



**Cláudia Patrícia
Pereira Pinto Cardoso**

Sistema de Monitorização de Frotas Automóveis



**Cláudia Patrícia
Pereira Pinto Cardoso**

Sistema de Monitorização de Frotas Automóveis

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. José Manuel Matos Moreira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho a família e amigos que sempre me apoiaram, sem eles este trabalho não seria possível.

o júri

presidente

Prof. Doutor Joaquim João Estrela Silvestre Madeira
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor Alexandre Miguel Barbosa Valle de Carvalho
Professor Auxiliar da Faculdade de Engenharia da Universidade do Porto (arguente)

Prof. Doutor José Manuel Matos Moreira
Professor Auxiliar da Universidade de Aveiro (orientador)

agradecimentos

Esta dissertação não seria possível ser concretizada sem o apoio da família e amigos.

Agradeço à Posi-Sat pelo respeito e disponibilidade demonstrada.

Estou verdadeiramente agradecida ao meu Orientador Professor Doutor José Moreira que sempre se mostrou disponível e que me incentivou sempre a fazer mais e melhor.

palavras-chave

Sistema de Informação, monitorização de frotas, geo-referênciação.

resumo

Em empresas que possuem frotas automóveis a diminuição dos custos operacionais, a proteção de recursos humanos e materiais bem como a satisfação dos clientes são essenciais para manter a competitividade, tendo em vista o aumento do número de clientes e das margens de lucro. Para ajudar nestes objetivos, é essencial a gestão e monitorização das frotas, possibilitando o estudo do comportamento destas ao longo do tempo bem como obter informação atualizada do estado dos veículos.

A empresa portuguesa Posi-Sat disponibiliza um serviço de monitorização de frotas automóveis em tempo-real, que no entanto pretende melhorar uma vez que o sistema se encontra numa fase inicial de desenvolvimento. Cada veículo tem um sensor instalado que envia dados de posicionamento para o sistema, sendo disponibilizada numa Aplicação Web. Esta aplicação também permite consultar o histórico de viagens dos veículos.

Neste projeto foi adicionado um novo módulo de aquisição e transmissão de dados entre os sensores e o sistema, um módulo de agendamento de eventos cujo propósito é ajudar na gestão das obrigações com veículos, como é o caso das revisões, foi atualizada a versão usada para disponibilização do mapa na Aplicação Web e foi adicionado um módulo para criação de relatórios de frota.

Todos os pontos planeados para este projeto foram cumpridos sendo que alguns deles já estão em fase de produção.

keywords

Information sistem, tracking, vehicle fleets, georeferencing

abstract

The management of the operating costs, the protection of human and material resources and client satisfaction are essential for companies working with vehicle fleets to keep the competitive edge, in order to increase the number of clients and profit margins. One of the fundamental tasks for assisting these essential operations is the management and tracking of vehicle fleets, which allows to study the fleets performance over time and to obtain updated information about the vehicles state.

Posi-Sat is a Portuguese company that provides services for real-time tracking of vehicle fleets and desires to improve its information system that is in an early stage of development. Each vehicle has a sensor to capture and send positioning data to a central system. This data is made available in real-time through a Web Application, which also allows to keep track of historical data.

In this project it was developed a new data acquisition and transmission module for the communication between the sensors and the central system, a module for the scheduling of events related to vehicles duties, for instance the annual inspections, it was updated the version of the web mapping tool and it was developed a module to generate fleet reports.

All the planned objectives were fulfilled and some of them are already in production stage.

Índice

Capítulo 1.....	1
Introdução.....	1
Capítulo 2.....	3
Sistemas e Tecnologias para a Monitorização de Frotas Automóveis	3
1 Projetos Relacionados	3
1.1. Cartrack	3
1.2. TFleet	4
1.3. Masternaut.....	4
1.4. 3dtracking.....	5
1.5. Frotcom	6
1.6. Inosat	6
1.7. Sumário	7
2 Tecnologias	8
2.1. Dispositivos de Monitorização	8
2.1.1. Teltonika FM1100.....	9
2.1.2. Atrack AT5i.....	10
2.2. Tecnologias Java para Acesso à Rede.....	10
2.2.1. Java IO vs Java NIO.....	11
3 APIs para <i>Web Mapping</i>	14
4 Google Web Toolkit (GWT).....	16
5 Jaspersoft Studio	16
Capítulo 3.....	19
Requisitos e Arquitetura do Sistema	19
1 Sistema Inicial.....	19
2 Requisitos do Sistema	19
2.1. Levantamento de Requisitos	20
2.2. Atores	21
3 Casos de Uso.....	22
3.1. <i>Proxy</i> de Mensagens.....	22
3.2. Aplicação Web	24
4 Requisitos Não Funcionais.....	26
5 Pressupostos e Dependências	27
6 Arquitetura do Sistema.....	28
6.1. Modelo de Domínio da Base de Dados Operacional.....	28
6.2. Modelo da Base de Dados Multidimensional.....	31
7 Modelo Físico.....	33
Capítulo 4.....	37
Implementação.....	37
1 Tecnologias Usadas.....	37
2 Base de Dados Operacional.....	38
2.1. Agenda	38
2.2. Relatórios de Frota	40
2.3. Configuração de Sensores	41
2.4. <i>Proxy</i> de Mensagens – Receção de Mensagens	44
2.5. <i>Proxy</i> de Mensagens – Configuração de Sensores	47
3 Base de Dados Multidimensional.....	48

4	<i>Report Service</i>	49
5	<i>Pivot Table Service</i>	49
6	Interação com os Utilizadores	50
6.1.	Agenda de Eventos.....	50
6.2.	<i>Web Mapping</i>	52
6.3.	Tabelas Dinâmicas	53
7	Implementação de Requisitos não Funcionais	54
Capítulo 5		57
Conclusão e Trabalho Futuro		57
Referências		59
Anexo A		63
Anexo B		64
Anexo C		66
Anexo D		69

Lista de Figuras

Figura 2.1: Servidor usando Java IO [33]	12
Figura 2.2: Servidor usando Java NIO [33]	13
Figura 2.3: Arquitetura de Acesso ao <i>Socket</i> em Java NIO, adaptado de [34]	13
Figura 2.4: Aspeto da Ferramenta Jaspersoft Studio [43]	17
Figura 3.1: Diagrama de Casos de Uso do Proxy de Mensagens	23
Figura 3.2: Diagrama de Casos de Uso da Aplicação Web	24
Figura 3.3: Modelo de Domínio da Base de Dados Operacional	28
Figura 3.4: Modelo de Domínio da Base de Dados Operacional – Configuração de Sensores	30
Figura 3.5: <i>Dimensional Fact Model</i> da Base de Dados Multidimensional	32
Figura 3.6: Diagrama de Implementação	33
Figura 4.1: Diagrama de Classes Referente à Agenda	39
Figura 4.2: Diagrama de Classes Referente aos Relatórios de Frota	41
Figura 4.3: Diagrama de Classes Referente à Configuração de Sensores	43
Figura 4.4: Diagrama de Classes do <i>Proxy</i> de Mensagens - Receção	46
Figura 4.5: Diagrama de Fluxo de Pacotes de Configuração	47
Figura 4.6: Diagrama de Classes do <i>Proxy</i> de Mensagens - Envio	48
Figura 4.7: Interface com Lista de Eventos	50
Figura 4.8: Interface com Lista de Seguros dos Veículos	51
Figura 4.9: Interface para Editar Informação de Seguro	52
Figura 4.10: Interface com Visão Geral dos Veículos do Utilizador	52
Figura 4.11: Interface com Informação Relativa a uma Viagem	53
Figura 4.12: Exemplo de Tabelas Dinâmicas em Microsoft Excel	54
Figura A.1: Diagrama de Classes Completo da Base de Dados Operacional	63
Figura B.1: Ficheiro de Configuração quartz.properties do <i>Quartz</i>	64
Figura B.2: Ficheiro de Configuração quartz.properties do <i>Quartz</i>	65

Capítulo 1

Introdução

Hoje em dia vivemos num mundo altamente industrializado e globalizado em que fazer chegar a todo o lado todo o tipo de produtos e serviços é imperativo para manter a competitividade. Mesmo na era da Internet nem tudo é possível sem deslocação, portanto as empresas para o fazer têm necessidade de efetuar viagens usando frotas automóveis.

Surge assim a necessidade de monitorizar e controlar estas frotas automóveis de forma a torná-las mais eficazes e rentáveis, pelo que os sistemas de informação para a gestão de frotas automóveis são essenciais para a sustentabilidade destas empresas.

Estes serviços ao monitorizarem em tempo-real a posição dos veículos e ao disponibilizarem informação relativa ao estudo do comportamento dos condutores e veículos durante as viagens vão possibilitar que os responsáveis possam tomar decisões informadas sobre a melhor forma de aumentar eficácia e distribuição dos seus recursos. Para além disso é um grande aliado no combate ao roubo de viaturas e na procura de viaturas roubadas, no caso de acontecer o furto. Estes serviços caracterizam-se por instalar sensores nos veículos para obter dados de posicionamento, disponibilizando numa Aplicação Web ou Desktop onde é possível visualizar esta informação. Pode ainda disponibilizar funcionalidades extra como relatórios de frota e outras ferramentas de apoio à decisão.

O trabalho desenvolvido trata-se de um projeto de dissertação de mestrado que se centra num serviço de monitorização de frotas automóveis desenvolvido pela empresa Posi-Sat, que pretendiam melhor, tendo em conta que este era muito simples, apenas disponibilizava informação na Aplicação Web sobre o estado atual dos veículos e histórico de viagens.

A motivação deste projeto foi o aumento acentuado da procura de sistemas capazes de ajudar na monitorização de frotas automóveis de empresas, bem como melhorar um

sistema já existente de forma a torná-lo mais atraente para novos clientes que necessitam de funcionalidades extra que até então não estavam disponíveis. Embora seja um sistema simples destaca-se pelo seu baixo preço, o que motiva pequenas empresas ou entidades individuais a aderirem a este serviço bem como médias e grandes empresas com grandes frotas automóveis.

O objetivo deste projeto foi então aumentar funcionalidades e melhorando existentes do sistema já existente. Para isso foi exigido que a base de dados que já se encontrava no sistema não fosse alterada para se manter compatível, no entanto, o aumento de tabelas e campos às já existentes não constituía um problema.

Os pontos fundamentais a implementar são um módulo de aquisição e transmissão de dados vindos dos sensores instalados nos veículos; um módulo de agendamento de eventos cujo objetivo é alertar para a proximidade da data de eventos relacionados com os veículos, como por exemplo as datas das revisões; atualizar a versão da API usada para disponibilização do mapa na Aplicação Web; e criar um módulo para criação de relatórios de frota.

Este documento está dividido em cinco partes que representam os capítulos. No primeiro capítulo é feita a introdução do problema, onde é dada uma contextualização do mesmo e as motivações para a realização deste trabalho. No segundo é feito o estado da arte abordando as soluções já existentes no mercado e tecnologias disponíveis para desenvolver esta solução. No terceiro capítulo é feita uma abordagem aos requisitos e à arquitetura geral do sistema. No quarto capítulo é dado a conhecer a forma como foi implementada a solução e os resultados do trabalho desenvolvido. E por último, no quinto capítulo, é feita a conclusão do projeto desenvolvido e são abordados possíveis trabalhos futuros.

Capítulo 2

Sistemas e Tecnologias para a Monitorização de Frotas Automóveis

1 Projetos Relacionados

Hoje em dia os sistemas de informação para gestão de frotas automóveis são essenciais para empresas com frotas uma vez que permitem uma melhor gestão de recursos e consequentemente a diminuição de custos operacionais bem como o aumento da satisfação do cliente final.

Tendo em vista esta problemática já existem diversas soluções no mercado sendo que esta secção pretende apresentar as mais relevantes no contexto português.

1.1. Cartrack

A Cartrack é uma empresa portuguesa que surge de uma parceria com a Cartrack sul-africana que desenvolve e comercializa sistemas de gestão de segurança de veículos. Disponibiliza quatro soluções, duas delas mais focadas na segurança de veículos individuais e as outras em monitorização de frotas [1].

A *Segurança - Queek* é o produto mais simples que permite monitorização em tempo real dos veículos através do acesso a uma Aplicação Web ou na sala de controlo da própria Cartrack, usando um sistema de localização por rádio frequência, sensor de ignição, receção de alertas em casos de falta de bateria, de tentativas de manipulação do sensor instalado em cada veículo ou outras situações consideradas anormais. Por outro lado a *Segurança - Quick Executive* acrescenta à solução anterior a possibilidade de efetuar o

controle de condutores autorizados, a imobilização remota do veículo e em caso de emergência pode ser acionado o botão de pânico [2].

Por outro lado, no que toca a monitorização de frotas automóveis a Cartrack apresenta a solução *Fleet Management* que dispõe das seguintes funcionalidades: localização em tempo real através da Aplicação Web de veículos e frotas, relatórios extensos de uma ou mais viaturas contendo a informação detalhada das suas viagens e relatórios de perfil de condução, possibilita a criação e gestão de *geofences* (áreas onde a circulação é permitida), são gerados alertas em casos de deteção de anomalias como condução brusca ou falta de bateria [3].

A *Safe Fleet Management* é uma solução que também está vocacionada para gestão de frotas mas mais completa do que a anterior, acrescentando funcionalidades como imobilização remota, botão de pânico e sensor de ignição [3].

1.2. TFleet

A TFleet é uma empresa europeia que disponibiliza uma solução de gestão de frotas automóveis. Os clientes podem aceder à TFleet através da Aplicação Web onde podem visualizar a localização dos automóveis no mapa e consultar relatórios diários de atividade das viaturas e condutores, incluindo gestão financeira com registo de custos por quilómetro. O sistema envia alertas automáticos via correio eletrónico ou SMS, permite identificação do condutor e disponibiliza uma plataforma móvel com navegação ponto-a-ponto e identificação do percurso [4].

1.3. Masternaut

A Masternaut é uma empresa internacional do grupo Aeroportos de Paris que se dedica ao desenvolvimento de soluções de gestão de frotas e de operações em mobilidade, apresentando três produtos distintos que assentam numa Aplicação Web e um sistema embarcado [5].

O *Geoloc* permite localização em tempo-real dos veículos, comunicação de informação com os colaboradores e disponibiliza relatórios de atividade dos veículos [6].

O *Geonav* acrescenta à solução anterior um sistema embarcado, o *Masterlink*, que facilita a navegação por parte dos condutores através da definição de trajetos a efetuar [7]. O *Ecocan* distingue-se por ajudar a otimizar a segurança dos veículos e manutenção da frota graças à recolha de dados do CAN Bus, possibilitando estudos de consumos e quilometragens e Eco-Condução (condução com consumos de combustível baixos) [8].

1.4. 3dtracking

A 3dtracking é uma multinacional de desenvolvimento de sistemas de gestão de frotas automóveis. Esta apresenta uma solução para empresas com frotas bem como uma solução para consumidores individuais.

Para os clientes individuais o produto *3dFind* permite a consulta do posicionamento dos veículos em tempo-real através de uma Aplicação Web e lança alertas automáticos em situações em que os veículos são utilizados em alturas não autorizadas [9]. O *3dProtect* permite efetuar alertas em casos de saída de *geofences* e velocidades excessivas [10].

A solução direcionada para as empresas inclui quatro grupos de funcionalidades. O *3dTrack* que possibilita a monitorização em tempo-real da posição e do estado dos veículos e condutores, consulta dos tempos de partida e chegada das viagens efetuadas, bem como detalhes das rotas, a disponibilização de qual o veículo que está mais próximo de uma determinada localização e a marcação de pontos de interesse [11].

O *3dReport* possibilita a consulta de relatórios detalhados de viagens (início e fim de viagem, quilometragem e horas de condução), relatórios de transgressões efetuadas pelos condutores, de detalhes de atividade e de violações de segurança como entrada em zonas não autorizadas ou em alturas não permitidas [12].

O *3dAlerta* proporciona o envio de avisos em casos de emergência, de condução perigosa, de partidas ou chegadas a locais específicos e de saída de zonas autorizadas e uso dos veículos em alturas não permitidas [13].

O *3dFuel* permite que a empresa possa monitorizar os gastos de combustível através de informações de abastecimentos, de gastos e de eficiência do veículo [14].

O *3dTelematics* é um serviço que permite análise de histórico de atividade possibilitando que essa informação seja usada para seguros dos veículos cujo valor de

pagamento depende do tipo de condução dos condutores, planeamento e otimização de rotas e gestão de combustível [15].

1.5. Frotcom

A Frotcom é uma empresa Portuguesa que comercializa um sistema de localização de viaturas por GPS para gestão de frotas [16].

A solução possibilita obter informação de localização, temperatura, abertura de portas, Eco-Condução e dados de CAN Bus. Permite ainda navegação integrada com a utilização de um navegador, identificação automática de condutor e gestão de custos através do controlo dos gastos de combustível. São ainda gerados relatórios pormenorizados de viagens e consumos [17].

1.6. Inosat

A Inosat é uma empresa portuguesa especializada na produção e comercialização de soluções de localização via GPS para particulares e empresas [18].

A solução para monitorização de veículos individuais intitula-se *Car Locator* e tem as seguintes funcionalidades: o cliente recebe uma mensagem quando a ignição da viatura é ativada, a bateria é desligada ou cortada, o carro saia do local onde foi estacionado sem ter a ignição ligada, a viatura ultrapassa uma velocidade pré-estabelecida e em casos de *carjacking* o cliente pode efetuar a imobilização remota do veículo. O condutor dispõe de um botão de pânico no veículo [19].

Para empresas apresenta o *InoFrota Start* como produto mais simples que possibilita que o gestor de frota possa visualizar a qualquer momento, através de uma Aplicação Web a localização de todos os veículos, tendo ainda acesso a relatórios de posição, de rota e de alarmes [20].

O *InoFrota Trace* vem aumentar a possibilidade de obter relatórios de passagem por marcos, de atividade, de quilómetros percorridos, de utilização diária, de utilização individual, de velocidade e de balizas. Este permite ainda a identificação do condutor [21].

O *InoFrota Pro* gera relatórios de gestão técnica de frota como informação de quilómetros percorridos, prazos de inspeções de automóveis e análise de média de consumos mensais. Disponibiliza ainda relatórios de rota, de atividade, de comunicações e de anomalias [22].

O *InoFrota Navigator* integra um navegador em cada veículo para orientar os condutores durante a viagem. Este gera, para além dos já referidos nos outros produtos, relatórios de navegação, que dão informação relativa aos dados gerados pelo navegador, como por exemplo quais as ordens enviadas [23].

1.7. Sumário

Este tipo de sistemas têm características comuns sendo que podem, em geral, adaptar-se ao tipo de necessidades dos clientes, através da aquisição de diferentes pacotes. As características gerais dos sistemas de monitorização de frotas automóveis são:

- **Monitorização em Tempo Real** – Sensor instalado no veículo que envia dados relativos ao estado do veículo para o sistema central; Aplicação Web que disponibiliza num mapa onde estão identificadas as posições atuais dos veículos. Estão ainda disponíveis dados relativos à velocidade instantânea, distância percorrida até então, estado da ignição, qual o condutor do veículo e temperatura das arcas frigoríficas (em veículos que se aplique), dados estes gerados pelos sensores instalados nos veículos.
- **Geofences** – Gestão de *geofences*, sendo estas áreas que delimitam os locais onde os veículos podem circular;
- **Segurança** – Envio de alertas via correio eletrónico ou SMS em casos de falta ou corte de bateria, tentativa de manipulação do sensor instalado, saída de *geofences*, uso do veículo por condutores não autorizados e uso do veículo em horários não autorizados; disponibilização na Aplicação Web do estado das portas (abertas ou fechadas); possibilidade de efetuar a

imobilização remota de veículos em caso de *carjacking* através da Aplicação Web.

- **Relatórios** – Relatórios de viagens por veículo, frota ou condutor; relatórios de perfil de condução, incluindo práticas de Eco-Condução; relatórios de avisos (velocidades acima das autorizadas, saída de *geofences*, uso do veículo por condutores não autorizados, uso do veículo fora do horário autorizado); relatórios de gestão financeira (gastos de combustível por quilómetro, abastecimentos efetuados e eficiência do veículo tendo em conta os trajetos).
- **Navegação** – Dispositivo de navegação dedicado (como o Garmin) ou aplicação móvel para navegação ponto-a-ponto e identificação do percurso; identificação do veículo mais próximo de determinada local.

2 Tecnologias

Esta secção tem como objetivo explorar as tecnologias usadas para o desenvolvimento deste projeto.

2.1. Dispositivos de Monitorização

Todos os sistemas de monitorização de frotas automóveis requerem um dispositivo instalado nos veículos ou que pelo menos esteja presente aquando das viagens. Este dispositivo é o responsável por fornecer dados remotamente sobre a localização do veículo a cada instante, bem como outras informações suplementares (temperatura, velocidade, distância percorrida, gastos de combustível, etc.). Estes podem ainda servir como fornecedor de informação ao veículo como no caso em que é usado para imobilização do veículo em situação de roubo.

Associado a este dispositivo podem estar outros para aumentar as funcionalidades presentes no sistema como é o caso do *iButton* [24] usado para identificar o condutor que está a conduzir o veículo.

De seguida são apresentados os dispositivos usados no sistema disponibilizado pela Posi-Sat.

2.1.1. Teltonika FM1100

Trata-se do dispositivo que está a ser usado desde que o serviço foi para o mercado, em Junho de 2012.

Este dispositivo envia mensagens com informação GPS do veículo através da rede GPRS e SMS. O acesso à rede é realizado através de um cartão SIM, sendo que não está limitado à rede local, podendo efetuar *roaming*. Para além de dados de posicionamento pode enviar informação relativa ao instante de tempo em que a mensagem foi enviada, distância percorrida desde a última mensagem e estado de ignição do veículo. Possui entradas e saídas digitais de dados, uma saída analógica e uma entrada para *iButton*. Possibilita que sejam definidas cinco *geofences* e que, no caso de não ter rede GPRS disponível, as mensagens sejam guardadas e enviadas assim que possível. Possui ainda estados para poupança de energia nos casos em que veículo não está com o motor ligado [25].

As mensagens com a informação de posicionamento e estado do veículo obedece a um protocolo proprietário, sendo que a Teltonika disponibiliza uma implementação em Java para descodificação destas mensagens a entidades que adquiram o equipamento. Importa referir que no caso de haver informação acumulada (quando não há acesso à rede GPRS) as mensagens são enviadas uma a uma assim que possível e é dada prioridade a mensagens produzidas no momento, isto é, se houver dez mensagens antigas e uma referente ao instante atual, na fila, a do instante atual é a primeira a ser enviada e só depois são enviadas as outras dez, começando pela mais antiga [26].

No entanto, a Posi-Sat não está satisfeita, pois o sensor tem-se mostrado pouco robusto uma vez que deixa facilmente de funcionar sendo necessário reiniciá-lo ou mesmo substituir. Esta situação é muito desagradável tanto para a Posi-Sat, uma vez que o

equipamento tem dois anos de garantia, mas também para o cliente porque necessita de um dispositivo fiável que esteja constantemente em funcionamento e que não cause perdas de serviço.

2.1.2. Atrack AT5i

Este é o novo sensor disponibilizado pela Posi-Sat para resolver os problemas que o sensor da Teltonika tem causado. Este, para além de o fabricante garantir a sua fiabilidade tem a vantagem de, se for comprado em grandes quantidades, ficar mais barato por unidade.

O sensor para além da fiabilidade anunciada, faz cifra das mensagens, isto é, existe a possibilidade das mensagens trocadas na rede estarem cifradas usando AES-128 (Advanced Encryption Standard 128 bits) [27], cada mensagem tem ainda um campo CRC16 (Cyclic Redundancy Check 16 bits) para deteção de erros que possam ocorrer durante a transmissão dos dados [28][29].

Este sensor para além das mensagens normais de posição tem as *multiple position message* que conseguem enviar mais do que uma informação de posicionamento, pelo que as mensagens armazenadas que não podem ser enviadas por falta de rede, quando houver rede disponível novamente, não é necessário enviar uma a uma. Desta forma, o processo de restabelecimento da normalidade é mais rápido [29]. Este sensor é ainda compatível com o sistema de navegação Garmin FMI 2.0 [30] e suporta *iButton*.

O protocolo usado para comunicação com o dispositivo Atrack AT5i é proprietário da ATrack Technology Inc [29].

2.2. Tecnologias Java para Acesso à Rede

A utilização de Java como plataforma para programação em redes é uma boa opção uma vez que:

- É compatível com múltiplas plataformas [31].

- Disponibiliza um modelo “leve” de concorrência o que é atrativo para desenvolvimento de servidores que façam uso de *threads* [31].
- O ambiente é considerado seguro tornando as aplicações pouco suscetíveis a falhas totais do sistema: não usa ponteiros o que impede problemas relacionado com referências [31]; as exceções e erros podem ser capturados [31]; quando o programa está a correr se ocorrerem erros ou exceções não capturados apenas são fatais para a *thread* onde ocorrem [31]; ao ser um ambiente de execução isolado cria uma barreira permitindo apenas acesso aos recursos dentro do próprio domínio [32]; o ambiente de execução implementa mecanismos de segurança, os *Byte Code Verifier*, *Class Loader* e *Security Manager* que juntos efetuam verificações no acesso a recursos como o sistema de ficheiros e a rede [32].

Nesta dissertação só é feita referência à parte de acesso aos *sockets* TCP (Transmission Control Protocol) das bibliotecas disponibilizadas pelo Java (Java NIO e Java IO), no entanto estas podem ser usadas para outras operações de IO (entrada e saída de dados) como é o caso do acesso a ficheiros.

2.2.1. Java IO vs Java NIO

A biblioteca Java IO assenta no paradigma de uma *thread* por cliente do lado do servidor, isto é, uma *thread* por cada conexão estabelecida cliente-servidor, como é possível observar na Figura 2.1 [31].

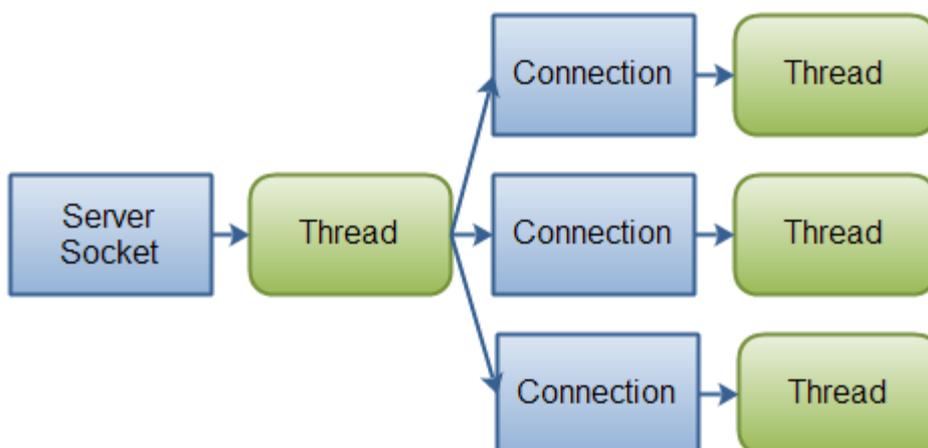


Figura 2.1: Servidor usando Java IO [33]

As operações de leitura e escrita são feitas de uma forma bloqueante, isto é, a *thread* que invoca a operação fica bloqueada até ter dados para ler ou até ter tudo escrito, respectivamente [31].

Existe a necessidade de uma forma mais escalável de acesso aos *sockets* uma vez que este paradigma, com o crescimento do número de clientes, torna-se insustentável, pois a memória para alocar *threads* numa máquina é finita. Para além disso, o número de *threads* que podem estar em execução ao mesmo tempo depende do número de núcleos do processador e o SO (Sistema Operativo), ao tentar atribuir o mesmo tempo de serviço a todas as *threads*, vai levar a que o processador esteja constantemente em mudança de contexto, agravando com o aumento do número de *threads* não bloqueadas. Isto leva a uma diminuição da qualidade de serviço na medida em que o tempo para atender os pedidos pode ser muito elevado ou pode mesmo provocar a queda de ligações por falta de resposta em tempo útil [31].

Para contornar este problema existe a biblioteca Java NIO que tem apenas uma *threads*, para atender todas as conexões (clientes), como é possível verificar na Figura 2.2. Podem ser usadas mais, mas o caso mais habitual é apenas uma.

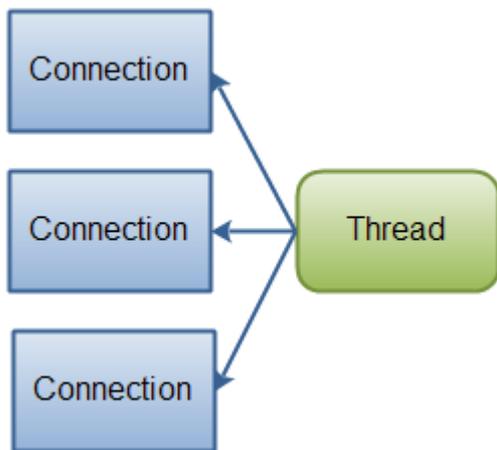


Figura 2.2: Servidor usando Java NIO [33]

A Figura 2.3 mostra como é feito o acesso ao *socket* usando Java NIO. Existe um seletor (*Selector*) que vai possibilitar que uma única *thread* possa monitorizar vários canais (*SocketChannel*), uma vez que este vai ativar uma *SelectionKey*, que é colocada no conjunto das *Keys*, de cada vez que existe um canal com conteúdo disponível para ler e o mesmo acontece na escrita, que é ativada uma *SelectionKey* para cada canal pronto para receber dados [31].

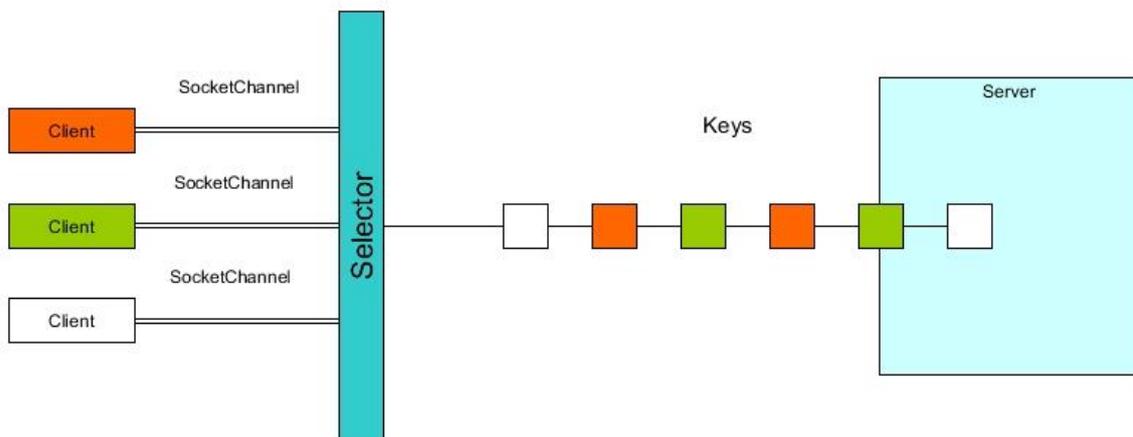


Figura 2.3: Arquitetura de Acesso ao *Socket* em Java NIO, adaptado de [34]

Usando o Java NIO de uma forma não bloqueante, quando uma *thread* solicita leitura esta apenas retira os bytes de dados disponíveis (ou nada se não houver conteúdo), não ficando bloqueada até que haja mais conteúdo disponível. O mesmo acontece com a

escrita em que uma *thread* pode pedir para escrever no *channel*, no entanto, não espera até que a operação fique concluída na totalidade. Assim, o sistema torna-se assíncrono o que pode levar a algumas dificuldades acrescidas a nível da programação [31], quando comparado com o Java IO.

De acordo com [33] o NIO é a opção mais indicada quando é necessário centenas de conexões em simultâneo e são enviados poucos de dados de cada vez, caso contrário a complexidade acrescida do NIO pode não compensar.

3 APIs para *Web Mapping*

Existem diversas APIs para disponibilização de mapas em Aplicações Web, como são os casos da Google Maps Javascript API e da OpenLayers. Estas são distintas no que toca a funcionalidades e termos de utilização, sendo estes últimos o ponto que mais as distingue.

O Google Maps disponibiliza a API livremente, no entanto, se a aplicação tiver mais do que vinte e cinco mil *map loads* durante noventa dias consecutivos a Google entra em contacto com os responsáveis para encontrarem uma solução: reduzir utilização, comprar usando a API de pagamento do Google de cada vez que o limite é atingido ou comprar uma licença de utilização [35]. Para além disso a Google define que as aplicações que usem esta API devem poder ser usadas livremente, sem qualquer tipo de conta paga para aceder [36].

Esta API possibilita disponibilizar mapas dinâmicos na Web, colocar marcadores em pontos de interesse e traçar percursos no mapa, personalizar o mapa e marcadores, visualização panorâmica de ruas (*Street View*) e visualização por satélite de alguns locais do globo [37].

Recentemente a API foi atualizada da versão dois para a versão três sendo que a Google deixou de dar suporte à versão mais antiga [38]. As principais alterações foram as seguintes:

- A maioria das funções suplementares foram migradas para bibliotecas [39];
- A performance de algumas funcionalidades foi melhorada, como por exemplo a disponibilização de marcadores no mapa [39];

- Passou a dar suporte ao padrão Modelo-Visão-Controlador [39];
- Passou a dar suporte a vários *browsers* modernos e deixou de dar suporte ao Internet Explorer 6 [39];
- Foi adicionada uma implementação de HTML5 para visão de rua (Street View) [39];
- Possibilita a customização dos mapas [39];
- Dá suporte a novos serviços como por exemplo Weather Layer [39];
- O sistema de navegação foi melhorado facultando funcionalidades como rotas alternativas, otimização de rotas e indicações para quem usa bicicleta [39].

Por sua vez, a Open Source Geospatial Foundation fornece a API OpenLayers para disponibilização de mapas dinâmicos e marcadores em Aplicações Web, sendo que para funcionalidades adicionais têm que ser acrescentadas bibliotecas. Esta é uma API em JavaScript completamente livre (*Open Source*) [40].

De notar que existem outras APIs para disponibilização de mapas na Web, no entanto não foram exploradas durante este trabalho.

4 Google Web Toolkit (GWT)

O GWT é uma ferramenta que permite o desenvolvimento de aplicações Ajax sofisticadas de uma forma fiável e simples. Uma aplicação em GWT é escrita usando a linguagem Java, no entanto esta é depois compilada para JavaScript, e é executada num *browser*, ou seja, do lado do cliente. O facto de existirem muito mais pessoas a programar bem Java do que JavaScript é um ótimo motivo para se usar GWT, no entanto, o principal motivo para o surgimento desta ferramenta foi facilitar o desenvolvimento de Aplicações Web complexas do lado do cliente. Providencia ainda facilidades como um conjunto completo de *widgets*, suporte a RPC (Remote Procedure Call), internacionalização intrínseca, integração com a camada de persistência e bibliotecas Open Source. Neste momento é uma ferramenta madura e estável, tendo em conta que já está disponível desde 2006 e tem vindo sempre a evoluir [41].

A programação em JavaScript tem o seu lado complicado uma vez que cada *browser* tem as suas próprias peculiaridades. Assim, para resolver este problema, o GWT disponibiliza *deferred bindings* que permite criar várias implementações da mesma funcionalidade para que seja compatível com os *browsers* pretendidos [41].

Tendo em conta que o código gerado para JavaScript corre do lado do cliente o GWT disponibiliza a *framework* RPC a fim de facilitar a troca de objetos Java entre o cliente e o servidor da Aplicação Web, usando HTTP (Hypertext Transfer Protocol). A parte do servidor que é invocada pelo cliente é normalmente chamada de serviço e a implementação do serviço RPC é baseada na arquitetura *Java Servlet* [42].

5 Jaspersoft Studio

O Jaspersoft Studio é uma ferramenta baseada em Eclipse para JasperReports e JasperReports Server. Esta permite que sejam criados relatórios com informação disposta em vários formatos como tabelas, gráficos, imagens e sub-relatórios. Possibilita o acesso a dados através de JDBC (Java Database Connectivity), TableModels, JavaBeans, XML, Hibernate, CSV e fontes customizadas. Os relatórios podem ser gerados em formato PDF, RTF, XML, XLS, CSV, HTML, texto, DOCX ou OpenOffice [43].

A Figura 2.4 mostra o interface de desenvolvimento de relatórios da ferramenta.

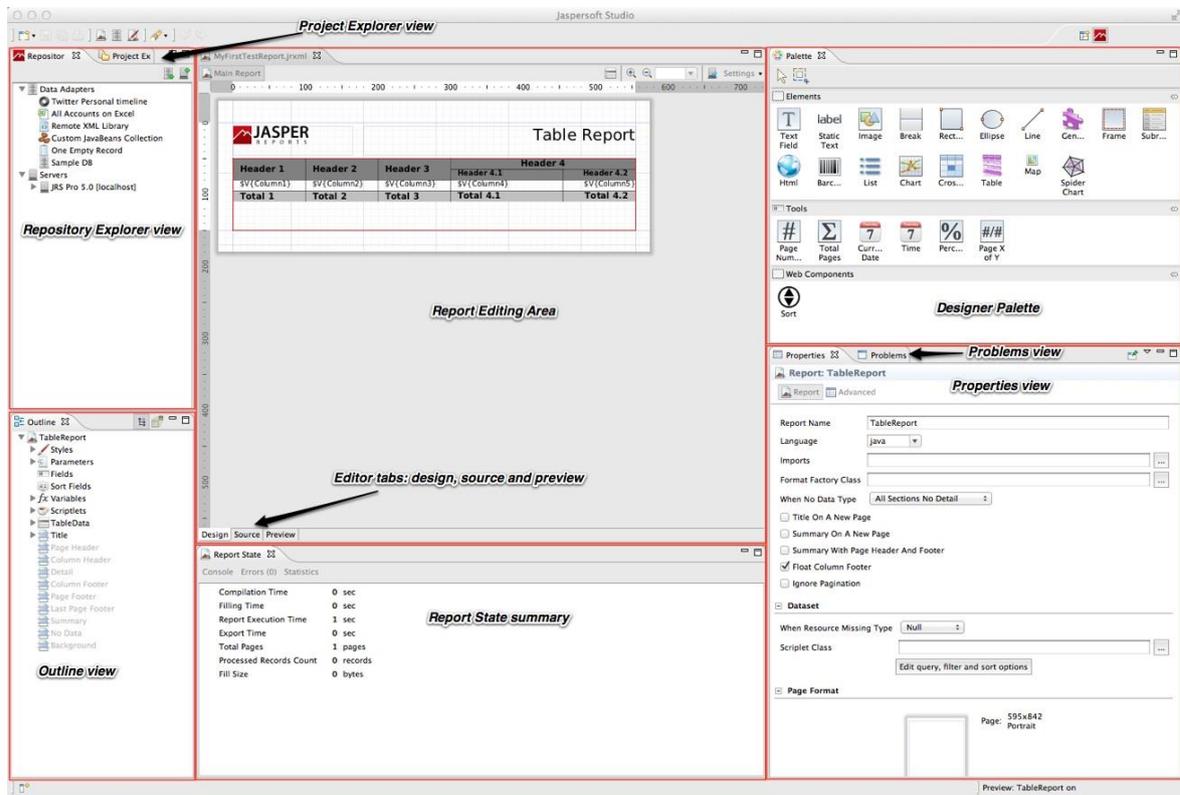


Figura 2.4: Aspeto da Ferramenta JasperSoft Studio [43]

Quando pretendemos criar um relatório usando esta ferramenta é criado um ficheiro JRXML que se trata de um ficheiro XML que contém a definição do aspeto do relatório. A manipulação do aspeto pode ser feita visualmente pelo que não é obrigatório perceber a estrutura do ficheiro XML. E para produzir o relatório basta passar o ficheiro JRXML e a fonte de dados para o JasperReport [43].

Capítulo 3

Requisitos e Arquitetura do Sistema

1 Sistema Inicial

No início do projeto, o sistema estava dividido em três camadas: aquisição de dados, persistência e apresentação. A camada de aquisição de dados funcionava como um servidor que recebe as mensagens enviadas pelos sensores instalados nos veículos. Cada mensagem transportava dados relativos à localização, velocidade, distância percorrida até então e o estado da ignição do veículo, sendo estas mensagens recebidas de um em um minuto quando o veículo se encontra em Portugal e de dez em dez minutos quando o veículo se encontra fora de Portugal.

A camada de persistência era a camada responsável por guardar os dados enviados pelos sensores bem como os dados relativos à gestão de utilizadores.

A camada de apresentação consistia numa Aplicação Web onde os clientes podiam visualizar informação relativa aos veículos das suas frotas: posição geográfica atual, velocidade atual e histórico de viagens incluindo data de início, data de fim, morada inicial, morada final, duração e distância percorrida.

2 Requisitos do Sistema

Esta secção apresenta a especificação dos requisitos do sistema que constituiu a primeira fase do desenvolvimento da nova versão do sistema. É feito o levantamento de requisitos, classificação de atores e descrição de caso de uso.

2.1. Levantamento de Requisitos

O processo de especificação de requisitos baseou-se em entrevistas aos responsáveis da Posi-Sat bem como a alguns dos clientes da empresa, tendo como objetivo avaliar as necessidades existentes. Também foi feito um levantamento de projetos similares, tecnologias mais usadas para este tipo de soluções e estudos efetuados na área. Assim, esta pesquisa ajudou a definir de uma forma mais clara os requisitos e quais as tecnologias a usar. No final deste processo foram identificados requisitos que a seguir se apresentam.

A Posi-Sat pretende instalar sensores mais robustos e com mais funcionalidades nos veículos dos novos clientes ou em clientes antigos que assim o pretendam. Para tal, é necessário contruir um servidor capaz de receber mensagens e armazenar dados vindos de ambos os tipos de sensores (os antigos e os mais recentes). O servidor deve ainda poder configurar remotamente os novos sensores, isto é, em caso de anomalia ou necessidade de alterar algumas das configurações do sensor.

A Posi-Sat pretende atualizar a versão da API utilizada para disponibilização do mapa na página web. Considerou-se a possibilidade de alterar a API usada (Google Maps API V3) para uma API *open-source*, no entanto, os responsáveis da empresa optaram por apenas alterar a versão utilizada tendo em conta que estavam satisfeitos, pois o único problema que identificaram foi o facto de a V2 já não ser suportada [38] e não despender recursos nesta altura.

Alguns clientes mencionaram a necessidade de associar ao sistema as obrigações relativas a revisões, pagamento de impostos e seguros, para que toda a informação relacionada com os veículos estivesse disponível numa única aplicação. Para tal, é necessário implementar uma agenda para registar os eventos associados a cada veículo e avisar o cliente para a aproximação da data do evento, tanto na Aplicação Web, como via correio eletrónico e SMS.

Os clientes devem receber no correio eletrónico relatórios diários, semanais ou mensais detalhados com informação referente às viagens efetuadas por cada veículo e condutor, destacando a hora e local de partida e chegada, tempo de condução, distância percorrida, velocidade média e velocidade máxima de cada viagem. O sistema também deve permitir elaborar relatórios detalhados com as posições geográficas de um

determinado veículo num espaço de tempo, apresentando para cada posição, o instante de tempo em que ocorreu, o local (endereço e coordenada) e a velocidade instantânea.

Para além da disponibilização dos relatórios com informação pré-definida e estática, sugerimos que os clientes tenham acesso aos dados sobre as viagens efetuadas pelas viaturas da sua frota de forma a poderem organizá-la usando tabelas dinâmicas, com o objetivo de obterem informação útil para a empresa ou pessoa individual.

2.2. Atores

Os utilizadores alvo da Aplicação Web podem ser pessoas de qualquer faixa etária desde que possuam uma conta. No entanto, é de esperar que como a Aplicação Web está associada a frotas de veículos os utilizadores sejam maioritariamente adultos com idades compreendidas entre os vinte e os sessenta e cinco anos, pelo que a Aplicação Web tem que estar preparada para utilizadores com diferentes níveis de experiência de navegação na internet.

Por outro lado, os utilizadores que têm acesso à configuração de dispositivos são um número reduzido e limitado, tendo em conta que esta parte do sistema se destina aos administradores da Posi-sat, pelo que o nível de experiência espectral é bastante elevado.

Assim, os atores identificados são os descritos na lista seguinte:

- **Utilizador Básico:** Este ator, depois de efetuar o login, tem acesso à Aplicação Web podendo visualizar as posições dos veículos, em tempo real, o histórico de posições e viagens.
- **Utilizador Autorizado:** Este utilizador pode visualizar o mesmo que o utilizador básico, podendo ainda aceder à agenda com os avisos relativos a revisões, seguros, complementos e impostos associados aos veículos e relatórios de frota, incluindo as tabelas dinâmicas.
- **Administrador Posi-Sat:** Este ator tem acesso à aplicação de configuração dos dispositivos, podendo visualizar o estado de atividade de cada um dos sensores instalados e enviar comandos de configuração.

- **Sensor – Teltonika FM1100:** Este ator representa os sensores mais antigos, isto é, sensores da Teltonika que já estavam instalados nos veículos antes do início deste projeto. Assim, este ator vai enviar mensagens para o servidor de recepção e envio de mensagens com informação relativa às posições dos veículos ao longo do tempo.
- **Sensor – Atrack AT5i:** Este ator representa os sensores da Atrack, sendo estes os mais recentes. Quando instalados nos veículos, enviam mensagens para o servidor de recepção e envio de mensagens, tal como os sensores Teltonika FM1100. Estes podem ainda receber mensagens de configuração remota.

3 Casos de Uso

O diagrama de casos de uso foi dividido em duas partes representando o sistema de recepção de mensagens e configuração dos sensores (*Proxy* de Mensagens), e por outro lado a interação dos utilizadores com a aplicação (Aplicação Web).

3.1. *Proxy* de Mensagens

A Figura 3.1 representa o diagrama de casos de uso referente à parte do sistema de recepção e envio de mensagens pelos sensores. Os atores que interagem com o sistema são os sensores instalados nos veículos e os administradores da Posi-Sat. O sensor Teltonika FM1100 apenas envia mensagens com as várias posições do veículo, enquanto o sensor Atrack AT5i também tem a possibilidade de ser configurado remotamente pelo administrador do sistema.

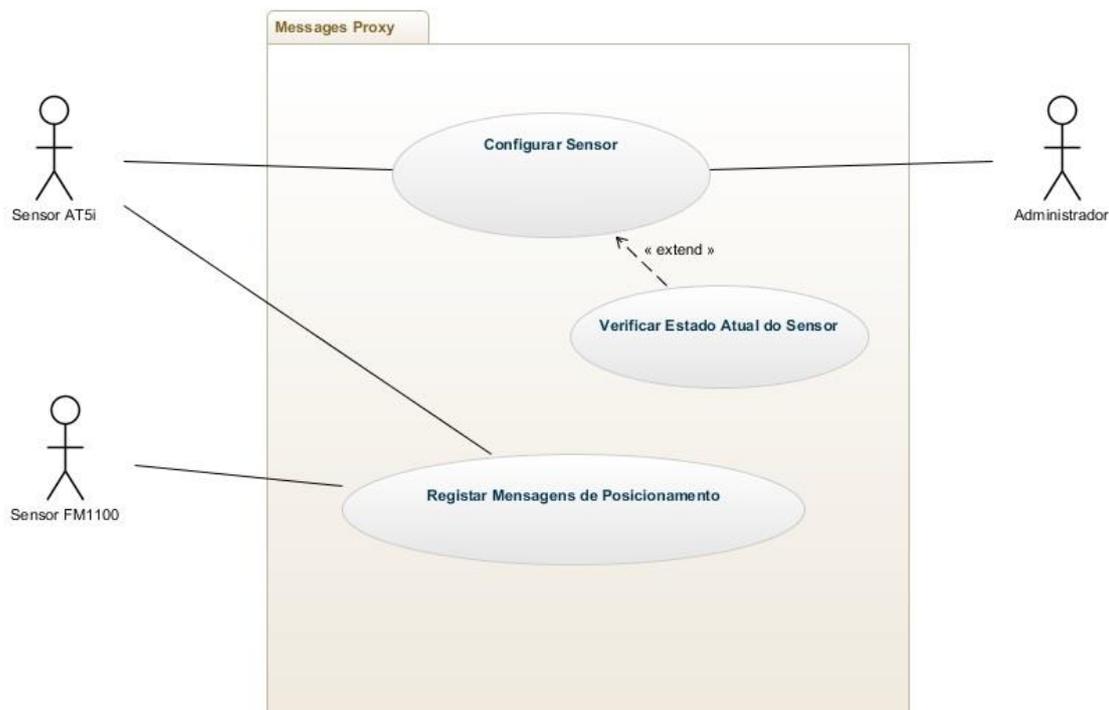


Figura 3.1: Diagrama de Casos de Uso do Proxy de Mensagens

- **Registrar Mensagem de Posicionamento:** Permite que o sistema receba mensagens com dados relativos ao estado do veículo em cada instante. Estas mensagens diferem no formato e tipos de dados enviados consoante o tipo de sensor.

Prioridade: Elevada.

- **Configurar Sensor:** O administrador tem a possibilidade de enviar comandos de configuração para os sensores do tipo Atrack AT5i. Estes comandos podem servir para tentar recuperar de anomalias ou para alterar definições dos sensores. O utilizador tem ainda a possibilidade de verificar o estado atual de cada sensor.

Prioridade: Baixa.

3.2. Aplicação Web

A Figura 3.2 representa o diagrama de casos de uso referente à Aplicação Web disponível para os utilizadores.

Esta parte do sistema permite que o utilizador tenha acesso a informação relativa às várias posições dos veículos e configurar opções de agenda e relatórios.

Seguidamente é feita uma descrição detalhada de cada caso de uso.

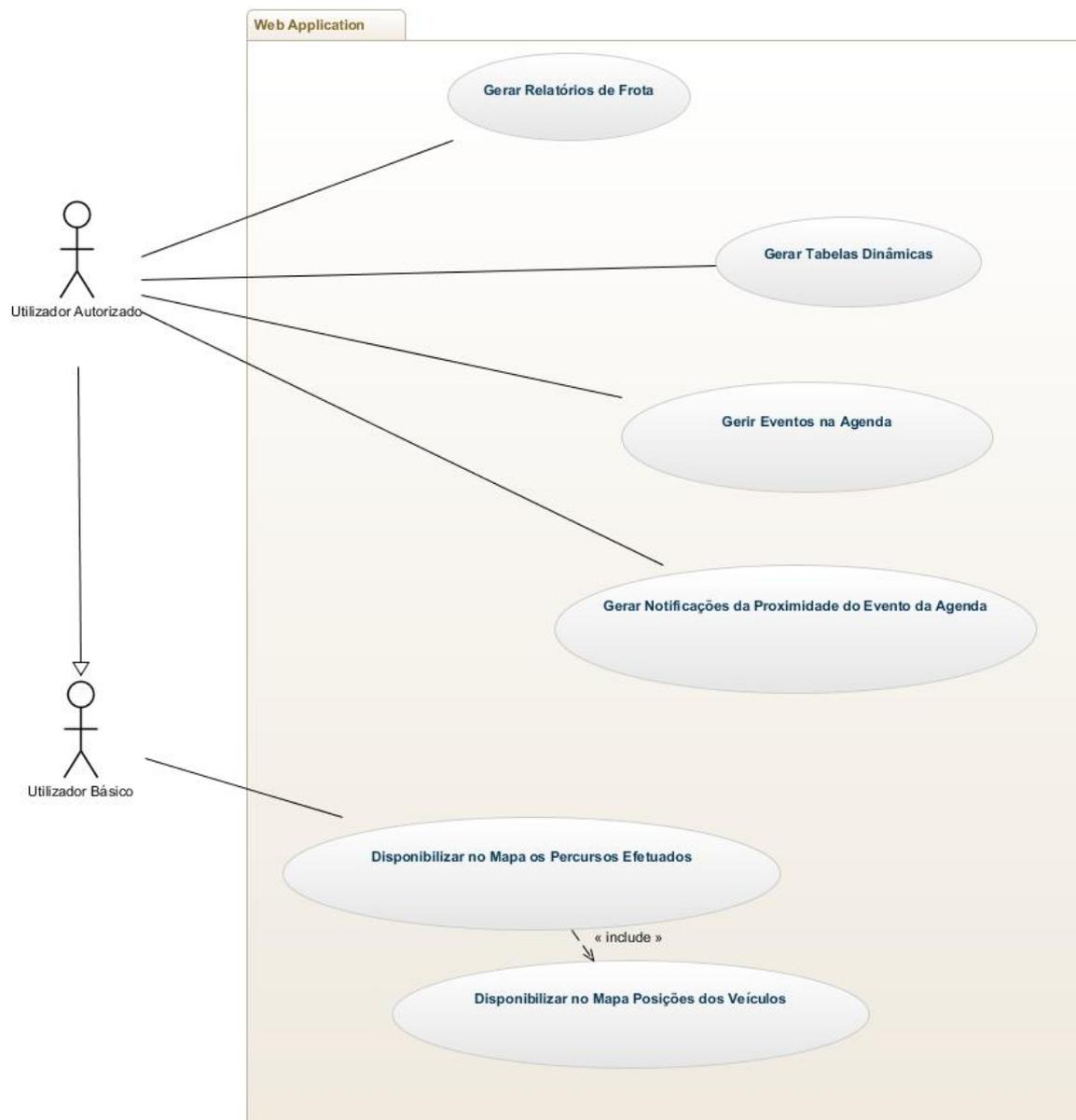


Figura 3.2:Diagrama de Casos de Uso da Aplicação Web

- **Gerir Eventos na Agenda:** Permite que o utilizador autorizado possa adicionar, remover e editar eventos, referentes aos veículos, na agenda.

Prioridade: Elevada.

- **Gerar Notificações da Proximidade do Evento da Agenda:** O utilizador autorizado pode pedir para ser avisado, via correio eletrónico ou SMS, para a proximidade da data do evento.

Prioridade: Elevada.

- **Disponibilizar no Mapa os Percursos Efetuados:** O utilizador tem acesso a um mapa na Aplicação Web onde pode visualizar em tempo real as posições dos vários veículos de cada uma das suas frotas. Estão também disponíveis os percursos efetuados por cada um dos veículos da sua frota, podendo ter acesso a informação relativa às posições ao longo do tempo de cada percurso.

Prioridade: Elevada.

- **Gerar Relatórios de Frota:** Permite que um utilizador autorizado tenha acesso a relatórios de frota com informação relativa a um intervalo de tempo definido. Pode ainda pedir que lhe sejam enviados diariamente, semanalmente ou mensalmente relatórios via correio eletrónico. Estes relatórios podem ser gerados em formato PDF (Portable Document Format) ou XLS (Microsoft Excel).

Os tipos de relatórios são:

- **Relatório de Viagens:** Contém informação relativa às viagens completas: hora de início e fim, morada de início e fim, velocidade média, velocidade máxima, distância percorrida e consumo médio de combustível.

Prioridade: Média.

- **Relatório de Posições:** Contém informação relativa às posições do veículo ao longo do tempo: hora, morada, longitude, latitude e velocidade instantânea.
Prioridade: Média.

- **Gerar Tabelas Dinâmicas:** Possibilita que o utilizador autorizado tenha acesso dados relativos a viagens efetuadas pelas suas frota, dentro de um intervalo de tempo definido, podendo organizá-la em tabelas dinâmicas.
Prioridade: Baixa.

4 Requisitos Não Funcionais

A lista seguinte apresenta os requisitos não funcionais do sistema, nomeadamente, a descrição de características que este deve exibir ou restrições que deve respeitar.

- **Usabilidade:** Tendo em conta que a Aplicação Web é usada por pessoas com idades variadas e níveis de experiência muito distintos é importante que esta seja fácil de usar e aprender.
Prioridade: Elevada.

- **Fiabilidade:** Tendo em conta que o sistema lida com informação em tempo real é importante que este esteja constantemente disponível. Ou seja, é fulcral que o número de falhas seja muito reduzido.
Prioridade: Elevada.

- **Segurança:** As mensagens enviadas pelos sensores devem estar protegidas para não sofrerem ataques de *Packet Sniffing* e *Man in the Middle*, porque é importante para os clientes manterem a privacidade das suas operações (neste caso das viagens) e para que não sejam induzidos em erro relativamente às posições dos veículos.

Prioridade: Média.

- **Interoperabilidade:** O sistema deve ser compatível com ambos os protocolos dos sensores, tanto o mais antigo como o mais recente.

Prioridade: Elevada.

- **Portabilidade:** Tendo em conta que a Aplicação Web vai ser usada por diversas pessoas com *browsers* distintos é importante que a aplicação tenha o mesmo comportamento em cada um deles.

Prioridade: Elevada.

- **Extensibilidade:** Tendo em conta que são usados dois tipos de sensores e que existe a possibilidade de serem acrescentados mais, é importante que o sistema esteja preparado de modo a que nesses casos a integração de um novo protocolo ou a receção de novos tipos de dados seja de fácil implementação.

Prioridade: Média.

5 Pressupostos e Dependências

Para que os sensores enviem mensagens e sejam configurados é necessário que estejam numa área com rede GPRS. Por outro lado, para aceder à Aplicação Web é necessário ter ligação à internet e um *browser* instalado.

Apenas as mensagens enviadas pelos sensores Atrack AT5i são cifradas usando AES-128 para que apenas quem tenha a palavra-passe possa visualizar o conteúdo da mensagem e usam CRC [29] para deteção de erros nas mensagens durante a sua transmissão que as possa corromper. Os sensores mais antigos (Teltonika FM1100) não implementam qualquer mecanismo de cifra nem de verificação de erros [26].

6 Arquitetura do Sistema

Esta secção apresenta uma visão geral da arquitetura do sistema, descrevendo o modelo de domínio da Base de Dados Operacional, o modelo da Base de Dados Multidimensional e as relações subjacentes e ainda o modelo físico do sistema.

6.1. Modelo de Domínio da Base de Dados Operacional

A Figura 3.3 mostra o modelo de domínio que representa a Base de Dados Operacional do sistema. Nesta figura a azul estão as entidades novas (acrescentadas no âmbito deste projeto) e a branco as entidades que se mantiveram inalteradas.

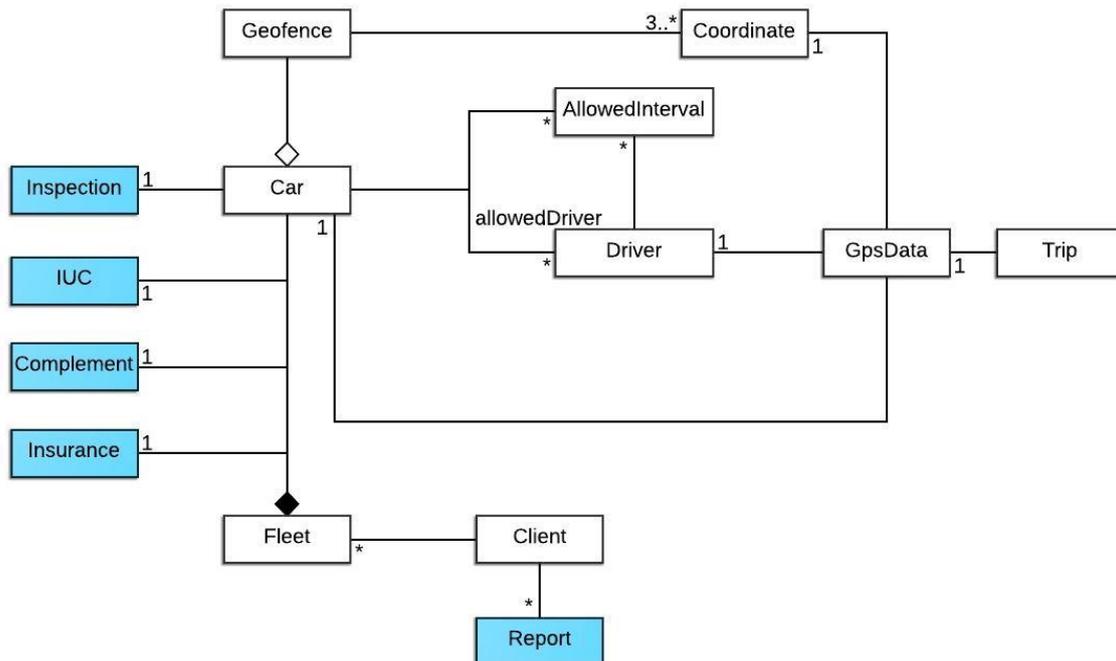


Figura 3.3: Modelo de Domínio da Base de Dados Operacional

A entidade *Client* representa os clientes que têm acesso à Aplicação Web. Cada cliente tem um conjunto de frotas, representadas pela entidade *Fleet*, e cada uma dessas frotas tem um conjunto de veículos, representados pela entidade *Car*.

Os clientes podem definir quais as áreas em que um determinado veículo pode circular. Para tal, é usada a entidade *Geofence* que é delimitada por pelo menos três coordenadas, representadas pela entidade *Coordinate*. Os veículos têm ainda intervalos de tempo em que o seu uso é permitido, representados pela entidade *AllowedInterval*, e condutores autorizados, representados pela entidade *Driver*. Estes condutores autorizados podem ser destacados para intervalos de tempo específicos, isto é, apenas serem autorizados em determinados intervalos de tempo (*AllowedInterval*).

Os veículos vão enviar dados relativos às várias posições que vão possuindo, sendo estes guardados na entidade *GpsData*. Existe ainda a noção de viagem representada pela entidade *Trip*, existindo duas instâncias para cada viagem: uma que representa o início da viagem e outra que representa o fim da viagem. Uma viagem tem início quando a ignição passa de desativa para ativa e o fim da viagem quando a ignição passa de ativa para desativa.

Para dar suporte aos relatórios foi introduzida a entidade *Report* na base de dados. Assim, para cada tipo de relatório que o cliente solicite receber é criada uma instância da entidade *Report*.

As entidades *Inspection*, *IUC*, *Insurance* e *Complement* representam respetivamente a inspeção, o imposto único de circulação, o seguro e o complemento de cada um dos veículos. Estas entidades foram acrescentadas na Base de Dados Operacional de modo a dar suporte à agenda de eventos associados aos veículos.

A Figura 3.4 representa o modelo de domínio de suporte à configuração dos sensores. Assim, os sensores de cada veículo podem ser configurados quanto a definições de energia (*PowerSettings*), entrada e saída de informação (*IOSettings*), rede GPRS (*GPRSSettings*), monitorização (*TrackingSettings*) e Roaming (*RoamingSettings*).

Tal como na Figura 3.3 as novas entidades introduzidas estão representadas a azul e a entidade que se mantém a branco.

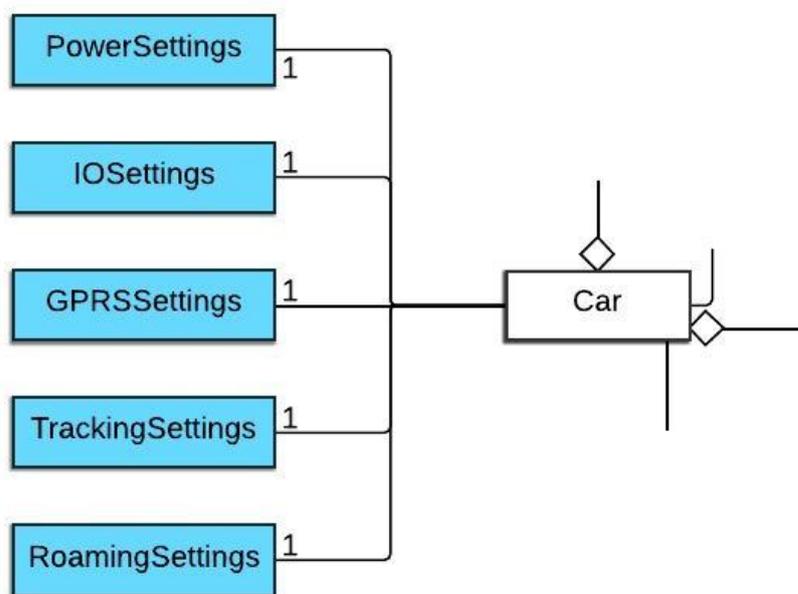


Figura 3.4: Modelo de Domínio da Base de Dados Operacional – Configuração de Sensores

6.2. Modelo da Base de Dados Multidimensional

Para gerar as tabelas dinâmicas e os relatórios de viagens é utilizada uma Base de Dados Multidimensional representada na Figura 3.5 usando a notação DFM (Dimensional Fact Model) [44].

Esta base de dados tem em vista responder a perguntas como: qual a distância percorrida por um determinado veículo com um determinado condutor? Quanto tempo demorou a fazer as viagens dos últimos três dias de um determinado condutor?

A tabela de factos (*Trip*) representa as viagens dos veículos.

As medidas são a distância percorrida (*distance*) em quilómetros, o tempo que o veículo esteve em movimento (*driveTime*) em horas, minutos e segundos, a duração da viagem (*duration*) em horas, minutos e segundos, a velocidade máxima (*maxVelocity*) em quilómetros por hora, a velocidade média (*avgVelocity*) em quilómetros por hora e ainda o gasto de combustível (*fuelConsumption*) em litros. O gasto em combustível é feito por estimativa, isto é, quando o sensor é colocado no veículo é inserido na base de dados qual o gasto litros a cada cem quilómetros e tendo em conta a velocidade média que teve é efetuado o cálculo, usando a expressão $\text{fuelConsumption} = (\text{avgFuelConsumption} * \text{distance}) / 100$.

As dimensões são o cliente (*Client*), o veículo (*Car*), o condutor (*Driver*), a localização (*Location*) e o tempo (*Time*). O cliente caracteriza-se por nome de utilizador (*Name*) e nome real (*Realname*); o carro por nome (*Name*), média de consumo por litros a cada cem quilómetros (*AvgFuelConsumption*), identificador da frota (*Fleet*) e nome da frota (*FleetName*); o condutor por valor do iButton (*iButtonValue*), nome (*Name*) e descrição do trabalho (*JobDescription*); a localização por endereço de início de viagem (*StartAddress*), valor da latitude da coordenada de início de viagem (*StartLatitude*), valor da longitude da coordenada de início de viagem (*StartLongitude*), endereço de fim de viagem (*EndAddress*), valor da latitude da coordenada de fim de viagem (*EndLatitude*) e valor da longitude da coordenada de fim de viagem (*EndLongitude*); o tempo por hora de início de viagem (*StartTime*), data de início de viagem (*StartDate*), dia da semana de início de viagem (*StartWeekDay*), hora de fim de viagem (*EndTime*), data de fim de viagem (*EndDate*) e dia da semana de fim de viagem (*EndWeekDay*).

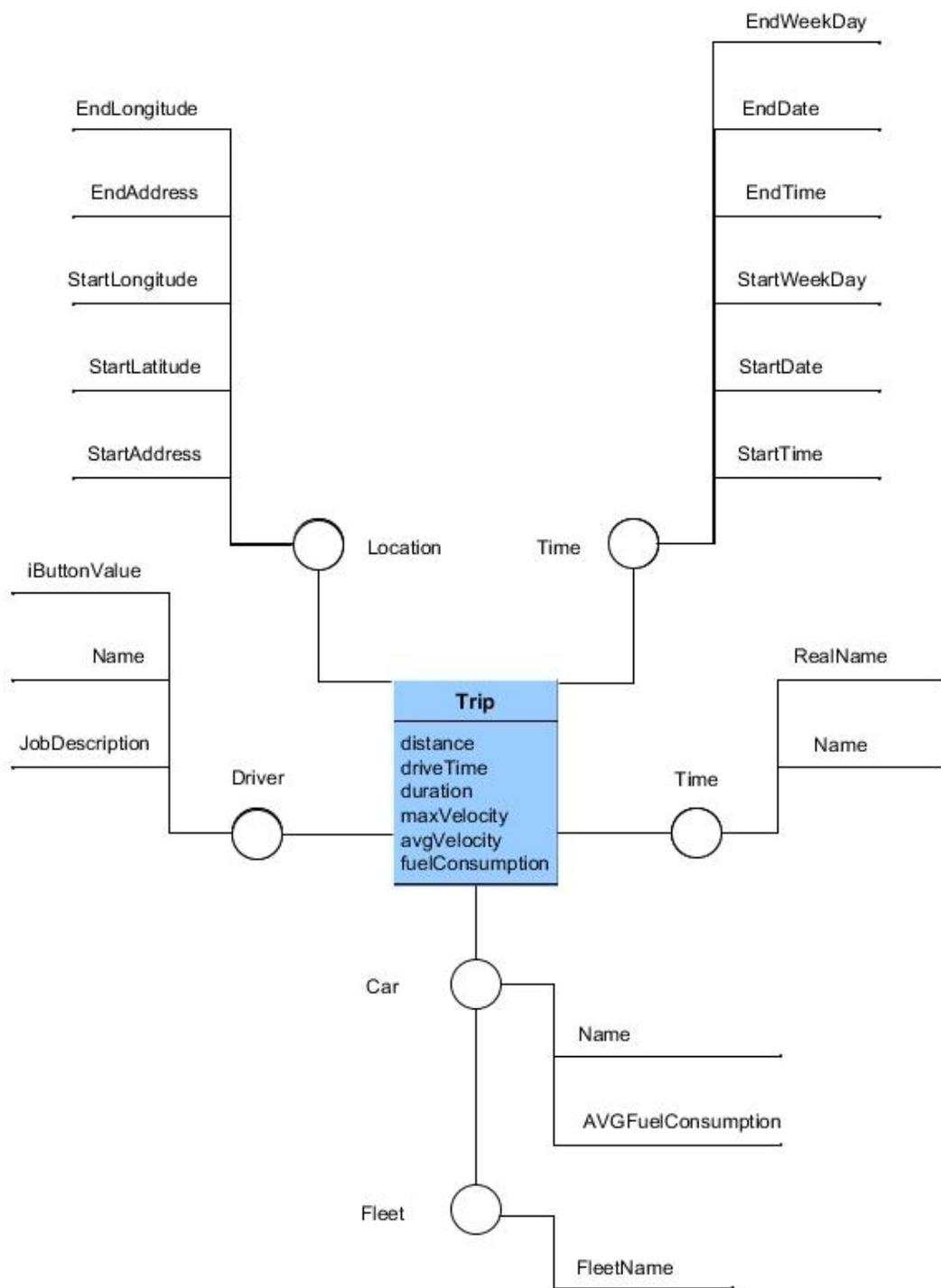


Figura 3.5: Dimensional Fact Model da Base de Dados Multidimensional

7 Modelo Físico

A Figura 3.6 apresenta o diagrama de implementação que mostra a organização em nós e componentes que formam o sistema.

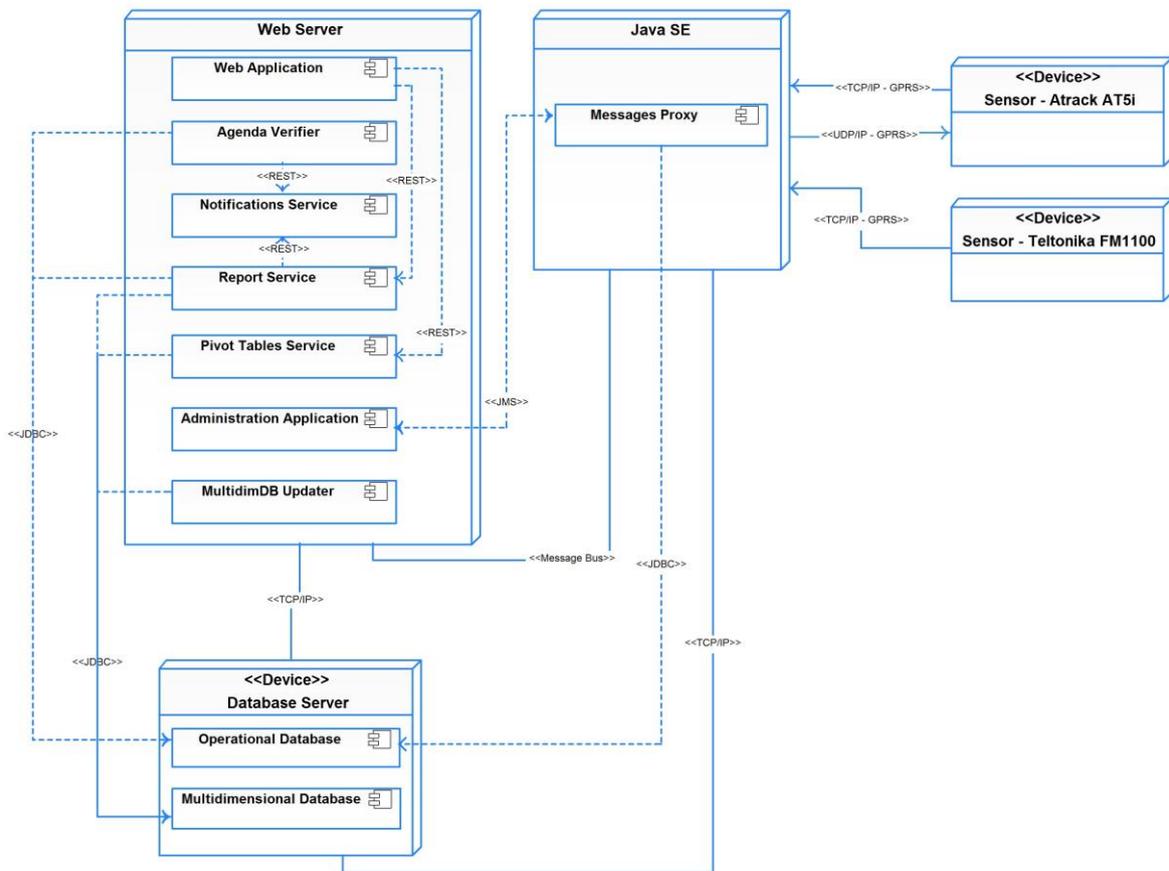


Figura 3.6: Diagrama de Implementação

- **Web Server** – Nó onde estão alojados os componentes *Web Application*, *Agenda Verifier*, *Notification Service*, *Report Service*, *Pivot Tables Service*, *Administration Application* e *MultidimDB Updater*.
 - **Web Application** – Representa a interface web à qual os clientes têm acesso para visualizar o estado das suas viaturas em tempo real,

informação sobre viagens efetuadas, gerir a agenda de eventos e gerir os relatórios.

- **Notification Service** – Responsável por enviar correio eletrónico e SMS utilizando métodos disponibilizados através de um interface REST.
- **Agenda Verifier** – Responsável por verificar a cada dia se existem eventos associados a veículos próximos da data atual. Este envia avisos aos clientes usando o componente *Notification Service*.
- **Report Service** – Representa os serviços que geram os relatórios de frota. Estes relatórios podem ser gerados espontaneamente (quando o cliente solicita naquele instante) ou periodicamente (quando agenda que lhe seja enviado diariamente, semanalmente ou mensalmente). Os dados provêm da *Multidimensional Database* para os relatórios de viagens e da *Operational Database* para os relatórios de posição.
- **Pivot Tables Service** – Representa o serviço REST que transforma os dados de viagens de um determinado cliente registados na *Multidimensional Database* num ficheiro CSV (Comma-Separated Values) que será usado para construir as tabelas dinâmicas.
- **Administration Application** – Responsável por disponibilizar uma interface de administração dos sensores ao administrador do sistema. Quando o administrador altera alguma configuração nos sensores esta mensagem é enviada para o barramento de mensagens (*Message Bus*) de modo a que o *Proxy* de Mensagens (*Messages Proxy*) as receba, esperando pela resposta por parte do *Proxy* de Mensagens se o sensor foi reconfigurando ou não.
- **MultidimDB Updater** – Responsável por atualizar a *Multidimensional Database*, passando os dados organizados da Base de Dados Relacional (*Operational Database*) para um esquema em estrela.

- **Database Server** – Servidor que aloja as duas bases de dados dos sistema: *Operational Database* e *Multidimensional Database*. As duas bases de dados estão alojadas no mesmo servidor por impossibilidade por parte da Posi-Sat em adquirir ou alugar um outro. Mas o objetivo, no futuro, é a *Multidimensional Database* migrar para outro servidor de modo a que haja balanceamento de carga, uma vez que as tarefas de gerar tabelas dinâmicas e relatórios de viagens ficam a cargo de um servidor e as restantes a cargo do outro.
 - **Operational Database** – Base de dados responsável pela gestão de todos os dados do sistema.
 - **Multidimensional Database** – Base de Dados Multidimensional com os dados organizados em esquema em estrela.

- **Java SE** – Ambiente onde corre o componente *Messages Proxy*.
 - **Messages Proxy** – Responsável por receber mensagens vindas dos sensores, via GPRS, interpretar o seu conteúdo e guardar os dados resultantes na *Operational Database*. É ainda responsável por receber mensagens vindas do barramento de mensagens, interpretá-las e enviar as instruções de configuração para os sensores (apenas para os Atrack AT5i).

- **Sensor Atrack AT5i** – Representa os sensores Atrack AT5i instalados nos veículos que enviam mensagens de posicionamento para o Proxy de Mensagens e recebem mensagens de configuração, também estas enviadas pelo *Proxy* de Mensagens.

- **Sensor Teltonika FM1100** – Representa os sensores Teltonika FM1100 instalados nos veículos que enviam mensagens de posicionamento para o *Proxy* de Mensagens.

Capítulo 4

Implementação

1 Tecnologias Usadas

As tecnologias usadas para a implementação desta solução são descritas seguidamente.

- **Web Server** – *Apache Tomcat 7* [45].
 - **Web Application** – GWT (*Google Web Toolkit*) [41] para desenvolver a Aplicação Web. *Google Maps JavaScript API V3* é a *Maps v3 bindings for GWT 3.8.0 prerelease 1*
 - **Notification Service** – Aplicação *Java EE6*. Faz uso da API *JavaMail* [46] para enviar correio eletrónico e da API *SMSGlobal* [47] para envio de SMSs.
 - **Agenda Verifier** – Aplicação *Java EE6* que faz uso da biblioteca de agendamento de tarefas *Quartz Scheduler 2.2.1* [48].
 - **Report Service** – Aplicação *Java EE6* que faz uso da biblioteca de agendamento de tarefas *Quartz Scheduler 2.2.1* [48]. É ainda usada a ferramenta *JasperStudio V5.5.1* [49] para produzir os relatórios de frota.
 - **Pivot Table Service** – Aplicação *Java EE6*.
 - **Administration Application** – Aplicação em GWT [41]. Faz uso da API JMS (*Java Message Service*) e do *Message Brocker RabbitMQ V3.3.1* [50] para aceder ao barramento de mensagens.
 - **MultidimDB Updater** – Aplicação *Java EE6* que faz uso da biblioteca de agendamento de tarefas *Quartz Secheduler 2.2.1* [48].

- **Database Server** – O Sistema de Gestão de Base de Dados usado é o *PostgreSQL* 8.4.21 [51] com a extensão *PostGIS* para operações espaciais.
- **Java SE – Java SE7.**
 - **Messages Proxy** – Aplicação *Java SE7*. Como API de acesso ao *socket* TCP usa *Java NIO* [31] para receção de mensagens vindas dos sensores. Faz uso da API *JMS* e do *Message Brocker RabbitMQ V3.3.1* [50] para aceder ao barramento de mensagens de configuração.

2 Base de Dados Operacional

A Figura A.1 em anexo apresenta o diagrama de classes completo da Base de Dados Operacional. Neste trabalho foram acrescentadas novas entidades (a azul) que dão suporte à agenda de eventos, aos relatórios de viagens e à configuração de sensores.

2.1. Agenda

A Figura 4.1 apresenta o diagrama de classes com apenas a parte referente à estrutura de dados que dá suporte à agenda.

Para cada inspeção (*Inspection*) é definida uma data de registo (*carRegistrationDate*), quanto tempo antes pretende ser alertado (*alert*) e observações (*observations*). O cliente é avisado anualmente deste evento.

Para cada IUC é definida uma data do primeiro pagamento efetuado (*dateStart*), quanto tempo antes pretende ser alertado (*alert*) e observações (*observations*). O cliente é avisado anualmente deste evento.

Para cada seguro (*Insurance*) é definida uma data do primeiro pagamento efetuado (*dateStart*), uma periodicidade para a qual o cliente deve efetuar os pagamentos

(*periodicity*), quanto tempo antes pretende ser alertado (*alert*) e observações (*observations*).

Para cada complemento (*Complement*) é definida uma data do primeiro pagamento efetuado (*dateStart*), uma periodicidade para a qual o cliente deve efetuar os pagamentos (*periodicity*), quanto tempo antes pretende ser alertado (*alert*), descrição do complemento (*description*) e observações (*observations*).

Para cada uma destas classes no caso do valor de *alert* ser *Null* o cliente não é avisado via correio eletrónico ou SMS.

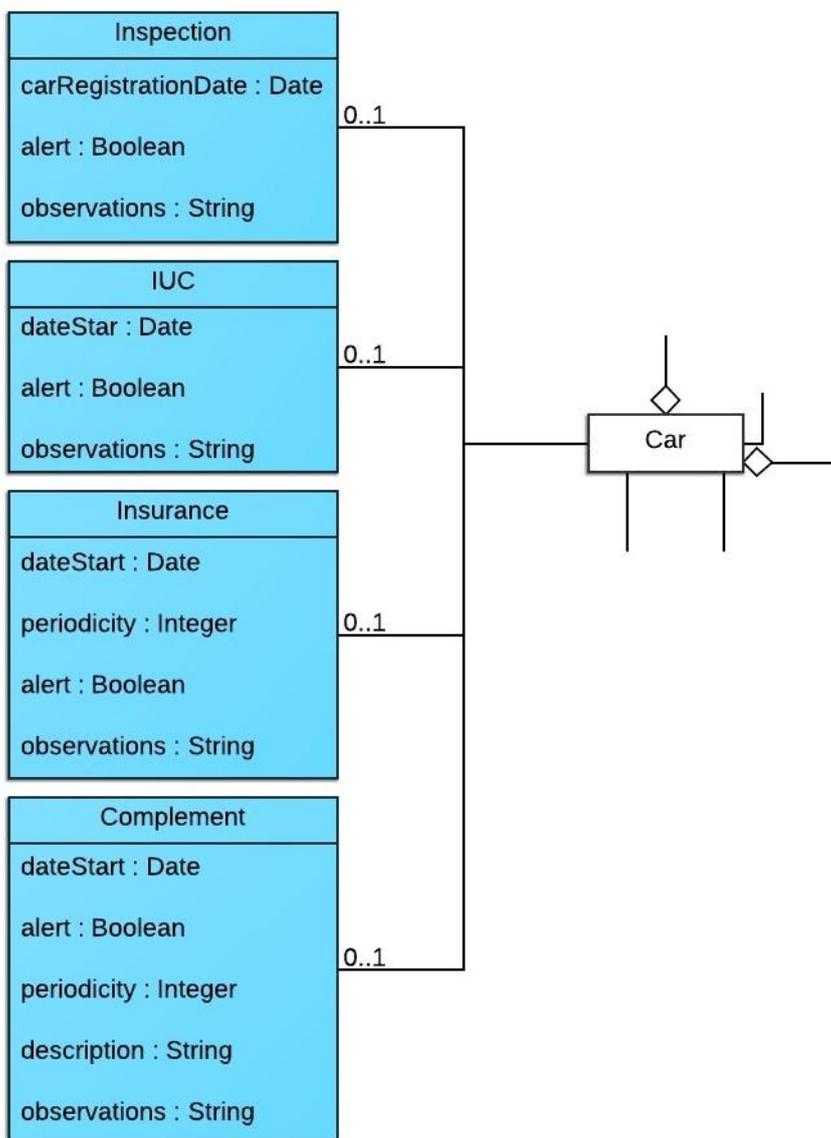


Figura 4.1: Diagrama de Classes Referente à Agenda

A parte do sistema referente à agenda foi implementada em dois módulos distintos, um associado à Aplicação Web em que se disponibiliza a interface para criar, editar, eliminar e consultar eventos, que será abordada na secção 6 do **Error! Reference source not found.** e o outro associado a uma tarefa periódica para verificar diariamente quais os eventos que se estão a aproximar (*Agenda Verifier*) para que o responsável pela frota respetiva seja avisado via correio eletrónico ou SMS.

O módulo *Agenda Verifier* faz uso da biblioteca de agendamento de tarefas *Quartz Job Scheduler*. Para configurar o *Quartz* são necessários dois ficheiros [48] o *quartz.properties* e o *quartz-config.xml*. O ficheiro *quartz.properties* define propriedades como o nome da instância, a sua identificação (ID), qual a classe usada como *threadPool*, o número de *threads*, o *plugin* responsável por inicializar a tarefa e o ficheiro de configuração do inicializador da tarefa. As configurações usadas no módulo implementado neste sistema são apresentados na Figura B.1 em anexo.

O ficheiro referenciado pelo *quartz.properties* encontra-se na Figura B.2 em anexo. É neste que se define qual a classe que é chamada quando o *trigger* associado à tarefa agendada é disparado e de quanto em quanto tempo este deve ser disparado. Neste caso, todos os dias pelas sete horas da manhã (*cron-expression*), perto da hora a que habitualmente os portugueses acordam.

O *SchedulerJob* irá por sua vez verificar os eventos próximos na Base de Dados Operacional e usar o serviço de notificações para enviar uma mensagem de correio eletrónico ou SMS a avisar os clientes da proximidade desses eventos.

As funções para gestão (criar, editar, eliminar e disponibilizar) desta estrutura de dados, as funções de cálculo de quais as próximas datas dos eventos e as funções para verificação de quais os eventos a aproximar-se são todas implementadas usando funções do lado da Base de Dados Operacional, sendo que a Aplicação Web e o *Agenda Verifier* apenas vão fazer uso destas funções.

2.2. Relatórios de Frota

A Figura 4.2 apresenta a estrutura na Base de Dados Operacional que permite definir quais os relatórios de frota que o cliente pretende e a periodicidade dos mesmos.

Para tal existe uma classe *Report* que representa um tipo de relatório (*type*) podendo este ser do tipo relatório de viagens ou relatório de posições. Para além disso é definido se pretende que seja enviado diariamente (*daily*), semanalmente (*weekly*) ou mensalmente (*monthly*).

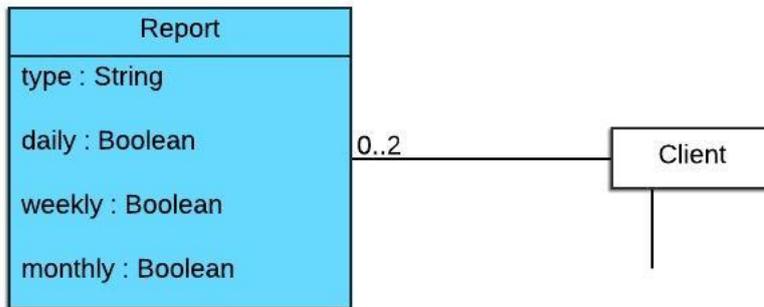


Figura 4.2: Diagrama de Classes Referente aos Relatórios de Frota

2.3. Configuração de Sensores

Para efetuar a configuração dos sensores Atrack AT5i é guardado na Base de Dados Operacional o estado de cada um desses sensores. O diagrama de classes da Figura 4.3 mostra a estrutura de classes que dá suporte a esta parte do sistema.

Os sensores podem ser configurados quanto às definições de energia (*PowerSettings*), de entrada e saída de dados (*IOSettings*), de rede GPRS (*GPRSSettings*), de monitorização (*TrackingSettings*) e de *roaming* (*RoamingSettings*).

Quanto às definições de energia pode ser definido se o *Sleep Mode* está ativo (*sleepModeState*), nesse caso, qual o tipo (*NORMAL* ou *DEEP*). Caso esteja ativo, em que situações deve ser usado (*sleepModeType*): a ignição é desligada (*ACC_OFF*), o motor é desligado (*ENGINE_OFF*) ou quando o veículo para de se movimentar (*MOTION_OFF*).

Existem três fios que ligam o sensor ao veículo configuráveis: o fio 1 (*IO1*) pode ser uma entrada (*INPUT1*) ou uma interface de um fio (*IWIRE_INTERFACE*), o fio 2 pode ser uma entrada (*INPUT2*), uma saída (*OUTPUT1*) ou uma entrada analógica (*ANALOG_INPUT*) e o fio 3 pode ser uma entrada (*INPUT3*) ou uma saída (*OUTPUT2*).

Para configurar a rede GPRS é necessário definir o nome da APN (*Access Point Name*) (*APNName*), o nome de utilizador associado ao sensor (*userName*), o número de tentativas de reenviar em casos de falha (*connectionRetry*), o tempo após o qual deve voltar a tentar (*retryTimeout*) e de quando em quanto tempo é enviada a mensagem de *keepalive* (*keepAlive*).

Quanto às definições de monitorização pode ser definido o modo de monitorização (*trackingMode*), isto é, enviar mensagens a uma determinada frequência temporal (*TIME*) ou quando é percorrida uma determinada distância (*DISTANCE*). Estes dois modos podem estar ativos individualmente (*LogicMode = OR*) ou ao mesmo tempo (*LogicMode = AND*). Pode ainda ser definida de quanto em quanto tempo enviar mensagens (*timeValue*) ou qual a distância (*distanceValue*), dependendo de qual o modo ativo.

As definições de *roaming* disponíveis são não enviar SMS quando em *roaming* (*stopSMS*) ou parar de usar a rede GPRS quando em *roaming* (*stopGPRS*).

Para além destas configurações os administradores podem mandar reiniciar os sensores, limpar a fila de mensagens, reiniciar as definições do dispositivo (definições por omissão) e ativar a imobilização do veículo em caso de roubo.

A Figura 4.3 não apresenta os métodos usados uma vez que para todas as entidades são apenas implementadas as operações básicas CRUD (*create, read, update e delete*).

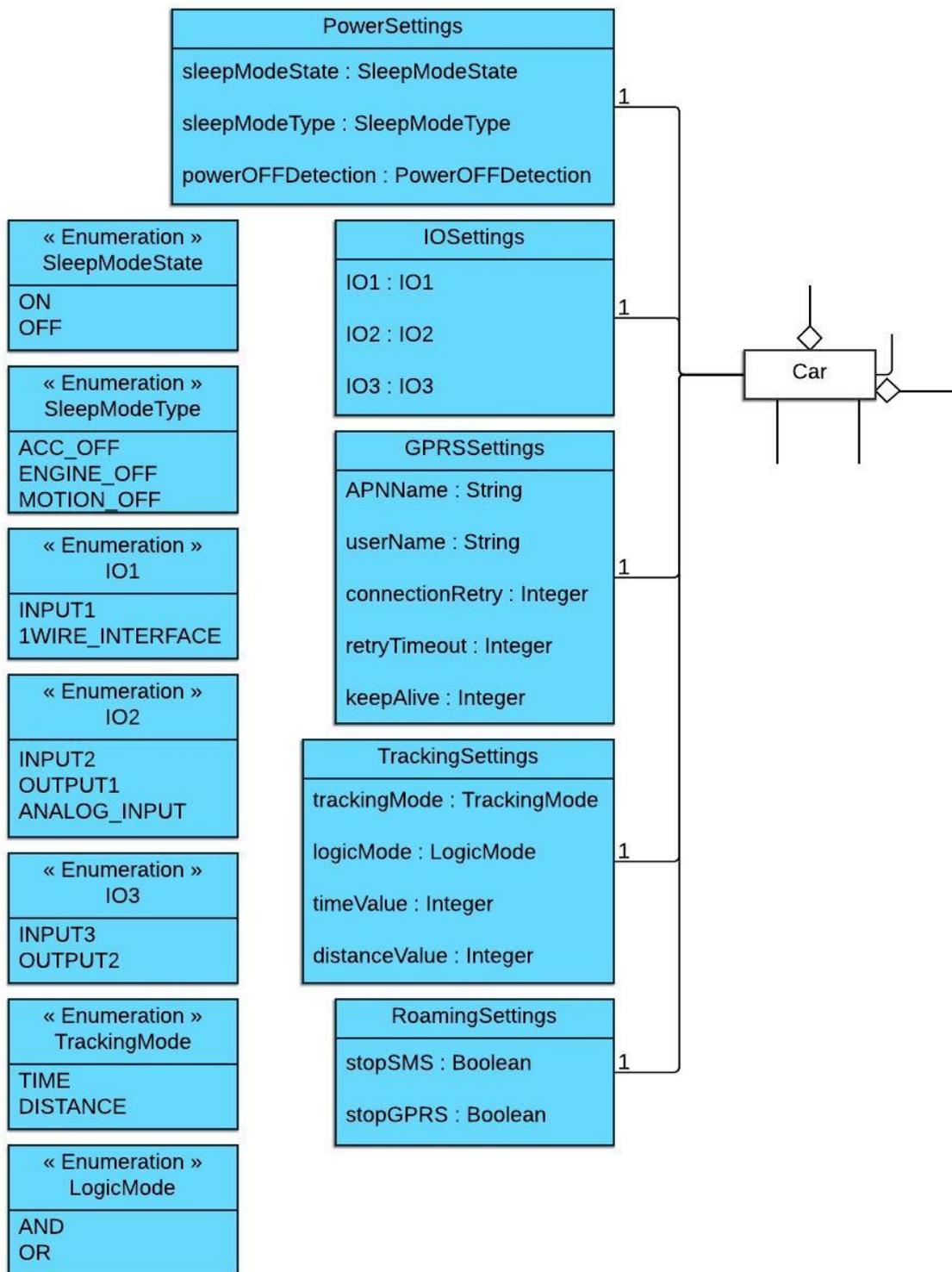


Figura 4.3: Diagrama de Classes Referente à Configuração de Sensores

2.4. Proxy de Mensagens – Receção de Mensagens

Para a receção de mensagens no sistema vindas dos sensores foi utilizada a biblioteca Java NIO. O diagrama de classes da Figura 4.4 mostra as classes que foram usadas para implementar esta solução.

A receção de mensagens é feita em modo não bloqueante, isto é, quando uma *thread* tenta ler uma mensagem do *channel*, retira do *buffer* os bytes de dados que tem disponível no momento, quando o *buffer* estiver vazio em vez de ficar bloqueada à espera de mais é libertada para poder efetuar qualquer outra tarefa.

Como o *proxy* vai ter o papel de servidor na receção de mensagens é-lhe associado um *ServerSocketChannel* que, por sua vez, tem um *Selector* que vai ser responsável por ativar uma chave, *SelectionKey*, por cada ligação estabelecida.

É na *thread Reactor* que todas as outras *threads* (associadas à receção de mensagens) e estruturas de dados são instanciadas. Para além disso, esta *thread* está sempre ativa e tem a tarefa de receber as *SelectionKey* selecionadas pelo *Selector* e enviá-las para uma das *SelectedKeysThread*. Estas últimas têm o papel de chamar o *Handler* respetivo, isto é, *AcceptHandler* quando a chave é do tipo *OP_ACCEPT* ou *SensorRequestHandler* quando a chave é do tipo *OP_READ*.

Quando o *AcceptHandler* é chamado este vai criar um *SocketChannel* para que mais tarde seja, efetivamente, feita a leitura da mensagem, associando à chave a operação de leitura e respetivo *Handler*, ou seja, o *SensorRequestHandler*.

A tarefa do *SensorRequestHandler* é ler a mensagem que está no *buffer* e colocá-la na fila com as *MessageToProcess*. Estas vão ser atendidas pelas *threads DataHandlerThread* que, por sua vez, vão verificar se existem mensagens antigas, nas *Message*, guardadas com este mesmo endereço IP, para os casos em as mensagens tenham sido recebidas em partes. A mensagem que foi recebida é colocada nas *ReadyMessage*, sendo que o método *getReadyMessage()* define qual é o tipo de mensagem que foi recebida. Caso as mensagens sejam dos sensores Atrack AT5i é neste mesmo método que o conteúdo é verificado quanto a erros de transmissão através do campo CRC e decifradas para que se possa analisar o conteúdo das mesmas. Quem trata as *ReadyMessage* é a *ReadyMessageHandlerThread*, sendo que esta chama o método

handleDataFromMessage() da *ReadyMessage* para tratar a mensagem e colocar os dados na Base de Dados Operacional, caso seja necessário.

O *DisconnectFinder* é uma *thread* que tem como função percorrer as *Message* e encontrar quais os endereços IP para os quais já não são recebidas mensagens há mais de *sensorTimeoutNS* nanosegundos. Isto permite libertar memória que, caso tenham passado *sensorTimeoutNS* não faz sentido continuar alocada.

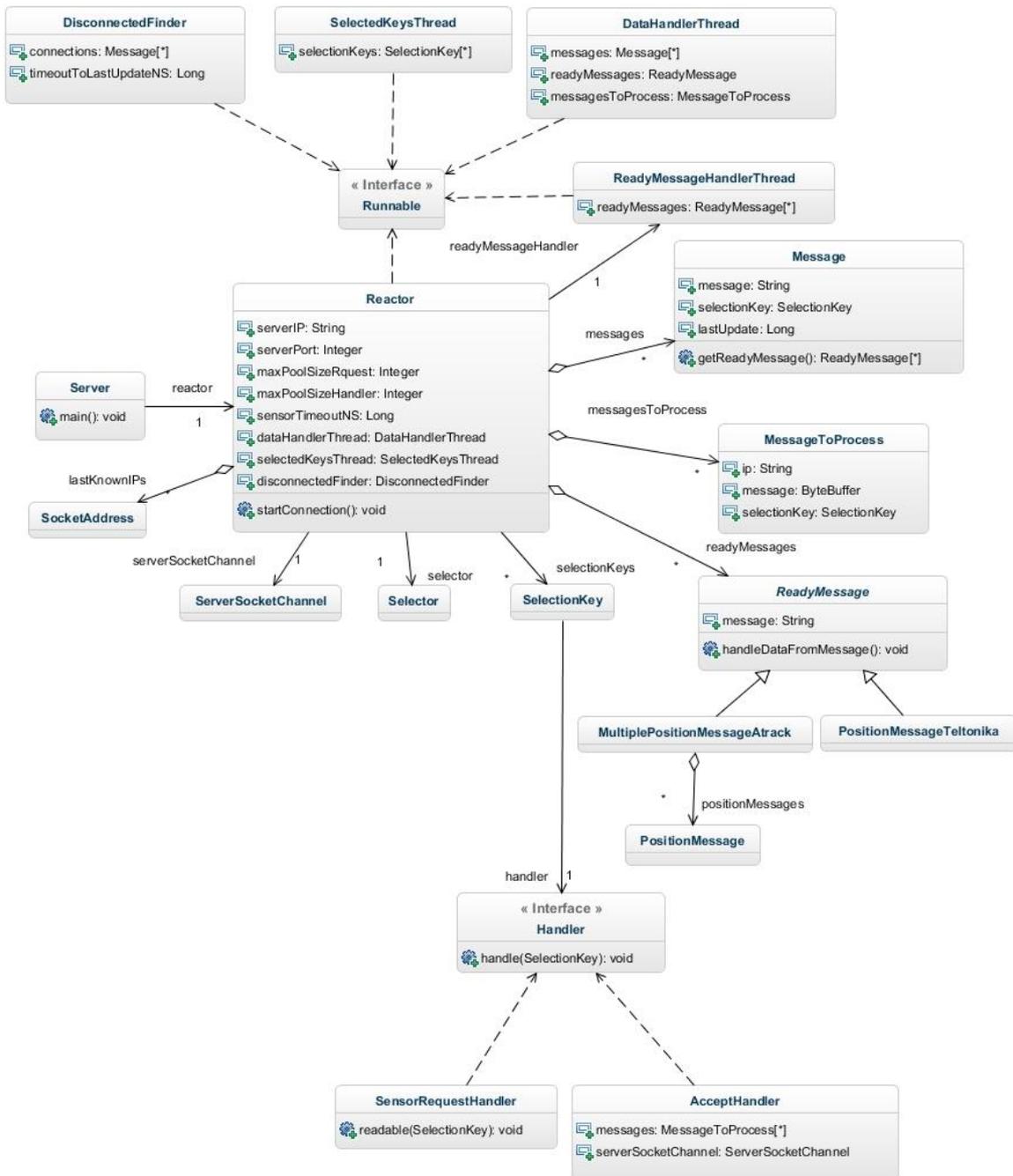


Figura 4.4: Diagrama de Classes do *Proxy* de Mensagens - Recepção

2.5. Proxy de Mensagens – Configuração de Sensores

A Figura 4.5 mostra como as mensagens de configuração são enviadas para os sensores. Ao contrário do que acontece com a recepção de mensagens, o *proxy* faz o envio de mensagens para os sensores de uma forma bloqueante, isto porque depende de uma resposta por parte dos sensores para que o *proxy* possa verificar que a instrução foi efetuada. Para além disso o envio deste tipo de mensagens será esporádico, pelo que, neste caso, o desempenho não é um fator importante.

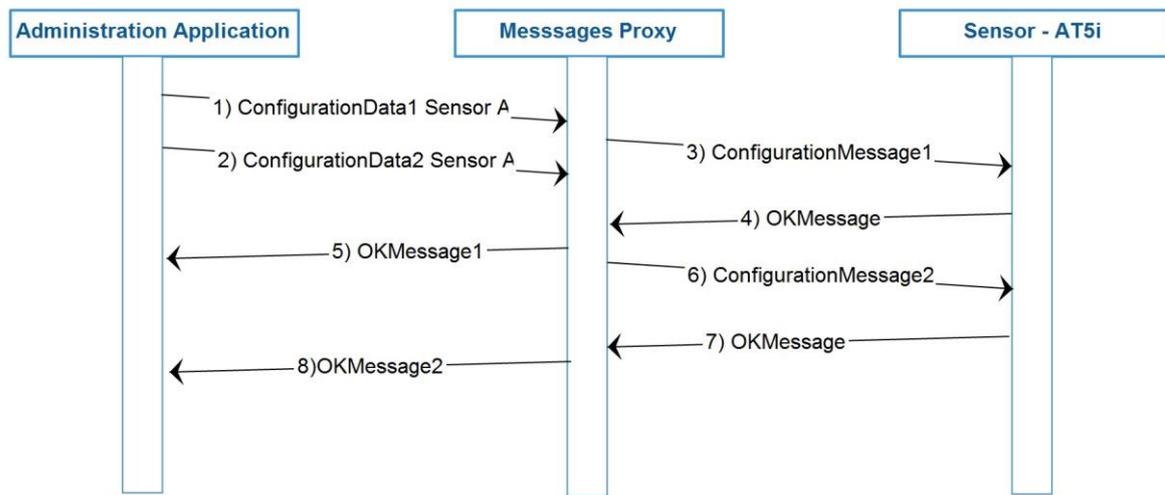


Figura 4.5: Diagrama de Fluxo de Pacotes de Configuração

A Figura 4.6 mostra o diagrama de classes da implementação do envio de mensagens de configuração para os sensores. Quando a aplicação de administração envia dados de configuração para o *proxy* estes são transformados numa mensagem que o sensor consegue perceber (*WriteMessage*), sendo este processo efetuado pela *MessageReceiverThread*. Estas mensagens são enviadas pelas *MessageWriterThread* usando *OutputStream* do *Socket*. E no caso de haver mais mensagens para enviar para um determinado sensor, essas mensagens são enviadas por ordem e sempre pela mesma *thread*, isto é, a *thread* que envia a primeira mensagem espera que venha uma resposta desse mesmo sensor para poder enviar a segunda mensagem, se esta existir, e assim sucessivamente. Se receber uma mensagem de OK, significa que a configuração foi concretizada, por parte do sensor, e essa informação é reencaminhada para a

Administration Application, caso contrário o *Proxy* de Mensagens vai tentar reenviar a mensagem duas vezes. No caso de o sensor não responder, essa configuração é cancelada e a *Administration Application* é informada. Se ao tentar conectar com o sensor este não estiver disponível, passa para a seguinte, sendo que cada configuração tem três tentativas, caso não seja possível, é descartada e a *Administration Application* é avisada que a configuração não se concretizou.

Para enviar mensagens de configuração da *Administration Application* para o *proxy* é usado o *RabbitMQ* implementando o paradigma produtor-consumidor [50]. O produtor é a *Administration Application* que coloca a mensagem na fila partilhada com o *proxy*, sendo o *message broker* responsável por gerir a fila partilhada pela *Administration Application* e pelo *Proxy* de Mensagens. Este como está à espera de mensagens vai recebê-la e processá-la e continuar com as suas tarefas.

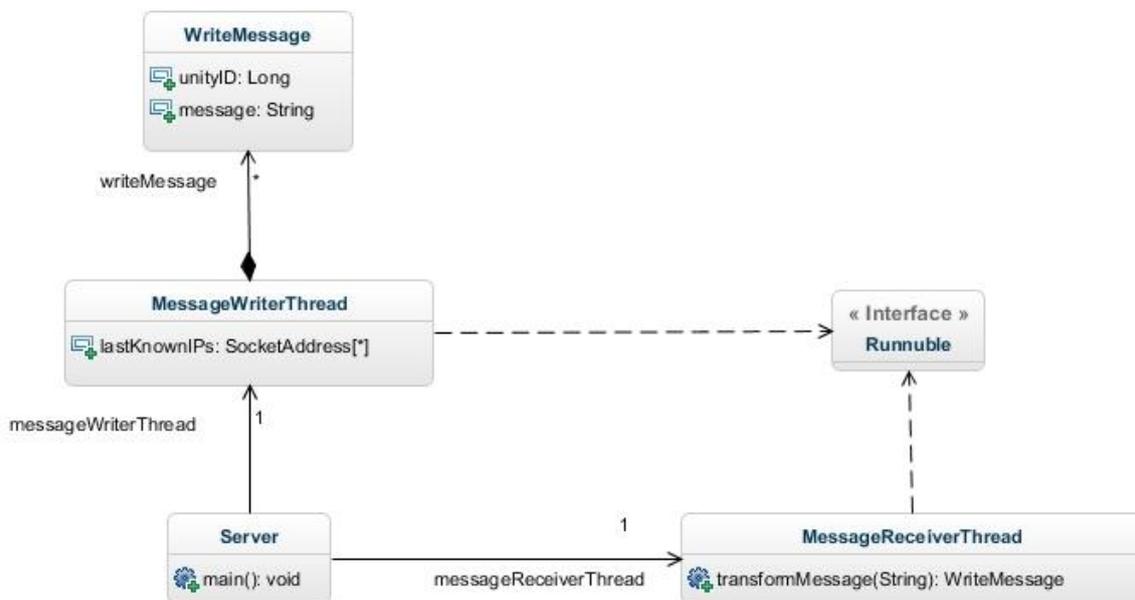


Figura 4.6: Diagrama de Classes do *Proxy* de Mensagens - Envio

3 Base de Dados Multidimensional

O módulo *MultidimDB Updater* faz diariamente atualização na Base de Dados Multidimensional onde vai transformar os dados que estão organizados de uma forma

relacional (Figura A.1 em anexo) para um esquema em estrela [52] (Figura 3.5), sendo que estes dados são usados para produzir tabelas dinâmicas e relatórios de frota. Neste módulo é usada a mesma configuração do *Quartz* feita no *Agenda Verifier* (Figura B.1 e Figura B.2 em anexo), para todos os dias efetuar esta atualização.

As funções que tratam de efetuar esta tarefa de transformar os dados de um esquema para o outro estão implementadas do lado da Base de Dados Multidimensional. Desta feita, o *MultidimDB Updater* apenas tem que chamar estas funções.

4 *Report Service*

Este módulo é o responsável por produzir os dois tipos de relatórios: de viagens e de posições. Os relatórios podem ser produzidos diariamente, semanalmente e mensalmente. Por outro lado, ao aceder à Aplicação Web é possível solicitar a extração destes mesmos relatórios, sendo que a Aplicação Web faz uso do serviço REST disponibilizado pelo *Report Service* para os produzir.

Neste módulo é usada a mesma configuração do *Quartz* feita no *Agenda Verifier* (Figura B.1 e Figura B.2 em anexo), para todos os dias verificar quais os relatórios que devem ser enviados.

A aparência e a fonte de dados de cada tipo de relatório na ferramenta *Jaspersoft Studio* é definido num ficheiro de configuração XML [49] que é referenciado quando o *Report Service* pretende produzir um relatório.

Para os relatórios de viagens os dados são retirados da Base de Dados Operacional, enquanto para os relatórios de posições os dados são retirados da Base de Dados Multidimensional.

É possível observar um exemplo de relatório de viagens no Capítulo 5 Anexo C e um exemplo de relatório de posições no **Error! Reference source not found..**

5 *Pivot Table Service*

Este é o módulo responsável por gerar os ficheiros CSV para que depois possam ser usados para gerar tabelas dinâmicas. Possui um interface REST que dadas uma data inicial,

uma data final e o nome do cliente, gera a informação que possui relativamente às viagens efetuadas pelas frotas desse cliente, sendo que a fonte de dados é a Base de Dados Multidimensional.

No sistema é a Aplicação Web que usa este interface quando um cliente solicita a geração do ficheiro CSV.

6 Interação com os Utilizadores

Nesta secção é apresentada a interação que o utilizador tem com o sistema para gerir a agenda de eventos, para navegar no mapa e como pode utilizar as tabelas dinâmicas.

6.1. Agenda de Eventos

A Figura 4.7 mostra a página que o utilizador tem acesso para verificar quais os eventos que se vão realizar no próximo mês, tendo a possibilidade de seleccionar eventos já efetuados para que saiam da lista.

	Tipo de Evento	Veiculo	Próxima Data	
<input type="checkbox"/>	Seguro	83-MT-55 Renault Master	02-06-2014	Passaram 14 dias
<input type="checkbox"/>	Complemento desc	80-LA-16 renault master	04-06-2014	Passaram 12 dias
<input type="checkbox"/>	Seguro	80-LA-16 renault master	04-06-2014	Passaram 12 dias
<input type="checkbox"/>	Complemento somedesc	66-AZ-38 Mercedes sprinter	22-06-2014	Faltam 6 dias
<input type="checkbox"/>	Seguro	66-AZ-38 Mercedes sprinter	22-06-2014	Faltam 6 dias
<input type="checkbox"/>	Seguro	83-MT-55 Renault Master	02-07-2014	Faltam 16 dias
<input type="checkbox"/>	Seguro	80-LA-16 renault master	04-07-2014	Faltam 18 dias
<input type="checkbox"/>	Complemento asdad	83-MT-55 Renault Master	05-07-2014	Faltam 19 dias
<input type="checkbox"/>	Seguro	21-CE-56 Sprinter	06-07-2014	Faltam 20 dias

Figura 4.7: Interface com Lista de Eventos

A Figura 4.8 mostra a página que faculta a lista de seguros associados aos veículos de um determinado cliente. Esta página permite seleccionar qual pretende eliminar ou editar ou ainda criar um novo. As restantes modalidades (Revisões, IUC e Complementos) não são apresentadas tendo em conta que a interação é muito semelhante à presente na Figura 4.8.

The screenshot shows the POSI-SAT web application interface. At the top, there is a navigation bar with 'Vista Mapa', 'Relatórios', and 'Agenda' tabs. Below the navigation, there is a calendar icon labeled 'AGENDA' with the number '1' and the word 'Agenda'. To the right, there is a 'PONTOS DE INTERESSE' icon. The main content area is titled 'Agendamento de Pagamento de Seguros'. It contains a table with the following columns: 'Veículo', 'Data de Início', 'Periodicidade', 'Antecedência do Alerta', 'Observações', and 'Próximo Pagamento'. The table lists four vehicles, with the last one, '83-MT-55 Renault Master', selected. Below the table, there are buttons for 'Eliminar', 'Editar', and 'Criar novo'.

Veículo	Data de Início	Periodicidade	Antecedência do Alerta	Observações	Próximo Pagamento
<input type="checkbox"/> 21-CE-56 Sprinter	06-03-2014	2 meses	1 dia		06-07-2014
<input type="checkbox"/> 80-LA-16 renault master	04-03-2014	1 mês			04-07-2014
<input type="checkbox"/> 66-AZ-38 Mercedes sprinter	22-12-2013	3 meses	0 dias	someobs	22-06-2014
<input checked="" type="checkbox"/> 83-MT-55 Renault Master	02-03-2014	1 mês	1 dia		02-07-2014

Figura 4.8: Interface com Lista de Seguros dos Veículos

A Figura 4.9 mostra a página que fornece o formulário para que o utilizador possa editar os dados referentes ao seguro associado a um determinado veículo. O formulário para criar um novo seguro é muito semelhante ao presente na Figura 4.9, diferenciando apenas no facto de ser possível seleccionar qual o veículo a que está associado o seguro.

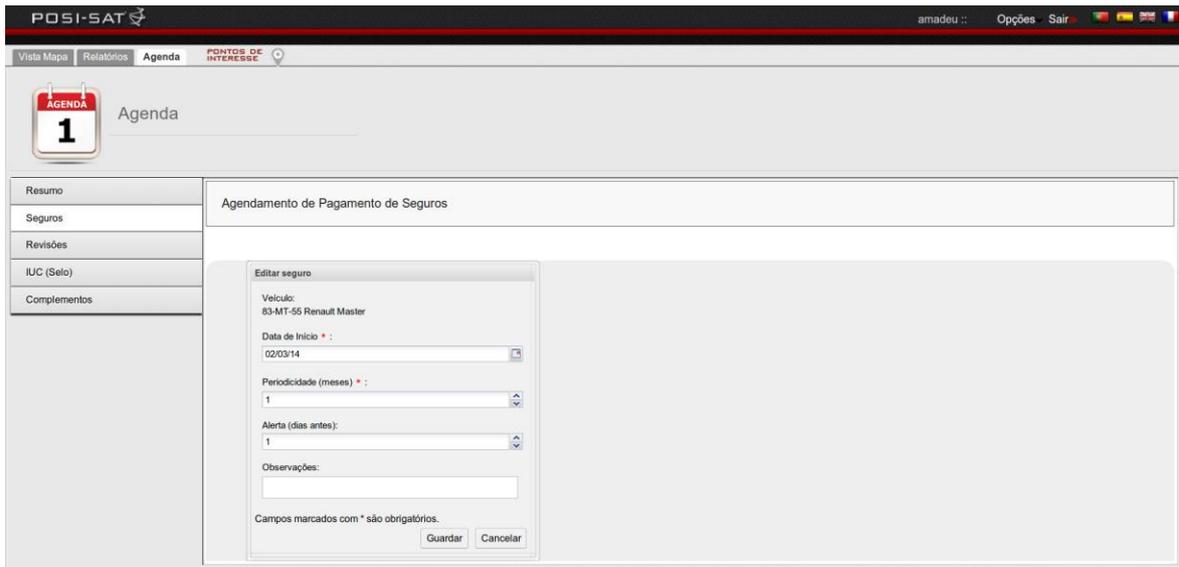


Figura 4.9: Interface para Editar Informação de Seguro

6.2. Web Mapping

Na Figura 4.10 é possível visualizar a última posição conhecida de cada veículo de uma determinada frota, anteriormente selecionada, no mapa. As posições vão sendo atualizadas em tempo real à medida que as mensagens de posicionamento chegam ao sistema, sendo necessário efetuar *refresh* na página para visualizar as novas posições.

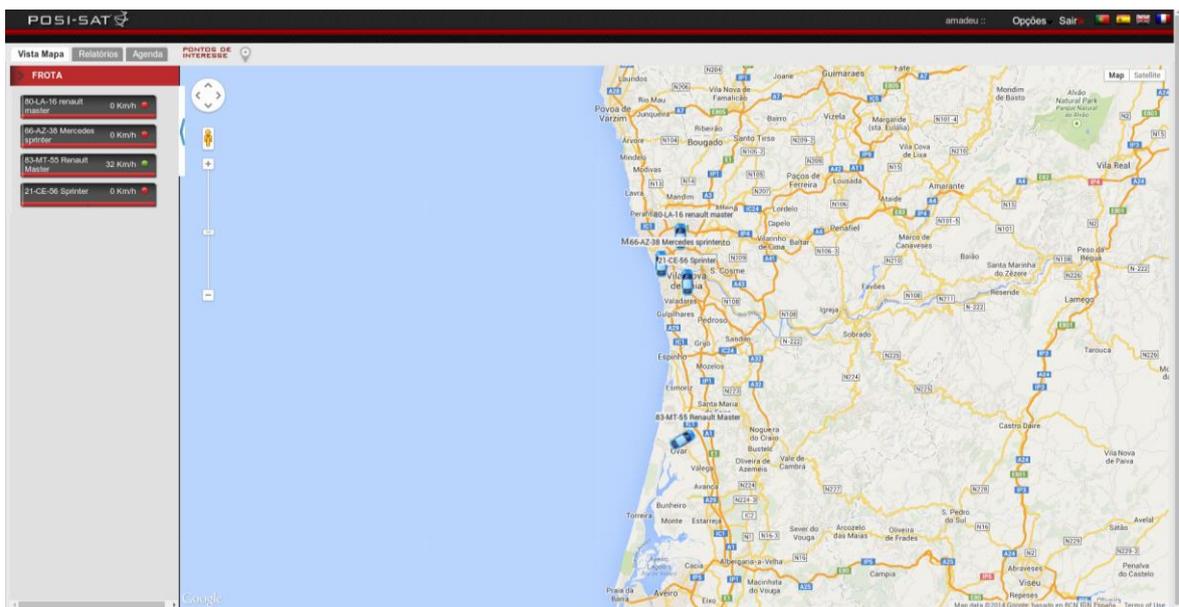


Figura 4.10: Interface com Visão Geral dos Veículos do Utilizador

Na Figura 4.11 é possível visualizar o percurso efetuado por um determinado veículo numa viagem. Cada percurso tem assinaladas as posições pelas quais o veículo passou podendo visualizar para cada uma delas pormenores de tempo de passagem, morada e velocidade instantânea.

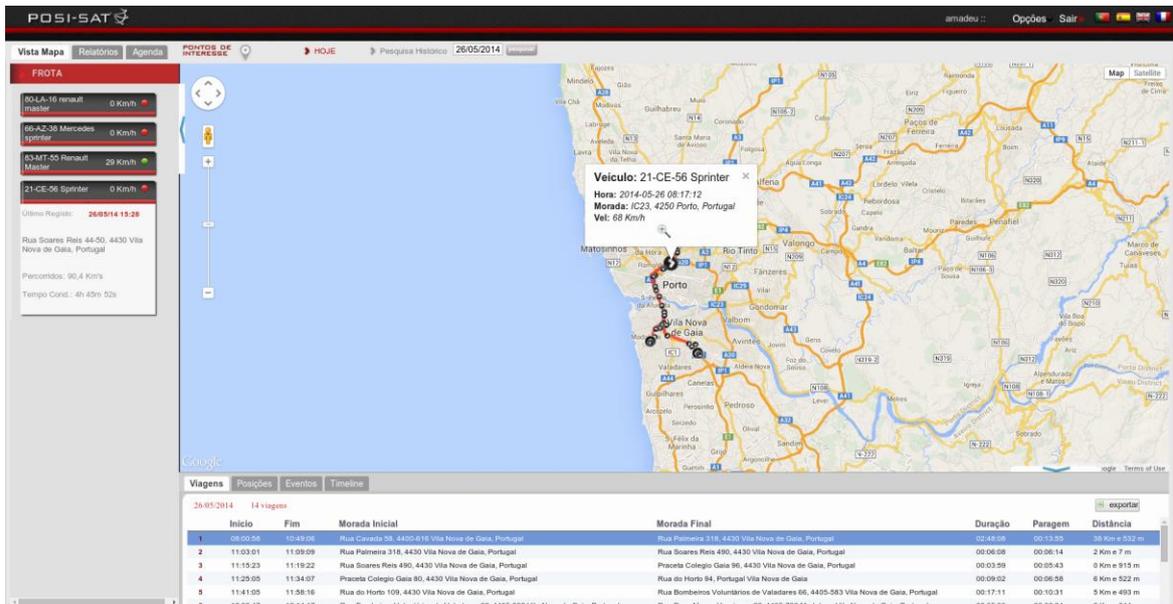


Figura 4.11: Interface com Informação Relativa a uma Viagem

As alterações que foram necessárias efetuar para passar da versão dois para a versão três da API concentraram-se principalmente em pormenores de chamadas de métodos (nomes, tipos e número de argumentos, etc.) [53].

6.3. Tabelas Dinâmicas

Na Aplicação Web o cliente pode solicitar que seja gerado um ficheiro CSV para que depois o possa usar para gerar as tabelas dinâmicas no Microsoft Excel. Na Figura 4.12 está um exemplo do uso de tabelas dinâmicas, onde é possível verificar para um determinado cliente, qual a distância total percorrida por frota, data de início e nome do veículo.

Row Labels	Soma de Distancia
Frota Centro	31637
21/04/2014	31637
00-JA-46 Fiat Doblo	31637
Frota Norte	16423
21/04/2014	16423
49-29-GM VOLVO FL6 porta 5	2627
77-HJ-39	1314
PG	6150
Renault Clio -11-GN-67	6332
Frota Sul	28016
21/04/2014	28016
07-BO-31 DAF XF	3353
14-CC-93 DAF CF	1135
21-CE-56 Sprinter	8950
51-12-JC FH12	666
66-AZ-38 Mercedes sprinter	11092
SKODA	2820
masteradminfleet	13636
21/04/2014	13636
Ford Fiesta	13636
Grand Total	89712

Figura 4.12: Exemplo de Tabelas Dinâmicas em Microsoft Excel

7 Implementação de Requisitos não Funcionais

Nesta secção é abordada em que situações o sistema obedece aos requisitos não funcionais levantados no início do trabalho.

- **Usabilidade:** Os clientes já estão a utilizar a nova versão da Aplicação Web com as novas funcionalidades introduzidas no sistema. Foi pedido aos clientes que no caso de terem dificuldades ou alguma situação que não lhes agradasse no interface informassem a administração da Posi-Sat, sendo que até ao momento não se sucedeu nenhum caso.
- **Fiabilidade:** Desde que esta nova versão do sistema está a ser utilizada, o sistema esteve sempre disponível.

- **Portabilidade:** A Aplicação Web está desenvolvida em GWT e já foi testado nos principais navegadores: Internet Explorer, Firefox, Safari e Chrome. Sendo que o mais usado pelos clientes é o Internet Explorer.
- **Segurança:** As mensagens enviadas pelos sensores Atrack AT5i faz a cifra de mensagens para AES-128 para evitar *Packet Sniffing* e *Man in the Middle*. Para além disso, as mensagens têm o campo CRC para ajudar na deteção de erros.
- **Interoperabilidade:** O *Proxy* de Mensagens é compatível com ambos os tipos de sensores presentes no sistema.
- **Extensibilidade:** No futuro, se for necessário adicionar novos sensores ao sistema, o *Proxy* de Mensagens é facilmente extensível para outros tipos de protocolo, bastando adicionar métodos capazes de interpretar as mensagens do novo protocolo.

Capítulo 5

Conclusão e Trabalho Futuro

Todos os requisitos que a Posi-Sat estabeleceu para este projeto foram cumpridos e alguns já estão em produção, uma vez que os clientes finais já estão a usufruir das novas funcionalidades e das que foram melhoradas. Até ao momento, o feedback dado pelos utilizadores tem sido positivo no que diz respeito à agenda de eventos, aos relatórios de frota e à nova versão do mapa. A parte do sistema que disponibiliza as tabelas dinâmicas ainda não está a ser usada pelos clientes, no entanto, logo que a Posi-Sat pretenda está pronta para tal. O novo *Proxy* de Mensagens é o que está de momento a ser utilizado e ainda não foram identificados problemas, pelo que é possível concluir que este projeto foi terminado com sucesso.

Embora ao sistema ainda seja necessário acrescentar algumas funcionalidades para se tornar tão completo quanto os demais concorrentes, este trabalho foi uma mais-valia para que o sistema se venha a tornar competitivo tanto ao nível de custo como ao nível de funcionalidades que fornece ao cliente final.

O produto disponibilizado pela Posi-Sat pode ainda evoluir significativamente de forma a torná-lo mais rico e atrativo. Possíveis trabalhos futuros que já foram propostos pelos próprios responsáveis e que foram surgindo ao longo deste projeto são:

- Disponibilizar informação na Aplicação Web relativa a Eco-Condução (travagens bruscas, acelerações, curvas efetuadas bruscamente). No entanto, para isso vai ser necessário aumentar a frequência de amostragem, uma vez que neste momento é enviada uma mensagem pelo sensor de um em um minuto quando o veículo se encontra em Portugal e de dez em dez minutos quando o veículo se encontra fora de Portugal.

- Assim que o veículo é estacionado, criar uma zona de segurança à volta da viatura. Se este sair dessa zona sem a ignição estar ligada, o responsável pelo veículo deve ser avisado.
- Incluir funcionalidades de definição de rotas integradas com dispositivos de navegação nos veículos de modo a que os condutores saibam o percurso que devem seguir, tendo em atenção quais as viaturas mais próximas de determinados locais e condições de tráfego.
- Relatórios com passagens em pontos de interesse (locais onde era suposto o veículo passar).
- Relatórios com informação acumulada de qual o percurso que consome menos ou demora menos tempo para locais habituais, tendo em conta o tipo de veículo.
- Relatórios com perfil de condução tendo em conta Eco-Condução.
- Integrar as tabelas dinâmicas com a Aplicação Web, não sendo necessário aceder usando Microsoft Excel. Por exemplo, a ferramenta WebPivotTable.
- Detecção de anomalias em sensores e atuação automática.
- Receção de mensagens via SMS de posicionamento no *Proxy* de Mensagens para situações de falta de rede GPRS.

Referências

- [1] Cartrack, “Sobre Nós | Cartrack,” 2013. [Online]. Available: <http://www.cartrack.pt/about-us/>. [Accessed: 15-Oct-2013].
- [2] Cartrack, “Segurança e Recuperação de Viaturas | Recuperação | Cartrack,” 2013. [Online]. Available: <http://www.cartrack.pt/stolen-vehicle-recovery/>. [Accessed: 15-Oct-2013].
- [3] Cartrack, “Gestão de Frotas | Monitoramento da frota | Cartrack,” 2013. [Online]. Available: <http://www.cartrack.pt/fleet-management/>. [Accessed: 15-Oct-2013].
- [4] Tfleet, “TFleet - Funcionalidades,” 2011. [Online]. Available: <http://www.tfleet.eu/tfleetcorp/funcionalidades.aspx>. [Accessed: 15-Oct-2013].
- [5] Masternaut, “MASTERNAUT,” 2009. [Online]. Available: <http://www.masternaut.pt/site2010/offres.php>. [Accessed: 15-Oct-2013].
- [6] Masternaut, “Geoloc,” 2009. [Online]. Available: <http://www.masternaut.pt/site2010/medias/pdf/pack-geoloc.pdf>. [Accessed: 15-Oct-2013].
- [7] Masternaut, “Geonav,” 2009. [Online]. Available: <http://www.masternaut.pt/site2010/medias/pdf/pack-geonav.pdf>. [Accessed: 15-Oct-2013].
- [8] Masternaut, “Ecocan,” 2009. [Online]. Available: <http://www.masternaut.pt/site2010/medias/pdf/pack-ecocan.pdf>. [Accessed: 15-Oct-2013].
- [9] 3dtracking, “3dtracking - 3dFind,” 2007. [Online]. Available: <http://3dtracking.com.pt/3dfind.aspx>. [Accessed: 15-Oct-2013].
- [10] 3dtracking, “3dtracking - 3dProtect,” 2007. [Online]. Available: <http://3dtracking.com.pt/3dprotect.aspx>. [Accessed: 15-Oct-2013].
- [11] 3dtracking, “3dtracking - 3dTrack,” 2007. [Online]. Available: <http://3dtracking.com.pt/3dtrack.aspx>. [Accessed: 15-Oct-2013].
- [12] 3dtracking, “3dtracking - 3dReport,” 2007. [Online]. Available: <http://3dtracking.com.pt/3dreport.aspx>. [Accessed: 15-Oct-2013].
- [13] 3dtracking, “3dtracking - 3dAlert,” 2007. [Online]. Available: <http://3dtracking.com.pt/3dalert.aspx>. [Accessed: 15-Oct-2013].
- [14] 3dtracking, “3dtracking - 3dFuel,” 2007. [Online]. Available: <http://3dtracking.com.pt/3dfuel.aspx>. [Accessed: 15-Oct-2013].
- [15] 3dtracking, “3dtracking - 3dTelematics,” 2007. [Online]. Available: <http://3dtracking.com.pt/3dtelematics.aspx>. [Accessed: 15-Oct-2013].

- [16] Frotcom Lusitana, “Empresa | Frotcom,” 2014. [Online]. Available: <http://pt.frotcom.com/empresa>. [Accessed: 02-Apr-2014].
- [17] Frotcom Lusitana, “Fleet Management | Frotcom,” 2014. [Online]. Available: http://pt.frotcom.com/solucoes/gestao-de-frotas/?no_redirect. [Accessed: 02-Apr-2014].
- [18] Inosat, “Localização pessoas, localização viaturas e Gestão de Frotas | Inosat,” 2013. [Online]. Available: <http://www.inosat.pt/inosat/hist%C3%B3ria.aspx>. [Accessed: 15-Oct-2013].
- [19] Inosat, “Nº1 no Combate ao Carjacking | Localizador GPS | Localizador de Carros | Localizador de Viaturas | Inosat Car Locator,” 2013. [Online]. Available: <http://www.inosat.pt/particulares/car-locator-1.aspx>. [Accessed: 15-Oct-2013].
- [20] Inosat, “Gestão de Frotas | Localização de Viaturas | Localização de Carros | Localização de Veículos | Inofrota Start,” 2013. [Online]. Available: <http://www.inosat.pt/empresas/inofrota-start.aspx>. [Accessed: 15-Oct-2013].
- [21] Inosat, “Gestão de Frotas | Localização de Viaturas | Localização de Carros | Localização de Veículos | Inofrota Trace,” 2013. [Online]. Available: <http://www.inosat.pt/empresas/inofrota-trace.aspx>. [Accessed: 15-Oct-2013].
- [22] Inosat, “Gestão de Frotas | Localização de Viaturas | Localização de Carros | Localização de Veículos | Inofrota PRO,” 2013. [Online]. Available: <http://www.inosat.pt/empresas/inofrota-pro.aspx>. [Accessed: 15-Oct-2013].
- [23] Inosat, “Gestão de Frotas | Localização de Viaturas | Localização de Carros | Localização de Veículos | Inofrota NAVIGATOR,” 2013. [Online]. Available: <http://www.inosat.pt/empresas/inofrota-navigator.aspx>. [Accessed: 15-Oct-2013].
- [24] Maxim Integrated, “iButton Standards,” 2008.
- [25] Teltonika, “FM1100 - Flyer,” 2013. [Online]. Available: http://www.teltonika.lt/uploads/docs/FM1100_Flyer_v1_08.pdf. [Accessed: 05-Dec-2013].
- [26] Teltonika, “FM1100 User Manual V0.18,” 2012.
- [27] A. Zúquete, “Cifras simétricas por blocos,” in *Segurança em Redes Informáticas*, 3rd ed., Lisboa: FCA - Editora de Informática, 2010, p. 45.
- [28] J. Henry S. Warren, “Hacker’s Delight,” in *Hacker’s Delight*, 2nd ed., Boston: Addison-Wesley Professional, 2012.
- [29] ATrack Technology Inc., “ATrack Protocol Document,” Taiwan, 2013.
- [30] ATrack Technology Inc., “AT5i | ATrack Technology Inc.” [Online]. Available: <http://www.atrack.com.tw/products/at5i/>. [Accessed: 11-Nov-2013].
- [31] E. Pitt, *Fundamental Networking in Java*. Springer, 2006, p. 381.

- [32] L. Gong, “Java Security Architecture.” [Online]. Available: <http://docs.oracle.com/javase/8/docs/technotes/guides/security/spec/security-specTOC.fm.html>. [Accessed: 24-Oct-2013].
- [33] J. Jenkov, “Java NIO vs. IO.” [Online]. Available: <http://tutorials.jenkov.com/java-nio/nio-vs-io.html>. [Accessed: 30-Mar-2014].
- [34] O’Reilly Media Inc., “Introducing Nonblocking Sockets - O’Reilly Media,” 2002. [Online]. Available: <http://www.onjava.com/pub/a/onjava/2002/09/04/nio.html?page=2>. [Accessed: 05-Jun-2014].
- [35] Google, “Usage Limits and Billing - Google Maps JavaScript API v3 — Google Developers,” 2013. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/usage>. [Accessed: 25-Oct-2013].
- [36] Google, “Google Maps/Google Earth APIs Terms of Service - Google Maps API — Section 9.1,” 2013. [Online]. Available: https://developers.google.com/maps/terms#section_9_1. [Accessed: 25-Oct-2013].
- [37] Google, “Google Maps JavaScript API v3 — Google Developers,” 2013. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/>. [Accessed: 25-Oct-2013].
- [38] Google, “Google Maps JavaScript API V2 Basics - Google Maps JavaScript API v2 (Deprecated) — Google Developers,” 2013. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/v2/introduction>. [Accessed: 25-Oct-2013].
- [39] Google, “Upgrading Your Google Maps JavaScript Application To v3 - Google Maps JavaScript API v2 (Deprecated) — Google Developers,” 2013. [Online]. Available: <https://developers.google.com/maps/documentation/javascript/v2/v2tov3#v3changes>. [Accessed: 25-Oct-2013].
- [40] OpenStreetMaps, “OpenLayers: Home,” 2013. [Online]. Available: <http://openlayers.org/>. [Accessed: 25-Oct-2013].
- [41] Adam Tacy, Robert Hanson, Jason Essington, and Anne Tökke, *GWT in Action*, 2nd ed. Manning, 2013, p. 680.
- [42] GWT Open Source Project, “[GWT] Using GWT RPC.” [Online]. Available: <http://www.gwtproject.org/doc/latest/tutorial/RPC.html>. [Accessed: 15-Nov-2013].
- [43] Jaspersoft Corporation, “Introduction to Jaspersoft Studio | Jaspersoft Community,” 2014. [Online]. Available: <http://community.jaspersoft.com/wiki/introduction-jaspersoft-studio>. [Accessed: 24-Jun-2014].
- [44] S. Rizzi, “Conceptual modelling solutions for the datawarehouse,” *Inf. Sci. Ref.*, pp. 208–227, 2008.

- [45] A. Vukotic and J. Goodwill, *Apache Tomcat 7*. Apress, 2011, p. 296.
- [46] Oracle, “JavaMail API documentation,” 2013. [Online]. Available: <https://javamail.java.net/nonav/docs/api/>. [Accessed: 12-Dec-2013].
- [47] SMSGlobal, “REST API | SMSGlobal.” [Online]. Available: <http://www.smsglobal.com/rest-api/>. [Accessed: 12-Dec-2013].
- [48] Terracotta, “Quartz-2.1.x-Documentation.” [Online]. Available: <http://quartz-scheduler.org/files/documentation/Quartz-2.1.x-Documentation.pdf>. [Accessed: 21-Feb-2013].
- [49] Jaspersoft Corporation, “Jaspersoft Studio User Guide | Jaspersoft Community,” 2013.
- [50] I. Pivotal Software, “RabbitMQ - Producer-Consumer tutorial,” 2014. [Online]. Available: <http://www.rabbitmq.com/tutorials/tutorial-one-java.html>. [Accessed: 04-Mar-2014].
- [51] The PostgreSQL Global Development Group, “PostgreSQL 8.4.21 Documentation,” 2009.
- [52] C. S. Jensen, T. B. Pedersen, and C. Thomsen, *Multidimensional Databases and Data Warehousing*, vol. 2, no. 1. Morgan & Claypool Publishers, 2010, pp. 1–111.
- [53] “Javadoc - Google Maps library for GWT 3.8.0,” 2012. [Online]. Available: <http://gwt-google-apis.googlecode.com/svn/javadoc/maps/3.8/index.html>. [Accessed: 05-Dec-2013].

Anexo A

Base de Dados Operacional

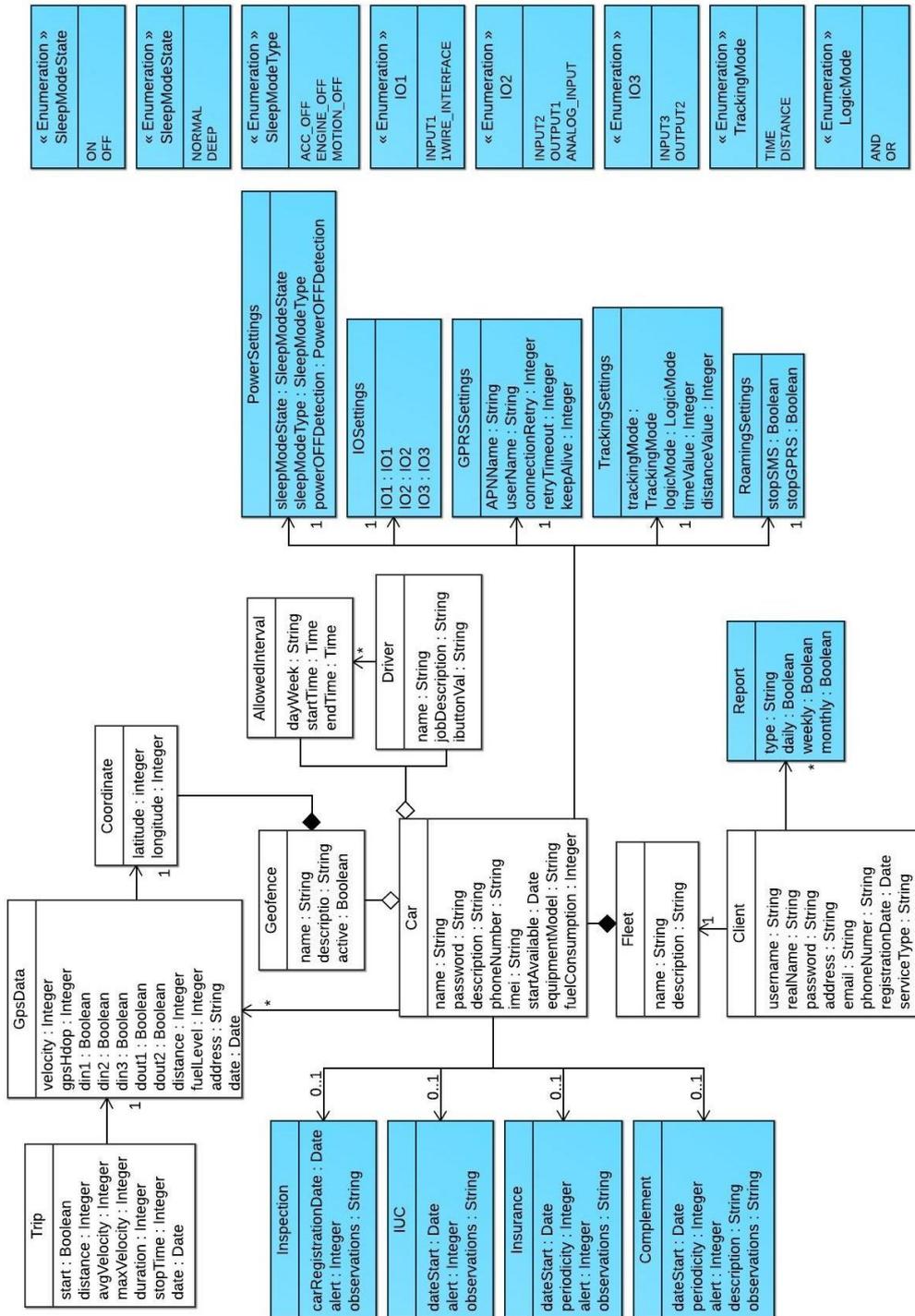


Figura A.1: Diagrama de Classes Completo da Base de Dados Operacional

Anexo B

Configuração da *Agenda Verifier*

```
org.quartz.scheduler.instanceName = MyScheduler
org.quartz.scheduler.instanceId = 1
org.quartz.scheduler.rmi.export = false
org.quartz.scheduler.rmi.proxy = false
org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 1
org.quartz.jobStore.class = org.quartz.simpl.RAMJobStore
org.quartz.plugin.jobInitializer.class =
org.quartz.plugins.xml.XMLSchedulingDataProcessorPlugin
org.quartz.plugin.jobInitializer.fileNames = quartz-config.xml
org.quartz.plugin.jobInitializer.failOnFileNotFound = true
```

Figura B.1: Ficheiro de Configuração `quartz.properties` do *Quartz*

```

<?xml version="1.0" encoding="UTF-8"?>

<job-scheduling-data
  xmlns="http://www.quartz-scheduler.org/xml/JobSchedulingData"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.quartz-
scheduler.org/xml/JobSchedulingData
  http://www.quartz-scheduler.org/xml/job_scheduling_data_1_8.xsd"
  version="1.8">

  <schedule>
    <job>
      <name>AgendaJob</name>
      <group>AgendaGroup</group>
      <description>Send email</description>
      <job-class>com.posisat.agendaevents.SchedulerJob</job-class>
    </job>

    <trigger>
      <cron>
        <name>agendaTrigger</name>
        <job-name>AgendaJob</job-name>
        <job-group>AgendaGroup</job-group>
        <cron-expression>0 0 7 * * ? *</cron-expression>
      </cron>
    </trigger>
  </schedule>
</job-scheduling-data>

```

Figura B.2: Ficheiro de Configuração quartz.properties do *Quartz*

Anexo C

Exemplo de Relatório de Viagens

Relatório de Viagens

de 2014-04-20 até 2014-04-23

POSI-SAT 

Hora Início	Endereço Início	Hora Fim	Endereço Fim	Velocidade Máxima	Velocidade Média	Distância Percorrida	Consumo Combustível	Duração
-------------	-----------------	----------	--------------	-------------------	------------------	----------------------	---------------------	---------

21-CE-56 Sprinter

2014-04-21

15:03:15	Rua Pádua Correia 378, 4405-585 Barroco Vila Nova de Gaia, Portugal	15:14:52	Rua António Francisco de Sousa 162, 4405 Madalena Vila Nova de Gaia, Portugal	88 Km/h	27 Km/h	4 Km 199m	0.55 L	00:11:37	
15:27:19	Rua António Francisco de Sousa 162, 4405 Madalena Vila Nova de Gaia, Portugal	15:33:42	Rua Orfeão 66, 4405 Madalena Vila Nova de Gaia, Portugal	45 Km/h	18 Km/h	0 Km 953m	0.14 L	00:06:23	
						Subtotal	5 Km 152 m	0.69 L	00:18:00

66-AZ-38 Mercedes sprinter

2014-04-21

15:32:35	Rua Entrecampos 421, 4150 Porto, Portugal	15:47:00	Rua Conselheiro Veloso da Cruz 609, Vila Nova de Gaia, Portugal	114 Km/h	46 Km/h	11 Km 92m	1.55 L	00:14:25	
						Subtotal	11 Km 92 m	1.55 L	00:14:25

Hora Início	Endereço Início	Hora Fim	Endereço Fim	Velocidade Máxima	Velocidade Média	Distância Percorrida	Consumo Combustível	Duração
-------------	-----------------	----------	--------------	----------------------	---------------------	-------------------------	------------------------	---------

21-CE-56 Sprinter

2014-04-21

15:43:58	Rua Orfeão 70, 4405 Madalena Vila Nova de Gaia, Portugal	15:54:54	Rua Canastreiros 407, 4400 Canidelo Vila Nova de Gaia, Portugal	36 Km/h	17 Km/h	3 Km 798m	0.49 L	00:10:56
----------	--	----------	---	---------	---------	-----------	--------	----------

						Subtotal	3 Km 798 m	0.49 L	00:10:56
--	--	--	--	--	--	----------	------------	--------	----------

Anexo A

Exemplo de Relatório de Posições

Relatório de

de 02/04/2014 15:20:00 até 02/04/2014

POSI-SAT 

Hora	Endereço	Latitude	Longitude	Velocidade
------	----------	----------	-----------	------------

21-CE-56 Sprinter

02/04/2014

15:20:21	Rua Carregal 39, 4410-279 Canelas, Portugal	41.0644096	-8.6036288	0 Km/h
15:27:29	Rua Carregal 39, 4410-279 Canelas, Portugal	41.0643072	-8.6036624	0 Km/h

66-AZ-38 Mercedes sprinter

02/04/2014

15:28:03	Rua Nova do Monte do Xisto 33, 4460-901 Guifões, Portugal	41.2004448	-8.6708888	0 Km/h
----------	---	------------	------------	--------

Hora	Endereço	Latitude	Longitude	Velocidade
------	----------	----------	-----------	------------

21-CE-56 Sprinter

02/04/2014

15:20:15	Rua Carregal 39, 4410-279, Portugal	41.0644160	-8.6036360	0 Km/h
15:23:21	Rua Carregal 39, 4410-279, Portugal	41.0643552	-8.6036608	0 Km/h
15:26:21	Rua Carregal 39, 4410-279, Portugal	41.0643264	-8.6036904	0 Km/h
15:28:29	Rua Carregal 39, 4410-279, Portugal	41.0642624	-8.6036376	0 Km/h
15:29:29	Rua Carregal 39, 4410-279, Portugal	41.0642560	-8.6036432	0 Km/h

66-AZ-38 Mercedes sprinter

02/04/2014

15:22:21	Rua Nova do Monte do Xisto 33, 4460-901, Portugal	41.2004448	-8.6708888	0 Km/h
15:25:21	Rua Nova do Monte do Xisto 33, 4460-901, Portugal	41.2004448	-8.6708888	0 Km/h
15:29:04	Rua Nova do Monte do Xisto 33, 4460-901, Portugal	41.2004448	-8.6708888	0 Km/h

Hora Endereço Latitude Longitude Velocidade

83-MT-55 Renault Master

02/04/2014

15:20:06	IP1, 3800, Portugal	40.6150848	-8.5431224	153 Km/h
15:20:30	IP1, 3850, Portugal	40.6229568	-8.5371640	150 Km/h
15:20:56	IP1, 3850, Portugal	40.6308544	-8.5314160	126 Km/h
15:21:27	IP1, 3850, Portugal	40.6384800	-8.5245984	127 Km/h
15:21:56	IP1, 3850, Portugal	40.6469600	-8.5196248	130 Km/h
15:22:24	IP1, 3850, Portugal	40.6560160	-8.5175752	134 Km/h
15:22:52		40.6649984	-8.5150400	136 Km/h
15:23:18	IP1, 3850, Portugal	40.6734720	-8.5191752	139 Km/h
15:23:45	IP1, 3850 Albergaria-a-Velha, Portugal	40.6824736	-8.5184584	129 Km/h
15:24:15	IP1, 3850-184 Albergaria-a-Velha, Portugal	40.6911232	-8.5218608	129 Km/h
15:24:46	IP1, 3850-184 Albergaria-a-Velha, Portugal	40.7019840	-8.5226368	137 Km/h
15:25:07	IP1, 3865, Portugal	40.7092672	-8.5215256	141 Km/h
15:25:38	IP1, 3865, Portugal	40.7184576	-8.5223600	92 Km/h
15:26:09	IP1, Portugal	40.7274816	-8.5222408	131 Km/h
15:26:37	IP1, 3850, Portugal	40.7367136	-8.5242088	146 Km/h
15:27:02	IP1, 3865, Portugal	40.7453344	-8.5286424	146 Km/h
15:27:28	IP1, 3865, Portugal	40.7534112	-8.5346304	141 Km/h
15:27:54	IP1, 3860 Estarreja, Portugal	40.7617536	-8.5399920	144 Km/h

Hora	Endereço	Latitude	Longitude	Velocidade
15:28:21	IP1, 3860 Estarreja, Portugal	40.7706464	-8.5436032	135 Km/h
15:28:50	IP1, 3860 Estarreja, Portugal	40.7796672	-8.5436392	99 Km/h
15:29:05	IP1, 3860 Estarreja, Portugal	40.7824224	-8.5432360	64 Km/h