

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

InfoPub na Web

André da Costa Meneses

VERSÃO DEFINITIVA

Relatório de Projecto
Mestrado Integrado em Engenharia Informática e Computação
Orientador: Eng. Gil Manuel Magalhães de Andrade Gonçalves

Junho de 2009

InfoPub na Web

André da Costa Meneses

Relatório de Projecto
Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Prof. João Francisco de Sousa Cardoso

Arguente: Prof. Luís Manuel Borges Gouveia

Vogal: Eng. Gil Manuel Magalhães de Andrade Gonçalves

29 de Junho de 2009

Resumo

Este relatório pretende descrever todo o trabalho realizado no âmbito do Projecto InfoPub na Web, inserido no Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

Hoje em dia existe uma necessidade cada vez maior de se fazer disponibilizar informação de uma forma rápida e eficaz. Para isso, é requisito primeiro que essa informação seja gerida e também gerada de uma forma igualmente rápida. É nesta premissa que residem as ideias basilares para o desenvolvimento de todo o trabalho que este projecto envolve.

Este projecto pretende que se torne possível ser efectuada uma gestão da informação acerca dos transportes públicos e que, com esta mesma informação, possam ser gerados de forma automática, documentos informativos (em formato SVG) para posterior afixação nas respectivas paragens.

Estes documentos terão informação relativa às Linhas que servem uma determinada paragem. Essa informação estará contida em forma de espinha que contém os vários pontos por onde passam os veículos nessa Linha e as Interfaces que servem esses pontos. Essas espinhas também terão informação relativa ao zonamento da rede e ao tempo médio para percorrer determinados troços do percurso dessa linha.

A ideia de geradores de informação automáticos não é pioneira dentro da instituição onde este projecto foi desenvolvido mas pretende-se tirar o maior partido de uma disponibilização online desta ferramenta. Para os utilizadores do sistema – entidades que gerem a informação de transportes ao público numa determinada área geográfica – esta aplicação pode trazer vantagens como a facilidade de instalação e de acesso ao serviço, assim a diminuição do custo da manutenção do hardware e software onde o sistema está alojado. Para quem disponibiliza o serviço também trará vantagens tendo em consideração a maior facilidade de actualização da ferramenta, de assistência ao utilizador e de correcção de anomalias que se tornam mais fáceis pelo facto de toda a informação estar concentrada num servidor ao seu cargo.

Neste relatório pretende-se mostrar todos os passos que foram feitos no decorrer deste projecto: como foram tomadas as decisões importantes, a razão dessas decisões, o modo como as dificuldades foram ultrapassadas e como decorreu todo o desenvolvimento da aplicação em questão.

Abstract

This report is intended to describe all the work developed during the Project Web InfoPub, as a part of the Integrated Masters in Information Technology and Computer Engineering in the Faculty of Engineering of the University of Oporto.

Nowadays there is always a bigger need to make information available in a fast and effective way. To achieve this goal it is required that that information is managed and generated in an equally fast way. All the work this project requires is based on these basic ideas.

The purpose of the Web InfoPub is that it becomes possible to manage the information related to public transportation and that, with this same information, informative documents (in SVG format), so that they can be posted in their respective stops.

These documents will have information relative to the Lines that serve a certain stop. This information will be held in a form of a spine that contains the several points where the vehicles that do that Line will stop and the Interfaces that serve those points. Those spines will also have information related to the network zoning and average time to go across certain tracks of that Line's path.

The idea of automatic information generators is not a pioneer one inside the institution where this project was developed, but it is intended that the concept of this kind of tool being available online is an upgrade. The advantage upcoming from this is valid for the two intervening parts of this process. On one hand users – entities who manage the public transportation information in a specific geographic area – will have the advantage of low installation requirements and service access as well as the withdrawn of the costs regarding software and hardware maintenance where this tool is hosted. On the other hand the entity that provides the service will have the advantage of being easier to update the tool, as well as to make user assistance and problem solving, which are made easier regarding that all the information is hosted on a server managed by this same entity/institution.

The main goal of this report is to show all the steps which were taken during the development of this project: how important decisions were made, the reason these decisions were taken, how all the difficulties were overcome and how all the development of this tool was made.

Agradecimentos

Gostaria de agradecer inicialmente ao Eng. Gil Gonçalves por toda orientação dada durante o projecto, assim como o interesse e apoio demonstrados na realização do mesmo.

Em segundo lugar gostava de agradecer ao Dr. João Marques e ao Eng. Luís Filipe Ferreira pelo acompanhamento e apoio dados na OPT no decorrer deste projecto e que foram determinantes para o bom decorrer do mesmo. Da mesma forma não posso deixar de esquecer todos os outros colegas de trabalho nesta empresa que sempre me deram apoio não só a nível pessoal como também a nível técnico sempre que foi necessário.

Não posso deixar de também de agradecer aos meus amigos que estiveram todas as horas do dia presentes comigo, mesmo que por via electrónica, e com quem sempre pude trocar os meus pensamentos, desabaços e opiniões. Que também me apoiaram nas situações mais difíceis e com quem passei bons momentos fora do trabalho. Obrigado {6+1} = 1.

Em último lugar, mas nunca menos importante que quaisquer outras pessoas, gostaria de agradecer à minha família e à minha namorada que sempre me deram apoio, carinho, compreensão e amor durante todo o decorrer deste projecto. Um grande obrigado.

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	A Instituição	2
1.3	Projecto	2
1.4	Motivação e Objectivos	3
1.5	Estrutura do Relatório.....	4
2	Revisão Tecnológica.....	7
2.1	Formatos de Dados para Importação no Sistema	8
2.1.1	Google Transit Feed Specification.....	8
2.1.2	TransXChange.....	9
2.2	Rich Internet Applications.....	12
2.3	Tecnologias Web	13
2.3.1	Tecnologias Adobe.....	13
2.3.2	Microsoft Silverlight	14
2.4	Tecnologia de Sistema de Gestão de Base de Dados	15
2.4.1	Oracle 11 vs. SQL Server 2008.....	15
2.5	Resumo e Conclusões	16
3	Descrição do Projecto	17
3.1	O que é o projecto?.....	17
3.2	Informação ao Público nas Paragens	18
3.3	Geradores de Informação ao Público.....	19
3.4	O GIST no InfoPub Desktop	21
3.5	Utilização da Aplicação.....	22
3.6	Utilizadores do Sistema	23
3.7	Requisitos do Sistema.....	24
3.7.1	Requisitos Funcionais	24
3.7.2	Requisitos Não Funcionais.....	26
3.8	Arquitectura Lógica.....	27
3.9	Arquitectura Física	29
3.10	Arquitectura da Base de Dados	30
3.11	Escolhas Tecnológicas.....	33

3.12	Metodologias e Ferramentas.....	33
3.13	Resumos e Conclusões	34
4	Implementação	37
4.1	Importação de Dados	37
4.1.1	Formatos de entrada na importação	37
4.1.2	Leitura dos ficheiros.....	38
4.1.3	Linhas	39
4.1.4	Paragens e Pontos de Mapa.....	40
4.1.5	Zonas	41
4.1.6	Horários.....	42
4.1.7	Percursos, Paragens por Percurso e Tempos de Passagem	42
4.1.8	Correspondências na Base de Dados.....	43
4.2	Funcionalidades na DataGrid Silverlight 2.0	46
4.3	Formulários em Silverlight.....	47
4.3.1	Utilização do sistema	47
4.3.2	Login	47
4.3.3	Users.....	48
4.3.4	Images	49
4.3.5	Identity	49
4.3.6	Interfaces	50
4.3.7	Lines, Stops e MapPoints	51
4.3.8	InterfacesStop.....	51
4.3.9	Colors	52
4.3.10	Data Import	53
4.3.11	Document Generation	53
4.4	Gerador de Informação	54
4.4.1	Documento	54
4.4.2	Espinha Horizontal.....	55
4.4.3	Espinha Vertical	56
4.5	Testes Efectuados	57
4.6	Resumo e Conclusões	58
5	Conclusões	59
5.1	Satisfação dos Objectivos	59
5.2	Trabalho Futuro	60
6	Bibliografia	62
Anexo A – Tabelas da Base de Dados		64
Anexo B – Exemplos de Documentos		75

Lista de Figuras

Figura 2.1: Esquema TransXChange	10
Figura 2.2: Modelo de Dados TransXChange / GTFS	11
Figura 3.1: Exemplo do BusMap na Carris, em Lisboa	19
Figura 3.2: Exemplo do BusSched na STCP, no Porto	20
Figura 3.3: Gestor de Dados do InfoPub Desktop	21
Figura 3.4: Planeamento de viagens no GIST	22
Figura 3.5: Diagrama dos utilizadores do sistema	23
Figura 3.6: Esquema da Arquitectura Lógica Horizontal	27
Figura 3.7: Esquema da Arquitectura Lógica Vertical.....	28
Figura 3.8: Esquema da Arquitectura Física	30
Figura 3.9: Esquema da Base de Dados InfoPub na Web.....	32
Figura 4.1: Esquema de Importação para a Base de Dados	38
Figura 4.2: Leitura dos Ficheiros para as respectivas Classes	39
Figura 4.3: Importação das Linhas	39
Figura 4.4: Importação dos MapPoints e Stops.....	40
Figura 4.5: Importação das Zonas	41
Figura 4.6: Importação dos Schedules	42
Figura 4.7: Importação dos Paths, PathStops e Passings	43
Figura 4.8: Interface Login.....	48
Figura 4.9: Interface Users	48
Figura 4.10: Interface Images.....	49
Figura 4.11: Interface Identiy.....	50
Figura 4.12: Interface Interfaces	50
Figura 4.13: Interface Stops	51
Figura 4.14: Interface InterfacesStop	52
Figura 4.15: Interface Colors	52
Figura 4.16: Interface Import Data.....	53
Figura 4.17: Interface Document Generation (Parametrization).....	54
Figura 4.18: Exemplo de Cabeçalho	55
Figura 4.19: Exemplo do Rodapé.....	55
Figura 4.20: Exemplo de Espinha Horizontal	56

Figura 4.21: Exemplo de Espinha Vertical 57

Lista de Tabelas

Tabela 3.1: Prioridade dos Requisitos funcionais	24
Tabela 3.2: Prioridade dos Requisitos não Funcionais	26
Tabela 3.3: Lista das Tabelas da Base de Dados.....	31
Tabela 4.1: Correspondência stops.txt para a Base de Dados (MapPoints).....	44
Tabela 4.2: Correspondência stops.txt para a Base de Dados (Stops)	44
Tabela 4.3: Correspondência routes.txt para a Base de Dados	44
Tabela 4.4: Correspondência calendar.txt para a Base de Dados.....	45
Tabela 4.5: Correspondência trips.txt para a Base de Dados	45
Tabela 4.6: Correspondência stop_times.txt para a Base de Dados.....	46

Abreviaturas e Símbolos

CSV	Comma Separated Values
GIST	Gestão Integrada de Sistemas de Transportes
GTFS	Google Transit Feed Specification
HTML	HyperText Markup Language
IIS	Internet Information Services
OPT	Optimização e Planeamento de Transportes
PDF	Portable Document Format
SaaS	Software as a Service
SGBD	Sistema de Gestão de Base de Dados
STCP	Sociedade de Transportes Colectivos do Porto
SVG	Standard Vector Graphics
WCF	Windows Communication Foundation
WMA	Windows Media Audio
WMV	Windows Media Video
WWW	<i>World Wide Web</i>
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

1 Introdução

Este documento pretende demonstrar todo o processo de pesquisa, análise e desenvolvimento do projecto InfoPub na Web.

O InfoPub na Web é uma ferramenta que pretende, como último objectivo, gerar documentos de informação ao público com informação relativa aos transportes públicos, aos seus percursos e às suas horas de passagem. Estes documentos são, regra geral, relativos a uma paragem específica, mostrando a parte do percurso mais relevante para essa paragem assim como as suas horas de passagem. Para que isto seja possível a ferramenta dispõe também de um sistema que permite que se possa fazer a importação dos dados usados para gerar esses documentos e interfaces que permitam fazer pequenas alterações aos dados importados assim como parametrizar a informação a ser mostrada nos documentos finais e a maneira como esta será mostrada.

Este capítulo pretende fazer uma introdução a todo o contexto onde este projecto se enquadra. Serão focados aspectos relacionados com a necessidade de este projecto ser desenvolvido, o âmbito onde ele se insere, o ambiente onde foi desenvolvido, a motivação para o realizar e os objectivos que se pretendem atingir, assim como será feito um pequeno resumo da estrutura do resto deste documento.

1.1 Enquadramento

O Projecto InfoPub na Web foi realizado como projecto de final de curso, no âmbito do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto. Este projecto foi proposto pela OPT – Optimização e Planeamento de Transportes S.A., em cujas instalações decorreu o seu desenvolvimento.

Todo o processo de desenvolvimento descrito neste relatório foi feito durante o segundo semestre do ano lectivo de 2008/2009, com início no mês de Fevereiro de 2009 e final no mês de Julho de 2009.

O projecto foi desenvolvido por André da Costa Meneses, aluno finalista do Mestrado Integrado em Engenharia Informática e Computação, sob a orientação do Eng. Gil Manuel Magalhães de Andrade Gonçalves.

1.2 A Instituição

A OPT (Optimização e Planeamento de Transportes, S.A.) foi a empresa onde este projecto foi desenvolvido, tendo esta disponibilizado todo o contexto para a realização do mesmo.

Esta empresa foi constituída em 1992, sendo presentemente uma Sociedade Comercial Anónima, tendo como principais accionistas algumas das maiores empresas de transportes públicos em Portugal, sendo também em alguns casos, os seus principais clientes [OPT05].

A principal missão da OPT é oferecer sistemas inovadores e eficientes no planeamento e gestão dos transportes públicos, assim como sistemas de informação ao público. Assim sendo, esta empresa caracteriza-se por dois núcleos que trabalham em paralelo: um que trabalha na optimização da gestão dos sistemas de planeamento e gestão dos transportes públicos, e aquele que trabalha no desenvolvimento do software que permite uma tarefa facilitada nesse sentido.

A OPT prima por tentar sempre imprimir agilidade aos processos de desenvolvimento de software, mesmo em projectos de maiores dimensões, dando prioridade aos objectivos mais importantes a curto prazo, estabelecendo prazos para que esses objectivos sejam atingidos.

O principal produto da OPT é o GIST (Gestão Integrada de Sistemas de Transportes). O GIST é um pacote de software com diversos módulos que permitem gerir a informação base relativa à rede de transportes, às linhas e às viagens a realizar. Permitem ainda gerar, de uma forma optimizada, os horários das viaturas e dos motoristas, incluindo o escalamento diário destes últimos.

Outro produto que tem vindo a ganhar particular destaque dentro da empresa é o InfoPub (Informação ao Público). Este produto está intimamente ligado ao projecto descrito neste relatório, visto ser uma versão Desktop e neste momento mais completa, da ferramenta que foi desenvolvida. O InfoPub divide-se em dois principais módulos: o BusMap e o BusSched.

O BusMap gera documentos com um diagrama principal representando a interacção dos percursos que passam por determinada paragem, mostrando os pontos notáveis que atravessam a partir desse ponto. O BusSched mostra uma espinha revelando as paragens de uma linha específica e com os horários de passagem para essa linha, nessa paragem.

1.3 Projecto

O Projecto InfoPub na Web surge da necessidade de expansão e globalização da OPT e conseqüente descentralização da localização geográfica dos clientes deste produto. Assim sendo, foi tomada a decisão de desenvolver o protótipo de uma aplicação Web (RIA – Rich Internet Application), que pelas suas características seria uma mais-valia quer para a OPT, quer para os potenciais clientes desta ferramenta.

A ideia principal é partir do conceito da aplicação InfoPub Desktop existente e a partir daí ir construindo um protótipo que no fim permita demonstrar as potencialidades desta aplicação aos clientes.

Existem portanto, uma série de requisitos que à partida são fundamentais para o sucesso desta aplicação e para que se possam atingir as metas propostas.

É importante que numa estratégia de globalização esta aplicação esteja a todos os níveis acessível ao maior número de clientes possível com o menor custo para estes. Para que isto seja possível é preciso escolher uma tecnologia que esteja ao alcance de todos, é necessário que o formato para importação dos dados necessários à geração de informação seja o mais standard possível, é necessário que seja possível às empresas fazerem uma gestão dos utilizadores que têm acesso ao produto e também poderem ter uma certa liberdade para parametrizar os documentos que serão gerados por esta ferramenta, tendo algumas opções disponíveis para o efeito.

Também é preciso que do lado da OPT se possa fazer uma gestão dos clientes, dos serviços adquiridos por estes e que haja um completo controlo do sistema de modo a que seja possível garantir o seu contínuo bom desempenho.

1.4 Motivação e Objectivos

Tendo em conta a crescente utilização das tecnologias Web e neste caso específico das Rich Internet Applications é um desafio poder tentar transpor uma aplicação que nos dias de hoje é executada em ambiente Desktop e fazer uma versão Web para que os utilizadores a possam utilizar sem necessidade de requisitos de sistema exigentes e instalações complexas e onerosas.

Assim será possível fazer uma análise das potencialidades de se criar uma aplicação nestes moldes e tendo como ponto de partida uma aplicação Desktop com duas principais funções: fazer uma gestão de dados e gerar informação recorrendo a esses mesmos dados.

Os Software as a Service (SaaS) caracterizam-se por ser software que não é desenvolvido à medida, mas em vez disso é produzido apenas localizado no provedor de serviços, a maioria das vezes apenas acessível através da Internet, onde são vendidas licenças para se poder utilizar os serviços que esse software oferece. Isto é cada vez mais uma realidade no mundo dos sistemas de informação pois cada vez mais os SaaS se afirmam como uma alternativa e as perspectivas de crescimento são cada vez maiores [Mah09].

Assim o InfoPub na Web passa também por ser um SaaS tendo como objectivo tirar o maior partido possível das vantagens deste tipo de software que tanto se aplicam a quem desenvolve as aplicações como também aos possíveis clientes. Facilidade de manutenção e de upgrade da aplicação para os desenvolvedores do serviço por esta ser executada num único local. Para os clientes a vantagem de o investimento poder ser menor e de haver menos risco ao investir num serviço deste género.

A informação é algo que as pessoas cada vez mais procuram, pois ter um conjunto de dados não é o mesmo que ter esses dados expostos de uma maneira que seja acessível para as pessoas compreenderem. Como tal, é necessário pegar nesses dados e torná-los em algo compreensível pelo cidadão comum, e neste caso específico o utilizador de transportes públicos, para que na paragem possa ter a informação que precisa para planear da melhor maneira as suas viagens.

Produzir uma aplicação deste género pode trazer vantagens a vários níveis. Desde logo o facto de ser algo mais acessível para as empresas que produzirão este tipo de

informação facilitará a sua adesão à aplicação. Esta adesão por parte das empresas de transportes públicos pode ter um segundo impacto que é o aumento da utilização dos mesmos. Este facto pode ser atingido visto que hoje em dia muitos cidadãos não têm acesso facilitado à informação porque é difícil a sua consulta ou porque simplesmente ela não está disponível. Existindo esta informação pode levar a uma maior motivação para as pessoas usarem os transportes públicos ao invés de se renderem à comodidade de utilizar o seu transporte particular.

Com tudo isto o objectivo deste projecto é atingir um protótipo funcional, que possa evidenciar as potencialidades de uma Rich Internet Application. Este protótipo terá de evidenciar algumas capacidades entre as quais se destacam a capacidade de importar a informação para o sistema, fazer a gestão dessa mesma informação, e numa última fase gerar os documentos informativos, de uma forma minimamente parametrizável, para que estes sejam exportados num formato SVG, para consulta ou impressão.

1.5 Estrutura do Relatório

Este documento pretende dar a conhecer, de uma forma estruturada, todas as etapas da realização deste projecto e assim mostrar de uma forma mais detalhada a ferramenta que foi desenvolvida e os aspectos que levaram a que o estado da aplicação fosse o presente.

O presente capítulo – a Introdução – pretende dar a conhecer de uma forma geral aquilo que este projecto é, o que lhe deu origem assim como o que é pretendido atingir no final do mesmo.

Seguidamente será apresentada toda a investigação que foi efectuada em torno deste projecto quer a nível teórico, quer a nível tecnológico, para que se pudessem escolher os melhores caminhos na sua realização. Esta investigação passou a nível mais teórico por perceber o trabalho que já é realizado na área de informação ao público e formatos standard para os dados de entrada. A nível tecnológico foi preciso investigar as ferramentas e tecnologias de desenvolvimento que melhor se adequavam a este projecto: foram analisadas diversas tecnologias Web e também os possíveis Sistemas de Gestão de Base de Dados.

No terceiro capítulo será explicado com mais detalhe o problema a resolver. De uma forma geral vão-se mostrar as decisões que foram tomadas tendo em conta aquilo que foi investigado e exposto no capítulo anterior e será apresentado o planeamento de implementação da aplicação.

O modo de implementação da aplicação será exposto no quarto capítulo. A implementação dividiu-se sobretudo na importação dos dados, seguida do desenvolvimento de interfaces que permitirão haver algum controlo dos dados importados assim como a parametrização de alguns aspectos dos documentos a produzir e por fim o desenvolvimento dos geradores de documentos, que são no fundo o produto e objectivo final da aplicação.

O quinto capítulo reserva-se para toda uma análise deste projecto. Irão ser focados os aspectos de onde se puderam tirar as conclusões mais importantes, será feita uma análise do cumprimento dos objectivos propostos no início do projecto, assim como serão feitas algumas considerações tendo em conta o trabalho que poderá vir a ser desenvolvido na sequência deste projecto e no sentido de o tornar cada vez melhor.

De realçar que cada um dos capítulos que se seguem terão uma pequena secção no final para pequenas considerações sobre o que foi tratado nesse mesmo capítulo.

2 Revisão Tecnológica

Neste capítulo será mostrada a investigação que foi feita em torno das áreas envolventes ao desenvolvimento deste projecto.

A aplicação desenvolvida tem também de constituir uma vantagem em relação a outras que possam existir dentro do mesmo género. Um dos aspectos a ter em conta é a “universalidade” do formato de importação de dados. Quanto mais generalizado estiver o formato que for utilizado para se poder fazer a importação de dados maior será a probabilidade de adesão por parte de possíveis clientes. Isto é algo que tem de se ter em conta visto que além de um trabalho que envolve alguma investigação é pretendido que a ferramenta (ou as suas funcionalidades) possa vir a ser comercializada futuramente.

Mostrar como a informação normalmente é disposta nas paragens também é importante para perceber o que se deve incluir nos documentos gerados e o que pode ser inovado ou melhorado em relação ao que existe. Isto faz deste ponto uma parte importante deste estudo.

Seguidamente será feito um estudo das tecnologias disponíveis para desenvolver o projecto. A nível de tecnologias Web é necessário considerar que este é o ponto de contacto com o utilizador da aplicação e isso é um factor importante. É importante ter a consciência que deve ser escolhida uma tecnologia que seja acessível a um número vasto de utilizadores sem necessidade de um esforço acrescido por parte dos mesmos. Ao mesmo tempo é preciso considerar o modo como cada tecnologia se adequa ao que se pretende fazer e as suas potencialidades tendo em conta os objectivos que são propostos atingir. Também será demonstrado o estudo feito em torno das tecnologias de Sistemas de Gestão de Base de Dados. Aqui a interacção com o utilizador não é tão importante pois a maneira como os dados ficam armazenados será invisível ao mesmo. De qualquer modo é preciso ter em conta a maneira como poderá interagir a tecnologia Web com o SGBD, sempre considerando a performance e capacidade destes últimos.

A terminar este capítulo, será feita uma pequena reflexão sobre o mesmo realçando os aspectos mais importantes que foram focados e fazendo a sua síntese.

2.1 Formatos de Dados para Importação no Sistema

Com a necessidade de haver um formato de dados para importação que fosse o mais standard possível foi feita uma pequena pesquisa do que existe nesse domínio. Dois formatos foram estudados e foi verificada a sua viabilidade para serem utilizados neste projecto.

2.1.1 Google Transit Feed Specification

Esta especificação da Google teve a sua primeira versão publicada a 25 de Setembro de 2006 [GoC09], sendo que a sua última revisão foi efectuada a 30 de Março de 2009. Esta especificação permite fazer uma importação de dados para uma ferramenta da Google, o Google Transit, que está desde finais de 2007 totalmente integrado com o Google Maps.

O Google Transit é uma ferramenta que permite programar viagens usando os transportes públicos. Os resultados desse planeamento podem então ser consultados no Google Maps. Para que isto seja possível é necessário que as empresas de transportes públicos disponibilizem os dados relativamente à sua rede, horários, tarifários, etc. ... Isto tornou-se possível através do GTFS.

A especificação GTFS consiste num conjunto de 12 ficheiros [Goo09], sendo que 6 destes são obrigatórios e os outros 6 são opcionais.

Estes ficheiros apesar de terem a extensão *.txt estão em formato CSV. O nome dos ficheiros encontra-se listado abaixo assim como uma pequena descrição de cada um deles. De salientar que os primeiros 6 ficheiros listados são os que têm de estar obrigatoriamente na especificação e os outros 6 são opcionais:

- **agency.txt** – Informação sobre a agência ou agências sobre a qual o feed tem informação.
- **stops.txt** – Informação sobre as localizações específicas das paragens dos transportes públicos.
- **routes.txt** – Este ficheiro contém informação sobre as Linhas. As Linhas são conjuntos organizados de “trips”(viagens) apresentados aos utilizadores como um único serviço.
- **trips.txt** – Este ficheiro contém informação sobre todas as trips e as linhas a que pertencem. Uma trip é uma sequência de duas ou mais paragens percorridas num momento específico.
- **stop_times.txt** – Contém todas as horas de chegada e de partida de um veículo, numa “trip”, para cada paragem.
- **calendar.txt** – Especifica os dias da semana a que corresponde determinado serviço. Também especifica a data de entrada em vigor e de fim de exercício desse serviço.
- **calendar_dates.txt** – Este ficheiro contém todas as excepções ao ficheiro calendar.txt.
- **fare_attributes.txt** – Informação relacionada com o tarifário das diferentes linhas.

- **fare_rules.txt** – Especificação das regras a aplicar aos tarifários.
- **shapes.txt** – Especifica as regras de desenho para se representar uma determinada Linha num mapa.
- **frequencies.txt** – Contém especificações das frequências de cada veículo que podem ser definidas em vez dos tempos de passagem específicos.
- **transfers.txt** – Regras para transbordo entre Linhas da Rede.

Existem cada vez mais ferramentas disponíveis onde se pode usar o GTFS. Alguns exemplos vêm mencionados na própria página de especificação do GTFS e são o TimeTablePublisher e o FeedValidator.

O TimeTablePublisher [GCo09] permite transformar os dados existentes num Feed em formato GTFS para um formato mais legível e compreensível. Por seu lado o FeedValidator verifica de uma forma rápida e simples se um determinado Feed GTFS está num formato válido.

Outras ferramentas que envolvem Feeds GTFS encontram-se em desenvolvimento. Um desses casos é a “Google Transit Data Tool for Small Transit Agencies” [TRB09] que está a ser desenvolvida pela Transportation Research Board [TrR09]. Esta ferramenta irá permitir a empresas de transportes de dimensões mais reduzidas produzir sem grande esforço os seus Feeds no formato GTFS para que a sua Rede e as suas viagens se tornem disponíveis no Google Transit e assim esteja mais facilmente ao alcance das pessoas planearem as suas viagens através dessas empresas.

O formato GTFS também já é usado para ferramentas que não apenas o Google Transit. O PedNav [RMT08], ferramenta para o Google Android, tem como input dados o formato GTFS [Eva08]. Esta ferramenta permite calcular um plano de viagem para um dia tendo como input determinados parâmetros desejados pelo utilizador.

Mesmo grandes entidades dos transportes públicos nos Estados Unidos da América, como a WMATA (Washington Metropolitan Area Transit Authority), que inicialmente se encontrava relutante em disponibilizar os seus dados para o Google Transit dizendo que “uma parceria com a Google não seria a melhor estratégia de negócio” [GGE08], acabou por, eventualmente acusando a pressão dos utilizadores com petições [GGE00], disponibilizar os seus dados para o Google Transit. Acrescendo a isto o facto de terem tornado públicas as suas feeds para qualquer pessoa no mundo poder consultar [Peg09].

Também são realizadas algumas iniciativas como foi o Google Transit Earth Day Challenge que incentivava as empresas de transportes públicos a disponibilizarem os seus dados para fazerem parte do Google Transit [Cam08]. Esta iniciativa teve algum sucesso visto que várias empresas pela América aderiram ao Google Transit.

2.1.2 TransXChange

O formato TransXChange é o formato standard para todo o Reino Unido para troca de horários de transportes públicos e dados relacionados.

Este formato é usado para registar dados das Linhas operadas pelos transportes públicos do Reino Unido junto das entidades responsáveis, assim como para trocar informação entre diferentes aplicações de software como sistemas de planeamento de viagens ou aplicações de acompanhamento de viaturas em tempo real [Hom09].

O TransXChange é patrocinado pelo Departamento de Transportes do Reino Unido e faz parte de uma família de uma série de outros *schemas* XML relacionados.

Muitos dados podem ser trocados usando este sistema. Ele serve sobretudo para conter informação acerca dos horários dos transportes públicos, incluindo as paragens, linhas, percursos, horas de passagem, mapas com os percursos e outras notas. Ele comporta uma série de tipos de percursos mais complexos como as rotas circulares ou os expressos. Pode ser incluída informação acerca dos horários em que o serviço está disponível, sendo possível definir excepções ou ocasiões especiais como feriados e pontes. Ele comporta uma série de outras informações técnicas, entre elas os tarifários, que permitem que apenas um ficheiro XML possa definir toda a rede.

Na Figura 2.1 podemos verificar toda a interacção que o formato TransXChange permite com ferramentas externas.

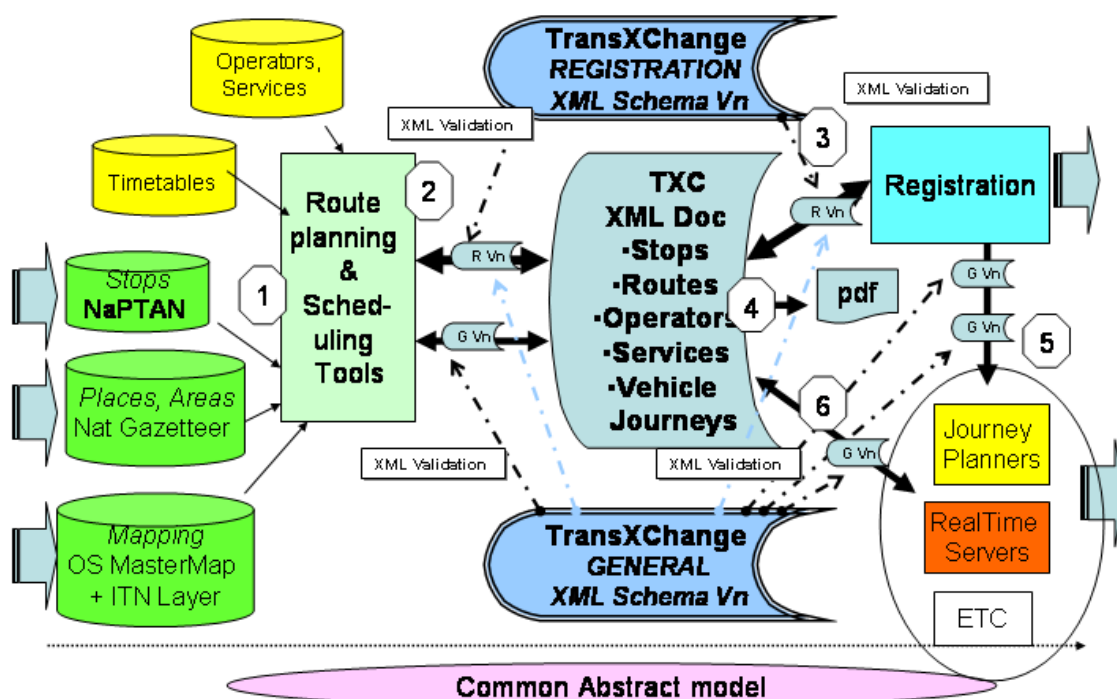


Figura 2.1: Esquema TransXChange

Existem ferramentas que são disponibilizadas pela organização responsável pelo formato TransXChange e que permitem a exportação de informação, sobretudo relacionada com os horários, para um formato PDF com o objectivo de uma consulta mais fácil da informação contida num ficheiro TransXChange.

A empresa OmniBus [Mic08] é o exemplo de uma empresa que nos seus produtos integra o formato TransXChange tanto como formato de importação como formato de exportação de dados, sendo assim possível a interacção do software produzido por esta empresa com outros existentes ou mesmo com as autoridades que requerem a informação das empresas neste formato. Isto espelha a vantagem de haver um formato

uniformizado de troca de dados neste caso especificamente para os Transportes Públicos.

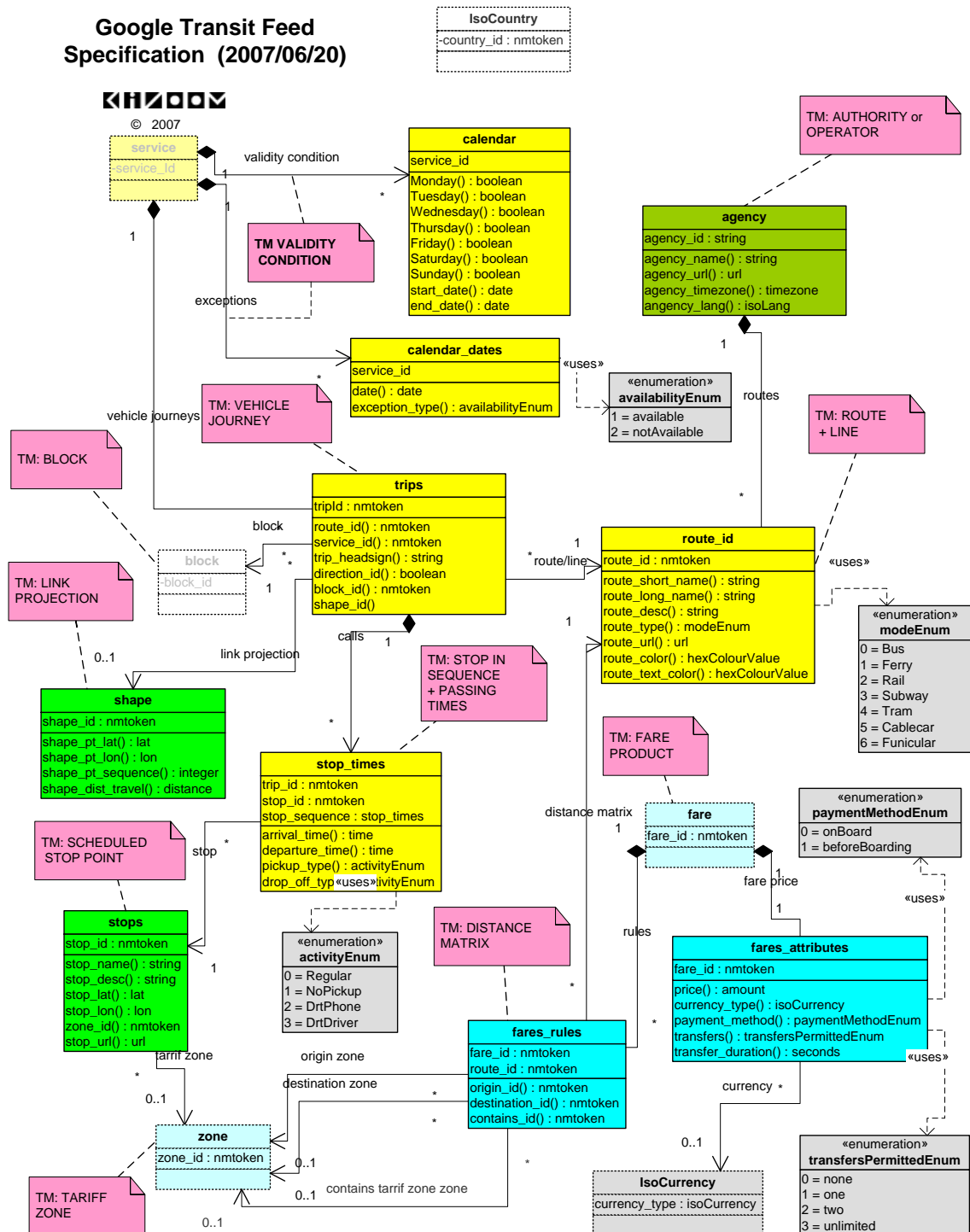


Figura 2.2: Modelo de Dados TransXChange / GTFS

O Department of Transport, entidade responsável pelo formato TransXChange, também emitiu um documento analisando o GTFS e comparando as potencialidades deste com o TransXChange. Nesse documento são estabelecidas associações entre os

dois modelos, mostrando a sua potencial compatibilidade ou possibilidade de transformação de formato.

Nas conclusões desse documento vem a informação que o formato GTFS cobre a maioria dos casos de percursos simples e viagens comuns. Contudo, sente-se que o GTFS fica um pouco aquém quando se tratam de percursos mais complexos, percursos estes que o TransXChange é capaz de tratar.

Na Figura 2.2 está patente a relação, atingida por engenharia reversa, que foi possível atingir e que foi demonstrada nesse mesmo documento entre o GTFS e o TransXChange.

De notar que o Verde corresponde a entidades do formato NaPTAN/NPTG (formato mais específico das paragens que pode também ser usado dentro de um documento TransXChange), o amarelo refere-se a entidades TransXChange, o cinzento a tipos de dados base e o azul a tarifas.

2.2 Rich Internet Applications

As Rich Internet Applications são aplicações Web que têm algumas características de aplicações Desktop. Isto é exactamente o conceito que se pretende atingir uma vez que este projecto consiste numa transformação da aplicação Desktop InfoPub para funcionamento na Web.

As aplicações Web tradicionais estão normalmente muito dependentes do servidor. Nessas aplicações tradicionais é apresentada ao utilizador uma página HTML estática e qualquer interacção necessária obrigará a uma comunicação/sincronização com o servidor. As RIA vêm prevenir essa necessidade constante de sincronização, uma vez que as suas capacidades de execução do lado do cliente são maiores e portanto isto leva a uma diminuição da comunicação com o servidor.

Estas aplicações RIA estão normalmente desenvolvidas em cima de uma Framework como o Microsoft Silverlight ou tecnologias da Adobe como o Flash/Flex/Air.

Com este tipo de tecnologias, as RIA vêm com uma série de vantagens em relação às páginas convencionais como a capacidade de realizarem uma série de operações sem precisarem da intervenção do servidor. Este facto faz com que a interacção com o utilizador ganhe uma velocidade superior e transmite a este a sensação de estar a usar uma aplicação Desktop na maioria das situações. A comunicação assíncrona também é uma das grandes vantagens das RIA, uma vez que não é estritamente necessário que o utilizador esteja à espera que uma operação acabe para poder realizar a operação seguinte. Noutra perspectiva, em alguns casos, a aplicação pode antecipar ou tentar antecipar as acções do utilizador fazendo as ligações ao servidor mesmo antes de elas serem necessárias.

Ainda assim nem tudo são vantagens nas aplicações RIA e existem algumas coisas que ainda podem constituir entraves à sua execução. Um exemplo disso mesmo é a dependência que este tipo de aplicações ainda tem da habilitação ou não do Javascript por parte do cliente, que pode impedir a execução destas aplicações, assim como uma má configuração de acesso aos recursos. Outro dos problemas que os clientes podem ter deve-se ao tempo que uma RIA pode levar a carregar. O facto de este tipo de aplicações terem uma grande autonomia no cliente também leva a que estas aplicações, por vezes,

tenham um tamanho considerável e assim o seu carregamento inicial pode ser mais demorado. Este será talvez um dos maiores problemas que as RIA enfrentam hoje em dia e por isso existem medidas que estão a ser tomadas no sentido de melhorar esta performance, como por exemplo carregar a aplicação à medida que os seus componentes forem sendo necessários. Mesmo considerando este facto, no fim da utilização da aplicação o tempo dispendido à espera de comunicações com o servidor deverá ser sempre menor do que nas páginas Web comuns. Há igualmente quem considere que uma das grandes desvantagens das RIA é a perda de visibilidade nos motores de busca, algo que se encontra também a ser melhorado.

A dependência de uma conexão com a Internet é por vezes referida como uma desvantagem, mas apenas se compararmos as RIA com aplicações Desktop. No caso de esta comparação ser feita com outras aplicações Web isto não se verifica e portanto não poderá ser considerada uma desvantagem.

2.3 Tecnologias Web

Para desenvolver esta aplicação é preciso escolher uma tecnologia Web na qual vai assentar toda a interface e que terá impacto na lógica de negócio. A Adobe tem uma série de tecnologias que tornam possível produzir aplicações RIA. Do outro lado temos o emergente Silverlight da Microsoft que começa a tomar protagonismo e a mostrar-se como uma séria ameaça ao domínio do Flash na Internet durante os últimos anos.

Com tudo isto, estas serão as tecnologias que serão estudadas para se poder futuramente tomar uma decisão em relação à tecnologia Web a utilizar no desenvolvimento deste projecto.

2.3.1 Tecnologias Adobe

A Adobe possui uma série de tecnologias que permitem construir Rich Internet Applications. Estas tecnologias começam numa camada inferior onde aparecem os servidores como o Adobe Flash Media Server ou o ColdFusion. Acima disto existem as Frameworks como o Flex. No último nível temos os clientes capazes de reproduzir os conteúdos desenvolvidos: o Flash e o Adobe Air.

Com tudo isto foi feito um estudo direccionado para o que era possível reproduzir em Flash para se poder perceber até que ponto era vantajoso desenvolver nessa tecnologia.

O Flash é uma tecnologia utilizada mundialmente. Cerca de 98% dos computadores com acesso à Internet têm o Flash instalado. Isto é facilitado pelo facto de o Flash funcionar na esmagadora maioria dos browsers e Sistemas Operativos. Por isto, ao longo dos anos o Flash veio a ganhar preponderância na Internet em mini-aplicações baseadas em animações. O facto de ser uma ferramenta “universal” é realmente uma das suas grandes vantagens. Por outro lado não é o único factor que deve ser analisado e há aspectos técnicos que devem ser tidos em conta para o desenvolvimento do InfoPub na Web os quais também serão revistos.

As animações em Flash são baseadas num sistema de *frames*. Este facto leva a um controlo pouco preciso sobre a duração das animações. Dependendo da performance do computador as mesmas animações podem levar quantidades de tempo substancialmente diferentes a serem reproduzidas.

O *encoding* de vídeo no Flash tende a ser um pouco complexo. Nem todos os formatos são aceites e são necessárias ferramentas para que este processo seja possível.

Com o Flash há a desvantagem de se poderem utilizar bibliotecas C# com partes do projecto que já existem. Usar Flash levaria à necessidade de se reescrever todo esse código.

Em último lugar o Flash precisa de um intermediário para acesso à Base de Dados. Isto pode ser feito com recurso a várias tecnologias podendo estar ser escolhida também em função do SGBD usado.

2.3.2 Microsoft Silverlight

O Microsoft Silverlight foi lançado precisamente pela Microsoft percebendo a necessidade que o mercado cada vez mais tinha de Rich Internet Applications. De acordo com estudos efectuados, os resultados RIA vão de encontro ou excedem as expectativas das empresas em relação ao sucesso das soluções deste tipo. A necessidade de atrair novos clientes que estejam abertos a novos tipos de tecnologias também é um ponto de necessidade para se criar uma plataforma de desenvolvimento de Rich Internet Applications. Para além destes novos clientes é preciso também manter os clientes existentes satisfeitos, visto que estes se tornam cada vez mais exigentes e não se contentam com qualquer solução que lhes seja proposta como em tempos aconteceu. Assim, a Microsoft desenvolveu o Silverlight, que permite desenvolver aplicações à medida destas novas necessidades [Mic09].

O Silverlight permite distribuir conteúdos em vídeo ou aplicações desenvolvidas na plataforma .NET nos maiores browsers a funcionar em Windows, Mac e Linux. Usando as ferramentas da Microsoft para desenvolvimento na plataforma .NET é possível cada vez mais desenvolver trabalho de uma forma colaborativa e orientada.

O Silverlight na altura do início deste projecto encontrava-se na sua versão 2.0 e as suas características na altura serão aquelas que vão ser aqui descritas. Entretanto foi lançada a versão 3.0 beta que já inclui uma série de novas e inovadoras funcionalidades.

Em termos mais técnicos o Silverlight também permite animações como o Flash, embora o Silverlight tenha animações orientadas ao tempo decorrido contrariamente ao sistema de *frames* do Flash.

Para a interface o Silverlight recorre a uma camada separada da lógica de negócio. Toda a interface e formas contidas nela são definidas em XAML. O XAML é uma linguagem XML que permite desenhar de uma forma fácil e intuitiva toda a interface para ferramentas .NET (não se aplica apenas ao Silverlight).

O XAML é uma linguagem declarativa, que tem um conjunto de objectos definidos e é possível desenvolver em cima desses objectos. De resto o XAML é em tudo idêntico a qualquer outra linguagem XML [FWT00].

Em termos de reprodução de vídeo e áudio, o Silverlight suporta nativamente WMV e WMA, não suportando os formatos AVI ou MOV. Contudo é fácil converter para os formatos WMV e WMA e por isso esta limitação não deverá colocar muitos entraves a quem quiser usar Silverlight para disponibilizar os seus conteúdos de vídeo.

Ligar o Silverlight a um SGBD parece ser relativamente simples se este SGBD for o SQL Server usando as tecnologias Linq to SQL que faz o mapeamento da Base de Dados e o WCF (Windows Communication Foundation) que é responsável por fazer a

ligação Cliente-Servidor. Todas estas tecnologias têm bastante suporte por parte da Microsoft e o Silverlight está bem preparado para trabalhar com elas. A ligação a um SGBD Oracle mostra-se um pouco mais complexa sendo preciso usar ferramentas de terceiros obrigando a um maior tempo de aprendizagem.

O Silverlight tem a vantagem acrescida de se poder usar muito do código que se encontra desenvolvido para algumas partes do projecto visto o desenvolvimento ter a possibilidade de ser feito em .NET, na linguagem C#, da mesma maneira que está a feita a versão Desktop do InfoPub.

2.4 Tecnologia de Sistema de Gestão de Base de Dados

Observando as diferentes tecnologias existem duas potenciais candidatas a servirem este projecto: Oracle e SQL Server. Por um lado a Oracle é a Base de Dados utilizada pela OPT no InfoPub versão Desktop e outros projectos existentes. O SQL Server é uma Base de Dados de fácil integração caso a ferramenta Web escolhida seja o Silverlight. Sendo assim, uma pequena análise tecnológica de comparação entre as duas foi efectuada.

2.4.1 Oracle 11 vs. SQL Server 2008

Neste tópico é mais propícia uma comparação entre as duas tecnologias, centrando-nos nas versões mais recentes de cada uma delas. Podem ser comparadas as suas características e as suas performances de modo a se poder tomar uma decisão acerca da que será a mais adequada para o projecto.

De acordo com estudos recentes o SQL Server mostra ser mais fiável em termos de segurança que o Oracle. Isto pode ser observado tendo em conta que o NIST (National Vulnerability Database) encontrou mais de 330 vulnerabilidades em torno da segurança do Oracle nos últimos quatro anos enquanto o SQL Server, ainda o 2005, não mostrou nenhuma vulnerabilidade.

Outra grande vantagem do SQL Server é a sua grande integração com o Visual Studio e que conseqüentemente torna mais fácil a sua integração com projectos que possam ser desenvolvidos nesta ferramenta de desenvolvimento. As ferramentas usadas para se trabalhar com as Bases de Dados Oracle são, na maioria das vezes proprietárias. Isto obriga a um custo monetário muito maior, acrescentando o facto de que se tem de trabalhar sobre muito mais interfaces e a curva de aprendizagem é muito maior. Logo a produtividade em Oracle acaba por ser reduzida.

O SQL Server tem como desvantagem o facto de apenas estar disponível para Windows, enquanto o Oracle está disponível para Windows, Linux e Mac. Também está patente que o Oracle é mais configurável, sendo considerada uma ferramenta de mais baixo nível. O SQL Server por seu lado tem a tendência a optar pelas configurações que lhe parecem mais indicadas ou então apenas tem algumas configurações pré-definidas.

As grandes comparações que existem entre estes dois sistemas são feitas sobretudo pela Microsoft e pela Oracle. Este facto é ingrato na medida em que cada uma das empresas se preocupa apenas em mostrar os pontos vantajosos da sua aplicação em relação à aplicação concorrente. No entanto pode-se notar uma tendência da Microsoft para fazer comparações mais gerais contra as comparações mais específicas da Oracle. Isto denota uma aposta da Oracle em ter soluções para problemas mais específicos ao

contrário da Microsoft que prepara o SQL Server para um conjunto mais generalizado de cenários de utilização.

Um dos grandes factores e muitas vezes o mais decisivo para muitas companhias é o elevadíssimo preço do SGBD da Oracle. Quando comparado com o SQL Server há uma diferença muito grande a nível de preço. Não é de admirar que a muitos níveis o SQL Server seja referido como o número 1 em termos de performance/custo. Talvez o Oracle ultrapasse o SQL Server em alguns níveis de performance mas o importante é perceber a necessidade desse aumento de performance em torno de tudo o resto que o SQL Server oferece como vantagem em relação ao Oracle.

2.5 Resumo e Conclusões

Existem vários níveis decisão neste projecto. Algumas dessas decisões passam pelo que será esta aplicação e o que permitirá fazer. Inicialmente foram observados os formatos possíveis para importação de dados. Os mais conhecidos e utilizados hoje em dia têm as suas semelhanças e foi possível verificar com a pesquisa efectuada que até têm pontos de contacto e que já existem ferramentas para tentar fazer a ligação entre eles. Cada um dos formatos analisados tem uma aplicação específica mas ambos contemplam a maioria daquilo que é necessário para se usar no InfoPub na Web.

Ao nível das tecnologias foi possível analisar que duas delas mostram ter uma compatibilidade bastante grande devido ao facto de ambas serem desenvolvidas pela Microsoft (Silverlight e SQL Server). O Flash é uma plataforma que tem um impacto grande na Internet de hoje, mas é preciso perceber se realmente está orientada para o verdadeiro propósito do InfoPub na Web. O Silverlight por seu lado tem a desvantagem de ser uma tecnologia emergente e em constante mudança. Está pouco madura e não se sabe quando nem como irá estabilizar. Tendo em consideração os SGBD é possível verificar que o Oracle pode ter algumas vantagens em termos de performance e de alguns requisitos específicos. Neste ponto é preciso perceber se realmente é vantajoso em relação ao SQL Server tendo em conta todos os outros pontos de comparação. Ainda considerando as tecnologias de SGBD, se a escolha na tecnologia Web recair sobre o Silverlight, a escolha do SQL Server será quase imediata devido à sua vantagem de integração que é decididamente um factor de diferenciação.

Não faltam alternativas para se poder fazer um projecto de qualidade. É necessário saber tomar as escolhas certas e fazer o melhor uso de cada uma das ferramentas para maximizar o tempo produzindo uma aplicação estável e eficiente.

3 Descrição do Projecto

Neste capítulo pretende-se descrever o que é, ou o que será, o InfoPub na Web. Para que este objectivo seja atingido pretende-se seguir uma estrutura lógica que dê a conhecer o percurso de conceptualização do problema.

Inicialmente será feito um pequeno resumo do que é o projecto, seguido do ambiente em que este se insere. Para isto será dado a conhecer um pouco do que existe ao nível de informação ao público nas paragens assim como os geradores de informação que já existem. Também será feita uma pequena referência às aplicações já existentes na empresa onde este projecto foi desenvolvido e a interacção entre elas assim como a sua importância para o funcionamento do sistema existente.

Uma vez concluída esta fase será dada uma pequena ideia daquilo que se pretende que venha a ser uma utilização normal da aplicação, assim como os diferentes utilizadores (actores) que vão interagir com o sistema. Uma vez conhecido isto será elaborada uma lista de requisitos que se pretendem ver concluídos na sua maioria no final deste projecto.

Seguidamente, e definidos estes passos, será explicada a conceptualização do sistema a nível da sua arquitectura lógica, física e da base de dados.

Por fim serão dadas a conhecer as decisões tomadas acerca das tecnologias a utilizar assim como as metodologias e ferramentas que irão contribuir para o desenvolvimento da aplicação.

3.1 O que é o projecto?

Pretende-se que o InfoPub na Web seja uma variante do InfoPub versão Desktop que já existe. Durante o decorrer deste projecto pretende-se desenvolver uma aplicação que seja capaz de mostrar as potencialidades do sistema InfoPub numa aplicação Web.

O InfoPub na Web terá de ser capaz de ter uma Base de Dados que suporte toda a estrutura da definição de uma rede de Transportes Públicos: as suas linhas, percursos, horários etc. Uma vez tendo essa estrutura para a definição da rede é pretendido que as companhias de Transportes Públicos (que serão os utilizadores desta aplicação) sejam

capazes de importar uma definição da sua rede para a Base de Dados do InfoPub na Web.

A importação de dados tem de ser desenvolvida de modo a que o formato de entrada seja o mais standard possível, estando assim ao alcance de várias empresas em vários pontos do mundo. É preciso então fazer durante o desenvolvimento deste projecto, um mapeamento entre o formato de entrada para a estrutura de definição da rede existente no InfoPub na Web.

Uma vez concluída a parte da importação de dados, ao utilizador terá de ser dada a oportunidade de poder editar alguns desses dados através de interfaces construídas na tecnologia Web escolhida para o efeito. Essa edição permitirá pequenas mudanças nos dados introduzidos, assim como a introdução de alguns dados que podem não estar patentes no formato que foi usado na importação desses dados.

Ao utilizador será também permitida uma pequena gestão dos acessos à informação da empresa em questão assim como aspectos de informação acerca dela em outros aspectos relacionados com a imagem e com a parametrização de informação nos dados de saída.

O objectivo final é que com a importação de dados e a gestão destes dados através das interfaces produzidas, o cliente possa ter acesso a alguns geradores genéricos que produzirão informação orientada à paragem ou à linha (mas cujo objectivo será a afixação na paragem) e então ter uma noção das potencialidades do sistema InfoPub através do InfoPub na Web.

A informação presente nos documentos gerados terá como fonte os dados inseridos na importação e possivelmente editados através da interface do InfoPub na Web, produzindo esquemas de linha, informações sobre zonamento e sobre o tempo médio entre determinados segmentos que essa linha percorre.

3.2 Informação ao Público nas Paragens

Muitas empresas de transportes públicos de passageiros disponibilizam informação sobre as suas linhas e percursos assim como os respectivos horários. Essa informação pode ser encontrada em postos específicos das empresas e, eventualmente, nas paragens.

Normalmente nas paragens de transportes públicos de passageiros podemos ver placas identificativas da paragem, normalmente colocadas pela empresa (ou empresas, no caso de haver mais do que uma placa), onde figura o nome da empresa e muitas vezes o código da linha que serve aquela paragem, assim como o destino dessa mesma linha.

Há um crescente número de tipos de folhetos com mapas, horários e outro tipo de informações que normalmente as empresas disponibilizam aos seus utentes nas suas lojas, sites, ou que afixam nas paragens.

Há empresas especializadas na produção de mapas e folhetos informativos com horários, como é por exemplo o caso da FWT, uma empresa Britânica que se dedica há mais de 30 anos a preparar informação na área dos transportes. Esta empresa pratica um princípio que diz: “8 segundos é o tempo máximo que provavelmente será possível conseguir para ganhar a atenção de um utilizador à procura de informação. Use esse tempo da maneira mais eficiente possível” [FWT00]. Juntamente com este lema a FWT

mostra algumas soluções para conseguir ganhar essa mesma atenção, entre elas a disponibilização de informação orientada à paragem onde ela é colocada.

A grande questão de tudo isto é que, na maioria dos casos, esta informação é gerada sem recurso a ferramentas automáticas.

3.3 Geradores de Informação ao Público

A OPT, em Portugal deparou-se com um desafio inovador. Produzir de forma automática informação para ser afixada nas paragens de forma a melhor informar o público sobre os recursos existentes no local.

Os objectivos passavam por ter dois tipos de geradores: um que fosse capaz da produção automática de mapas esquemáticos das linhas por paragem (BusMap) e outro que fosse capaz de produzir espinhas, zonamentos e horários por linha e por paragem (BusSched). Estes projectos enquadram-se no que é hoje designado por InfoPub.

O BusMap teve como fonte de inspiração mapas esquemáticos de linhas por paragem existentes em Londres. A partir daí o projecto sofreu uma série de iterações até chegar à fase onde se encontra hoje, que se encontra numa versão estável e já se encontra a funcionar na STCP no Porto e na Carris em Lisboa.

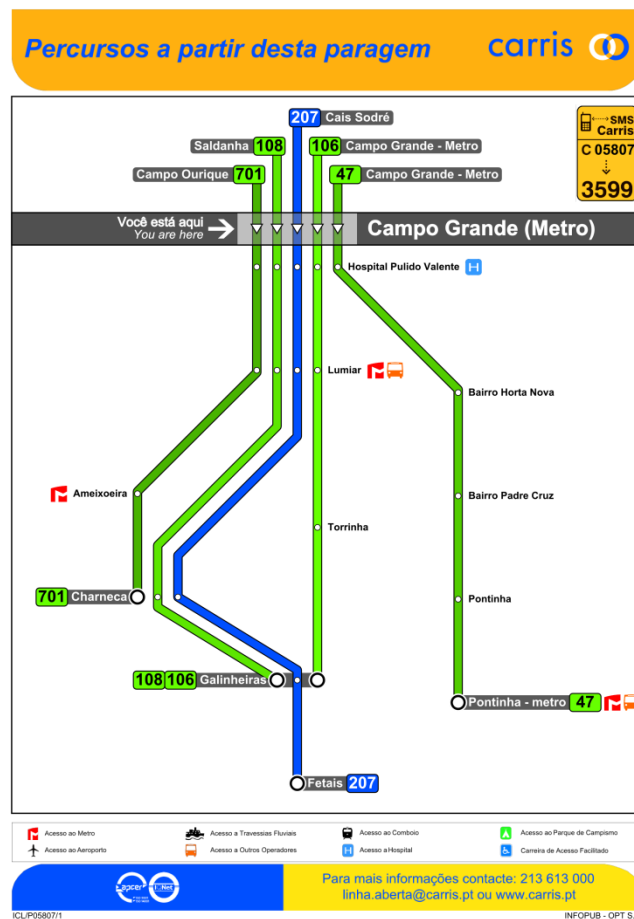


Figura 3.1: Exemplo do BusMap na Carris, em Lisboa

No caso do BusSched a ideia era produzir um documento que ilustrasse uma espinha com todas as paragens (sempre que possível) de uma determinada linha sendo que essa espinha teria informação acerca do zonamento e estimativas de tempo de passagem em determinados troços dessa mesma linha. Igualmente este documento deveria ilustrar as diferentes horas de passagem (ou frequência de passagem) dos veículos que servem essa linha pela paragem à qual o documento se destinava a ser afixado e de acordo com o tipo de dia, visto que os horários variam com os tipos de dia.



Figura 3.2: Exemplo do BusSched na STCP, no Porto

O BusSched na sua primeira versão tinha como fonte de dados a Base de Dados da rede STCP e também os dados na Base de Dados GIST na STCP sendo a relação entre estes dados garantidos pela própria STCP. Este aspecto foi evoluindo de forma a ir de encontro às necessidades dos novos clientes e hoje em dia.

Na sua adaptação para a Carris foi produzido um Gestor de Dados para que os dados referentes à rede fossem importados por ficheiros (três, contendo informação sobre as paragens, as linhas e os percursos), e fosse feita uma associação desses dados aos dados existentes no GIST da Carris para se poderem saber os horários (no caso do BusSched).

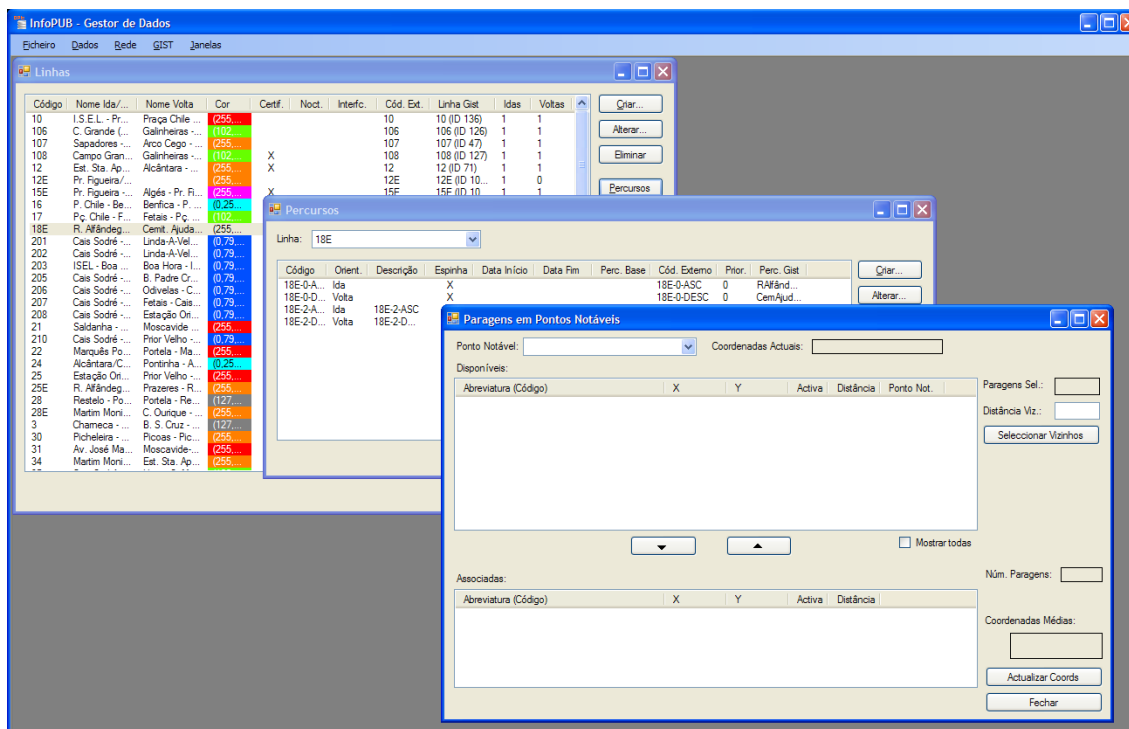


Figura 3.3: Gestor de Dados do InfoPub Desktop

Hoje em dia está a ser desenvolvida uma versão do InfoPub que será completamente independente do GIST e em os horários serão importados através de ficheiros à semelhança do que acontece para o resto da definição da Rede na Carris.

3.4 O GIST no InfoPub Desktop

O GIST é uma parte fundamental das primeiras versões do InfoPub, versão Desktop. Estas primeiras versões apenas importavam de ficheiros externos os dados relativos às paragens, às linhas e aos percursos. Uma vez importados esses dados era feita uma associação entre o que no GIST são os Nós com as paragens correspondentes especificadas no InfoPub.

No GIST não existe a necessidade de especificar cada uma das paragens e por isso o recurso apenas a Nós. Estes Nós são pontos que têm relevância na rede. São pontos de recolha dos veículos, pontos de início ou de fim de uma linha, ou pontos de renição de motoristas ou veículos. Em suma, são os pontos relevantes para que se possa fazer um planeamento adequado da rede sem que se sobrecarregue o sistema.

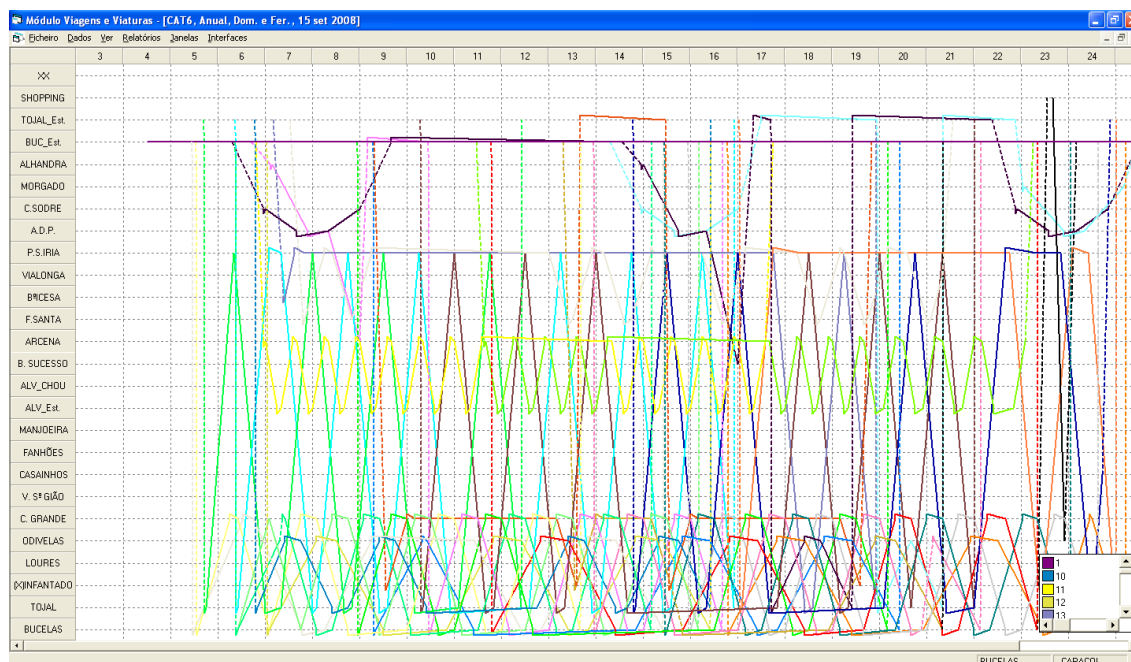


Figura 3.4: Planeamento de viagens no GIST

O GIST também está responsável por planear, nas empresas onde se encontra instalado, os horários referentes às viagens das viaturas. É nesse ponto que depois o InfoPub versão Desktop vai buscar a informação relativa aos horários para apresentar nos documentos gerados. Isso é feito através da interpolação entre os nós e as paragens e a inferência dos restantes tempos de passagem recorrendo a estatísticas que usam como parâmetro distâncias entre paragens, horas do dia, entre outros.

3.5 Utilização da Aplicação

Um utilizador que, pela primeira vez utilize a aplicação terá que efectuar alguma importação de dados. Sendo esta aplicação um protótipo funcional ainda não será possível que todos os dados sejam importados através do sistema de importação que se implementado nem sequer apenas pelas interfaces de edição de dados que também se pretende que façam parte do programa.

Assim, este utilizador começará normalmente por importar um conjunto de dados relativos à definição da sua rede. Esta importação será feita num dos formatos disponíveis para o efeito.

Seguidamente o utilizador pode pretender alterar alguns desses dados através dos formulários existentes, quer por detecção de erros, quer por ajustes necessários fazer à rede. O utilizador pode igualmente pretender que sejam adicionadas informações que os formatos de dados não permitam mas que façam parte da especificação da rede e que os formulários pretendam editar.

Seguidamente o utilizador terá a opção de produzir os documentos de informação ao público usando os geradores disponíveis para o efeito, tendo como meio os campos de parametrização que permitirão definir opções como a linha e as paragens para as quais é pretendido que essa informação seja criada.

Existe uma série de outras acções que se pretende que o utilizador possa fazer no decorrer da utilização da aplicação. Tal passa por alterar os nomes relativos à empresa que dispõe os serviços ou o seu logótipo. Também se pretende que seja possível fazer uma gestão dos utilizadores que têm acesso aos dados daquele cliente específico.

3.6 Utilizadores do Sistema

O InfoPub na Web apenas contempla dois tipos de utilizadores. Os utilizadores normais e os administradores, sendo que não existem diferenciações dentro de cada uma destas categorias.

Os administradores têm acesso a um menu que permite fazer uma gestão dos Providers que acedem ao sistema. Podem alterar alguns dados como o nome de registo dos Providers e a data de limite de acesso ao serviço. Podem igualmente fazer uma gestão dos utilizadores com poderes administrativos.

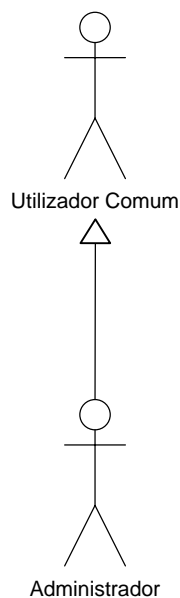


Figura 3.5: Diagrama dos utilizadores do sistema

Os administradores têm também uma funcionalidade extra que pode ser útil na assistência técnica aos utilizadores. Essa funcionalidade consiste em entrar no sistema com o seu nome de utilizador e a sua password, mas como fazendo parte de um qualquer provider adicionado “@nome_do_provider” a seguir ao seu nome de utilizador. É autenticado de acordo com a sua password mas entra no sistema identificado como utilizador daquele provider específico. Isto permite que este administrador acesse à interface exactamente com os mesmos dados que são apresentados aos utilizadores desse Provider, tornando a assistência técnica e a resolução de problemas num processo mais fácil.

Os utilizadores comuns têm acesso a todas as funcionalidades de sistema, excepto às de administração. O acesso destes utilizadores restringe-se igualmente aos dados pertencentes ao Provider que deu acesso a esse mesmo utilizador.

3.7 Requisitos do Sistema

Neste ponto serão referidos os requisitos funcionais e não funcionais que se pretendem ver atingidos no final deste projecto.

3.7.1 Requisitos Funcionais

A aplicação, como referenciado anteriormente, é composta por vários módulos. É mais importante que todos estejam funcionais no final do projecto, contudo existem funcionalidades dentro dos módulos que têm mais importância relativamente a outras.

Em primeiro lugar é necessário que haja um sistema de controlo de acessos e de gestão de utilizadores. Esse acesso vai permitir o acesso controlado ao sistema dos Providers que adquirirem o serviço e posteriormente dos utilizadores criados por esse Provider.

Tabela 3.1: Prioridade dos Requisitos funcionais

Requisitos funcionais	Prioridade
Fazer importação num formato standard de dados	Alta
Fazer importação em mais que um formato de dados	Baixa
Desenvolver uma gestão de utilizadores	Média
Desenvolver gestão de administração	Baixa
Formulários simples de edição de uma tabela da BD	Alta
Formulários que envolvam várias tabelas	Média
Elementos acessórios nos formulários de edição de dados	Baixa
Parametrização de dados para o Gerador	Alta
Gerador de um documento genérico	Alta
Gerador de uma espinha simples	Alta
Gerador de uma espinha com vários elementos	Média
Gerador de mais tipos de espinha	Baixa
Gerador de horários	Baixa
Gerador de mapas esquemáticos à paragem	Baixa

Depois disso existem os três grandes módulos do sistema. Cada um desses módulos tem as suas necessidades específicas e a sua hierarquia na importância das funcionalidades a implementar.

3.7.1.1 Importador de Dados

O Importador de Dados é uma parte importante da aplicação. Sem a importação de dados, não há possibilidade de fazer uso dos outros módulos. Isto significa que é um módulo que tem prioridade alta.

É importante que se consiga fazer a importação de dados pelo menos em um dos formatos standard que foram previamente analisados. Ainda assim existem pequenos detalhes que ainda podem vir a ser implementados.

Nenhum dos formatos de importação tem o nível de detalhe da Base de Dados que será implementada. Da mesma maneira existem elementos especificadas nos formatos de entrada que não estão contempladas na Base de Dados. O objectivo principal é conseguir importar a informação mais importante para o sistema e que permita sobre este módulo construir os restantes módulos do sistema.

As Paragens, Linhas e Percursos são dos pontos mais importantes e que têm obrigatoriamente de ser importados. Seguidamente existem outros pontos como os Pontos de Mapa, os Calendários e os Horários de Passagem que apesar de serem importantes para o sistema não é imperativo que a sua importação seja feita numa fase inicial. Por fim existe a especificação geográfica de cada um dos percursos que nesta fase do projecto é algo que não pode ser sequer considerado um requisito.

3.7.1.2 Gestor de Dados

O Gestor de Dados é o módulo onde existe maior margem para serem desenvolvidas mais ou menos funcionalidades pois uma vez estando os dados inseridos na Base de Dados passa a existir informação suficiente para os geradores funcionarem.

A implementação do Gestor de Dados tem então alguns objectivos principais como mostrar as potencialidades da tecnologia escolhida e provar que é possível fazer a gestão deste tipo de informação através da Internet e em segundo lugar acrescentar informação adicional não disponibilizada nos formatos de importação, ou fazer pequenas alterações ou correcções aos dados inseridos para que em caso de erro não seja necessário proceder a uma nova importação.

Neste ponto a importância dos formulários de edição de dados para cada tipo de informação é relativamente idêntica ao da importação, podendo também atribuir-se alguma importância a dados que não são contemplados nos formatos de importação analisados e que terão interesse constar da Base de Dados para o Gerador. As Interfaces são um ponto importante dos geradores e que não estão especificados nos formatos de importação elevando a prioridade de desenvolvimento de uma interface que permita fazer a sua gestão. Também são de prioridade elevada os formulários de edição de dados relativos às Linhas, Paragens e que contemplam detalhes relacionados com estes.

3.7.1.3 Gerador

O Gerador é também uma parte fundamental deste projecto visto que este é o módulo responsável por produzir o resultado final da aplicação. O Gerador deve pedir ao utilizador para fazer uma selecção dos dados sobre os quais quer gerar informação (Paragem e Linha por exemplo) e a partir daí gerar um folheto informativo que corresponda à informação da Base de Dados e que seja capaz de informar os utilizadores.

Existem vários tipos de informação que podem estar contidos no folheto informativo: uma “espinha” representativa do conjunto de paragens que compõem uma linha, o horário de passagem dos transportes de uma determinada linha por uma determinada paragem ou então um esquema que inclua os percursos das linhas que servem determinada paragem.

Aqui é pretendido que pelo menos se consiga produzir um destes tipos de informação, tendo como objectivo ilustrar o que pode vir a ser atingido. Gerar outro tipo de informação é já um objectivo de prioridade mais baixa e ainda depois desse conseguir combinar vários tipos de informação num mesmo folheto informativo.

O Gerador deverá ser capaz de gerar o documento ou o conjunto de documentos seleccionados pelo utilizador e devolvê-los num ficheiro comprimido quem também terá contido uma pasta com as imagens incluídas nesses folhetos informativos.

3.7.2 Requisitos Não Funcionais

A par dos requisitos funcionais existem alguns requisitos não funcionais que devem também ser mencionados.

Tabela 3.2: Prioridade dos Requisitos não Funcionais

Requisitos não funcionais	Prioridade
Usabilidade	Alta
Eficiência	Alta
Fiabilidade	Alta
Manutenção	Alta
Segurança	Alta

3.7.2.1 Usabilidade

É importante que a interface com o utilizador seja intuitiva e fácil de usar. Quanto melhor forem estes parâmetros, mais fácil é que o utilizador se sinta à vontade com a aplicação, tenha mais facilidade em usá-la e se sinta mais satisfeito com o resultado no fim da sua utilização.

Por estes motivos é necessário que esta seja agradável à vista, sem introduzir animações ou efeitos exagerados, tendo um esquema de cores suave e uma usabilidade fácil em que o utilizador possa encontrar facilmente a funcionalidade que procura.

3.7.2.2 Eficiência

O sistema deverá apresentar uma boa eficiência e um bom tempo de resposta.

A importação é um exemplo de uma funcionalidade que deverá apresentar boa eficiência, tendo em conta a quantidade de dados envolvidos especialmente no caso dos tempos de passagem, sendo que esta deverá ser feita da forma mais eficaz e rápida possível.

Ao mesmo tempo os formulários de edição devem ter uma boa e rápida resposta nas suas comunicações com o servidor e consequentemente com a Base de Dados.

Isto tudo deverá ser garantido gastando o mínimo de recursos possível para obter uma optimização dos mesmos.

3.7.2.3 Fiabilidade

Em termos de fiabilidade a aplicação deverá estar preparada para lidar com possíveis falhas, permitindo ao utilizador estar tranquilo na sua utilização e seguro de que as suas operações não serão perdidas.

Para ajudar a isto é necessário que o sistema devolva mensagens de erro explicativas das possíveis falhas de modo a que o utilizador possa corrigir a sua acção ou alternativamente pedir assistência técnica acerca do problema ocorrido.

3.7.2.4 Manutenção

A aplicação deverá ser desenvolvida de maneira a que a sua manutenção possa ser garantida sem comprometer o normal funcionamento da mesma. A sua arquitectura deverá estar construída de maneira a permitir alterações na mesma e até extensões e melhoramentos sem que seja afectado o seu normal funcionamento ou os dados existentes.

3.7.2.5 Segurança

A segurança deve ser um ponto importante da aplicação uma vez que o utilizador querera sentir que os seus dados estão em segurança, que não serão alvos de ataques e que outros utilizadores ou pessoas externas ao sistema tenham acesso ao mesmo.

Assim é necessário garantir que apenas os administradores de sistema têm acesso a todos os dados e apenas os utilizadores autorizados por esse provider tenham acesso aos dados armazenados por parte de esse Provider e possam usufruir dos serviços adquiridos pelo mesmo.

3.8 Arquitectura Lógica

A arquitectura Lógica do InfoPub na Web pode perfeitamente ser descrita pela decomposição horizontal representada na Figura 3.6.

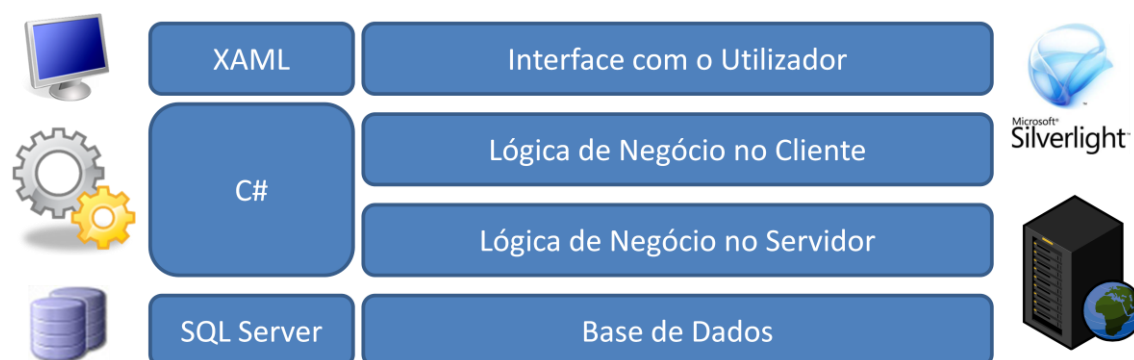


Figura 3.6: Esquema da Arquitectura Lógica Horizontal

Esta imagem mostra que o sistema está bem dividido em camadas distintas. No topo existe a interface com o utilizador. Esta interface com o utilizador está definida em ficheiros XAML. Logo abaixo desta camada existe a Lógica de Negócio no cliente que é definida em código C#. Estas duas camadas são executadas no cliente através do Microsoft Silverlight. O Silverlight executa a lógica de negócio e exibe a interface com

o utilizador através da interacção entre o código C# e o código XAML definidos separadamente.

Toda a parte do sistema definida em Microsoft Silverlight tem capacidade de execução no cliente mas ainda assim precisa de informação que reside no servidor. O servidor tem a sua própria lógica de negócio que funciona num servidor IIS (Internet Information Services) sob a tecnologia ASP .NET. Esta camada de lógica de negócio é sobretudo necessária para receber os pedidos do cliente para acesso à Base de Dados e para devolver esses resultados ao cliente. Adicionalmente a lógica de negócio no servidor encarrega-se de algum processamento mais pesado como é o caso da geração de documentos.

A camada mais baixa, a camada dos dados, é garantida por um SGBD SQL Server que contém todos os dados necessários e é a fonte persistente onde esses dados se encontram.

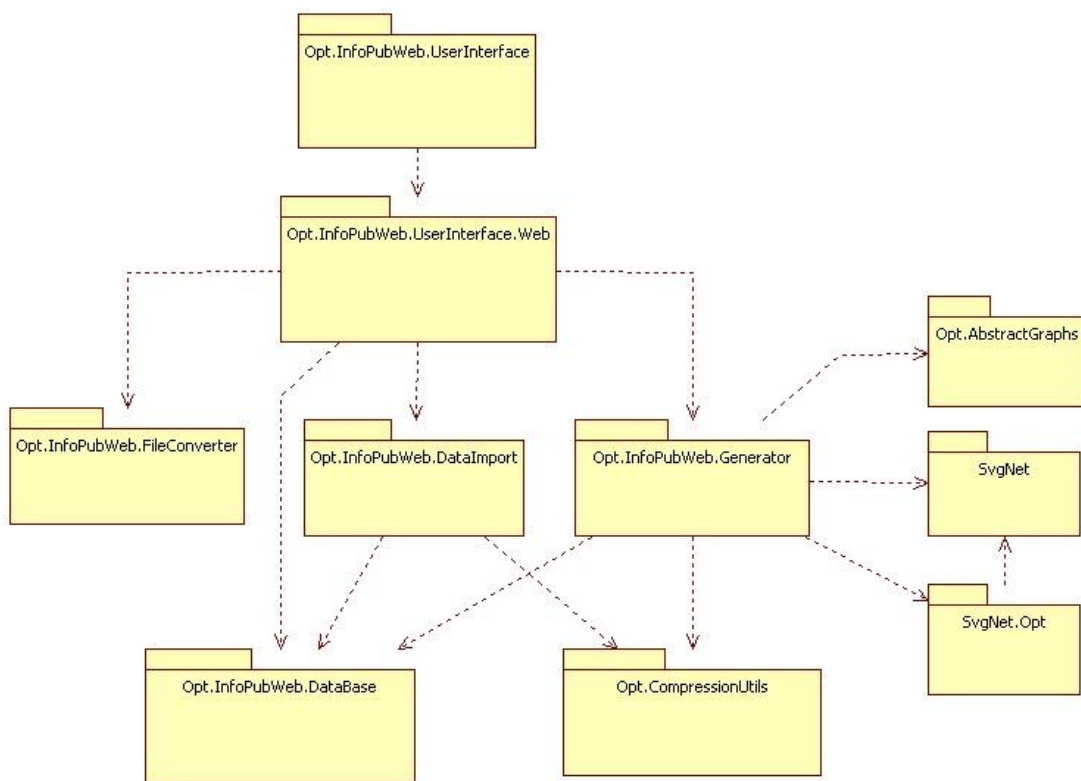


Figura 3.7: Esquema da Arquitectura Lógica Vertical

Na Figura 3.7 podemos ver uma decomposição vertical da arquitectura lógica do InfoPub na Web. No topo está o módulo da Interface com o utilizador que corre do lado do cliente. Esse módulo comunica com o módulo UserInterface.Web, sendo que este último é responsável por tratar os pedidos dos clientes e os processar no servidor. Este módulo é complexo e sendo assim dependente de várias outras partes.

Existem três partes importantes da aplicação: o Importador, o Gestor de Dados, e o Gerador.

O Importador usa os módulos FileConverter, DataImport e o DataBase. O módulo FileConverter é usado no caso do ficheiro de importação estar no formato TransXChange de modo a convertê-lo no formato GTFS. O módulo DataImport também é exclusivo do Importador e é responsável por fazer o mapeamento dos dados que vêm no formato GTFS para o esquema da Base de Dados implementada. Este módulo utiliza outro módulo, o CompressionUtils, que tem como função descomprimir o ficheiro *.zip em onde se encontram os vários ficheiros que compõem o formato GTFS.

O Gestor de Dados está sobretudo implementado nos módulos UserInterface e UserInterface.Web. Estes dois módulos comunicam entre si, e o módulo UserInterface.Web usa o DataBase para aceder à Base de Dados e ir buscar os dados lá contidos ou fazer as alterações necessárias decorrentes da utilização do Gestor de Dados.

Por fim existe o Gerador que usa o módulo Generator para fazer grande parte do processamento da geração dos dados. Também acede ao módulo DataBase de modo a ir buscar a informação necessária à geração dos dados. Este Gerador usa a classe AbstractGraphs para criar um documento deste tipo. Uma vez produzido o documento AbstractGraphs é convertido para um ficheiro SVG usando os módulos SvgNet e SvgNet.Opt. No fim de criados todos os ficheiros relativos a uma geração, estes são comprimidos junto com as imagens que utilizam recorrendo para isso ao módulo CompressionUtils.

O módulo DataBase é usado por todas as partes da aplicação para interacção com a Base de Dados. Este módulo é também responsável por fazer o mapeamento da Base de Dados para Classes para que mais facilmente os dados possam ser usados, e as relações entre eles possam ser feitas na execução da aplicação e não apenas no momento da inserção na Base de Dados.

3.9 Arquitectura Física

O InfoPub na Web usa a tecnologia Microsoft Silverlight que precisa de estar instalada em cada uma das máquinas destinadas a correr a aplicação. O acesso à aplicação é feito através da Internet, onde o cliente acedendo ao endereço da aplicação faz um pedido ao servidor que lhe devolve a página onde a aplicação corre como Plug-in Silverlight.

Uma das vantagens desta arquitectura física é o facto de um cliente poder aceder através de vários computadores e de uma forma deslocalizada. Ou seja, tudo o que é preciso para aceder à aplicação é uma ligação à Internet, um browser que suporte o Silverlight e ter o Silverlight instalado, algo que é relativamente acessível especialmente quando comparado com outras tecnologias e ferramentas que normalmente são usadas nas versões Desktop deste tipo de aplicações acrescentando o facto de muitas dessas ferramentas serem pagas e o plug-in Silverlight ser completamente livre de encargos.

Qualquer pedido de dados ao servidor é efectuado sempre da mesma maneira, sendo que o cliente faz um pedido através da Internet ao servidor. O servidor responde e se tiver necessidade faz uma chamada à Base de Dados que pode estar localizada na mesma máquina ou numa máquina diferente acessível através de uma rede LAN.

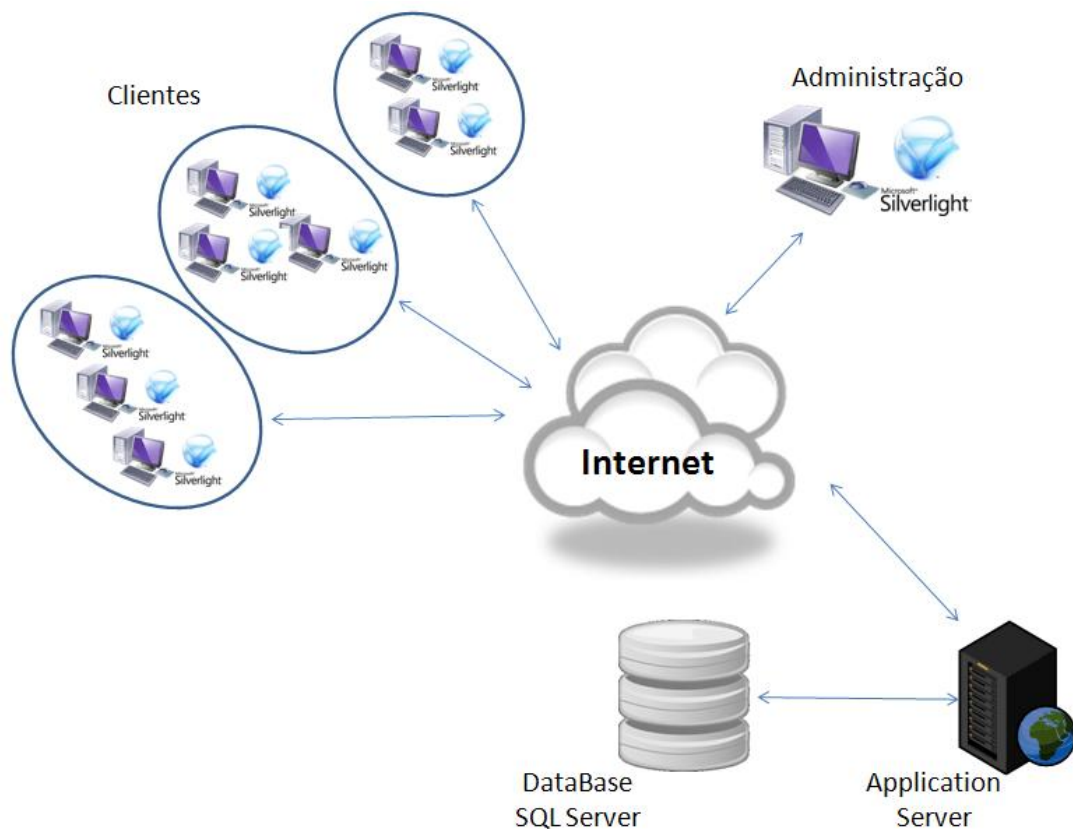


Figura 3.8: Esquema da Arquitectura Física

3.10 Arquitectura da Base de Dados

Na Figura 3.9 podemos ver um esquema representativo da Arquitectura de Base de Dados presente no InfoPub na Web. Este esquema espelha a definição da Rede no InfoPub na Web, assim um pouco como a gestão de utilizadores necessária neste sistema.

Este esquema foi obtido após uma série de iterações que começaram com o esquema do InfoPub versão Desktop existente. Após isso foram feitas as alterações necessárias para se tornar compatível com o projecto em questão e posteriormente foram efectuados pequenos ajustes que permitem cada vez melhor ajustar este esquema à realidade do InfoPub na Web.

Algumas partes desta arquitectura não foram exploradas no âmbito do projecto por serem requisitos de baixa prioridade mas, de qualquer modo, é pertinente que se encontrem na Base de Dados, para que esta esteja preparada para futuras evoluções do Projecto.

Todo este esquema se centra no Provider. Este é um cliente do serviço e pode definir a sua rede com a definição que se pretende implementar neste esquema de base de dados. O Provider é constituído por vários utilizadores (User).

Um Provider começa a definir a sua rede através das Linhas. Estas são da sua responsabilidade e representam a sua oferta de serviço aos clientes. Uma Linha (Line) é composta por um ou mais Percursos (Path). Vários Percursos podem pertencer à mesma Linha e percorrer diferentes trajectos dependendo da hora do dia ou do dia da semana. Cada Percurso é por sua vez definido por uma Sequência de Paragens (PathStops). As

PathStops são definidas por um conjunto Percurso-Paragem. As Paragens são também algo que o Provider pode definir (Stop). Para se poder definir a oferta que cada Linha tem, são associados a cada PathStop as horas de passagem (Passing) dos veículos por essa Paragem, estes que pertencem a um Calendário (Schedule), pois as horas de passagem definem uma hora do dia e o seu Calendário define os dias da semana, épocas do ano etc.

Existem alguns conceitos acessórios que também ajudam a definir toda a rede. As Linhas podem ter associada uma cor (LineColor) que ajuda a identificar conjuntos de Linhas. A Rede pode também ter vários tipos de Zona (ZoneType), ou seja, cada Zona (Zone) pertence a um ZoneType. Cada PathStop pode pertencer a várias Zonas (em princípio pretende-se que seja uma de cada tipo, mas este facto não é obrigatório).

Por último, a rede define-se por alguns Pontos de Interesse (PointofInterest) que são pontos relevantes da Rede que podem conter monumentos ou outros locais de interesse. Os Pontos de Mapa (MapPoints) são conjuntos de paragens que por norma se situam numa área geográfica próxima umas relativamente às outras, para determinar um ponto com várias paragens. Existem também ainda as Interfaces que definem pontos de ligação nas paragens ou identificam características especiais de uma Linha.

Uma definição mais detalhada de cada uma das tabelas da Base de Dados e dos seus campos encontra-se no Anexo A.

Tabela 3.3: Lista das Tabelas da Base de Dados

Tabela	Prioridade
Provider	Cliente que adquiriu um serviço ou serviços
User	Utilizador pertencente a um provider
PointOfInterest	Ponto de Interesse na Rede
Stop	Paragem de Autocarro
Schedule	Calendário que define tipos de dia da semana e época do ano
ZoneType	Tipos de Zona
Zone	Zonas
Line	Linhas. Serviços disponibilizados na Rede
Path	Percurso que pretence a uma Linha
LineColor	Cor de uma Linha
LineColorShade	Conjunto de gradientes de uma Cor
PathStop	Conjunto Percurso-Paragem
Interface	Ligações nas Paragens ou propriedades de uma Linha
InterfacesStop	Ligação das Paragens às Interfaces
MapPoint	Ponto de Mapa. Conjunto de várias paragens próximas.
InterfacesMapPoint	Interfaces dos Pontos de Mapa
Passing	Horas de Passagem das viaturas pelas paragens
PathStopZone	Ligação dos PathStops às Zonas

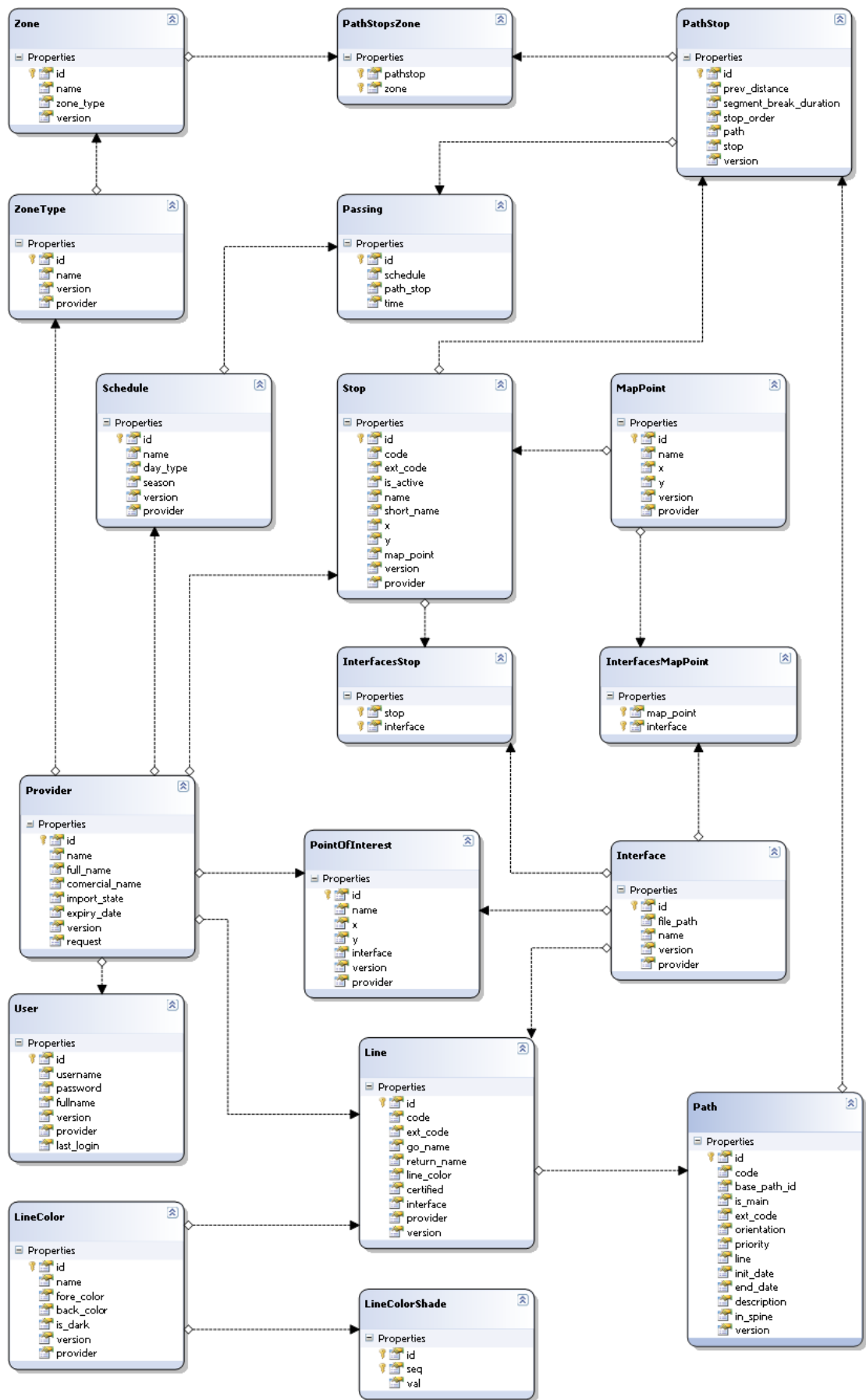


Figura 3.9: Esquema da Base de Dados InfoPub na Web

3.11 Escolhas Tecnológicas

Analisando todos os prós e contras das várias tecnologias foi decidido optar pela utilização do Silverlight juntamente com o SQL Server.

Tendo em conta todos os factores apresentados na análise das tecnologias e o âmbito da aplicação a desenvolver, a escolha do Silverlight foi quase imediata tendo em conta as suas características, facilidade de desenvolvimento, debugging e todas as outras vantagens anteriormente enumeradas.

Tendo a tecnologia Web escolhida, a grande dúvida era qual o SGBD a escolher. Por um lado, ao escolher Oracle e usando a BD existente no InfoPub versão Desktop, havia a grande vantagem de os geradores estarem completamente funcionais e adaptados a essa Base de Dados.

Por outro lado escolher SQLServer significa uma facilidade maior de desenvolvimento do produto. Nomeadamente as tecnologias de integração entre o Silverlight e o SQLServer como o WCF e o Linq to SQL levam também a uma maior liberdade para serem escolhidos caminhos diferentes do produto original e a uma investigação de tecnologias até agora não usadas na empresa onde decorreu o projecto. O grande factor contra esta decisão era o facto de os “Geradores” (bibliotecas que são usadas para a criação dos ficheiros SVG com a informação ao público) terem de ser alterados e adaptados à realidade SQLServer.

Ainda assim os factores a favor do SQLServer acabaram por pesar mais do que os factores contra. Tendo em conta que também muito do código dos Geradores pode ser aproveitado na sua reformulação para o InfoPub na Web.

Assim sendo ficou feita a escolha e será com base nestas tecnologias que todo o desenvolvimento foi feita.

3.12 Metodologias e Ferramentas

Na OPT, não é seguida à risca nenhuma metodologia de desenvolvimento de software, no entanto é usada uma versão adaptada das metodologias tradicionais. Inicialmente é feito um levantamento de requisitos para cada projecto, seguidamente são estabelecidos os modelos conceptuais e analisam-se os casos de uso mais críticos. A partir daqui existe uma certa flexibilidade no desenvolvimento, passando este por estabelecer metas semanais, com vários *milestones* ao longo do projecto.

Várias ferramentas foram usadas no desenvolvimento da aplicação InfoPub na Web. Tanto na fase de planeamento do projecto, como posteriormente no desenvolvimento e ainda outras para se poder fazer *debugging* e ver o resultado final.

- **StarUML** – Usado no planeamento para produzir diagramas de classes e modelos conceptuais.
- **InkScape** – Usado para ver os documentos SVG gerados e ajudar na sua construção.
- **Visual Studio 2008** – Principal ferramenta de desenvolvimento, compilação e debugging. Aqui é editado todo o código C# e também algum código XAML.

- **Expression Blend 2** – Esta ferramenta é um editor visual de XAML, ela ajuda a produzir as interfaces e é possível também fazer edição directa no código XAML através desta ferramenta. Trabalha em conjunto com o Visual Studio podendo ser invocada através do mesmo
- **Silverlight Tools for Visual Studio 2008** – Permite o desenvolvimento de Silverlight no Visual Studio 2008. Alguns outros componentes foram também usados embora não se possam verdadeiramente incluir nas ferramentas, mas fazem parte do desenvolvimento da aplicação, como algumas bibliotecas externas disponibilizadas com conteúdos Silverlight.
- **Microsoft SQL Server 2008 Management Studio** – Esta ferramenta permite criar e gerir a Base de Dados, ajudando na detecção de erros e de outros problemas na mesma. Aqui é possível aceder aos dados directamente e introduzi-los se for necessário fazer algum teste que não seja possível efectuar introduzindo dados através da interface da aplicação.

3.13 Resumos e Conclusões

Neste capítulo foi falado acerca de vários pontos acerca do trabalho como o seu planeamento e o seu desenvolvimento. O objectivo passa por fazer uma versão Web da aplicação InfoPub versão Desktop existente. Para isto foi revista um pouco da história da informação ao público referente aos transportes públicos nas respectivas paragens.

Seguidamente foi feita uma abordagem ao que será a arquitectura física, lógica e da Base de Dados. A nível físico esta aplicação é semelhante a qualquer outra aplicação Web, estando alojada num servidor, recorrendo a uma Base de Dados que pode correr numa mesma máquina ou num servidor diferente, com a particularidade que existe código da aplicação Silverlight que corre no cliente.

A arquitectura lógica é uma arquitectura de três camadas: interface com o utilizador, lógica de negócio e acesso a dados com a particularidade de, neste caso, a lógica de negócio se encontrar dividida entre cliente e servidor.

A nível de Base de Dados esta tenta reproduzir toda a especificação necessária para definir a Rede de transportes públicos.

A nível de requisitos funcionais temos três principais módulos que são precisos desenvolver: o importador de dados, o gestor de dados e o gerador. Ao longo deste capítulo é também definido o nível de detalhe que é pretendido que se atinja em cada um destes módulos. A nível de requisitos não funcionais as necessidades focam-se sobretudo ao nível da usabilidade, eficiência, fiabilidade, manutenção e segurança.

Foi feita também uma passagem pelos vários papéis que os utilizadores podem ter nesta aplicação havendo dois tipos: os utilizadores comuns e os administradores. Os utilizadores comuns têm acesso às funcionalidades do sistema para o cliente final. Um administrador por seu lado tem acesso a todas estas funcionalidades mas tem também a capacidade de gerir todo o sistema e de conseguir aceder aos dados de todos os clientes do sistema.

Neste capítulo foi também revelada a decisão acerca das tecnologias investigadas e anteriormente referidas neste relatório. A decisão passou por se usar o Silverlight como tecnologia Web e também o SQL Server como Sistema de Gestão da Base de Dados.

Adicionalmente foi feita uma pequena descrição das ferramentas mais importantes que foram utilizadas ao longo do desenvolvimento desta aplicação.

4 Implementação

Ao longo deste capítulo serão abordadas todas as questões relacionadas com a implementação da aplicação, os métodos usados na resolução de alguns problemas e pormenores pertinentes para o funcionamento da mesma.

Serão abordadas as três grandes componentes do projecto, ilustrando aquilo que efectivamente foi implementado e se encontra funcional à data da escrita deste relatório.

4.1 Importação de Dados

Após a implementação da Base de Dados que já foi mostrada no capítulo anterior, o primeiro passo foi passar à importação de dados. O desenvolvimento desta parte do projecto foi feito independentemente da aplicação Silverlight em si, uma vez que o objectivo era obter uma biblioteca capaz de fazer a importação através de alguns dos formatos de entrada e fazer a sua leitura, correspondência e respectiva escrita na Base de Dados.

4.1.1 Formatos de entrada na importação

Após alguma pesquisa foi descoberta uma ferramenta Open-Source que permite fazer a conversão de ficheiros no formato TransXChange para o formato GTFS. Esta ferramenta, chamada TransXChange2GoogleTransit e que está incluída no projecto GoogleTransitDataFeed. Com esta ferramenta será possível fazer a conversão para o formato GTFS se o utilizador pretender fazer a sua importação de dados em formato TransXChange e posteriormente fazer o processamento dos dados sempre no formato GTFS como ilustra o esquema na Figura 4.1.

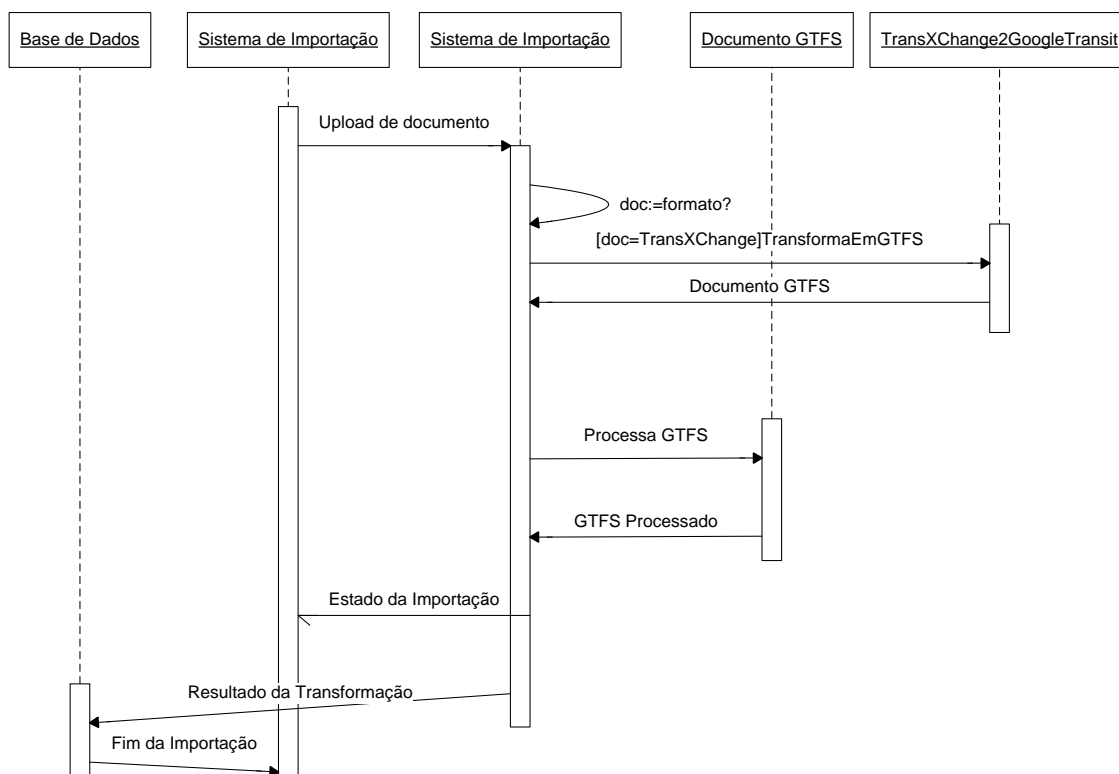


Figura 4.1: Esquema de Importação para a Base de Dados

4.1.2 Leitura dos ficheiros

Nesta altura começa verdadeiramente a importação dos dados existentes nos ficheiros para a Base de Dados. A primeira fase passa por uma leitura dos ficheiros com relevância para a Base de Dados implementada. Nem todos os ficheiros da especificação GTFS interessam para a importação de dados que é necessária fazer neste ponto do trabalho, portanto prosseguiu-se apenas com os que realmente têm utilidade.

Na Figura 4.2 é possível identificar os ficheiros que foram tratados para a sua importação na Base de Dados. Cada um dos ficheiros foi processado, sendo usada a classe `GoogleTransitLineParser` para interpretar cada uma das linhas desses ficheiros visto existirem regras específicas na maneira como os dados se encontram especificados nesses ficheiros.

Para cada um dos ficheiros é criado um objecto da classe correspondente ilustrada do lado direito sendo que as linhas de texto dos ficheiros ficam numa lista de objectos para que estejam em memória e assim se evitar ter de efectuar uma leitura dos ficheiros cada vez que seja necessário aceder aos dados dos mesmos.

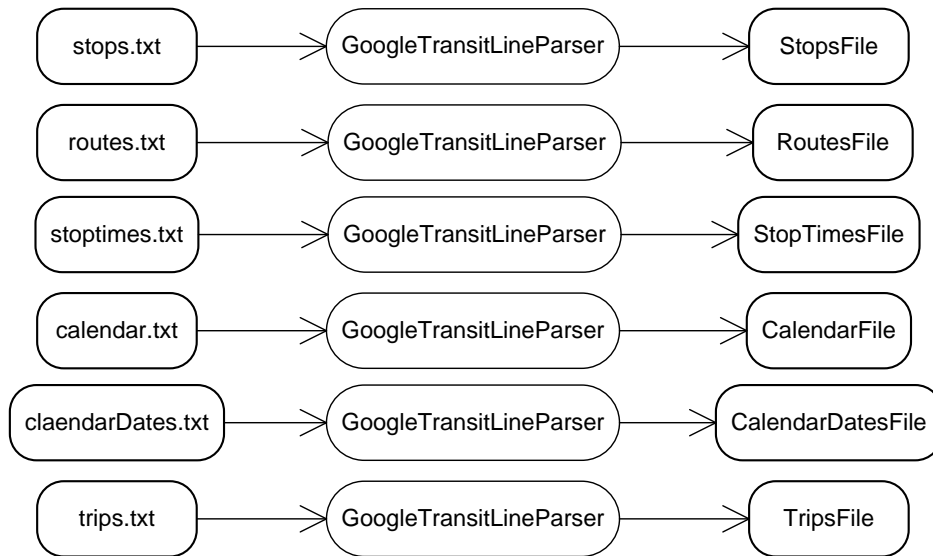


Figura 4.2: Leitura dos Ficheiros para as respectivas Classes

4.1.3 Linhas

A informação relativa às Linhas é a primeira a ser importada para a Base de Dados. O principal factor que leva a este facto é a sua independência relativamente a quaisquer outros dados e que leva a que não sejam precisas outras referências senão para o Provider em questão, algo que já se encontra previamente na Base de Dados.

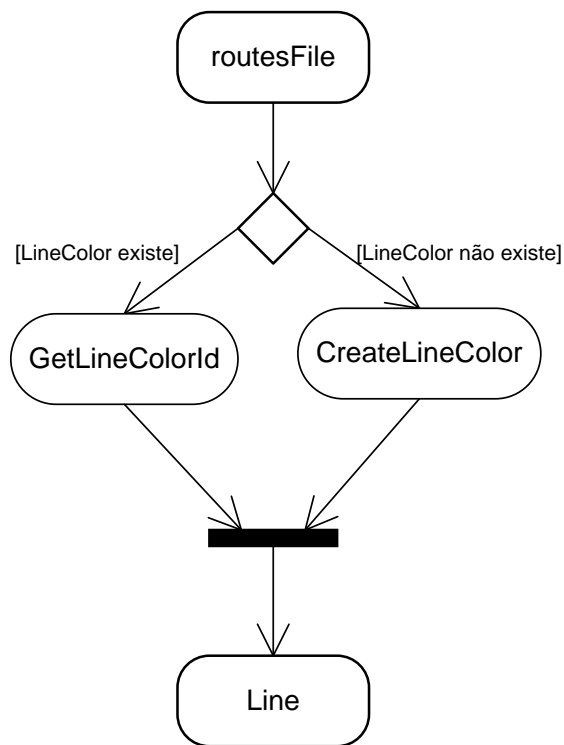


Figura 4.3: Importação das Linhas

Como podemos identificar na Figura 4.3 os dados relativos às Linhas estão num objecto da classe routesFile. Segundo a especificação GTFS a cor relativa a cada uma das linhas encontra-se especificada junto com cada linha. Na Base de Dados da aplicação existe uma tabela para que sejam armazenadas todas as cores, logo isto exige que quando é importada uma linha se faça a verificação se já existe uma cor correspondente na Base de Dados e assim associar essa cor à Linha, caso contrário será criada uma cor nova com as características especificadas no ficheiro e essa nova cor será associada à Linha em questão.

4.1.4 Paragens e Pontos de Mapa

Posteriormente à importação das linhas a aplicação procede à importação das Paragens e dos Pontos de Mapa. Estas entidades também são independentes de outras e é por este motivo que a sua importação é feita nesta altura. A única dependência existente aqui é das Paragens relativamente aos Pontos de Mapa, visto que uma paragem se pode encontrar num Ponto de Mapa.

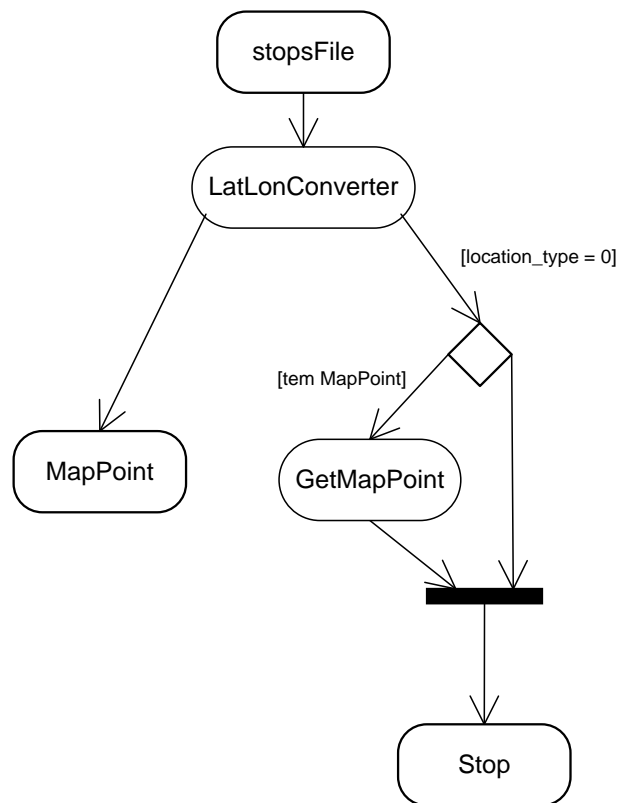


Figura 4.4: Importação dos MapPoints e Stops

Como se pode observar na Figura 4.4 a origem dos dados relativamente aos Pontos de Mapa e às Paragens é o mesmo. Na verdade, segundo a especificação GTFS, os Pontos de Mapa não são mais do que Paragens especiais que podem aglomerar outras Paragens. Assim sendo, este conceito foi adaptado ao conceito de Ponto de Mapa do InfoPub na Web.

Esta diferença entre Pontos de Mapa e Paragens é feita segundo um campo do ficheiro stops.txt, que identifica a linha como referente a um Ponto de mapa se tiver esse campo (location_type) igual a 1, ou então indentificando-a como uma Paragem se esse campo for igual a 0 ou se nem sequer estiver especificado no ficheiro.

Posteriormente a esta identificação é preciso passar por um processamento das coordenadas geográficas. Tendo em conta que os dados podem ser provenientes de uma especificação original TransXChange, por vezes as coordenadas geográficas estão especificadas no sistema OSGB36, ao contrário do que é pretendido que seja guardado na Base de Dados (formato WGS84). Desta maneira é identificado o formato das coordenadas e em caso de necessidade é feita a sua conversão para o formato WGS84.

Após isto é feita a importação dos Pontos de Mapa e de seguida a importação das Paragens, que assim já têm as referências para os Pontos de Mapa caso haja necessidade de recorrer às mesmas.

4.1.5 Zonas

A importação das Zonas é feita directamente do ficheiro que contém a informação relativa às Paragens. A Zona de cada Paragem vem identificada a seguir à mesma, contudo nesta altura da importação o objectivo é apenas identificar todas as zonas existentes no Sistema para que estas fiquem guardadas na Base de Dados quando as suas referências forem solicitadas.

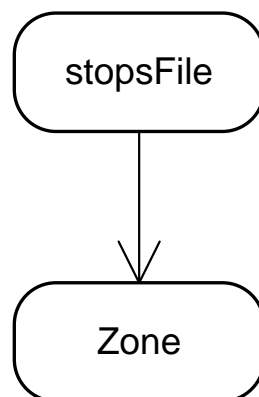


Figura 4.5: Importação das Zonas

Como é possível identificar na Figura 4.5 a importação é feita directamente, apenas sendo verificado se essa zona já existe na Base de Dados ou não. Caso exista é ignorada, em caso contrário é adicionada.

É contudo necessário fazer referência ao facto de o formato GTFS não especificar Tipos de Zona. Devido a este facto quando se pretende introduzir a primeira zona é adicionado um Tipo de Zona genérico que passa a ser usado em todos os dados de importação. Os tipos de Zona são algo que não foi usado nesta fase da aplicação mas que fazem parte da definição da Rede e podem vir a ser aproveitados com o avançar do projecto em desenvolvimentos futuros.

4.1.6 Horários

Os horários são dados que muito à semelhança dos já falados nos tópicos anteriores também estão independentes de qualquer outro tipo de dados. Conforme a Figura 4.6 ilustra estes dados são recolhidos do conjunto de dados de CalendarFile.

Os dados aí presentes são então adaptados aos campos da Base de Dados e assim é feita a sua importação de uma maneira bastante directa.

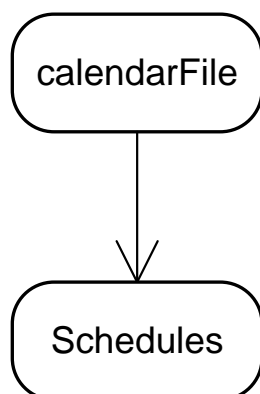


Figura 4.6: Importação dos Schedules

4.1.7 Percursos, Paragens por Percurso e Tempos de Passagem

Esta é a parte mais complexa da importação de dados do formato GTFS para a Base de Dados InfoPub na Web devido a algumas diferenças no modo de especificação da Rede.

A grande diferença entre as duas especificações é o facto de o formato GTFS considerar cada viagem de uma viatura como um único Percurso, algo que não é o pretendido no InfoPub na Web. Deste modo é necessário que todas as viagens correspondentes a um mesmo Percurso sejam identificadas e agrupadas nesse percurso. A partir daí será possível identificar a sequência de Paragens que constituem um Percurso e também adicionar os Tempos de Passagem às Paragens por Percurso respectivas.

Esta operação é demorada devido ao facto de os dados relativos à ordem da sequência de paragens nos percursos estar incluída apenas no ficheiro com os dados relativos aos tempos de passagem. Isto verifica-se como um problema pois este é um ficheiro com uma quantidade muito grande de dados, especialmente quando comparado com os outros ficheiros que fazem parte desta especificação.

O algoritmo que identifica os Percursos redundantes passa por uma estratégia de agrupar as passagens com a mesma identificação e gerar um código único a partir da sequência de paragens que forma esse percurso. No fim desta operação comparam-se os vários códigos únicos dos vários percursos, agrupam-se os percursos iguais e eliminam-se os que estão repetidos, juntando toda a informação sobre as paragens num só percurso.

Depois é necessário associar toda a informação e inseri-la na Base de Dados. Se algum percurso já existir e for importada informação acerca desse percurso, toda a informação anterior é substituída pela nova informação. A comparação dos percursos é

feita pelo path_id que na Base de Dados fica guardada na tabela Path no campo ext_code.

Na Figura 4.7 pode ver-se um diagrama exemplificativo das fontes de informação deste conjunto de dados, o processamento a que estão sujeitas e o resultado final em termos de objectos durante o processamento, sendo que esses são os objectos inseridos na Base de Dados.

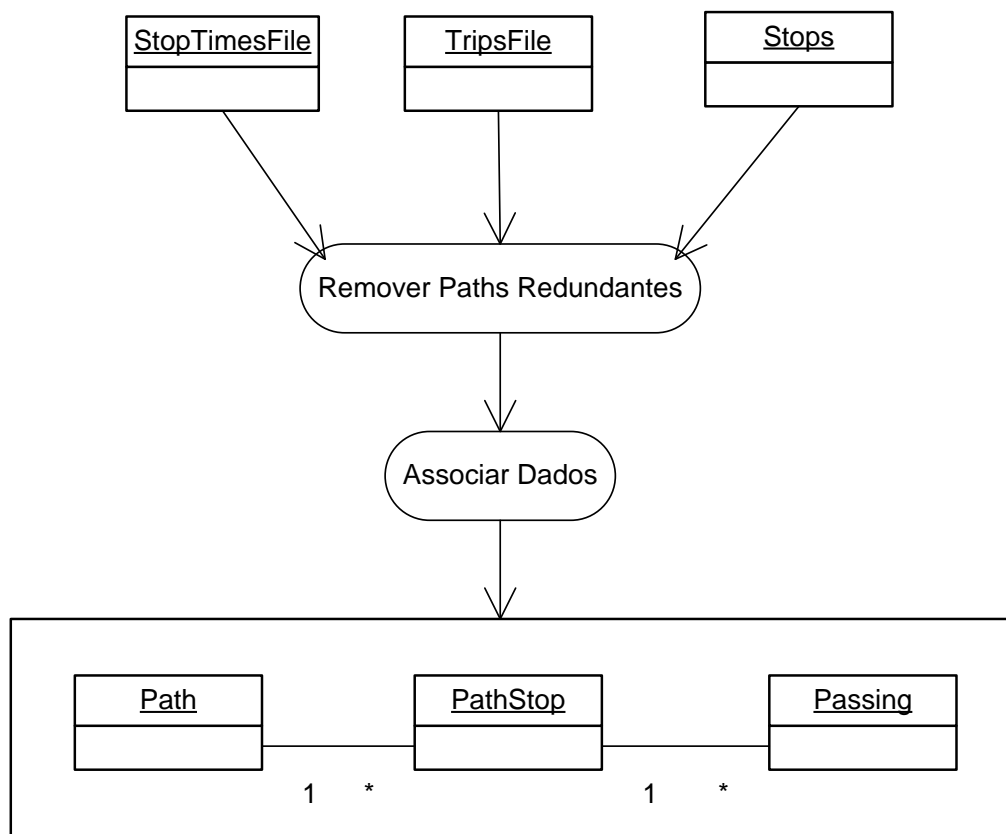


Figura 4.7: Importação dos Paths, PathStops e Passings

4.1.8 Correspondências na Base de Dados

Neste subcapítulo é pretendido ilustrar através de uma série de tabelas as correspondências que são feitas entre a informação contida nos ficheiros de formato GTFS e os campos da Base de Dados.

4.1.8.1 Ficheiro stops.txt

Este ficheiro tem destinos diferentes consoante uma das suas propriedades. Como explicado anteriormente se a propriedade location_type tiver o valor 1 essa ocorrência é considerada um Ponto de Mapa, caso contrário é considerada uma Paragem.

- Se a propriedade location_type tiver o valor 1:

Tabela 4.1: Correspondência stops.txt para a Base de Dados (MapPoints)

Campo	Correspondência na Base de Dados
name	MapPoint.name
stop_lat	MapPoint.x
stop_lon	MapPoint.y

- Se a propriedade location_type tiver o valor 0 ou não existir:

Tabela 4.2: Correspondência stops.txt para a Base de Dados (Stops)

Campo	Correspondência na Base de Dados
stop_id	Stop.ext_code
stop_code	Stop.code
stop_name	Stop.name
stop_lat	Stop.x
stop_lon	Stop.y
zone_id	Pode definir Zone.name
parent_station	Stop.map_point

4.1.8.2 Ficheiro routes.txt

Este ficheiro tem informação que diz respeito na sua maioria às Linhas ou às Cores das Linhas que ficam associadas a essas mesmas linhas:

Tabela 4.3: Correspondência routes.txt para a Base de Dados

Campo	Correspondência na Base de Dados
route_id	Line.ext_code
agency_id	Determina Line.provider
route_short_name	Line.code
route_long_name	Line.go_name
route_color	Pode definir LineColor.back_color
route_text_color	Pode definir LineColor.fore_color

4.1.8.3 Ficheiro calendar.txt

O ficheiro representado nesta tabela contém informação relativa aos Calendários.

Tabela 4.4: Correspondência calendar.txt para a Base de Dados

Campo	Correspondência na Base de Dados
service_id	Schedule.name
start_date	Definem Schedule.season
end_date	
monday	Definem Schedule.day_type
tuesday	
wednesday	
thursday	
friday	
saturday	
sunday	

4.1.8.4 Ficheiro trips.txt

Este é o ficheiro relativo às viagens. Os seus campos são mapeados para diferentes partes da base de dados após o seu tratamento.

Tabela 4.5: Correspondência trips.txt para a Base de Dados

Campo	Correspondência na Base de Dados
trip_id	Path.est_code
service_id	Determina Passing.schedule
route_id	Determina Path.line
direction_id	Path.orientation

4.1.8.5 Ficheiro stop_times.txt

Este ficheiro tem sobretudo informação dos tempos de passagem, contudo também se consegue extrair informação relativa à composição dos percursos, nomeadamente a ordem das suas paragens.

Tabela 4.6: Correspondência stop_times.txt para a Base de Dados

Campo	Correspondência na Base de Dados	
trip_id	Determina PathStop.path	Definem Passing.path_stop
stop_id	Determina PathStop.stop	
stop_sequence	PathStop.stop_order	
shape_dist_traveled	PathStop.prev_distance	

4.2 Funcionalidades na DataGrid Silverlight 2.0

O componente DataGrid disponível no Silverlight 2.0 foi o principal objecto para se poder fazer a gestão de dados. A partir da DataGrid é possível consultar os dados existentes numa tabela, alterar esses dados e fazer com que essas alterações sejam tornadas persistentes na Base de Dados.

O facto de o Silverlight ainda estar numa versão relativamente recente fez com que houvesse um trabalho acrescido considerando que a DataGrid não suportava todas as funcionalidades que eram esperadas que ela tivesse. Sendo assim foi preciso fazer um estudo exaustivo das suas capacidades actuais para que usando essas capacidades se conseguisse tirar o melhor rendimento possível para ter um funcionamento dentro do desejado.

A DataGrid suporta DataBinding o que desde logo é uma vantagem. Assim, indo à Base de Dados buscar informação relativa a um conjunto de Dados que se pretende editar pode-se estabelecer uma ligação entre os dados que estão nas propriedades da aplicação e aqueles que são apresentados na interface com o utilizador.

Um dos grandes problemas no funcionamento da DataGrid era a falta de um *handler* que executasse uma função assim que uma linha acabasse de ser editada, pois o comportamento que se esperava da DataGrid era que uma vez uma linha fosse acabada de editar esses dados fossem persistidos à Base de Dados. Desta maneira usando os *handlers* existentes como o GotFocus, o LostFocus, o SelectionChanged e o BegginingEdit foi possível simular o comportamento de um simples *handler* que identificasse o final da edição de uma linha da tabela.

Outra funcionalidade da DataGrid que foi importante no desenvolvimento da aplicação é a possibilidade de uso de Caixas de Selecção (em casos de campos pré-definidos), ou Caixas de Verificação (no caso de campos booleanos).

Também é possível mostrar um calendário na DataGrid se o campo for uma data para ser possível consultar o calendário quando se escolhe o novo valor.

Estes campos especiais de edição/selecção estão disponíveis para aplicar como templates a uma coluna específica, sendo que a todos os campos dessa coluna será aplicado o template definido para essa coluna.

4.3 Formulários em Silverlight

Neste subcapítulo serão evidenciados alguns pormenores de implementação das interfaces, tendo em conta que o grande desafio desta parte do projecto era desenvolver as interfaces de uma forma simples.

4.3.1 Utilização do sistema

O utilizador ao tentar aceder ao sistema terá que fazer a sua autenticação com o seu nome de utilizador e a sua palavra-passe. Após uma autenticação com sucesso o utilizador tem acesso às funcionalidades que lhe estão disponíveis no sistema.

O sistema apresenta do lado esquerdo do monitor um menu de navegação que permite escolher entre as diferentes opções de utilização da aplicação. Existem quatro opções principais: Configuração, Gestão da Rede, Geração de documentos e Logout.

A opção de configuração conta com as funcionalidades de gestão de utilizadores, assim como parametrização dos detalhes acerca da empresa cliente à qual este utilizador está associado e também permite fazer o upload das imagens que o utilizador pretenderá utilizar na geração de documentos.

A opção de gestão da rede mostra ao utilizador uma série de formulários onde este pode encontrar os dados inseridos na base de dados e fazer alteração sobre os mesmos. Esta opção inclui ainda um formulário para importação de dados que permite ao utilizador fazer *upload* do ficheiro com os dados de importação nos formatos especificados.

Existe a opção de geração de documentos onde é possível fazer a parametrização de dados para essa geração. Aqui o utilizador será capaz de especificar os documentos que quer gerar e alguns outros parâmetros disponíveis na geração desses mesmos documentos.

A última opção é a de Logout, onde o utilizador pode sair do sistema para impedir que este seja utilizado por terceiros.

4.3.2 Login

A página de Login mostra um formulário onde o utilizador se poderá autenticar.

Se algum erro ocorrer na autenticação uma pequena mensagem de erro aparecerá por baixo do botão de submissão.

Nesta página é mostrada também a informação relativa à versão do sistema que se encontra a funcionar.

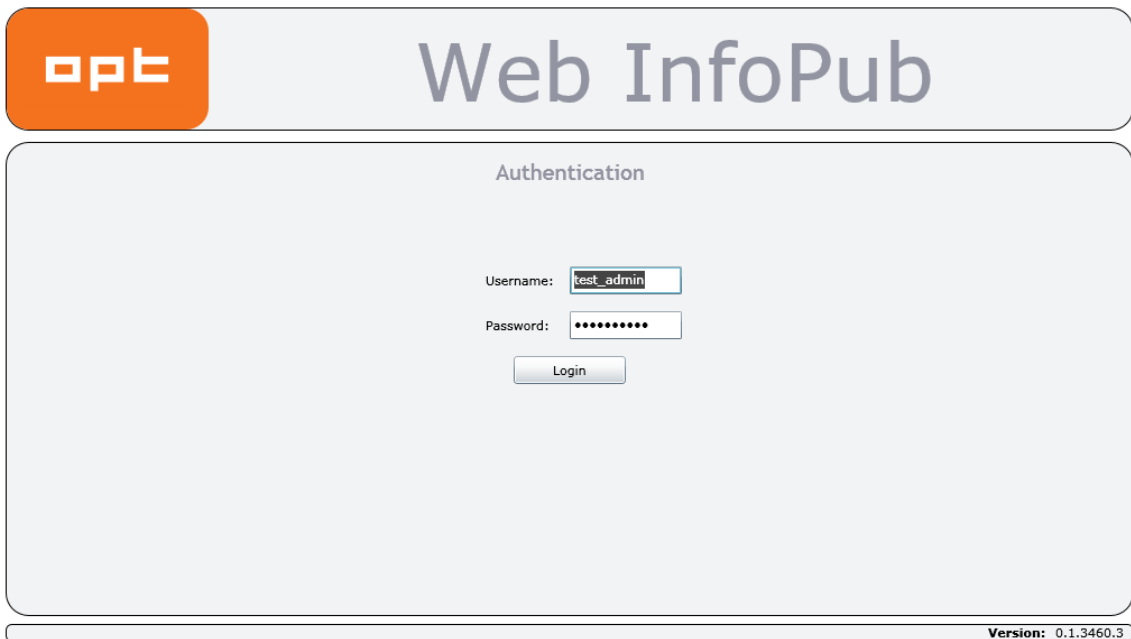


Figura 4.8: Interface Login

4.3.3 Users

Neste formulário um utilizador do sistema pode fazer a gestão de todos os utilizadores relativos ao seu provider. Poderá editar os seus nomes de utilizador, os seus nomes completos, assim como a sua password.

Tem também uma informação relativa à data do último login no sistema, informação que não é editável.

Todos os formulários depois da autenticação mostram no fundo da página informação relativa ao utilizador autenticado no sistema, o seu correspondente provider e a data da sua última entrada no sistema.

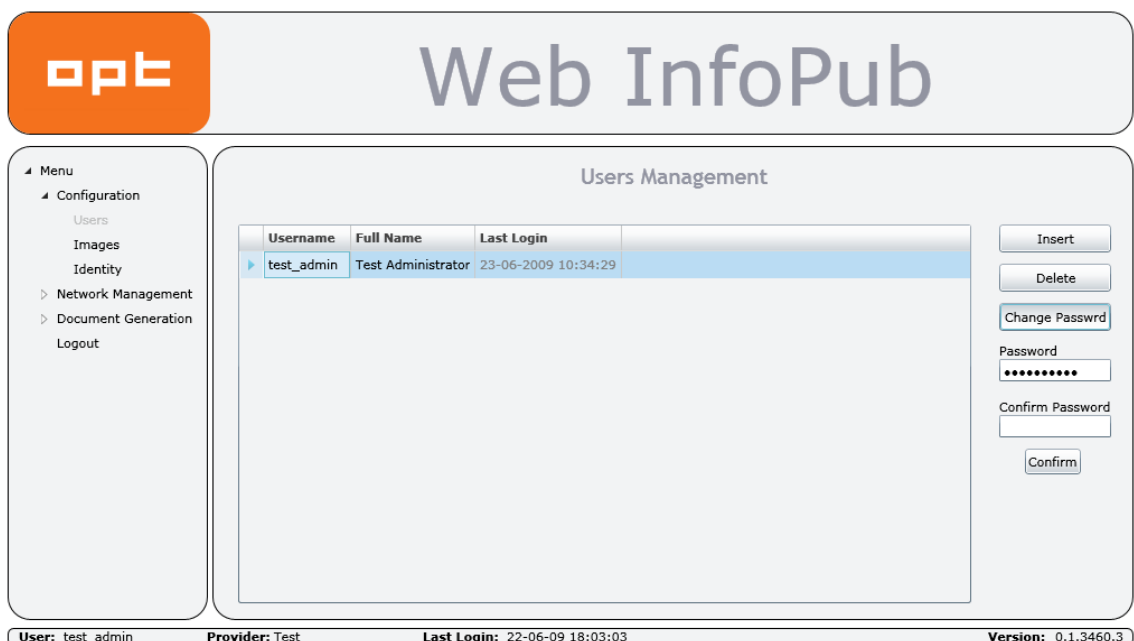


Figura 4.9: Interface Users

4.3.4 Images

O formulário que se pode ver nesta imagem permite fazer a gestão das imagens de determinado Provider. Cada provider tem um limite de upload de 50 imagens, não podendo nenhuma delas exceder um Megabyte de tamanho.

A partir deste formulário é possível fazer uma pré-visualização das imagens carregando sobre o nome das imagens aparecendo do lado direito uma miniatura das imagens usadas. Também se poderão apagar imagens assim como fazer upload de novas imagens para o sistema.

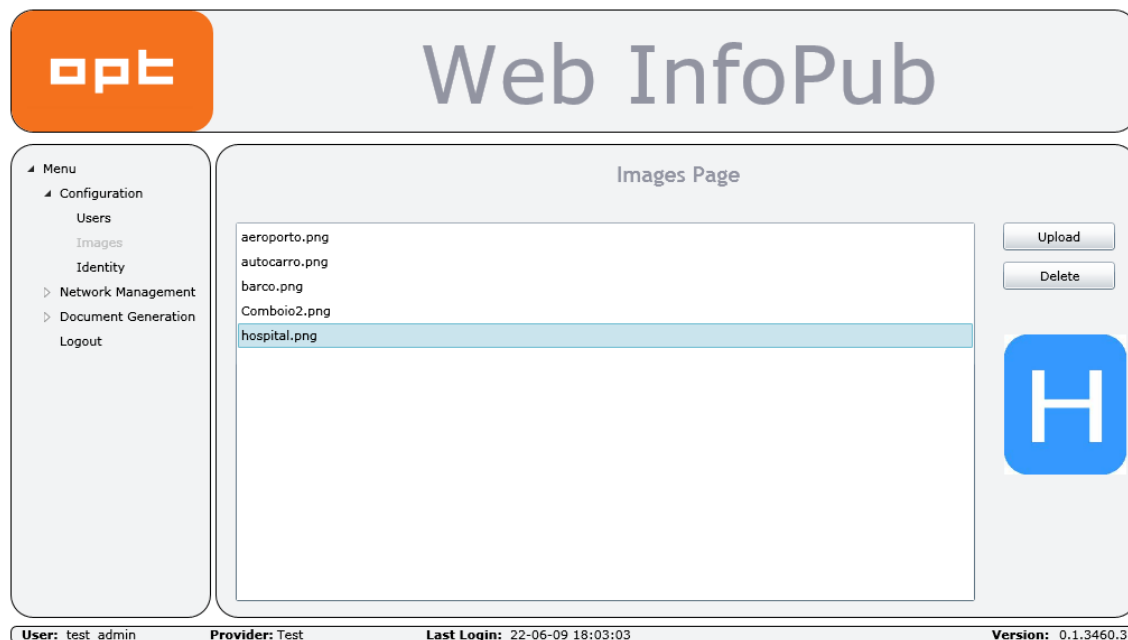


Figura 4.10: Interface Images

4.3.5 Identity

Neste formulário é possível alterar dados relativos à identidade da empresa. Estes dados aparecem nos documentos gerados, nomeadamente o nome da empresa e o seu logótipo.

Aqui é então possível alterar o nome completo da empresa, o seu nome comercial e fazer upload do seu logótipo. A gestão da imagem que serve de logótipo à empresa é feita independentemente da gestão das outras imagens desse Provider.

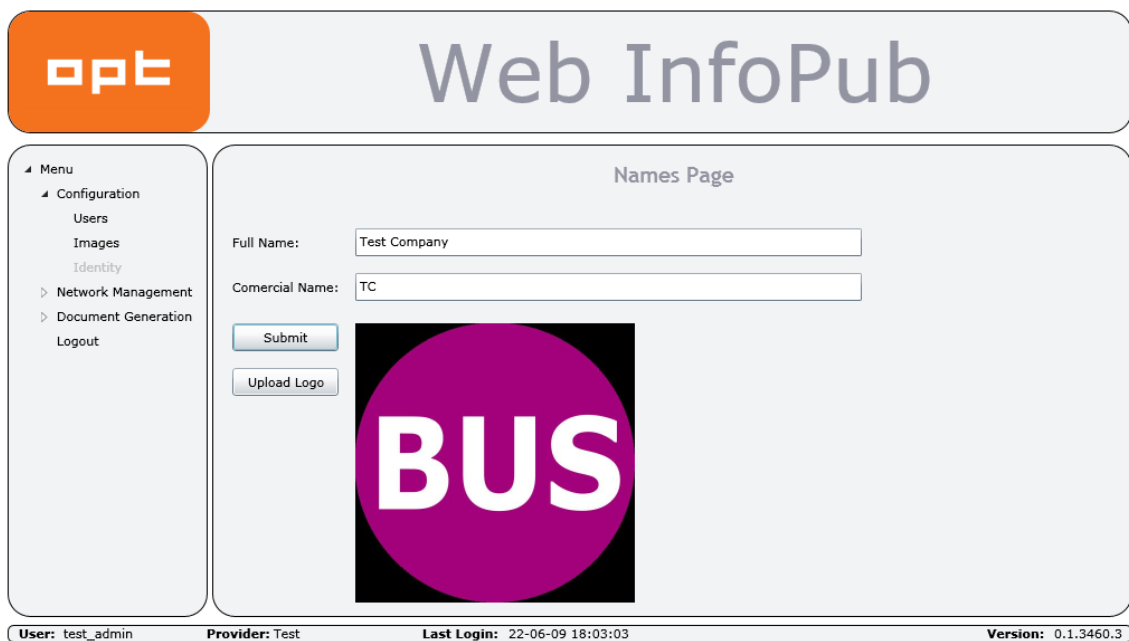


Figura 4.11: Interface Identiy

4.3.6 Interfaces

Neste formulário é possível criar, editar, ou apagar Interfaces. Os Interfaces são criados com recurso às imagens que o Provider tem na sua conta. Assim é possível criar um novo Interface, atribuindo-lhe um nome e seleccionando a imagem correspondente através de uma caixa de selecção que aparece quando se pretende seleccionar o ficheiro.

Adicionalmente pode-se ver uma pequena miniatura da imagem associada a esse Interface na linha correspondente ao mesmo.

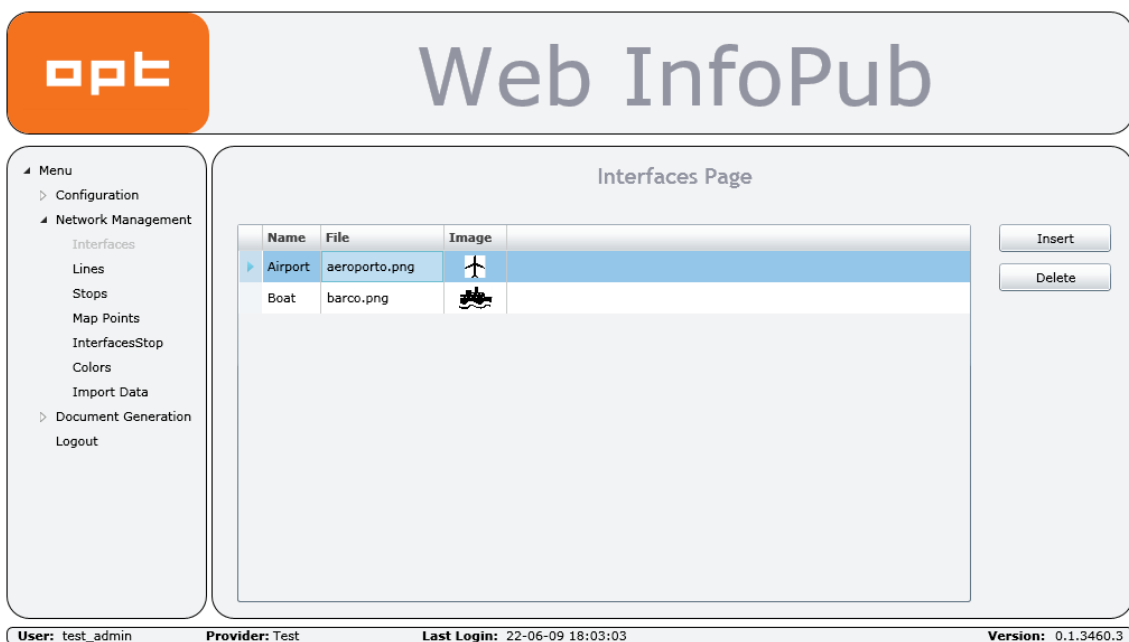


Figura 4.12: Interface Interfaces

4.3.7 Lines, Stops e MapPoints

Os formulários correspondentes às Linhas, às Paragens e aos MapPoints, são todos muito idênticos, baseando-se numa DataGrid como a que aparece na Figura 4.13, com um DataBinding puro das respectivas tabelas e podendo-se fazer a edição e inserção de novos dados assim como apagar alguns dos dados já existentes.

Code	External Code	Name	Short Name	X	Y	MapPoint	Is Active
0002	2	BROWARD CENTRAL TI		26.12	-80.14	No MapPoint	<input checked="" type="checkbox"/>
0003	3	BROWARD B/SE 3 A		26.12	-80.14	No MapPoint	<input checked="" type="checkbox"/>
0004	4	SE 3 A/SE 7 S		26.11	-80.14	No MapPoint	<input checked="" type="checkbox"/>
0005	5	SE 3 A/SE 9 S		26.11	-80.14	No MapPoint	<input checked="" type="checkbox"/>
0006	6	SE 3 A/SE 11 S		26.10	-80.14	No MapPoint	<input checked="" type="checkbox"/>
0007	7	SE 3 A/DAVIE B (S)		26.10	-80.14	No MapPoint	<input checked="" type="checkbox"/>
0008	8	SE 3 A/SE 14 S		26.10	-80.14	No MapPoint	<input checked="" type="checkbox"/>
0009	9	SE 3 A/SE 16 S BROW		26.10	-80.13	No MapPoint	<input checked="" type="checkbox"/>
0010	10	SE 17 S/SE 3 A		26.10	-80.13	No MapPoint	<input checked="" type="checkbox"/>
0011	11	SE 17 S/US 1 (W)		26.10	-80.13	No MapPoint	<input checked="" type="checkbox"/>
0012	12	US1/SE 21 S		26.09	-80.13	No MapPoint	<input checked="" type="checkbox"/>
0013	13	US1/SE 26 S		26.09	-80.13	No MapPoint	<input checked="" type="checkbox"/>

Figura 4.13: Interface Stops

4.3.8 InterfacesStop

Este formulário é um pouco mais complexo do que os restantes visto ser constituído por três DataGrids cada uma com dados correspondentes a uma tabela diferente. O objectivo é associar as Interfaces às respectivas paragens ou, alternativamente, remover algumas dessas ligações.

Este formulário revelou-se a certa altura algo complexo devido à quantidade de dados que era necessário ser transmitida entre cliente e servidor, tendo em conta a multiplicidade de ligações que podiam ser efectuadas simultaneamente. Após alguns melhoramentos no algoritmo de inserção na Base de Dados, este formulário passou a ser executado com uma performance bastante melhor sendo que o tempo de execução de algumas alterações é praticamente apenas o tempo que leva a fazer a inserção dessas associações na tabela respectiva da Base de Dados.

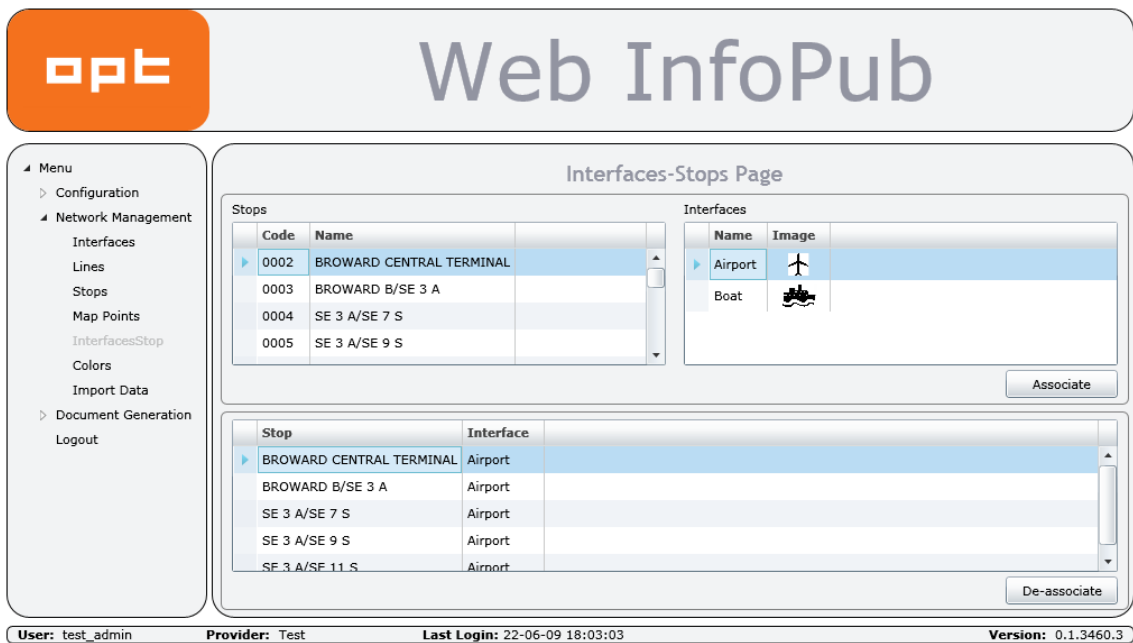


Figura 4.14: Interface InterfacesStop

4.3.9 Colors

Este formulário permite fazer uma gestão das Cores de um Provider específico.

Ao fazer a inserção de uma nova Cor é possível definir o seu nome, sendo que as restantes três colunas da DataGrid são apenas de consulta. Para alterar as cores de uma Cor deve-se seleccionar a Cor a alterar e usando o menu de selecção de cores do lado direito, escolher a nova cor pretendida e usar os botões em baixo para a definir como cor de fundo ou cor das letras. Assim é possível em tempo real ver o efeito das alterações aplicadas na célula de exemplo e ao mesmo tempo usar os códigos hexadecimais nas células respectivas para consulta e possível utilização para outros fins.



Figura 4.15: Interface Colors

4.3.10 Data Import

Este formulário é correspondente à importação de dados. Primeiro deve-se seleccionar o formato dos dados de entrada, visto o sistema permitir a importação nos formatos Google Transit ou TransXChange.

Seguidamente pode passar-se à escolha do ficheiro de importação que deverá ser um ficheiro XML no caso do formato TransXChange ou um ficheiro ZIP no caso do formato Google Transit.

Depois desses passos efectuados e se o estado de importação apresentado for “Ready to Import” a importação poderá ser feita. Se a mensagem for uma diferente significa que ou se encontra uma importação em curso, sendo que enquanto essa não acabar não será possível fazer outra, ou então que algo não correu dentro da normalidade durante a última importação, sendo que se deve contactar a Administração para se poder passar à resolução do problema.

The screenshot displays the 'Web InfoPub' interface. At the top left is the 'opt' logo. The main header reads 'Web InfoPub'. A left-hand navigation menu lists: Menu, Configuration, Network Management (expanded), Interfaces, Lines, Stops, Map Points, InterfacesStop, Colors, Import Data (highlighted), Document Generation, and Logout. The main content area is titled 'Import Data' and contains the following elements: 'Data Import Type' with radio buttons for 'Google Transit' (selected) and 'TransXChange'; 'Source File' with a 'Choose File' button and the text 'No File Selected...'; 'Import File' with an 'Import' button; and 'Import Status: Ready to Import'. The footer bar includes: 'User: test_admin', 'Provider: Test', 'Last Login: 22-06-09 18:03:03', and 'Version: 0.1.3460.3'.

Figura 4.16: Interface Import Data

4.3.11 Document Generation

Este último formulário é o que permite a geração da documentação. O utilizador escolhe a Linha referente à qual pretende que seja gerada a informação e as paragens para as quais essa informação deve ser gerada. O utilizador pode então escolher uma data de referência, que indicará a data de entrada em vigor dessa informação e o tipo de folheto informativo que pretende gerar.

Neste caso pode optar entre três opções, sendo as duas primeiras os dois geradores produzidos e numa terceira opção informação dos dois geradores combinada num mesmo documento.

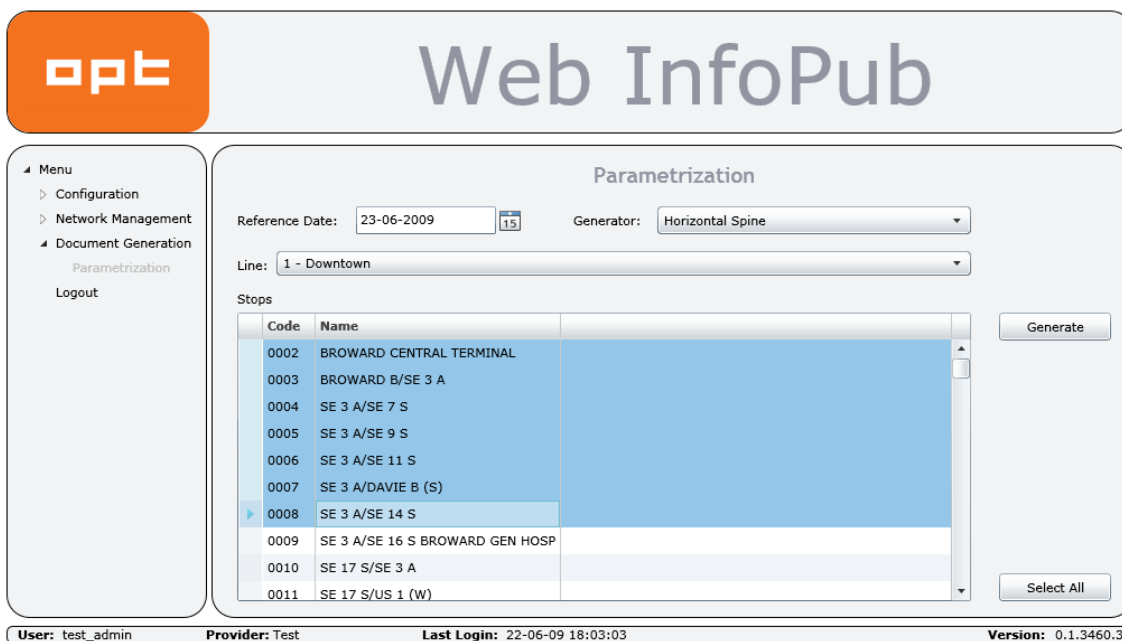


Figura 4.17: Interface Document Generation (Parametrization)

4.4 Gerador de Informação

O Gerador de Informação é a última parte da aplicação desenvolvida. Este Gerador pretende que haja um documento com informações relativas à Rede. Inicialmente foi produzido um documento geral, com cabeçalho e rodapé que pudesse ser usado como base genérica para o resto do documento, sendo posteriormente produzidos os outros geradores.

Foram usados projectos desenvolvidos anteriormente na empresa onde o projecto foi realizado, nomeadamente o projecto AbstractGraphs que permite construir documentos genéricos para depois se usar um outro projecto que permita exportar esses documentos para um qualquer formato. Neste caso foi usado um outro projecto, chamado SvgNet que permite converter esse objecto do tipo AGDocument (Abstract Graphs Document) para um ficheiro no formato SVG.

O formato SVG foi escolhido para este tipo de documentos devido a ser um formato vectorial que torna possível a impressão em qualquer tamanho sem haver perda de qualidade das imagens incluídas.

4.4.1 Documento

O documento base é produzido para que possa incorporar o resultado dos outros geradores.

No cabeçalho desse documento contém informação acerca da linha, como o seu código e o nome identificativo dessa linha. Ainda se encontra representado um pequeno código relativo à paragem à qual o documento se destina a ser colocado e eventuais Interfaces que essa linha possa ter.



Figura 4.18: Exemplo de Cabeçalho

No rodapé é mostrado o logótipo do Provider que gerou o documento, assim como o nome completo desse Provider. Em ponto pequeno é também exibida a data em que a informação contida entra em vigor.

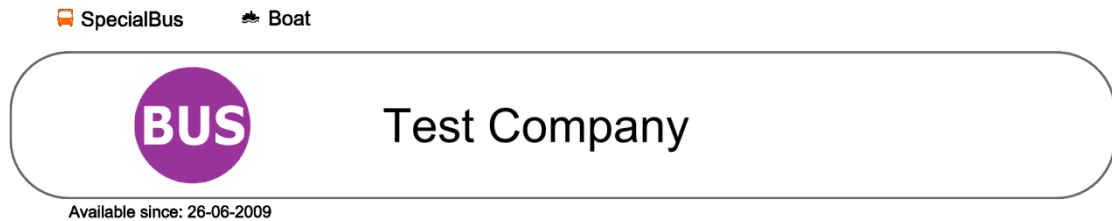


Figura 4.19: Exemplo do Rodapé

Adicionalmente logo acima do rodapé é exibida a legenda das Interfaces que são exibidas na informação mostrada pelos geradores. Essa legenda é feita tendo em conta as imagens que aparecem no documento, portanto é necessário saber à partida os geradores que irão ser usados e as imagens que irão aparecer no resultado final. Este algoritmo é também usado para saber as imagens que devem ser incluídas no ficheiro com todos os conteúdos gerados e que é devolvido ao utilizador no final da geração.

4.4.2 Espinha Horizontal

A Espinha Horizontal pretende identificar todas as paragens referentes a uma linha, dando destaque à paragem actual. Esta espinha é representada com as cores da linha a que se refere, ou a preto e branco no caso de essa Linha não ter nenhuma cor associada. Ela mostra também informação relativa às zonas dessa Linha e tempo identificativo do tempo que leva a percorrer determinadas partes do percurso.

Todas as paragens dessa Linha que tenham interfaces aparecem com as imagens das respectivas interfaces representadas.

As paragens de uma Linha podem ser em número demasiado grande para caber na representação de uma espinha e se isto acontecer é necessário recorrer aos designados de Percursos Abreviados. Estes percursos são partes da Linha que não se vão encontrar representadas na espinha.

Foi implementado um algoritmo que decide as partes do percurso que são abreviadas e as partes que devem ser representadas na espinha. Esse algoritmo obedece às seguintes regras:

- A primeira e a última paragem devem ser sempre representadas.
- Tanto a paragem actual como a anterior devem ser sempre representadas.
- Logo a seguir a estas têm mais prioridade as paragens imediatamente seguintes à paragem actual.

- Se todas as paragens posteriores conseguirem ser representadas as que têm mais prioridade são as anteriores à paragem actual que se encontrarem mais próximo desta.

Isto pode levar a uma espinha que tenha até dois percursos abreviados e que a existir se localizarão logo a seguir à paragem inicial, imediatamente antes da paragem final ou então nos dois locais.

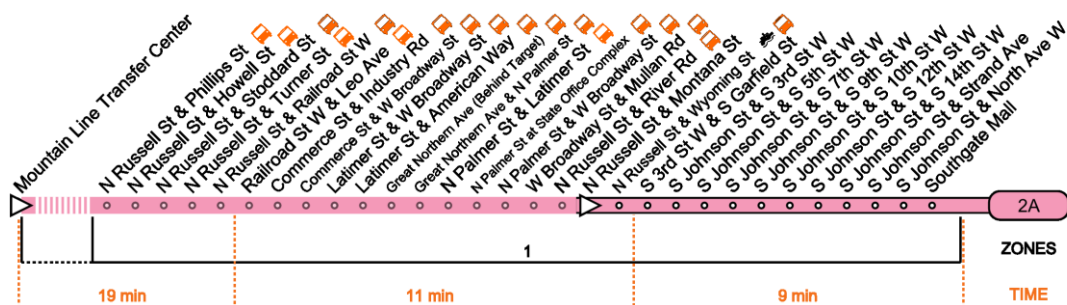


Figura 4.20: Exemplo de Espinha Horizontal

A representação das paragens, como ilustrado na Figura 4.20, tem uma inclinação de 45° e se a combinação do nome com as Interfaces que a compõem excederem o espaço disponível para a representação das mesmas, o tamanho das letras é ajustado de maneira a ser todo ele visível e ao mesmo tempo ocupar o máximo de espaço possível.

4.4.3 Espinha Vertical

A Espinha Vertical pretende ser uma alternativa à espinha horizontal em termos de nível de detalhe. Esta espinha ao contrário da apresentada anteriormente não mostra informação relativa a todas as paragens e também não dá especial atenção à paragem actual.

A informação aqui representada é referente aos principais pontos por onde determinada linha passa, ou seja, os Pontos de Mapa.

Esta espinha à semelhança da anterior mostra as durações, neste caso entre Pontos de Mapa, mostra também as interfaces presentes em cada Ponto de Mapa e também informa sobre as Zonas que um passageiro pode ter de atravessar quando viajar entre dois Pontos de Mapa identificados na espinha.

À semelhança da espinha anterior esta é desenhada tendo em conta as cores da espinha em questão ou a preto e branco no caso de não ter cor associada.

A Espinha Vertical quando usada em conjunto com a Horizontal é representada ligeiramente mais em baixo nos documentos, tendo em conta o espaço usado pela outra espinha.

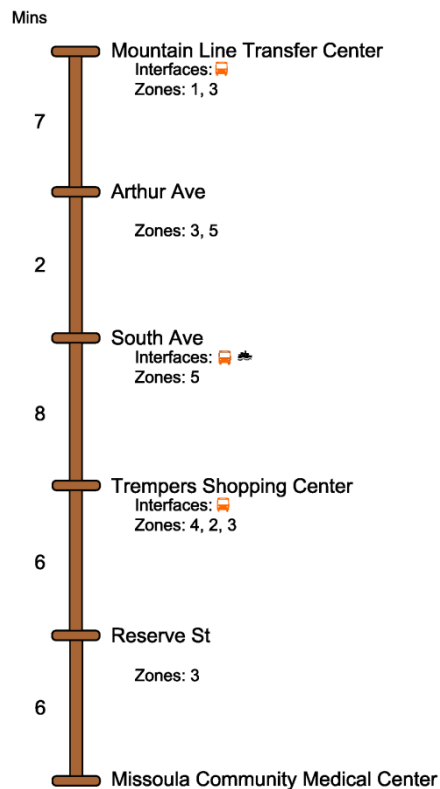


Figura 4.21: Exemplo de Espinha Vertical

4.5 Testes Efectuados

Os testes desta aplicação foram sempre sendo efectuados ao longo do seu desenvolvimento. Contudo, existiram alguns testes que apenas foram possíveis fazer na parte final de desenvolvimento do projecto. Estes testes destinavam-se a perceber o bom funcionamento da aplicação com grandes quantidades de dados.

Neste momento a limitação dos testes era definida pelos dados que estivessem disponíveis. Assim foi feita a importação do maior conjunto de dados disponível num dos dois formatos (neste caso GTFS).

Estes dados correspondem a toda a definição da Rede da Washington Metropolitan Area Transit Authority, mostrando todos os dados definidos por esta instituição no formato GTFS.

Alguns dos números mais importantes dos dados disponibilizados nesta especificação são os seguintes:

- 19 Tipos de Calendários diferentes
- 332 Linhas
- 10763 Paragens
- 46460 Viagens
- 1741108 Tempos de Passagem

Estes dados foram importados para a Base de Dados e convertidos. Não tendo a eficácia desejada, foram feitos ajustes à aplicação para que esta tornasse a sua importação mais célere.

Também foram feitos ajustes a nível de interface, onde alguns formulários que carregavam DataGrids referentes às Paragens, transpareciam alguma falta de performance devido à inúmera quantidade de dados transferidos entre a Base de Dados e a aplicação a correr no cliente. Também este problema de performance foi resolvido, tendo a aplicação ficado com um desempenho aceitável e passando neste que foi o grande teste de carga da mesma.

4.6 Resumo e Conclusões

O desenvolvimento passou sobretudo pelos três módulos anteriormente mencionados, o Importador de Dados, o Gestor de Dados e o Gerador de Documentos.

O Importador de dados acabou por suportar mais do que um formato de entrada, o que supera as expectativas iniciais, embora graças a uma aplicação externa, mas que não deixa de ter trabalho envolvido, nomeadamente de pesquisa e de adaptação entre os dois formatos. O resto do trabalho foi um grau relativamente alto de complexidade devido às diferenças de especificação da Rede no formato GTFS e o facto de isso se tornar relativamente pesado no processamento dos dados de entrada.

Foi necessário proceder a várias tentativas de aperfeiçoamento da solução para que ficasse suficientemente rápida e ao mesmo tempo eficaz, com as funcionalidades que eram previstas como a actualização de alguns dados no caso de eles já se encontrarem inseridos na Base de Dados.

O grande desafio na implementação do Gestor de Dados prendeu-se com o facto de ser preciso fazer com que a interface correspondesse aquilo que era pretendido do seu funcionamento. Aqui foi preciso tirar o máximo proveito das funcionalidades que o Silverlight oferece, tentar apresentar os dados da forma mais intuitiva possível e analisar os possíveis erros para que pudessem ser tratados da melhor maneira e de modo a dar sempre a maior informação possível ao utilizador.

O Gerador de Dados passou por um estudo das tecnologias que existiam na empresa, e a maneira como a informação como a informação era gerada, de forma a se poder usar o máximo de código possível e produzir algo que tivesse uma interacção visual agradável e servisse como um bom protótipo de informação. Foram conseguidos fazer dois geradores diferentes sendo que um deles, o que gera a Espinha Vertical, ficou com uma estrutura bastante diferente do que havia sido feito até este momento permitindo ver as várias potencialidades das ferramentas utilizadas.

5 Conclusões

Neste capítulo serão abordadas questões de apreciação global do projecto, tendo em conta o trabalho planeado e o que foi atingido, tendo em conta aquilo que poderia ter sido feito de maneira diferente. De igual modo se irão abordar questões relacionadas com o trabalho futuro do projecto, que inclui melhoramentos inicialmente não pensados ou então funcionalidades que o tempo limitado de duração deste projecto não permitiu atingir.

5.1 Satisfação dos Objectivos

Tendo em conta os objectivos a que este projecto se propunha é com satisfação que é possível olhar para o resultado atingido nesta altura. A meta passava por atingir um protótipo funcional capaz de mostrar as potencialidades que uma aplicação Desktop poderia ter na Internet.

Começando na investigação que foi feita a nível de tecnologias, ferramentas e igualmente o trabalho desenvolvido na área, foi possível fazer uma análise madura do problema e decidir qual o melhor caminho a seguir em termos do desenvolvimento do projecto. A partir daqui foi possível fazer um planeamento com fases bem determinadas permitindo que todos os objectivos fossem conseguidos.

Este planeamento que foi efectuado foi importante devido ao facto de este projecto consistir em três fases tão distintas a nível de implementação. Este facto requereu que se definissem metas para cada uma destas fases para que em todas elas se pudessem atingir os objectivos mínimos sem que isso pudesse interferir com as outras fases.

O que se verificou durante o desenvolvimento do projecto foi que em algumas das fases sobrou algum tempo em relação ao inicialmente proposto e isto foi uma mais-valia no sentido de se poderem produzir algumas soluções que inicialmente não estavam previstas superando algumas expectativas ao nível do número de funcionalidades.

A escolha combinada das tecnologias Silverlight com SQL Server revelou-se uma boa escolha devido à imensa facilidade que se verificou trabalhar com ambas as ferramentas, usando em conjunto o WCF e o Linq to SQL para a comunicação Cliente-Servidor-Base de Dados. As vantagens foram certamente muitas quando comparadas

com as desvantagens. A passagem do modelo de definição de Rede existente no InfoPub Desktop foi relativamente simples, e a passagem dos tipos de dados de Oracle para SQL Server também foi algo relativamente simples e conseguido sem esforço suplementar.

Pessoalmente uma das grandes desvantagens de ter escolhido as tecnologias em questão foi o facto de o Silverlight ainda ser uma ferramenta em crescente evolução, o que faz com que algumas das suas características e funcionalidades estejam em grande mudança. Tendo como exemplo algumas funcionalidades da DataGrid, que fizeram com que tivesse de ser feito algum desenvolvimento ao que se encontra feito e, no momento de finalização do projecto, com a divulgação da nova versão beta do Silverlight (3.0), esse trabalho não seria de todo necessário.

O trabalho mostrou igualmente que é perfeitamente possível uma ferramenta deste tipo funcionar como SaaS, fazendo com que haja mais facilidade em vender este tipo de solução a um leque de clientes que normalmente não teriam ao seu alcance a oportunidade de adquirir o serviço na sua versão Desktop. Ao mesmo tempo permite oferecer um tempo de experiência para os utilizadores poderem avaliar a qualidade da ferramenta e o que realmente podem atingir com ela.

5.2 Trabalho Futuro

Devido ao resultado deste projecto ser um protótipo existe muito trabalho que pode ser efectuado posteriormente. Alguns desses pontos são contemplados logo no início do projecto, outros são melhoramentos que com o decorrer do projecto foram surgindo como boas ideias para melhoramento dos mesmos.

Algo que certamente tornará este projecto melhor será uma melhor qualidade na importação de dados, ou seja, interpretar dados que podem estar nos ficheiros de entrada e que neste momento a sua análise ainda não é feita pelo módulo de importação, nem suportada pelo modelo de definição da rede. Um exemplo de informação útil que por exemplo o formato GTFS tem e que seria interessante poder-se consultar nos documentos gerados são os tarifários praticados pela operadora de transportes.

É também importante haver uma gestão de completa da compatibilidade entre os formatos de entrada e a própria Base de Dados do InfoPub na Web. Essa compatibilidade tem de ir sendo gerida ao longo do tempo, com as constantes actualizações dos formatos. Seria também interessante contribuir para o desenvolvimento da ferramenta open-source que permite a transformação de dados do formato TransXChange para o formato GTFS.

No fim deste projecto existe um número limitado de formulários que foram produzidos, o que significa que ainda existem muitos mais formulários de gestão de dados que podem ser produzidos de maneira a haver um completo controlo sobre os dados de cada Provider sem que tenham de estar a fazer novas importações para o sistema sempre que se tenham alterações a fazer. Talvez um dos maiores desafios será fazer a gestão das horas de passagem num formulário na Web devido à enorme quantidade de dados que seria preciso gerir.

Sendo os Geradores uma parte importante do projecto também é importante que fossem criados mais geradores e que eventualmente estes pudessem ser combinados entre si. Um outro desafio seria separar completamente os Geradores em bibliotecas, de maneira a que um utilizador ao adquirir um serviço pudesse fazer uma compra de

acordo com os Geradores que pretendem utilizar. Alguma da informação que falta incluir em novos geradores diz respeito aos horários, ou a mapas esquemáticos orientados às paragens (à semelhança do projecto BusMap, parte do InfoPub versão Desktop).

Neste momento o poder sobre o que se pode disponibilizar a um cliente ou não é bastante reduzido: apenas é possível determinar se o cliente tem acesso ao sistema e por quanto tempo ele tem permissão para aceder ao mesmo. Seria importante poder diferenciar tipos de acesso ao sistema, consoante os serviços que cada cliente tivesse adquirido. Isto faria bastante sentido numa lógica de disponibilização de serviço “trial”, ou então com os diferentes tipos de geradores como referido anteriormente. A isto pode-se juntar a questão da segurança, pois é necessário garantir ao cliente que os seus dados estão em segurança. Isto já está a ser garantido pela aplicação mas será necessário fornecer ao utilizador garantias em como isto acontece.

Este último ponto conduz também a uma outra questão que se prende com a disponibilidade dos dados para o cliente e o seu poder sobre os mesmos. O sistema permite que o cliente tenha os seus dados em seu poder e depois faça a sua importação, mas existem dados que são introduzidos na base de dados e que não podem ser especificados nos ficheiros de importação. Isto leva à conclusão que poderia ser útil criar uma ferramenta de exportação de todos os dados do cliente, ou alternativamente, dos dados que o cliente pode editar na interface da aplicação mas que não tem a possibilidade de especificar nos ficheiros de importação.

Como último ponto é importante que seja feita uma manutenção do sistema, o que também pode ser contado como trabalho futuro. Não só ao nível de correcção de anomalias que o sistema apresente mas também ao nível da introdução das mais recentes tecnologias de forma a manter o sistema actual e apelativo para os utilizadores. Um bom exemplo de actualização de tecnologias é por exemplo a actualização para Silverlight versão 3.0 que se encontra neste momento em versão beta e que contempla mudanças que trariam melhorias substanciais para a aplicação.

6 Bibliografia

- [OPT05]. **OPT.** *OPT.* [Online] '05. <http://www.opt.pt>.
- [Mah09]. **Mahowald, Robert.** IDC - Press Release. [Online] 26 Janeiro '09. <http://www.idc.com/getdoc.jsp?containerId=prUS21641409>.
- [GoC09]. **Google Code.** FeedSpecHistory. [Online] 27 Maio '09. <http://code.google.com/p/googletransitdatafeed/wiki/FeedSpecHistory>.
- [Goo09]. **Google.** Google Transit Feed Specification. [Online] 30 Março '09. http://code.google.com/transit/spec/transit_feed_specification.html.
- [GCo09]. **Google Code.** timetablepublisher. [Online] 10 Fevereiro '09. <http://code.google.com/p/timetablepublisher/>.
- [TRB09]. **Transportation Research Board of the National Academies.** Google Transit Data Tool for Small Agencies. *TRB.* [Online] '09. <http://www.trb.org/TRBNet/ProjectDisplay.asp?ProjectID=2695>.
- [TrR09]. **Transportation Research Board for National Academies.** Transportation Research Board. [Online] '09. <http://www.trb.org/>.
- [RMT08]. **RouteMe² Technologies, Inc.** PedNav. *RouteMe² Technologies.* [Online] 12 Dezembro '08. <http://www.routeme2.com/products/pednav/>.
- [Eva08]. **Evans, Martin.** Martin Evans' Web Portfolio. *Google Transit Feed Project.* [Online] '09. <http://www.martin-evans.com/portfolio/view/30#nogo>.
- [GGE08]. **Greater Greater Washington.** Metro refuses to participate in Google Transit. *Greater Greater Washington.* [Online] 13 Dezembro '08.
- [GGE00]. **Greater Greater Washington.** Washingtonians want Google Transit! *Greater Greater Washington.* [Online] <http://greatergreaterwashington.org/campaigns/googletransit/>.
- [Peg09] **Pegoraro, Rob.** Upgrading Transit's Interface: Metro Releases Google Transit Data. *Faster Forward.* [Online] 23 Março '09. http://voices.washingtonpost.com/fasterforward/2009/03/upgrading_transits_interface_m.html.
- [Cam08]. **Camino, Fred.** We Want Our Google Transit! *MetroRiderLA.* [Online] 14 Abril '08. <http://metroriderla.com/2008/04/14/we-want-our-google-transit/>.

[Hom09]. Home. *TransXChange*. [Online] 02 Junho '09.
<http://www.transxchange.org.uk/>.

[Omn08]. **OmniBus**. Home. *OmniBus*. [Online] '08. <http://www.omnibus.uk.com/>.

[Mic08]. **Microsoft**. Light Up the Web. *Microsoft Silverlight*. [Online] '08.
<http://www.microsoft.com/SILVERLIGHT/overview/default.aspx>.

[Mic09]. **Microsoft**. XAML Overview. [Online] '09.
<http://msdn.microsoft.com/en-us/library/ms752059.aspx>.

[FWT00]. **FWT**. FWT - providing maps, guides, timetables and stop information to the public transport industry. *FWT*. [Online]

Anexo A – Tabelas da Base de Dados

Provider

Cada elemento da tabela provider representa um cliente do InfoPub na Web. Um cliente (Provider) tem então uma data limite de acesso ao sistema. Esta tabela proporciona fazer esta gestão assim como guardar alguma outra informação relevante para os Providers.

Tabela Provider na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
name	VARCHAR(255)	Nome identificativo do Provider
full_name	VARCHAR(255)	Nome completo do Provider
comercial_name	VARCHAR(255)	Nome comercial do Provider
import_state	VARCHAR(1024)	Descrição do estado de importação
expiry_date	DATETIME	Data de expiração do fornecimento do serviço ao Provider
version	DATETIME	Campo verificação de problemas de concorrência
request	NUMERIC(5)	Contagem de pedidos de geração de documentos

User

Um User é um utilizador do sistema. Este utilizador está inevitavelmente ligado a um provider, visto que o provider é a verdadeira entidade cliente. Um utilizador no sistema tem o poder de gerir todos os utilizadores do seu Provider. Não existe diferenciação entre utilizadores de um mesmo Provider.

Tabela User na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
<i>username</i>	VARCHAR(255)	Nome do User
password	VARCHAR(255)	Password de acesso ao sistema
fullname	VARCHAR(255)	Nome completo do User
version	DATETIME	Campo verificação de problemas de concorrência
provider	NUMERIC(19)	Referência ao Provider (FK)
last_login	DATETIME	Data do último login do User no sistema

Lines

As Lines representam uma linha de transporte público. Uma linha está normalmente associada a um transporte que circula entre dois pontos, por isso existe um campo para o nome de ida e outro para o nome de volta. Estes nomes normalmente espelham o ponto de chegada da viagem actual ou então é feita uma descrição “Partida – Destino”. As linhas têm também uma cor associada e podem ter uma interface derivado de alguma característica especial dessa linha.

Tabela Lines na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
<i>code(1)</i>	VARCHAR(255)	Código da Line
<i>ext_code</i>	VARCHAR(255)	Código externo da Line
<i>go_name</i>	VARCHAR(255)	Nome de ida
<i>return_name</i>	VARCHAR(255)	Nome de volta
<i>line_color</i>	NUMERIC(19)	Referência à LineColor (FK)
<i>certified</i>	NUMERIC(5)	Certificação ou não da Line
<i>interface</i>	NUMERIC(19)	Referência à Interface (FK)
<i>provider(1)</i>	NUMERIC(19)	Referência ao Provider (FK)
<i>version</i>	DATETIME	Campo verificação de problemas de concorrência

LineColor

A LineColor representa a cor da linha. Esta cor é caracterizada pela cor de fundo e pela cor das letras que compõem elementos dessa cor.

Tabela LineColor na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
<i>name(1)</i>	VARCHAR(255)	Nome da LineColor
<i>fore_color</i>	NUMERIC(19)	Cor das letras da LineColor
<i>back_color</i>	NUMERIC(19)	Cor de fundo da LineColor
<i>is_dark</i>	NUMERIC(5)	Verificação se a LineColor é escura
<i>version</i>	DATETIME	Campo verificação de problemas de concorrência
<i>provider(1)</i>	NUMERIC(19)	Referência ao Provider (FK)

LineColorShade

Um elemento LineColorShade faz parte de uma LineColor e pretende mostrar pequenas variações de uma LineColor quando, num esquema, existem duas linhas com a mesma LineColor.

Tabela LineColorShade na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
seq	NUMERIC(5)	Número de sequência da LineColorShade
val	FLOAT	Cor da LineColorShade

ZoneType

ZoneType define tipos de Zona. Dependendo do Provider e do tipo de serviço este pode ter diferentes tipos de zonamento consoante o serviço a utilizar. Isto permite definir mais do que um tipo de zonas para cada Provider.

Tabela ZoneType na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
name	VARCHAR(255)	Nome da ZoneType
version	DATETIME	Campo verificação de problemas de concorrência
provider	NUMERIC(19)	Referência ao Provider (FK)

Zone

Zone define uma zona específica. Esta zona tem obrigatoriamente de pertencer a um ZoneType. Mesmo que o Provider apenas tenha um tipo de zona, terá de ser criado um tipo de zona por defeito para que todas as zonas possam ser criadas.

Tabela Zone na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
name	VARCHAR(255)	Nome da Zone
zone_type	NUMERIC(19)	Referência à ZoneType (FK)
version	DATETIME	Campo verificação de problemas de concorrência

Schedule

Um Schedule é horário é aplicado aos tempos de passagem das viaturas nas paragens. Este horário define os dias da semana e a época do ano em que essas horas de passagem vigoram.

Tabela Schedule na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
name	VARCHAR(255)	Nome do Schedule
day_type	VARCHAR(255)	Tipo de dias afectados pelo Schedule
season	VARCHAR(255)	Época do ano afectada pelo Schedule
version	DATETIME	Campo verificação de problemas de concorrência
provider	NUMERIC(19)	Referência ao Provider (FK)

Stop

Stop representa uma paragem. Todas as paragens têm um código e podem ter um código externo para efeitos de importação de dados. Uma paragem pode também existir mas estar inactiva. As paragens têm um nome e alternativamente um nome mais curto. Também é armazenada a localização geográfica da paragem no sistema WGS84. Uma paragem está associada a um Provider e pode também estar associada a um MapPoint.

Tabela Stop na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
<i>code(1)</i>	VARCHAR(255)	Código da Stop
ext_code	VARCHAR(255)	Código externo da Stop
is_active	NUMERIC(5)	Verificação se a Stop se encontra activa
name	VARCHAR(255)	Nome da Stop
short_name	VARCHAR(255)	Nome curto da Stop
x	FLOAT	Posição geográfica 'x' da Stop
y	FLOAT	Posição geográfica 'y' da Stop
map_point	NUMERIC(19)	Referência ao MapPoint (FK)
version	DATETIME	Campo verificação de problemas de concorrência
<i>provider(1)</i>	NUMERIC(19)	Referência ao Provider (FK)

PointOfInterest

Um PointOfInterest é um ponto da área geográfica da rede que pode ser relevante ser apresentado numa representação com referências geográficas. Este ponto pode ter uma interface e está também associado a um Provider.

Tabela PointOfInterest na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
name	VARCHAR(255)	Nome do PointOfInterest
x	FLOAT	Posição geográfica 'x' do PointOfInterest
y	FLOAT	Posição geográfica 'y' do PointOfInterest
interface	NUMERIC(19)	Referência à Interface (FK)
version	DATETIME	Campo verificação de problemas de concorrência
provider	NUMERIC(19)	Referência ao Provider (FK)

Interface

Uma Interface pode ter várias utilidades no sistema InfoPub na Web. Primariamente uma Interface pode representar um conjunto de serviços que estão

disponíveis numa determinada paragem. Contudo esta definição foi aumentando e o Interface tornou-se em algo que pode ser associado a outras coisas. Ele representa um nome e tem uma imagem associada, sendo que o caminho dessa imagem no sistema de ficheiros é o que fica guardado na Base de Dados e não a imagem em si. As Interfaces estão também elas ligadas a um Provider.

Tabela Interface na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
file_path	VARCHAR(255)	Nome relativo do caminho do ficheiro que ilustra a Interface
<i>name(1)</i>	VARCHAR(255)	Nome da Interface
version	DATETIME	Campo verificação de problemas de concorrência
<i>provider(1)</i>	NUMERIC(19)	Referência ao Provider (FK)

InterfacesStop

InterfacesStop é a tabela que representa a ligação entre as paragens e as interfaces. Esta é uma ligação “de muitos para muitos” e portanto é necessária esta tabela para que essa ligação seja possível.

Tabela InterfacesStop na Base de Dados

Campo	Tipo	Descrição
stop	NUMERIC(19)	Referência à Stop (FK)
interface	NUMERIC(19)	Referência à Interface (FK)

MapPoint

Um MapPoint é um ponto na rede do qual fazem parte uma ou várias paragens que naturalmente estarão próximas umas das outras. A este Ponto é associado um nome e ele está ligado a um Provider. A sua localização geográfica também fica armazenada na Base de Dados.

Tabela MapPoint na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
<i>name(1)</i>	VARCHAR(255)	Nome do MapPoint
x	FLOAT	Posição geográfica 'x' do MapPoint
y	FLOAT	Posição geográfica 'y' do MapPoint
version	DATETIME	Campo verificação de problemas de concorrência
<i>provider(1)</i>	NUMERIC(19)	Referência ao Provider (FK)

InterfacesMapPoint

À semelhança de outras tabelas esta é uma tabela que realiza uma ligação “de muitos para muitos” e visto que um MapPoint pode ter várias Interfaces associadas e vice-versa surge a necessidade de esta tabela existir na Base de Dados.

Tabela InterfacesMapPoint na Base de Dados

Campo	Tipo	Descrição
map_point	NUMERIC(19)	Referência ao MapPoint (FK)
interface	NUMERIC(19)	Referência à Interface (FK)

Path

Path representa um percurso de uma linha. Embora uma linha se represente normalmente por uma ligação entre dois pontos, esta pode ter percursos diferentes, mais curtos ou mais longos, com desvios ou pontos de começo alternativos, e ainda assim serem a mesma linha. Daí a necessidade de criar esta Entidade.

Como quase todas as entidades tem o seu ID, um código, uma referência a um possível percurso que lhe tenha dado origem, a referência se é o percurso principal dessa linha, a sua orientação.

Também se encontra representada a sua prioridade em relação aos outros percursos dessa mesma linha. É feita naturalmente uma referência à linha à qual o percurso presente e pode ser especificada uma data de início e de fim consoante o propósito do percurso.

Este percurso pode ter uma descrição e existe um último campo que permite dizer se ele deve ser representado ou não na espinha, sendo esta uma opção mais avançada do gerador.

Tabela Path na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
code	VARCHAR(255)	Código do Path
base_path_id	NUMERIC(19)	Path de origem do Path em questão
is_main	NUMERIC(5)	Verifica se o Path é principal
ext_code	VARCHAR(255)	Código externo do Path
orientation	CHAR(1)	Orientação do Path (Ida, Volta ou Circular)
priority	NUMERIC(19)	Prioridade do Path (entre os Paths da Line)
line	NUMERIC(19)	Referência à Line (FK)
init_date	DATETIME	Data de início do Path em vigor
end_date	DATETIME	Data de fim do Path em vigor
description	VARCHAR(2048)	Descrição do Path
in_spine	NUMERIC(5)	Verifica se o Path deve aparecer na espinha
version	DATETIME	Campo para verificação de problemas de concorrência

PathStop

Um Path é composto por um conjunto de Stops em sequência que definem o percurso em si. Contudo é necessária uma entidade especial que ligue os Paths e os Stops por ser uma ligação de “muitos para muitos”, para além da informação adicional que é necessária guardar como a ordem da paragem no percurso e possivelmente informações sobre distância e duração média entre paragens nesse percurso.

Tabela PathStop na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
prev_distance	FLOAT	Distância para o PathStop anterior cuja prev_dist seja diferente de zero
segment_break_duration	FLOAT	Tempo de duração de viagem entre este PathStop e o anterior cujo segment_break_duration seja diferente de zero
stop_order	NUMERIC(19)	Ordem do PathStop no Path
path	NUMERIC(19)	Referência ao Path (FK)
stop	NUMERIC(19)	Referência à Stop (FK)
version	DATETIME	Campo verificação de problemas de concorrência

Passing

O Passing é a entidade que representa uma hora de passagem de um veículo a fazer uma linha num determinado percurso, numa paragem desse percurso. O passing também tem associado um Schedule para definir em que dias da semana e épocas do ano este Passing está em vigor.

Tabela Passing na Base de Dados

Campo	Tipo	Descrição
id	NUMERIC(19)	ID único de identificação na Tabela
schedule	NUMERIC(19)	Referência ao Schedule (FK)
path_stop	NUMERIC(19)	Referência ao PathStop (FK)
time	NUMERIC(19)	Tempo em quantidade de minutos desde a meia-noite do dia a que este Passing se refere

PathStopZone

PathStopZone é a tabela que faz a ligação de “muitos para muitos” entre as PathStops e as Zones pois tendo em conta que podem haver vários tipos de zonas uma PathStop pode estar associada a várias zonas, da mesma maneira que uma zona tem muitas PathStops associadas.

Tabela PathStopZone na Base de Dados

Campo	Tipo	Descrição
pathstop	NUMERIC(19)	Referência à PathStop (FK)
zone	NUMERIC(19)	Referência à Zone (FK)

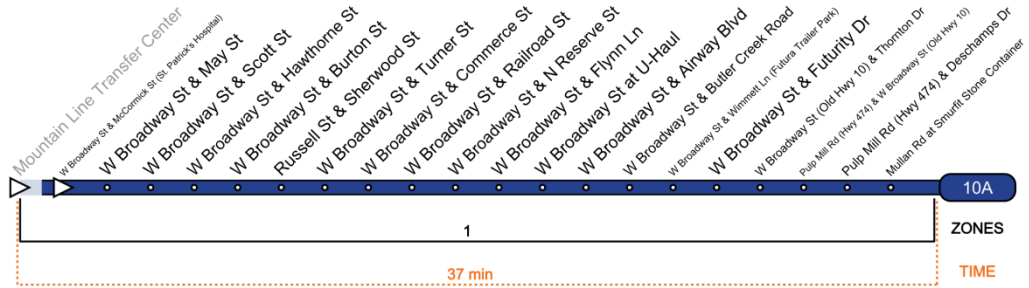
Anexo B – Exemplos de Documentos

10A

AIRPORT / SMURFIT STONE

This line has the following interface: ↑ Airport

(0057)



Test Company

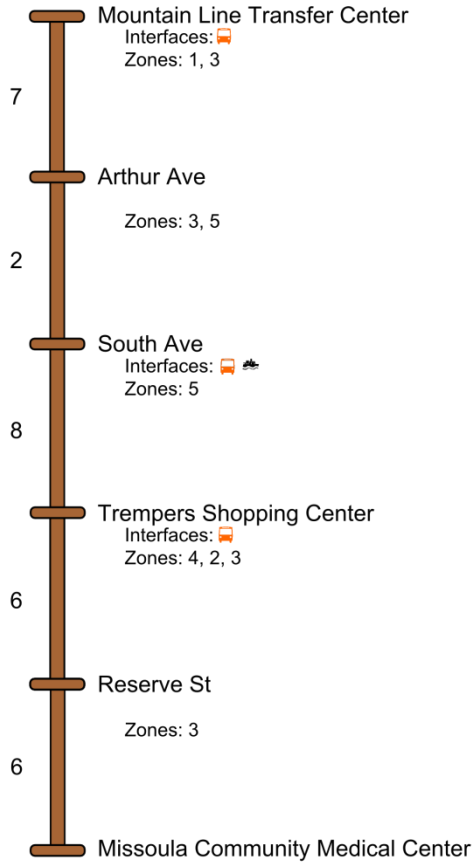
Available since: 26-06-2009

1A

UNIVERSITY / COMMUNITY HOSPITAL

(0013)

Mins



🚌 SpecialBus

🚤 Boat



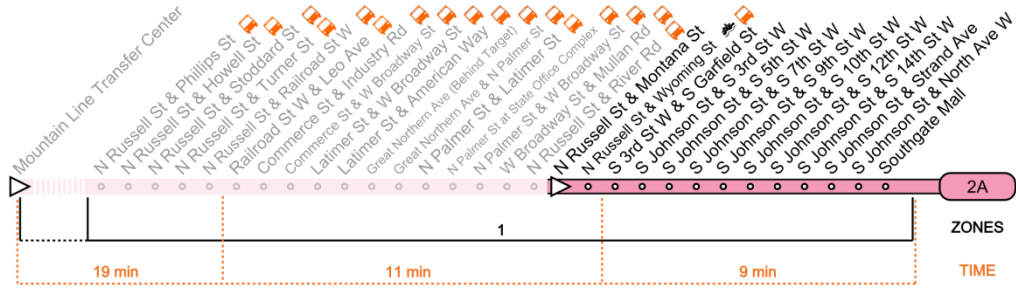
Test Company

Available since: 26-06-2009

2A

TARGET / SOUTHGATE MALL

(0063)



SpecialBus Boat



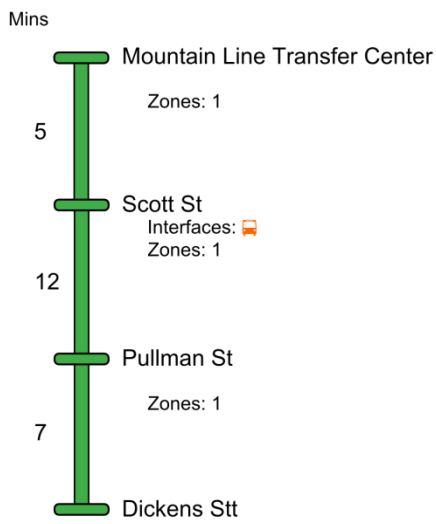
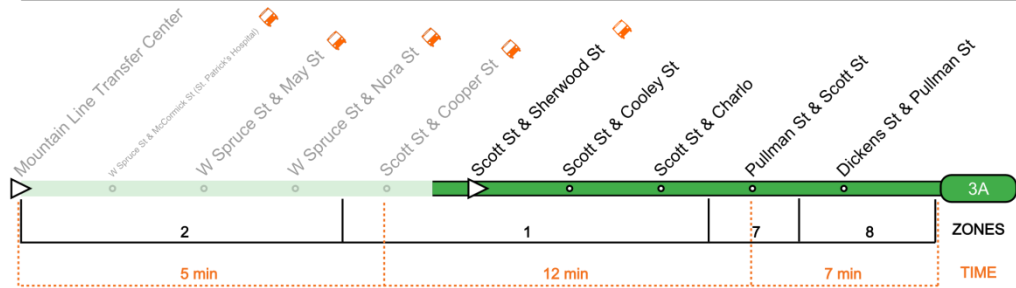
Test Company


Available since: 26-06-2009

3A

NORTHSIDE (OUTBOUND)

(0127)



 SpecialBus



Bus Company

Available since: 26-06-2009