistemas de gestão bastante reduzido. A escolha incidiu sobretudo na avaliação das características que cada SGBD gratuito oferece.

Recorreu-se ao *PostgreSql* (<http://www.postgresql.org/>) como sistema de gestão de base de dados e à sua extensão *PostGIS* (<http://postgis.refractory.net/>) para suporte espacial da mesma (Anexo 1: Script de criação da BD).

Na representação dos gráficos de dispersão e histogramas (Análise de Dados) existem várias bibliotecas que podem realizar a tarefa, sendo as que mais se adequam para o pretendido: *Nplot*, *Nevron Chart*, *NShape*, *WebChart*, *ZedGraph*. A escolha para a representação e criação de gráficos (e.g. histogramas, gráficos de dispersão) foi a biblioteca *ZedGraph* (<http://csharpopen-source.com/zedgraph.aspx>) escrita em C#.

Por último para acesso a tipos de dados geográficos, realização de consultas espaciais e representação desses mesmos dados em mapas recorreu-se à biblioteca *SharpMap* (<http://sharpmap.codeplex.com>) também esta desenvolvida em *C#*. Foram analisadas outras opções sendo que a escolhida se destacou das restantes: *GeoTools.Net* e *Proj.Net*.

De seguida é realizada a descrição mais detalhada de cada uma das tecnologias, identificadas previamente e que foram utilizadas no desenvolvimento da segunda versão do STAR.

#### 4.1.1. *C# e Visual Studio*

A segunda versão do STAR foi desenvolvida utilizando a linguagem *C#*. Para tal recorreu-se ao *Microsoft Visual Studio 2010*. Este representa um conjunto de ferramentas de desenvolvimento baseado em componentes e tecnologias, usadas com o intuito de criar aplicações robustas e de alto desempenho.

Por outras palavras, o *Microsoft Visual Studio* é um pacote de programas para desenvolvimento de *software* especialmente dedicado ao *.Net* e às linguagens *Visual Basic (VB)*, *C*, *C++*, *C# (C Sharp)* e *J#*. Este também pode ser usado para desenvolvimento na área *web*, usando para tal a plataforma *ASP.NET*.

De acordo com Marques e Pedroso (2005), o *C#* é uma linguagem que tira total partido dos serviços de segurança, da gestão automática de memória, da compilação no momento (*just-in-time*) e de um conjunto alargado de bibliotecas *.NET*.

A linguagem *C#* não deve ser vista de uma forma isolada, mas sim considerada em paralelo com a *framework .NET*, pois todo o código escrito em *C#* será sempre executado dentro da *framework .NET* (Nagel et al., 2010). Esta plataforma é definida como uma *framework* que suporta programação em várias linguagens, em conjunto com uma série de produtos para desenvolvimento e execução de aplicações. A sua arquitectura tem como principal objectivo permitir ao utilizador fácil acesso às suas aplicações e dados, em qualquer lugar, a qualquer hora e utilizando qualquer dispositivo.

O *.NET* define duas importantes entidades (Hebert Schildt, 2010).

- CLR (*Common Language Runtime*) – Sistema que gere a execução do programa. Entre outros benefícios, esta é a componente que permite que as aplicações sejam portáteis, permite o suporte a várias linguagens e oferece segurança na sua execução;

- *.NET Class library* - Esta componente permite às aplicações acesso aos ambientes gráficos ou qualquer tipo de entrada ou saída do sistema definido na aplicação executada (e.g. *Windows Forms*)

A plataforma apresenta um conjunto vasto de tecnologias, sendo os principais *ASP.NET*, *Windows Forms*, *ADO.NET*, *XML*, *XML Web Services*, aplicações móveis, entre outros. Para o desenvolvimento da segunda versão do STAR usaram-se algumas destas tecnologias, sendo estas posteriormente descritas.

As *Windows Forms* são a tecnologia recomendada para o desenvolvimento em ambiente local (*Desktop*), apresentando importantes características:

- Utilização de herança, trazida para todas as linguagens que agora são orientadas a objectos, ou seja existe a possibilidade de existir herança entre os *Forms* (janelas) do Windows e também herança entre *Controls* (controles). Isto apresenta uma vantagem que é permitir que sejam criados padrões de interface gráfica, para posteriormente serem usados de uma forma mais simples. Também é possível criar componentes visuais personalizados e adiciona-los à barra de componentes visuais do *Visual Studio .NET*, para que esses sejam usados em qualquer aplicação;
- Facilidade de implementação através do *Visual Studio .NET* (uso de *drag-and-drop* e programação orientada a eventos) e integração com os aplicativos do MS Office, como Word e Excel, além do próprio Windows, (a partir do *namespace System.Windows*).

O *ADO.NET* é um conjunto de classes do *.NET Framework Class Library* que permite o acesso a dados em aplicações *.NET*. Este é a evolução do *ActiveX Data Objects (ADO)*, apresentando várias evoluções em relação à versão anterior, sendo a transformação do *Recordset* do *ADO* em um conjunto formado pelos objectos *DataTable*, *DataSet*, *DataAdapter* e *DataReader*. Cada um destes objectos apresentam diferentes funções:

- *DataTable*: Representa um conjunto de linhas de uma tabela.
- *DataSet*: Representa um conjunto de *Datatables* e seus relacionamentos. Esta é uma estrutura em memória, que pode ser passada entre camadas de uma aplicação multicamadas. É importante referir que o *DataSet* não está ligado à origem dos dados, ou seja, este não se conecta com a base de dados, apenas tem uma cópia das informações contidas nele. O responsável

pela conexão com a base de dados e preenchimento dos dados do *DataSet* é uma entidade denominada *DataAdapter*.

- *DataAdapter*: É o objecto que se conecta com a base de dados e preenche o *DataSet* com as informações da base de dados. O tipo de *DataAdapter* depende do SGBD, enquanto o *DataSet* é genérico.
- *DataProvider*: Componente responsável por fornecer a API de acesso aos dados do SGBD (*driver*).

Como síntese desta secção são identificadas as principais vantagens do uso da plataforma *.NET* (Nagel et al., 2010):

- Programação orientada a objectos - Tanto o *.NET* como o *C#* são completamente sustentados nos princípios orientados a objectos.
- Bom *design* - Uma biblioteca de classes de suporte, que é projectada de baixo para cima de uma forma totalmente intuitiva.
- Independência de Linguagem – Com o *.NET*, todas as linguagens (*C#*, *C++*, *VB*) são compiladas para uma linguagem intermédia. Isto significa que as linguagens são interoperáveis de uma forma que não era vista antes.
- Acesso aos Dados eficiente – Um conjunto de componentes *.Net*, denominados de *ADO.NET*, permite um acesso eficiente a bases de dados relacionais e a uma variedade de fontes de dados. Estes também permitem acesso a ficheiros do sistema, ou directorias.
- Partilha de código – O *.NET* reformulou completamente a maneira como o código é compartilhado entre as aplicações, introduzindo o conceito da montagem, conjunto (*assembly*), que substitui a DLL tradicional. Estas montagens têm medidas formais para controlo de versões, podendo diferentes versões coexistirem lado a lado.
- Maior segurança – Cada conjunto ou montagem (*assembly*) pode conter informação de segurança, que pode indicar com precisão quem ou qual categoria de utilizador ou processo pode invocar métodos e em que classes. Isto confere um grau de controle sobre como as montagens podem ser usadas.
- Impacto de instalação nulo – Existem dois tipos de montagens (*assembly*): partilhadas e privadas. As partilhadas são bibliotecas comuns disponíveis para todo o *software*, sendo que os privados são definidos para o uso

particular de cada *software*. As montagens privadas são completamente autónomas, portanto o processo de instalação é bastante simples. Não existe registo de entradas, os ficheiros introduzidos são colocados na pasta correspondente no sistema de ficheiros.

#### 4.1.2. PostgreSQL

O *PostgreSQL* é um sistema de gestão de base de dados objecto-relacional, gratuito de código fonte aberto, desenvolvido a partir do projecto *Postgres*, iniciado em 1986, na Universidade Califórnia em Berkley.

Este apresenta um conjunto de características gerais que levaram à sua escolha para o desenvolvimento da segunda versão do STAR, entre estas:

- Licenciamento BSD – Esta licença garante total liberdade de uso, alteração, modificação e distribuição do sistema. Esta permite inclusive que versões proprietárias e comerciais sejam criadas a partir do sistema licenciado nestes termos.
- SGBD Objecto-Relacional (Classes e Hierarquia) – O *PostgreSQL* está na mesma categoria de outros SGBD's, tais como Oracle, Microsoft SQL Server e IBM DB2, que evoluíram da arquitectura relacional para objecto-relacional.
- Modularidade – Este recurso facilita o trabalho na implementação de novas funcionalidades (e.g. *PostGIS*).
- Inúmeras interfaces nativas – O *PostgreSQL* pode ser acedido através de várias interfaces: *ODBC*, *JDBC*, *NPGSQL*, *C*, *C++*, *PHP*, *TCL*, *Python*, *Ruby*, etc. Este supera os restantes SGBD's pois nenhum outro possui tanta flexibilidade.
- Flexibilidade – Preparado para várias plataformas (BSD, Linux, Windows, Mac, etc.).
- Robusto – Desenvolvido para suportar grandes volumes de dados e transacções concorrentes.
- Funções – *SQL*, *C*, *C++*, *Perl*, *Python*, etc.
- Totalmente ACID (Atomicidade, Consistência, Isolamento e Durabilidade/Persistência) - Atomicidade pois uma transacção é totalmente

executada ou totalmente revertida sem deixar efeito na base de dados. Consistência, uma vez que os resultados são coerentes com as operações realizadas. Isolamento, porque a execução de uma transacção não interfere ou sofre interferência em relação às demais transacções em execução. Durabilidade, o resultado das transacções deve persistir fisicamente na base de dados.

De forma a chamar a atenção para algumas das limitações que este sistema apresenta, a tabela 7 enumera um conjunto de limitações:

Limite	Valor
Tamanho máximo da Base de Dados	Ilimitado
Tamanho máximo de uma Tabela	32TB
Tamanho máximo de uma Linha	1.6TB
Tamanho máximo de um Campo	1GB
Máximo de coluna por Tabela	250 – 1600 dependendo do tipo de coluna
Máximo de Índices por Tabela	Ilimitado

**Tabela 7 - Limites do PostgreSQL**

O *PostgreSQL*<sup>10</sup> apresenta ainda um conjunto de características específicas que são necessárias a qualquer SGBD e que contribuíram para a sua escolha:

- Recuperação automática após *crash* de sistema (WAL);
- MVCC (controle de concorrência de multi-versão). Neste mecanismo, processos de leitura não bloqueiam processos de escrita e vice-versa, reduzindo drasticamente (às vezes eliminando) a contenção entre transacções concorrentes e paralisação parcial ou completa (*deadlock*);

<sup>10</sup> Para obter informação mais detalhada sobre o Sistema de Gestão de Base de Dados consulte o manual oficial do mesmo (<http://www.postgresql.org/docs/9.1/static/index.html>).



- Modelo Cliente/Servidor. O *PostgreSQL* utiliza um processo por utilizador. Este pode tratar várias ligações simultâneas de cliente, sendo isto conseguido iniciado um novo processo (*fork*) por cada ligação.
- *Logging* de transações;
- *Commit / Rollback / Checkpoints*;
- *Triggers / Stored Procedures*;
- *Constraints / Foreign Keys*;
- *Backup On-line*.

Para a implementação utilizou-se a ferramenta PGAdminIII. Esta é uma interface completa de administração do *PostgreSQL*, sendo considerada o cliente oficial deste.

#### 4.1.3. PostGIS

O *PostGIS* é uma extensão do *PostgreSQL* que fornece o suporte de objectos geográficos. Inicialmente foi desenvolvida pela comunidade de software livre, sendo actualmente a empresa *Refractions Research Inc* que mantém o desenvolvimento desta extensão, seguindo as especificações da *SFSSQL (Simple Features Specification for SQL)*. Efectivamente, o *PostGIS* habilita do ponto de vista espacial o *PostgreSQL*, adicionando suporte a objectos geográficos.

Este podia ser denominado como “PostgreSQL espacial”, uma vez que funciona da mesma forma que o proprietário das extensões de base de dados espaciais:

- Adição de um tipo de dados “*geometry*” aos tipos de dados convencionais de uma base de dados usual (e.g. *varchar*, *integer*, *date*, etc).
- Adição de novas funções que usam o tipo *geometry* (e.g. *st\_distance(geo,geo)*, *st\_length(geo)*, etc.).
- Acrescenta um mecanismo de indexação para permitir consultas com restrições espaciais, com o intuito de retornar registos de uma forma rápida.

As funcionalidades essenciais de uma base de dados espacial são simples de enumerar: tipos, funções e índices. Mais complexo é a quantidade de processamento espacial que pode ser feito numa base de dados, ao fim de estarem presentes estas funcionalidades: análises de sobreposição, projecções, pesquisas de proximidade, filtros espaciais, atributos, entre outros.

A estrutura dos dados no *PostGIS* é definida pelo padrão *OGC SFS (Simple Feature Specification)*. Esta suporta vários tipos de geometrias, sendo de seguida definidas e especificados exemplos recorrendo a coordenadas de pontos para cada uma delas:

- Point (0 0);
- Linestring ( 0 0,1 1,1 2);
- Polygon (0 0, 4 0, 4 4, 4 0, 0 0),(1 1, 3 1, 3 3, 1 3, 1 1);
- Multipoint (1 1, 1 3);
- Multilinestring ((0 0, 1 1, 1 3),(3 2, 2 3, 1 4));
- Multiolygon (((0 0, 3 0, 3 3, 0 3, 0 0),( 2 2, 2 1, 2 2, 1 2, 2 2)),((-2 -2, -2 -1, -2 -2, -1 -2, -2 -2)));
- GometryCollection (POINT( 1 3), LINESTRING(2 3,3 4)).

Resumindo o *PostGIS*<sup>11</sup> confere ao *PostgreSQL* uma extensão espacial, que permite à base de dados poder armazenar e consultar dados espaciais (pontos, linhas e polígonos).

#### 4.1.4. ZedGraph

*ZedGraph* é uma biblioteca de classes, *Windows Forms*, *UserControl* e *ASP*, acessível para a criação de vários tipos de gráficos (gráficos 2D, gráficos de barra, gráficos de dispersão, histogramas, etc.) a partir de um conjunto de dados arbitrários. Esta é uma biblioteca de código aberto, ou seja, livre, desenvolvida em C# sendo mantida como um projecto em desenvolvimento no *SourceForge* (<http://sourceforge.net/projects/zedgraph/>).

Esta biblioteca fornece um alto nível de flexibilidade, uma vez que quase todos os aspectos do gráfico podem ser modificados pelo utilizador. Em paralelo, o seu uso é bastante simples não existindo restrições e limitações no seu uso, uma vez que o utilizador tem controlo completo sobre os gráficos, e uma alargada gama de escolha dos mesmos.

---

<sup>11</sup> Para obter informação mais detalhada sobre a extensão *PostGIS* poderá consultar o manual oficial do mesmo. (<http://postgis.refractor.net/documentation/manual-1.5/>)

Por último esta biblioteca encontra-se muito bem documentada e em constante actualização. Existe uma grande quantidade de exemplos e de formas de uso da mesma *on-line* sendo que a principal componente de apoio encontra-se disponível *on-line* de uma forma simples e clara (<http://zedgraph.sourceforge.net/documentation/default.html>).

#### 4.1.5. Sharpmap

*Sharpmap* é uma biblioteca espacial fácil de usar, tanto em aplicações *web* como *Desktop*. Esta permite acesso a vários tipos dados espaciais, possibilitando consultas, e geração de mapas. A biblioteca foi desenvolvida em C# utilizando a plataforma .NET e está sob a licença *GNU Lesser General Public License* (Código livre).

Esta suporta um conjunto bastante alargado de tipos de dados, sendo que no caso da segunda versão do STAR os mais importantes são Pontos, Linhas e Polígonos.

Como conclusão, o *Sharpmap* é uma biblioteca em constante desenvolvimento, existindo uma excelente documentação em torno da mesma.

## 4.2. Implementação Realizada

Nesta secção são descritas e explicadas algumas funcionalidades do STAR v2.0. Para a explicação das funcionalidades recorre-se a diagramas e Figuras ilustrativas de como é realizada a tarefa descrita.

### 4.2.1. Conexão à Base de Dados - NPGSQL

A conexão à base de dados é fundamental para o desenvolvimento e bom funcionamento de qualquer aplicação que necessite de aceder a dados armazenados numa base de dados. Esta é uma operação relativamente lenta.

Para efectuar a conexão da segunda versão do STAR à sua base de dados usou-se a biblioteca *NPGSQL* implementada em C#. Esta não apresenta dependência de bibliotecas nativas o que facilita a sua instalação e utilização, permitindo que aplicações *.NET* acedam ao *PostgreSQL*. O *NPGSQL* é um *.Net Data Provider* para o *PostgreSQL*. que oferece suporte a consultas comuns (*insert*, *update*, *delete* e *select*) e implementa as classes base do ADO.NET (*NpgsqlConnection*, *NpgsqlCommand*, *NpgsqlDataReader*, etc.). De acordo com o manual (<http://npgsql.projects.postgresql.org/docs/manual/UserManual.html>) ao estabelecer uma

conexão (*NpgsqlConnection*), existem vários parâmetros que modificam o comportamento da mesma (*SSL, Pooling, Integrated, security, etc.*). Por definição uma conexão utiliza uma *pool* de conexões (*Pooling = True;*) passando a ser da responsabilidade do *NPGSQL* a gestão das mesmas. Uma *pool* de conexões contém todas as conexões abertas e reutilizáveis, sendo que o seu uso melhora o desempenho e escalabilidade.

A Figura 43 apresenta as classes implementadas para estabelecer a conexão entre a aplicação e a base de dados do sistema STAR, apresentando todos os parâmetros contidos na sua configuração.

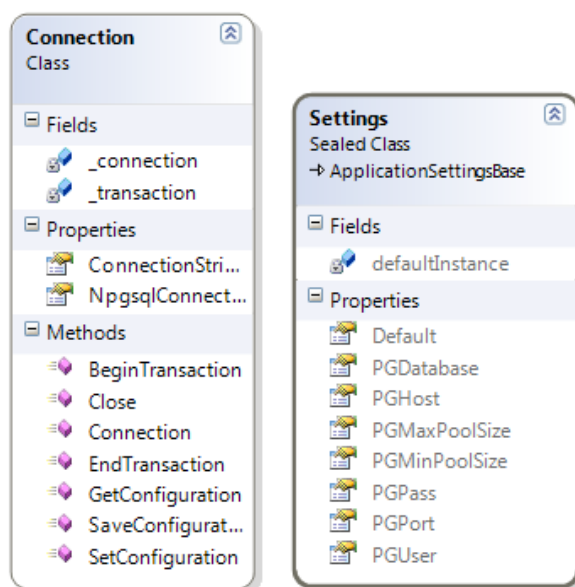


Figura 43 - Conexão

#### 4.2.2. Importar Ficheiro

No desenvolvimento da segunda versão do STAR surgiu a necessidade de importação de dados (geográficos) contidos em ficheiros, para a base de dados espaço-temporal já desenvolvida (funcionalidade não existente na primeira versão).

A necessidade de importar dados geográficos para uma base de dados não é uma tarefa trivial, pois existem várias limitações e restrições que necessitam de ser ultrapassadas para que esta tarefa seja simples e eficaz. Entre as várias limitações e restrições identificam-se as seguintes:

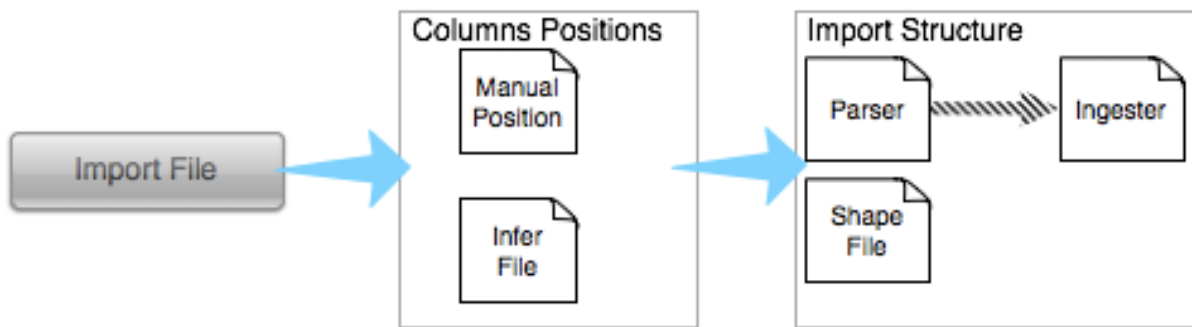
- Tipo de ficheiro a importar (extensão);
- Dados contidos no ficheiro (campos);
- Definição da estrutura dos dados;
- Tamanho do ficheiro a importar;

- Tempo de importação do ficheiro;
- Estrutura/Modelo de Importação.

De acordo com estas limitações, a realização de um único processo que ultrapassasse todas estas limitações e efectivasse a inserção dos dados na base de dados, não representava a melhor solução.

Assim, a funcionalidade de importação de um ficheiro foi dividida em várias tarefas, ultrapassando a restrição de uma estrutura única, que limitava necessidades futuras (novos requisitos).

A Figura 44 demonstra as opções que a funcionalidade apresenta, ilustrando todos os processos em que esta se encontra dividida:



**Figura 44 - Funcionalidade Importe**

Existem dois passos fundamentais no importe dos dados na base de dados:

1. Localizar/Identificar a posição dos dados no ficheiro;
2. O importe do ficheiro e respectivo *shapefile* (opcional) na base de dados.

Cada um destes processos é realizado separadamente sendo que os dois são fundamentais para o funcionamento correcto e eficaz da funcionalidade.

De uma forma mais específica, localizar/identificar os dados num ficheiro está intrinsecamente ligado ao formato do mesmo. Este processo pode ser manual, sendo o utilizador a localizar os dados no ficheiro (mediante um padrão de repetição), ou de forma mais sofisticada através da funcionalidade *infer* automático.

O segundo processo (Importe Ficheiro), realiza o carregamento de todos os dados contidos no ficheiro para memória realizando de seguida a carregamento dos mesmos para a base de dados do sistema.

Todo o processo de importe dos dados de um ficheiro para a base de dados passa por várias etapas/mecanismos até à conclusão final da funcionalidade. Todos os mecanismos e evolução da funcionalidade ao nível da arquitectura são detalhados na Figura 45.

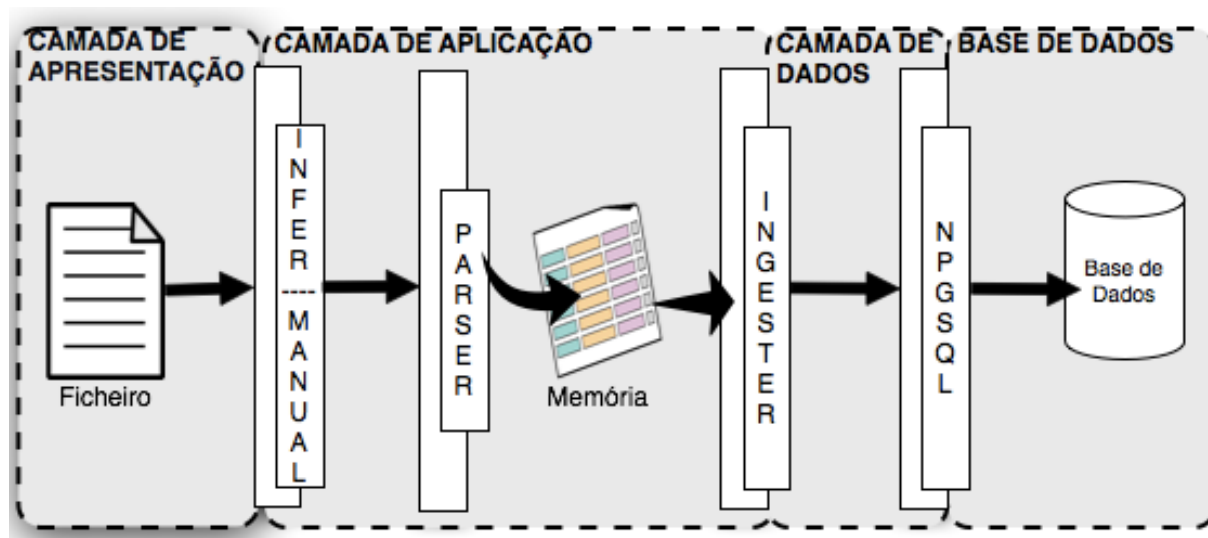


Figura 45 - Funcionamento do Importe

O modelo é composto por 4 componentes fundamentais que realizam diferentes funções:

- Infer – Camada opcional que permite localizar/identificar automaticamente os dados num ficheiro, mediante um padrão de repetição.
- Parser – Componente que realiza leitura dos valores do ficheiro procedendo ao seu armazenamento (estruturado) em memória.
- Ingester – Componente que realiza a passagem dos dados estruturados em memória para a base de dados.
- Npgsql – *Driver* que permite a comunicação da aplicação com a base de dados espaço-temporal.

Este processo interage com todas as camadas existentes na arquitectura do sistema, uma vez que vai desde a interface com o utilizador até ao armazenamento dos dados no sistema. De referir que o modelo apresenta uma estrutura vantajosa para o decorrer do bom funcionamento do sistema bem como de futuros avanços nesta funcionalidade, pois, por exemplo, se for necessário a leitura de um novo formato de ficheiro, ou de geração de um ficheiro de saída com um formato específico (e.g. GML), não é necessário a implementação de uma nova estrutura, basta implementar uma nova componente *parser*, ou *ingester*, respectivamente.

Também opcionalmente pode ser desenvolvida uma componente *infer* para localizar os dados no ficheiro.

Todos os mecanismos acima descritos foram implementados para funções específicas, e reflectem um processo crucial, que ultrapassa diversas dificuldades, tornando o sistema de armazenamento, análise e visualização de objectos móveis, a um nível global mais capaz, eficaz e completo.

A estrutura implementada, como já referido, para além de executar tarefa de forma eficiente, foi elaborada, com o intuito de apresentar uma estrutura de suporte para a importação de dados, onde novas funcionalidade, requisitos podem ser acrescentados não sendo necessário a implementação de raiz de todo o processo.

De seguida todas estas componentes são descritas separadamente de forma a detalhar e explicar o funcionamento de todos os processos.

### ***Infer***

Esta funcionalidade tem como objectivo identificar a localização dos campos que estão presentes no ficheiro a importar, baseando-se no facto dos dados seguirem padrões repetitivos. Isto é fundamental para que a funcionalidade importe seja bem-sucedida, pois se a localização dos valores não estiver correcta, os processos de análise e visualização de dados não estarão correctos.

Para o *infer* automático ser possível a primeira linha do ficheiro (.csv) tem de identificar todos os campos do mesmo (cabeçalho).

Ao fim de introduzido o ficheiro (directoria), esta funcionalidade segue os seguintes passos:

- Leitura dos valores da primeira linha do ficheiro.
- Para cada valor é realizada uma filtragem de caracteres, permanecendo unicamente caracteres alfabéticos.
- Os valores são percorridos guardando como estado a respectiva posição.
- Comparação *case-insensitive* entre o valor actual e todos os *enums* ilustrados na Figura 48. Se forem iguais (nas condições mencionadas) é associado ao campo em causa a posição actual.

A Figura 46 identifica as classes implementadas para a realização desta funcionalidade. Nesta estão retratados atributos que foram classificados em dois grupos: *FieldType* e

*OtherFieldType*. Estes dois grupos resultaram da análise de vários ficheiros sendo os atributos incluídos no *FieldType* identificados como os mais comuns (mais utilizados). O primeiro grupo possui os campos (atributos) que estão habitualmente presentes em ficheiros de dados sobre objectos móveis. O segundo grupo identifica outros atributos que podem estar presentes nos ficheiros.

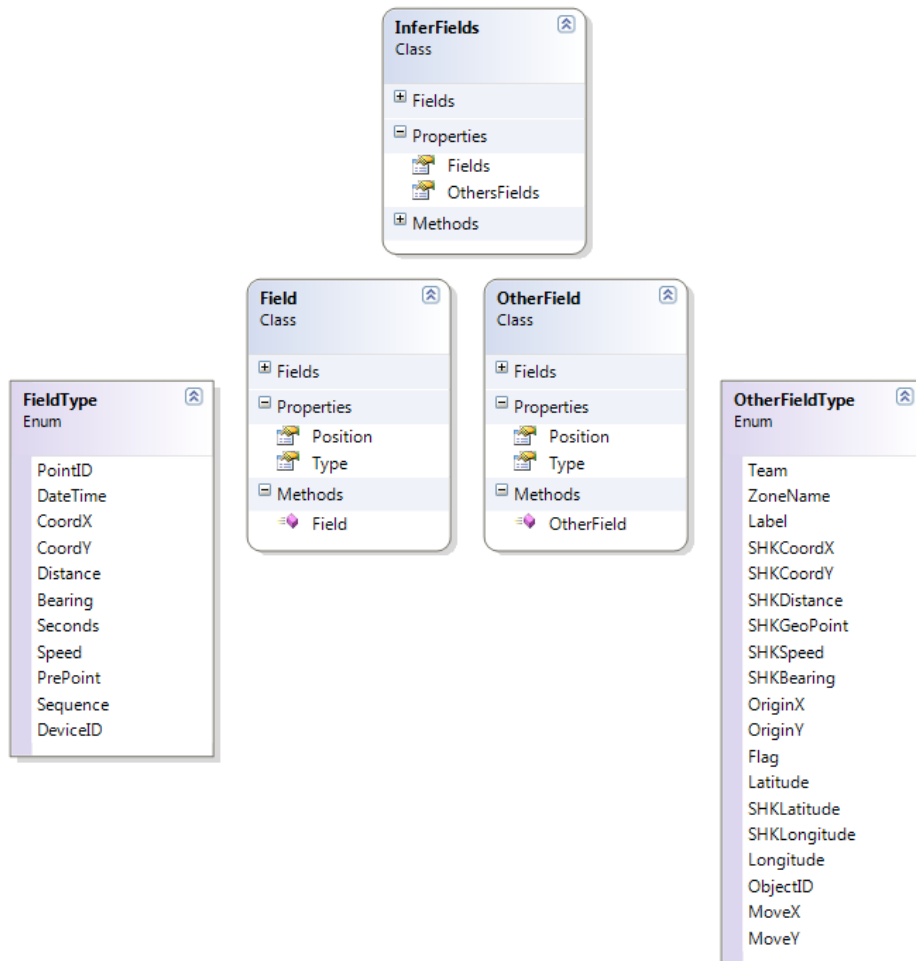


Figura 46 - Infer

### **Parser**

Um *parser* é um componente de *software* que interpreta uma entrada detectando e agrupando construções hierárquicas pré-definidas. A componente desenvolvida neste sistema é geralmente denominada de “*parser* de padrões” (*pattern parser*), ou “*parser* de expressões regulares” (*regular expression parser*). Esta consiste na construção de uma expressão regular (uso do conceito de grupos), por forma a conseguir detectar de uma forma eficiente todos os valores presentes no ficheiro a importar.



As expressões regulares fornecem um método eficiente e flexível para processamento rápido de grandes quantidades de texto. Esta exige duas componentes fundamentais:

- O padrão da expressão regular para identificar no texto;
- O texto a analisar para o padrão da expressão regular.

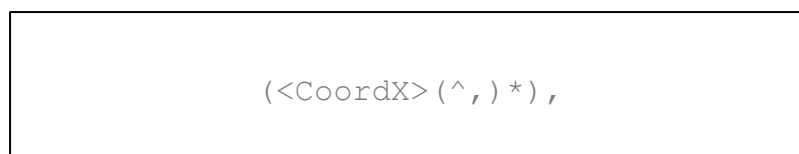
Estas podem ser usadas para vários casos, entre eles, captura de dados, validação de dados de entrada, comparação e substituição de *strings* e redução de tamanho de código.

Na segunda versão do STAR esta componente foi desenvolvida para interpretar ficheiros com o formato “.csv” (*Comma separated values*). Como o próprio nome indica, os valores presentes em ficheiros neste formato encontram-se separados por vírgulas. Este tipo de formato é normalmente utilizado para transferência de informação, sendo adequado para a importação e exportação de dados entre aplicações. Permite importar e exportar uma grande quantidade de dados de forma expedita, sendo ao mesmo tempo legível, manualmente editável e fácil de usar, pois pode ser lido em vários programas (e.g. bloco de notas, Excel). Para além das razões já referidas, a escolha incidiu neste tipo, pois a conversão de outros tipos de formatos para o formato “.csv” é bastante comum, existindo diversos aplicativos que realizam conversões de vários tipos de formatos para o formato csv.

O formato da expressão regular que é utilizado para a importação de dados é um padrão que é utilizado no mecanismo desta componente, sendo este padrão, na teoria de expressões regulares, um grupo.

Este padrão, em específico, está otimizado para o *parsing* implementado, recorrendo ao uso de conceitos avançados de expressões regulares, mais precisamente *Named Groups*. Com os *Named Groups* é possível aceder ao valor de um grupo através do seu nome, em vez de utilizar um índice.

O padrão usado (Figura 47) identifica o campo e o formato do mesmo. A identificação é feita usando o nome do grupo, que é idêntico ao nome do campo. O formato resume-se a todo e qualquer carácter excepto o carácter de separação (vírgula).



```
(<CoordX> (^,)*),
```

**Figura 47 – Formato da Expressão Regular**

Este excerto da expressão regular utilizada contém apenas a expressão regular referente à coordenada X. É a construção na ordem correcta desta expressão que permite um *parsing* correcto e eficiente.

Todos os atributos do ficheiro importado serão identificados da mesma forma, tal como está representado na Figura 47, sendo que o que muda é o nome do grupo e o seu valor. Os nomes dos grupos são identificados na componente descrita anteriormente (*Infer*), ou na introdução manual das posições do ficheiro a importar, sendo identificados todos os campos existentes no ficheiro. A Figura 48 apresenta as classes e métodos implementados para a realização da tarefa descrita.

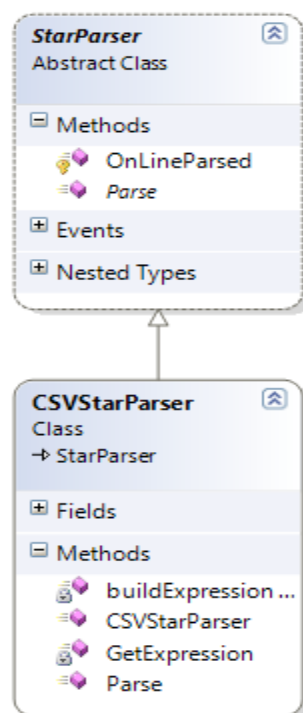


Figura 48 - *Parser*

Ao fim de localizadas/identificados os atributos do ficheiro, esta funcionalidade segue os seguintes passos:

- Leitura de uma linha do ficheiro.
- Aplicação da expressão regular a linha lida.
- Acesso aos valores reconhecidos na expressão regular.
- Construção de uma estrutura de dados em memória com os dados lidos.

A estrutura de dados baseia-se na representação em memória de uma lista de objectos, onde estes contêm toda a informação retirada de uma linha do ficheiro.

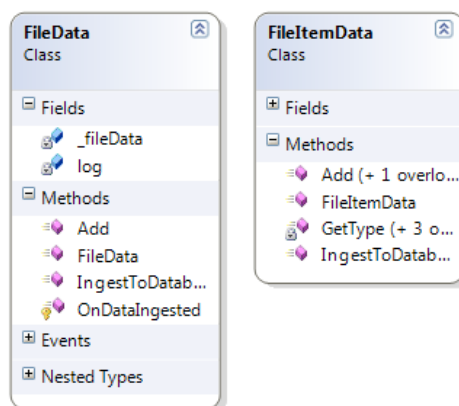
## ***Ingester***

O *Ingester* tem como função efectuar a leitura dos dados que estão estruturados em memória (função *parser*) armazenando-os na base de dados espaço-temporal do sistema STAR. Para que esta operação seja bem sucedida é preciso garantir que a mesma não é interrompida, ou seja, é efectuada de forma atómica<sup>12</sup>.

O mecanismo disponível no SGBD adoptado, que faculta esta propriedade, é a *transaction*. Uma transacção<sup>13</sup> por definição deve ser atómica, consistente, isolada e persistente, sendo estas propriedades geralmente referidas com o acrónimo ACID. A operação de introdução dos dados na base de dados (*ingester*) é efectuada da seguinte forma:

- Início da transacção (*BEGIN*).
- Inserção de todos os dados na BD (*INSERT*).
- Fim da transacção, que pode ocorrer de duas formas distintas: *COMMIT* se não ocorreu nenhum erro, ou *ROLLBACK (ABORT)* se ocorreu algum erro.

A Figura 49 apresenta as classes e métodos implementados para a realização da tarefa descrita.



**Figura 49 - Ingester**

<sup>12</sup> Por forma atómica entende-se uma operação que realiza todo o processamento em caso de sucesso, ou nenhum em caso de falha. O processamento inclui diversas tarefas integradas numa única transacção.

<sup>13</sup> Mais informações acerca de transacções consultar manual do *PostgreSQL* (<http://www.postgresql.org/docs/9.0/static/transaction-iso.html>).

Na segunda versão do STAR esta componente realiza a introdução dos dados na base de dados espaço-temporal. Futuramente, novas funcionalidades de introdução dos dados (e.g. exportar dados para ficheiro com formato .gml), podem ser identificadas, não sendo necessário a implementação de todas as componentes já descritas, pois basta implementar uma nova componente *Ingestor*.

#### 4.2.3. Visualizar Dados

A visualização de dados permite ao utilizador realizar três tipos de funcionalidades diferentes referentes a dados previamente importados para a base de dados. Podem ser realizadas visualizações a um conjunto de dados (entidades móveis) seleccionados pelo utilizador, ou a todos os dados contidos num ficheiro. Podem ser representados os pontos, as trajectórias ou os vectores direccionais, das entidades móveis escolhidas pelo utilizador.

Se na importação do ficheiro o utilizador seleccionou um *shapefile*, os dados seleccionados serão representados com recurso a um mapa. Se não, as representações decorrem em tela branca.

A Figura 50 demonstra as opções que a funcionalidade apresenta, ilustrando todos os processos em que esta se encontra dividida:

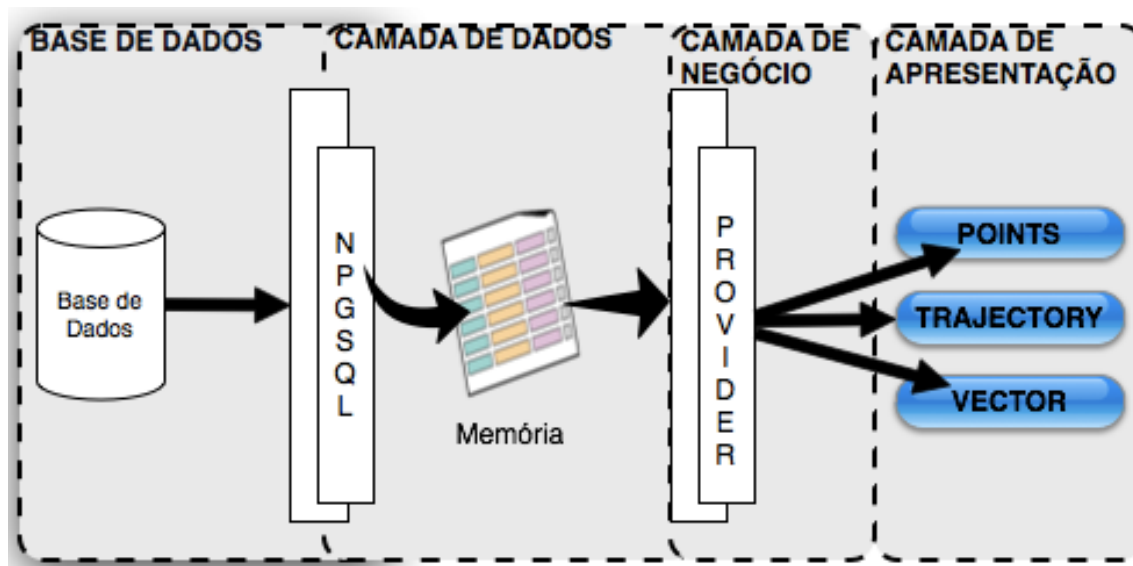


Figura 50 - Funcionamento da Visualização de Dados

### **Provider**

O *provider* realiza a leitura dos dados estruturados em memória (Camada de Dados) inserindo-os na estrutura de dados *SharpMap* (Camada de Apoio). Este é essencial pois transforma os dados provenientes da camada de dados de forma a realizar a tarefa requerida pelo utilizador.

Para cada funcionalidade disponibilizada para a visualização de dados, o *provider* realiza diferentes transformações de acordo com as visualizações pretendidas. Cada funcionalidade necessita de construções específicas tendo em conta a estrutura da biblioteca *SharpMap*.

A biblioteca *SharpMap* apenas possui os conceitos de Figuras geométricas no plano geoespacial, o que levou à necessidade de desenvolvimento de uma componente que realize as transformações necessárias a cada visualização.

Isto faz com que o *provider* dê semântica específica aos dados provenientes da camada de dados, isto é, transformar os dados em Figuras geométricas.

Para as funcionalidades desenvolvidas na segunda versão do STAR, o *provider* necessita apenas de realizar três tipos de transformações: transformações de pontos, linhas e vectores.

Todo este mecanismo de transformação de dados é invisível ao utilizador, pois este para realizar qualquer tipo de visualização tem apenas que seleccionar um, vários ou todos os dados (de um ficheiro) e carregar no botão que define o tipo de visualização pretendido.

As classes desenvolvidas para a realização das visualizações sobre os dados estão ilustradas na Figura 51.

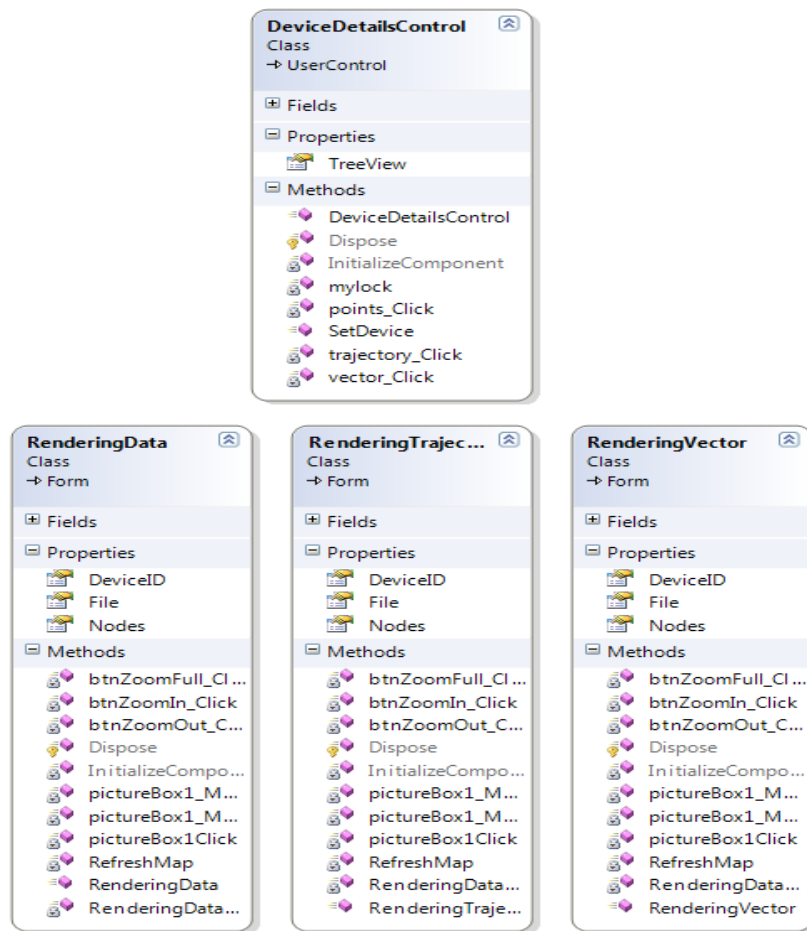


Figura 51 –Visualizar Dados

### ***Point Provider***

Na representação de dados de um ficheiro, a função do *provider* é realizar o mapeamento entre a camada de dados (valores dos dados) e a estrutura de dados do *SharpMap*

O mecanismo de alteração dos dados em pontos não necessita de grandes transformações pois estes cumprem os requisitos necessários às estruturas de dados da biblioteca adoptada para a representação dos mesmos.

### ***Trajectory Provider***

Para a representação de trajectórias de um determinado conjunto de dados, a componente *provider* necessita de realizar alguns cálculos específicos. Para tal, este componente tem que percorrer o conjunto de dados a representar (seleccionados pelo utilizador), agregando-os em pares de linhas consecutivas, sendo que esta estrutura de dados existe na biblioteca *SharpMap*. A representação de uma trajectória consiste numa única linha.

***Vectors Provider***

O *provider* assume um papel decisivo na construção dos vectores direccionais. Os dados provenientes da camada de dados estão estruturados sobre a forma de uma lista de pontos, ordenada pela data de captura dos mesmos. Assim a camada de negócio implementa num *provider* o seguinte algoritmo (Figura 52).

```

C = 10
Para cada i em (2 ... numPontos)
    P1 = pontos(i-1)
    P2 = pontos(i)
    deltaX = P2.X - P1.X
    deltaY = P2.Y - P1.Y

    Se deltaX = 0
        teta =  $\pi / 2$ 
    Senao
        m = deltaY / deltaX
        teta = arctan(m)

    Se deltaY < 0
        teta = teta +  $\pi$ 

    retorna Vector(C, P2, teta)

```

**Figura 52 - Algoritmo para Cálculo de Vectores Direccionais**

O cálculo dos vectores direccionais de cada ponto é feito com base no ponto anterior. Deste modo o primeiro ponto de cada leitura não tem vector associado, não sendo portanto representado. Este vector de comprimento fixo *c* (definido previamente) tem como ponto inicial, o ponto sobre qual é feito o calculo *P2*, com uma direcção *teta*, que resulta num prolongamento do segmento de recta formado por este e pelo ponto anterior.

Este algoritmo teve de ser implementado uma vez que nem todos os ficheiros contêm informações acerca do *Bearing*

#### 4.2.4. Analisar Dados

A análise de dados permite ao utilizador realizar dois tipos de funcionalidades diferentes referentes a dados previamente importados para a base de dados. Esta componente trabalha unicamente com ficheiros, fazendo análises a todos os dados contidos no mesmo.

A Figura 53 demonstra as opções que a funcionalidade de análise de dados apresenta, ilustrando todos os processos em que esta se encontra dividida:

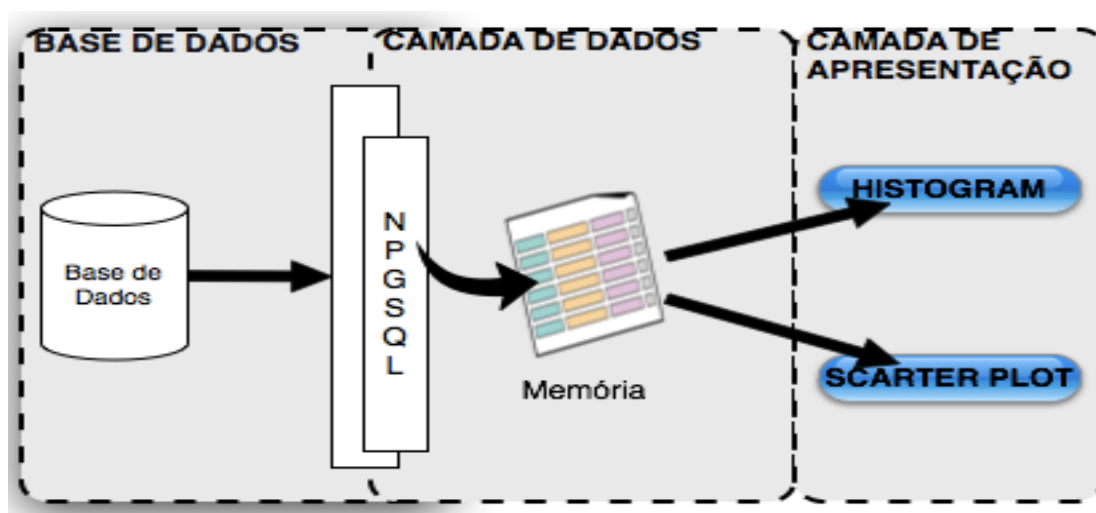


Figura 53 - Funcionamento da Análise de Dados

Nesta funcionalidade, não existe uma camada de cálculo (*Provider*), uma vez que os dados provenientes da camada de dados já se encontram no formato correcto e prontos para serem usados pela biblioteca *ZedGraph*.

Os dois tipos de análise decorrem sobre a forma de gráfico com eixos, sendo representados todos os valores de um ficheiro recorrendo a pontos (gráfico de dispersão) ou barras (histograma). As duas representações contêm um vasto leque de opções sobre as mesmas (funcionalidades já descritas no capítulo anterior).

As classes desenvolvidas para a realização das análises de dados estão ilustradas na Figura 54.



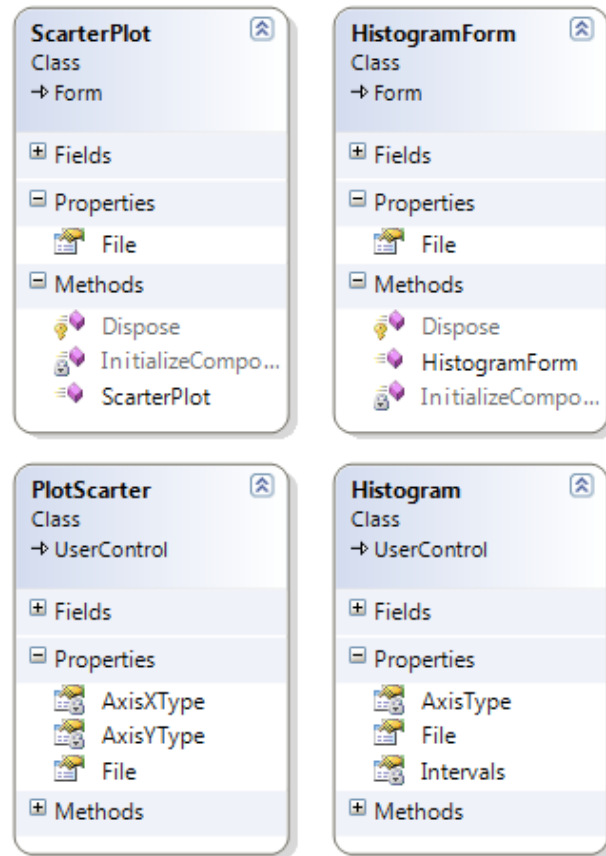


Figura 54 – Análise de Dados

A criação de histogramas permite ao utilizador especificar o atributo (a partir de uma lista de atributos) e número de intervalos que pretende para a sua análise. O gráfico de dispersão consiste na especificação das duas variáveis que se pretendem analisar, especificando o eixo em que pretende analisar cada uma.

#### 4.2.5. Correção de Dados

A funcionalidade de correção de dados permite ao utilizador realizar a limpeza de alguns atributos de um ficheiro, previamente seleccionado. Esta funcionalidade confere ao utilizador a possibilidade de realizar operações sobre os dados, através de uma interface simples e intuitiva. Aplicando o processo de limpeza ou correção de dados o utilizador passa a poder analisar ou visualizar dados com maior precisão, uma vez que valores errados ou omissos são eliminados do conjunto de dados em análise.

Resumidamente, esta operação consiste num conjunto de *queries* que eliminam dados na base de dados. Os dados a eliminar são dados que satisfazem condições especificadas pelo utilizador, num formulário que foi desenhado especificamente para a tarefa.

O utilizador pode assim eliminar quatro tipos de atributos: velocidade, ângulo, coordenada X e coordenada Y. A opção por estes quatro atributos deriva do facto destes serem os atributos numéricos considerados no primeiro grupo de atributos, os *FieldType*. De futuro sugere-se que esta funcionalidade seja revista no sentido de passar a considerar outros atributos.

Para realizar a eliminação dos dados, o utilizador precisa unicamente de seleccionar o atributo, ou os atributos, que pretende eliminar, definido o valor para o qual se verifica a a eliminação ou a partir do qual esta deve ocorrer (e.g. eliminar velocidade > 10, ângulo < 300, velocidade = Null, coordX > 757690,00). Este procedimento oferece um nível de total independência de correcção de dados, pois cabe ao utilizador escolher o atributo e os valores a eliminar.

## 5. APRESENTAÇÃO E AVALIAÇÃO DO PROTÓTIPO IMPLEMENTADO

Este capítulo começa a apresentação da segunda versão do STAR através de um conjunto de Figuras e respectiva descrição. Serão descritos os passos necessários à instalação do sistema, prosseguindo com um descrição de todos os componentes e utilidades do protótipo implementado.

Seguidamente, apresentam-se todas as funcionalidades implementadas neste protótipo, dando particular ênfase às análises e visualizações de dados, funcionalidades cruciais para o sucesso do sistema. As análises e visualizações de dados serão realizadas a um conjunto de dados que será posteriormente descrito no decorrer deste capítulo.

Na demonstração de todas as funcionalidades serão ilustradas Figuras do sistema em funcionamento recorrendo ao uso de *print screens*.

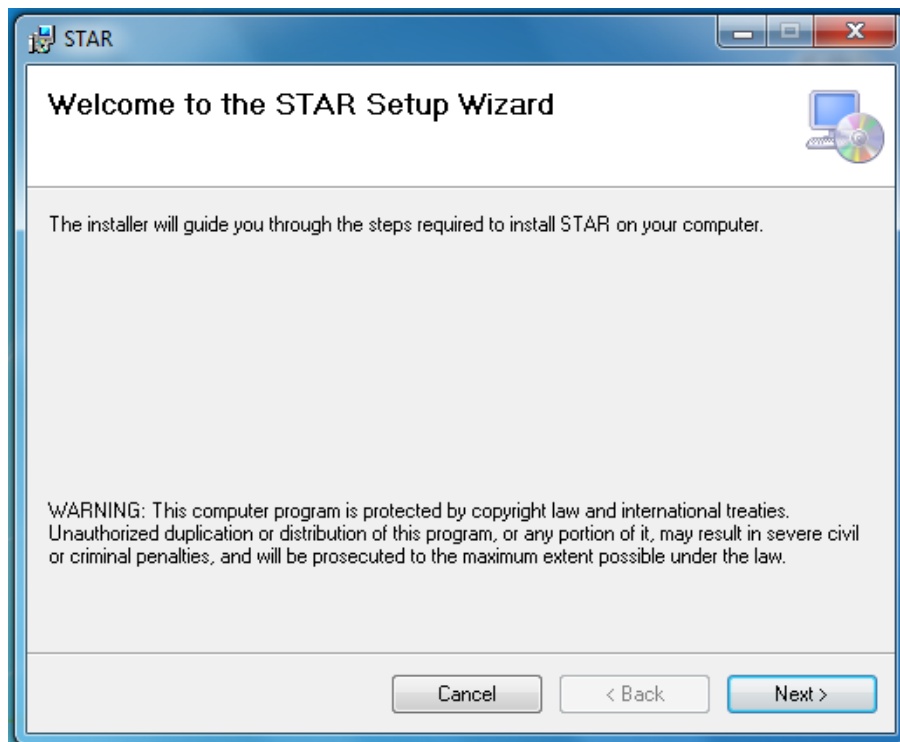
O capítulo termina com uma avaliação do trabalho realizado, apresentando um balanço das funcionalidades implementadas e identificando as que ficaram por implementar.

### 5.1. Apresentação do STAR v2.0

Esta secção pretende apresentar o protótipo da segunda versão do STAR. Para tal, primeiramente será descrito o procedimento necessário à instalação do mesmo, sendo posteriormente descritas todas as componentes que constituem esta versão.

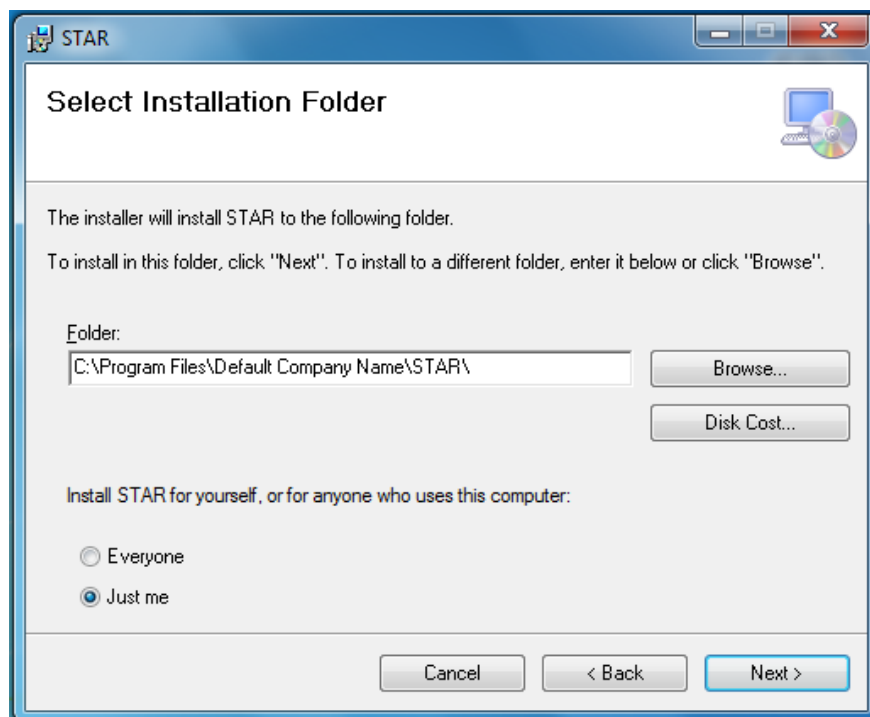
#### 5.1.1. Instalação do STAR v2.0

O processo de instalação da segunda versão do STAR é bastante simples e fácil de concluir. Basta clicar no ícone de *setup* (*starSetup.exe*) gerado pelo protótipo que o processo de instalação inicia automaticamente (Figura 55). Este processo de instalação é igual a muitos outros processos de instalação de qualquer aplicação.



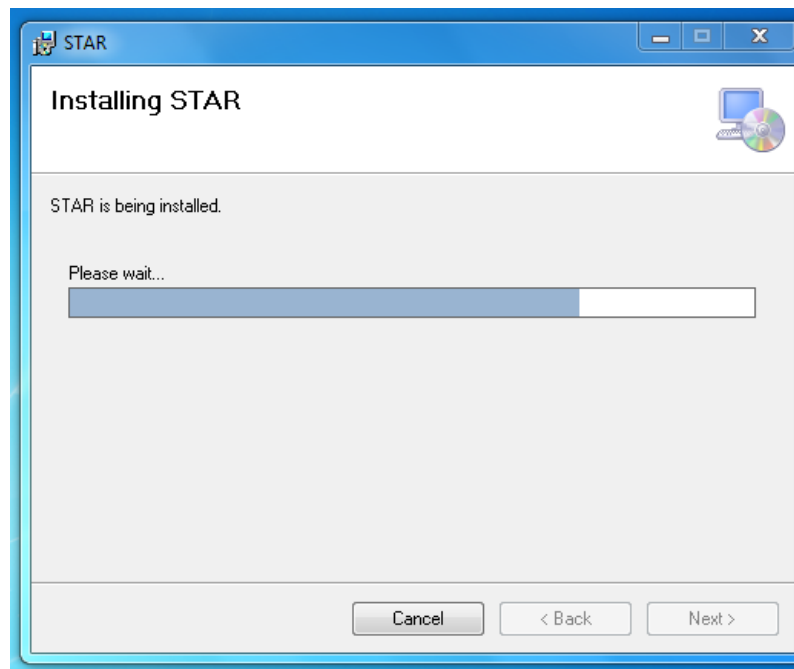
**Figura 55 - Instalação do STAR v. 2.0**

Para prosseguir a instalação basta clicar no botão *Next* (Figura 55). Se pretender sair da mesma pode cancelar a operação (Botão *Cancel*). Na janela seguinte o utilizador escolhe qual a directoria onde pretende instalar a aplicação e prossegue a instalação (Figura 56).



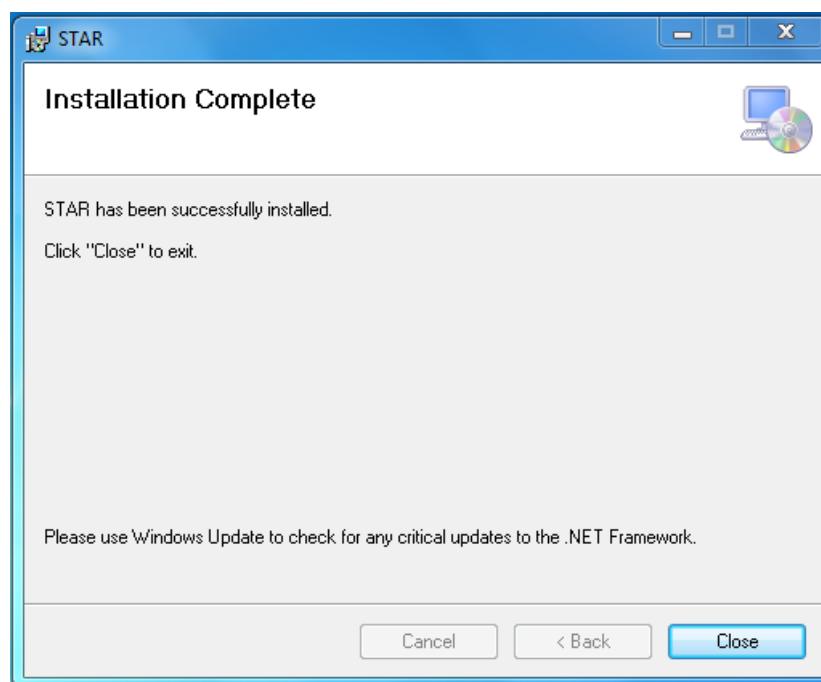
**Figura 56 - Instalação do STAR v. 2.0**

De seguida é ilustrada uma última janela antes de começar a instalação da aplicação na directoria especificada, o utilizador pode cancelar todo o processo, retroceder para especificar outra directoria, ou avançar para a instalação da aplicação (Figura 57).



**Figura 57 - Instalação do STAR v. 2.0**

O processo termina ao fim de alguns segundos, sendo o utilizador notificado que a instalação da segunda versão do STAR ocorreu com sucesso (Figura 58), podendo terminar a instalação.



**Figura 58 - Instalação do STAR v. 2.0**

A instalação da aplicação implementada cria um atalho no ambiente de trabalho para poder utilizar a ferramenta de uma forma mais simples.

Ao fim de realizado todo este processo, a segunda versão do STAR fica pronta para utilização, estando automaticamente configurada (conexão) para acesso a uma base de dados remota.

Se o novo utilizador pretender utilizar uma base de dados local, precisa de realizar mais alguns passos para o poder fazer. Primeiramente efectuar o *download* e instalação do PostgreSQL versão 9.0.5.1, para o sistema operativo específico (<http://www.enterprisedb.com/products-services-training/pgdownload>). Este pacote de instalação inclui o servidor base de dados, o *pgAdmin* e o *StackBuilder*.

A instalação decorre de forma idêntica à apresentada anteriormente, sendo que o utilizador antes do processo se iniciar tem de indicar uma *password* para o serviço *postgres*, como demonstra a Figura 59 (e.g. password = star).

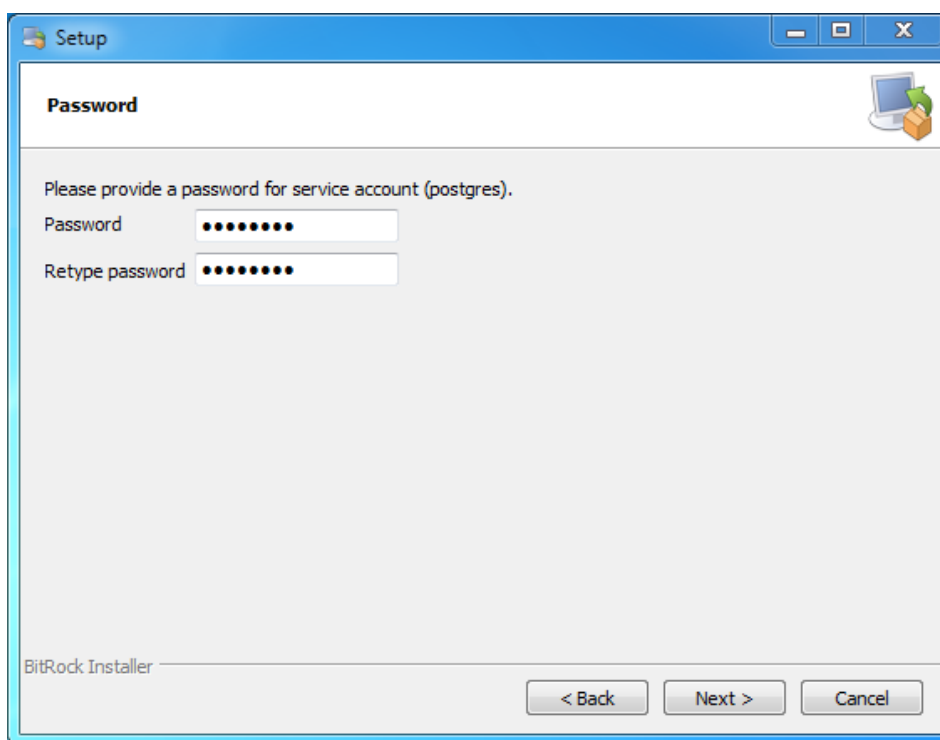
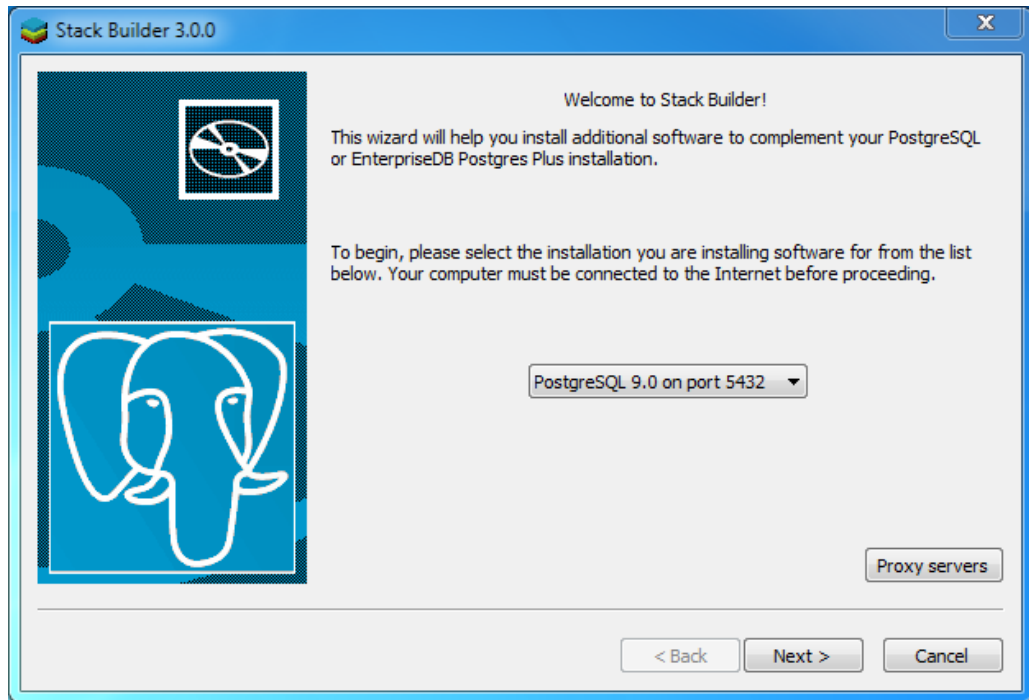


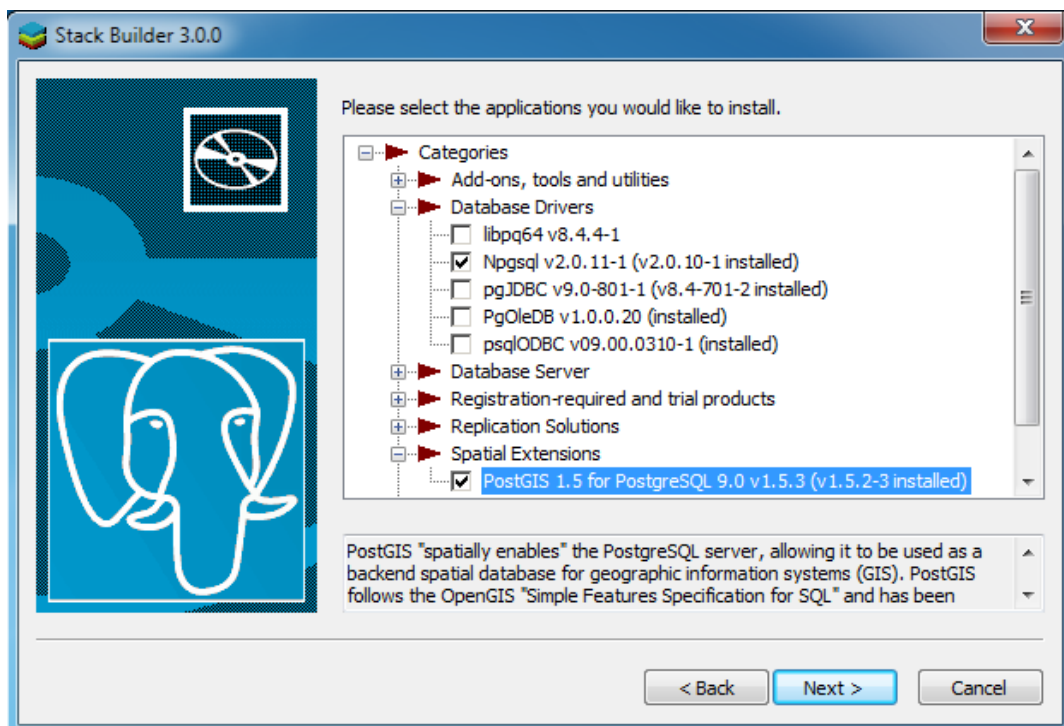
Figura 59 - Instalação PostgreSQL

Finalizada a instalação, é despoletada uma janela para autorização da abertura da componente *StackBuilder*, que auxilia a instalação de *software* adicional necessário como o *PostgreSQL*. O utilizador autoriza o processo e de seguida é apresentada a janela ilustrada na figura 60, onde o utilizador indica a porta e qual a versão do *PostgreSQL* instalado.



**Figura 60 - StackBuilder**

De seguida são ilustradas um conjunto de aplicações adicionais, que podem ser instaladas, sendo que as obrigatórias para o funcionamento da segunda versão do STAR são as ilustradas na Figura 61.



**Figura 61 - Aplicações Adicionais: StackBuilder**

Realizado este processo, o utilizador necessita apenas de um último passo para poder usar a segunda versão do STAR com uma base de dados remota. Para tal, é necessário importar o ficheiro *star.backup*, que se encontra na pasta onde foi instalado o STAR 2.0. Este ficheiro tem toda a estrutura de base de dados usada pelo sistema.

Para importar, o utilizador terá de abrir o *pgAdminIII*, anteriormente instalado, criar uma nova base de dados, onde bastará indicar o nome e o proprietário (Figura 62), e de seguida importar o ficheiro que contém a *backup* da base de dados (*star.backup*), como está ilustrado na Figura 63.

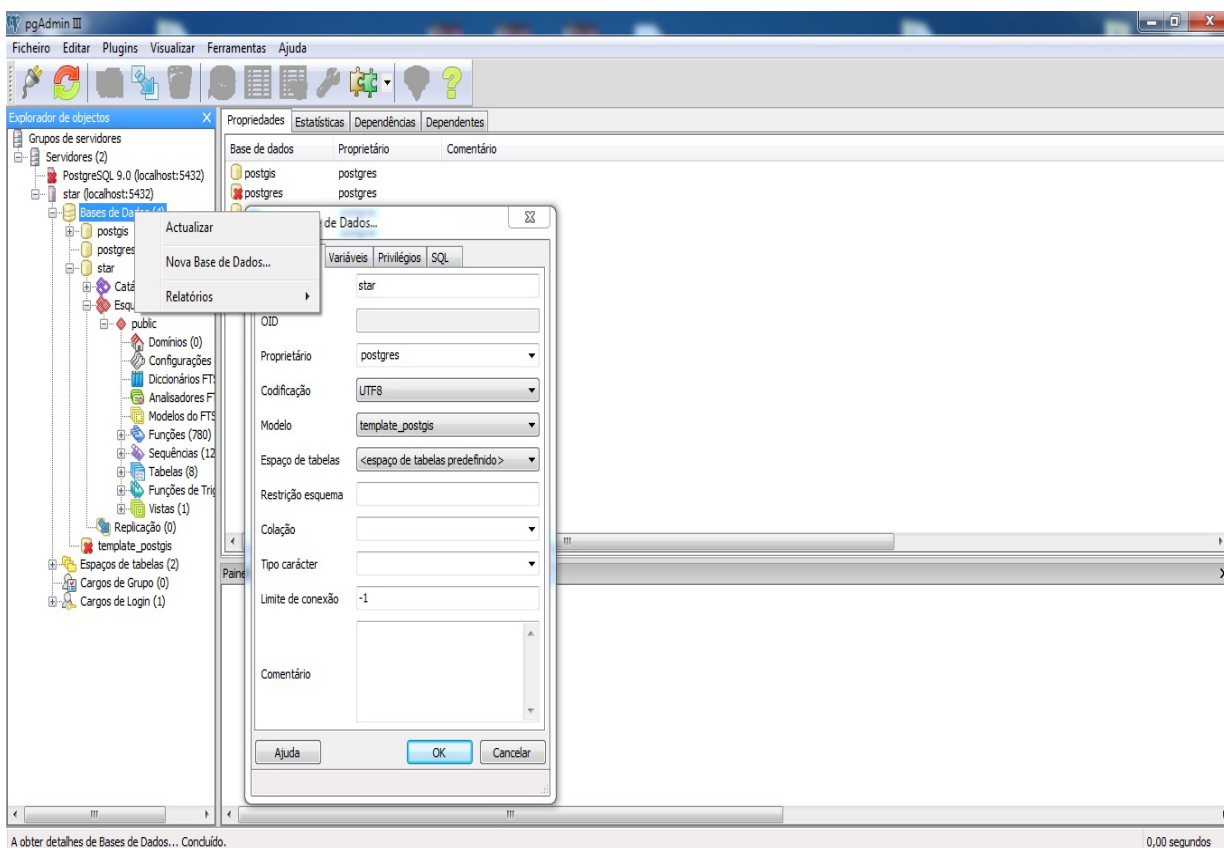


Figura 62 – PgAdminIII: Nova Base de Dados



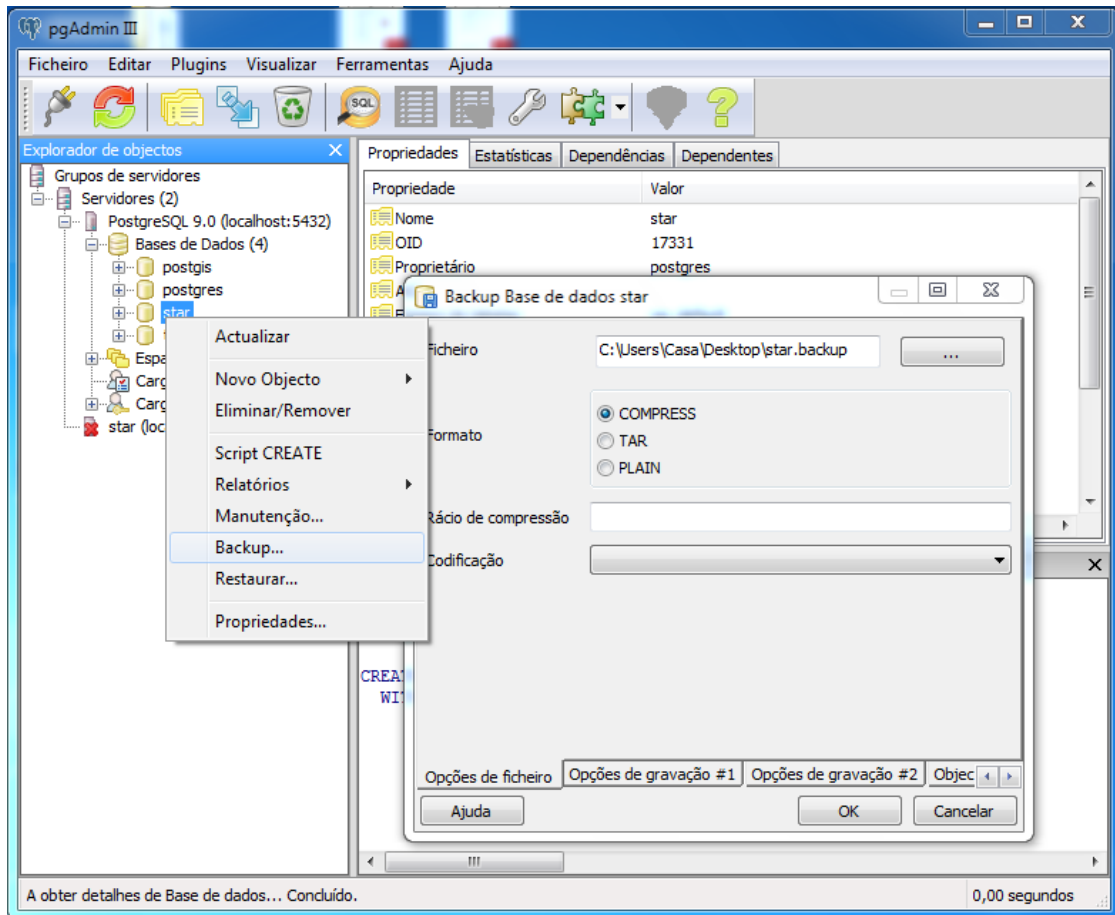


Figura 63 – PgAdminIII: Importar Backup

### 5.1.2. Apresentação da Interface do STAR v2.0

A interface gráfica do protótipo implementado tinha como requisito ser simples e intuitiva (*user-friendly*). Esta pode ser utilizada por vários utilizadores, exibindo conjuntos de dados, análises e visualizações distintas, mas sempre mantendo a mesma estrutura.

A interface do utilizador está dividida em quatro elementos (Figura 64): a área dos botões e menus (superior esquerdo), o gestor de dados (*Tree*), os controlos (inferior esquerdo) e a área de visualização. Cada uma das áreas especificadas cumpre tarefas distintas tornando a interface simples e bastante intuitiva de utilizar.

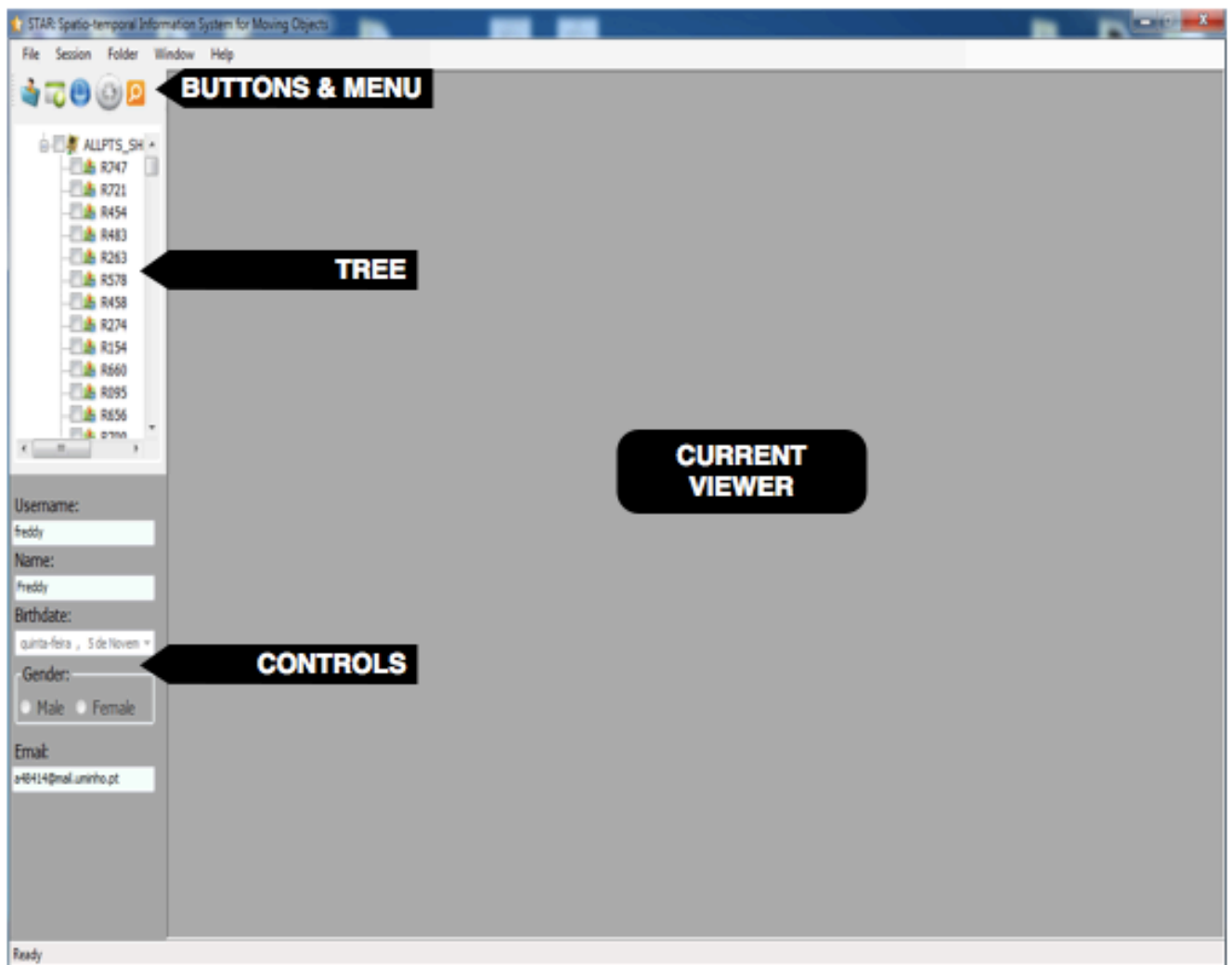


Figura 64 - STAR v 2.0: Interface Utilizador

De seguida todas as componentes que interagem com o utilizador são explicadas separadamente, referindo todas as funções das mesmas. Serão ilustradas Figuras para melhor entendimento de cada uma das componentes. Todas as funcionalidades identificadas na explicação de cada componente, serão descritas detalhadamente na secção seguinte (Secção 5.2), sendo demonstrado o funcionamento das mesmas.

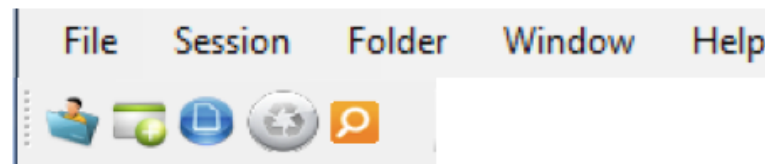
### ***Buttons & Menu***

Na área dos botões e menu, o utilizador pode realizar uma série de funcionalidades já descritas (Secção 3.4) e implementadas neste protótipo.

Ao aceder ao menu, o utilizador tem cinco opções (*File, Session, Folder, Window e Help*) e respectivas subopções (Figura 65). A primeira opção apresenta quatro subopções: permite ao utilizador importar um novo ficheiro para o sistema (*Import*), listar o conjunto de ficheiros

existentes no sistema (*List*), pesquisar por um ficheiro (*Search*) ou sair da aplicação (*Exit*). A segunda opção possibilita ao utilizador verificar a sua conta podendo realizar alterações à mesma se assim o entender (*Account*). Este poderá alterar o nome, data de nascimento, *email*, género e *username*. Esta opção apresenta ainda a possibilidade de sair do sistema (*Logout*).

A terceira opção apresenta um conjunto de funcionalidades sobre as pastas do sistema. O utilizador poderá criar uma nova pasta (*New*), realizar uma pesquisa de pastas no sistema (*Search*) e listar (*List*) todas as pastas existentes. Na quarta opção o utilizador tem várias funcionalidades de gestão das janelas do sistema presentes na componente *current viewer*. Assim este poderá organizar as janelas verticalmente, horizontalmente ou em escada (*Tile Horizontal, Vertical, Cascade*). Existe ainda uma quinta opção que não foi desenvolvida, que terá como intuito apresentar um manual de ajuda à utilização da ferramenta, explicando todos os componentes e funcionalidade (*Help*).



**Figura 65 – STAR v. 2.0: Componente *Buttons & Menu***

Nos botões o sistema apresenta cinco opções/ funcionalidades distintas. Algumas funcionalidades são idênticas às pertencentes nos menus, só que com características de acesso conferindo versatilidade ao utilizador do sistema.

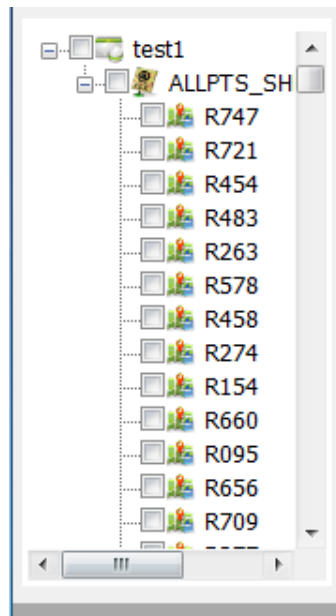
O primeiro botão (*User AccountDetails*) permite ao utilizador visualizar os dados da sua conta. Este pode ver todos os dados recolhidos no momento do registo excepto a *password* especificada, podendo fazer alterações aos mesmos se assim o entender. O segundo botão (*New Folder*) permite criar uma pasta no sistema, sendo esta funcionalidade idêntica à pertencente ao menu.

O terceiro botão (*Import File*) permite importar um ficheiro para a sistema. O quarto e quinto botão (*Refresh Database* e *Data Cleanning*, respectivamente) permitem ao utilizador actualizar a componente *Tree* do sistema e corrigir dados de um ficheiro seleccionado.

## **Tree**

A componente *tree* apresenta ao utilizador todas as pastas, ficheiros e dispositivos importados para o sistema. Esta apresenta os dados (*Folders, Files, Devices*) na forma de árvore, permitindo ao utilizador seleccionar cada um deles (Figura 66).

Ao efectuar *Login* no sistema esta árvore é carregada automaticamente, apresentando todas as pastas existentes no sistema e ficheiros que o utilizador tem acesso. Por acesso, entende-se ficheiros importados pelo próprio utilizador, ou ficheiros classificados como públicos no sistema.



**Figura 66 - STAR v. 2.0: Componente *Tree***

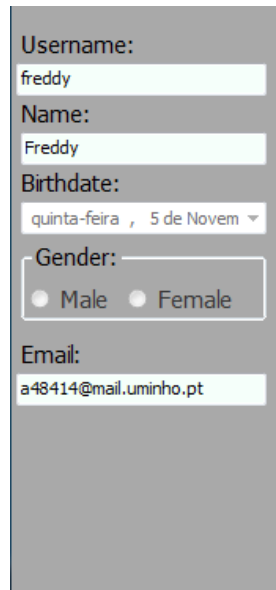
O utilizador ao seleccionar uma das componentes da árvore desencadeia o aparecimento de diferentes controlos na componente *Controls*. De referir que se o utilizador seleccionar um ficheiro, todos os dados contidos nele são automaticamente seleccionados. Para actualizar esta lista basta clicar no botão *refreshDatabase* que está na componente acima descrita (*buttons & menu*).

## **Controls**

A componente controlos (*Controls*) apresenta um conjunto de informações e funcionalidades acerca do utilizador e dos constituintes da componente *Tree*. O conjunto de funcionalidades específico dos dados pertencentes à árvore é apresentado na forma de botões.

Existem quatro tipos de controlos, sendo cada um deles desencadeado por acções específicas. O primeiro controlo é apresentado assim que é efectuada a entrada no sistema por parte de um

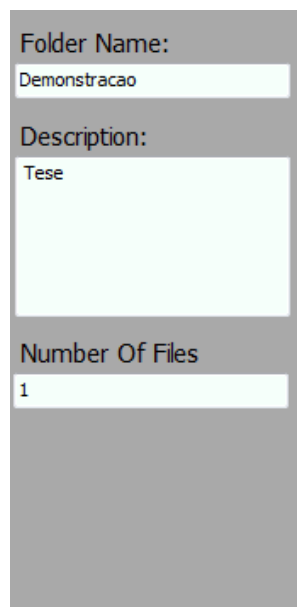
utilizador. Este descreve um conjunto de informações acerca do utilizador que efectuou o *Login* (Figura 67).



A vertical form with a grey background. It contains the following fields: 'Username:' with the value 'freddy'; 'Name:' with the value 'Freddy'; 'Birthdate:' with a dropdown menu showing 'quinta-feira , 5 de Novem'; 'Gender:' with two radio buttons labeled 'Male' and 'Female'; and 'Email:' with the value 'a48414@mail.uminho.pt'.

**Figura 67 - STAR v. 2.0: User Account Details Control**

O segundo controlo apresenta informações acerca do arquivo (*folder*) seleccionado pelo utilizador. É ilustrado o nome do mesmo, a sua descrição e o numero de ficheiros que este contém (Figura 68).

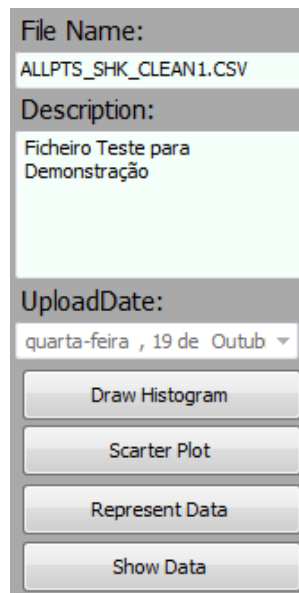


A vertical form with a grey background. It contains the following fields: 'Folder Name:' with the value 'Demonstracao'; 'Description:' with the value 'Tese'; and 'Number Of Files' with the value '1'.

**Figura 68 - STAR v. 2.0: Folder Control**

O terceiro controlo é projectado a partir da selecção de um ficheiro (Figura 69). Este apresenta o nome do ficheiro seleccionado, a descrição do mesmo e a data em que foi importado para o sistema. Para além disto, apresenta um conjunto de funcionalidades, através

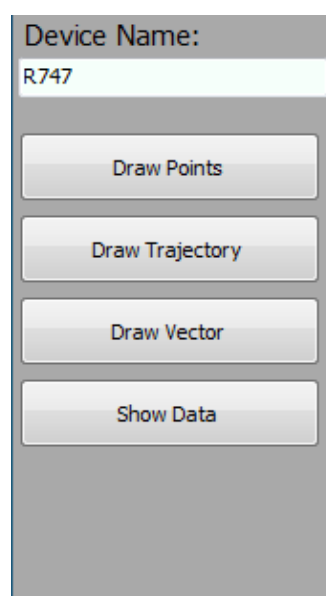
de botões, descritas nos capítulos anteriores (Capítulo 3 e 4) como, por exemplo, analisar dados. O utilizador pode representar os dados contidos num ficheiro, pedir um histograma ou um gráfico de dispersão.



The screenshot displays a vertical control panel for file management. At the top, the 'File Name' field contains 'ALLPTS\_SHK\_CLEAN1.CSV'. Below it, the 'Description' field contains 'Ficheiro Teste para Demonstração'. The 'UploadDate' field is a dropdown menu showing 'quarta-feira, 19 de Outub'. At the bottom of the panel, there are four stacked buttons: 'Draw Histogram', 'Scarter Plot', 'Represent Data', and 'Show Data'.

**Figura 69 - STAR v2.0: File Control**

O quarto e último controlo apresenta descrições e funcionalidades relativas às entidades móveis (*dados*) de um ficheiro (Figura 70). As funcionalidades são apresentadas através de três botões, onde é possível ao utilizador seleccionar várias entidades (dados) e representar os pontos no mapa, desenhar as suas trajectórias e vectores. Estas funcionalidades são descritas nos capítulos anteriores com as opções de visualização de dados (Capítulo 3 e 4).



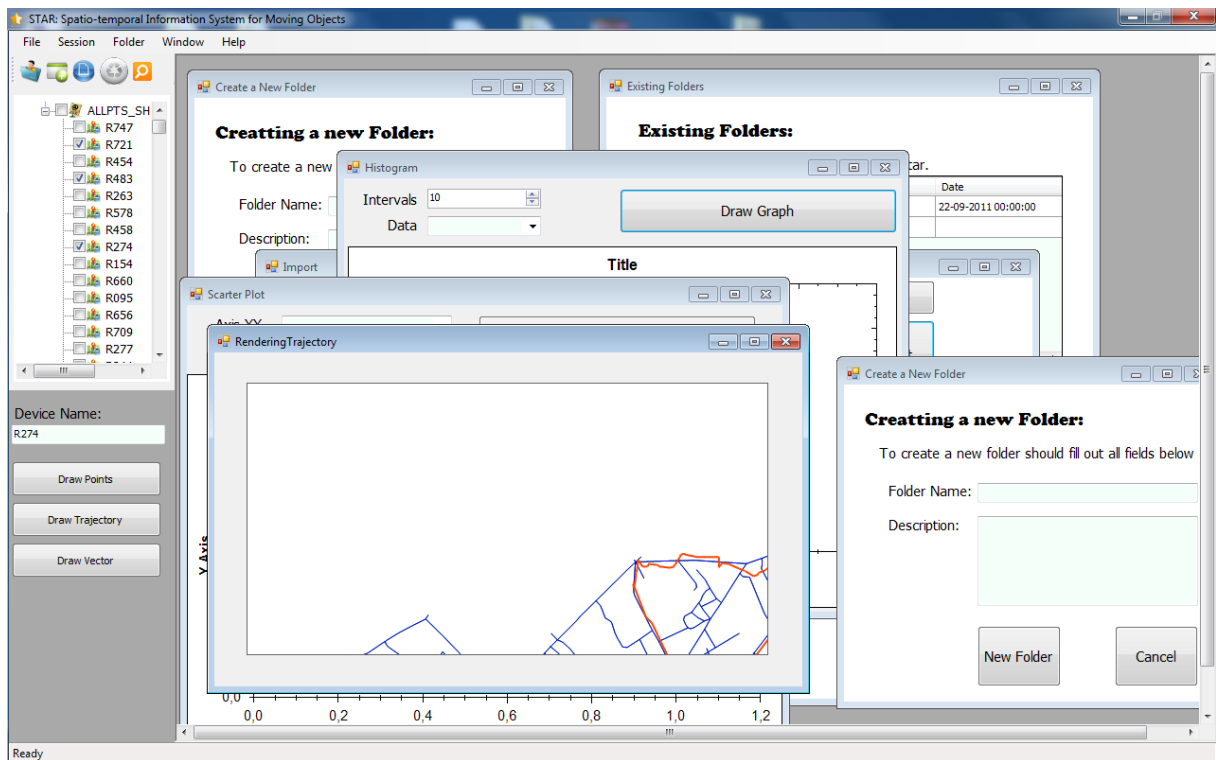
The screenshot displays a vertical control panel for device management. At the top, the 'Device Name' field contains 'R747'. Below it, there are four stacked buttons: 'Draw Points', 'Draw Trajectory', 'Draw Vector', and 'Show Data'.

**Figura 70 - STAR v. 2.0: Device Control**

## ***Current Viewer***

A área de visualização (*current viewer*) é onde estão os resultados de todas as funcionalidades presentes no sistema. Esta área apresenta todas as janelas de interacção do sistema com o utilizador, todos os resultados das análises e visualizações requeridos. As várias janelas (funcionalidades) podem ser arrastadas por toda a área, sendo possível ter várias janelas abertas em simultâneo (Figura 71).

Qualquer interacção entre o utilizador e o sistema que seja realizada em uma janela é projectada nesta componente. O utilizador pode usar a funcionalidade presente na componente *Buttons & Menu*, de organizar as janelas verticalmente, horizontalmente ou em cascata, para uma melhor percepção destas.



**Figura 71 - STAR v. 2.0: *Current Viewer***

## **5.2. Apresentação das Funcionalidades do STAR v. 2.0**

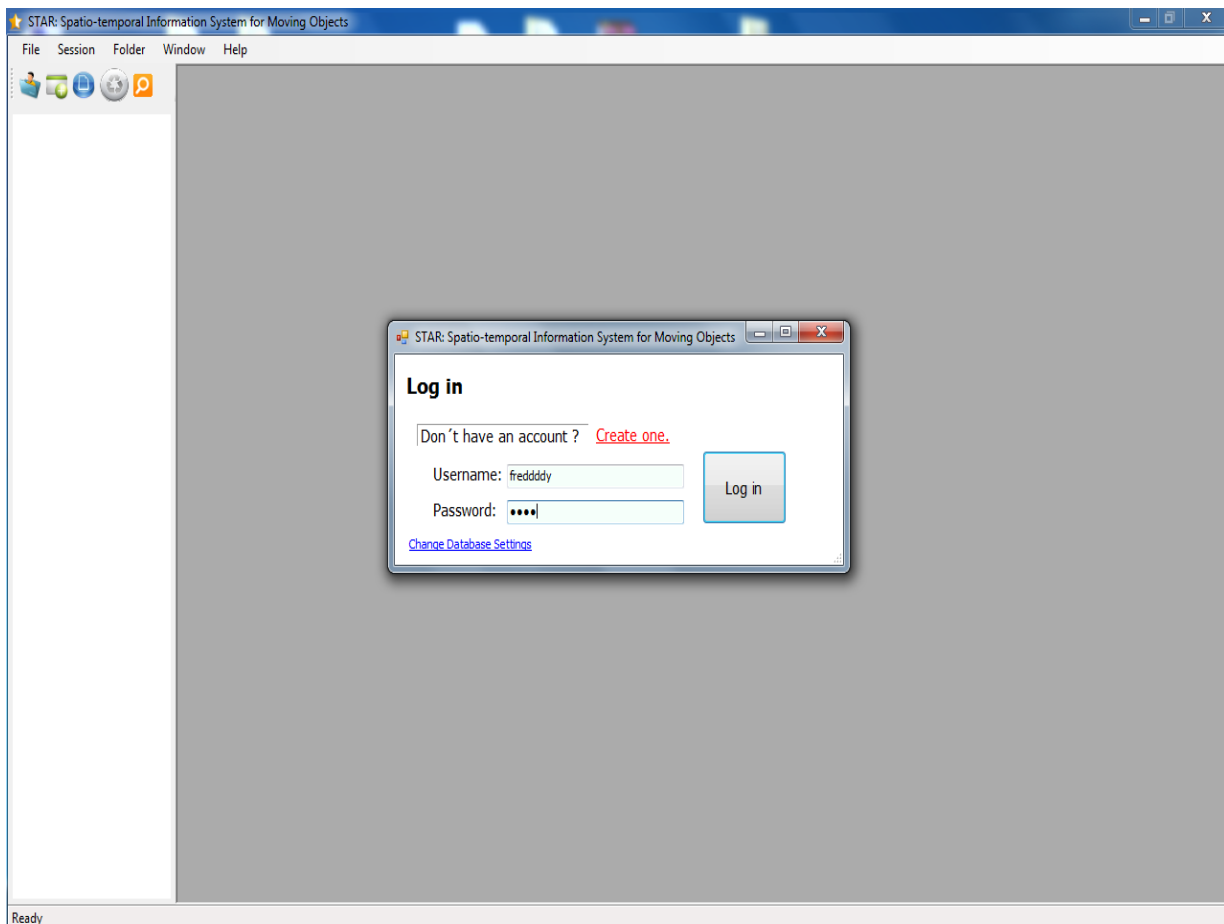
Esta secção pretende apresentar todas as funcionalidades implementadas no protótipo da segunda versão do STAR. Serão descritas todas as funcionalidades analisadas nos capítulos anteriores, ilustrando o funcionamento das mesmas recorrendo a *print screens* do sistema em execução.

Para melhor acompanhamento da demonstração das funcionalidades implementadas, esta secção está organizada da mesma forma (com a mesma estrutura) que a descrição das funcionalidades (diagramas de caso de uso) realizada no capítulo 3 (Secção 3.4.2).

### 5.2.1. Autenticar

A Figura 72 apresenta o ecrã de autenticação do STAR. Como se pode verificar este é bastante simples e intuitivo. Esta janela apresenta três opções ao utilizador: efectuar *login*, registar novo utilizador e alterar conexão à base de dados.

Para efectuar a *login* (utilizador já registado), basta apenas inserir os dados e seleccionar o botão de *Log in* (Figura 72). Enquanto nenhum utilizador estiver autenticado, não é possível usar qualquer funcionalidade do sistema, uma vez que estas estão todas desactivas.



**Figura 72 - STAR v. 2.0: Autenticação**

Se o sistema não foi capaz de executar uma determinada instrução indicada pelo utilizador ou se a mesma foi executada com sucesso, então o sistema emite uma mensagem com o resultado da operação (Figura 73). Este processo é idêntico em todas as funcionalidades do mesmo.



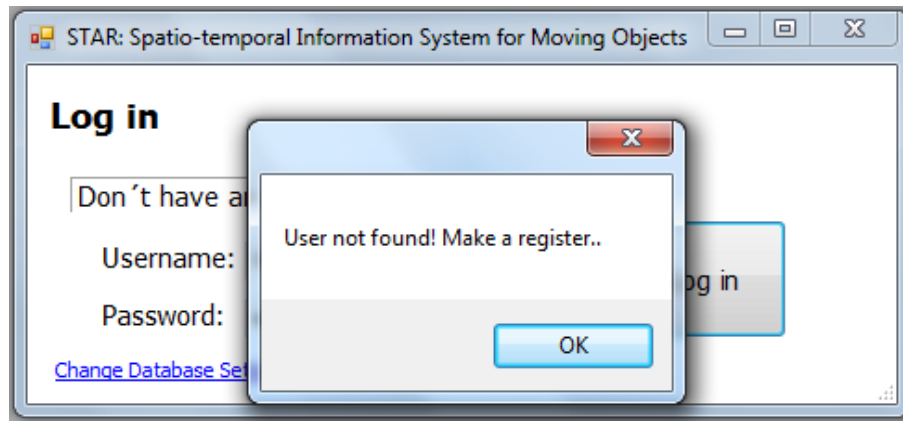


Figura 73 - STAR v. 2.0 – Erro Autenticação

A Figura 74 ilustra o formulário de registo de um novo utilizador devidamente preenchido. Se todos os dados estiverem correctos o sistema insere o novo utilizador na BD.

A screenshot of the STAR v. 2.0 registration form. The window title is "STAR: A Spatio-temporal Information System for Moving Obj...". The main content area is titled "Your Account" and contains the text "You wil use this information to Login to your STAR account". The form fields are: "Username:" (demonstracao), "Password:" (demo), "Name:" (demonstracao), "Birthdate:" (sábado, 8 de Outubro de 2011), "Gender:" (Male selected, Female unselected), and "Email:" (a48414@alunos.uminho.pt). A "Create My Account" button is located at the bottom right.

Figura 74 - STAR v. 2.0: Novo utilizador

A última opção presente na funcionalidade de autenticação é a de alterar a conexão à base de dados. A Figura 75 ilustra a alteração da conexão à base de dados. Ao seleccionar o botão para guardar (*Save*) a nova configuração fica armazenada para a próxima execução do sistema.

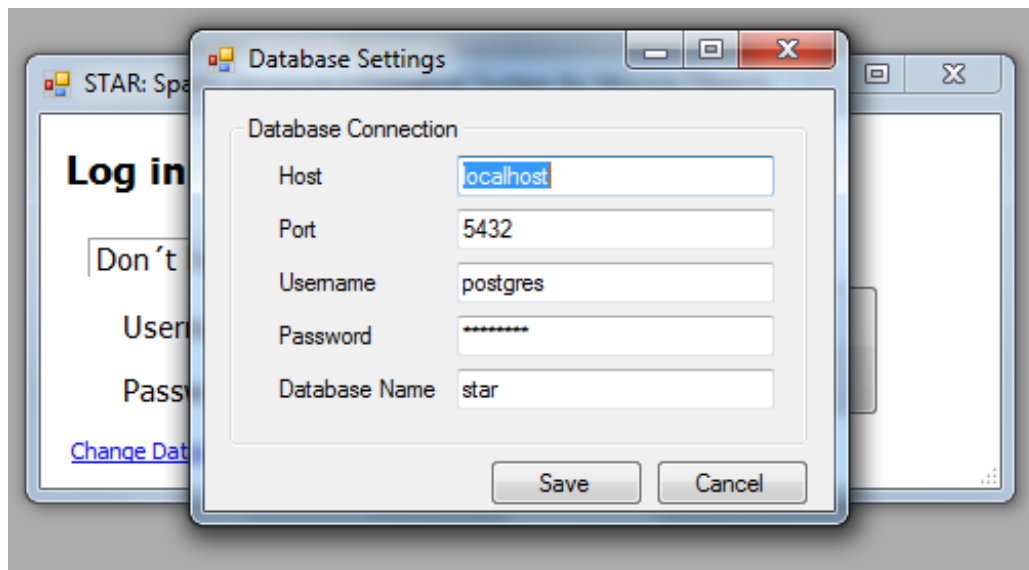


Figura 75 - STAR v. 2.0: Alteração da conexão à BD

Como referido, caso os dados inseridos para a nova conexão estejam incorrectos, o sistema envia uma mensagem de erro (Figura 76), possibilitando de seguida nova tentativa de autenticação ou de alteração da configuração.

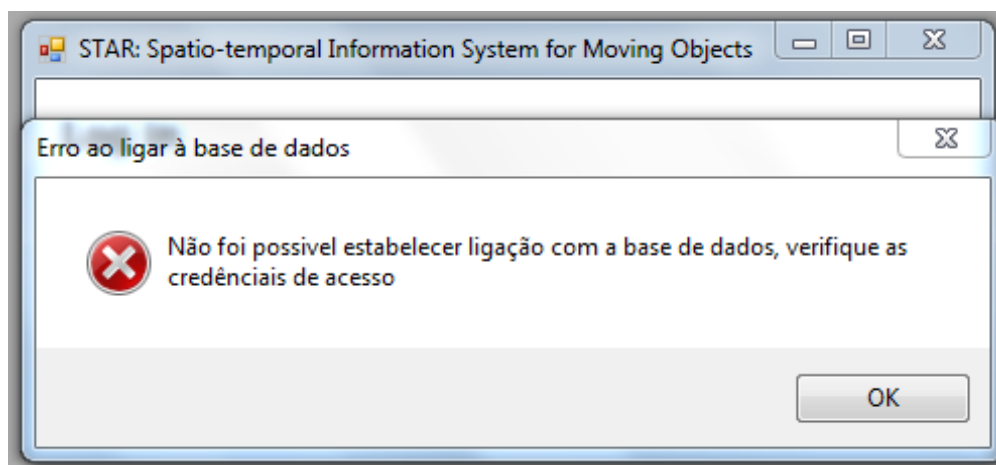


Figura 76 - STAR v. 2.0: Erro Configuração Conexão

### 5.2.2. Gerir Arquivos e Gerir Ficheiros

O Utilizador poderá criar/importar, pesquisar e listar arquivos e ficheiros. Estas funcionalidades foram agrupadas na mesma secção, pois apresentam estruturas e mecanismos de funcionamento semelhantes. A grande diferença nas funcionalidades apresentadas por ambos os grupos consiste no facto que para criar arquivos o mecanismo é relativamente simples, já a criação de ficheiros, engloba mais processos, sendo ambos demonstrados nesta

secção. Por similaridade de processos e de interface (e.g. listar arquivos e listar ficheiros, pesquisar arquivos e pesquisar ficheiros) serão apenas apresentadas imagens de um dos grupos sendo as restantes figuras ilustradas no anexo 2.

A Figura 77 apresenta o formulário de criação de um novo arquivo devidamente preenchido. Depois de seleccionar o botão *New Folder*, o sistema introduz o novo arquivo na BD. O utilizador pode também cancelar a criação seleccionando o botão *Cancel*.

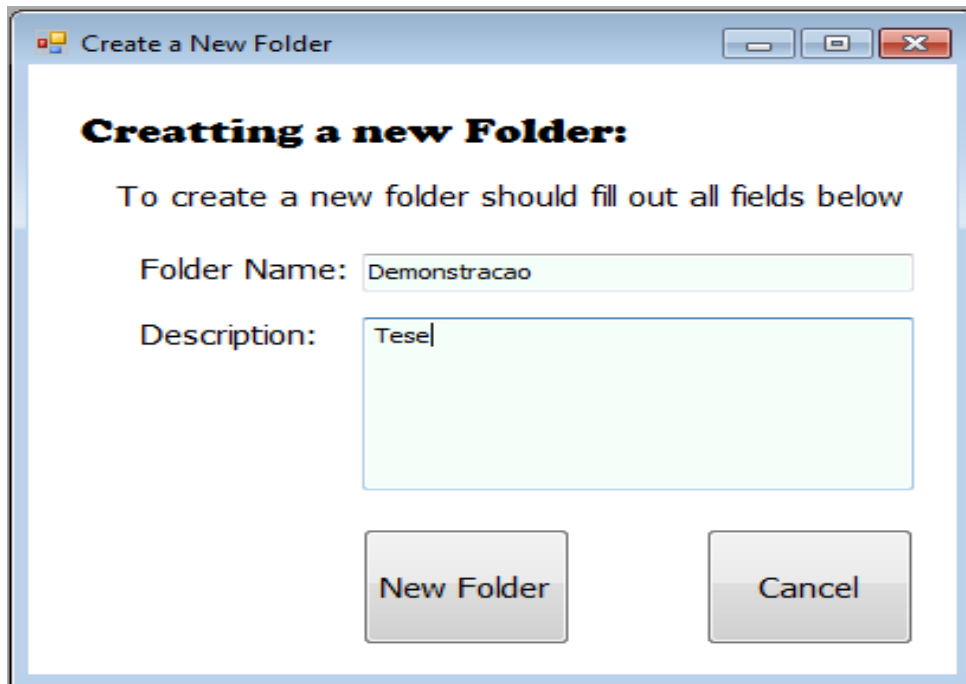
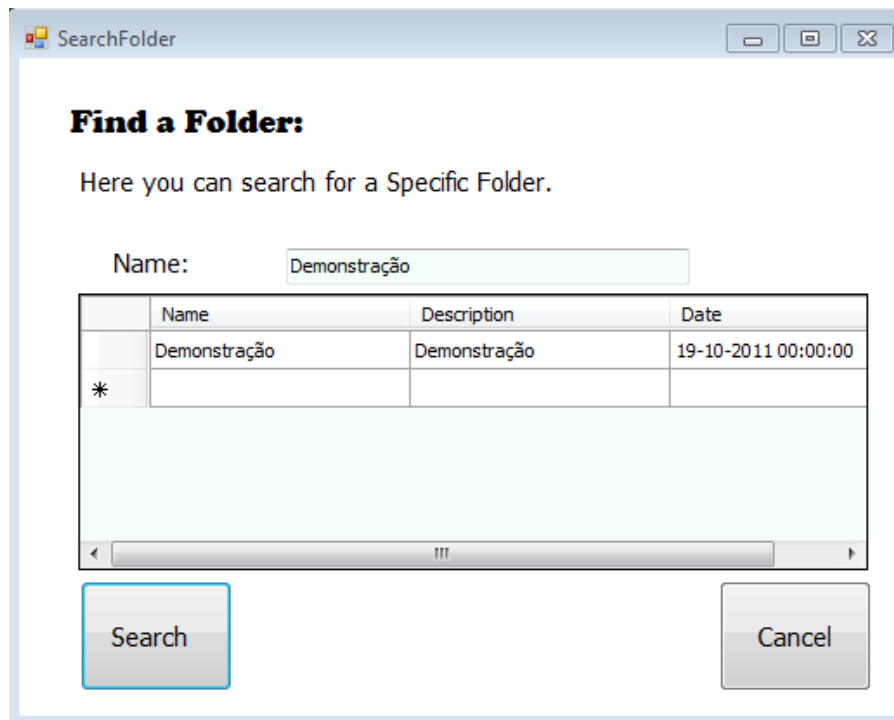


Figura 77 - STAR v. 2.0: Criar novo Arquivo

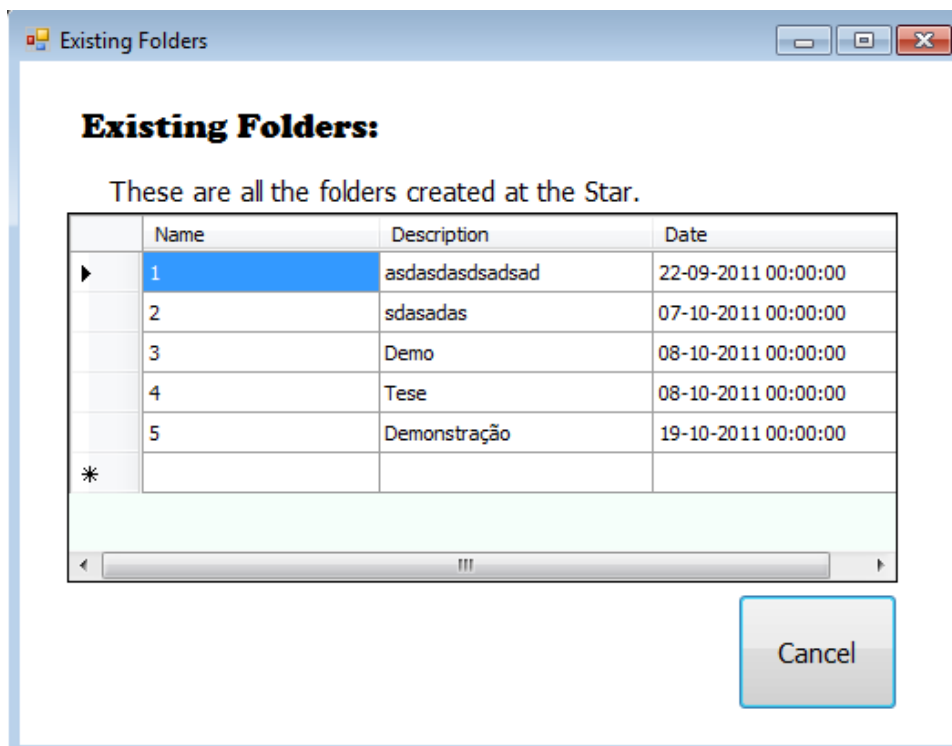
A Figura 78 apresenta o resultado de uma pesquisa feita através do nome do Arquivo. O sistema apresenta o resultado sobre a forma de lista, apresentando alguns detalhes sobre o arquivo encontrado. O utilizador pode em qualquer momento pressionar o botão *Cancel* ou sair da janela fechando-a.



**Figura 78 - STAR v. 2.0: Pesquisar Arquivos**

A pesquisa de Ficheiros está representada no Anexo 2: Figura 1.

O utilizador poderá ainda listar arquivos no sistema. Na Figura 79 está representada a listagem dos arquivos visíveis ao utilizador. A qualquer momento o utilizador pode cancelar a funcionalidade através do botão *Cancel*.



**Figura 79 - STAR v. 2.0: Listar Arquivos**

A listagem de ficheiros está ilustrada na Figura 2 do Anexo 2.

Na Figura 80 está representado o formulário, correctamente preenchido, que permite importar um novo ficheiro para o sistema. Para importar um ficheiro o utilizador tem que preencher um conjunto de opções no formulário apresentado.

Primeiramente, o utilizador tem que indicar qual o arquivo a que pretende adicionar o ficheiro (Lista: *Folder Name*), classificar o ficheiro como publico ou privado (Opções *Public: Yes* ou *No*), introduzir uma descrição do ficheiro a importar e se pretender indicar um mapa onde deseja visualizar os dados (*ShapeFile Name*). De seguida o utilizador tem duas opções obrigatórias para o correcto funcionamento desta opção de importação. Este poderá introduzir manualmente todos os atributos presentes no ficheiro a importar especificando qual a sua posição no mesmo (*Columns Positions*), ou então realizar esta tarefa automaticamente através da funcionalidade *Infer*. Ao seleccionar o botão *Infer* o utilizador terá de introduzir a directoria do ficheiro a importar, sendo de seguida demonstrado o resultado das posições dos atributos do ficheiro nos campos *Columns Positions* do formulário. O utilizador poderá sempre corrigir alguma posição de algum atributo que não esteja correcto ou de acordo com o que este pretende. Sempre que algum atributo presente no ficheiro não esteja detalhado no formulário, o utilizador poderá seleccionar o botão *Add Other Field*, seleccionando um atributo presente na lista apresentada.

A lista do lado esquerdo (*Device ID, DateTime, CoordX, etc.*) representam os atributos fixos de importação, sendo estes armazenados na Tabela *data* da BD desenvolvida. A lista do lado direito é extensível representando um conjunto bastante alargado de atributos, que resultaram da análise de vários ficheiros acerca de entidades móveis (Anexo 2: Figura 3). Estes dados são armazenados na Tabela *otherdata*.

Ao fim de preenchido todo o formulário a importação dos dados para o sistema inicia-se com a selecção do botão *Import*, que poderá pedir a directoria do ficheiro a importar se o utilizador optar por introduzir manualmente as posições do atributo. Se este optar pela funcionalidade *infer* o processo inicia-se automaticamente.

**Importing a new File**

**To Import a new File you must indicate the file path. Clicking the button.**

ShapeFile Name: C:\Users\Freddy\Desktop\Dados\DNP\_PathNetwork\pathnetwork\_ ...

Folder Name: Demonstração

Description: Demonstração

Public:  Yes  No

Infer

Import

Cancel

**Columns Positions**

Device ID	Position	1
Date Time	Position	2
Coord X	Position	4
Coord Y	Position	5
Distance	Position	10
Bearing	Position	8
Seconds	Position	11
Speed	Position	9
Previous Point	Position	12
Sequence	Position	13
Point ID	Position	3

Add Other Field

Type	OriginX	Position	6
Type	OriginY	Position	7
Type	Flag	Position	14

Figura 80 - STAR v. 2.0: *Import File*

Para além da sua função crucial neste sistema, permitir a importação de dados, esta funcionalidade apresenta duas características bastante importantes para o bom e eficiente funcionamento do sistema:

- **Formulário dinâmico:** O formulário desta funcionalidade não é fixo, este depende do ficheiro a importar, pois os atributos de cada ficheiro variam tanto em número (mais ou menos atributos) como em posição no mesmo;
- **Funcionamento em *Background*:** Como esta funcionalidade pode demorar alguns minutos dependendo do tamanho do ficheiro a importar e do tipo de conexão realizada, esta foi implementada de modo a que o utilizador possa continuar a usar o sistema enquanto está a ser executada a importação de todos os dados para o sistema. Assim o formulário apresenta uma barra de progresso (canto inferior direito) que ilustra o progresso da importação dos

dados para o sistema. Enquanto este decorre (Processo em *Background*) o utilizador poderá realizar qualquer outra funcionalidade no sistema.

A Figura 81 mostra a importação de um ficheiro com mais atributos que a figura apresentada anteriormente. A realização de outras tarefas ao mesmo tempo que o sistema importa os dados, está ilustrada no Anexo 2: Figura 4.

**Importing a new File**

**To Import a new File you must indicate the file path. Clicking the button.**

ShapeFile Name: C:\Users\Freddy\Desktop\Dados\Amsterdam\_shp\OSM\_Streets\_W ...

Folder Name: Demonstração

Description: Ficheiro Teste: demonstração

Public:  Yes  No

Infer

Import

Cancel

**Columns Positions**

Field Name	Position	Type	Position
Device ID	8	Team	2
Date Time	24	ZoneName	3
Coord X	9	Label	4
Coord Y	10	Latitude	5
Distance	11	Longitude	6
Bearing	14	Longitude	7
Seconds	12	MoveX	16
Speed	13	MoveY	17
Previous Point	23	SHKDistance	20
Sequence	15		
Point ID	23		

Add Other Field

Figura 81 - STAR v. 2.0: Import File

Quando um ficheiro é totalmente importado o sistema envia uma mensagem confirmando o sucesso da operação emitindo ao mesmo tempo um som sinalizador do mesmo (Anexo 2: Figura 5).

### 5.2.3. Gerir Dados

O Utilizador poderá realizar três tipos de funcionalidades: Visualização de Dados, Análise de Dados e Correção de Dados. Estas funcionalidades foram agrupadas na mesma secção, pois todas realizam operações sobre os dados importados para o sistema.

As funcionalidades de visualização de dados dividem-se em quatro tipos de visualizações: visualização de vectores, visualização de trajectória, visualização de dados no mapa e visualizações dados em tabela. Sempre que um utilizador seleccionar um ou vários registos (entidades móveis) na componente *Tree*, a componente de controlo (*Control*) apresenta quatro botões referentes à visualizações de dados: *Draw Points*, *Draw Trajectory*, *Draw Vector* e *Show Data*.

As três primeiras funcionalidades são apresentadas em uma tela branca ou recorrendo ao mapa (*shapefile*) do ficheiro importado. Esta representação contém ainda três botões que permitem ampliar, reduzir e voltar à representação original (*Zoom In*, *Zoom Out*, *Zoom to Full*), e informações dos valores das coordenadas do mapa (canto inferior direito), processo idêntico à representação dos dados (Ficheiros: *Represent Data*).

O utilizador ao seleccionar o botão *Draw Trajectory* o sistema representa as trajectórias do ou das entidades móveis seleccionadas. Na Figura 82, o sistema representa a trajectória de três entidades móveis (R274, R095 e R709) seleccionados na componente *Tree*.

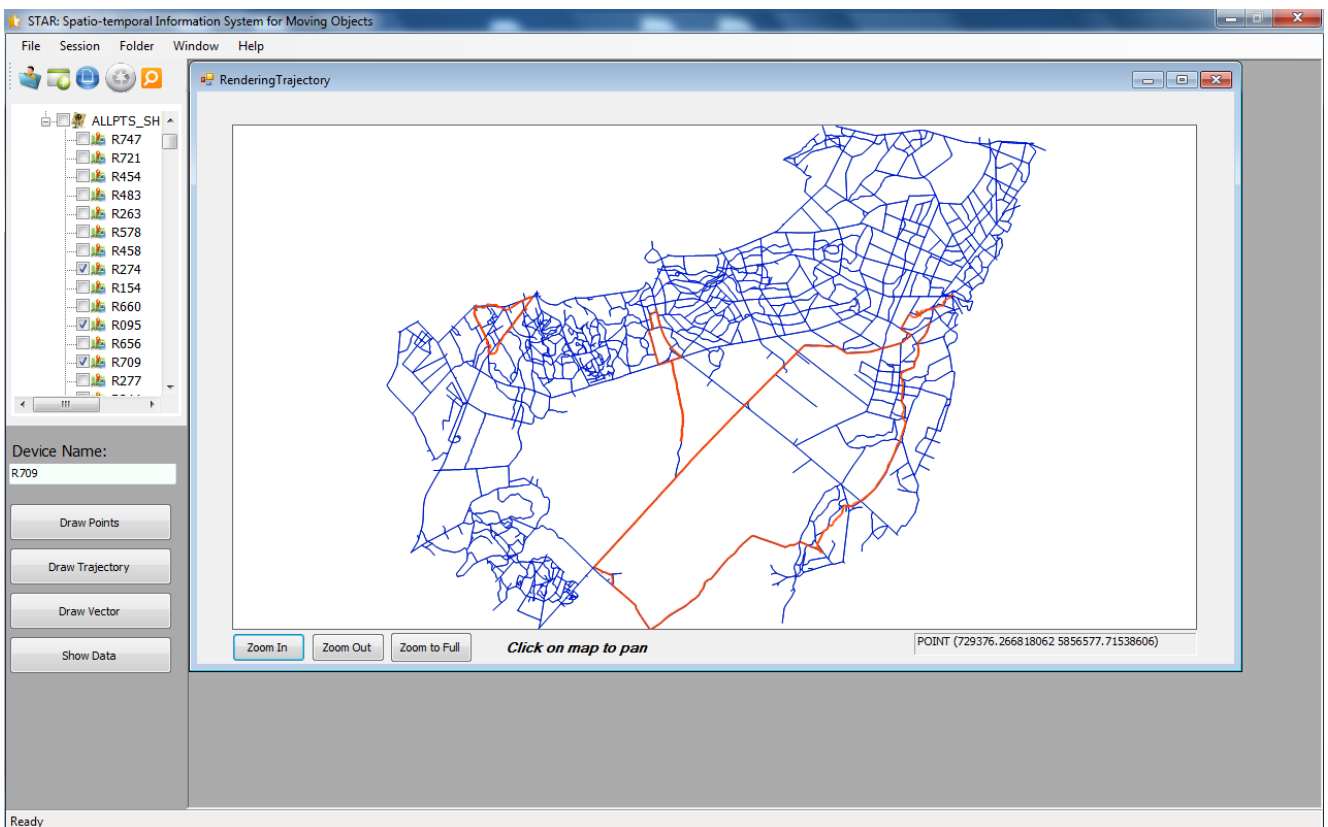
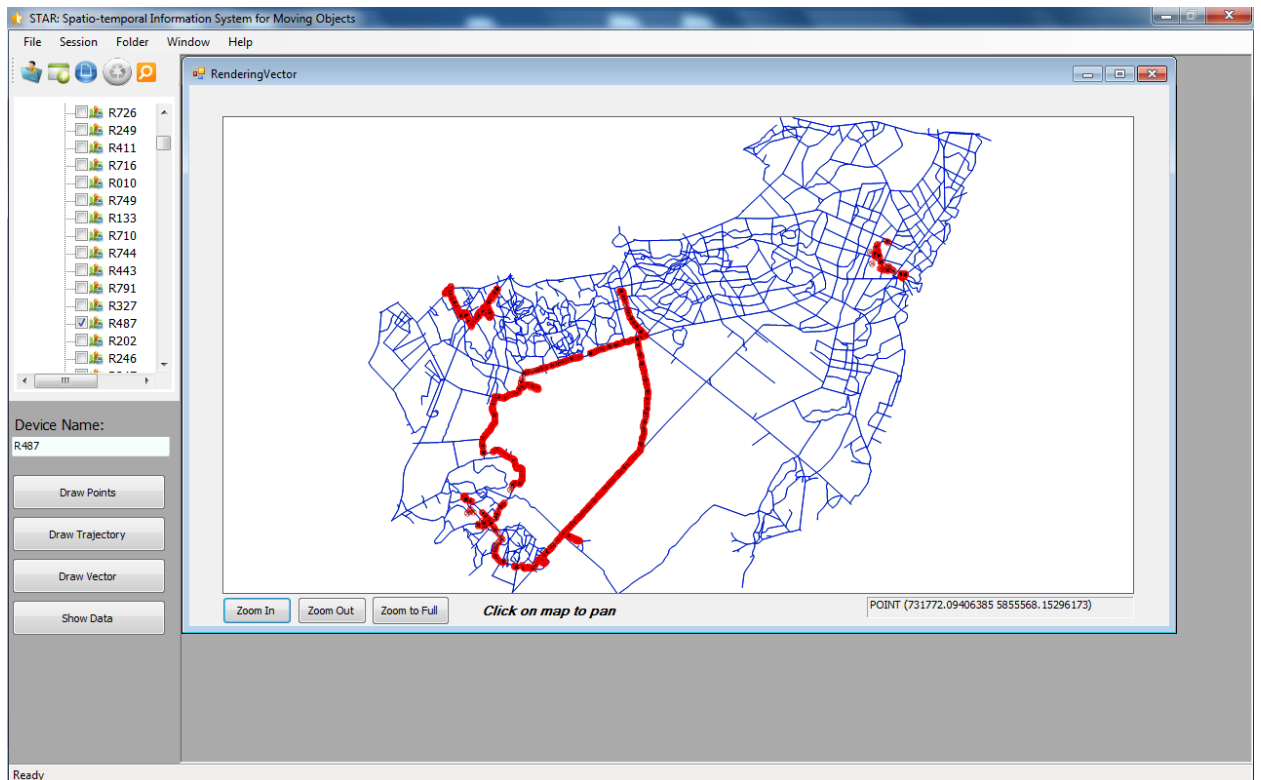


Figura 82 - STAR v. 2.0: *Draw Trajectory*



Na Figura 6 do Anexo 2 está representada a trajetória da entidade móvel com a identificação R095, sendo que neste caso foi utilizada a opção *Zoom In* para ampliar a trajetória apresentada.

O processo de visualização dos vectores dos pontos representados no mapa é muito similar à representação de trajetórias. Basta seleccionar os dados na componente *Tree* e carregar no botão *Draw Vector*. O sistema apresenta os vectores direccionais dos pontos, ou seja, estes vectores permitem verificar qual a direcção que determinado utilizador segue até atingir um novo ponto no seu percurso. Na Figura 83 estão representados os vectores direccionais dos pontos de três entidades (R487, R721, R277).



**Figura 83 - STAR v. 2.0: Draw Vector**

Dada a dificuldade de análise da Figura anterior, foi utilizada a opção de *Zoom In* à imagem para analisar com mais detalhe os diversos vectores associados à entidade móvel com a identificação R487 (Figura 84).

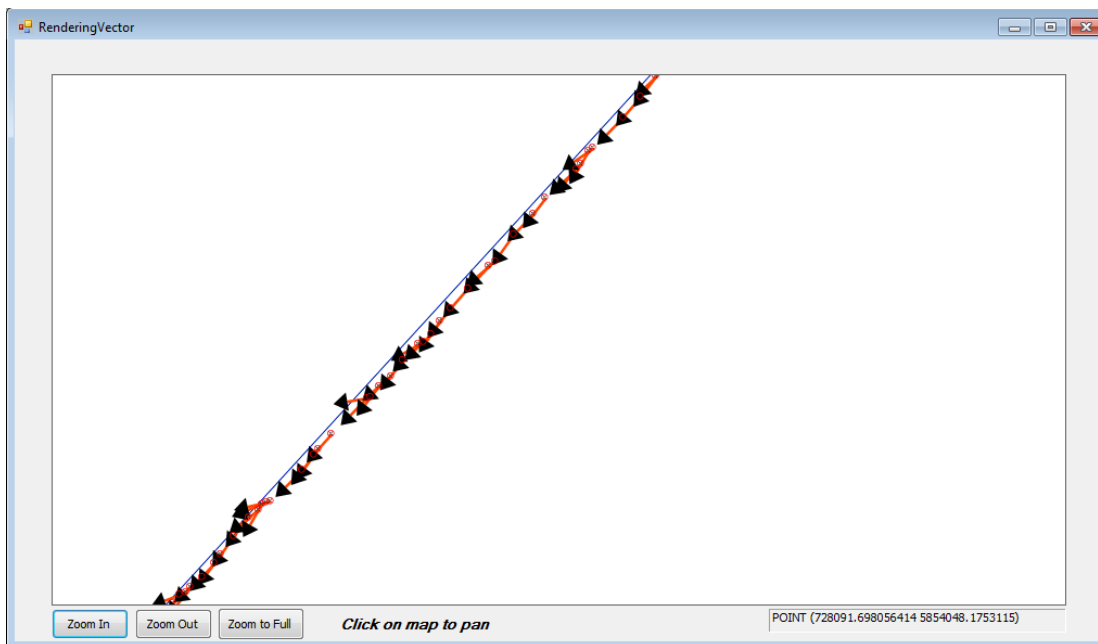


Figura 84 - STAR v. 2.0: *Draw Vector*

Para a representação dos pontos, o mecanismo é muito similar às demais funcionalidades descritas (Representação de trajetórias e vectores).

Assim, após seleccionar as entidades móveis e carregar no botão *Draw Points*, o sistema apresenta todos os pontos de cada entidade. Na Figura 85 estão representados os pontos de três entidades (R658, R261 e R472).

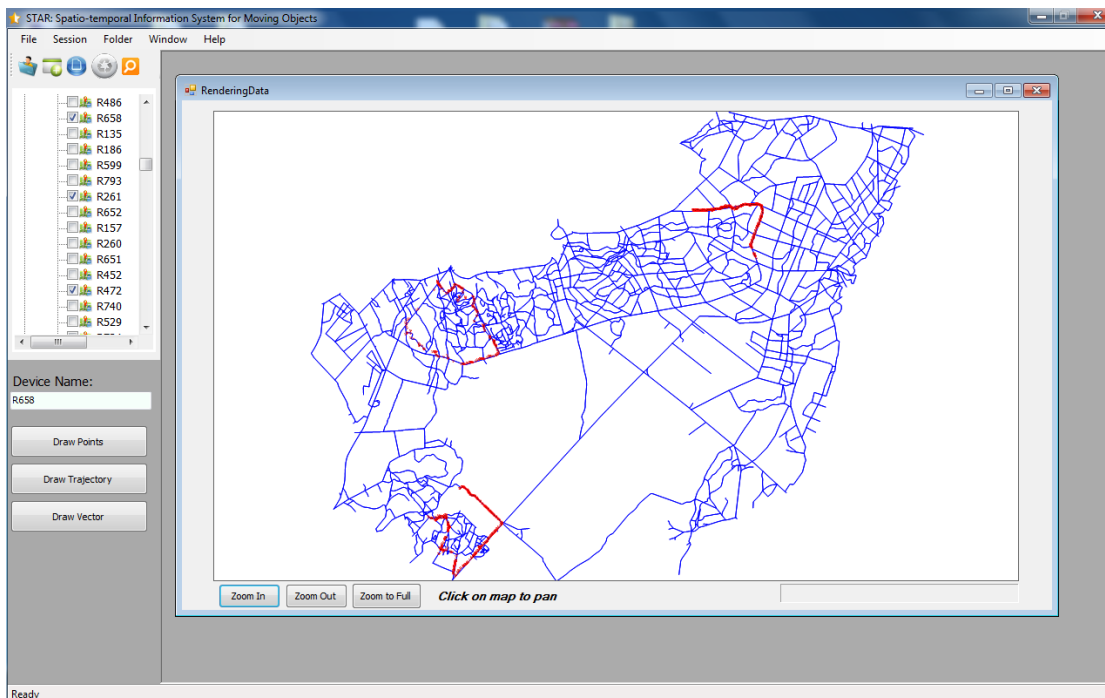


Figura 85 - STAR v. 2.0: *Draw Points*

Para melhor percepção, foram ampliados os pontos da entidade R658 (Anexo 2: Figura 7).

Por último a funcionalidade de visualização dos dados numa tabela apresenta um processo de selecção semelhante às demais funcionalidades. O utilizador ao fim de seleccionar a ou as entidades móveis que pretende visualizar, e pressionar o botão *Show Data (File control)*, é despoletada uma janela com uma tabela com os dados referentes a cada entidade seleccionada. Esta janela permite ao utilizador ordenar as entidades por qualquer das colunas apresentadas (*deviceID, X, Y, dateTime, speed e bearing*), permitindo organizar a apresentação por forma ascendente, descendente ou como estão armazenados (ordenados) no sistema (forma por defeito). Para tal, basta carregar na coluna que se pretende ordenar (e.g. *speed*) sendo reestruturada a apresentação dos dados.

A Figura 86 representa a visualização de dados numa tabela, da entidade móvel R274, R709 e R483, pertencente a um ficheiro previamente importado para o sistema.

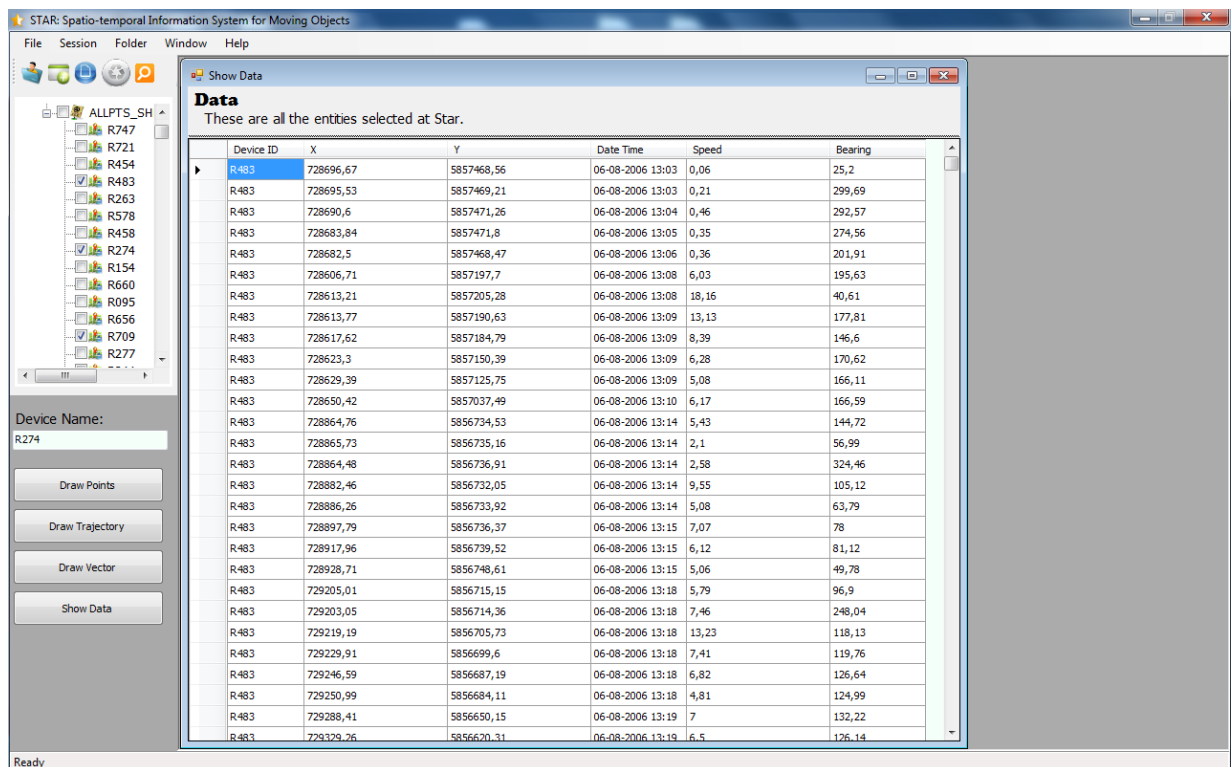


Figura 86 –Show Data: R274, R709 e R483

Na Figura 8 do Anexo 2 estão representados os dados das entidades móveis R274, R709 e R483, ordenados de forma descendente de velocidade (*speed*).

As funcionalidades de análise de dados dividem-se em quatro tipos: criação de histogramas, criação de gráfico de dispersão, representação de dados no mapa e visualizações dados em tabela. Sempre que um utilizador seleccionar um ficheiro na componente *Tree*, a componente

de controlo (*Control*) apresenta quatro botões referentes à análise de dados: *Draw Histogram*, *Scarter Plot*, *Represent Data* e *Show Data*

A representação dos dados do ficheiro (Figura 87) é despoletada através da selecção do botão *Represent Data* contido no componente *Control (FileControl)*.

De uma forma mais detalhada, para realizar uma representação dos dados de um ficheiro o utilizador tem de seleccionar o ficheiro pretendido (Componente *Tree*) e posteriormente seleccionar o botão *Represent Data*, sendo apresentada uma janela na componente *Current Viewer* que contém a representação de todos os dispositivos do ficheiro.

Esta representação disponibiliza um conjunto de botões (*Zoom In*, *Zoom Out*, *Zoom to Full*) e informações dos valores das coordenadas (canto inferior direito) tanto do mapa como dos dados. O utilizador pode assim ampliar e reduzir a imagem representada, podendo a qualquer momento retornar à representação inicial (botão *Zoom to Full*). A deslocação para outras zonas do mapa é possível através do cursor, para tal basta arrastar o mapa para qualquer uma das direcções pretendidas. No canto inferior direito são apresentados o valor das coordenadas de um ponto no mapa. Assim o utilizador pode identificar qualquer ponto no mapa apenas colocando o cursor do rato sobre um ponto específico.

A Figura 88 ilustra a representação dos dados descritas anteriormente, sendo realizado um *zoom* para melhor percepção.

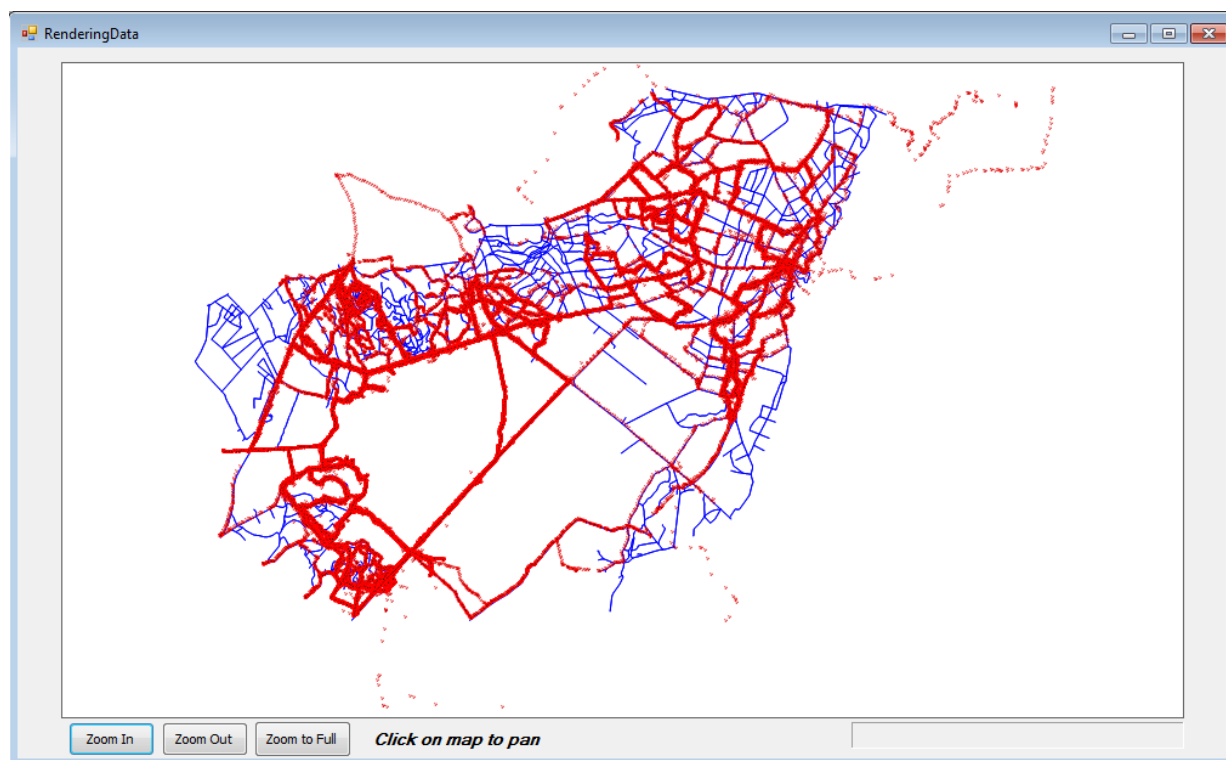
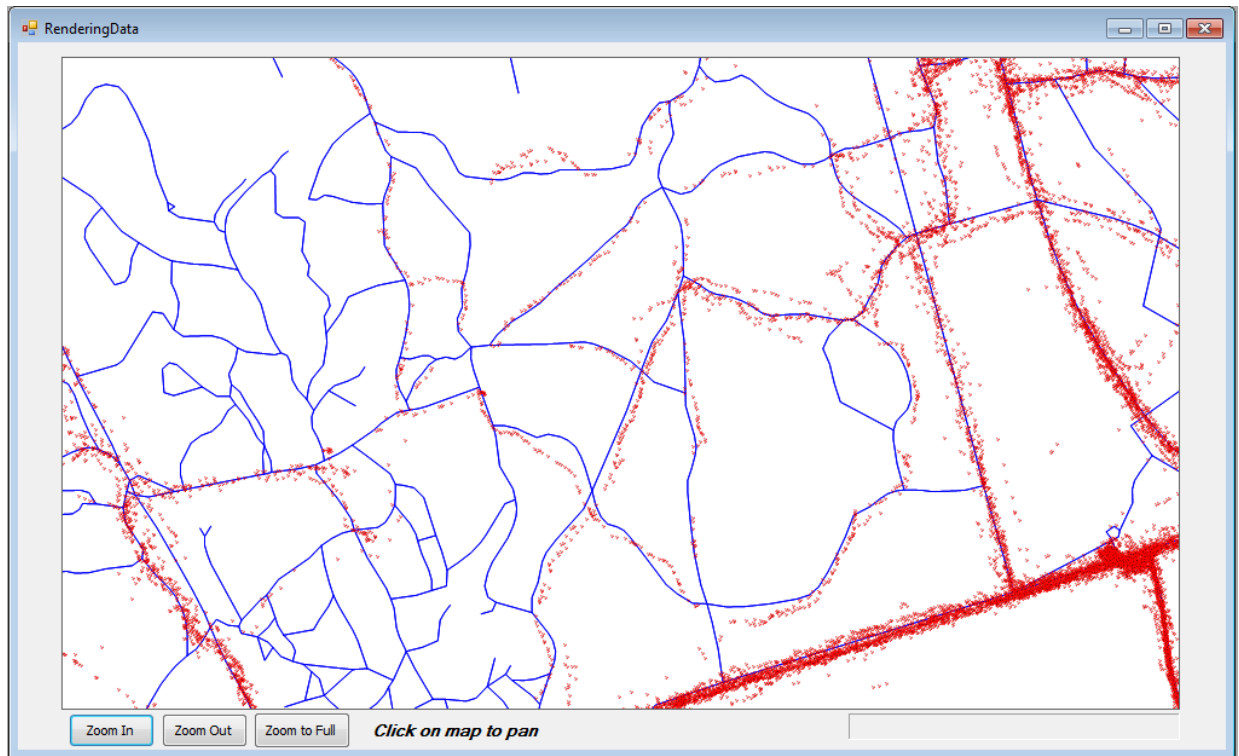


Figura 87 - STAR v. 2.0: *Represent Data*



**Figura 88 - STAR v. 2.0: Represent Data**

Esta funcionalidade pode representar os dados de um ficheiro em tela branca, não apresentando o respectivo mapa. Se qualquer utilizador importar um ficheiro não seleccionando a directoria de um mapa (*shapefile*), o sistema realiza todas as representações e visualizações de dados sem recurso ao mesmo.

Como referido, a análise de dados apresenta duas funcionalidades referentes à criação de gráficos. O sistema permite a criação de gráficos de dispersão e histogramas. Para criar um gráfico de dispersão, seleccionamos um ficheiro da componente *Tree* e de seguida seleccionamos o botão *Scarter Plot (File control)*. É aberta uma janela na componente *Current Viewer*, onde teremos de indicar (escolher na lista) as variáveis que irão constar no eixo do XX e do YY. Depois de seleccionar o botão *Draw Graph* o sistema apresenta o gráfico na mesma janela (Figura 89).

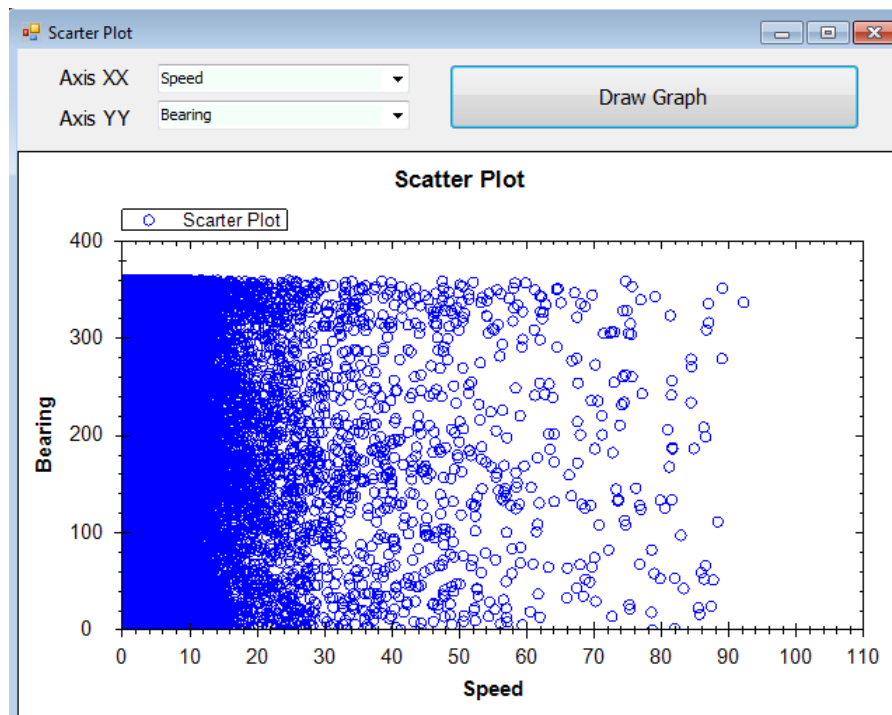


Figura 89 - STAR v. 2.0: *Scarter Plot*

O gráfico criado pelo sistema permite realizar um conjunto de funções sob os dados bastante úteis para os utilizadores do sistema. O utilizador ao carregar com o botão do lado direito do rato sobre o mapa é apresentado um menu com um conjunto de opções: copiar, guardar como, imprimir, informação da página, ampliar, reduzir, mostrar pontos, configuração página, mostrar valores dos pontos (Anexo 2: Figura 9). Todas as opções deste menu acrescentam um contributo bastante importante para o utilizador. Este pode imprimir e guardar o gráfico em vários formatos, pode copiar o gráfico para outra ferramenta, fazer *zoom* para pontos específicos, podendo visualizar o valor dos mesmos, entre outras opções.

A criação de um histograma é bastante idêntica ao processo descrito para a criação do gráfico de dispersão. Para tal basta seleccionarmos um ficheiro (da *Tree*) e seleccionar o botão *Draw Histogram*. É aberta uma janela onde temos de indicar duas informações: o campo que queremos visualizar e o número de intervalos do histograma (Figura 86). Ao seleccionar o botão *Draw Graph*, o sistema apresenta um histograma com os valores do atributo seleccionado (Figura 90).

Ao gráfico representado é possível aplicar o mesmo conjunto de operações já descritas no gráfico de dispersão (Anexo 2: Figura 10).

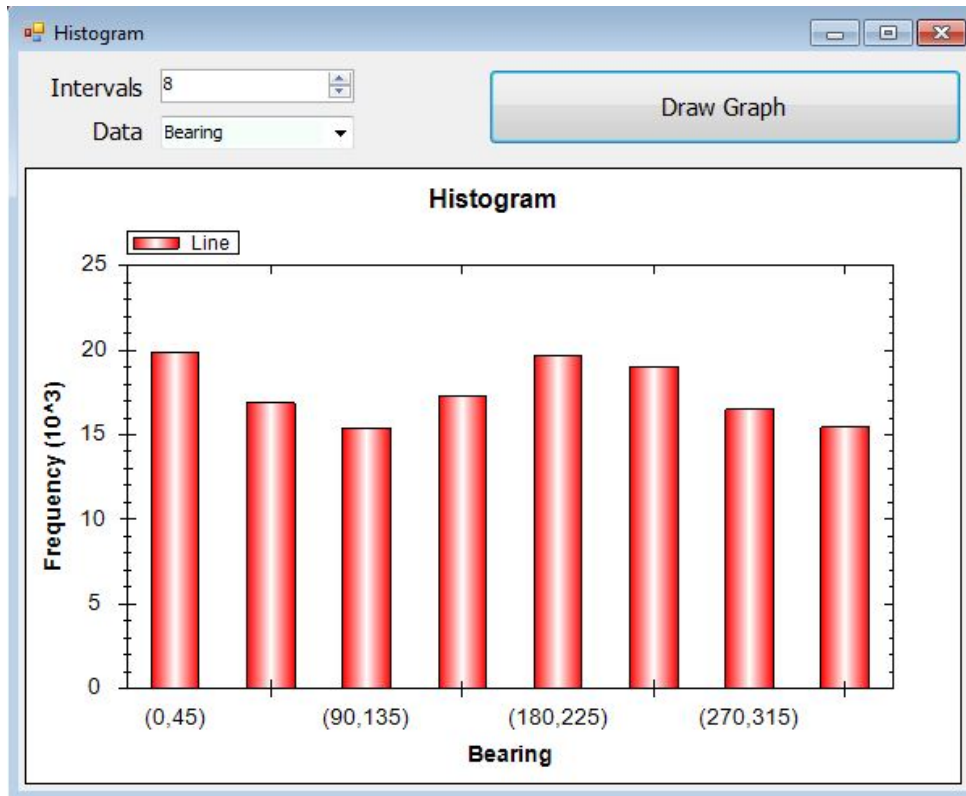


Figura 90 - STAR v. 2.0: *Histogram*

Por último a funcionalidade de visualização de dados numa tabela. Ao fim de seleccionar um ficheiro, basta seleccionar o botão *Show Data (File control)*, para visualizar os dados. É apresentada uma janela na componente *Current Viewer*. Esta janela apresenta todos os dados relativos ao ficheiro seleccionado, sobre a forma de tabela (Figura 91).

Os dados apresentados na tabela, correspondem aos dados que foram importados para o sistema (*import file*) e armazenados na Tabela *data* da BD do sistema. Como já referido na visualização de dados numa tabela (funcionalidade visualização de dados), esta permite a ordenação da apresentação dos mesmos.

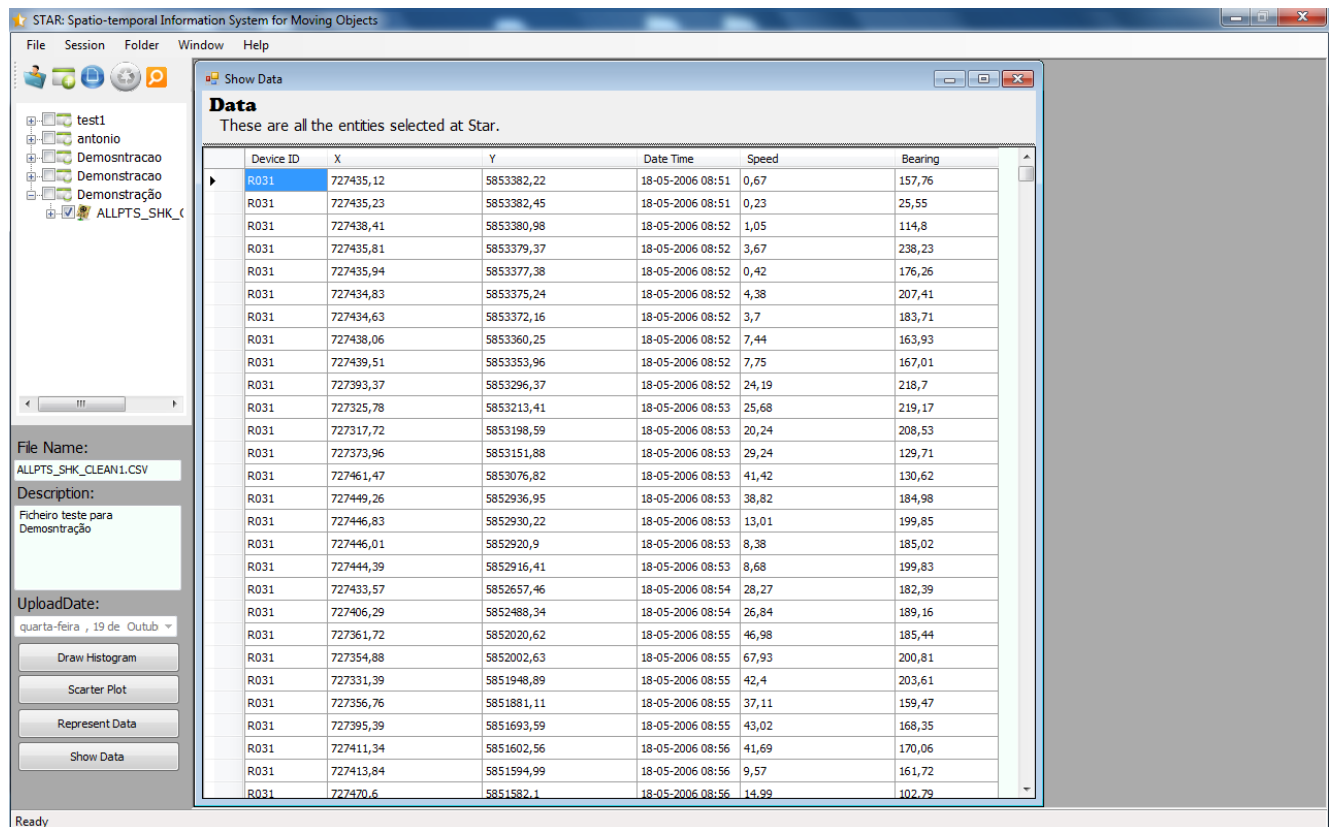


Figura 91 - STAR v. 2.0: Show Data

Na Figura 11 do Anexo 2 estão representados os dados do mesmo ficheiro, ordenados de forma ascendente de velocidade (*speed*).

Por fim, a Figura 92 ilustra a correcção de dados (*Data Cleaning*). Esta funcionalidade efectua a correcção dos dados de um ficheiro. O sistema apresenta correcções a quatro tipos de atributos: Velocidade (*Speed*), Ângulo (*Bearing*), coordenada X (CoordX) e coordenada Y (CoordY). O utilizador terá sempre de especificar (preencher) os valores que pretende eliminar, podendo portanto realizar qualquer tipo de correcção.

Para efectivar uma correcção aos dados, o utilizador selecciona qual o ficheiro (*File Name: List*) que pretende corrigir, e quais os atributos e condições que devem ser verificadas.



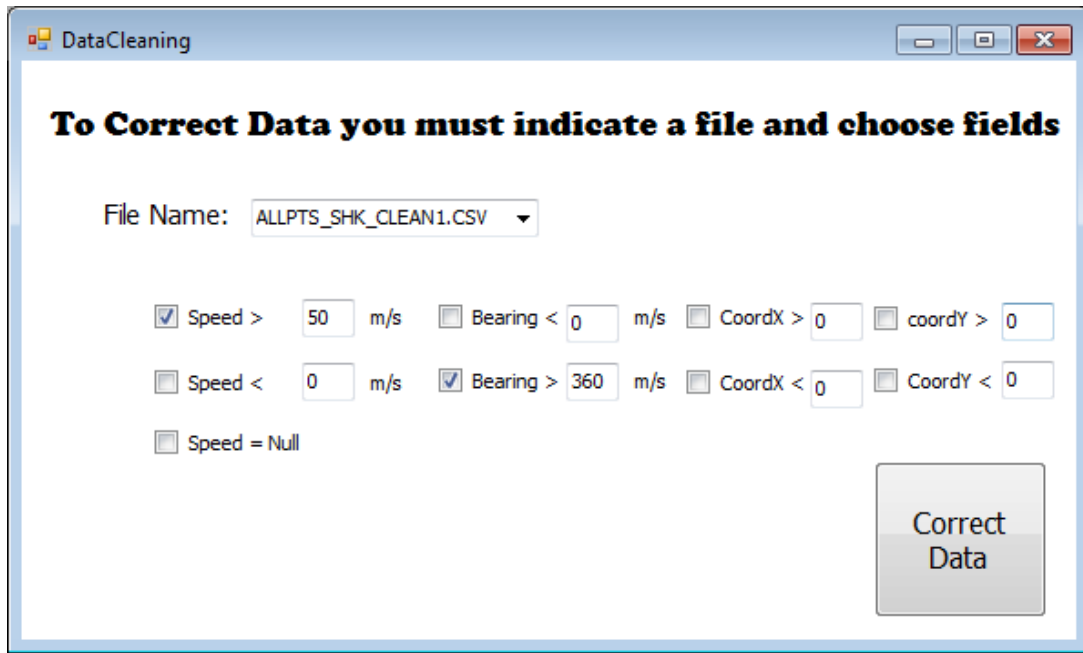


Figura 92 - STAR v. 2.0: *Data Cleaning*

Na Figura 92 está representada a correcção do ficheiro *ALLPTS\_SHK\_CLEAN1.CSV*, sendo os valores maiores que cinquenta (metros por segundo) ou com um ângulo superior a trezentos e sessenta (graus) eliminados do sistema.

O utilizador pode realizar correcções de dados a todos os ficheiros a que tem acesso no sistema, tanto na componente *Tree* como na listagem dos mesmos.

### 5.3. Caracterização das Análises e Visualização dos Dados

Nesta secção pretende-se demonstrar o potencial do STAR através da análise e visualização de dados importados para o sistema. Serão analisados os dados de dois ficheiros distintos relativos ao movimento de pedestres.

Esta secção inicia-se com a descrição dos conjuntos de dados usados para demonstrar o funcionamento do protótipo implementado. Seguindo-se a demonstração das funcionalidades acima referidas.

#### 5.3.1. Descrição dos conjuntos de dados

A análise de um conjunto de dados como demonstração do protótipo implementado é um dos objectivos deste projecto. Como parte deste projecto, usaram-se dois conjuntos de dados sobre o movimento de pedestres ao longo do espaço e do tempo. Estes dados foram recolhidos na

Holanda, por duas aplicações diferentes que recorriam ao GPS para obter a posição de cada pedestre.

De acordo com Moreira et al. (2010), o primeiro conjunto de dados foi recolhido na cidade de Amsterdão, durante um jogo denominado de Caça ao Tesouro. Estes dados foram recolhidos durante 10 dias no mês de Julho de 2007, totalizando 55,577 registos de entre 419 jogadores, com idades compreendidas entre os doze e os catorze anos. O segundo conjunto centra-se no movimento de pedestres que visitaram o parque *Dwingelderveld National Park*, também na Holanda. Os dados foram recolhidos num período de sete dias, entre o mês de Maio e Agosto de 2006, sendo recolhidos mais de 140.000 registos, para 370 visitantes.

Os ficheiros apresentam estruturas diferentes sendo que o primeiro conjunto é constituído por catorze atributos distintos, entre eles: *device\_id*, *dateTime*, *point\_id*, *coorX* e *coorY*, *originX* e *originY*, *bearing*, *speed*, *distance*, *seconds*, *pre\_point*, *sequence* e *flag*. O segundo conjunto apresenta um conjunto mais alargado de atributos (24): *team*, *zoneName*, *label*, *latitude*, *longitude*, *shkX*, *shkY*, *shkDistance*, *shkSpeed* e *shkBearing*. Neste ficheiro foi realizado o *shaking*<sup>14</sup> dos dados.

### 5.3.2. Análise e Visualizações de Dados

Nesta secção pretende-se demonstrar o potencial do STAR através da análise e visualização de dados importados para o sistema. Serão analisados os dados de dois ficheiros distintos relativos ao movimento de pedestres.

Ao longo deste trabalho as funcionalidades representar dados, criar gráfico de dispersão, histogramas e visualizar dados em tabela são agrupadas e descritas como análise de dados, já as funcionalidades desenhar pontos, desenhar trajectória, desenhar vectores e visualizar dados tabela são agrupadas e descritas como visualização de dados. A grande diferença entre os dois grupos é que as análises de dados operam sobre a totalidade dos dados de um ficheiro no sistema, enquanto que as visualizações realizam representações sobre um conjunto de

---

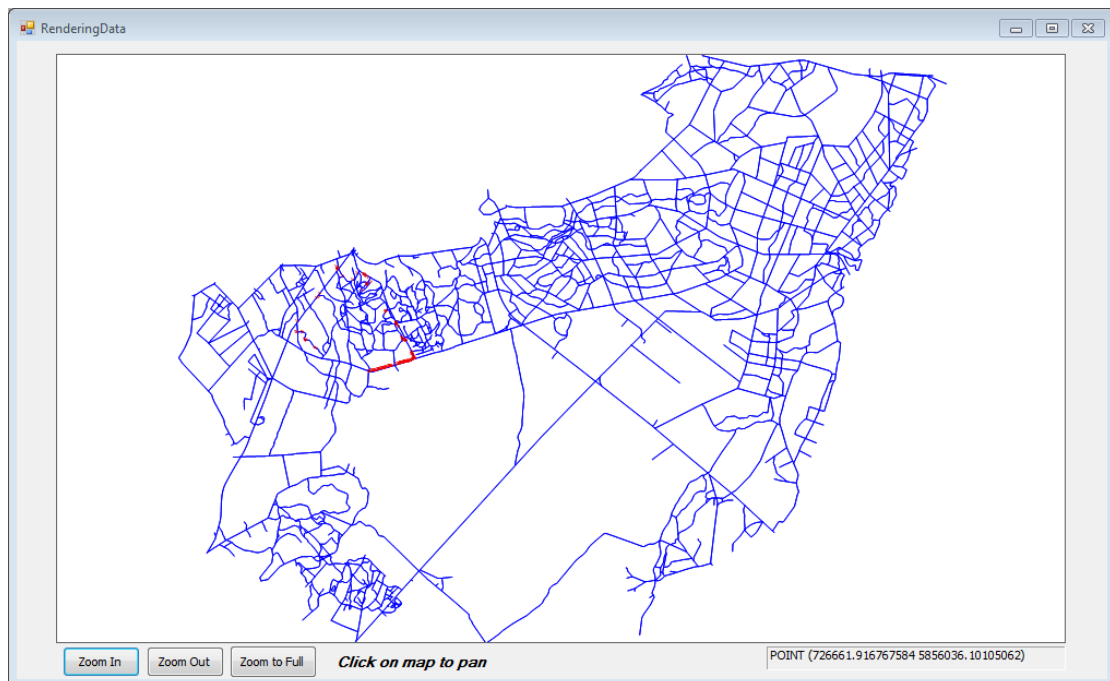
<sup>14</sup> O *shaking* consiste na adição de uma pequena quantidade de ruído Gaussiano às posições X e Y. Para mais informação sobre o *shaking* de dados consultar Moreira et al. (2010).

entidades móveis (dados) de um ficheiro importado para o sistema. A selecção das entidades é feita pelo utilizador, podendo conter uma única entidade ou a totalidade das entidades do ficheiro.

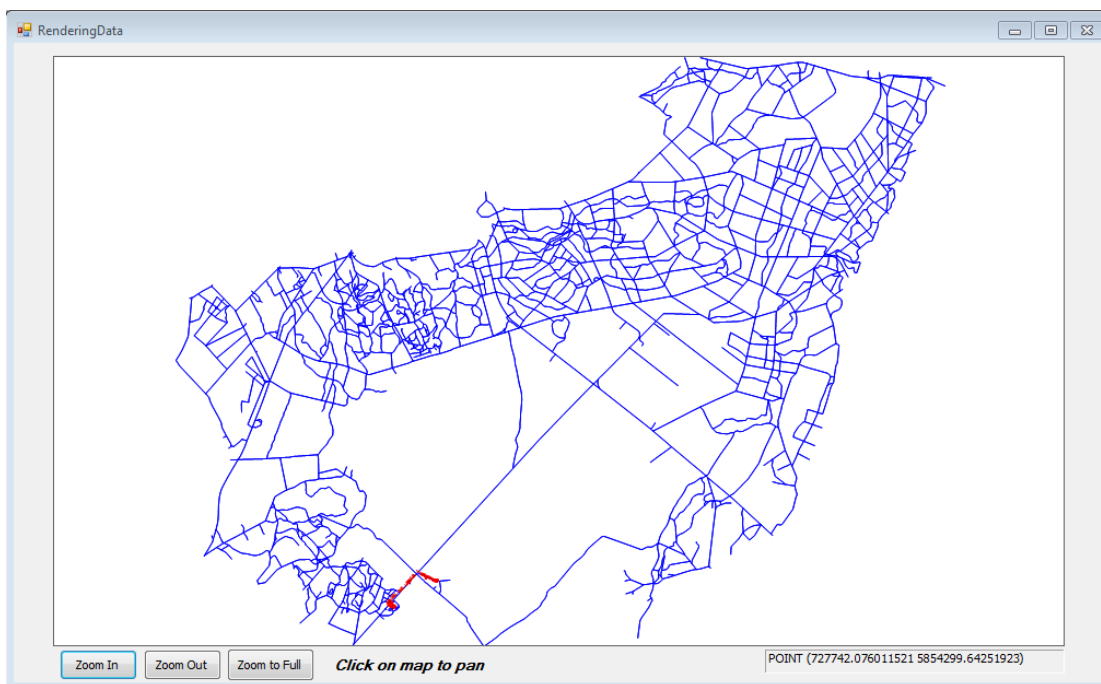
Neste trabalho pretende-se demonstrar o potencial do STAR através da análise e visualização dos dados importados para o sistema. Assim sendo, serão analisados os dados do primeiro e do segundo ficheiro, sendo ambos denominados pelo local onde foram captados: *Amsterdam* e *Dwingelderveld National Park*. No primeiro ficheiro serão realizadas as funcionalidades agrupadas no caso de uso de alto nível Visualização de Dados. Ao segundo ficheiro serão realizadas as funcionalidades agregadas na Análise de Dados.

### **Dwingelderveld National Park**

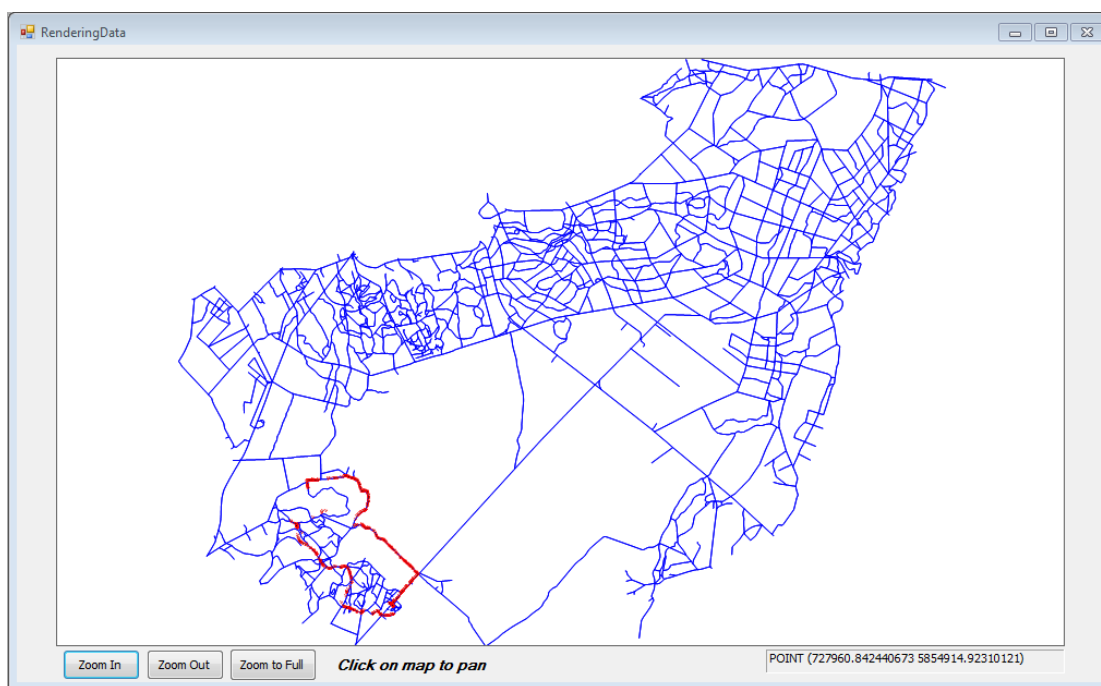
A visualização dos dados deste ficheiro iniciou-se com a selecção dos dados referentes a três entidades móveis, neste caso pedestres. Estes são identificados pelo seu ID, sendo que neste caso seleccionaram-se os pedestres R263, R660 e R656 (Figura 93, Figura 94 e Figura 95). Esta selecção permite visualizar o percurso de cada pedestre e restringe os dados para as operações que se apresentam de seguida.



**Figura 93 – Desenho dos pontos do pedestre R263**

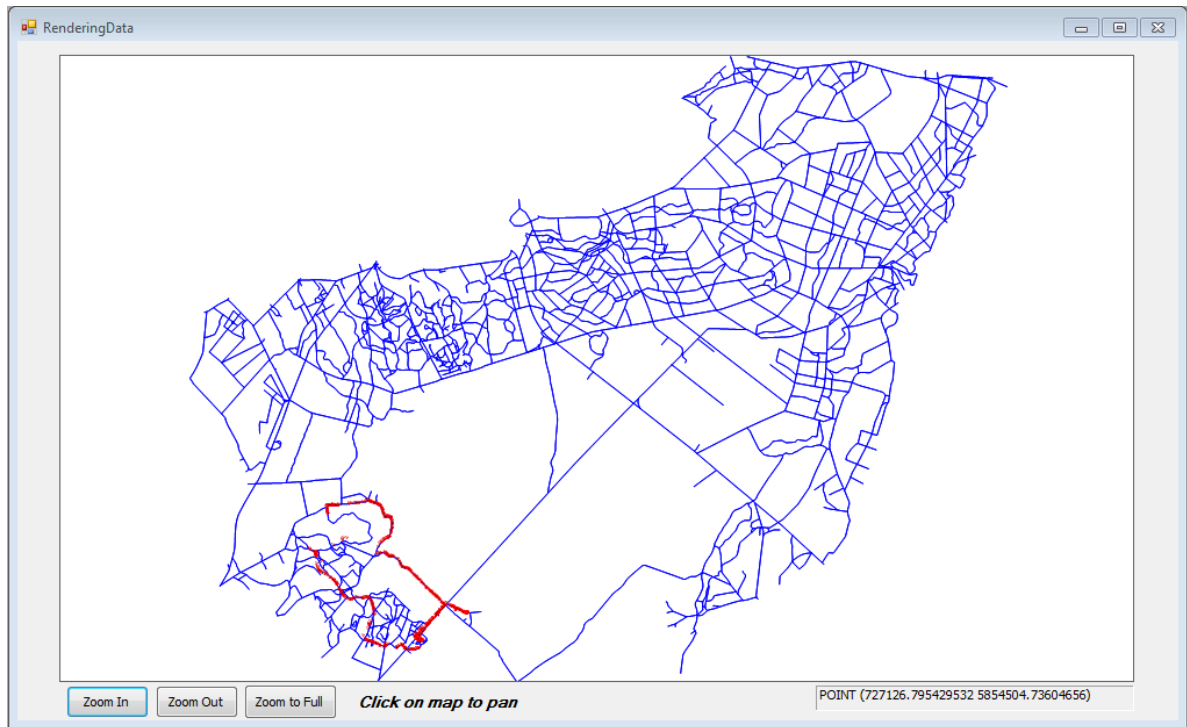


**Figura 94 – Desenho dos pontos do pedestre R660**



**Figura 95 – Desenho dos pontos do pedestre R656**

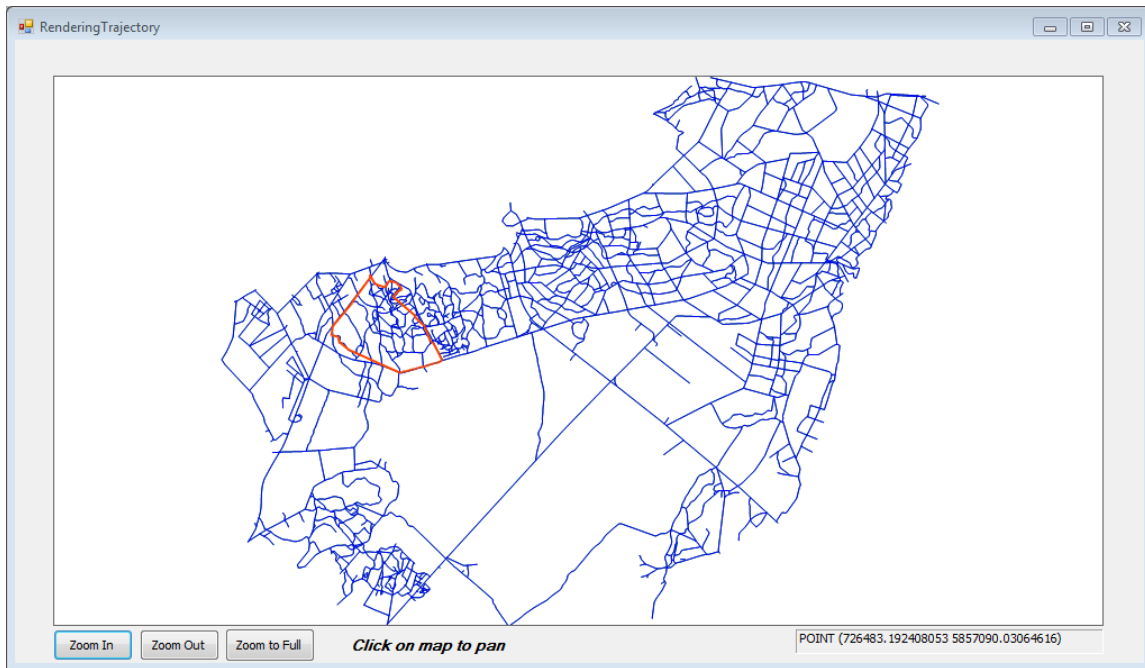
Conforme se pode verificar pela análise da Figura 93, Figura 94 e Figura 95, os pedestres estão situados em diferentes zonas do mapa. Analisando as figuras referidas, podemos afirmar que o pedestre R263 não esteve em contacto com os outros dois pedestres, uma vez que não se verifica nenhuma sobreposição de pontos. No entanto, os dois últimos pedestres seguiram uma rota semelhante, conforme se pode ver pela Figura 96, através da sobreposição de pontos.



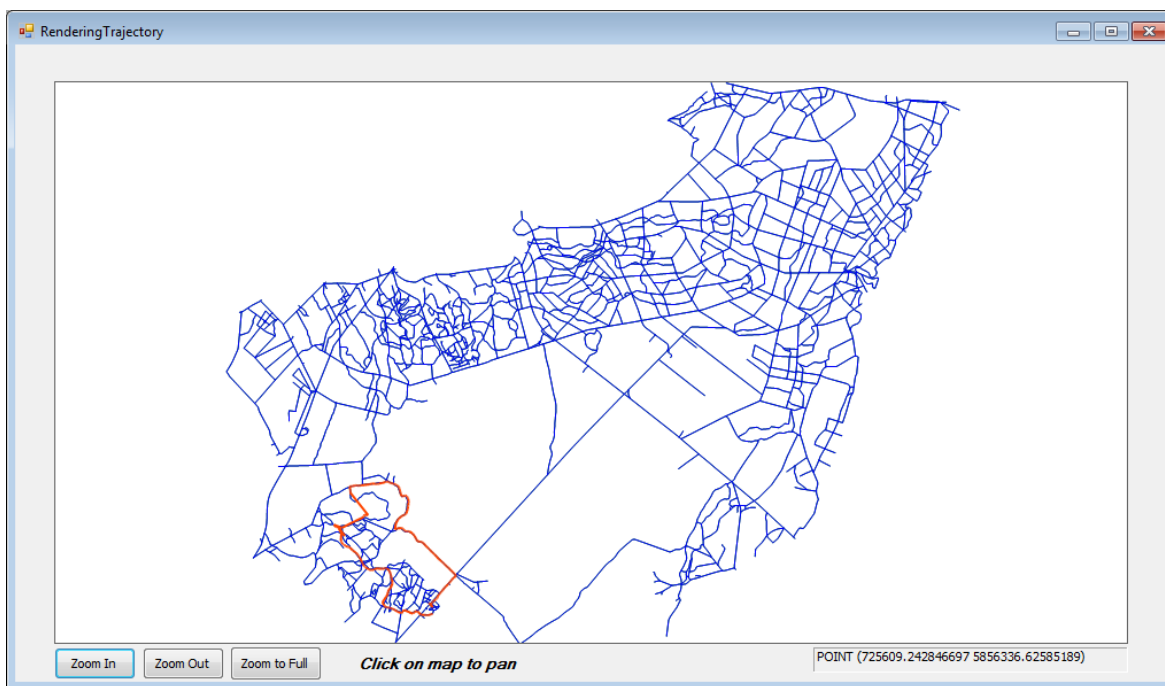
**Figura 96 - Desenho dos pontos dos pedestres R660 e R656**

As visualizações dos dados em tabela, relativos a cada um dos pedestres seleccionados estão ilustradas no Anexo 3, podendo estas representar uma ajuda na análise de cada visualização realizada.

Para analisar a trajectória de cada um dos pedestres, utilizou-se a funcionalidade do sistema *desenhar trajectória*. Como demonstrado na Figura 97 e 98, os pedestres R263 e R656 voltaram ao ponto de origem. Isto significa que o ponto de partida e chegada são iguais, sendo de uma forma geral, estes estão associados a sítios onde há parques de estacionamento.

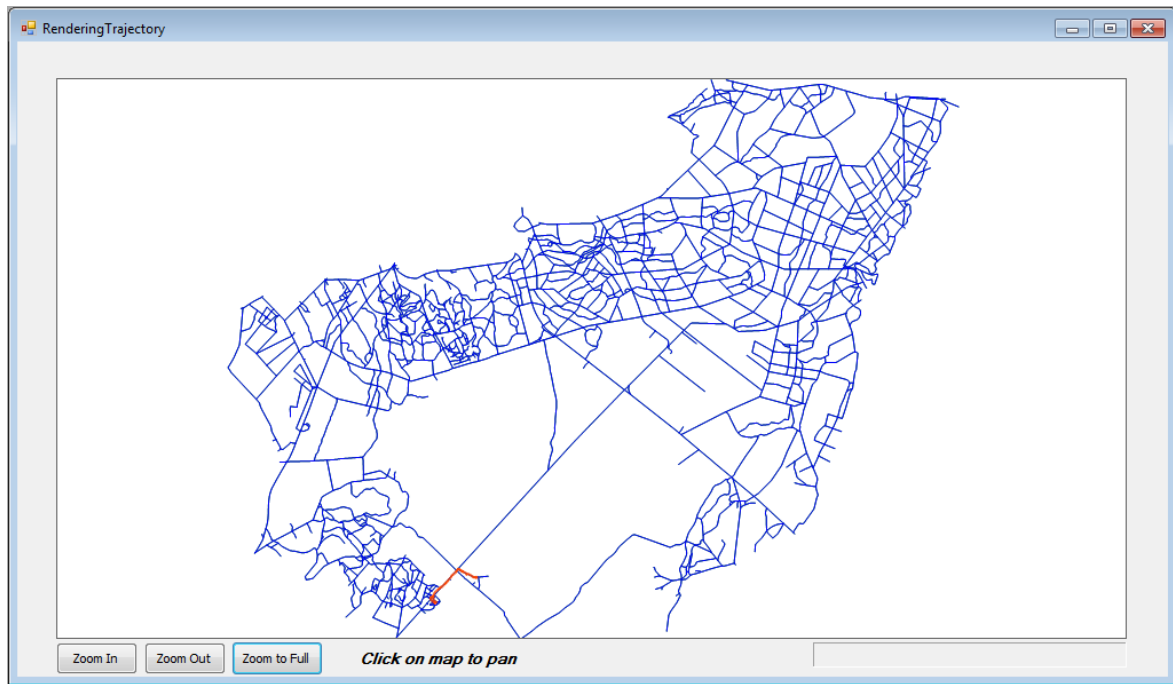


**Figura 97 –Trajectória do pedestre R263**



**Figura 98 - Trajectória do pedestre R656**

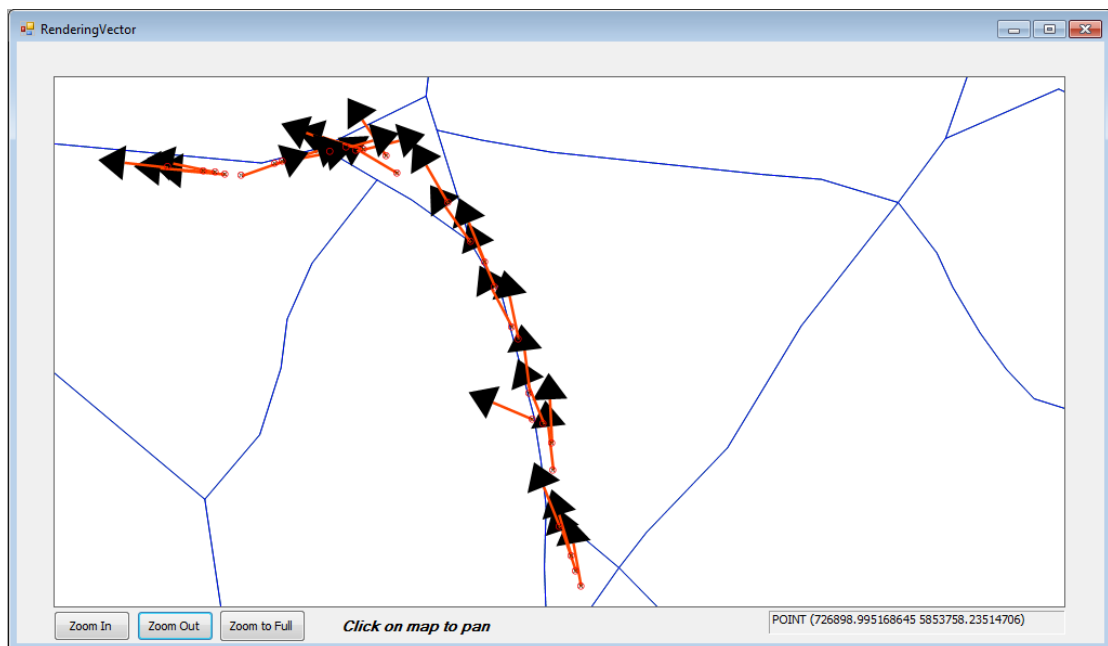
Já a trajectória do pedestre R660 (Figura 99), apresenta uma forma não fechada, o que indica que este não voltou ao ponto inicial.



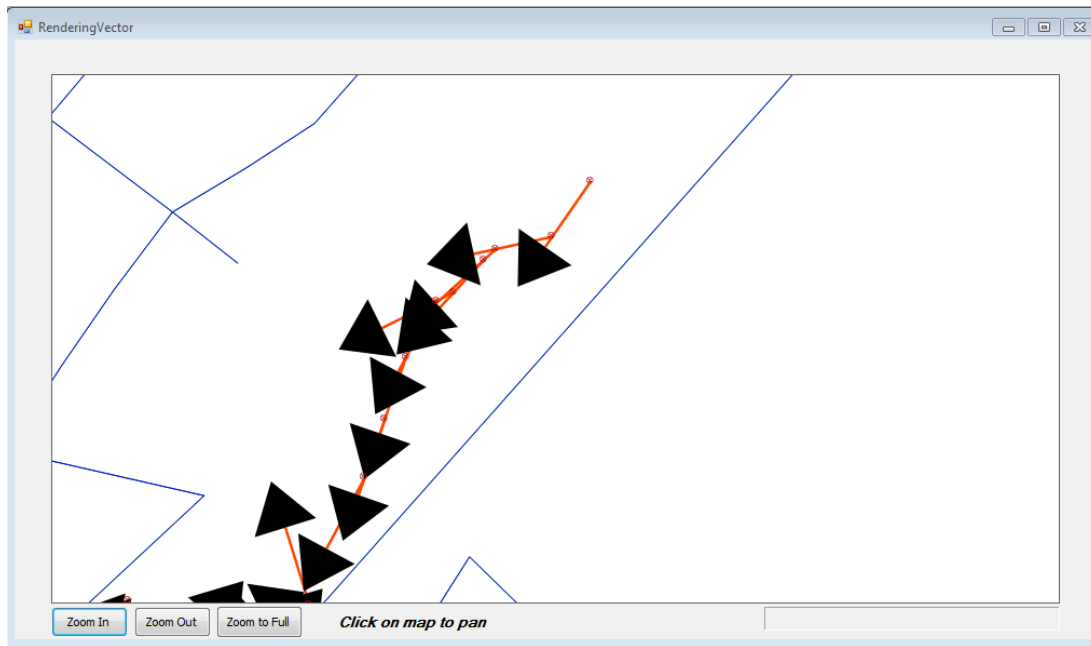
**Figura 99 - Trajectória do pedestre R660**

A análise do percurso dos três pedestres termina com a apresentação dos vectores direccionais.

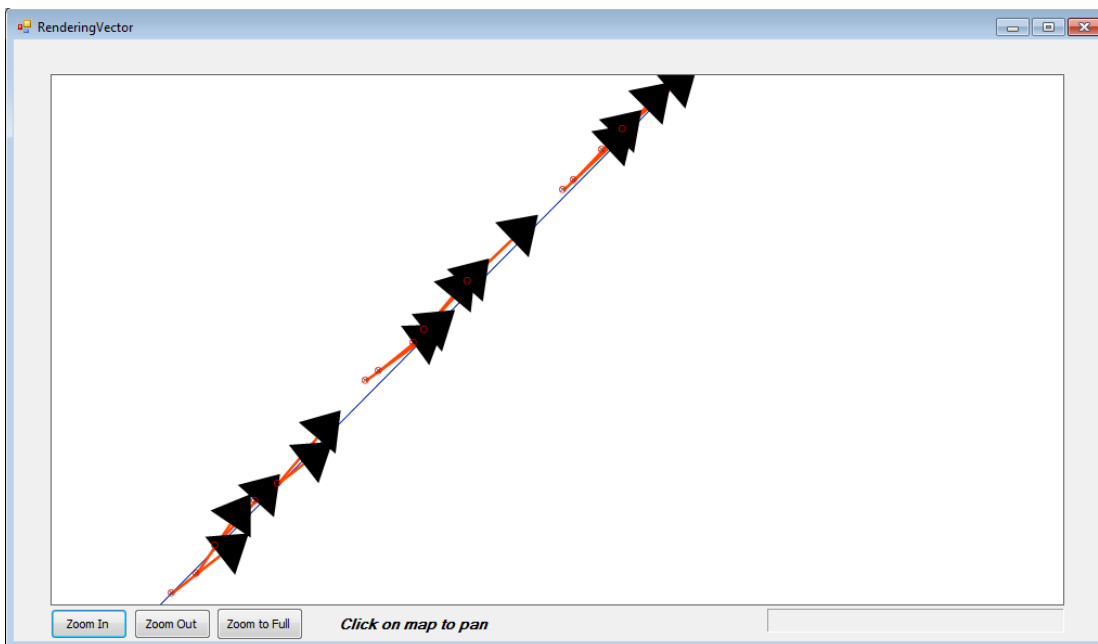
As Figuras 100, 101 e 102 permitem visualizar qual a direcção seguida pelos pedestres durante todo o percurso. Para uma melhor percepção sobre a representação dos mesmos, ampliou-se o percurso de cada pedestre.



**Figura 100 –Vectores do pedestre R656**



**Figura 101 - Vectors do pedestre R660**



**Figura 102 - Vectors do pedestre R263**

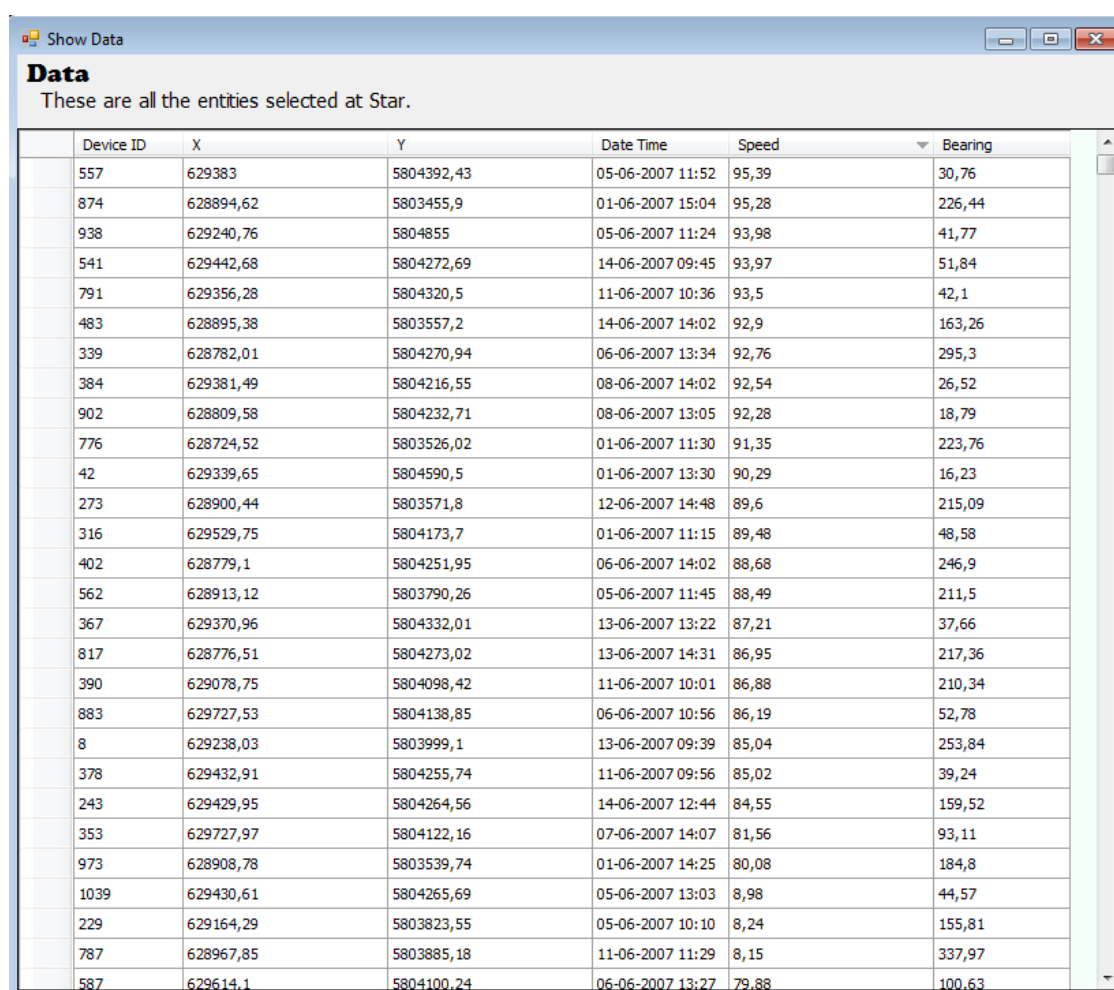
Através da visualização das figuras, verifica-se que estão apenas representados determinados locais do percurso, isto porque os pontos estão muito próximos e em vários locais quase sobrepostos, e por conseguinte, existe uma sobreposição de vectores, não conferindo a melhor percepção e representação dos vectores direccionais. Este problema pode ser corrigido através do *zoom in* e do *pan* para uma determinada área como está realizado nas imagens apresentadas.



## Amsterdam

A análise de dados deste ficheiro iniciou-se com a selecção do ficheiro (*Amsterdam*) para realização de análises. O ficheiro previamente importado para o sistema totaliza 55,577 registos de entre 419 pedestres (Figura 103).

O desenho do histograma e do gráfico de dispersão encontra-se representado na Figura 104. Apesar dos resultados obtidos estarem correctos, de acordo com o que foi representado pelo sistema, os valores da velocidade máxima não coincidem com a realidade, pois existem valores de velocidade bastante elevadas, como se pode verificar tanto no histograma como no gráfico de dispersão. De acordo com Moreira et al. (2010), a velocidade máxima de um pedestre situa-se entre os 5-6 m/s. Como resultado, pode afirmar-se que os pontos analisados foram afectados por erros de captação do posicionamento. Estes erros de leitura (*outliers*) podem ser tratados pelo STAR, através da funcionalidade de correcção de dados.



Device ID	X	Y	Date Time	Speed	Bearing
557	629383	5804392,43	05-06-2007 11:52	95,39	30,76
874	628894,62	5803455,9	01-06-2007 15:04	95,28	226,44
938	629240,76	5804855	05-06-2007 11:24	93,98	41,77
541	629442,68	5804272,69	14-06-2007 09:45	93,97	51,84
791	629356,28	5804320,5	11-06-2007 10:36	93,5	42,1
483	628895,38	5803557,2	14-06-2007 14:02	92,9	163,26
339	628782,01	5804270,94	06-06-2007 13:34	92,76	295,3
384	629381,49	5804216,55	08-06-2007 14:02	92,54	26,52
902	628809,58	5804232,71	08-06-2007 13:05	92,28	18,79
776	628724,52	5803526,02	01-06-2007 11:30	91,35	223,76
42	629339,65	5804590,5	01-06-2007 13:30	90,29	16,23
273	628900,44	5803571,8	12-06-2007 14:48	89,6	215,09
316	629529,75	5804173,7	01-06-2007 11:15	89,48	48,58
402	628779,1	5804251,95	06-06-2007 14:02	88,68	246,9
562	628913,12	5803790,26	05-06-2007 11:45	88,49	211,5
367	629370,96	5804332,01	13-06-2007 13:22	87,21	37,66
817	628776,51	5804273,02	13-06-2007 14:31	86,95	217,36
390	629078,75	5804098,42	11-06-2007 10:01	86,88	210,34
883	629727,53	5804138,85	06-06-2007 10:56	86,19	52,78
8	629238,03	5803999,1	13-06-2007 09:39	85,04	253,84
378	629432,91	5804255,74	11-06-2007 09:56	85,02	39,24
243	629429,95	5804264,56	14-06-2007 12:44	84,55	159,52
353	629727,97	5804122,16	07-06-2007 14:07	81,56	93,11
973	628908,78	5803539,74	01-06-2007 14:25	80,08	184,8
1039	629430,61	5804265,69	05-06-2007 13:03	8,98	44,57
229	629164,29	5803823,55	05-06-2007 10:10	8,24	155,81
787	628967,85	5803885,18	11-06-2007 11:29	8,15	337,97
587	629514,1	5804100,24	06-06-2007 13:27	79,88	100,63

Figura 103 - Tabela Data: conjunto de dados Amsterdam

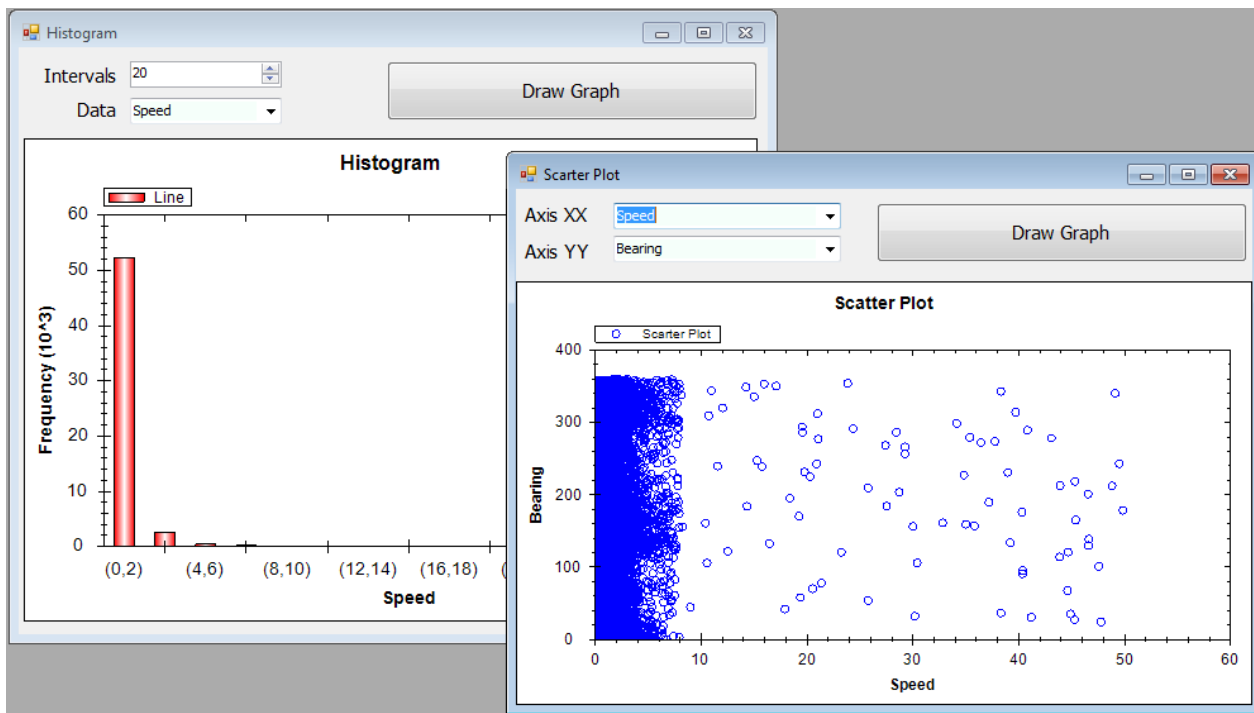


Figura 104 - Desenho de Histograma e do Gráfico de Dispersão: *Amsterdam*

A Figura 105 representa a formulário de correcção de dados. Todas as linhas com velocidades superiores a 5 m/s serão eliminadas da base de dados espaço-temporal do sistema.

The 'DataCleaning' window has a title bar with standard window controls. The main content area has a bold heading: "To Correct Data you must indicate a file and choose fields". Below this, there is a 'File Name:' label followed by a dropdown menu showing 'Shaked\_pts.txt'. There are several rows of checkboxes and input fields:
 

- Row 1:  Speed > [5] m/s,  Bearing < [0] m/s,  CoordX > [0],  coordY > [0]
- Row 2:  Speed < [0] m/s,  Bearing > [360] m/s,  CoordX < [0],  CoordY < [0]
- Row 3:  Speed = Null

 A large 'Correct Data' button is located at the bottom right of the window.

Figura 105 - Correcção de Dados: *Amsterdam*

Esta funcionalidade aumenta a eficiência nas análises e visualizações dos dados, pois elimina dados de acordo com as condições impostas pelo utilizador. Esta é bastante funcional pois não

restringe para um tipo específico de dados. A imagem 106, 107 e 108 ilustram o histograma, o gráfico de dispersão e a visualização dos dados em tabela criados (ordenada por ordem decendente de *speed*) pelo sistema ao fim de corrigido o ficheiro *Amsterdam*.

Foram eliminadas 393 linhas de dados com erros de posicionamento, verificando-se assim a utilidade desta funcionalidade. Através da análise do gráfico de dispersão, do histograma e da visualização de dados em tabela, pode-se afirmar que todos os pedestres que apresentavam velocidades superiores a 5 m/s foram eliminados. Pode-se também verificar que a maioria (mais de 50,000 de um total de 55,000) dos pedestres desloca-se a uma velocidade entre os 0 e o 2 m/s.

Todo o processo de análise efectuada ao ficheiro *Amsterdam* foi também realizado ao ficheiro *Dwingelderveld National Park*, estando as Figuras ilustradas no Anexo 3. Estas cumprem a mesma ordem de apresentação e descrição seguida nesta secção, sendo importante referir que a correcção de dados neste ficheiro torna-se fundamental, pois existe uma grande quantidade de dados com erros de leitura (cerca de 35000 dados).

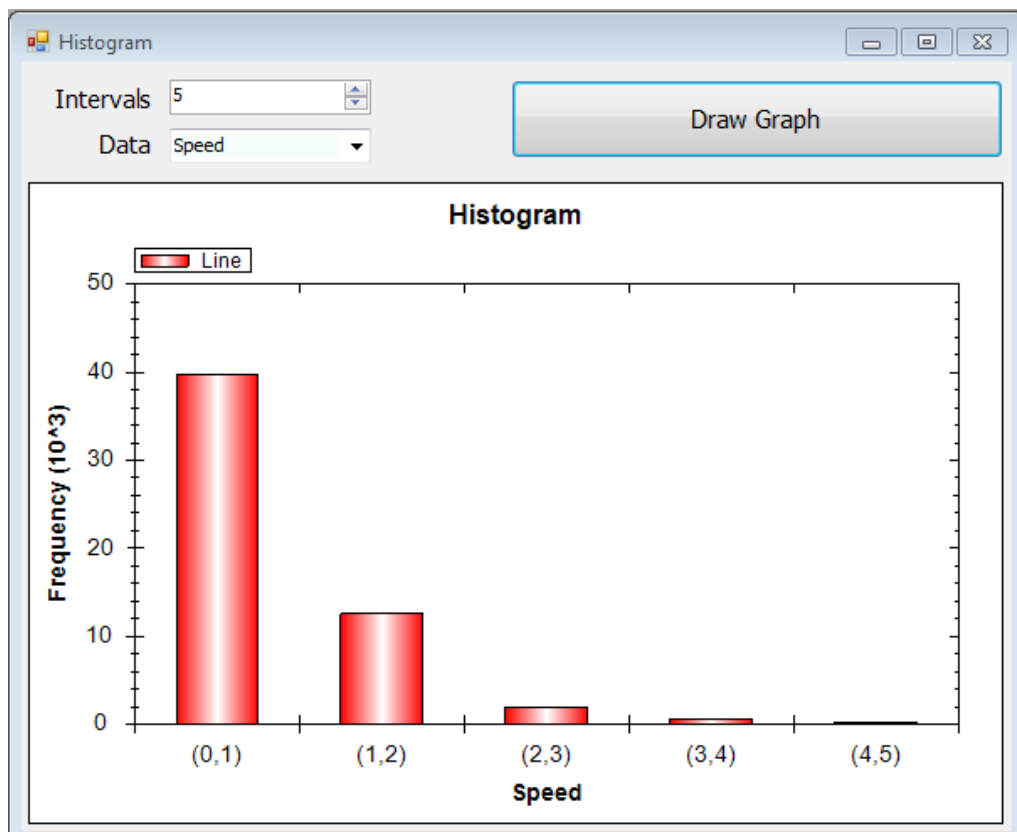


Figura 106 – Histograma: *Amsterdam*

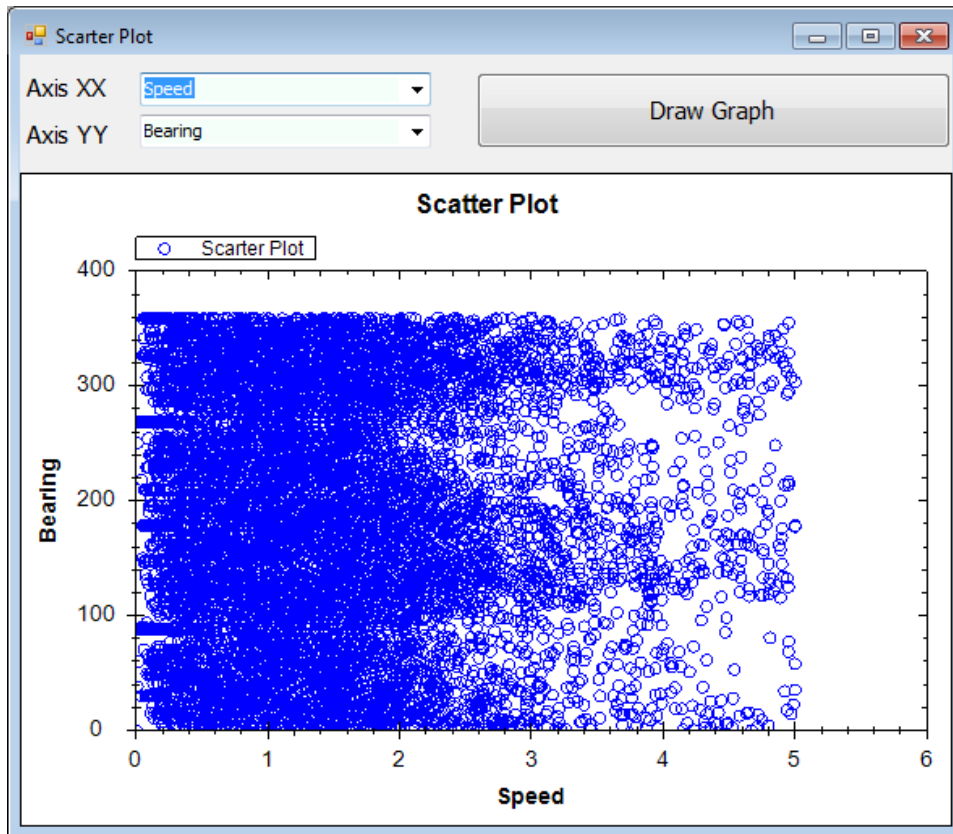


Figura 107 – Gráfico de dispersão: *Amsterdam*

**Data**  
These are all the entities selected at Star.

Device ID	X	Y	Date Time	Speed	Bearing
895	629571,81	5804102,47	05-06-2007 09:45	5	35,28
123	628658,18	5804135,27	13-06-2007 14:47	5	177,53
541	629728,8	5804142,22	14-06-2007 10:12	5	303,79
622	629062,93	5803922,17	15-06-2007 13:10	5	58,06
579	628886,87	5804142,36	05-06-2007 14:23	5	178,49
564	629013,9	5804389,42	13-06-2007 10:12	4,99	22,61
305	629506,81	5804553,72	13-06-2007 12:03	4,97	14,38
328	629009,75	5804443,84	11-06-2007 14:09	4,97	307,24
701	629639,89	5804129,86	08-06-2007 10:33	4,95	124,89
367	628982,01	5804410,84	13-06-2007 14:04	4,95	216,7
598	629273,17	5804035,64	05-06-2007 09:48	4,95	295,2
900	629080,26	5804482,41	07-06-2007 10:38	4,95	68,58
543	628715,7	5803967,61	15-06-2007 11:19	4,95	355
791	629367,58	5804564,52	11-06-2007 12:03	4,95	77,41
900	628958,23	5804512,6	07-06-2007 11:02	4,95	328,82
433	628916,8	5804300,07	06-06-2007 10:26	4,95	213,85
800	628996,42	5804406,76	08-06-2007 10:20	4,94	17,77
229	628874,8	5803484,32	05-06-2007 11:00	4,94	293,1
654	628971,73	5804387,2	12-06-2007 10:34	4,94	312,74
467	629006,85	5804372,54	06-06-2007 14:32	4,94	132,74
467	628985,02	5804399,79	06-06-2007 14:33	4,93	317,23
266	629659,27	5804092,53	12-06-2007 10:52	4,92	214,01
924	628597,65	5804236,07	15-06-2007 11:55	4,92	34,03
874	629131,83	5803840,51	01-06-2007 15:18	4,91	349,79
328	629200,25	5804296,37	11-06-2007 14:01	4,91	134,03
360	628870,19	5803504,23	01-06-2007 14:21	4,91	320,91
378	629484,78	5804614,35	11-06-2007 12:08	4,9	167,83
909	629023,82	5803750,87	11-06-2007 15:14	4,9	144,78

**Figura 108 - Data: conjunto de dados *Amsterdam***

#### 5.4. Avaliação do Trabalho Implementado

Esta secção pretende realizar uma síntese, em termos de avaliação, do trabalho realizado durante o desenvolvimento da segunda versão do STAR. Nesse sentido, a avaliação será dividida em duas vertentes: i) balanço das funcionalidades implementadas e ii) melhorias a serem consideradas em trabalhos futuros.

De acordo com o objectivo principal deste projecto, o desenvolvimento do sistema STAR foi estruturado em duas fases. A primeira refere-se ao desenvolvimento de uma base de dados espaço-temporal, que cumpra todos os requisitos do projecto (e.g. armazenamento de dados com características espaciais). A segunda etapa é referente à implementação de uma aplicação (protótipo) que realize todos os processos de gestão, análise e visualização de dados sobre objectos móveis.

Para o desenvolvimento da base de dados de suporte a todo o sistema, foi necessária a realização do estudo dos vários sistemas passíveis de serem utilizados (de acordo com os

requisitos), e como resultado da adopção de uma base de dados foi necessária a aprendizagem de um novo sistema gestor de base de dados (*PostgreSQL*). Para administrar este sistema, foi utilizada a ferramenta *pgAdmin*, que permite realizar todas as tarefas necessárias de administração da base de dados. A necessidade de armazenar dados espaciais (e.g. ponto, linha e polígono) nesta estrutura de suporte, levou ao uso da extensão espacial do *PostgreSQL* denominada de *PostGIS*. O desenvolvimento desta etapa, apesar de ter decorrido sem problemas, não foi trivial, pois nunca tinha trabalhado com este sistema, o que levou à necessidade do estudo e aprendizagem de conhecimentos relativamente ao mesmo.

O desenvolvimento da segunda fase foi dividido em duas grandes componentes, a primeira referente às funcionalidades de gestão (e.g. criação de arquivos, listagem de ficheiros) e a segunda, funcionalidades dedicadas à visualização e análise de dados (e.g. representação dos dados, criação de histogramas, trajectórias).

Apesar de durante a minha formação não adquirir conhecimentos relativos à linguagem *C#*, esta foi mantida como a escolha para o desenvolvimento da segunda versão do STAR. Esta opção foi encarada como bastante motivadora para o desenvolvimento do protótipo, pois para além do enriquecimento pessoal a nível programático, esta apresenta como *framework* de desenvolvimento o *Visual Studio 2010*, uma das ferramentas mais completas para desenvolvimento de *software*.

O desenvolvimento da primeira componente do STAR decorreu sem problemas, pois os conhecimentos adquiridos durante a minha formação académica, ao nível da programação por objectos, permitiram-me ultrapassar as dificuldades de lidar com uma nova linguagem de programação. De seguida são apresentadas todas as funcionalidades implementadas através da indicação dos casos de uso (consultar secção 3.4.2):

- Caso de uso: interface do sistema;
- Caso de uso: gerir arquivos;
- Caso de uso: gerir ficheiros;
- Caso de uso: gerir dados (excepção: analisar e visualizar dados).

O desenvolvimento da segunda componente revelou-se bem mais complexo por diversas razões: i) nunca tinha trabalhado com um SIG; ii) pesquisa de bibliotecas que realizem análises de dados (e.g. criação de histogramas e gráficos de dispersão), cumprindo todos os requisitos do sistema (e.g. bibliotecas *open source*, compatíveis com *C#*); iii) pesquisa de bibliotecas para visualização de dados (e.g. representação de pontos, trajectórias e vectores direccionais), cumprindo todos os requisitos do sistema; iv) necessidade de conhecer todas as

bibliotecas de escolhidas, devido à sua integração com o STAR; v) adaptação de todas bibliotecas com o protótipo desenvolvido por forma a funcionar como um todo.

Seguindo a mesma linha de apresentação utilizada anteriormente, apresentam-se de seguida as funcionalidades de visualização e análise de dados implementadas no STAR:

- Caso de uso: gerir dados;
- Caso de uso: analisar dados;
- Caso de uso: visualizar dados.

Concluída a primeira parte da avaliação, inicia-se a segunda parte com a apresentação de melhorias a serem consideradas em trabalhos futuros. Todas as funcionalidades implementadas no STAR estão 100% operacionais, podendo algumas destas serem melhoradas, pois apresentam algumas limitações. Tais limitações não foram consideradas neste projecto devido ao factor tempo, que não permitiu a eliminação/implementação destas limitações e devido ao facto de estas apenas serem identificadas para melhorias futuras dos resultados finais.

De seguida é apresentada uma lista que visa apenas as funcionalidades implementadas, com possíveis melhorias a serem consideradas em trabalho futuro:

- Alteração das características dos dados representados no mapa (e.g. alterar a cor e a forma dos pontos, agrupar os dados em escala de cor);
- Melhorar o módulo de criação da trajectória de forma a identificar o ponto de partida e chegada de uma determinada trajectória;
- Na criação dos histogramas e gráficos de dispersão, possibilidade de realizar análises a todos os atributos do ficheiro seleccionado (isto é, os que estão considerados na componente *FieldType* ou *OtherFieldType*).

Como conclusão realça-se a passagem da primeira versão do STAR para a versão actual, cumprindo os três principais requisitos deste projecto de investigação: i) o desenvolvimento de todo o protótipo em código livre, o que permite que o sistema esteja acessível a todos os investigadores ou cidadãos interessados nesta temática; ii) a melhoria das funcionalidades de análise e visualização de dados; iii) introdução de diversas opções sobre as representações e cumprimento dos requisitos do sistema, identificados na tabela 4 na secção 3.1.

Por último destaca-se a aprendizagem e utilização de várias tecnologias, bibliotecas e ferramentas (*C#, Visual Studio 2010, PostgreSQL, PostGIS, PgAdminIII, ZedGraph, SharpMap*), para o desenvolvimento da segunda versão do STAR.



## 6. CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS

Este capítulo conclui a descrição do trabalho realizado ao longo deste projecto, apresentando uma revisão do trabalho realizado recorrendo à finalidade e aos objectivos traçados no início deste projecto. Posteriormente são identificadas as perspectivas de trabalho futuro, sendo identificadas novas funcionalidades e perspectivas de melhoramento do protótipo implementado. Por último tecem-se algumas considerações finais acerca de todo o projecto.

### 6.1. Trabalho Realizado

A principal finalidade deste projecto, definida no início deste trabalho, foi a concepção e implementação de um sistema de armazenamento, análise e visualização de dados sobre objectos móveis capaz de representar e armazenar dados com características espaciais e de realizar análises sobre os mesmos. Esta apresentava como requisito o desenvolvimento da segunda versão do STAR recorrendo unicamente a *Software* aberto (*Open Source*), requisito que não foi possível satisfazer na implementação da primeira versão.

De acordo com esta finalidade foram estabelecidos quatro objectivos, cuja concretização permitirá a obtenção dos resultados e contributos esperados para este projecto. De seguida é descrito o nível de sucesso atingido em cada um dos objectivos definidos.

O primeiro objectivo consistia na apresentação do estado da arte dos SIG, modelos de base de dados para objectos móveis, representação de objectos móveis e sistemas similares ao STAR. Nesse sentido, começou-se por apresentar a definição e enquadramento conceptual dos SIG, posteriormente apresentaram-se os modelos de bases de dados para objectos móveis, sendo identificados e descritos os vários tipos de modelos (BDE, BDT e Bases de Dados Espaço-Temporais). Nesta secção é dada particular ênfase às bases de dados espaço-temporais, pois são as que se enquadram no projecto de investigação. De seguida foram abordados os conceitos relacionados com as representações dos dados espaciais e dos objectos móveis, sendo individualizados e definidos cada um dos conjuntos. Finalmente são apresentados três sistemas similares ao STAR: DMOTIS, HERMES e SECONDO. São descritos os seus objectivos, as suas arquitecturas e as suas funcionalidades.

Este enquadramento teórico/bibliográfico permitiu criar os alicerces que suportaram o restante desenvolvimento do projecto, demonstrando a importância dos trabalhos realizados pelos autores e investigadores identificados neste trabalho.

Após a pesquisa bibliográfica o segundo objectivo do trabalho consistia na análise do trabalho realizado para a concretização da primeira versão do STAR. Em particular a análise incidiu sobre a arquitectura do sistema, e as funcionalidades implementadas e identificadas nesta versão. Esta análise permitiu identificar deficiências na estrutura do sistema, bem como em algumas funcionalidades implementadas, servindo de suporte à definição de requisitos para a segunda versão do STAR.

Completa a análise da primeira versão do STAR, o terceiro objectivo consistia em definir os requisitos da versão 2 do STAR. No âmbito deste objectivo foram identificadas melhorias a implementar, novas funcionalidades e funcionalidades já especificadas mas não implementadas, tendo como intuito melhorar aspectos de desempenho da versão anterior e dar prioridade à identificação e implementação de novas funcionalidades, principalmente ao nível da análise de dados. Todo este processo permitiu a assimilação e identificação das lacunas estruturais e funcionais do sistema, conferindo um avanço para o desenvolvimento da segunda versão do sistema.

Depois de realizada a especificação dos requisitos do sistema (secção 3.1: tabela 4) e contextualizadas todas as novas opções de desenvolvimento, foi possível desenhar o novo modelo de dados e a nova arquitectura do sistema que agrega quatro camadas: camada de apresentação, camada de negócio, camada de dados e camada de apoio. Esta reestruturação do STAR em camadas reduz a complexidade do sistema (agrupamento de funcionalidades), favorece a coesão (agrupamento de componentes com responsabilidades relacionadas), promove a reutilização e futuros avanços do sistema (pois as camadas podem ser utilizadas em outros sistemas, ou podem ser substituídas, sem danos no mesmo) e reduzem a dependência (a regra de comunicação evita dependências directas entre componentes de camadas diferentes).

A camada de apresentação permite a interacção entre o utilizador e a aplicação. Esta implementa uma interface de entrada e saída, com o papel fundamental de realizar as validações das informações fornecidas pelo utilizador.

A segunda camada implementa toda a lógica de negócio da aplicação, fornecendo a API necessária para todas as funcionalidades do sistema. Esta agrupa operações como o importe,

correção e normalização de dados fornecidos, geração de mapas, histogramas, gráficos de dispersão, cálculo de vectores, trajectórias, entre outras.

A camada de dados mapeia os dados existentes na base de dados em entidades persistentes em memória. Esta é o único elo de ligação com a base de dados, sendo de importância maior para garantir a consistência dos mesmos.

Por último a camada de apoio representa a infra-estrutura de toda a aplicação, apresentando todas as classes e bibliotecas necessárias para a realização de todas as tarefas implementadas na segunda versão do STAR.

O modelo de dados da segunda versão do STAR é composto por 8 tabelas: *staruser*; *folder*; *file*; *data*; *otherdata*; *shapefile*; *geometry\_columns* e *spatial\_ref\_sys*.

Em resumo, este objectivo foi cumprido, apontando-se como uma das contribuições deste trabalho as funcionalidades do STAR especificadas ao longo deste trabalho, a nova arquitectura do STAR e o cumprimento dos requisitos do sistema, na implementação da segunda versão do STAR.

A implementação da segunda versão do STAR representa o último objectivo deste trabalho. Esta engloba uma fase de testes, onde pode ser verificado o desempenho do sistema implementado. Na implementação da segunda versão do STAR recorreu-se ao *PostgreSQL* com a extensão *PostGIS* para o desenvolvimento/armazenamento da base de dados de objectos móveis. Para a visualização dos dados ao nível geográfico, recorreu-se à biblioteca *SharpMap*, sendo que para os restantes gráficos recorreu-se à biblioteca *ZedGraph*. O uso de todas estas tecnologias permitiu cumprir o principal requisito desta segunda versão, a utilização única de *Software* aberto, para o desenvolvimento de todo o sistema.

Depois de verificado o cumprimento de cada um dos objectivos, apresentam-se de seguida as principais contribuições deste projecto, relativas às aplicações analisadas no estado da arte e à primeira versão do STAR:

- A possibilidade de efectuar análises a vários objectos em movimento (e.g. pedestres, veículos, animais) é uma das principais vantagens em relação às aplicações analisadas na secção 2.3. É possível verificar na secção 5.2 que o STAR possui a capacidade de importação de ficheiros para a base de dados independentemente do tipo de dados que estão armazenados no ficheiro e de quais e quantos atributos se encontram no mesmo;

- A possibilidade de importar dados de diversos tipos de dispositivos de posicionamento (e.g. *GPS*, *RFID*, *bluetooth*), não sendo verificada esta característica nas aplicações apresentadas no capítulo 2;
- A representação de movimento não se baseia unicamente nas trajectórias (Aplicações apresentadas no capítulo 2), pois no STAR verificamos que este pode representar trajectórias, pontos e vectores direccionais. Nas representações do movimento, existe ainda possibilidade de ampliar, reduzir e mover o mapa de acordo com o que é pretendido pelo utilizador (secção 5.3.2);
- O STAR permite criar gráficos de dispersão e histogramas para análise de dados, permitindo uma série de opções acerca dos mesmos (secção 5.2.3). Nos casos estudados, não foi possível encontrar funcionalidades iguais ou similares às referidas anteriormente;
- A capacidade de importação de ficheiros de dados sobre objectos móveis para a base de dados independentemente do tamanho e do conteúdo do mesmo (independente dos tipos atributos e da sua quantidade);
- A correcção de dados de um ficheiro, importados para o sistema, que permite realizar análises e visualizações de dados de uma forma mais eficiente (5.2.3). Em nenhum dos sistemas analisados é possível a correcção de dados de um ficheiro;
- Alteração da conexão à base de dados. Isto possibilita a ligação a bases de dados diferentes, podendo estas serem remotas ou locais. Nenhuma das aplicações analisadas apresenta esta funcionalidade;
- O desenvolvimento de todo o sistema recorrendo unicamente a *software* livre.

## 6.2. Trabalhos Futuros

Concluída a síntese do trabalho realizado, dando cumprimento à finalidade e objectivos definidos para este trabalho, esta secção formula propostas de trabalho futuro que visam a melhoria, a promoção e a evolução do STAR.

As propostas de trabalho futuro visam sobretudo a melhoria das opções funcionais que permitiram chegar à segunda versão do STAR. Assim numa primeira fase, pretende-se melhorar e otimizar as funcionalidades desenvolvidas utilizando as sugestões referidas na

secção 5.4. Numa segunda fase a possibilidade de implementação de novas funcionalidades como: i) a possibilidade do STAR receber dados em tempo real através de vários dispositivos de posicionamento. Isto só será possível através da implementação de um ambiente de testes, que permitirá avaliar os STAR em cenários reais; ii) Implementação de algoritmos de *Clustering* na análise dos dados; iii) evolução do STAR para um sistema cliente-servidor.

### 6.3. Considerações Finais

Este trabalho foi realizado tendo como finalidade a concepção e implementação de um sistema de armazenamento, análise e visualização de dados sobre objectos móveis capaz de representar e armazenar dados com características espaciais e de realizar análises sobre os mesmos.

Este projecto iniciou-se com o levantamento do estado da arte que enquadrou um conjunto de fundamentos teóricos relevantes associados aos modelos de bases de dados para objectos móveis, aos SIG, à representação dos objectos móveis e a análise a sistemas similares ao STAR. Através da revisão de literatura foi possível identificar um conjunto de aplicações que possibilitaram a definição de um conjunto de funcionalidades essenciais a um sistema de análise de dados sobre o movimento bem como o desenho da sua arquitectura.

A análise de um conjunto de modelos de dados, e da primeira versão do STAR, permitiu construir o modelo de dados do sistema através da proposta de um diagrama de classes. Os diagramas de caso de uso, que representam as funcionalidades do sistema, o modelo de dados do sistema e a sua arquitectura, auxiliaram o desenvolvimento e implementação da segunda versão do STAR.

A demonstração das funcionalidades do sistema e a análise de dados de dois conjuntos de dados carregados no sistema com o formato *csv*, permitiram testar e comprovar a utilidade do STAR. Verificou-se que, apesar de não possuir mais informação do que a que era fornecida pelos dados dos ficheiros, foi possível verificar o comportamento das entidades móveis analisadas.

Conclui-se, assim, com a sistematização dos principais resultados obtidos neste projecto e com a constatação da concretização de todos os objectivos traçados inicialmente. Apesar de ter sido um projecto muito exigente, devido aos diversos desafios que foi necessário ultrapassar, termino este projecto gratificado e com um sentimento de realização pelo facto de ter atingido e concretizado as metas traçadas para este projecto de investigação.



---

## REFERÊNCIAS BIBLIOGRÁFICAS

- Abdul-Kadar, H. M. (2009). *Location Updating Strategies in Moving Object Databases*. International Journal of Computer Theory and Engineering, Vol. 1, No. 1, 1793-8201.
- Adam, N. R., e A. Gangopadhyay, (1997). *Database Issues in Geographic Information Systems*, Kluwer. Academic Publishers, Massachusetts.
- Anda, B., e Sjoberg, D. I. K. (2005). *Investigating the Role of Use Cases in the Construction of Class Diagrams*. Empirical Software Engineering, 10, 285-309.
- Andrienko, N., Andrienko, G., Pelekis, N. e Spaccapietra, S. (2008). "Basic Concepts of Movement Data," *Mobility, Data Mining and Privacy*, F. Giannotti and D. Pedreschi, pp. 15-38.
- Almeida, V.T., Gutting, R.H. e Behr, T. (2006). *Querying Moving Objects in Secondo*. Fernuniversität Hagen D-58084 Hagen, Germany.
- Aronoff, S. (1989), *Geographic Information Systems: a management perspective*. WDL Publications, Ottawa.
- Bauer, C.; King, G. (2007). *Java Persistence with Hibernate – 1o Ed. Manning*. Second Edition of Hibernate in Action, ISBN: 1-932394-88-5,
- Booch, G., Rumbaugh, J. e Jacobson, I. (1999). *The Unified Modeling Language User Guide: Addison Wesley Longman, Inc.*
- Burrough, P. A. (1986). *Principles of Geographical Information Systems for Land Resources Assessment*. Oxford: Oxford University Press.
- Calçado, P. (2006). *Arquitetura de Camadas em Java EE*. Mundo Java. Rio de Janeiro, v.3, n.15.
- Câmara, G., Davis, C., e Monteiro, A. (2001). *Introdução à Ciência da Geoinformação*. Ministério da Ciência e Tecnologia: Instituto Nacional de Pesquisas Espaciais, INPE-10506-RPQ/249.
- Câmara, G., e Freitas, U. (1995). *Perspectivas em Sistemas de Informação Geográfica*. Fator GIS, 1 (4).
- Câmara, G., e Medeiros, J. S. (2003). *Princípios Básicos em Geoprocessamento*. Em E. D. Assad e E. E. Sano (Eds.), *Sistema de informações geográficas: aplicações na agricultura* (pp. 3-11). Brasília: Embrapa – CPAC.

- Cowen, D.J. (1988). *GIS versus CAD versus DBMS: what are the differences?* Photogrammetric Engineering and Remote Sensing 54: 1551-4.
- Davis, F. W., e Simonett, D. S. (1991). *GIS and Remote Sensing*. In D. J. Maguire, M. F. Goodchild e D. W. Rhind (Eds.), *Geographical Information Systems: Principles and Applications* (Vol. 1, pp. 191- 213). London: Longman Scientific and Technical.
- Dieker, S., e Güting, R.H., (2000). *Plug and Play with Query Algebras: SECONDO*. A Generic DBMS Development Environment. Proc. Int. Database Engineering and Applications Symposium (IDEAS, Yokohama, Japan), 380-392.
- Dobing, B., e Parsons, J. (2006). *How UML is used*. Communications of the ACM, 49, 109-113.
- Dodge, S., e Alesheikh A. (2005). *Evaluating different approaches of spatial database management for moving objects*. Faculty of Geodesy and Geomatics Eng., K.N. Toosi University of Technology.
- Egenhofer, M. (1994). *Spatial SQL: A Query and Presentation Language*. IEEE Transactions on Knowledge and Data Engineering, 6(1), 86-95.
- Erwig, M., Güting, R. H., Schneider, M. e Vazirgiannis, M. (1999). *Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases*. GeoInformatica, vol. 3, pp. 269- 296.
- Ferros, L. (2009). *Extracção e concentração de metainformação distribuída por vários repositórios*. Universidade do Minho, Escola de Engenharia, Departamento de Informática.
- Fotheringham, S., e Rogerson, P. (1994). *Spatial analysis and GIS*. Bufalo: Taylor and Francis.
- Gatrell, A. C. (1991). *Concepts of space and geographical data*. Em D. J. Maguire, M. F. Goodchild e D. W. Rhind (Eds.), *Geographical Information Systems: Principles and Applications* (Vol. 1, pp. 119–134). Harlow: Longman Scientific and Technical.
- Gutting, R.H. (1989). *Gral: An Extensible Relational Database System for Geometric Applications*. Proc. of the 15th Intl. Conf. on Very Large Data Bases, Amsterdam, 33-44.
- Gutting, R.H. (1993). *Second-Order Signature: A Tool for Specifying Data Models, Query Processing, and Optimization*. Em: Proc. ACM SIGMOD Conference. Washington, USA, 277-286.
- Gutting, R. H. (1994). *An Introduction to Spatial Database Systems*. VLDB Journal, 3(4).



- Gutting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M. e Vazirgiannis, M. (2000). *A foundation for representing and quering moving objects*. ACM Transactions on Database Systems, vol. 25, pp. 1-42.
- Gutting, R.H., Behr, T., Almeida V., Ding, Z., Hoffmann, F. e Spiekermann, M. (2004). *SECONDO: An Extensible DBMS Architecture and Prototype*. Fernuniversität Hagen, Informatik-Report 313.
- Gutting, R.H. and Schneider, M. (2005). *Moving Objects Databases*. Morgan Kaufmann Publishers.
- Gutting, R.H., Ansorge, D., Behr, T. e Spiekermann, M. (2007). *Secondo User Manual*. Praktische Informatik IV, Fernuniversität Hagen D-58084 Hagen, Germany.
- Hagerstrand, T. (1970). *What about people in Regional Science?*. Papers in Regional Science, vol. 24, pp. 6-21, December.
- Hornsby, K. e Egenhofer, M.J. (2002). *"Modeling Moving Objects over Multiple Granularities,"*. Annals of Mathematics and Artificial Intelligence, vol. 36, pp. 177-194.
- Lee, E. J. e Ryu K. H. (2005). *Design of Vehicle Information Management System for Effective Retrieving of Vehicle Location*. Em: Int. Conf. on Computational Science and its Applications, Springer-Verlag.
- Maguire, D. J., e Dangermond, J. (1991). *The Functionality of GIS*. In D. J. Maguire, M. F. Goodchild e D. W. Rhind (Eds.), *Geographical Information Systems: Principles and Applications* (Vol. 1, pp. 319–335). London: Longman Scientific and Technical.
- Mendes, J. M., (2010). *STAR: Um Sistema de Armazenamento, Análise e Visualização de dados sobre o Movimento Humano*. Tese de Mestrado, Universidade do Minho.
- Miller, H. e Shaw, S. (2001). *Geographic information systems for transportation: principles and applications*. Oxford: Oxford University Press.
- Nóbrega, E., Rolim, J.T. e Times, V. (2004). *Representing uncertainty, profile and movement history in mobile objects databases*. Em: GeoInfo'2004, Brazil.
- Orellana, D., Wachowicz, M., Andrienko, N., e Andrienko, G. (2009). *Uncovering interaction patterns in mobile outdoor gaming*.
- Ott, T. e Swisaczny, F. (2001). *Time-Integrative Geographic information system: Management and analysis of spatio-temporal data*. Springer.

Pelekis, N., Frentzos, E., Giatrakos, N. e Theodoridis, Y. (2008). *HERMES: Aggregative LBS via a Trajectory DB Engine*. Department of Informatics, University of Piraeus, 80 Karaoli-Dimitriou St, GR-18534 Piraeus, Greece.

Pelekis, N., Theodoridis Y., Vosinakis S. e Panayiotopoulos T. (2006). *HERMES - A Framework for Location-Based Data Management*. Proceedings of EDBT.

Pelekis, N. e Theodoridis Y. (2006). *Boosting Location-Based Services with a Moving Object Database Engine*. Proceedings of MobiDE.

Pessoa, J. (2001). *Um estudo sobre sistema de banco de dados cliente/servidor*. ASPER- Associação Paraibana de Ensino Renovado, Faculdade Paraibana de processamento de dados.

Pfoser, D., Jensen, C. S., Theodoridis Y. (2000). *Novel Approaches to the Indexing of Moving Object Trajectories*. Proceedings of VLDB.

Praing, R. e Schneider, M. (2007). *A Universal Abstract Model for Future Movements of Movement Objects*. In Fabrikant, S. I. and Wachowicz, M. (Eds.): *The European Information Society: Leading the way with geo-information*, Springer-Verlag.

Qajary, Y., e Alesheikh, A. (2008). *Design and Implementation of a Moving Object Database for Truck Information Systems*.

Qiu, k., Hachem, N. I., Ward, M. O. e Gennert, M. A. (1992). *Providing Temporal Support in Data Base Management Systems for Global Change Research*. Data/Knowledge Base Research Group, Computer Science Department, Worcester Polytechnic Institute, MA 01609.

Roshannezhad, Aliasghar (1996). *The Management Of Spatio-Temporal Data in a National Geographic Information System*.

Rumbaugh, J., Jacobson, I. e Booch, G., (1998). *The Unified Modeling Language Reference Manual*. Addison-Wesley.

Samet, H., e Aref, W. (1994). *Spatial data models and query processing*. Modern Database Systems: The Object Model, Interoperability and Beyond.

Santos, M. (2001). *Padrão: um sistema de descoberta de conhecimento em bases de dados geo-referenciadas*. Unpublished PhD thesis, Universidade do Minho, Guimarães.

Santos, M. Y., e Amaral, L. A. (2002). *Técnicas de modelação de informação geográfica: uma síntese*. Conferência da Associação Portuguesa de Sistemas de Informação, Coimbra.

- Santos, M. Y., Mendes, J., Moreira, A. E Wachowicz, M. (2011). *Towards a Spatio-Temporal Information System for Moving Objects. Moving Objects*, Springer-Verlag, 978-3-642-21928-3.
- Sarace, M H. e Theodoulidis, B. (1995). *Knowledge Discovery in Temporal Databases*. London: The Institution of Electrical Engineers, IEE, WC2R 0BL.
- Schmitz, L. K. (2010). *Geotecnologias Aplicadas ao SIG*. Departamento de Arquitetura. Cieg.
- Schunemann, F. (2010). *Diferenças entre Sistemas Gerenciadores de Banco de Dados para GIS – SGBD*.
- Shekhar, S., Coyle, M., Goyal, B., Liu, D. e Sarkar, S. (1997). *Data Models in Geographic Information Systems*. Communication of the ACM, Vol 40, No.4.
- Shekhar, S., Ravada, S. e Liu, X. (1999). *Spatial Databases Accomplishments and Research Needs*, IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1.
- Shekhar, S., Chawla, S., Ravada, S., Fetterer, A., Liu, X., e Lu, C. T. (1999). *Spatial Databases - Accomplishments and Research Needs*. IEEE Transactions on Knowledge and Data Engineering, 11(1), 45-55.
- Silva, C. (2007). *Análise e Avaliação do Framework HIBERNATE em uma aplicação cliente/servidor*. Jaguariúna.
- Silva, J. e Santos M.Y. (2010). *Spatiotemporal Database Models and Languages for Moving Objects: A Review*.
- Snodgrass, R. e Ahn, I. (1986). *Temporal Databases*. University of North Carolina at Chapel Hilln AT&T Bell Laboratories, 0018-9162/86/0900-0035\$0100.
- Theodoridis, Y. e Benchmark, T. (2003). *Database Queries for Location-based Services*. The Computer Journal, 46, 6, 713-725.
- Tobler, W.R. (2003). "Movement Mapping," Center for Spatially Integrated Social Science. <http://www.csiss.org/clearinghouse/FlowMapper/2003>.
- Wachowicz, M., Ligtenberg, A., Renso, C., e Gürses, S. (2008). *Characterising the Next Generation of Mobile Applications Through a Privacy-Aware Geographic Knowledge Discovery Process*. Em F. Giannotti e D. Pedreschi (Eds.), *Mobility, Data Mining and Privacy* (pp. 39-72). Berlin: Springer-Verlag.
- Weng, J., Wang, W., Fan K. e Huang J. (2005). *Design and Implementation of Spatial-temporal Data Model in Vehicle Monitor System*. In: The 8th Int. Conf. on GeoComputation, University of Michigan.

Wolfson O., Cao H., Lin H., Trajcevski G., Zhang F. e Rishe N. (2002). *Management of dynamic location information in domino*. Proceedings of EDBT.

Worboys, M. F., Hearnshaw, H. M., e Maguire, D. J. (1990). Object-Oriented Modelling for Spatial Databases. *International Journal of Geographical Information Systems*, 4(4), 369-383.

Worboys, M.F. (1995). *Geographic Information Systems: A Computing Perspective*, Taylor and Francis.

Yeung, A. K. W., e Hall, G. B. (2007). *Spatial Data and Spatial Database Systems* (Vol. 87). Dordrecht: Springer.

Zhang, J. (2005). *Spatio-Temporal Database*. Ecole Polytechnique Fédérale de Lausanne (EPFL) School of Computer and Communication Sciences CH-1015 Lausanne, Switzerland.

## **ANEXOS**

## Anexo 1 – Script de Criação da BD

```
CREATE DATABASE star
  WITH OWNER = postgres
       ENCODING = 'UTF8'
       TABLESPACE = pg_default
       LC_COLLATE = 'Portuguese_Portugal.1252'
       LC_CTYPE = 'Portuguese_Portugal.1252'
       CONNECTION LIMIT = -1;

CREATE TABLE staruser
(
  user_id integer NOT NULL DEFAULT nextval('user_user_id_seq'::regclass),
  username character varying(15) NOT NULL,
  "password" character varying(15) NOT NULL,
  "name" character varying(30),
  birthdate date,
  gender character(1),
  email character varying(30),
  CONSTRAINT pk_user PRIMARY KEY (user_id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE staruser OWNER TO postgres;

CREATE TABLE folder
(
  folder_id serial NOT NULL,
  user_id serial NOT NULL,
  foldername character varying(300),
  description character varying(500),
  date date DEFAULT now(),
  CONSTRAINT folder_pmk PRIMARY KEY (folder_id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE folder OWNER TO postgres;
```

```
CREATE TABLE file
(
  file_id serial NOT NULL,
  folder_id serial NOT NULL,
  filename character varying(300),
  public boolean,
  uploaddate date DEFAULT now(),
  description character varying(500),
  user_id serial NOT NULL,
  CONSTRAINT file_pmk PRIMARY KEY (file_id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE file OWNER TO postgres;

CREATE TABLE shapefiles
(
  id_shapefile serial NOT NULL,
  file_id serial NOT NULL,
  geom geometry,
  CONSTRAINT shapefile_pmk PRIMARY KEY (id_shapefile)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE shapefiles OWNER TO postgres;

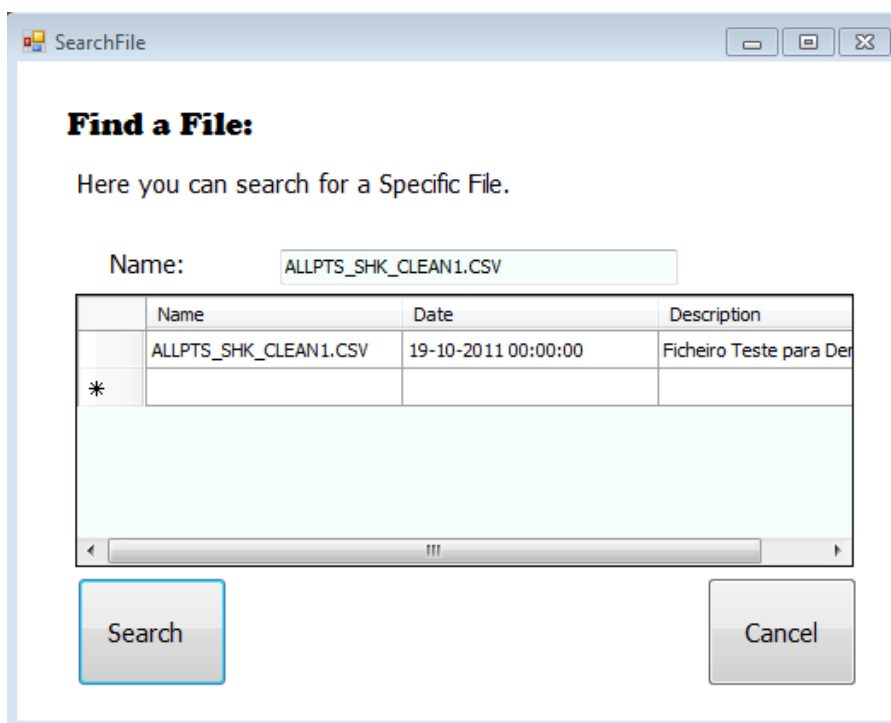
CREATE TABLE data
(
  data_id serial NOT NULL,
  file_id serial NOT NULL,
  x double precision,
  y double precision,
  t timestamp without time zone,
  speed double precision,
  bearing double precision,
  device_id character varying(300),
  point geometry,
  CONSTRAINT data_pmk PRIMARY KEY (data_id)
```

```
)  
WITH (  
    OIDS=FALSE  
);  
ALTER TABLE data OWNER TO postgres;  
  
CREATE TABLE otherdata  
(  
    otherdata_id integer NOT NULL DEFAULT nextval  
    ("otherData_otherData_id_seq"::regclass),  
    data_id integer NOT NULL DEFAULT nextval  
    ("otherData_data_id_seq"::regclass),  
    otherfieldname character varying(30) NOT NULL,  
    "value" character varying(300),  
    CONSTRAINT "pk_otherData" PRIMARY KEY (otherdata_id)  
)  
WITH (  
    OIDS=FALSE  
);  
ALTER TABLE otherdata OWNER TO postgres;
```

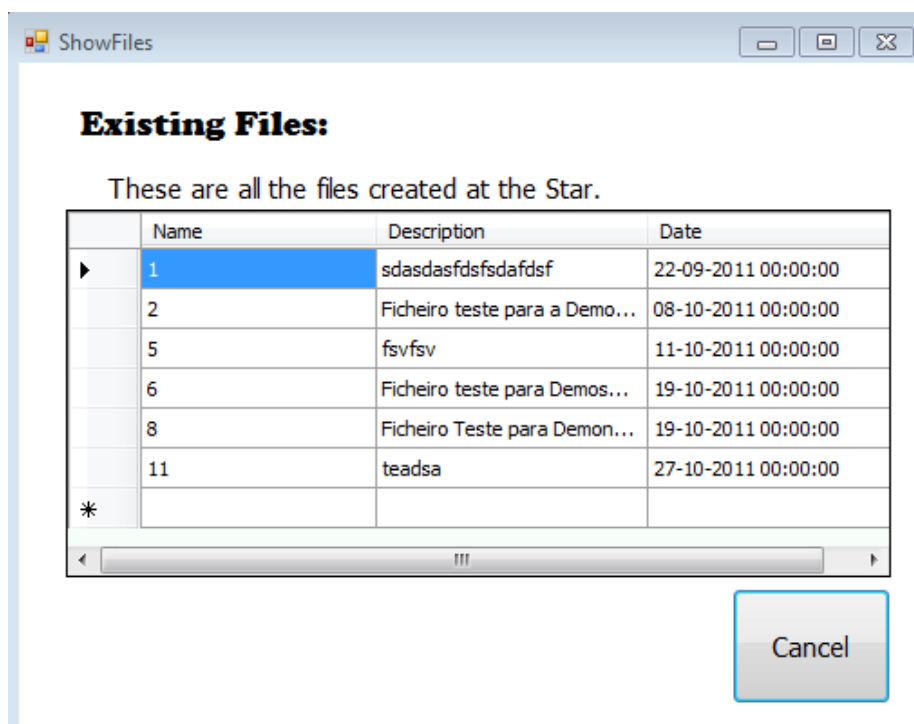


## Anexo 2 – Figuras do Sistema STAR v. 2.0

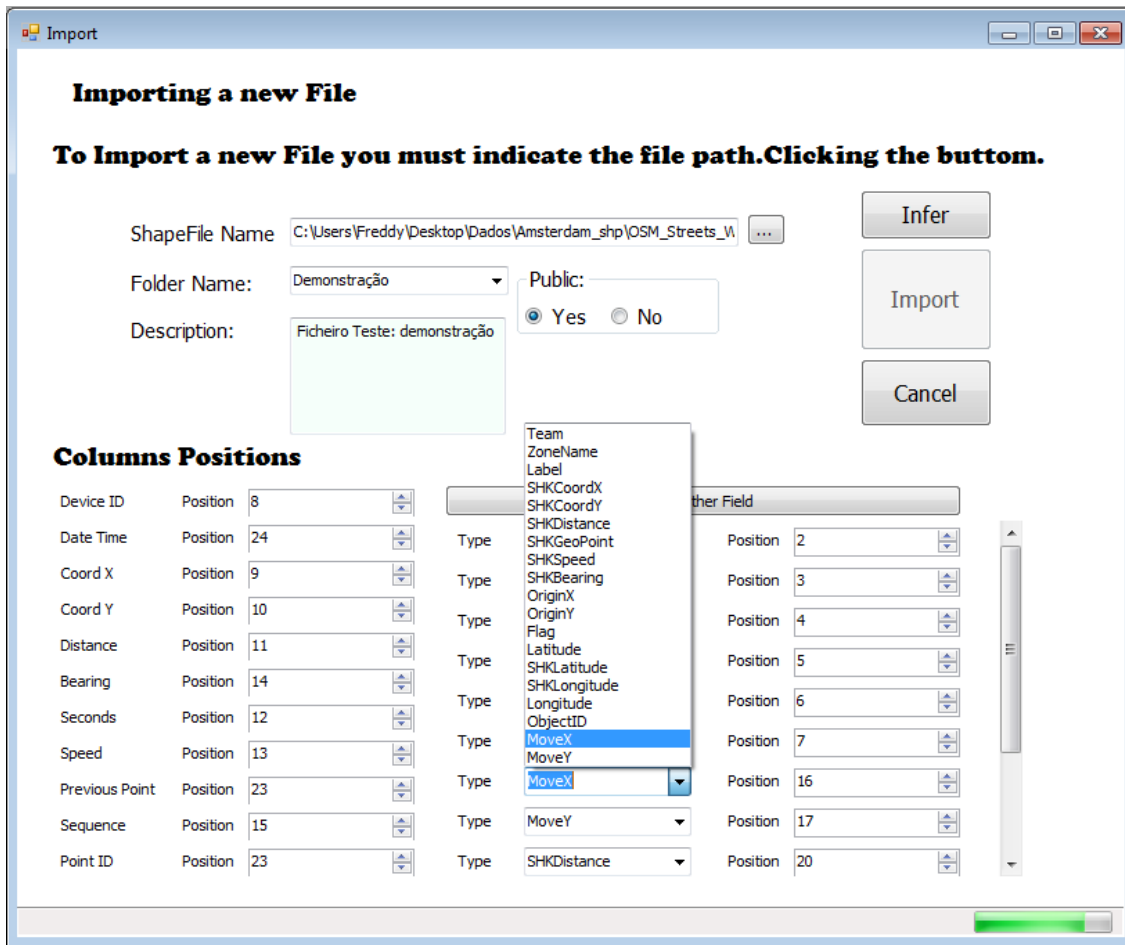
As Figuras a seguir apresentadas demonstram o funcionamento do STAR.



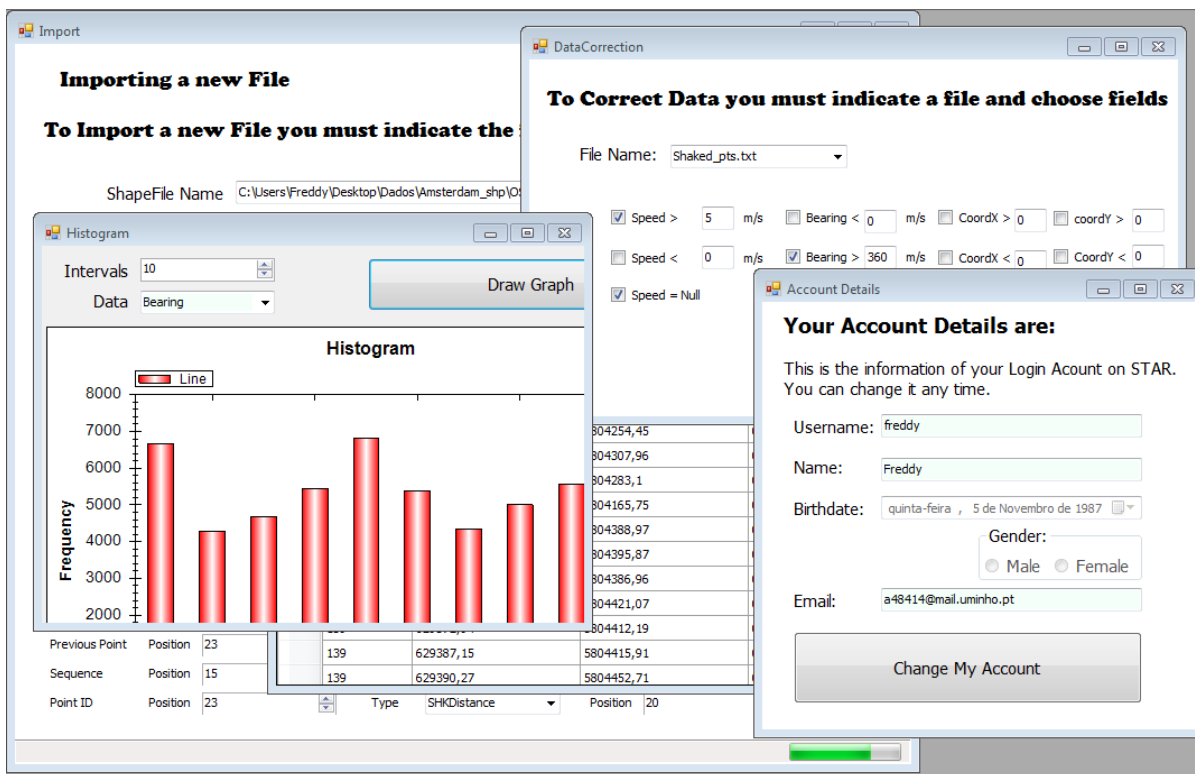
Anexo 2: Figura 1 – Search File



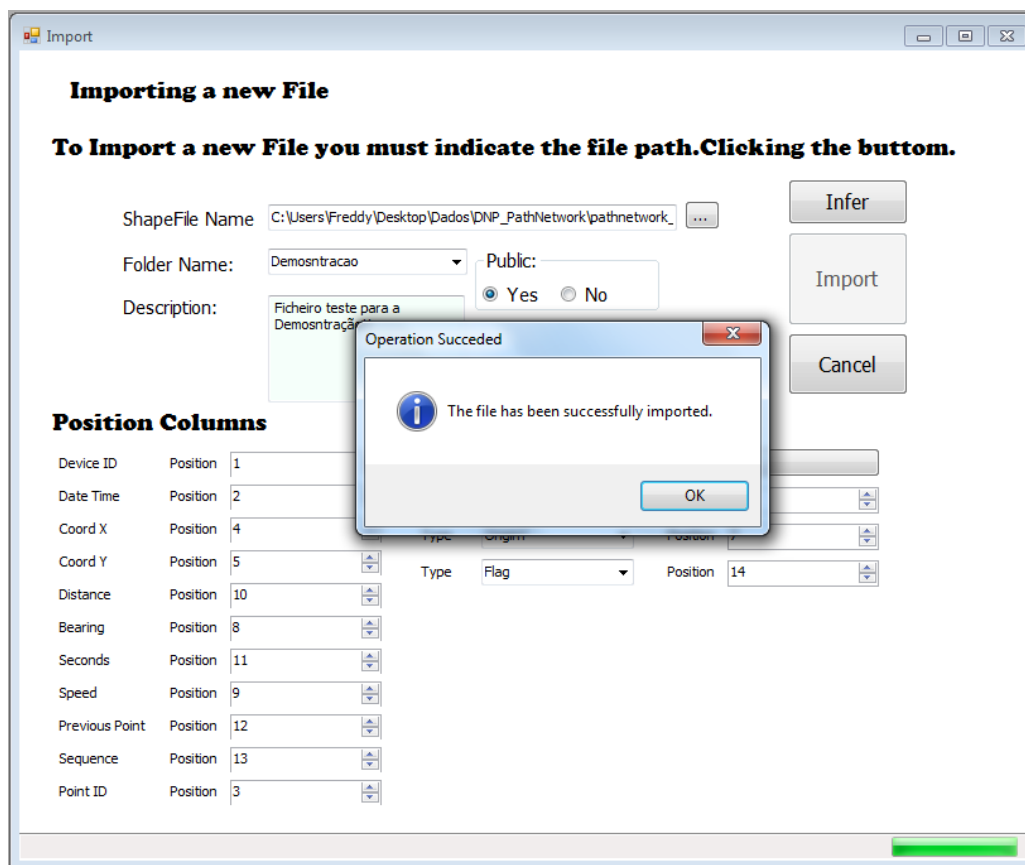
Anexo 2: Figura 2 – List of Files



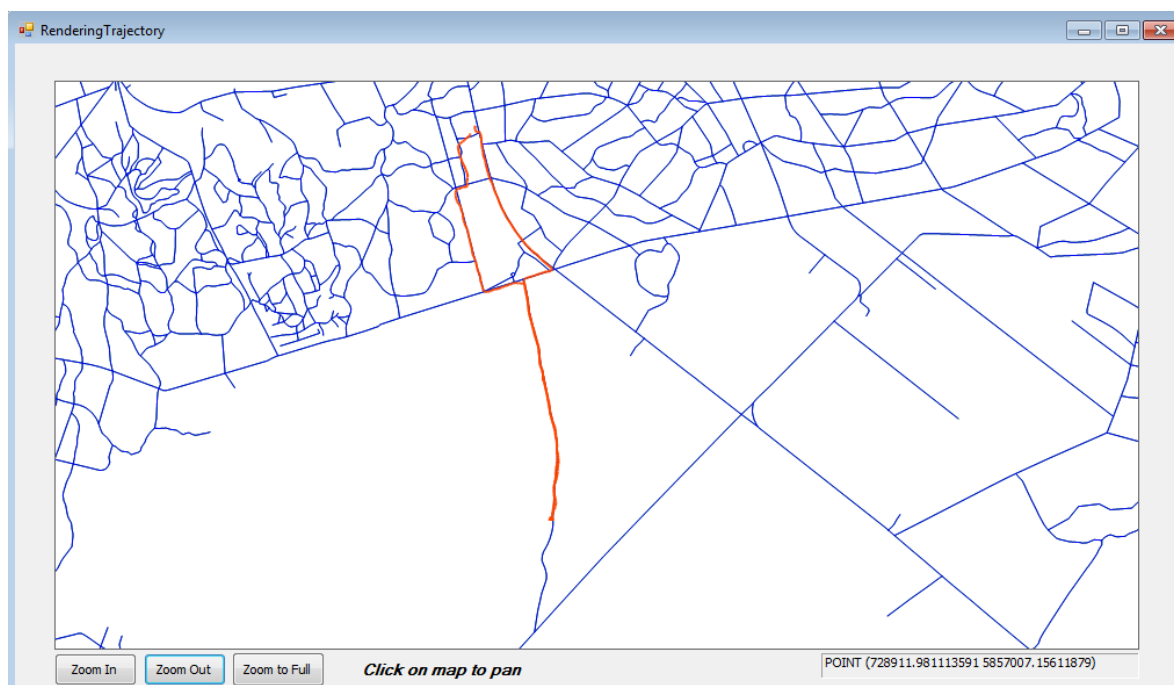
Anexo 2: Figura 3 – Import File



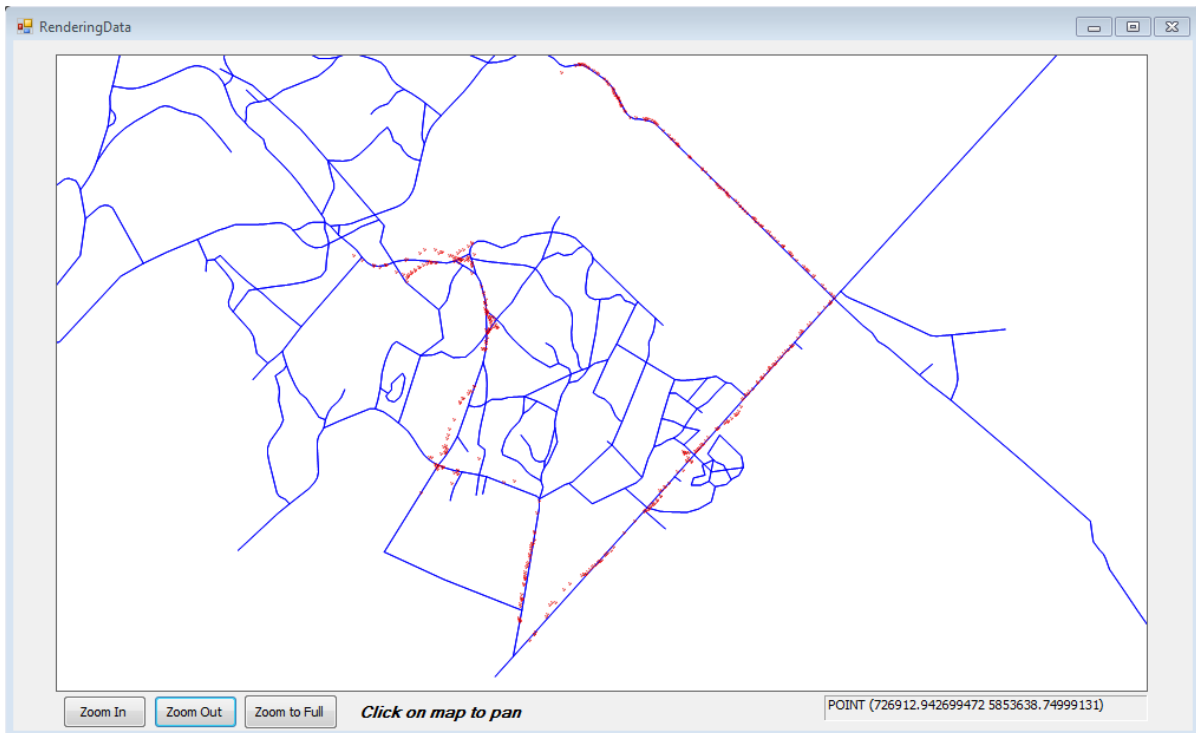
Anexo 2: Figura 4 – Import File



Anexo 2: Figura 5 – Import File



Anexo 2: Figura 6 – Draw Trajectory

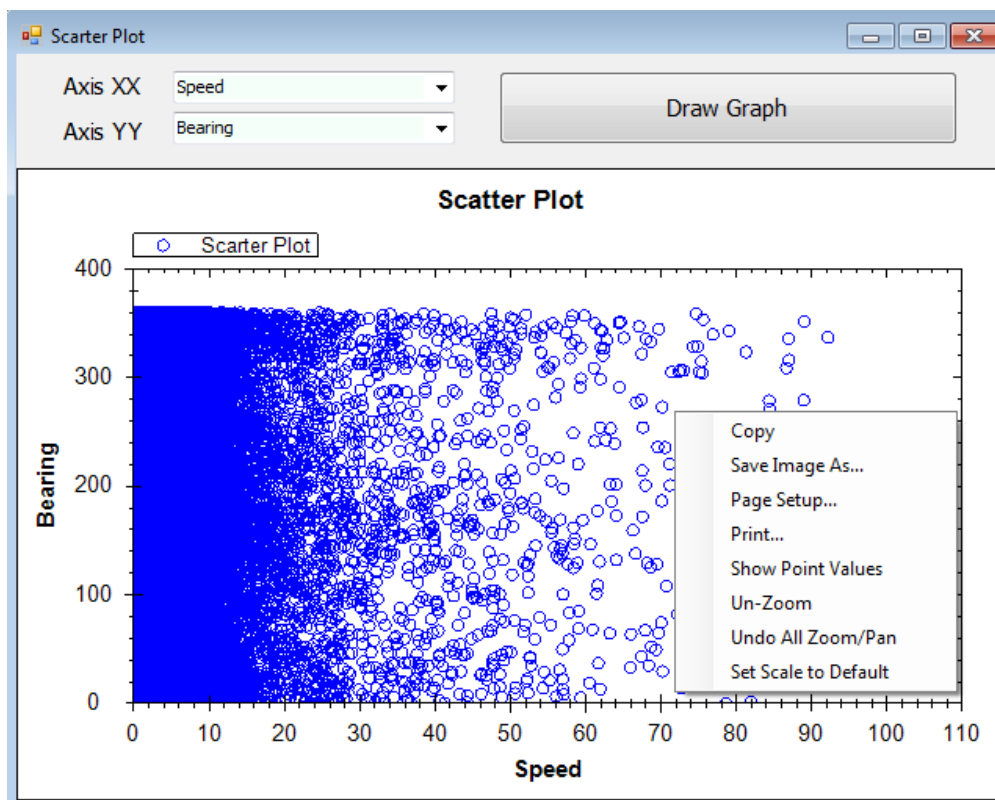
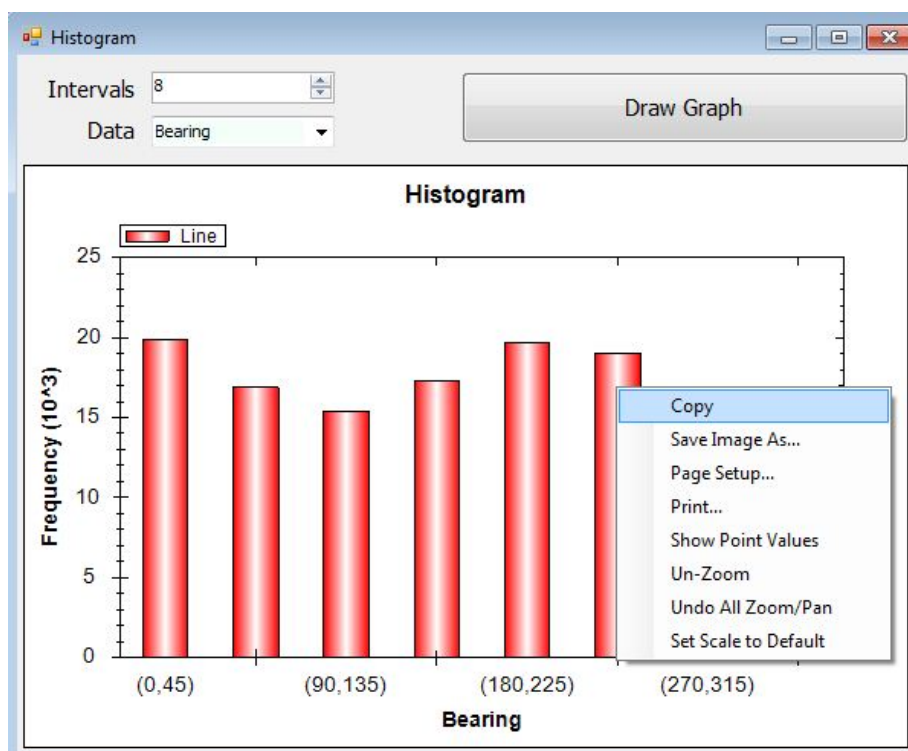


Anexo 2: Figura 7 – Draw Points

The screenshot shows a window titled 'Show Data' with a table of data points. The table has the following columns: Device ID, X, Y, Date Time, Speed, and Bearing. The data is as follows:

Device ID	X	Y	Date Time	Speed	Bearing
R483	728923,54	5857416,37	06-08-2006 13:52	9,94	289,46
R483	729568,13	5856597,3	06-08-2006 13:22	9,86	25,33
R709	731286,58	5854345,06	19-08-2006 15:03	9,85	100,94
R709	729774,59	5853867,19	19-08-2006 14:27	9,75	35,12
R483	728882,46	5856732,05	06-08-2006 13:14	9,55	105,12
R709	731294,44	5854359,14	19-08-2006 15:03	9,48	15,33
R483	729806,06	5856990,18	06-08-2006 13:32	9,46	100,66
R709	732545,33	5857426,58	19-08-2006 11:14	9,45	312,34
R483	728627,01	5857447,25	06-08-2006 14:00	9,34	351,22
R709	731444,3	5854571,19	19-08-2006 15:06	9,31	17,49
R709	728266,35	5853312,25	19-08-2006 13:56	9,29	144,16
R709	732459,97	5857391,95	19-08-2006 11:15	9,17	294,68
R483	729533,78	5856527,22	06-08-2006 13:22	9,15	6,32
R483	729652,93	5856699,25	06-08-2006 13:24	9,11	349,73
R709	732512,19	5857391,65	19-08-2006 11:15	8,97	244,26
R483	729653,74	5856689,86	06-08-2006 13:24	8,9	356,95
R709	732546,2	5856958,31	19-08-2006 15:54	8,9	7,74
R709	732523,49	5856791,8	19-08-2006 15:52	8,82	41,91
R483	729353,68	5856602,97	06-08-2006 13:19	8,74	116,31
R483	728653,07	5857228,37	06-08-2006 13:58	8,72	7,02
R709	732329,2	5855911,42	19-08-2006 15:38	8,67	312,19
R709	729478,05	5853552,83	19-08-2006 14:20	8,64	51,92
R709	729416,09	5853470,76	19-08-2006 14:18	8,64	347,49
R709	730440,93	5854087,21	19-08-2006 14:41	8,63	140,81
R709	732950,3	5857643,69	19-08-2006 11:04	8,61	217,28
R709	732418,01	5857135,59	19-08-2006 11:21	8,56	258,85
R709	728100,5	5853454,32	19-08-2006 13:53	8,54	170,96

Anexo 2: Figura 8 – Show Data

Anexo 2: Figura 9 – *Scarter Plot*Anexo 2: Figura 10 – *Histogram*

Show Data

**Data**  
These are all the entities selected at Star.

	Device ID	X	Y	Date Time	Speed	Bearing
	R729	730003,99	5856107,22	17-08-2006 11:37	2,88	300,96
	R201	731697,79	5858082,93	25-05-2006 12:53	2,88	173,2
	R195	727878,24	5856432,65	18-05-2006 12:29	2,88	271,15
	R440	727815,12	5853727,28	06-08-2006 12:47	2,88	229,06
	R519	727999,15	5853645,12	06-08-2006 12:35	2,88	290,75
	R236	727319,72	5854157,71	28-05-2006 13:36	2,88	277,42
	R147	733715,61	5858572,81	25-05-2006 10:40	2,88	33,91
	R708	732500,78	5856750,81	19-08-2006 11:15	2,88	209,77
	R119	726942,6	5853703,36	28-05-2006 10:01	2,88	1,43
	R114	731677,28	5858059,68	25-05-2006 11:04	2,88	207,61
	R280	727678,19	5856523,83	28-05-2006 13:28	2,88	157,19
	R740	727479,42	5853317,86	19-08-2006 09:37	2,89	136,68
	R273	727346,86	5857251,04	28-05-2006 11:41	2,89	164,67
	R585	727153,03	5854575,51	09-08-2006 12:21	2,89	344,3
	R253	727449,15	5853375,37	25-05-2006 13:32	2,89	156,64
	R263	727293,67	5856248,8	25-05-2006 14:09	2,89	269,63
	R255	727524,5	5853305,4	28-05-2006 10:36	2,89	0,36
	R711	732676,14	5858192,28	19-08-2006 13:34	2,89	203,22
	R118	725531,71	5855141,62	28-05-2006 11:41	2,89	358,91
	R755	728643,24	5857224,36	17-08-2006 11:02	2,89	183,81
	R485	729609,26	5856892,25	06-08-2006 13:14	2,89	244,38
	R094	729068,26	5857200,9	18-05-2006 13:24	2,89	79,48
	R759	729543,58	5856904,52	17-08-2006 14:12	2,89	287,93
	R689	728807,51	5857171,59	09-08-2006 10:23	2,89	354,99
	R243	728097,35	5853526,63	28-05-2006 11:54	2,89	178,33
	R280	726544,25	5856473,65	28-05-2006 14:01	2,89	330,68
	R408	727584,5	5856325,26	06-08-2006 12:48	2,89	287,48

Anexo 2: Figura 11 – *Show Data*

### Anexo 3 – Figuras do Sistema STAR v. 2.0

As Figuras seguintes demonstram a visualização de dados em tabela, dos pedestres R263, R660 e R656.

**Data**  
These are all the entities selected at Star.

	X	Y	Date Time	Speed	Bearing
	727512,67	5856803,9	25-05-2006 13:53	3,87	175,68
	727538,24	5856775,67	25-05-2006 13:54	2,92	137,83
	727537,24	5856774,86	25-05-2006 13:54	4,54	230,99
	727580,61	5856646,96	25-05-2006 13:56	3,99	161,26
	727582,14	5856642,94	25-05-2006 13:56	1,11	159,16
	727601,35	5856635,19	25-05-2006 13:56	14,97	111,97
	727605,07	5856620,24	25-05-2006 13:56	7,9	166,02
	727612,39	5856613,23	25-05-2006 13:57	1,26	133,76
	727630,35	5856611,76	25-05-2006 13:57	4,32	94,67
	727636,32	5856604,21	25-05-2006 13:57	5,78	141,66
	727638,13	5856602,91	25-05-2006 13:57	0,73	125,68
	727630,73	5856611,54	25-05-2006 13:58	13,64	319,38
	727631,44	5856606,99	25-05-2006 13:58	5,53	171,13
	727633,17	5856607,6	25-05-2006 13:58	3,34	70,57
	727634,71	5856600,75	25-05-2006 13:58	0,87	167,32
	727702,08	5856469,64	25-05-2006 14:00	5,1	152,8
	727702,61	5856470,1	25-05-2006 14:00	0,63	49,04
	727711,72	5856444,81	25-05-2006 14:00	32,26	160,18
	727713,03	5856443,02	25-05-2006 14:00	4,03	143,8
	727719,37	5856433,82	25-05-2006 14:00	4,47	145,42
	727719,42	5856434,45	25-05-2006 14:00	0,76	4,53
	727724,43	5856420,16	25-05-2006 14:01	2,48	160,67
	727731,19	5856405,62	25-05-2006 14:01	4,43	155,06
	727736,38	5856399,02	25-05-2006 14:01	4,31	141,81
	727737,59	5856397,02	25-05-2006 14:01	4,25	148,82
	727737,11	5856393,76	25-05-2006 14:01	5,99	188,37
	727742,25	5856387,6	25-05-2006 14:02	0,8	140,15

Anexo 3: Figura 1 – *Show Data* (pedestre R263)

**Data**  
These are all the entities selected at Star.

	X	Y	Date Time	Speed	Bearing
	727484,1	5853309,57	09-08-2006 08:13	4,19	137,85
	727485,85	5853307,85	09-08-2006 08:13	2,94	134,5
	727488,04	5853304,95	09-08-2006 08:13	3,25	142,94
	727488,39	5853304,71	09-08-2006 08:13	0,77	124,43
	727488,22	5853305,54	09-08-2006 08:13	0,61	348,42
	727516,07	5853300,04	09-08-2006 08:14	2	101,17
	727516,79	5853299,46	09-08-2006 08:15	0,07	128,85
	727498,11	5853285,13	09-08-2006 08:23	0,17	232,5
	727529,1	5853298,38	09-08-2006 08:23	40,44	66,85
	727534,83	5853303,65	09-08-2006 08:24	1,47	47,39
	727533,24	5853305,22	09-08-2006 08:24	0,62	314,63
	727533,16	5853305,1	09-08-2006 08:24	0,05	213,69
	727533,11	5853302,11	09-08-2006 08:24	0,98	180,95
	727530,81	5853304,16	09-08-2006 08:24	3,7	311,71
	727524,89	5853306,73	09-08-2006 08:25	1,45	293,46
	727523,24	5853307,1	09-08-2006 08:25	1,51	282,63
	727518,82	5853313,19	09-08-2006 08:25	1,23	324,02
	727516,95	5853316,24	09-08-2006 08:25	2,59	328,48
	727515,32	5853316,83	09-08-2006 08:25	1,55	289,89
	727509,56	5853319,09	09-08-2006 08:25	2,48	291,42
	727508,05	5853320,11	09-08-2006 08:25	6,43	304,03
	727505,32	5853320,61	09-08-2006 08:25	3,33	280,37
	727501,99	5853319,96	09-08-2006 08:25	3,04	258,95
	727499,12	5853320,46	09-08-2006 08:25	5,3	279,88
	727496,26	5853324,81	09-08-2006 08:26	2,67	326,67
	727496,47	5853323,93	09-08-2006 08:26	1,64	166,57
	727495,18	5853326,82	09-08-2006 08:26	3,8	335,94

Anexo 3: Figura 2 – Show Data (pedestre R660)



Show Data

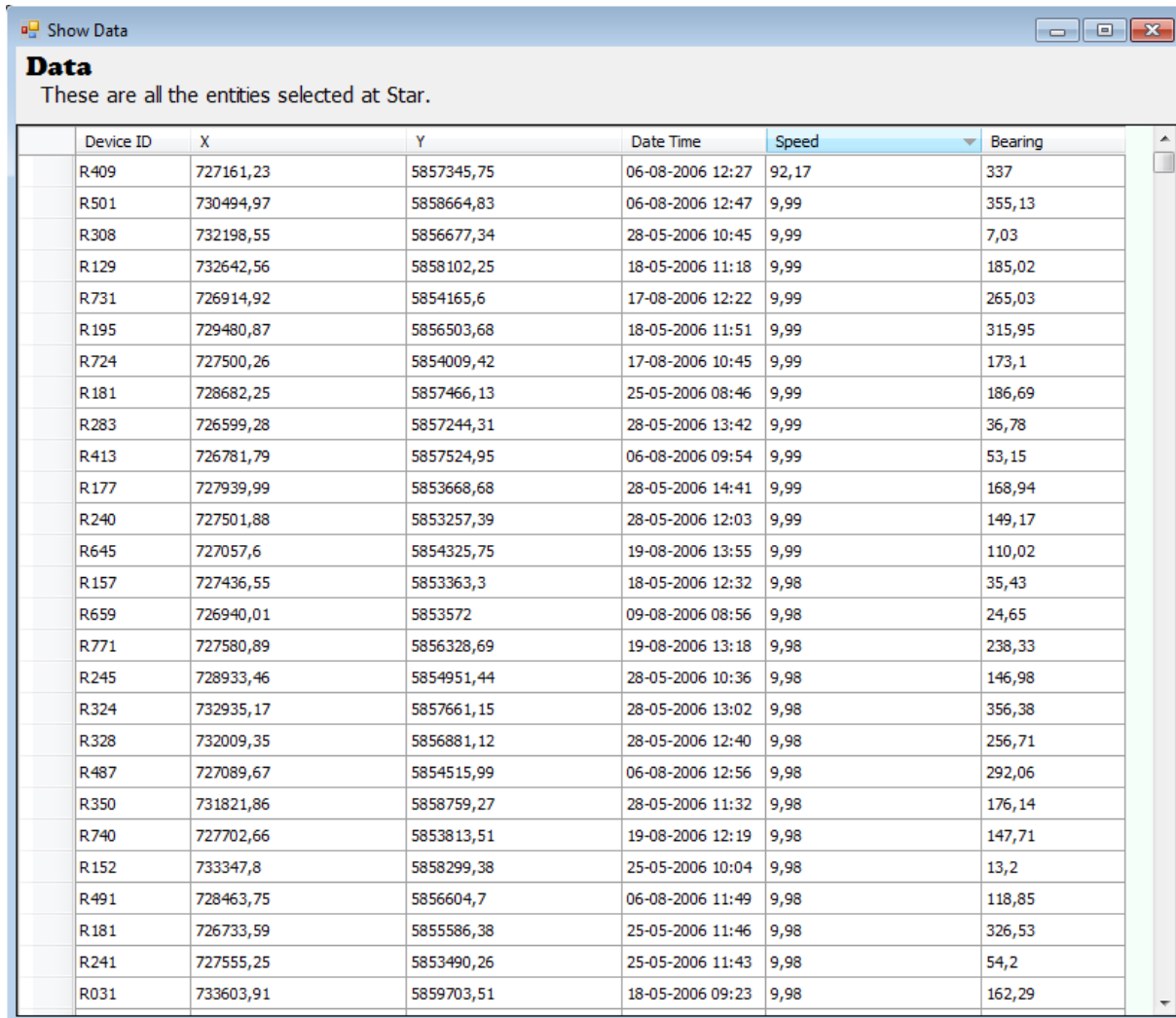
**Data**  
These are all the entities selected at Star.

	X	Y	Date Time	Speed	Bearing
	727018,9	5853250,7	09-08-2006 10:43	4,81	310,84
	727000,68	5853262,84	09-08-2006 10:43	7,18	303,67
	726996,38	5853265,51	09-08-2006 10:43	6,07	301,83
	726994,9	5853265,48	09-08-2006 10:43	5,22	268,83
	726989,92	5853267,87	09-08-2006 10:44	2,83	295,63
	726984,17	5853271,58	09-08-2006 10:44	6,13	302,83
	726967,51	5853281,36	09-08-2006 10:44	5,34	300,41
	726960,22	5853283,24	09-08-2006 10:44	4,52	284,46
	726958,5	5853284,34	09-08-2006 10:44	7,21	302,6
	726951,26	5853284,14	09-08-2006 10:44	3,27	268,41
	726948,4	5853286,41	09-08-2006 10:44	4,38	308,43
	726942,44	5853289,35	09-08-2006 10:44	4,8	296,25
	726941,15	5853288,97	09-08-2006 10:44	4,75	253,58
	726939,63	5853292,36	09-08-2006 10:44	6,75	335,84
	726933,43	5853291,31	09-08-2006 10:44	4,55	260,38
	726928,22	5853294,04	09-08-2006 10:45	2,11	297,65
	726920,34	5853294,66	09-08-2006 10:45	5,71	274,49
	726913,49	5853298,35	09-08-2006 10:45	3,99	298,31
	726910,54	5853298,8	09-08-2006 10:45	5,43	278,67
	726907	5853299,52	09-08-2006 10:45	4,33	281,49
	726903,51	5853300,36	09-08-2006 10:45	4,31	283,53
	726891,24	5853301,67	09-08-2006 10:45	1,65	276,09
	726887,96	5853302,12	09-08-2006 10:45	2,39	277,81
	726873,29	5853305,37	09-08-2006 10:46	5,4	282,49
	726870,65	5853308,37	09-08-2006 10:46	4,8	318,65
	726862,92	5853313,34	09-08-2006 10:46	4,15	302,73
	726858,48	5853322,68	09-08-2006 10:46	6,2	334,57

Anexo 3: Figura 3 – Show Data (pedestre R656)

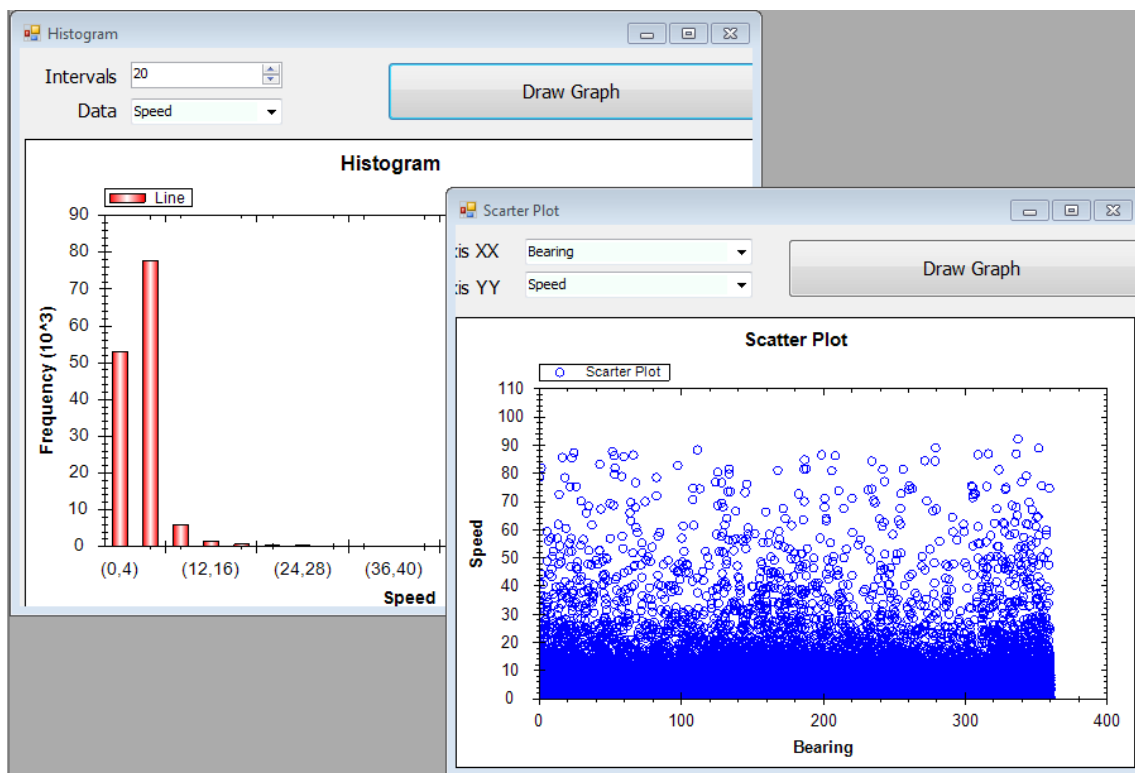
## Anexo 4 – Figuras do Sistema STAR v. 2.0

As Figuras a seguir apresentadas demonstram a correcção de dados realizada ao ficheiro *Dwingelderveld National Park*, sendo apresentados os resultados e análises de dados do ficheiro antes e depois da ocorrência da funcionalidade.



Device ID	X	Y	Date Time	Speed	Bearing
R409	727161,23	5857345,75	06-08-2006 12:27	92,17	337
R501	730494,97	5858664,83	06-08-2006 12:47	9,99	355,13
R308	732198,55	5856677,34	28-05-2006 10:45	9,99	7,03
R129	732642,56	5858102,25	18-05-2006 11:18	9,99	185,02
R731	726914,92	5854165,6	17-08-2006 12:22	9,99	265,03
R195	729480,87	5856503,68	18-05-2006 11:51	9,99	315,95
R724	727500,26	5854009,42	17-08-2006 10:45	9,99	173,1
R181	728682,25	5857466,13	25-05-2006 08:46	9,99	186,69
R283	726599,28	5857244,31	28-05-2006 13:42	9,99	36,78
R413	726781,79	5857524,95	06-08-2006 09:54	9,99	53,15
R177	727939,99	5853668,68	28-05-2006 14:41	9,99	168,94
R240	727501,88	5853257,39	28-05-2006 12:03	9,99	149,17
R645	727057,6	5854325,75	19-08-2006 13:55	9,99	110,02
R157	727436,55	5853363,3	18-05-2006 12:32	9,98	35,43
R659	726940,01	5853572	09-08-2006 08:56	9,98	24,65
R771	727580,89	5856328,69	19-08-2006 13:18	9,98	238,33
R245	728933,46	5854951,44	28-05-2006 10:36	9,98	146,98
R324	732935,17	5857661,15	28-05-2006 13:02	9,98	356,38
R328	732009,35	5856881,12	28-05-2006 12:40	9,98	256,71
R487	727089,67	5854515,99	06-08-2006 12:56	9,98	292,06
R350	731821,86	5858759,27	28-05-2006 11:32	9,98	176,14
R740	727702,66	5853813,51	19-08-2006 12:19	9,98	147,71
R152	733347,8	5858299,38	25-05-2006 10:04	9,98	13,2
R491	728463,75	5856604,7	06-08-2006 11:49	9,98	118,85
R181	726733,59	5855586,38	25-05-2006 11:46	9,98	326,53
R241	727555,25	5853490,26	25-05-2006 11:43	9,98	54,2
R031	733603,91	5859703,51	18-05-2006 09:23	9,98	162,29

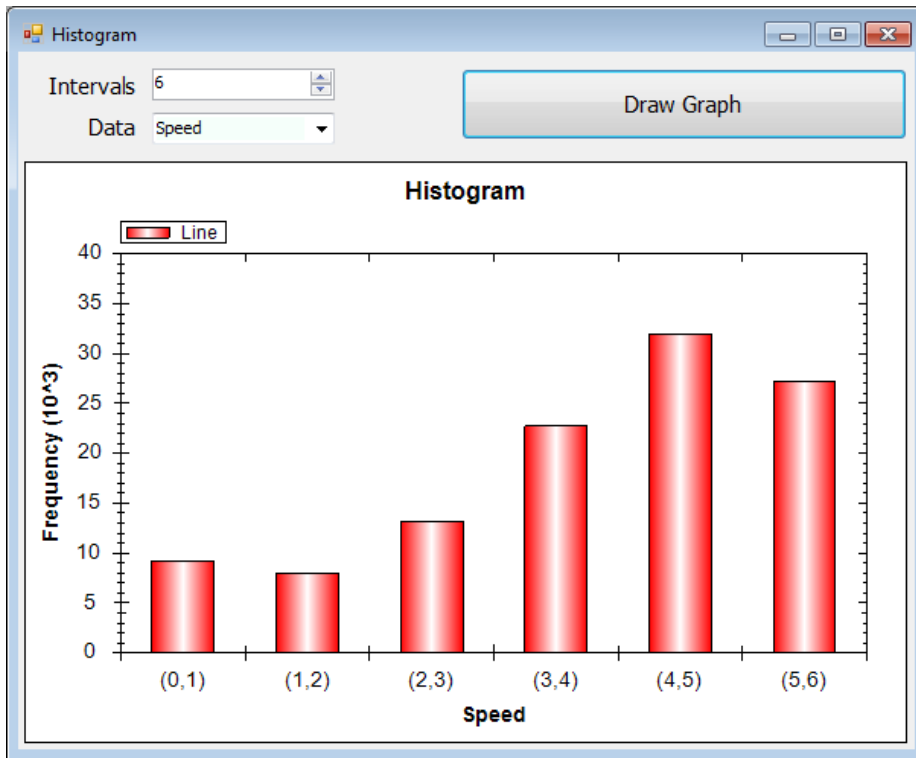
Anexo 4: Figura 1 – Show Data: Dwingelderveld National Park



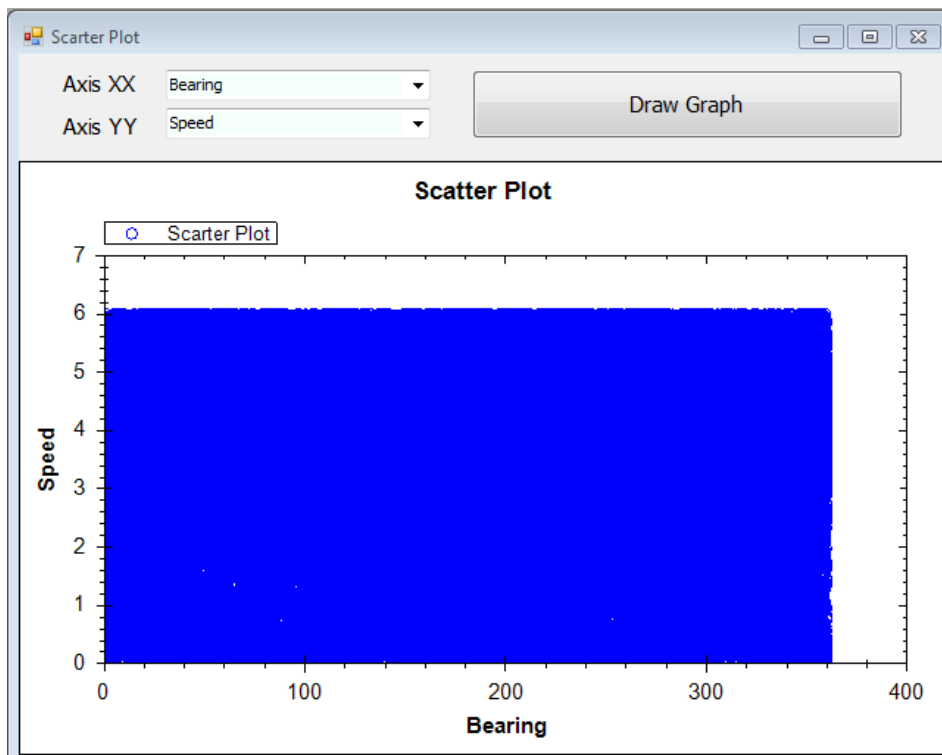
Anexo 4: Figura 1 – Histograma e Gráfico de Dispersão: *Dwingelderveld National Park*

The 'DataCorrection' window has a title bar with standard window controls. It displays the instruction: **To Correct Data you must indicate a file and choose fields**. Below this, there is a 'File Name:' label followed by a dropdown menu showing 'ALLPTS\_SHK\_CLEAN1.CSV'. There are several checkboxes and input fields for defining correction rules:   
 -  Velocity > 6 m/s   
 -  Bearing < 0 m/s   
 -  CoordX > 0   
 -  coordY > 0   
 -  Velocity < 0 m/s   
 -  Bearing > 360 m/s   
 -  CoordX < 0   
 -  CoordY < 0   
 -  Velocity = Null   
 A 'Correct Data' button is located in the bottom right corner.

Anexo 4: Figura 2 – Correção de Dados: *Dwingelderveld National Park*



Anexo 4: Figura 3 – Histograma após correção: *Dwingelderveld National Park*



Anexo 4: Figura 4 - Gráfico de Dispersão após correção: *Dwingelderveld National Park*

Show Data

**Data**  
These are all the entities selected at Star.

	Device ID	X	Y	Date Time	Speed	Bearing
	R.347	731648,04	5858060,96	28-05-2006 12:42	6	48,4
	R.135	732612,75	5858373,45	25-05-2006 13:48	6	116,44
	R.348	728711,65	5854729,72	28-05-2006 11:38	6	208,05
	R.659	727549,51	5853455,82	09-08-2006 08:36	6	229,33
	R.276	727616,38	5857574,51	28-05-2006 14:13	6	308,22
	R.792	731707,33	5858014,97	19-08-2006 10:54	6	346,74
	R.405	728316,39	5856541,15	06-08-2006 13:00	6	69,08
	R.106	727742,53	5853658,51	25-05-2006 13:41	6	217,25
	R.405	728480,46	5856593,73	06-08-2006 13:02	6	103,89
	R.726	728550,4	5854550,46	17-08-2006 12:08	6	39,12
	R.015	727375,01	5856276,98	28-05-2006 13:03	6	56,02
	R.647	726578,94	5855227,64	19-08-2006 12:59	6	226,27
	R.105	726809,22	5854930,29	25-05-2006 11:46	6	88,26
	R.034	727219,58	5853219,31	18-05-2006 11:40	6	253,7
	R.111	731650,88	5859007,26	25-05-2006 11:39	6	298,66
	R.348	728875,99	5854905,47	28-05-2006 11:35	6	223,76
	R.451	729338,89	5856639,56	06-08-2006 10:48	6	131,91
	R.304	732456,77	5856665,57	28-05-2006 09:54	6	241,2
	R.246	727171,68	5854688,78	28-05-2006 11:44	6	170,05
	R.180	727732,61	5853782,95	28-05-2006 14:17	6	139,74
	R.725	727607,65	5853908,73	17-08-2006 11:37	6	130,17
	R.235	728436,12	5854421,27	28-05-2006 14:08	6	219,6
	R.032	727651,55	5853566,92	18-05-2006 14:32	6	209,84
	R.722	727483,76	5853307,56	17-08-2006 09:28	6	318,19
	R.501	733616,29	5859121,28	06-08-2006 15:05	6	38,92
	R.444	727407,54	5853199,39	06-08-2006 10:22	6	275,29
	R.444	726941,04	5853718,2	06-08-2006 10:37	6	356,44

Anexo 4: Figura 5 –Show Data: Dwingelderveld National Park

