

UNIVERSIDADE DE LISBOA  
FACULDADE DE CIÊNCIAS  
DEPARTAMENTO DE INFORMÁTICA



## **SMART CITY BUS: FLEXIBLETRANSPORT.AI**

Alexandre José Rodrigues Cabral

**Mestrado em Engenharia Informática**  
Especialização em Engenharia de Software

Versão Pública

Trabalho de Projeto orientado por:  
Professor Doutor José Manuel da Silva Cecílio

2019



# Agradecimentos

O maior agradecimento é para os meus pais e irmã, que sempre fizeram de tudo para eu seguir aquilo que gosto, pelos valores, pela força e pelos conselhos que me passaram e permitiram chegar a esta fase académica. Graças a eles consegui manter sempre o foco, mesmo em momentos mais baixos da vida. Quero agradecer ainda a outros elementos da família que ajudaram também a ser quem sou hoje.

Gostaria ainda de agradecer ao orientador Professor Doutor José Cecílio, por me ter cativado e ter tido a disponibilidade necessária para a realização deste trabalho. Sem ele não seria possível a elaboração deste relatório.

Agradeço também aos amigos que conheci ao longo do meu percurso académico, foram eles que me ajudaram a chegar a esta fase e contribuíram ativamente para o meu sucesso.

Deixo ainda uma palavra à instituição onde o projeto foi realizado, CARD4B, e a todos os seus elementos, por terem uma disponibilidade incrível para me ajudar e me integrarem tão depressa no espírito da empresa, nomeadamente ao Jorge Mendes e ao João Almeida.

Por fim agradecer à Faculdade de Ciência da Universidade de Lisboa, por me ter dado a oportunidade de frequentar a instituição, onde foram adquiridos conhecimentos na área da informática, mas também valores para a vida que permitiram a chegada a esta fase académica.



## Resumo

Este relatório foi realizado no âmbito da disciplina de Dissertação/Projeto de Engenharia Informática do 2º ano de Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa e reporta o trabalho desenvolvido no âmbito do projeto *SMART CITY BUS: FLEXIBLETRANSPORT.AI* na empresa Card4B – Systems, S.A, projeto que está relacionado com a temática dos transportes públicos e pretende desenvolver um conjunto de mecanismos e soluções que têm por objetivo melhorar o funcionamento destes.

Nos dias de hoje, os serviços públicos de mobilidade em meios urbanos (serviços de transportes) são maioritariamente divididos em dois tipos, antagónicos: serviços coletivos regulares com horários e percursos pré-definidos (ex. carreiras da Carris) e serviços a pedido (ex. táxis, Uber). A flexibilidade dos serviços a pedido tem um acréscimo de preço significativo face aos serviços coletivos regulares, o que restringe a sua utilização por vastas camadas da população.

Com as tecnologias existentes nas áreas de informação e de comunicação começa a ser possível obter informação sobre os movimentos e a concentração dos cidadãos na malha urbana. Estas tecnologias permitem ainda gerir de forma mais dinâmica grandes frotas de veículos de transportes, pelo que é expectável o aparecimento de serviços coletivos com trajetos e horários mais flexíveis, dinâmicos e ajustados aos padrões de mobilidade dos utilizadores deste tipo de transporte.

Este trabalho consistiu na implementação e alteração de um sistema de transportes inteligentes (STI), que permite a um utilizador executar a função pretendida sem dificuldades no menor tempo possível, com uma arquitetura modular, permitindo a independência de módulos e uma instalação simples do sistema, com módulos de *back-end* e *frond-end*.

As implementações e alterações efetuadas originam melhoramentos no sistema, como uma troca de mensagens mais rápida por parte de um *WebService*, novas funcionalidades em vários módulos e ainda um conversor que permite a integração de dados no sistema, sendo estes dados provenientes de diferentes empresas clientes.

O trabalho realizado permite a compra de bilhetes em papel por parte dos motoristas e das bilheteiras, e ainda a validação de passes em dispositivos presentes nos autocarros. Este trabalho foi desenvolvido sobre uma plataforma que é explorada comercialmente pela empresa Card4B – Systems, S.A, o que levou a que existissem algumas restrições de divulgação de detalhes e/ou implementações dos diferentes componentes que constituem o sistema. Neste sentido, e de acordo com as regras da Faculdade de Ciências da

Universidade de Lisboa, foram elaboradas duas versões do presente relatório (uma versão pública e outra confidencial). Na versão pública, a atual, o sistema é descrito por meio das suas funcionalidades, sendo omitidos os detalhes dos mecanismos internos, bem como a sua implementação. No que diz respeito à versão confidencial, os aspetos omitidos na versão pública são descritos em detalhe de forma a descrever o trabalho que foi feito durante este estágio.

No que diz respeito à plataforma resultante deste projeto e às funcionalidades implementadas, esta permite um controlo pormenorizado de dados existentes no sistema, permitindo avaliar o sistema atual com base na informação existentes nas variadas bases de dados de cada módulo e comparar o estado do sistema antes e depois das alterações realizadas.

**Palavras-chave:**

Sistema de Transportes Inteligente; Flexibilidade; Mobilidade; Utilizadores.

# Abstract

This report was done in the context of the course of Computer Engineering Dissertation/Project, as part of the 2<sup>nd</sup> year of MSc in Computer Engineering of the School of Sciences of the University of Lisbon. It reports the work developed under the SMART CITY BUS: FLEXIBLETRANSPORT.AI project that ran at the company CARD4B – Systems, S.A. The SMART CITY BUS: FLEXIBLETRANSPORT.AI project is related to public transportation theme and aims to develop a set of mechanisms and solutions to improve the efficiency of public transportation systems.

Nowadays, public mobility services in urban areas (transportation services) are divided in two antagonistic types: regular collective services with predetermined schedules and routes (e.g. Carris buses) and services on request (e.g. Taxis, Uber). The flexibility of on-demand services has a significant price compared with regular collective services, which restricts their use by many sectors of the population.

Based on information and communication technologies, it is now possible to obtain information about the movements and concentration of citizens in the urban network. These technologies also allow a more dynamic management over large fleets of transport vehicles, so it is expected that new services with more flexibility, dynamic routes and schedules will appear, adjusted to user needs and fleets.

The work reported here consisted on the implementation and modification of an intelligent transport system, designed to present a modular architecture, allowing the modules independence and a simple system installation, with backend and front-end modules. Besides those capabilities it presents a user-friendly interaction, allowing users to execute the desired function without difficulties in the shortest possible time.

The implementations and changes lead to improvements in the system, such as faster exchange of messages by a Webservice, new functionalities in several modules and a converter that allows the integration of data in the system, being this data coming from different client companies.

The work done allows the purchase of paper tickets by drivers and ticket offices, as well as the validation of passes inside buses.

This work was developed on a commercial platform that is exploited by the Card4B - Systems, S.A company, which led to some restrictions on the description of details and/or implementations of the different components that make up the system. In this sense, and in accordance with the rules of the Faculty of Sciences of the University of Lisbon, two versions of this report (a public and a confidential version) have been created and

delivered. In the public version, the current one, the system is described through its functionalities, being omitted the details of the internal mechanisms, as well as their implementations. Regarding to the confidential version, all modules, mechanisms and functionalities are described in detail.

In terms of final solution, it allows a detailed control of existing data in the system, allowing to evaluate the current system status based on information coming from the various databases of each module and compare the state of the system before and after the changes made.

**Keywords:**

Intelligent Transport Systems; Flexibility; Mobility; Users.

# Conteúdo

Agradecimentos .....	ii
Resumo .....	iv
Abstract.....	vi
Lista de Tabelas .....	xii
Lista de Gráficos.....	xiv
Lista de Abreviaturas.....	xvi
1. Introdução.....	1
1.1. Objectivos .....	1
1.2. Contribuições do Trabalho.....	2
1.3. Enquadramento Institucional .....	3
1.4. Estrutura do Relatório .....	3
2. Trabalho Relacionado .....	5
2.1. Contextualização.....	5
2.2. Sistemas de Transporte Inteligentes.....	5
2.2.1. Serviços .....	6
2.2.2. Energia.....	6
2.2.3. Meio Ambiente .....	6
2.2.4. Segurança.....	6
2.3. Categorias dos STI.....	7
2.4. Inteligência Artificial nos transportes públicos .....	8
3. Arquitetura para o sistema de transportes inteligentes.....	11
3.1. Requisitos.....	11
3.1.1. Usabilidade .....	11
3.1.2. Arquitetura.....	12
3.1.3. Instalação .....	12
3.2. Arquitetura .....	12
3.3. Ticketing backoffice .....	13
3.3.1. LEGACY CONFIGURATION DATA .....	14
3.3.2. NEW CONFIGURATION DATA .....	14
3.3.3. Conversor .....	14

3.3.4.	Base de Dados .....	14
3.3.5.	Web Service.....	14
3.3.6.	Web Interface .....	14
3.3.7.	Utilizador.....	15
3.4.	Tracking backoffice.....	15
3.4.1.	Base de Dados .....	15
3.4.2.	API.....	16
3.4.3.	Web Interface .....	16
3.4.4.	Utilizador.....	16
3.5.	Planning backoffice .....	16
3.5.1.	Base de Dados .....	16
3.5.2.	Web Interface .....	17
3.5.3.	Utilizador.....	17
3.6.	Driver app .....	17
3.6.1.	Base de Dados .....	17
3.6.2.	Dispositivo Móvel .....	17
3.6.3.	Utilizador.....	18
3.7.	Comunicação entre módulos.....	18
4.	Implementação do sistema de transportes inteligente.....	19
4.1.	Adaptação/conversão Webservice SOAP para REST.....	19
4.2.	Expansão do módulo Ticketing backoffice .....	19
4.3.	Conversor LEGACY CONFIGURATION DATA para NEW CONFIGURATION DATA .....	20
4.4.	Expansão Tracking backoffice.....	20
5.	Avaliação do sistema em execução.....	21
5.1.	Serviços e turnos realizados.....	21
5.2.	Vendas e validações.....	22
5.3.	Delay posição real autocarro – posição Tracking backoffice .....	23
5.4.	Comparação tempos SOAP vs REST .....	24
5.5.	Arranque do Driver app .....	25
6.	Conclusões e trabalho futuro.....	27
6.1.	Sumário do trabalho realizado .....	27
6.2.	Trabalho futuro .....	28
6.2.1.	Ticketing backoffice .....	28

6.2.2. Tracking backoffice .....	29
Bibliografia.....	31



## Lista de Tabelas

Tabela 2.1 - Aplicações dos SIT (Chen & Miles, 1999) .....	8
Tabela 5.1 - Vendas e validações .....	23
Tabela 5.2 - Tempos de comunicação Driver app - Tracking backoffice .....	23
Tabela 5.3 - Dados sobre replicações totais .....	25
Tabela 5.4 - Dados sobre replicações parciais.....	26



## Lista de Gráficos

Gráfico 5.1 - Serviços e turnos por dia.....	21
Gráfico 5.2 - Duração média de serviços e turnos por dia .....	22
Gráfico 5.3 - tempos SOAP vs tempos REST.....	24



# Lista de Abreviaturas

API	Application Programming Interface.
BD	Base de Dados.
CV	Controlo de Veículos.
GPS	Global Positioning System.
GT	Gestão de Tráfego.
IA	Inteligência Artificial.
ISA	International Organization for Standardization.
IV	Informação para Viajantes.
OVC	Operação de Veículos Comerciais.
REST	Representational State Transfer.
RITA	Real-time Intelligent Transport Assistant.
RTA	Roads and Transport Authority.
SOAP	Simple Object Access Protocol.
STI	Sistemas de Transporte Inteligentes.
TfL	Transport for London.
TP	Transportes Públicos.
TPA	Terminal de Pagamento Automático.



# 1. Introdução

No Inquérito à Mobilidade nas Áreas Metropolitanas do Porto e de Lisboa, realizado pelo Instituto Nacional de Estatística em 2017 (Instituto Nacional de Estatística, 2017), foi pedido aos utilizadores frequentes de transportes públicos que classificassem um conjunto de características sobre estes em cada região.

Numa escala de 1 a 6 (sendo 1 a pior classificação e 6 a melhor) os utilizadores classificaram as seguintes características: proximidade à rede (paragens), segurança, sistema tarifário e bilhética (diversidade de bilhetes e passes), conforto/ comodidade/ limpeza, informação ao público, lotação, facilidade de transbordo (mesmo operador ou para outro), duração do percurso/ rapidez, qualidade dos veículos/ frota, frequência dos serviços, horários de período de funcionamento, fiabilidade/ pontualidade e acesso por pessoas portadoras de deficiência e preço/custo do transporte público.

Nos resultados obtidos, para a região do Porto, verificou-se que a lotação (com uma média de 3,44) e a frequência de serviços (média de 3,49) obtiveram as duas piores classificações. De destacar que existe ainda um pouco de insatisfação com a qualidade dos veículos/frota (média de 3,79) e com a duração do percurso/rapidez (média de 3,89). Na região de Lisboa, tanto a lotação (média de 2,90), como a frequência de serviços (média de 3,16) apresentam também os piores resultados.

Resultados, com classificações intermédias, são obtidos quando se consideram a qualidade dos veículos/frota (média de 3,42) e a duração do percurso/rapidez (média de 3,78).

Nestas regiões (Lisboa e Porto) apenas existem "soluções estáticas", ou seja, soluções que apenas fazem com que existam horários e rotas pré-definidas, sem ajustes destes consoante as necessidades em tempo real.

Pelo mesmo inquérito, sabe-se que as pessoas estão descontentes com os atuais serviços prestados, havendo assim uma oportunidade para tentar melhorar um conjunto de características, cativando as pessoas a utilizar mais os transportes públicos.

## 1.1. Objectivos

O projeto *SMART CITY BUS: FLEXIBLIETRANSPORT.AI* tem como objetivo o desenvolvimento duma camada de software de abstracção que possua um conhecimento

global da presença e da procura de serviços numa cidade, tendo por base dados operacionais recolhidos e relacionados entre os diferentes sistemas da cidade. Assim, recorrendo a Inteligência Artificial e a diversas fontes de dados (*multi-sensing*) definiram-se os seguintes objetivos para este projeto:

- Melhorar o *back-office* do Sistema de Transportes Inteligentes, mantendo a estrutura modular já existente;
- Melhorar comunicações existentes entre módulos, permitindo uma troca de informação menos demorada;
- Permitir ao utilizador a consulta de nova informação, melhorando assim a gestão dos recursos da empresa correspondente.

Estes objetivos serviram como base para o desenvolvimento do sistema de software necessário à interligação de diferentes fontes de dados de forma a construir uma solução ajustável às necessidades num dado momento.

Esta necessidade de adaptação de um sistema já existente provém do facto de existir uma empresa cliente (portanto, uma empresa de transportes públicos, nomeadamente autocarros) a querer mudar a forma como atualmente têm o seu negócio, adaptando-se ao complexo sistema já existente na empresa.

## 1.2. Contribuições do Trabalho

Antes da adaptação do sistema, foi necessário listar os problemas que estas alterações iriam implicar:

- Adaptação da empresa cliente ao sistema (e conseqüentemente aos módulos que o compõem);
- Adaptação do sistema às restrições provenientes da empresa cliente;

Estas adaptações seriam necessárias para a continuação da proposta, mas era também necessário que estas adaptações ocorressem de tal forma que permitisse a continuidade do funcionamento da plataforma existente (visto que se encontra a ser utilizado por outras empresas cliente).

Assim, foi pensada uma solução contendo várias funcionalidades (algumas destas propostas pela empresa cliente e outras já existentes), nomeadamente:

- Gestão;
- Informação;
- Contabilização;
- Eficiência dos transportes.

No âmbito desta tese, e no contexto do projeto em que esteve enquadrada, as principais contribuições residem em:

- Adição de nova informação ao módulo *back-office* responsável pela bilhética;
- Alteração da comunicação existente entre o módulo de *front-end* e o módulo de *back-office* responsável pela bilhética. Esta contribuição residiu na alteração de um Web Service SOAP para REST;
- Implementação de um conversor de ficheiros que permite a conversão/adaptação de formatos de dados provenientes de empresas cliente;
- Implementação de um algoritmo de correspondência de paragens;
- Adição de novas formas de visualização dos dados presentes no módulo *back-office* responsável pelo posicionamento do autocarro;
- Adição de um serviço de mensagens que permite a troca de alertas entre o módulo *front-end* e o módulo *back-office* responsável pelo posicionamento do autocarro.

### 1.3. Enquadramento Institucional

A realização deste trabalho enquadra-se no âmbito da disciplina Dissertação/Projeto de Engenharia Informática (DPEI) da Faculdade de Ciências da Universidade de Lisboa (FCUL), sendo esta disciplina um requisito para a conclusão do Mestrado em Engenharia Informática (MEI). O trabalho foi desenvolvido numa instituição de acolhimento externa, neste caso uma empresa privada, Card4B – Systems, S.A.

### 1.4. Estrutura do Relatório

O relatório encontra-se organizado em capítulos, sendo que dentro de cada capítulo podem existir várias secções, de maneira a que o documento disponibilize uma leitura facilitada ao leitor. Os capítulos principais estão divididos da seguinte forma:

- Introdução – introdução e motivação para a realização do trabalho, do contexto no qual o trabalho se enquadra, as contribuições do mesmo e o enquadramento institucional do relatório;
- Trabalho relacionado – contextualização do trabalho baseada na identificação de trabalhos semelhantes que permitem compreender os requisitos e necessidades de uma plataforma de gestão para sistemas de transportes inteligentes;
- Arquitetura para o sistema de transportes inteligentes – descrição detalhada da arquitetura/sistema sobre o qual o trabalho foi desenvolvido;
- Implementação do sistema de transportes inteligente – detalhe dos módulos e mecanismos desenvolvidos para acomodar os requisitos do sistema;
- Avaliação do sistema em execução – demonstração de dados estatísticos sobre a utilização e performance do sistema perante as alterações realizadas no contexto deste trabalho;

- Conclusões e trabalho futuro – identificação das principais conclusões extraídas da realização deste projeto e identificação de novas necessidades que deverão ser tidas em consideração no trabalho futuro a realizar.

O trabalho reportado neste relatório foi desenvolvido sobre uma plataforma que é explorada comercialmente pela empresa Card4B – Systems, S.A. Neste sentido, e de forma à Card4B – Systems, S.A. se proteger relativamente à concorrência comercial que está sujeita, este relatório está sujeito a certas restrições de confidencialidade, nomeadamente em termos de divulgação de detalhes e/ou implementações dos diferentes componentes que constituem o sistema. Neste sentido, e de acordo com as regras da Faculdade de Ciências da Universidade de Lisboa, foram elaboradas duas versões do presente relatório (uma versão pública e outra confidencial). Na versão pública, a atual, o sistema é descrito por meio das suas funcionalidades, sendo omitidos os detalhes dos mecanismos internos, bem como a sua implementação. No que diz respeito à versão confidencial, os aspetos omitidos na versão pública são descritos em detalhe de forma a descrever/documentar o trabalho desenvolvido durante este estágio.

## 2. Trabalho Relacionado

### 2.1. Contextualização

Devido ao aumento da mobilidade urbana, a quantidade de pessoas a utilizar os transportes públicos para chegarem ao seu destino aumentou (Silva, 2003). Este aumento influencia o número de viagens individuais nos transportes públicos, e provém do facto da sociedade estar a alterar o seu comportamento (Ferreira, 2004). Dada a litoralização existente e ao facto de 40% da população morar a menos de 100 quilómetros da costa (PERCENTAGE OF TOTAL POPULATION LIVING IN COASTAL AREAS, 2016), os transportes públicos dessas regiões têm de acomodar este acréscimo de população num menor raio de intervenção.

O aumento da concentração da população nas mesmas zonas arrecada dificuldades, dado que existirão assim mais pessoas a utilizar tanto o transporte público como o transporte privado para as suas deslocações. Estes dois tipos de transporte influenciam o quotidiano não só de quem os utiliza, mas também de quem os conduz e até de quem necessita dos seus serviços. Assim, é possível concluir que os transportes têm uma grande importância no dia-a-dia da população (Sussman, 2000).

Por outro lado, os sistemas de informação têm vindo a crescer, revelando-se uma mais valia para a mobilidade urbana. Esta evolução tem permitido a interação entre dispositivos heterogéneos, permitindo a recolha de dados em tempo real e a avaliação da mesma para efeitos de monitorização, mesmo com informações em quantidades consideráveis (Perera, Zaslavsky, Christen, & Georgakopoulos, 2015).

### 2.2. Sistemas de Transporte Inteligentes

Os Sistemas de Transporte Inteligentes podem ajudar tanto os condutores dos veículos como os passageiros. Os STI permitem um melhor acesso à informação como o tempo de viagem, o gasto de combustível e as emissões provenientes dos veículos que, sem estes sistemas, não seria facilmente acedida. Estes sistemas permitem ainda o acesso a acidentes e incidentes, podendo assim os motoristas travar atempadamente, em vez de terem de esperar pela ocorrência para começarem a travar (Ghosh & Lee, 2000).

Assim, os STI têm como principal funcionalidade a otimização da gestão dos serviços tanto dos transportes públicos como dos transportes privados. Os STI são centrados em problemas como (Antunes, 2016):

- capacidade de os transportes conseguirem executar os seus serviços atempadamente;
- ineficiência na utilização da energia necessária para a execução dos serviços;
- impacto destes serviços no meio ambiente;
- perigos envolventes na execução destes serviços (segurança rodoviária).

### 2.2.1. Serviços

Os STI, para os motoristas, ajudam com a execução do serviço, dado que permitem a apresentação de informação como trânsito, tempo previsto do serviço e trajetos (sendo que estes podem ser melhorados em tempo real em caso de estrada cortada). Para os passageiros os STI permitem saber os horários com os respetivos atrasos, dando assim uma maior liberdade para que eles possam escolher previamente o transporte público a usar (diminuindo assim o tempo desperdiçado). Os STI permitem também criar percursos dinâmicos, ou seja, criados para o típico movimento do passageiro, tornando mais prático e aplicado ao quotidiano da generalidade dos passageiros. Tal como os percursos dinâmicos, os STI permitem a criação de horários dinâmicos, de modo a haver mais recursos nas horas mais solicitadas (as denominadas “horas de ponta”).

### 2.2.2. Energia

Com estes sistemas, os condutores conseguem ter acesso a informação que antes não era possível e assim obter mais dados sobre o estado atual da viatura. Se nessa nova informação constatar dados como o estado de peças mecânicas ou o combustível gasto desnecessariamente (com possíveis acelerações e travagens repentinas), o condutor poderá adaptar o seu estilo de condução e assim poupar efetivamente combustível.

### 2.2.3. Meio Ambiente

Assumindo que o consumo de combustível está diretamente relacionado com a emissão de dióxido de carbono (CO<sub>2</sub>) para a atmosfera, os STI influenciam positivamente o meio ambiente. Tal acontece, porque eles permitem a poupança de combustível por parte dos automobilistas (avisando sobre possíveis acidentes ou incidentes e propondo novas rotas). Esta redução de emissões de gases prejudiciais para a atmosfera é conseguida por meio de sistemas que indicam aos condutores os trajetos mais curtos para chegarem ao seu destino (evitando grandes aglomerados de trânsito e o uso de rotas com maior distância); sistemas que avisam os condutores da disponibilidade de lugares de certos parques de estacionamento (poupando assim o uso de combustível gasto à procura de lugares); e de sistemas que, ao melhorarem a informação disponível para os utilizadores dos variados tipos de transporte público, fazem com que estes tenham uma maior afluência.

### 2.2.4. Segurança

Com o aparecimento dos STI, também a segurança rodoviária melhorou. Os STI introduziram novos sistemas capazes de assistir a condução como por exemplo, na

mudança de faixa de rodagem e na prevenção de acidentes ou na detecção de cansaço ou sonolência do condutor. Os STI ajudam também os condutores indiretamente, visto que permitem aos controladores das infraestruturas a obtenção de nova informação como disposição de informação em painéis (sobre condições climatéricas ou tráfego congestionado), controlo do tráfego existente e detecção de possíveis acidentes. Assim, os sistemas permitem um escoamento de tráfego mais eficiente, melhorando as condições de condução.

### 2.3. Categorias dos STI

Com o aparecimento dos STI, a *International Organization for Standardization (ISA)* estabeleceu um conjunto de categorias (Figueiredo, 2005) referentes às aplicações envolvidas. A Tabela 2.1 faz a correspondência entre as categorias e as suas aplicações (Chen & Miles, 1999).

<b>Categorias</b>	<b>Aplicações</b>
Gestão de Tráfego (GT)	Apoio ao planeamento de transportes
	Controlo de Tráfego
	Gestão de incidentes
	Gestão da procura
	Política de regulamentos de trânsito
	Gestão da manutenção de infraestruturas
Informação para Viajantes (IV)	Informação antes da viagem
	Informação durante a viagem
	Informação durante a viagem em transporte público
	Serviços de informação pessoais
	Planeamento de rotas e apoio à navegação
Sistemas Avançados de Controlo de Veículos (CV)	Melhoramento da visão
	Operação de veículos automáticos
	Impedimento de colisão longitudinal
	Disponibilização de segurança
	Difusão de restrições pré colisão
Operação de Veículos Comerciais (OVC)	Distância de folga de veículos comerciais
	Processos administrativos de veículos comerciais
	Inspeção automática da segurança das vias
	Monitorização da segurança dentro dos veículos comerciais
	Gestão de frotas de veículos comerciais
Transportes Públicos (TP)	Gestão de transportes públicos
	Gestão da procura/resposta de transportes
	Gestão partilhada de transportes

Gestão de emergências	Notificação de emergências e segurança pessoal
	Gestão de veículos de emergência
	Notificação de incidentes e matérias perigosos
Pagamento eletrónico	Transações financeiras eletrónicas
Segurança	Segurança das viagens públicas
	Aumento da segurança para utilizadores idosos ou com deficiência
	Junções inteligentes

Tabela 2.1 - Aplicações dos SIT (Chen & Miles, 1999)

## 2.4. Inteligência Artificial nos transportes públicos

De forma a tornar os STI mais eficientes e alargar o conjunto de serviços disponíveis, os STI passaram a integrar mecanismos de Inteligência Artificial (IA). Com o recurso à IA podemos prever possíveis perturbações como, em casos de emergência, melhorando a vida dos utilizadores dos transportes. Indiretamente permite melhorar o fluxo de transportes nas grandes cidades, acabando por melhorar também o dia-a-dia no ambiente urbano para não-utilizadores dos serviços de transportes. A Inteligência Artificial permite usar dados de telemetria dos transportes (como o desgaste dos travões, a quilometragem dos pneus, a temperatura do motor, o nível do óleo), para antever a possível necessidade de manutenção dos mesmos, melhorando a segurança e a fiabilidade.

A melhoria dos horários e das rotas seria uma outra consequência da aplicação de IA. O que permite aumentar a eficiência e produtividade dos transportes e reduzir os tempos de espera dos utilizadores. Usando dados como o tempo entre cada paragem e o tempo gasto em cada uma delas, é possível tornar os transportes públicos mais fiáveis, no sentido em que a informação é disponibilizada aos clientes, nomeadamente o tempo de espera, que passa a ser mais preciso. Se for usada a capacidade de *machine-learning* da Inteligência Artificial, é ainda possível melhorar os horários já existentes, usando dados provenientes dos atrasos observados anteriormente, cuja causa é atribuída a incidentes.

A IA contribui para o melhoramento dos horários em tempo real, tendo como base o número de pessoas à espera, e a diminuição do trânsito (principalmente dentro das cidades).

Um típico STI consiste numa frota de autocarros que enviam dados sobre os mesmos, em tempo real, para um centro de controlo. Um STI destes tem como intuito ajudar os operadores de um centro de controlo nas suas decisões, nomeadamente para resolver problemas numa frota de autocarros, no meio urbano. Este sistema tem como objetivo principal tornar mais eficiente o funcionamento deste tipo de serviço, de modo a respeitar os horários dos condutores, garantir a segurança do serviço prestado e minimizar o desconforto dos passageiros e dos condutores.

Para garantir o sucesso deste sistema, foi necessário perceber primeiro as funções dos operadores que iriam utilizar o mesmo. As funções dos operadores foram divididas em três categorias: diagnosticar, prever e planejar. A primeira consiste em identificar a presença de situações anormais para depois as classificar em relação ao seu nível de gravidade, tendo como base informação proveniente dos veículos (como a sua localização e o seu estado) em tempo real. A função de prever faz uma estimativa do impacto que uma anomalia terá nos serviços, usando também informação enviada pelo autocarro, como o número de passageiros, e informação histórica, como o comportamento dos veículos e a quantidade de passageiros nas paragens seguintes. A última tem como objetivo descobrir as ações apropriadas face ao problema encontrado, de maneira a melhorar o serviço de transporte.

Para este assistente fornecer um suporte adequado à gestão dos transportes públicos, é necessário que simule o processo de pensamento natural dos operadores, ou seja, que processe as três funções praticadas pelos operadores, com o maior nível de precisão possível.

Na função de diagnóstico o assistente produz valores qualitativos em relação à informação recolhida pela central de controlo, de modo a que, posteriormente, consiga determinar o tipo de problema existente com base numa classificação previamente definida.

Na segunda função, o assistente estima o número de pessoas em cada paragem, com base em dados históricos, e simula o movimento dos autocarros atrasados para determinar que paragens irão ser afetadas. Por fim, calcula o impacto dos atrasos e o número de pessoas afetadas. Estes valores são armazenados e eventualmente aproveitados quando surgirem situações de conflitos diferentes em linhas diferentes, mas que podem exigir soluções semelhantes. Antes da execução da última tarefa, existe já uma hierarquia de objetivos e um plano com as medidas a tomar para a correção dos problemas detetados.

Na última função, o assistente vai melhorar o plano já existente, subindo na hierarquia de objetivos até chegar a objetivos que não serão alcançados. O assistente vai então gerar um plano final, estendendo o existente e preenchendo-o com planos mais pequenos. Esta extensão de plano pode ser baseada em dois métodos (sendo o método escolhido dependente do nível da hierarquia chegado): um método que seleciona um plano usando conhecimento de classificações heurísticas; outro método que constrói o plano usando um algoritmo que depende de certos parâmetros, como a escolha dos veículos de reserva.

Este modelo foi concebido de modo a ser facilmente integrado em várias aplicações e locais. Uma destas aplicações surgiu num projeto europeu denominado *Fluids* (Hernández, Molina, & Cuenca, 2004) e financiado pela Comissão Europeia relativo ao Programa *Telematics Application Programme*, onde um dos objetivos principais consistia em fornecer métodos avançados para melhorar a interação entre o utilizador e o sistema no contexto de suporte de decisões em tempo real. Esta aplicação foi utilizada em Turim,

Itália, e foi mais tarde alterada de modo a poder ser utilizada nos transportes de outras cidades, como foi o caso de Vitoria, Espanha (Molina, 2005).

Nos dias de hoje já existem várias empresas de transportes públicos a usarem Inteligência Artificial para melhorarem os serviços efetuados. A *Transport for London* (TfL) implementou um *chatbot* (denominado *TravelBot*) (Transport for London, 2017) na aplicação Messenger no seu Facebook e que responde às perguntas dos utilizadores sobre os tempos de chegada, os mapas com as rotas existentes e as possíveis atualizações nos serviços. A implementação de Inteligência Artificial neste *chatbot* permite que as respostas fornecidas pelo mesmo sejam mais precisas quantas mais perguntas recebe, visto que vai "aprendendo" com estas.

Para além de operar em Londres, a TfL ajuda ainda outros países, nomeadamente a Austrália, onde existe um *chatbot* denominado RITA (*Real-time Intelligent Transport Assistant*) (Transport for NSW, 2018). Este *chatbot* consegue responder a mais de 300 frequentes perguntas, aconselhar planeamento de viagens e aceitar feedback dos utilizadores sem que sejam preenchidos formulários. Estão previstos melhoramentos neste *chatbot*, nomeadamente a incorporação de mensagens “micro conteúdo”, ou seja, mensagens sobre a segurança e a fadiga do condutor e providenciar informação sobre os destinos em plena viagem (Transport for NSW, 2018).

Na rede de transportes públicos *Roads and Transport Authority* (RTA), nomeadamente nos serviços de táxi a operar no Dubai, estão a criar um *heat map* (Roads & Transport Authority, 2019) sobre as localizações com maior número de utilizadores a requerer os serviços dos taxistas, melhorando o tempo de chegada dos táxis em 9% e aumentando a taxa de reservas em 17%.

O *heat map* funciona através da capacidade de *machine learning* (capacidade de aprendizagem para melhorar progressivamente as suas respostas), tendo por base dois tipos de dados, históricos e em tempo real. Para este sistema ter sucesso, é necessário que cada táxi esteja diretamente ligado ao centro de controlo, o que acontece através de sensores inteligentes, que servirão para receber e enviar dados. O sistema compara o número de utilizadores que pretendem utilizar o serviço de táxi com o número de táxis nas proximidades, para conseguir prever a capacidade de resposta a esses pedidos. Esta informação é depois transformada no *heat map*, que será enviado para os taxistas através dos sensores inteligentes.

Em Hong Kong, a Inteligência Artificial é usada para, através de um algoritmo de análise de horários, maximizar o trabalho possível usando os recursos disponíveis (pessoas, equipamentos, espaço e tempo) (He, Tao, Hou, & Jiang, 2018) . A Suíça, em contraste com os tipos de utilização da Inteligência Artificial até agora, tira partido desta para eliminar o condutor e automatizar a tarefa da condução nos serviços de transportes públicos (operação em fase de testes) (TrapezeTM).

## 3. Arquitetura para o sistema de transportes inteligentes

Neste capítulo descreve-se a arquitetura de um STI desenvolvido pela empresa CARD4B onde este estágio foi realizado. Este sistema é composto por vários módulos, com variadas comunicações entre estes. No entanto, neste relatório apenas são descritos os módulos sobre os quais se trabalhou e os módulos que interagem diretamente com estes.

O capítulo é composto por oito secções onde se começa por uma análise cuidada dos requisitos relevantes para a definição da arquitetura. Na secção seguinte é apresentada a visão geral de todo o sistema, identificando os módulos que a compõem, sendo a descrição dos módulos onde o aluno trabalhou efetuada nas subsecções seguintes. Neste capítulo é também descrita a interação entre os diversos módulos.

### 3.1. Requisitos

Antes do começo da implementação do sistema, foi necessária uma análise de requisitos. Estes requisitos podem ser agrupados em três grupos, a seguir representados:

- Usabilidade – referentes à interação entre o sistema e os vários tipos de utilizador;
- Arquitetura – referentes a características importantes necessárias ao sistema;
- Instalação – referentes às necessidades para a instalação e o funcionamento dos equipamentos presentes nos autocarros.

#### 3.1.1. Usabilidade

Os requisitos de usabilidade são necessários para que, qualquer utilizador a servir-se do sistema, consiga executar a função que pretende sem dificuldades no menor tempo possível. Para isso, é necessário identificar os seguintes requisitos:

- *User friendly* – o utilizador deve ter a capacidade de aceder a qualquer tipo de informação (disponível no módulo a ser utilizado) sem qualquer dificuldade;
- Configurável – o utilizador deve ser capaz de alterar informação presente no *back-end*, através das interfaces disponibilizadas de cada módulo;
- Aprendizagem – o utilizador deve conseguir aprender a utilizar o *front-end* de cada módulo o mais rápido possível;
- Retenção – o utilizador deve adquirir a maior informação possível sobre o uso de cada *front-end*;
- Precisão – o utilizador deve receber a informação a qual tentou aceder.

### 3.1.2. Arquitetura

Os requisitos de arquitetura são mais direcionados ao desenvolvimento do sistema (comparativamente aos requisitos de usabilidade), complementando as normas de arquitetura e desenho deste. De seguida são listados os requisitos identificados:

- Modular – o sistema deverá ser composto por vários módulos, sendo que estes deverão fornecer funcionalidades diferentes e deve ser possível o funcionamento de um módulo sem a dependência de outros;
- Responsivo – o sistema deverá concretizar todos os pedidos executados, respondendo sempre aos pedidos do lado do utilizador;
- Confidencial – o sistema deverá apresentar um *login* e permitir que apenas utilizadores registados tenham acesso ao mesmo;
- Tolerante a falhas – o sistema deverá ser capaz de responder aos pedidos, mesmo tendo ocorrido uma falha;
- Contemporâneo – o sistema deverá verificar a informação presente no mesmo quotidianamente, de modo a que esta informação seja do dia em questão.

### 3.1.3. Instalação

Os requisitos de instalação servem para guiar os processos de instalação dos *tablets* nos autocarros (*front-end*) e as bases de dados e domínios (*back-end*). Estes requisitos são apresentados de seguida:

- Fonte energia (*front-end*) – o *tablet* necessita de estar ligado a um circuito que o permita ter sempre bateria ao longo dos serviços;
- Internet (*frond-end*) – o *tablet* necessita de Internet móvel, rede 3G/4G ou Internet Wi-Fi, de modo a conseguir comunicar com os módulos de *back-end*;
- Global Positioning System (GPS) (*frond-end*) – o *tablet* necessita de captar a posição GPS em tempo real, para a enviar para os módulos de *back-end*;
- Bluetooth (*frond-end*) – o *tablet* necessita de ter a capacidade de comunicar através de Bluetooth, para que consiga interagir com o Terminal de Pagamento Automático (TPA), desde que este respeite o protocolo aplicacional do fornecedor;
- Base de Dados (BD) – a informação de cada módulo precisa de ser guardada em bases de dados, neste caso, relacionais;
- Domínios (*back-end*) – a informação de cada módulo deve ser mostrada organizada aos utilizadores utilizando, neste caso, *Web Interfaces*, que necessitam de domínios de Internet.

## 3.2. Arquitetura

O sistema no qual este projeto se encontra é composto por vários módulos, havendo comunicações entre estes. A arquitetura definida assenta num modelo modular que

permite a que possíveis clientes possam adquirir funcionalidades referentes a cada módulo, ao invés de ter de adquirir funcionalidades para além das requeridas.

O módulo referente ao *front-end* é o Driver app, onde se inclui o *tablet* presente nos autocarros. Este módulo comunica com os módulos Ticketing backoffice e Tracking backoffice que implementam o STI. Foi nestes dois módulos onde incidiram as maiores implementações presentes neste relatório. O Ticketing backoffice serve para fazer a gestão de produtos e serviços. Dispõe de um *Web Interface* para apresentar os dados existentes no *back-office* (organizados de maneira a que qualquer utilizador os consiga perceber). Por sua vez o Tracking backoffice é usado para a gestão de informação sobre as posições dos autocarros (tanto em tempo real como históricas) e sobre os horários. Este também dispõe de um *Web Interface* para visualização da informação. Outro módulo importante é o Planning backoffice, visto que este alimenta, em parte, o Tracking backoffice e que permite a qualquer utilizador verificar os horários existentes.

### 3.3. Ticketing backoffice

O Ticketing backoffice é o módulo de gestão da informação presente na BD. Esta informação é inserida com base em ficheiros provenientes da empresa cliente e pode também ser inserida/alterada através da *Web Interface*. Assim, o Ticketing backoffice apresenta as seguintes funcionalidades:

- Consultar, modificar, adicionar e remover linhas;
- Consultar, modificar, adicionar e remover zonas;
- Consultar, modificar, adicionar e remover paragens;
- Consultar, modificar, adicionar e remover motoristas;
- Consultar, modificar, adicionar e remover viaturas;
- Consultar, modificar, adicionar e remover produtos;
- Consultar, modificar, adicionar e remover tarifas;
- Consultar, modificar, adicionar e remover cartões;
- Consultar, modificar, adicionar e remover caixas;
- Consultar, modificar, adicionar e remover faturação;
- Consultar, modificar, adicionar e remover serviços;
- Consultar, modificar, adicionar e remover vendas;
- Consultar, modificar, adicionar e remover máquinas (tablets).

Com base nas funcionalidades requeridas, desenvolveu-se o módulo Ticketing backoffice que recolhe informação proveniente da(s) empresa(s) clientes, pré-processa a informação e guarda-a numa base de dados. Este módulo inclui também uma *Web Interface* para facilmente se explorar a informação guardada.

### 3.3.1. LEGACY CONFIGURATION DATA

O LEGACY CONFIGURATION DATA é um *data external source* proveniente da empresa cliente. Consiste num ficheiro XML com a informação necessária a alimentar o Ticketing backoffice. Este ficheiro vem com uma organização, que depois é alterada para ser possível importar essa informação para o Ticketing backoffice.

### 3.3.2. NEW CONFIGURATION DATA

O NEW CONFIGURATION DATA é o ficheiro com a informação presente no LEGACY CONFIGURATION DATA, mas organizada de modo a que outros módulos, para além do Ticketing backoffice, a consigam utilizar. Os outros módulos usam este ficheiro para, por exemplo, verificarem se um dado produto (passe, bilhete de papel ou *zapping*) é válido na paragem onde se encontra o autocarro. O Ticketing backoffice faz uso de vários importadores para ler o ficheiro e guardarem a informação na base de dados.

### 3.3.3. Conversor

O conversor que traduz a informação presente no LEGACY CONFIGURATION DATA para o NEW CONFIGURATION DATA foi implementado como parte deste projeto. Para além do ficheiro LEGACY CONFIGURATION DATA, existem outros dois ficheiros dos quais é necessário ir recolher informação: *LineSpine* e *TranslationTable*.

### 3.3.4. Base de Dados

A base de dados do módulo Ticketing backoffice é preenchida de duas formas. A primeira, e principal, consiste no uso dos importadores de informação existentes que leem a informação presente no NEW CONFIGURATION DATA e a guardam na BD. A segunda, menos programática, consegue-se através da *Web Interface*, onde um utilizador consegue alterar/adicionar a informação que ache necessária.

### 3.3.5. Web Service

O módulo Ticketing backoffice apresenta um *Web Service* com a função de comunicação entre a aplicação *front-end* Driver app com o módulo *back-end* Ticketing backoffice. Este *Web Service* está implementado em *SOAP*.

### 3.3.6. Web Interface

A *Web Interface* existente neste módulo permite ao utilizador ver informação presente na BD, como adicionar informação nova e alterar a informação já existente. Existe ainda a possibilidade de, para certos tipos de informação, existir a opção de exportar a informação consultada para um ficheiro Excel, que é depois enviado para o *e-mail* do utilizador. Para esta funcionalidade estar ativa é necessário que o utilizador tenha efetuado o *login* e tenha as permissões necessárias para o fazer.

Esta informação é mostrada de maneira a que o utilizador a consiga entender e consiga realizar as ações pretendidas de maneira a demorar o menos tempo possível. Esta *Web Interface* apresenta uma barra lateral, onde estão agrupados os variados itens por categorias.

Consoante as permissões do utilizador, as categorias e respetivos itens podem ou não estar disponíveis. Alguns dos subtópicos apresentam uma tabela com a informação principal, existindo depois a possibilidade de consultar a informação sobre uma linha dessa tabela.

### 3.3.7. Utilizador

O utilizador presente neste módulo refere-se a qualquer pessoa que queira aceder à informação presente no mesmo, ou seja, será sempre uma pessoa ligada à empresa cliente. Para a execução das tarefas descritas no tópico anterior, é necessário que o utilizador tenha permissões para as aceder, sendo que existem três tipos de utilizadores.

## 3.4. Tracking backoffice

O módulo Tracking backoffice consiste no módulo de gerenciamento da informação *GPS* proveniente do módulo Driver app. Para além da informação sobre a localização, em tempo-real, dos autocarros, o módulo contém também informação sobre os serviços que foram executados e o historial da localização dos autocarros. Esta informação é proveniente do módulo, Planning backoffice. O módulo Planning backoffice faz parte do STI, mas o seu desenvolvimento não faz parte deste trabalho.

O Tracking backoffice inclui as seguintes funcionalidades:

- Consultar a posição do veículo em tempo real;
- Consultar informação sobre o serviço a ser prestado em tempo real;
- Consultar informação sobre os *tablets* presentes nos veículos.

Com as funcionalidades listadas, criou-se então o módulo Tracking backoffice que inclui uma base de dados e uma *Web Interface*.

### 3.4.1. Base de Dados

O módulo Tracking backoffice possui uma base de dados que é preenchida através do módulo Planning backoffice. Todos os dias, uma vez por dia, existe uma sincronização entre as bases de dados dos dois módulos. Esta sincronização é configurável, estando neste momento perto das 00h00, para ser executado antes do horário de exploração da empresa cliente. O Tracking backoffice recolhe toda a informação que necessita do Planning backoffice para o próprio dia e para o próximo dia.

### 3.4.2. API

A comunicação existente entre os módulos Driver app e Tracking backoffice ocorre a partir de uma *Application Programming Interface* (API). Este API é do tipo RESTful, ou seja, utiliza os pedidos HTTP GET, PUT, POST e DELETE.

### 3.4.3. Web Interface

A *Web Interface* presente neste módulo permite ao utilizador consultar informação sobre os serviços prestados e as localizações *GPS* dos autocarros. Contrariamente ao módulo Ticketing backoffice, a *Web Interface* do Tracking backoffice não permite ao utilizador criar, editar ou remover dados.

### 3.4.4. Utilizador

Tal como no módulo Ticketing backoffice, no Tracking backoffice o utilizador, que será alguém dentro da empresa cliente, necessita de inserir as suas credenciais para conseguir ter acesso à informação referida anteriormente. No entanto, ao contrário do Ticketing backoffice, no módulo Tracking backoffice não existe uma hierarquia de permissões de utilizadores – existe antes aquisição, ou não, por parte da empresa cliente, para aceder às informações pretendidas. Caso certas funcionalidades não tenham sido adquiridas, as opções para as mesmas não estarão disponíveis.

## 3.5. Planning backoffice

O Planning backoffice é um módulo do STI no qual não foram efetuadas alterações para a realização do trabalho descrito neste relatório, não obstante foi necessário perceber como está organizado e como interage com os módulos já descritos anteriormente. Este módulo consiste na apresentação e possível alteração de informação relacionada com a calendarização de serviços. Mais concretamente, este módulo permite:

- Consultar, modificar, adicionar e remover épocas;
- Consultar, modificar, adicionar e remover tipos de dia;
- Consultar, modificar, adicionar e remover paragens;
- Consultar, modificar, adicionar e remover viaturas;
- Consultar, modificar, adicionar e remover motoristas;
- Consultar, modificar, adicionar e remover serviços;
- Obtenção de relatórios das informações presentes na base de dados.

### 3.5.1. Base de Dados

Tal como no módulo Ticketing backoffice, no Planning backoffice a base de dados é preenchida de duas maneiras distintas: uma, que consiste em importadores que leem a

informação dos ficheiros fornecidos pela empresa cliente; e uma outra que permite a criação e alteração de dados na *Web Interface*. Esta BD foi, tal como o próprio módulo, implementada por outro colega.

### 3.5.2. Web Interface

O Planning backoffice apresenta também uma *Web Interface* onde é possível ao utilizador criar ou modificar qualquer informação que esteja a ser mostrada. A informação está categorizada numa barra horizontal na parte superior da página principal a partir da qual, consoante as suas intenções, conseguirá aceder às informações pretendidas.

### 3.5.3. Utilizador

Tal como no módulo Tracking backoffice, no Planning backoffice o utilizador será um trabalhador da empresa cliente e terá de ter as suas credenciais para conseguir visualizar a informação que pretende. No entanto, ao contrário do Tracking backoffice, no módulo Planning backoffice não existe uma discriminação de informação, ou seja, o utilizador terá sempre acesso a qualquer informação presente na *Web Interface*, basta que a sua empresa tenha adquirido as capacidades deste módulo.

## 3.6. Driver app

O Driver app é o módulo correspondente ao *front-end* do sistema. O Driver app ao invés de incluir uma *Web Interface* como os outros módulos, este módulo apresenta a sua informação num *tablet* ou num *smartphone* para ser utilizada pelo motorista.

### 3.6.1. Base de Dados

No caso do Driver app, a sua base de dados local é preenchida através de um processo de sincronização tanto com o Ticketing backoffice como com o Tracking backoffice. Ao instalar a aplicação, logo após o login do utilizador e através do *Web Service SOAP*, são feitos vários pedidos de informação, um para cada tabela existente na base de dados local da aplicação (à exceção de tabelas específicas cujo objetivo passa por guardar informação gerada no dispositivo para eventualmente ser enviada para o *back-office*).

### 3.6.2. Dispositivo Móvel

A aplicação pode ser utilizada num *tablet* ou *smartphone*. Funciona como uma consola com dois modos principais: modo de serviço de motorista e modo de bilheteira. Esta consola está emparelhada com um TPA que permite a impressão de bilhetes vendidos a partir da aplicação e talões e, no caso do modo de serviço de motorista, a leitura de cartões para validação ou carregamento dos mesmos através do Driver app. Ainda neste modo, a aplicação tem a capacidade de, a partir da receção de coordenadas GPS, acompanhar em *real time* o percurso de um dado autocarro num determinado horário, tanto no dispositivo

como no *back-office*. As coordenadas GPS são enviadas para o Tracking backoffice de 30 em 30 segundos. Este acompanhamento geográfico também permite “personalizar” as vendas em relação às paragens e à zona.

### 3.6.3. Utilizador

O típico utilizador será um motorista de autocarro que se faz acompanhar de um dispositivo móvel com a aplicação ou um vendedor de bilheteira. Cada conta de utilizador, ao ser criada no *back-office* contém uma configuração que especifica o modo de operação do utilizador, permitindo assim diferenciar entre o modo de bilheteira e o modo de serviço de motorista.

## 3.7. Comunicação entre módulos

Tendo em conta os requisitos enumerados no início do capítulo e as especificações de cada um dos módulos apresentados, as interações existentes com o Driver app foram já descritas anteriormente: o Ticketing backoffice comunica com este através do *Web Service*, de modo a obter a informação necessária, enquanto o Tracking backoffice usa a API para troca de informação. Por fim, é possível concluir que o Planning backoffice alimenta o Tracking backoffice com informação referente a motoristas, serviços e rotas. Por sua vez, o Tracking backoffice e o Ticketing backoffice têm ligações bilaterais com o Driver app que oferece os serviços de gestão de bilhetes. Por outro lado, o Driver app também permite localizar os veículos e representar rotas do serviço. Todas estas interações e trocas de informação são complementadas com informações proveniente das diferentes bases de dados do sistema.

## 4. Implementação do sistema de transportes inteligente

Neste trabalho foram implementados os módulos Ticketing backoffice e Tracking backoffice descritos anteriormente. Nestes dois módulos foram implementadas as *Web Interface* de modo a que os utilizadores da empresa cliente consigam aceder a informação. No Ticketing backoffice foi implementada a base de dados e os *Web Services* usados para comunicar com o módulo Driver app. Neste módulo *back-end* foi ainda implementado o conversor entre LEGACY CONFIGURATION DATA e NEW CONFIGURATION DATA.

### 4.1. Adaptação/conversão Webservice SOAP para REST

O módulo Ticketing backoffice possui um *Web Service*, utilizado para a comunicação deste com o módulo Driver app. Este *Web Service* está implementado usando a arquitetura SOAP e foi alterado para a arquitetura REST (*Representational State Transfer*). A necessidade desta alteração consta no facto de os tempos de resposta teóricos serem menores na arquitetura REST, em comparação com a arquitetura SOAP, permitindo assim uma troca de mensagens entre os módulos Ticketing backoffice e Tracking backoffice menos demorada. Assim, a primeira etapa deste projeto consistiu na transformação do *Web Service* SOAP para um *Web Service* REST.

### 4.2. Expansão do módulo Ticketing backoffice

O desenvolvimento associado ao módulo Ticketing backoffice assentou numa metodologia por etapas de forma a testar todas as funcionalidades e identificar possíveis erros. Estas alterações influenciaram também o módulo Driver app, devido à necessidade de adicionar nova informação à base de dados (colunas novas na base de dados do Ticketing backoffice), levando à necessidade de adaptação do módulo Driver app de forma a contemplar também esta nova informação.

Esta etapa teve o propósito de adquirir conhecimento sobre o Ticketing backoffice, nomeadamente as suas funcionalidades, a organização da sua base de dados e a implementação da *Web Interface*. Assim, estas adaptações começaram por correções de *bugs* existentes na *Web Interface*, como *querys* mal formuladas, elementos da página *Web* em falta ou mal configurados ou a falta de colunas na base de dados, visto a empresa cliente ser um cliente novo e nesta altura o Ticketing backoffice estava configurado para outras empresas clientes.

### 4.3. Conversor LEGACY CONFIGURATION DATA para NEW CONFIGURATION DATA

A tradução do ficheiro LEGACY CONFIGURATION DATA para o ficheiro NEW CONFIGURATION DATA é feita através de um conversor, resultado de uma implementação necessária para a realização deste projeto. Esta conversão é necessária para que outros sistemas, já implementados, consigam ler informação de um ficheiro XML, modificando o mínimo possível estes sistemas (visto estes já estarem em utilização para outras empresas cliente). O começo desta etapa consistiu no mapeamento de informação do LEGACY CONFIGURATION DATA para o NEW CONFIGURATION DATA. Este conversor lê ainda informação de dois ficheiros CSV com informação não presente no LEGACY CONFIGURATION DATA. O primeiro ficheiro CSV (denominado *TranslationTable*) é também proveniente da empresa cliente e contém informação que não é possível demonstrar em ficheiros XML.

### 4.4. Expansão Tracking backoffice

A expansão do módulo Tracking backoffice consistiu na alteração visual de uma funcionalidade e na implementação de duas novas. A funcionalidade melhorada consistiu na apresentação da informação para cada autocarro quando se o selecciona no mapa. As novas funcionalidades consistem na implementação de um *dashboard* para apresentação de um grafo representativo da execução do serviço e de um menu novo para troca de alertas com o motorista.

## 5. Avaliação do sistema em execução

### 5.1. Serviços e turnos realizados

Para se entender o uso do sistema por parte dos motoristas, é possível obter informação tanto sobre o número de serviços e turnos por dia, como a duração dos mesmos (um turno consiste num horário de exploração do motorista, e é constituído por um ou mais serviços). Estes valores são obtidos a partir da análise da base de dados do módulo Ticketing backoffice.

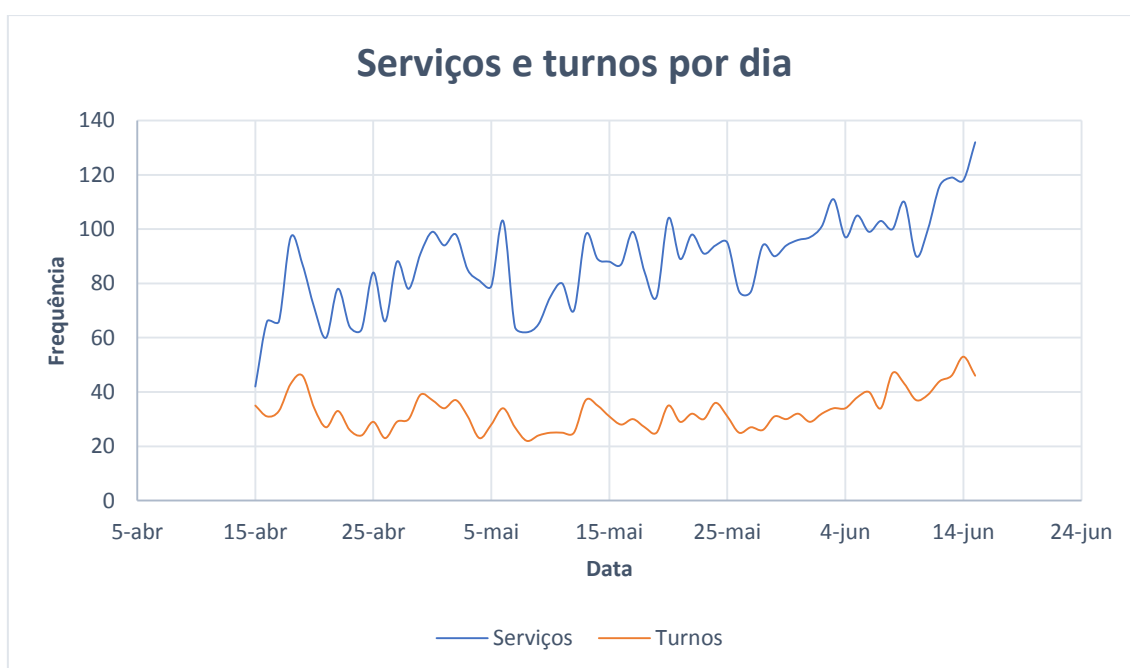


Gráfico 5.1 - Serviços e turnos por dia

O Gráfico 5.1 apresenta informação sobre os números de serviços e turnos por dia. Tal como esperado, o número de serviços é superior ao número de turnos. Em relação ao número de serviços é possível verificar que existe um crescendo dos mesmos ao longo do tempo, muito em parte porque no início a empresa cliente usou o sistema para a realização de testes. O número de turnos não tem variado muito, visto que para a realização dos testes referidos anteriormente era mais importante os resultados sobre os serviços.

É possível verificar o número de serviços por dia através do Gráfico 5.1, podendo-se afirmar que estes se encontram entre os 60 e os 100 por dia. No entanto, com o passar do tempo, estima-se que este valor venha a ser superior (o que já acontece, nomeadamente a partir de junho). Em relação aos turnos, é possível também verificar que estes se encontram entre os 20 e os 40 por dia, sendo que a partir de junho, o valor tenha tendência a aumentar. Este aumento, tanto para os turnos como para os serviços, a partir de junho,

consta no facto de a empresa cliente começar a utilizar o sistema em linhas que antes não o utilizava.

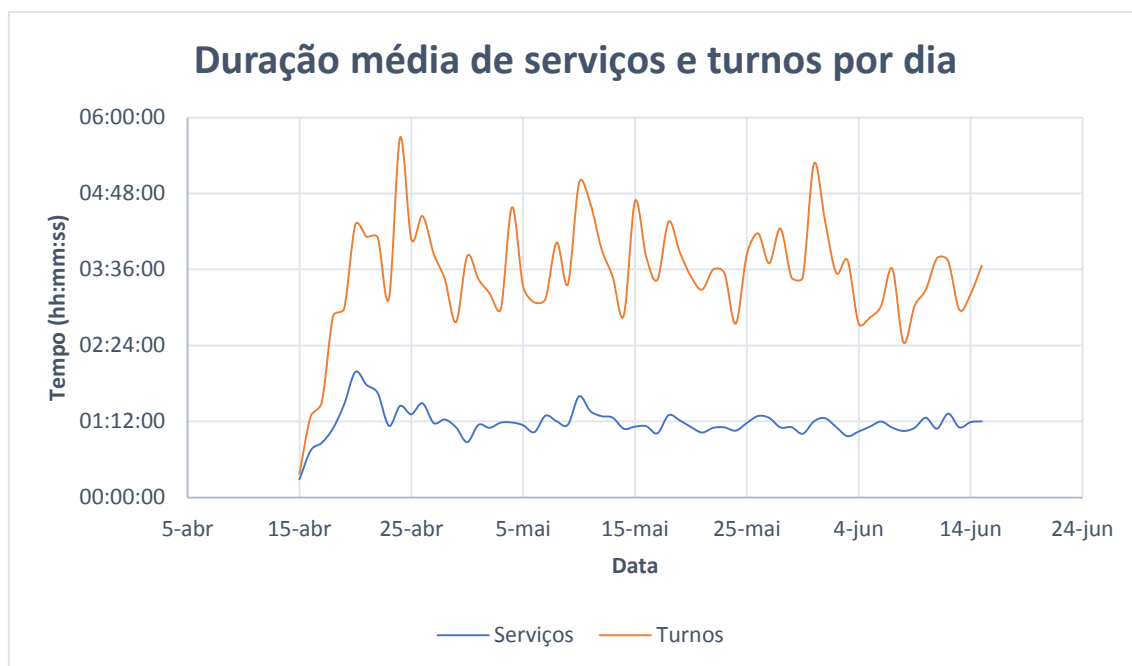


Gráfico 5.2 - Duração média de serviços e turnos por dia

Dado que um turno é constituído por um ou mais serviços e que existiram mais serviços que turnos (Gráfico 5.2), é expectável que o tempo médio de duração dos serviços seja inferior ao tempo médio de duração dos turnos.

Em relação à duração média de serviços é possível verificar que no início existia uma enorme amplitude de valores, dado que foi nesta altura que a empresa cliente realizou os seus testes. Durante a fase de testes o número de turnos foi inferior, visto que poucos turnos davam a realização dos mesmos (tendo estes vários serviços).

Em relação a valores, é possível verificar que os serviços têm uma duração média entre 1h10m e 1h15m (retirando os valores iniciais, de testes). Estima-se que este valor venha a variar mais com a entrada de serviços em novas linhas. Por outro lado, os turnos apresentam valores principalmente entre 2h20m e 5h. Esta discrepância pode ser fundamentada com o facto de existirem turnos com menor duração do que outros (enquanto serviços na mesma linha terão durações semelhantes).

## 5.2. Vendas e validações

O Driver app, para além de auxiliar o motorista, serve também para controlar o número de vendas de bilhetes em papel e de validações de cartões nos serviços executados. Estes valores estão guardados nas tabelas *DayTickets* (bilhetes) e *DayObliterations* (validações). Atualmente apenas bilhetes em papel podem ser vendidos, mas espera-se

que em breve possa também ser possível a qualquer passageiro carregar o seu passe no autocarro.

	<b>Vendas (Bilhetes) / Validações (Cartões)</b>	<b>Média p/ serviço</b>	<b>Desvio-padrão p/ serviço</b>
Bilhetes	120075	27,8	64,9
Cartões	62489	15,1	10,8

*Tabela 5.1 - Vendas e validações*

Pela Tabela 5.1 é possível verificar o número total de bilhetes vendidos, tal como o número total de validações de passes nos serviços executados e ainda a média e o desvio-padrão por serviço para cada um destes. O número total de bilhetes é maior que as validações de cartões dado que no início não existia a hipótese de validar cartões.

Por esta tabela é possível verificar que o número de bilhetes vendidos é aproximadamente o dobro do número de validações de cartões. Verifica-se ainda que a dispersão de vendas de bilhetes é substancialmente maior em comparação com as validações de bilhetes, comparando o desvio-padrão por serviço para cada um destes. Esta diferença deve-se ao facto de existirem serviços nos quais o número de bilhetes vendidos é relativamente maior que os outros (valores, por exemplo, na ordem dos 400 bilhetes vendidos num serviço).

### 5.3. Delay posição real autocarro – posição Tracking backoffice

Para que o utilizador consiga perceber a posição atual do autocarro, a mesma é mostrada o mais fiel à realidade possível na *Web Interface* do módulo Tracking backoffice. Esta posição é conseguida através do Driver app que, de 30 em 30 segundos, envia a sua posição para o Tracking backoffice. Por outro lado, a *Web Interface* atualiza-se, através de *Jquery* e *Javascript*, de 5 em 5 segundos. O primeiro valor foi escolhido para diminuir a probabilidade de serem enviados dados em demasia comparativamente ao que o sistema suporta, enquanto o segundo foi escolhido para informar o utilizador o melhor possível da localização do autocarro.

<b>Tempo (ms)</b>	<b>Repetições</b>	<b>Média (ms)</b>	<b>Desvio-padrão (ms)</b>
[0, 100[	2	76,00	5,66
[100, 200[	84	122,69	12,65
[200, 300[	0	0,00	0,00
[300, 400[	2	330,00	11,31
[400, 500[	5	423,40	22,45
Total	93	142,31	75,33

*Tabela 5.2 - Tempos de comunicação Driver app - Tracking backoffice*

Pela Tabela 5.2 é possível verificar os tempos de comunicação desde o envio da posição GPS até à receção destes por parte do Tracking backoffice. Usando o valor médio, a saber 142,31 milissegundos, e usando os valores referidos anteriormente, é possível calcular que o *delay* existente entre a posição real do autocarro e a posição mostrada no Tracking backoffice varia entre  $0\text{ s} + 142,31\text{ ms} = 0,142\text{ s}$  (quando a *Interface Web* é atualizada imediatamente após a chegada dos dados relativos à posição do autocarro ao Tracking backoffice) e  $30\text{ s} - 5\text{ s} + 142,31\text{ ms} = 25,142\text{ s}$  (quando a *Interface Web* é atualizada imediatamente antes da chegada dos dados relativos à posição do autocarro ao Tracking backoffice).

## 5.4. Comparação tempos SOAP vs REST

A adaptação/conversão WebService SOAP para REST permitiu diminuir o tempo de comunicação entre os módulos Ticketing backoffice e Driver app.

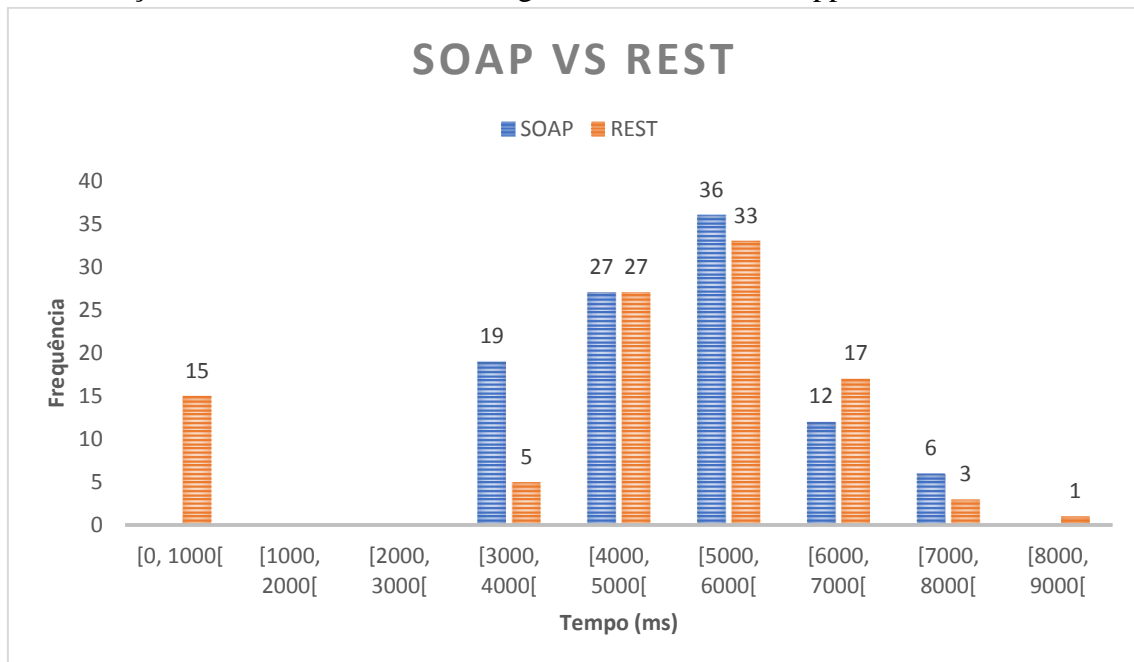


Gráfico 5.3 - tempos SOAP vs tempos REST

O Gráfico 5.3 mostra os tempos de 100 pedidos, nomeadamente o pedido *GetCards* (que serve para obter os cartões existentes na base de dados do *back-office*). Com estes dados é possível calcular a média de valores para cada tipo de envio e resposta:

- SOAP

$$\frac{510685}{100} = 5106,85\text{ ms}$$

- REST

$$\frac{462053}{100} = 4620,53\text{ ms}$$

Em apenas 100 testes (para o mesmo pedido) é possível verificar que a média de valores já apresenta uma diferença de  $5106,85 - 44620,53 = 486,32 \text{ ms}$ . É previsível que, aumentando o número de testes, esta diferença venha a aumentar. Assim, é possível concluir que esta alteração de pedidos SOAP para pedidos REST foi vantajoso tanto para o sistema como para os utilizadores.

## 5.5. Arranque do Driver app

A aplicação Driver app comunica com os módulos *back-office* Ticketing backoffice e Tracking backoffice tanto para obter informação essencial para a execução de serviços (por parte do motorista) como para enviar informações sobre estes no fim de cada serviço e de cada turno.

A obtenção da informação necessária para a execução dos serviços é feita após a autenticação do motorista, ou sempre que se seleccionar a opção para a obtenção da mesma (esta operação é denominada sincronização). Existem dois tipos de sincronização: sincronização total e sincronização parcial. A primeira é executada sempre que se trata de um dispositivo móvel novo, sem qualquer tipo de informação. Já a segunda consiste na atualização de dados, ou seja, apenas se alteram os dados que tenham sido alterados desde a última replicação.

Tempo (s)	Replicação Total											
	Número			Dados Enviados			Dados Apagados			Dados Recebidos		
	Total	Média (s)	D.P. (s)	Total	Média	D.P.	Total	Média	D.P.	Total	Média	D.P.
]0, 3]	27	01.632	00.400	29	01,07	02,42	0	0,00	0,00	201	07,44	09,09
]3, 6]	09	04.184	00.531	261	29,00	43,47	0	0,00	0,00	312	34,67	46,55
]6, 9]	15	07.239	00.709	311	20,73	23,07	2	0,13	0,52	765	51,00	35,49
]9, 12]	11	10.903	00.773	424	38,55	27,63	0	0,00	0,00	812	73,82	42,89
]12, 15]	03	13.640	00.225	151	50,33	39,00	4	1,33	2,31	182	60,67	09,61
]15, 18]	07	16.523	01.760	516	73,71	74,11	2	0,29	0,76	769	109,86	82,25
]18, 21]	03	19.472	00.366	148	49,33	20,26	0	0,00	0,00	258	86,00	08,00
]21, 24]	01	21.873	00.000	57	57,00	00,00	0	0,00	0,00	134	134,00	00,00
]24, 27]	02	26.323	00.430	80	40,00	07,07	0	0,00	0,00	188	94,00	00,00
]27, 30]	01	27.506	00.000	13	13,00	00,00	0	0,00	0,00	887	887,00	00,00
]30, +∞]	21	55.530	58.837	3556	169,33	137,23	8	0,38	1,16	5615	267,38	297,39

Tabela 5.3 - Dados sobre replicações totais

Replicação Parcial												
Tempo (s)	Número			Dados Enviados			Dados Apagados			Dados Recebidos		
	Total	Média (s)	D.P.	Total	Média	D.P.	Total	Média	D.P.	Total	Média	D.P.
]0, 3]	212274	02.224	00.654	26922	00,13	2,35	1495	0,01	0,15	374837	01,77	02,10
]3, 6]	3493	04.988	00.588	16374	04,69	19,86	235	0,07	0,68	29206	08,36	23,80
]6, 9]	1206	08.232	00.729	19562	16,22	31,49	168	0,14	0,67	28111	23,31	46,81
]9, 12]	2484	11.267	00.795	17209	06,93	25,24	102	0,04	0,46	27092	10,91	37,64
]12, 15]	1237	13.890	00.442	14713	11,89	32,78	110	0,09	0,91	22732	18,38	48,66
]15, 18]	602	17.533	01.965	9794	16,27	41,15	78	0,13	1,25	15261	25,35	61,06
]18, 21]	486	19.978	00.723	7193	14,80	40,96	55	0,11	1,34	13698	28,19	75,51
]21, 24]	351	22.043	00.873	4378	12,47	36,89	32	0,09	0,57	10701	30,49	95,39
]24, 27]	194	26.566	00.890	2761	14,23	39,58	04	0,02	0,18	6973	35,94	113,26
]27, 30]	167	28.871	00.354	1578	09,45	31,57	02	0,01	0,15	3725	22,31	72,08
]30, +∞]	1156	58.576	59.632	168421	145,69	1826,64	245	0,21	1,37	65832	56,95	140,74

Tabela 5.4 - Dados sobre replicações parciais

Pela Tabela 5.3 e pela Tabela 5.4 é possível verificar alguns dados sobre os dois tipos de replicação, respetivamente. Para os dois tipos de replicação é possível verificar que a maioria demora menos de 3 segundos para obter e enviar os dados a atualizar. Tal pode ser justificado dado que a consola aplica replicações em *background* (ou seja, sem que o utilizador o faça explicitamente) e assim sempre que o utilizador faz uma replicação, na maioria dos casos a informação existente no *tablet* já se encontra atualizada.

Este facto pode ser corroborado com os valores das médias das mesmas tabelas, dado que estes são inferiores comparando com os outros tempos. É possível também concluir que nas replicações, a troca principal consiste nos dados enviados do Ticketing backoffice para o Driver app, enquanto os dados apagados (entre os mesmos módulos, no mesmo sentido) é a troca na qual menos dados são precisos.

## 6. Conclusões e trabalho futuro

### 6.1. Sumário do trabalho realizado

Este trabalho teve como início a percepção e organização de um sistema de transportes inteligentes. O sistema considerado é constituído por módulos e o trabalho desta tese incidiu principalmente nos módulos Planning backoffice, Ticketing backoffice e Tracking backoffice (módulos de *back-end*). Mais tarde foi preciso entender a organização do Driver app (módulo *front-end*).

O projeto iniciou-se com a percepção de como o Ticketing backoffice está organizado, efetuando alterações ao mesmo (principalmente para correção de bugs). Este começo permitiu verificar a importância deste módulo no sistema, visto ser nele que está presente parte da informação essencial para o motorista conseguir executar os serviços.

De seguida, dentro do mesmo módulo, foi alterado o *WebService* existente, usado para comunicar com o Driver app. Esta alteração consistiu na mudança para *WebService* REST e permitiu melhor a *performance* do Driver app, nomeadamente na recolha dos dados necessários (provenientes do Ticketing backoffice) para a execução dos serviços.

De forma a que o módulo fosse utilizável por empresas-cliente foi implementado o conversor que mapeia os dados do ficheiro XML proveniente da empresa-cliente para dados reconhecíveis ao sistema. Esta implementação começou pelo mapeamento LEGACY CONFIGURATION DATA – NEW CONFIGURATION DATA, de modo a ser possível ir ao ficheiro proveniente da empresa-cliente obter os dados de forma correta.

Após este mapeamento passou-se à implementação do conversor que, com o auxílio de outros dois ficheiros (*TranslationTable* e *LineSpine*), geram o ficheiro NEW CONFIGURATION DATA, cujos dados são reconhecíveis no sistema. Por fim foram efetuadas alterações ao Tracking backoffice a nível da *Web Interface* do mesmo, de modo a existirem mais informações a mostrar ao utilizador. Estas alterações permitem que o utilizador tenha acesso à posição do autocarro num grafo representativo do percurso e permitirá a troca de alertas entre o motorista e o utilizador, quanto o módulo Driver app for adaptado a esta alteração.

Relativamente aos resultados obtidos, nomeadamente sobre os serviços e os turnos dos motoristas, pode-se afirmar que tanto o número de serviços como o número de turnos têm vindo a aumentar, e que estes valores terão tendência a aumentar cada vez mais com a entrada de novas linhas no sistema nas quais se executarão os serviços.

Na duração média destes serviços e turnos verifica-se que a média de serviços não varia muito (o que poderá vir a mudar com a entrada de linhas novas), enquanto os turnos apresentam uma discrepância grande entre eles, dado que os turnos dos motoristas variam entre estes (ou seja, um motorista pode executar um turno de curta duração num dia, e outro de maior duração noutro dia, enquanto nos serviços, a mesma linha, em princípio, terá o mesmo tempo de execução).

Em relação às vendas de bilhetes em papel e às validações dos passes, é possível verificar que, em média por serviço, existe aproximadamente, o dobro de bilhetes vendidos em relação às validações. Estima-se que estes valores se venham a aproximar quando for possível carregar os passes nos próprios autocarros.

Já para a posição do autocarro na Web Interface do módulo Tracking backoffice, calculou-se que existe uma variação de *delay* resultante do tempo de envio dos dados GPS provenientes do Driver app para o Tracking backoffice e do tempo de atualização tanto destes dados no Driver app como da Web Interface no Tracking backoffice.

Os testes efetuados entre o *Web Service* SOAP e o *Web Service* REST do módulo Ticketing backoffice permitiram concluir que, com esta última arquitetura, o motorista obtém os resultados necessários mais rapidamente, permitindo uma melhor execução do serviço (permitindo que seja mais atempada), o que refletirá também no contentamento dos passageiros. Por fim foram efetuadas medições relativamente aos dados trocados entre o Driver app e os módulos *back-end*, Tracking backoffice e Ticketing backoffice.

Estas medições permitiram concluir que, tanto nas replicações parciais como nas replicações totais, a maior troca de dados consiste nos dados enviados do Driver app para os módulos referidos. Permitiram também concluir que a maioria destas replicações demoram de 3 segundos.

## 6.2. Trabalho futuro

Dado o sistema ainda não estar totalmente concluído, vai ser necessário continuar a efetuar alterações ao mesmo até apresentar todas as funcionalidades pretendidas.

### 6.2.1. Ticketing backoffice

Em relação ao Ticketing backoffice, poderão vir a existir dois tipos de alterações: alterações ao módulo em si e alterações à *Interface Web*. O primeiro consiste em alterações à base de dados, enquanto o segundo apenas será necessário se for necessário acrescentar mais informação de forma a tornar a gestão mais eficiente. Estes dois tipos de alteração irão surgir naturalmente, uma vez que novos produtos e linhas irão aparecer e poderão ter configurações ainda não previstas no Ticketing backoffice.

### 6.2.2. Tracking backoffice

No Tracking backoffice, uma vez que este módulo serve para gerir a informação sobre os autocarros, existe a ideia de reformular o mapa existente, de modo a tornar a interação com o utilizador mais eficiente e dinâmica.



## Bibliografia

- Antunes, P. D. (2016). *Sistema Inteligente Transporte do IST*.
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*.
- Chen, K., & Miles, J. C. (1999). *ITS Handbook 2000: Recommendations from the World Road Association (PIARC)*. Artech House.
- Dihua Sun, H. L. (2007). *Predicting Bus Arrival Time on the Basis*.
- Ferreira, S. M. (2004). PLANEAMENTO DE TRANSPORTES COM APOIO DE SISTEMAS DE INFORMAÇÃO GEOGRÁFICA. *A REDE DE TRANSPORTES URBANOS DE BRAGA*.
- Figueiredo, L. M. (2005). *Sistemas Inteligentes de Transporte*. Porto.
- Ghosh, S., & Lee, T. (2000). *Semantic Scholar*. Obtido de <https://pdfs.semanticscholar.org/ee46/e5cdec5547f13d76d1b2d68d35cc9e87a79c.pdf>
- He, S., Tao, S., Hou, Y., & Jiang, W. (2018). *Mass Transit Railway, transit-oriented development, and spatial justice: The competition for prime residential locations in Hong Kong since the 1980s*. Obtido de [https://www.researchgate.net/publication/323725893\\_Mass\\_Transit\\_Railway\\_transit-oriented\\_development\\_and\\_spatial\\_justice\\_The\\_competition\\_for\\_prime\\_residential\\_locations\\_in\\_Hong\\_Kong\\_since\\_the\\_1980s](https://www.researchgate.net/publication/323725893_Mass_Transit_Railway_transit-oriented_development_and_spatial_justice_The_competition_for_prime_residential_locations_in_Hong_Kong_since_the_1980s)
- Hernández, J., Molina, M., & Cuenca, J. (2004). Towards an Advanced HCI through Knowledge Modelling Techniques. Em J. Cuenca, Y. Demazeau, A. G. Serrano, & J. Treuer, *Knowledge Engineering and Agent Technology*. IOS Press.
- Instituto Nacional de Estatística. (2017). Obtido de [https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine\\_publicacoes&PUBLICACOESpub\\_boui=349495406&PUBLICACOESmodo=2](https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_publicacoes&PUBLICACOESpub_boui=349495406&PUBLICACOESmodo=2)
- Jianmei Lei, D. C. (2017). *A Bus Arrival Time Prediction Method Based on*.
- Marko Čelan, M. L. (2017). Bus arrival time prediction based on network model.
- Molina, M. (2005). *An Intelligent Assistant for Public Transport Management*. Obtido de [https://www.researchgate.net/publication/220777857\\_An\\_Intelligent\\_Assistant\\_for\\_Public\\_Transport\\_Management](https://www.researchgate.net/publication/220777857_An_Intelligent_Assistant_for_Public_Transport_Management)
- PERCENTAGE OF TOTAL POPULATION LIVING IN COASTAL AREAS. (2016). Obtido de [https://sedac.ciesin.columbia.edu/es/papers/Coastal\\_Zone\\_Pop\\_Method.pdf](https://sedac.ciesin.columbia.edu/es/papers/Coastal_Zone_Pop_Method.pdf)
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2015). *Context Aware Computing for The Internet of Things: A Survey*. Obtido de IEEE Communications Surveys Tutorials, : <https://ieeexplore.ieee.org/document/6512846>

- Roads & Transport Authority. (2019). *Press Releases*. Obtido de Roads & Transport Authority: <https://www.rta.ae/wps/portal/rta/ae/home/news-and-media/all-news/NewsDetails/accomplishing-75-smart-city-and-artificial-intelligence-projects>
- Silva, F. N. (2003). *Será a mobilidade urbana ingovernável?* Lisboa: Ambiente 21, nº 11, ano II.
- Sussman, J. (2000). *Introduction to transportation Systems*. London: Artech House.
- Transport for London. (2017). *Transport for London*.
- Transport for NSW. (2018). Future Transport Strategy 2056.
- TrapezeTM. (s.d.). ARTIFICIAL INTELLIGENCE in Public Transport. Obtido de <http://trapezgroup.ae/resources/ebook-artificial-intelligence-in-public-transport>