TÉCNICO LISBOA

Desenvolvimento de um Módulo de Business Intelligence para o XTraN Passenger

Produto para Gestão de Frotas de Transportes de Passageiros

Catarina Da Conceição Filipe

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Orientador: Prof. Alberto Manuel Ramos da Cunha

Júri

Presidente: Prof. Daniel Jorge Viegas Gonçalves
Orientador: Prof. Alberto Manuel Ramos da Cunha
Vogal: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

outubro 2021

Agradecimentos

Gostaria de agradecer à Tecmic S.A. e ao António Marcelo pela oportunidade e desafio proposto, ao Jóni Batista, e em especial ao André Leal pelo seu tempo e por todos os conselhos que me deu ao longo deste projeto. Uma palavra de agradecimento ao Professor Alberto Cunha pela orientação e valiosas recomendações. Aos meus amigos do secundário, aos amigos da faculdade, e aos amigos do secundário que só já conheci na faculdade, agradeço a amizade e dicas que me deram ao longo deste projeto. Uma palavra especial à Raquel e à Rita por serem uma constante e por me incentivarem a acabar esta dissertação, e à Marta e ao Lucas, os meus eternos companheiros de projeto. Por fim quero agradecer à minha família, principalmente ao Pedro, pelo seu companheirismo incalculável, e aos meus pais, por todo o apoio incondicional que me deram durante este percurso.

Abstract

As companies face an ever-increasing flow of raw data, it is a challenge to deconstruct and gain detailed insight into that information. With a Business Intelligence process, it is possible to get a comprehensive view of the data and translate it into new insights about the activity, enabling informed business decisions. The aim of this project is to develop a Business Intelligence system for XTraN Passenger, Tecmic's fleet and passenger management product. For this process to be supported, a technological infrastructure is required. This document presents the development of the Data Warehouse and Business Intelligence modules of the system. The DW module involves a model that transforms the data extracted from the existing operating aid system in passenger transports, and its insertion into a SQL Server DW. The BI module allows the analysis of the stored data, through reports created by Microsoft's Power BI, which are integrated in the XTraN Passenger's web platform. The solution was evaluated in usability tests, which concluded that the refreshing of the graphics as the reports are manipulated, is within reasonable latency parameters, until at least, 10 million travel records stored in the DW, 2 years, in an average of 15 thousand daily trips. With this, passenger transport companies, who purchase the product from Tecmic, will be able to monitor key indicators of the activity, and with this make decisions regarding the business model.

Keywords

Business Intelligence, Data Warehouse, Fleet Management, Power BI

Resumo

À medida que as empresas enfrentam um fluxo cada vez maior de dados em bruto, é um desafio desconstruí-los e obter uma visão detalhada da informação que representam. Com um processo de Business Intelligence, é possível obter uma visão abrangente dos dados e traduzi-los em novas perspetivas sobre a sua atividade, permitindo a tomada de decisões de negócio informadas. É objetivo deste projeto, o desenvolvimento de um sistema de Business Intelligence para o XTraN Passenger, produto para a gestão de frotas e passageiros da empresa Tecmic. Para este processo ser suportado, é necessária uma infraestrutura tecnológica. Neste documento apresenta-se o desenvolvimento dos módulos de Data Warehouse e Business Intelligence do sistema. O módulo DW passa por um modelo que transforma os dados extraídos do sistema de ajuda à exploração existente em transportes de passageiros, e sua inserção num DW em SQL Server. O módulo BI permite a análise dos dados armazenados, por meio de relatórios criados pela ferramenta da Microsoft, Power BI, que estão integrados na plataforma web do XTraN Passenger. A solução foi avaliada em testes de usabilidade, que permitiram concluir que a atualização dos gráficos nos relatórios funciona dentro dos parâmetros de latência razoáveis, até pelo menos, com 10 milhões de registos de viagens armazenados no DW, correspondentes a 2 anos, numa média de 15 mil viagens diárias. Com isto, as empresas de transporte de passageiros que adquirem o produto à Tecmic, poderão monitorizar indicadores chave da atividade, e com isto tomar decisões a nível estratégico.

Palavras Chave

Business Intelligence, Data Warehouse, Gestão de Frotas, Power BI

Conteúdo

Lista de Figuras

Lista de Tabelas

Listagens

Acrónimos

API	Application Program Interface
BD	Base de Dados
BI	Business Intelligence
CSV	Comma Separated Values
DAX	Data Analysis Expressions
DW	Data Warehouse
ETL	Extract, transform, load
GPS	Global Positioning System
GTFS	Especificação Geral de Feed de Trânsito
GUI	Graphical User Interface
ISV	Independent Software Vendor
JDBC	Java Database Connectivity
MPP	Massive parallel processing
NeTEx	Network Timetable Exchange
ODBC	Open Database Connectivity
RDD	Resilient Distributed Dataset
SAEIP	Sistema de Ajuda à Exploração e Informação aos Passageiros
SCD	Slowly Changing Dimensions
SIRI	Service Interface for Real Time Information
SMS	Short Message Service
SQL	Structured Query Language
SSAS	SQL Server Analysis Services

SSIS	SQL Server Integration Services
SSISDB	SSIS Catalog database
SSMS	SQL Server Management Studio
SSRS	SQL Server Reporting Services
VPN	Virtual Private Network
WAN	Wide Area Network
XML	eXtensible Markup Language

1

Introdução

Conteúdo

1.1 Motivação

A mobilidade de pessoas é um dos elementos essenciais do desenvolvimento urbano. Nos últimos 50 anos, o transporte pessoal tem sido dominado por veículos particulares movidos por motores de combustão interna. Embora ofereçam aos utilizadores um grande grau de liberdade, a sua adoção em massa nas cidades levou ao sucessivo congestionamento do tráfego, a impactos negativos no meio ambiente, saúde humana e redução da sua habitabilidade geral. À medida que a população mundial se continua a concentrar nas cidades, estes impactos negativos associados ao transporte são exacerbados. [1]

Uma maneira de inverter esta tendência é a existência de uma rede robusta e fiável de transportes públicos, que proporcione, de uma forma acessível, mobilidade à população. Para que isso ocorra, uma extensiva rede de transportes públicos necessita de um sistema que ajude a gerir e monitorizar a frota e o seu planeamento, para garantir uma elevada qualidade dos mesmos, e para que assim se mantenham atrativos à população.

Esta tese enquadra-se no contexto do produto XTraN Passenger desenvolvido pela empresa multinacional portuguesa, Tecmic - Tecnologias de Microeletrónica, fundada em 1988, que tem a sua atividade direcionada para soluções de gestão e monitorização de frotas.

A Tecmic é uma empresa de relevo no mercado nacional, oferecendo serviços a empresas como a Carris, empresa de referência no transporte público de passageiros da área metropolitana de Lisboa, que ultrapassa os 139 milhões [2] de passageiros anuais. Exporta também os seus produtos para mais de 17 países na Europa, África e América Latina, com perspetivas de sucessiva expansão.

Para se evidenciar como líder no mercado, a Tecmic necessita de uma constante evolução e de disponibilizar nos seus produtos algo que a diferencie dos restantes, e conseguir dar resposta às necessidades específicas das empresas usufruem dos seus produtos.

No conjunto de soluções que a Tecmic dispõe, surge então o XTraN Passenger, um sistema de ajuda à exploração e informação aos passageiros, que permite o acompanhamento de frotas em tempo real, ao recolher um vasto conjunto de dados da atividade, através de hardware a bordo dos veículos. Para além disto, reúne a informação relativa ao planeamento das rotas, tanto a nível horário, como a nível do itinerário.

A recolha de dados da atividade das frotas é a base deste produto, mas entende-se que este não atinge o seu potencial máximo, se não tiver na sua composição um mecanismo que permita analisar estes dados de uma forma estruturada e simplificada. De momento, o sistema carece então de um módulo que permita analisar a performance da atividade, numa forma de controlo periódico, sem necessidade de conhecimentos informáticos por parte de quem os supervisiona. Uma funcionalidade fundamental para as empresas que adquirem o XTraN Passenger poderem monitorizar como estão os indicadores da sua atividade, e com base neles poder tomar decisões quanto à forma do negócio, como, por exemplo, fazendo alterações nos horários ou na supressão de troços redundantes nas rotas, numa forma de otimização da atividade e redução de custos, e em última análise a evolução da empresa.

É meta deste projeto a construção de um módulo que permita complementar o XTraN Passenger neste aspeto, e com isto torná-lo um produto mais competitivo no seu sector.

1.2 Objetivos do Trabalho

O desafio deste projeto passa pela criação de um BackOffice para o produto XTraN Passenger. Uma adição ao sistema que permitirá às empresas fornecedoras de serviço de transporte público clientes da Tecmic analisar a performance da atividade da sua frota, com uma automatização na disponibilização de relatórios, e interfaces que forneçam dados relativos a cada dia da exploração, em que seja possível fazer em paralelismo entre o planeamento e a execução.

A atividade da frota produz continuamente uma grande quantidade de dados no decorrer da sua operação. Como tal, é importante que as ferramentas de análise associadas tenham a capacidade de processar e interpretar este vasto volume de dados para apoiar a melhoria da qualidade do serviço.

A solução passa por realizar o tratamento, transformação e organização de dados provenientes das Base de Dados (BD) do Sistema de Ajuda à Exploração e Informação aos Passageiros (SAEIP), migrando-os para um Data Warehouse (DW), para que mais tarde seja possível gerar análises de Business Intelligence (BI) sobre esta.

Tem-se então como principais passos:

- A seleção de atributos e criação de views na BD de exploração, de forma a selecionar informação que seja relevante para as análises.
- A definição e concretização do processo Extract, transform, load (ETL), onde os dados provenientes das bases de dados serão tratados

- A definição da estrutura do DW
- A escolha de uma ferramenta de BI que disponibilize uma camada de apresentação
- A integração da camada de apresentação na plataforma web do XTraN Passenger
- Encontrar as ferramentas adequadas para lidar com o volume de dados criados

1.3 Estrutura do Documento

Este documento encontra-se estruturado da seguinte forma: O Capítulo 2 discute trabalho relacionado, descrevendo todo o sistema XTraN Passenger, os seus módulos e mais especificamente as bases de dados fonte de interesse a este projeto, fala também de metodologias e tecnologias usadas hoje em dia. No Capítulo 3 é descrita a proposta de solução, com a arquitetura geral pensada para este modelo. O Capítulo 4 fala em detalhe da implementação do novo módulo do XTraN Passenger, incluindo todos os componentes e passos dados no seu desenvolvimento. O Capítulo 5 aborda o método de testagem do sistema e é feita a avaliação geral dos objetivos. Por fim, no Capítulo 6 são retiradas as conclusões deste projeto, há uma reflexão sobre os desafios encontrados, se os objetivos foram ou não alcançados e são feitas prospeções para trabalho futuro.

2

Trabalho Relacionado

Conteúdo

2.1 XTraN Passenger

O XTraN Passenger é um SAEIP, que decorre de uma estratégia de desenvolvimento que pretende conferir uma maior competitividade ao transporte público de passageiros.

Tem a capacidade de monitorizar as frotas em tempo real baseada na localização de cada veículo e sua interpretação no âmbito de cada carreira, sendo possível seguir cada veículo sobre a planta da cidade, através de um sistema de GPS, complementado ainda através de um sistema de odómetro e de abertura de portas nas paragens. O sistema permite que toda a frota em circulação seja acompanhada em tempo real na central de comando.

O motorista de serviço dispõe de uma consola de bordo, onde se regista no início de serviço e possui informação sobre as rotas a efetuar, o avanço ou atraso em relação ao horário planeado, e a disponibilização de mensagens de texto pré-definidas para comunicar com a central, útil em casos de congestionamentos ou interrupção da circulação.

Possuí também um módulo de disponibilização de informação ao público em tempo real através de painéis eletrónicos colocados nas paragens, internet e SMSs, proporcionando uma maior qualidade do serviço prestado e consequentemente, a satisfação dos clientes.

De seguida segue-se a síntese dos módulos que atualmente compõem o XTraN Passenger [3]:

InfoPublic: Plataforma que disponibiliza aos passageiros informação precisa e em tempo real, através de painéis, internet e via SMS.

Eco-Driver: Controla a eficiência energética da condução, promovendo melhores práticas de condução para tal, visando assim a redução do consumo de combustível, outros custos operacionais e um aumento geral da segurança e conforto dos passageiros.

Counter: Sistema inteligente de contagem de passageiros, que fornece dados relativos à afluência do serviço, em termos horários e de localização.

Bus DVR: Regista imagens de alta qualidade em movimento relativo.

InfoDesigner: Permite a geração dinâmica de informação ao público, como posters direcionados à rede de transportes, linha ou paragem.

Infotainer Informação dinâmica a bordo relativa ao serviço, fornecendo entretenimento e publicidade baseada na localização.

Operations & Resource Planner: Módulo de planeamento, otimização e atribuição dos serviços aos veículos e condutores.

2.1.1 Base de dados do SAEIP

Os dados registados durante a operação das frotas são persistidos na base de dados do sistema de ajuda à exploração. As tabelas da base de dados contêm dados sobre o planeamento e a execução das viagens na rede. De uma forma abstrata, existem tabelas a nível da estrutura da rede, que podemos considerar uma hierarquia de carreiras, variantes de carreira, troços, ligações e paragens. Ao iniciar funções, cada condutor terá um serviço a realizar, constituído por um conjunto de viagens, que podem ser realizadas na mesma carreira ou não. Este agrupamento de viagens é identificado por algo denominado de chapa. Para obter o planeamento para cada serviço são tidas em atenção variantes temporais, como a época sazonal, o tipo de dia da semana e a hora do dia, dado que a frequência de cada carreira e o tempo de ligação entre duas paragens será dependente destas. Durante o serviço, os dados do planeamento nesta base de dados, auxiliam o motorista quanto às rotas que tem de efetuar e quanto à sua execução face ao planeamento. À medida que o serviço é efetuado, os eventos referentes a este são recolhidos para a base de dados.

2.2 Background

Esta secção está estruturada da seguinte forma: em 2.2.1 e 2.2.2 é feito um background tecnológico das temáticas em torno de BI, sendo feita a descrição do estado do conhecimento de sistemas de BI e DW. No fim de cada uma destas secções é feita uma análise de tecnologias com potencial para serem utilizadas no desenvolvimento deste projeto. Na secção 2.2.3 é feito um levantamento de práticas comuns na construção de um sistema de BI para frotas de transportes de passageiros, incluindo a identificação de indicadores relevantes da atividade.

2.2.1 Business Intelligence

Existem diversas definições do que é Business Intelligence, mas podemos sintetizá-la como um conjunto de ferramentas e sistemas inteligentes que ajudam uma empresa a monitorizar e explorar o que ocorre na sua atividade.

É um conceito que inclui a recolha, integração, análise e visualização de dados operacionais para apoiar e melhorar o processo de tomada de decisão [4]. As etapas de um processo de BI, segundo Eckerson [5] são: a *aquisição de dados*, que passam por um processo de ETL, e são *armazenados* numa base de dados multidimensional, geralmente um Data Warehouse, onde podem ser feitas *análises* e onde há uma camada de *apresentação*. Esta apresentação dos dados é feita sob a forma de relatórios

dinâmicos, alertas e dashboards operacionais.

Este conjunto de ferramentas e sistemas referidos ajudam a empresa a transformar dados, em informação útil e significativa, que ajuda a encontrar e resolver potenciais problemas, identificar e aproveitar novas oportunidades, prever e planear o futuro e alinhar as operações conforme com os objetivos, podendo ser empregue em qualquer nível hierárquico dentro de uma empresa: estratégico, tático ou operacional.

2.2.1.A BI Tradicional

Na abordagem tradicional, um sistema de BI explora várias tecnologias, onde são produzidos relatórios, a fim de melhorar a eficácia da tomada de decisões. São usados mecanismos de criação de relatórios, para aceder aos dados armazenados no DW. Um sistema tradicional é composto por três camadas distintas: camada de apresentação, da aplicação e do DW [4].

Costuma ser usado numa vertente estratégia dentro da empresa. A este nível, permite a trabalhadores com tarefas executivas, uma visão clara e concisa sobre a atividade da empresa, ajudando a gerir o desempenho empresarial de acordo com objetivos estratégicos. Alguns destes objetivos incluem novas prospeções de desenvolvimento no mercado, decisões de investimento, ou modificações no modelo de negócio.

2.2.1.B BI Nova Geração

Os sistemas de BI da nova geração, são caracterizados por novas aplicabilidades. Enquanto um sistema de BI tradicional se direciona para aspetos estratégicos, surgem agora sistemas para ser usados a nível operacional. É o caso de sistemas *Real-Time*, que têm como objetivo transmitir informação em tempo real aos operadores em vez de dados históricos. Há a redução da latência entre o tempo de aquisição de dados e o tempo de análise. A diminuição desta janela temporal, permite a monitorização imediata, e a tomada de ações apropriadas ao desenvolvimento da atividade à medida que ocorrem novos eventos.

Outro tipo de BI a emergir é o *Self-Service BI*, que permite a utilizadores sem conhecimentos técnicos criar relatórios e fazer consultas analíticas. Caracteriza-se pelas suas interfaces user-friendly, intuitivas e de fácil utilização [4], existindo uma barreira de abstração entre o utilizador e o DW.

Da mesma maneira, os avanços tecnológicos e os novos Sistemas Ciber-Físicos oferecem novas capacidades de BI, como a existência indicadores preditivos e adaptativos. Estas funcionalidades caracterizam-se por privilegiar a previsão à reação, conseguem, usando *machine learning*, prever

padrões de atividade futura, para qual as empresas podem tomar medidas preventivas. É uma mais-valia a nível operacional, e principalmente a nível tático, numa empresa.

2.2.1.C Ambiente de Implementação

Com o crescimento de cloud computing, também se deu o aparecimento de soluções de BI na cloud, ou seja a oferta deste serviço através de uma arquitetura na cloud. BI como Software-as-a-Service é uma opção cada vez mais generalizada pela sua rapidez de implementação e flexibilidade [6], sendo um modelo onde não há infraestrutura para gerir por parte da empresa.

A implementação de sistemas de BI na cloud está em crescimento, mas ainda existem razões para uma empresa preferir o método tradicional, armazenamento num sistema local. Estas razões recaem em dois grandes fatores, a velocidade na transferência dos dados, e a segurança. Embora novas soluções para estes problemas estejam a emergir, ainda se verificam obstáculos para algumas empresas.

Para negócios que lidam com vastos conjuntos de dados, ou Big Data, pode revelar-se uma tarefa pouco eficiente transferir todos estes dados para a cloud. Alguns vendedores de software, estão a desenvolver soluções que envolvem otimizar as conexões WAN para acelerar este processo de transferência de dados. Todavia, este tipo de solução ainda está em desenvolvimento e requer investimento [7].

Deste modo, as empresas que produzem os seus dados internamente, optam muitas vezes por uma solução on-premises.

O segundo problema referido é a segurança. Embora a compra da infraestrutura e a manutenção de servidores locais signifique um investimento inicial acrescido, a empresa tem controlo direto sobre os servidores e sistemas de segurança. Não sendo imunes a possíveis ataques, é mais fácil a restrição do acesso ao hardware.

Neste momento, as mais reconhecidas empresas no ramo de BI, oferecem na sua maioria, ambas as opções de armazenamento, na cloud e on-premises.

2.2.1.D Interação

Um aspeto importante a ter em conta, aquando da implementação da camada de apresentação de um sistema de BI, é o tempo de resposta das interfaces. Em *Usability Engineering* [8] é descrito que numa interação com a interface:

0.1 segundos é a janela temporal limite para que o utilizador sinta que o sistema está a reagir instantaneamente.

1 segundo é o limite temporal para a linha de pensamento de um utilizador permanecer ininterrupta, mesmo que se aperceba do ligeiro atraso na resposta. Com este *delay*, perde a sensação de controlo direto nos dados.

10 segundos é o limite para manter a atenção do utilizador. Em atrasos superiores, os utilizadores querem realizar outras tarefas enquanto esperam por resposta. Poderá ser necessário *feedback* adicional.

2.2.1.E Soluções de BI

A escolha de uma plataforma BI é crucial para o desenvolvimento do novo módulo da XTraN Passenger. As operações possibilitadas por esta ferramenta terão de estar incorporadas na plataforma web. Terá de ser feito um balanço entre funcionalidade, usabilidade, custo e estética, tendo em conta que este serviço é a camada entre os dados e o utilizador. Serão analisadas ferramentas dentro de 3 patamares de preço.

Google Data Studio

Google Data Studio é uma ferramenta de BI criada pela Google. Funciona exclusivamente em ambiente web e necessita apenas de uma conta Google para criar conteúdos. Permite a criação de relatórios, dashboards e gráficos personalizados. O seu uso tem uma baixa curva de aprendizagem, com features interativas como o drag-and-drop, e é simples de incorporar numa plataforma web. Suporta poucas fontes de dados, nomeadamente, não possui um conector nativo para o Microsoft SQL Server. Há ferramentas com um baixo custo associado que podem fazer de conector, como o "Analytics Canvas". Outra opção seria usar outra base de dados com conectores nativos ao Data Studio, como MySQL ou PostgreSQL, ou usar ficheiros CSV como intermediários. É uma ferramenta sem qualquer custo associado, apelativo em termos financeiros, mas bastante rudimentar em termos de funcionalidade. Não permite a visualização da partilha sem que o utilizador esteja associado a uma conta Google e carece de qualquer suporte de fornecedor.

Power BI

O Power BI, é uma ferramenta de Business Intelligence desenvolvida pela Microsoft, que permite ao utilizador analisar e criar visualizações de grandes conjuntos de dados provenientes das mais variadas fontes. Facilita a exploração dos dados ao detalhe, através de, por exemplo, *drill downs* e *drill throughs*, possibilita a exposição dos dados de forma a encontrar padrões e, assim, ganhar perspetivas que podem ser úteis às tomadas de decisão inerentes a qualquer negócio [9].

A utilização de fórmulas não é essencial na composição de relatórios, mas para representar informação proveniente de cálculos mais aprofundados, existem Data Analysis Expressions (DAX). O DAX é uma biblioteca de funções e operadores que podem ser combinados para criar fórmulas e expressões no Power BI [10].

O Power BI Desktop é aplicação onde são criados e compostos os relatórios, que mais tarde são publicados numa das duas opções: o serviço Power BI, alojado na cloud, ou Power BI Report Server. De seguida são descritas particularidades do Report Server, solução de particular relevância.

O PBIRS é uma versão on-premises do Power BI, vantajosa para empresas que escolhem manter os dados e relatórios armazenados localmente. O PBIRS é uma extensão do SSRS report server, encontra-se incluído na licença Premium do Power BI e no SQL Server Enterprise Edition com Software Assurance. Os custos desta última traduzem-se na licença do SQL Server para o número de cores necessários para executar tanto o SQL Server como o PBIRS, a taxa do Software Assurance, e pelo menos uma licença Pro necessária para publicar relatórios. Para visualizar os conteúdos publicados, os end users não necessitam de licenças próprias, apenas de acesso à infraestrutura onde está instalado o Report Server, normalmente na rede da empresa, protegido pela sua firewall [11].

Para incorporar os conteúdos numa plataforma web externa, podem ser utilizadas iframes HTML e APIs que onde estão incluídas as seguintes: APIs REST, acesso por URL e WMI Provider. Algumas capacidades disponíveis nas versões Power BI na cloud, como Dashboards e Natural Language Query, não estão ainda incluídas no PBIRS.

Tableau

Tableau pertence à empresa americana Tableau Software, dedicada a serviços de BI. Assim como a maioria do software de BI, o Tableau permite a criação de dashboards, relatórios dinâmicos e um vasto conjunto de visualizações, com recurso a ferramentas que permitem uma pormenorizada exploração de amplos conjuntos de dados, e a descoberta de novas informações e tendências de negócio. Apresenta uma interface simplificada, com recursos de *drag-and-drop*, não sendo necessário um grande *know-how* para utilizá-lo, permite que o utilizador faça análises com facilidade e liberdade criativa. Funciona em todos os tipos de dispositivos, não tendo o utilizador necessidade de se preocupar em que ambiente vai executar o Tableau, porque ele não necessita de nenhum hardware ou software específico para funcionar. Tal como o Power BI, possui soluções na cloud, Tableau Online e on-premises, Tableau Server. [12, 13]

2.2.2 Data Warehouse

Segue-se uma introdução ao termo Data Warehouse, um componente chave de um sistema de BI. Um DW funciona como uma BD relacional projetada para consulta e análise, em vez de processamento de transações. A principal diferença está na estrutura, onde as BDs são projetadas e otimizadas para armazenar dados, e um DW é projetado e otimizado para responder a perguntas e pesquisas de análise críticas para o negócio de uma organização. Geralmente contém dados históricos derivados de dados de transações, ou seja, armazena dados redundantes, estruturados de uma nova forma.

Esta definição vai de em conta com a definição de Kimball de DW, em que *"Um DW é uma cópia de dados de transação, especificamente estruturada para consulta e análise"* [14].

Outra das definições mais consensuais para DW vem de Inmon, que diz *"Um DW é uma coleção de dados orientada ao assunto, integrada, variável no tempo e não volátil, que suporta processos de tomada de decisão"* [15]. Como resumo destas características temos:

Orientado a assunto: Tem como principal objetivo gerar informações sobre um assunto particular.
Integração: Os dados de um DW são extraídos de dados descentralizados, provenientes de várias fontes, que passam por um processo de colheita e reorganização.
Variabilidade Temporal: Para efeitos de decisão, os dados no DW precisam de estar associados a um atributo temporal, pois um DW tem como funcionalidade a conservação de um histórico de dados por um período superior que os sistemas comuns.
Não volatilidade: Os dados existentes num DW pautam-se pela sua estabilidade, na qual novos dados são inseridos num DW, mas nunca modificados.

Os DW e as suas arquiteturas variam dependendo das especificidades e necessidades de uma organização: A arquitetura de um DW pode ser considerada básica, sendo apenas o DW, pode conter também uma área de *staging*, o sítio onde os dados são processados antes de entrarem no DW. E pode ser composta por uma área de *staging* e por *Data Marts* [16] que são bastante comuns quando se quer personalizar a arquitetura para diferentes repartições da organização, onde cada um dos *Data Marts* é direcionado para cada um destes grupos e as suas necessidades .

Associado a um DW está normalmente associado um processo ETL. *Extract*, *Transform*, and *Load* é um método de integração de dados, o processo de migrar dados de um sistema para outro. Normalmente utilizado para construir DWs, é realizado em 3 etapas:

Extract: Os dados são extraídos dos sistemas fonte, neste caso as bases de dados de ajuda à exploração.

Transform: Os dados extraídos são limpos e uniformizados de acordo com a estrutura para onde vão ser carregados. Onde normalmente são aplicadas as seguintes técnicas:

- Limpeza, tratamento de anomalias e padronizar dados;
- Eliminação de campos desnecessários;
- Junção de diferentes tabelas de dados provenientes de fontes diferentes;
- Criação de novas chaves primárias;
- Construção de agregados de modo a acelerar as pesquisas;

Load: Alimentação do DW com os dados tratados.

Os dados seguem este processo até que estarem de acordo com a estrutura do DW.

2.2.2.A Soluções para o Processo ETL

Para realizar manipulações no volume acrescido de dados provenientes do SAEIP, será preciso um método robusto e escalável, aqui surgem os seguintes:

Apache Spark

Apache Spark é uma framework open source, que tem o propósito de analisar e processar eficientemente grandes volumes de dados, de uma forma paralela e distribuída. Foi concebido para ser rápido e de uso geral.

O Spark amplia o modelo MapReduce do Hadoop, para suportar com eficiência diversos tipos de computação, como consultas interativas e processamento em stream. Além disto, e sendo a rapidez um critério essencial quando falamos do processamento de consideráveis quantidades de dados, a rapidez do Spark advém do facto de poder ser executado em memória principal.

Oferece APIs que permitem a escrita de aplicações em várias linguagens de programação e integra um conjunto de bibliotecas, que fornecem capacidades adicionais para outras áreas de análise, como se observa na Figura 2.1.

O Spark pode ser configurado para ser executado num cluster Hadoop juntamente com um gestor como o Hadoop Yarn ou Apache Mesos, ou numa versão mais simples, pode ser instalado utilizando o gestor

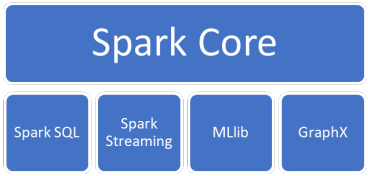

Figura 2.1: Componentes Apache Spark

do próprio Spark Core, denominado de standalone.

O Spark core, é responsável pelas funcionalidades básicas do Spark, incluindo o escalonamento de processos, gestão da memória e tolerância a falhas entre outras. É também no core que está a API de Resilient Distributed Datasets (RDDs), o principal objeto da abstração da programação em Spark, que se encontram repartidos entre computadores, é nestes objetos que é executado o processamento dos dados.

O Spark SQL, é um módulo utilizado para trabalhar com dados estruturados. Disponibiliza uma interface similar à de SQL para realizar consultas nos dados. Permite ligações JDBC ou ODBC para aceder a dados de bases de dados. Uma das suas mais valias é a abstração de programação chamada DataFrames cuja manipulação pode ser feita em Scala, Java e Python, e pode também funcionar como motor de consultas Structured Query Language (SQL) distribuída. Com recurso ao Spark é possível a escrita de um processo ETL robusto, capaz de processar um grande volume de dados [17].

SSIS

SSIS, acrónimo para SQL Server Integration Services, é uma plataforma para a construção de soluções de integração e transformação de dados. Automatiza funções administrativas e o carregamento de dados, ao permitir responder ao mais diverso tipo de problemas como carregar e descarregar ficheiros, limpar e transformar dados e gerir objetos do SQL Server e popular DWs. Consegue aceder e extrair dados de variadas fontes tais como ficheiros de dados XML, CSV, e fontes de dados relacionais, e depois carregar os dados para o número de destinos que for necessário. Dentro das suas mais valias, oferece a possibilidade de um tratamento robusto de erros e eventos, a identificação e processamento de alterações em dados, e consegue processar inputs com milhões de registos da fonte ao destino numa questão de minutos. Contêm uma Graphical User Interface (GUI), que permite operar uma série de ferramentas gráficas que ajudam o utilizador a construir pacotes e a compor o seu modelo, em vez de ter de recorrer a código em si.

Quanto à arquitetura do SSIS, temos uma hierarquia de solução, projeto e pacote. Dentro do pa-

cote, temos o Control Flow, contém um conjunto de tarefas a executar, como uma tarefas de envio de email, execução de uma tarefas SQL, Script ou de Data Flow. Visando a produção de um ambiente de integração, há um especial interesse na tarefa de Data Flow onde internamente há uma toolbox, com elementos para a construção de um pipeline de fluxo de dados, com fontes, transformações e locais de destino para inserir os dados. Tem-se também um Event Handler que permite lidar com eventos que surjam, um Package Explorer, que oferece uma vista geral para todo o pacote, e parâmetros, que permitem interação com o utilizador. Os Serviços de Integração incluem também a base de dados do Catálogo de Serviços de Integração, onde se armazenam, executam e gerem pacotes [18].

2.2.2.B Soluções de DW

Azure Synapse Analytics

O Microsoft Azure Synapse é uma evolução do Azure SQL Data Warehouse. É uma solução end-to-end que reúne Data Warehousing e análise de Big Data num único serviço. Está integrada num todo sistema Microsoft, que inclui ferramentas como o Apache Spark e o Power BI.

Nas suas especificidades de enquanto serviço de DW, é uma base de dados localizada na cloud, escalável e permite processar grandes quantidades de dados, uma característica possibilitada pela sua arquitetura Massive parallel processing (MPP), onde cada instância é processada por um nó dedicado que tem o seu próprio processador e memória [19]. O tamanho deste DW é teoricamente ilimitado, sendo os limites impostos apenas pelo custo associado.

SQL Server

O SQL Server, também conhecido como MSSQL, foi desenvolvido pela Microsoft nos anos 80. Desde então, tornou-se a plataforma de eleição para grandes empresas, devido à sua escalabilidade e fiabilidade.

Suporta uma grande variedade de aplicações de processamento de transações, análise e BI. Embora o SQL Server seja considerado um *Relational Database Management System*, nos últimos anos a Microsoft tem vindo a implementar um conjunto de ferramentas e novas funcionalidades importantes para atender a necessidades do mercado de DW, sendo capaz de comportar um esquema multi-dimensional utilizando o Analysis Services.

SSMS

O SSMS é um ambiente integrado para a gestão de qualquer infraestrutura SQL, como o SQL Server. Fornece ferramentas para configurar, monitorizar, e administrar instâncias do SQL Server e bases de

dados. Pode ser utilizado para implementar, monitorizar e atualizar os componentes em bases de dados, fazer consultas e gerar scripts [20].

2.2.3 Transporte de Passageiros

2.2.3.A Indicadores de Performance

Uma das questões que precede a análise de práticas comuns na construção de um sistema de BI para a atividade de frotas de transporte de passageiros, é identificar os indicadores mais relevantes da atividade. A monitorização e a tomada de decisões apoiada por estes indicadores será mais pertinente. Friman, identificou a frequência, tempo de viagem, preço, informação, limpeza, conforto do autocarro, conduta dos funcionários, disponibilidade de lugares, segurança da paragem de autocarro, segurança face a acidentes, segurança a bordo, condição da paragem de autocarro e informação na paragem de autocarro como os atributos que melhor medem a qualidade do serviço [21]. Numa seleção dos atributos da atividade a analisar, estes podem ser um bom ponto de partida.

Noutro estudo, Litman, indica que existem três tipos gerais de indicadores de desempenho: medidas de qualidade de serviço, que refletem a qualidade do serviço experienciada pelos utilizadores; indicadores de resultados que refletem os resultados ou produtos; e indicadores de eficiência de custos, que refletem a razão entre custos e resultados.

Entre vários indicadores de qualidade de serviço, a fiabilidade do serviço continua a ser um dos aspetos de maior importância entre os passageiros de transportes públicos. Turnquist e Blume definem fiabilidade do serviço de trânsito como “a capacidade do sistema de trânsito de cumprir a programação ou manter intervalos regulares entre veículos e um tempo de viagem consistente” [22]. Um serviço sem estas características resulta em tempos de espera e de viagem superiores, e com isto pode dar-se a perda de passageiros, verificando-se o contrário com o aumento da fiabilidade. Operadores de transporte público desenvolveram vários indicadores para medir a fiabilidade do serviço, mas as três que se destacam são a pontualidade, a regularidade entre a passagem de veículos e o tempo de viagem. A performance da pontualidade das partidas, por exemplo, pode ser avaliada considerando a percentagem de veículos que partem e chegam ao destino a horas. Nakanishi, apresenta um indicador de pontualidade como a percentagem de viagens que partem de todas as paragens nos tempos planeados, não incluindo terminais, entre 0 e 5 minutos após o horário de partida programado [23, 24].

2.2.3.B BI no Mercado

Abordando soluções de BI direcionadas a empresas de transporte de passageiros, foi verificada [25–29] a crescente disponibilidade de sistemas BI com recursos de análise preditiva. A geração de modelos preditivos de métricas futuras com base em dados históricos para prever comportamentos e tendências por meio de análises estatísticas e aprendizagem, oferece às empresas a possibilidade de se prepararem para os desafios que se advinham.

Um ponto em comum em todos os sistemas de BI analisados, é o requerimento dinamismo e a interatividade da camada de apresentação, nomeadamente, nos relatórios. Há também uma promoção da automatização de todos os processos e sistemas, e do envio de notificações automáticas no caso de algum indicador chave de desempenho cair abaixo de um determinado valor.

As soluções podem recair num variado conjunto de domínios associados com o transporte de passageiros como:

- Vendas
- Monitorização financeira
- Gestão da qualidade
- Mão de obra
- Manutenção e gestão da frota
- Dados da atividade

Este último, é o domínio abordado por este projeto, englobando registos desde consumos energéticos ao cumprimento de horários por parte dos veículos.

Temos também o exemplo da empresa Cygnet [30], um ISV que fornece serviços a empresas que operam transportes públicos, entre outros, no Reino Unido. Elaborou uma solução de BI que permite um rastreio da informação em tempo real, para que assim seja possível implementar as melhores práticas para uma gestão ideal da frota. Entre os problemas que esta solução pretende resolver, estão o minimizar o tempo de espera, dar a capacidade de análise preditiva para tomar decisões sobre o número ideal de serviços necessários e de alertar e justificar atrasos nos mesmos. Para tal foi concebido um DW multi-dimensional com esquema em estrela, com a capacidade de recolher milhões de registos diários. Para o processo BI, recorreram ao seguinte conjunto de ferramentas da Microsoft: SSIS para integração, SQL Server Analysis Services (SSAS) para análise e SQL Server Reporting Services (SSRS) para a criação de relatórios. Para além disto foi desenvolvida uma solução web com ASP.NET e #C, que permite a visualização desta informação em tempo real a utilizadores não técnicos.

3

Design e Arquitetura

Conteúdo

3.1 Arquitetura Geral

A solução tem como base a criação de um Data Warehouse onde serão carregados dados relevantes extraídos das bases de dado de ajuda à exploração numa dada periodicidade. A extração de dados das bases de dados do SAEIP, será feita por via de views, a serem criadas, que reúnem os indicadores base das viagens se pretendem analisar. Entre estas duas bases de dados, existirá um processo de ETL, que irá filtrar e limpar os dados vindos do sistema fonte. A informação será organizada por período, num modelo de dados adaptado ao mecanismo de BI.

A segunda parte do modelo passará pela integração de uma ferramenta de BI, alimentada pelo DW, na plataforma web do XTraN Passenger. Esta ferramenta permitirá analisar e apresentar os dados de forma relevante aos end-users, fornecendo perspetivas que auxiliam a tomada de decisões informadas quanto ao negócio. Estas consultas e análises multidimensionais pretendem responder às questões que estão na base da definição das tabelas de factos do modelo, os dados serão disponibilizados em forma de relatórios.

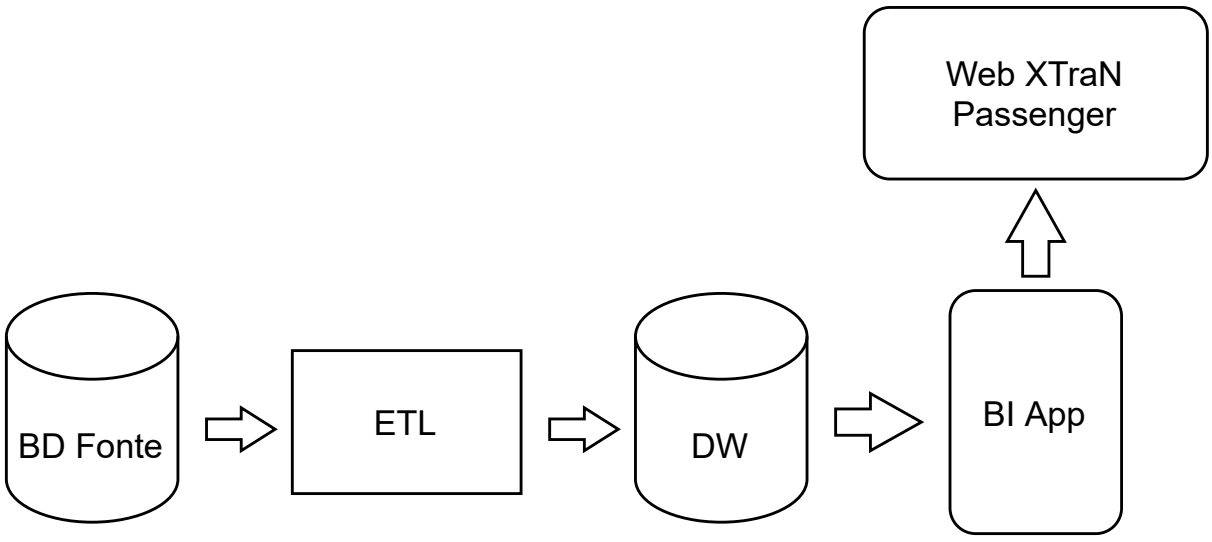

Figura 3.1: Arquitetura Geral do XTraN Passenger DW e BI

3.2 Fontes de Dados

As fontes de dados para este módulo, são internas, capturadas por sistemas operacionais da Tecmic, que ativamente registam informação referente às viagens realizadas pela frota. A estrutura da base de dados encontra-se desenvolvida em SQL Server.

O trabalho a ser feito nesta camada será a criação de views no sistema fonte, de modo a simplificar a extração aos dados que são considerados relevantes para o novo módulo. A fazer parte da informação destas views temos dados relativos à rede e planeamento e dados relativos à execução e respetivo cruzamento com o planeamento.

Na figura 3.2, podemos observar parte da estrutura dos sistemas fonte, a base de dados do SAEIP. Para salvaguardar os dados da Tecmic, os campos das tabelas encontram-se omitidos.

Diversos indicadores podem ser extraídos das bases de dados do SAEIP, que proporcionam uma visão quanto a qualidade do serviço, mas para o efeito deste projeto, os objetos de análise passam pelo controlo da realização e regularidade das carreiras e serviços, a sua pontualidade, o controlo da duração, tempo médio e velocidade dos mesmos. Transitam para o próximo passo, atributos que ajudam a definir estes indicadores.

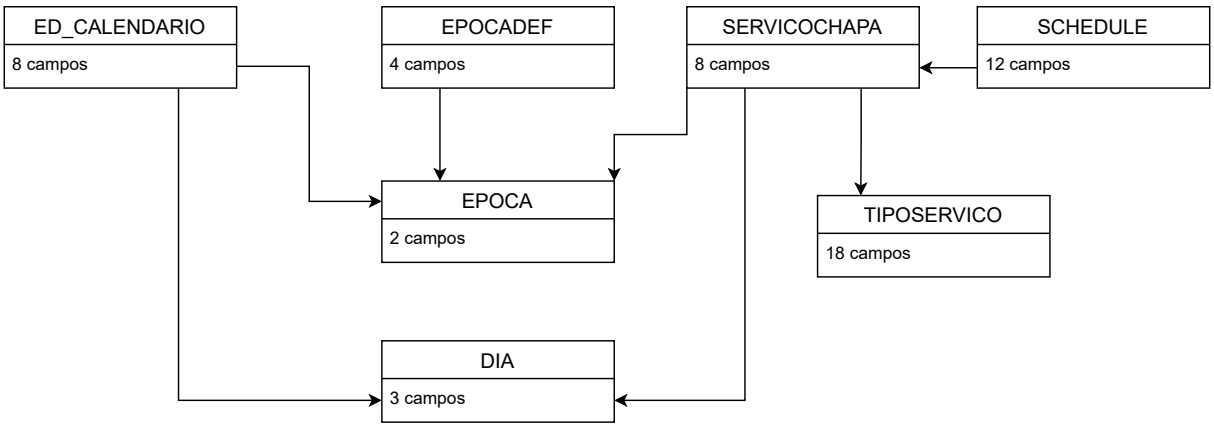

Figura 3.2: Exemplo de parte do schema da BD SAEIP

3.3 Data Warehouse

Os dados processados vindos do processo de ETL são armazenados num DW, atualizado em macro períodos. A sua estrutura permite armazenar dados históricos referentes à atividade.

Requerimentos
O primeiro passo deverá passar por definir os requerimentos do processo de BI. É essencial identificar as questões às quais se pretende responder com a ferramenta de BI, são estas que irão definir as granularidades da pesquisa e quais as tabelas de factos necessárias e as respetivas dimensões.

Configuração do ambiente físico
Tendo em conta o volume dos dados recolhidos e precavendo possíveis escalas ao sistema, terá de ser escolhido um ambiente robusto com suporte para tal. Para além do preço que advinha dessa solução, foi determinado que é importante proteger os dados da empresa, deste modo, expô-los num DW na cloud não seria uma opção desejável.

Por esses motivos, será utilizada um DW local, em SQL Server, que poderá ser acedido através do SQL Server Management Studio num computador pessoal, através de uma VPN, permitindo trabalho remoto ou presencial na empresa.

Definição do modelo dimensional
Para a definição do DW será necessária a criação de um modelo dimensional, que assenta nos seguintes:

- Factos: representa as métricas de desempenho decorrentes da atividade das frotas
- Dimensões: o contexto associado à atividade das frotas, utilizadas para analisar as métricas através de várias perspetivas.

Estes conceitos são convertidos em tabelas de facto e dimensão. As tabelas de dimensão servem para complementar os factos, valor calculados, sendo que é possível identificar algumas delas como os condutores, paragens, carreiras e veículos.

Os atributos das tabelas de dimensão formam hierarquias, que permitem a exploração dos factos a diferentes níveis de detalhe. As tabelas de facto são associadas a tabelas de dimensões através de primary e foreign keys. A solução para este DW passará idealmente pela criação de um esquema em estrela, como exemplifica a Figura 3.3.

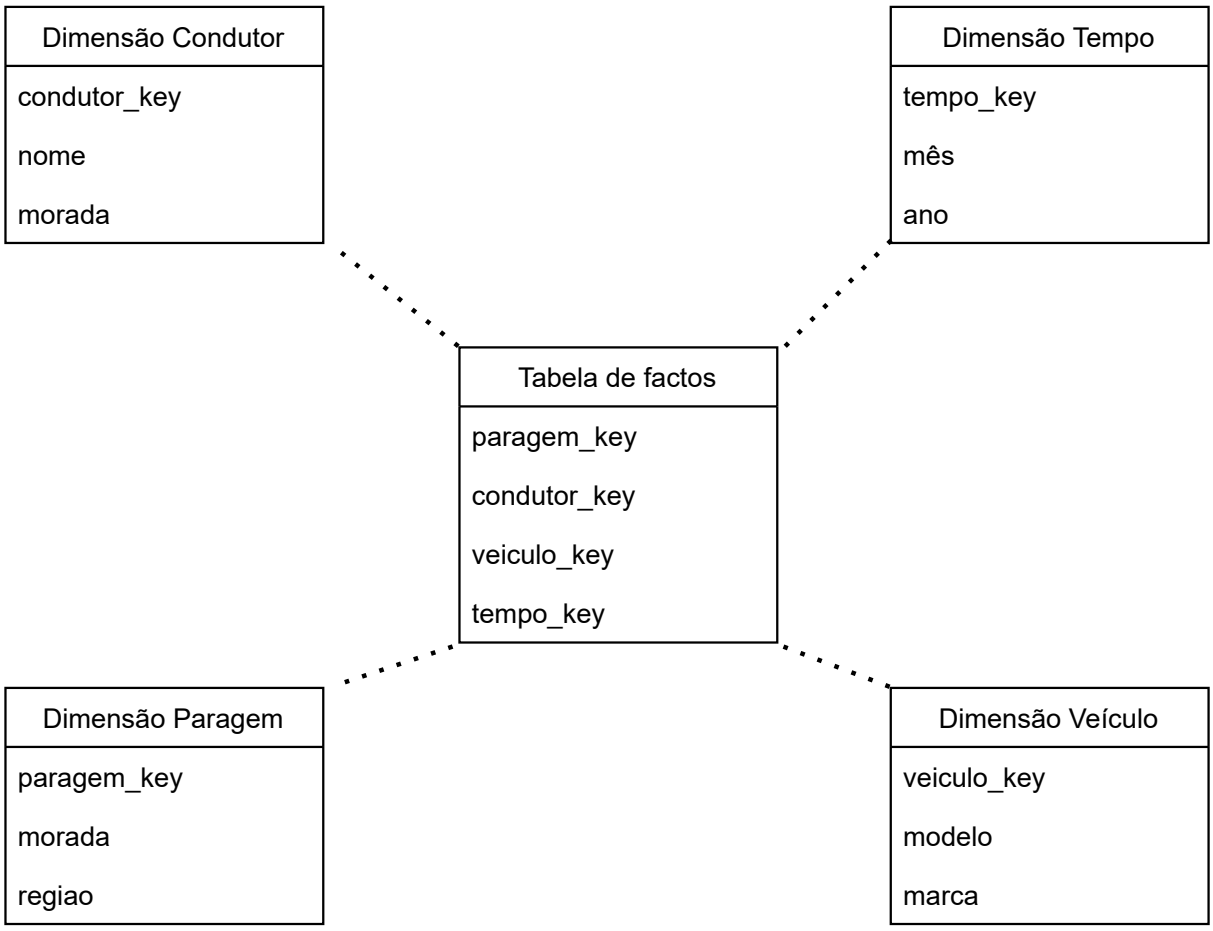

Figura 3.3: Exemplo de um esquema estrela

O esquema em estrela retira a informação da tabela de factos e divide-a em tabelas de dimensões desnormalizadas, levando a consultas em SQL mais simples e rápidas, necessário tendo em mente as queries complexas próprias das tarefas de análise.

Quanto à forma como serão alojados os dados dos clientes, cada empresa que dispõe do XTraN Passenger, terá uma instância do schema no DW. Será um sistema multi-inquilino. Aqui, a mesma infraestrutura, é partilhada por múltiplos clientes, ou seja, cada cliente tem um único schema e várias instâncias são executadas no mesmo servidor físico.

Tínhamos a opção de alocar um cliente por cada servidor, mas essa opção verificava-se excessivamente dispendiosa e pouco eficiente caso os servidores não fossem executados na capacidade total. Dentro das opções de multi-inquilino existem outras modalidades, como a de um único schema compartilhado, mas aí haveria um menor isolamento dos dados de cada uma das empresas. Assim, os dados estão no mesmo servidor, mas existe uma relativa segurança entre os dados dos clientes.

3.4 Camada BI

A última camada do modelo será a camada da criação e integração de relatórios sobre os dados, numa plataforma web, já existente, do XTraN Passenger. Esta plataforma, construída em .NET possuí outros módulos do XTraN Passenger referidos previamente. Como ferramenta de BI, vai ser utilizado o Power BI, produto da Microsoft. Os relatórios serão criados com o Power BI Desktop, e depois alojados no Report Server, solução on-premises, onde serão acedidos pelas empresas por meio de uma VPN.

Quanto ao conteúdo dos relatórios, das características anteriormente consideradas como determinantes da qualidade do serviço, vão ser avaliadas, entre outras, a frequência dos autocarros, tempo de viagem e a pontualidade. Estão propostos ser obtidos os seguintes relatórios:

- Relatório de Indicadores de Serviços
  - %Serviços Previstos
  - % Serviços Realizados
  - % Serviços Completos
  - % Serviços Incompletos
  - Tempo médio de duração de um serviço
- Relatório de Indicadores de Linha / Viagens
  - % Viagens Previstas / Realizadas
  - % Viagens Completas / Incompletas
  - Duração média das viagens
  - Desvio médio da duração das viagens
  - % Viagens duração real $>$ % duração planeada
  - % Viagens duração real $<$ % duração planeada
  - % Viagem com início real $<$ planeado
  - % Viagem com início real $>$ planeado
  - Velocidade Média
- Relatório de Indicadores de Pontos de Paragem
  - % Passagens / Linha
  - % Passagens / Hora
  - Regularidade Média / Linha
  - Tempo Médio Paragem

4

Implementação

Conteúdo

4.1 Ferramentas utilizadas

Para a criação do novo módulo DW e BI do XTraN Passenger foi utilizado um conjunto de ferramentas e software, enumerando: o SSMS, Visual Studio, SQL Server, Power BI, tSQLt.

Será usado o SQL Server Integration Services (SSIS), que desempenha o processo ETL, pela sua integração com todo o sistema Microsoft e pela facilidade de utilização, derivada da sua GUI. Para manipular qualquer infraestrutura SQL existente no projeto, foi utilizado o SSMS, quer para criar o DW, como para quaisquer consultas e manipulações que se verificaram necessárias no DW ou nas BD fontes. O Visual Studio foi usado em diferentes partes da criação: para criar e elaborar um pacote SSIS, sendo instalada a extensão SSIS, e para lidar com o desenvolvimento da parte web tendo com este aplicado a framework ASP.NET.

O servidor utilizado já extistia previamente nas instalações da Tecmic, funcionam em SQL Server. O servidor é uma máquina virtual com o processador Intel(R) Xeon(R) E5645 @ 2.40GHz e 20GB de RAM. Um servidor físico vem com um custo inicial mais elevado, mas o não pagamento de taxas mensais e principalmente a segurança que advém desta solução são essenciais para a empresa. Os relatórios foram desenvolvidos em Power BI Desktop, e depois carregados no Report Server. A criação de testes unitários foi feita com a framework tSQLt.

Primeiramente o uso destas ferramentas e trabalho foi desenvolvido num computador pessoal e só posteriormente transferido para um ambiente de qualidade na Tecmic. Para o controlo de versões foi usado o Github.

4.2 Escolha de atributos e criação views

Para a extração de dados da BDs fonte é importante a criação de views, de forma a expor os campos que existem dentro desta, de um modo simplificado em que a lógica das relações está desconstruída e pela barreira de segurança que estas conferem às tabelas da BD [31].

O primeiro passo foi perceber que atributos seriam retirados por estas views. Para isto foi necessário compreender como todo o sistema se relacionava, as dependências e relações que existiam entre os dados, e que atributos estavam presentes e em condições de ser utilizados. Como forma de filtrar os campos necessários, foi então feito um mapeamento entre os dados previamente pensados serem mostrados nos relatórios, com os campos da BD dos quais estes poderiam ser extraídos. Para além destes, foram adicionados campos que se visavam essenciais para a construção lógica de um DW.

Foram então retirados os seguintes campos das diferentes tabelas das BD fonte, sistematizados nas figuras 4.1 e 4.2.

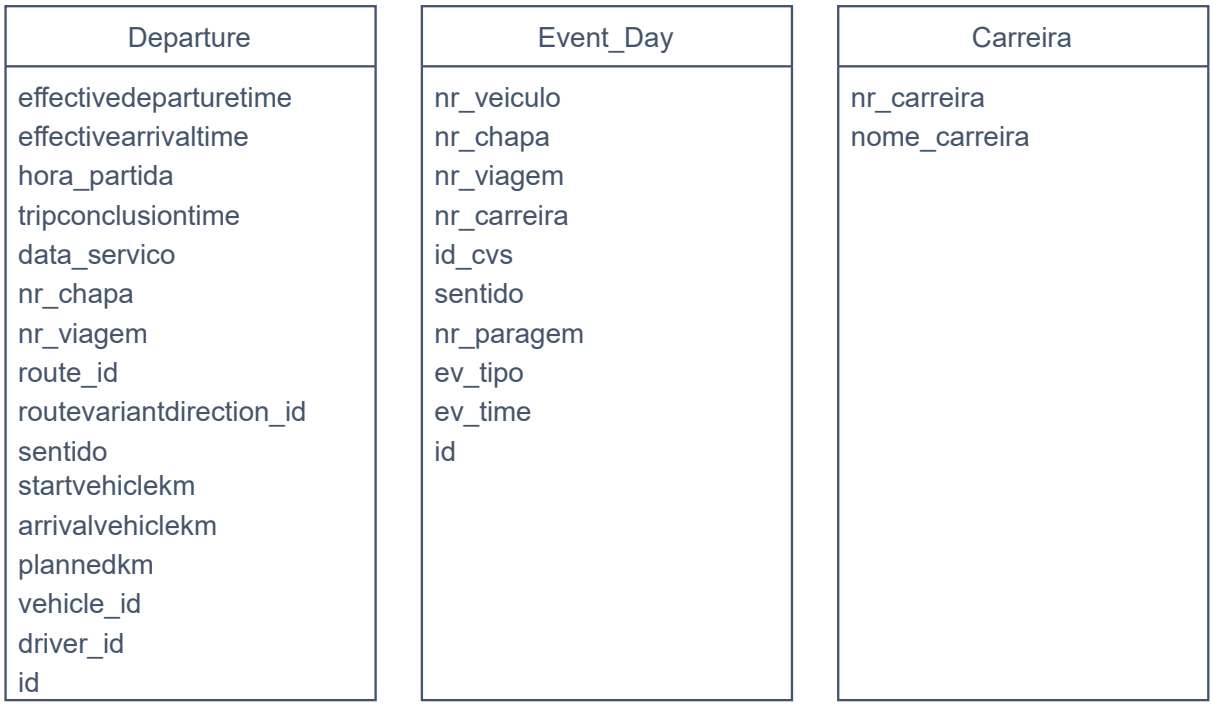

Figura 4.1: Campos retirados de Departure, Event_Day e Carreira

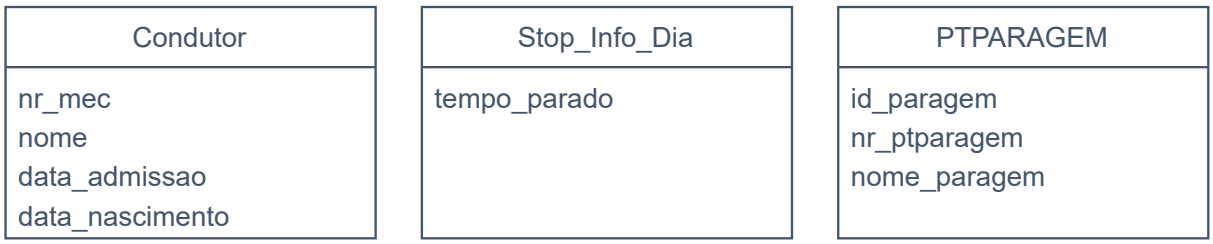

Figura 4.2: Campos retirados de Condutor, Stop_Info_Dia e PTPARAGEM

A tabela Event_Day agrega diferentes tipos de evento que ocorrem durante a operação dos transportes e são detetados pelos sistemas coletores a bordo. Os eventos são distinguidos uns dos outros pelo atributo EV_TIPO e dentro das variadas opções de evento, foram considerados os seguintes os mais relevantes para transitar para o DW:

- 05644 — Paragem
- 05646 — Excesso de velocidade
- 65536 — Excesso de aceleração
- 65537 — Excesso de travagem
- 65538 — Excesso de rotações

- 65540 — Excesso de tempo em ralenti
- 65543 — Excesso de aceleração lateral

Temos os eventos de paragem, que são imprescindíveis se se quiser fazer uma análise das viagens a este nível de detalhe e expor nos relatórios a distribuição do uso das paragens.

E têm-se eventos de excessos ocorridos durante a condução, maioritariamente atributos associados a acontecimentos menos desejáveis durante a condução. Além da performance temporal e cumprimento de horários é importante perceber um que condições está o serviço a ser realizado, uma análise destes atributos pode ajudar a diferentes perspetivas sobre a condução e desempenho do responsável pela condução.

Retiram-se de Departure atributos que identificam uma viagem, e métricas de exploração. Das restantes tabelas, são retirados atributos que ajudam a caracterizar as viagens, paragens, eventos e toda a sua envolvente.

Como apenas vão ser necessários dados que surgiram nas bases de dados desde a última carga, foi também criada uma view que estará situada no DW, que revela a data mais recente entre a última execução do processo ETL, ou um valor determinado como limite de meses a ser depositados no DW, apresentada em maior detalhe na secção 4.4.2.B.

4.3 Definição da Estrutura do Data Warehouse

No DW foi necessário delinear uma estrutura que armazene ambos os dados do planeamento e da execução das viagens. De notar, que para a definição da estrutura foi originalmente pensado ser criado um schema em estrela, a forma mais vantajosa de DW para uma eficiência nas pesquisas, que se caracteriza pela sua forma não normalizada e tabela de facto única.

4.3.1 Dimensões

As tabelas de dimensão fornecem um contexto descritivo às tabelas de facto, e contêm dados relativamente estáticos em relação a estas. Foram estabelecidas as seguintes tabelas de dimensão:

- Time
- Plate
- Route
- Driver
- Stop

Numa descrição breve das tabelas de dimensão temos: Time, onde são depositados dados temporais, agrega o ano, semestre, trimestre, mês e dia; Plate, tem o número da chapa e o número da viagem associada a essa chapa; Route, com o número interno da carreira, o número vulgar da carreira, o nome da carreira, a variante e a direção. Na tabela de dimensão Stop temos, o número de identificação interno Stop_Nr, Stop_Street_Nr com o número vulgar da paragem e Stop_Name com o nome desta. Existe ainda a tabela Driver, que armazena o número interno e o nome de um condutor de um veículo para além da sua data de nascimento e data de admissão na transportadora.

A combinação de dimensões, reunidas com factos, permite a resposta a pesquisas detalhadas. Juntando isto à forma hierárquica em que as dimensões foram pensadas, será possível visualizar os dados por vários prismas, com as agregações de dados em diferentes granularidades.

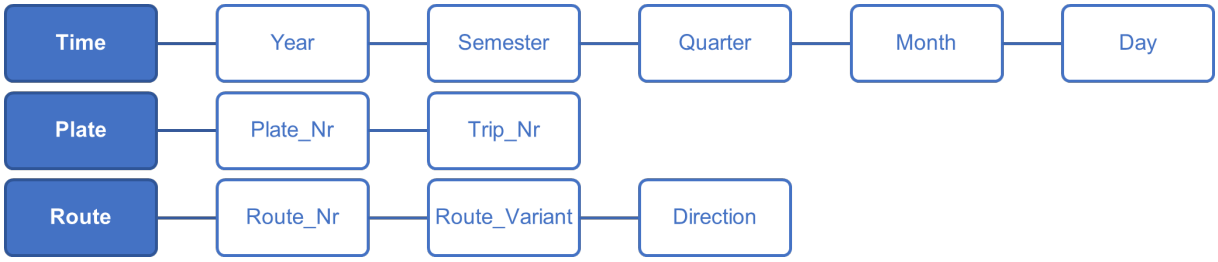

Figura 4.3: Hierarquias naturais do DW

Slowly Changing Dimensions

No âmbito de um DW existem certos atributos que estão suscetíveis a mudanças ao longo do tempo, estes são chamados de Slowly Changing Dimensions (SCD). No contexto do XTraN Passenger temos passíveis a uma mudança, por exemplo, nomes de locais identificativos numa cidade, assim como os números identificativos de objetos associados aos meios de transporte como número da carreira ou o número da paragem de autocarro. Assim, foram identificados entre as dimensões os seguintes atributos como SCD:

- Route_Name
- Route_Street_Nr
- Stop_Name
- Stop_Nr

Existindo diferentes alternativas para tratar uma SCD, foi colocada a questão de decidir se o DW iria armazenar apenas os dados atuais ou também dados históricos relativos a estes atributos. Numa abordagem resumida, foram consideradas 2 metodologias que poderiam ser usadas para este projeto, SCD tipo 1 e tipo 2. Uma SCD Tipo 1 não armazena o histórico da dimensão, o valor dos registos é substituído e assumido o novo valor. Uma SCD tipo 2 é adicionado um novo registo com o novo valor, e o registo antigo é marcado como inativo. Na ótica do que vai ser a utilização dos relatórios do XTraN Passenger, e quais são os valores a observar, foi considerado que faz mais sentido considerar os atributos referidos SCDs Tipo 1, em que os valores antigos são substituídos, na medida em que faz sentido apresentar apenas os valores atualizados das instâncias de qualquer um destes atributos, mesmo que no passado tenham tido outros valores, além do mais, numa SCD Tipo 2 seria investida uma complexidade superior.

Surrogate Keys

As tabelas de dimensão contêm uma coluna considerada como chave primária, que ajuda a identificar cada registo como único na dimensão. Quando chegou a altura de atribuir chaves primárias às tabelas, levantou-se a questão de usar chaves naturais ou substitutas, surrogate. Seria possível utilizar chaves naturais na maioria das dimensões, provenientes das bases de dado fonte, evitava-se a complexidade acrescentada de uma tabela extra e teriam significado intrínseco [32]. No entanto, nada pode assegurar que estas chaves não sofram alterações na fonte, o que o próprio sistema fonte não mudar ou seja fundido com outro a dado momento, provocando assim inconsistências a nível das chaves naturais. Deste modo, embora haja alguns inconvenientes, foi decidido manter independência face às chaves naturais das BD fonte e usar chaves surrogate, sem significado de negócio, usadas exclusivamente para identificar os registos no DW, com inteiros sequenciais para todas as dimensões.

4.3.2 Tabelas de facto

Na linha de pensamento para delinear a estrutura deste DW, teria imperativamente de existir uma estrutura que armazenasse os dados relativos a cada viagem, neste estavam incluídas métricas que ocorriam apenas uma vez, como a hora de partida e os quilómetros percorridos, bem como eventos que potencialmente ocorriam múltiplas vezes ao longo de uma viagem: como a passagem numa paragem e a deteção de excesso de velocidade. Estes últimos com uma menor granularidade. Como em cada viagem existem múltiplas paragens, e cada paragem tem associado a si diversos elementos, verificou-se que seria desajustado associá-los à tabela FactTrip. Foi estabelecido que existiriam duas tabelas de facto, uma com os eventos de ocorrência única numa viagem e de ocorrência múltipla, mas sumarizáveis, e outra, a FactStop, dedicada exclusivamente a eventos de paragem.

Sendo deparados com o constrangimento de ter diferentes níveis de granularidade nas métricas, a desejada, ideal e otimizada solução de ter um esquema em estrela, já não é viável, e temos então um Fact Constellation schema, onde se têm várias tabelas de facto, neste caso duas, uma para as viagens e outra para as paragens, com dimensões partilhadas entre estas. Time, Plate e Route são as tabelas de dimensão fundamentais para a identificação de uma viagem, quer na tabela FactTrip como na FactStop.

Em relação à estrutura usada para armazenar viagens temos a FactTrip, que agrega ambas métricas de planeamento e execução, entre estas temos:

- Tempos de chegada e partida
- Quilómetros percorridos e planeados
- Número de vezes que o veículo entrou em excesso de velocidade
- Número de vezes que o veículo acelerou em excesso
- Número de vezes que o veículo travou em excesso
- Número de vezes que foi reportada aceleração lateral em excesso
- Número de vezes que foi reportado tempo em ralenti excessivo
- Número de vezes que foi reportado um número de rotações excessivas

Na FactStop, tabela que armazena dados sobre cada uma das paragens que os autocarros fazem em cada viagem, temos armazenadas foreign keys semelhantes às da FactTrip correspondente,TimeID, RouteID e PlateID, e uma StopID, que identifica a paragem. Deste modo é possível estabelecer correspondência entre uma instância de FactStop e FactTrip. Como métricas desta tabela temos Event_Time, o tempo exato da ocorrência do evento de passagem pela paragem e Stopped_Time o tempo em segundos que o veículo lá esteve parado.

Na BD fonte havia alguma incongruência quanto à nomenclatura dos campos, estando uns listados em português outros em inglês, uns em letra maiúscula, outros em minúscula. É reportado que em projetos onde não existem convenções de nomenclatura, existem mais situações de equívocos derivadas desta decisão, isto leva a atrasos e a uma menor produtividade da equipa em certos momentos. Deste modo, todas as incongruências que existiam na BD fonte foram uniformizadas: temos nomes em inglês e maiúsculas apenas em cada início de palavra.

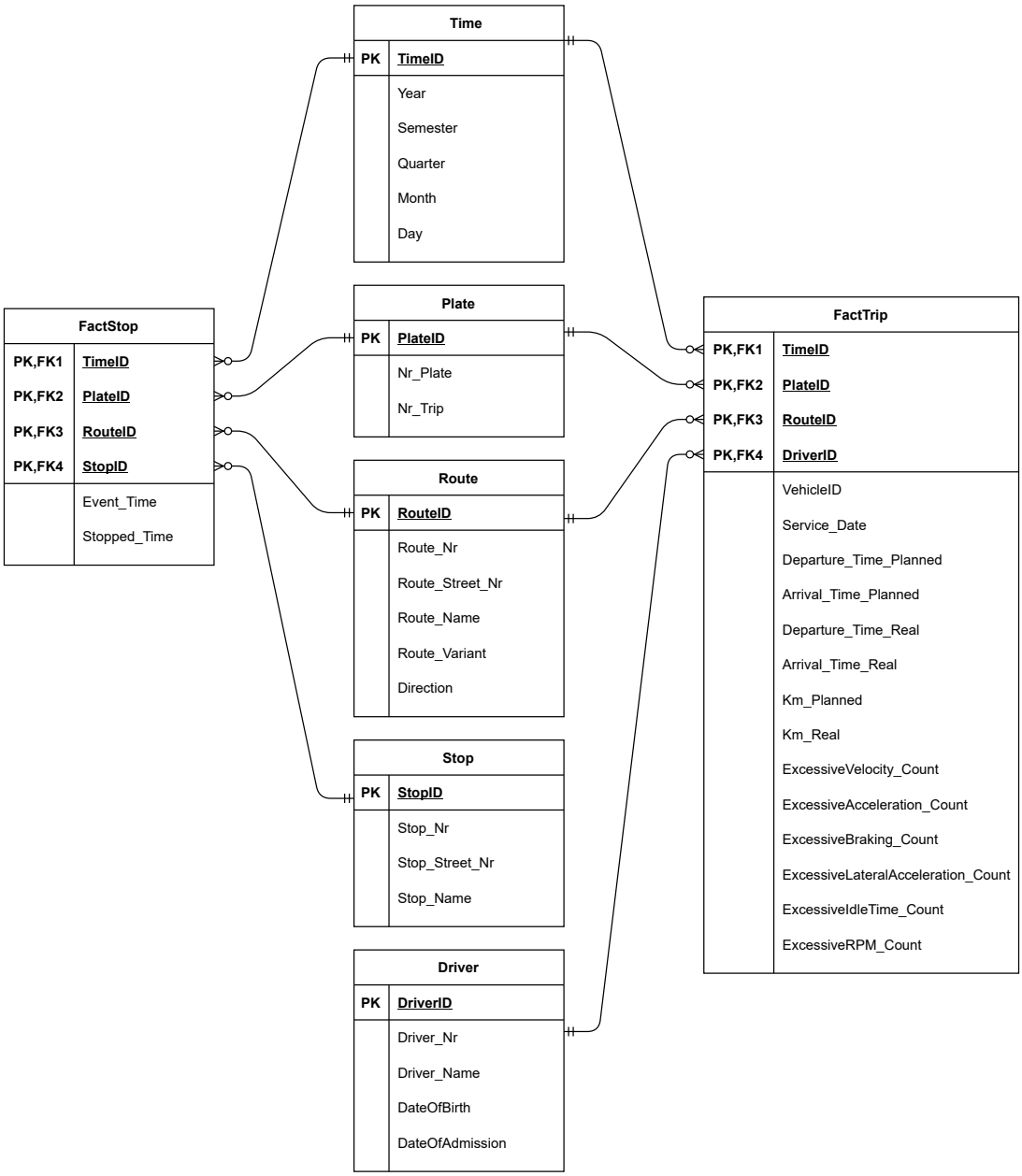

Figura 4.4: Schema do DW

4.3.3 Métricas derivadas por Views

Para além das métricas, que conseguiam ser extraídas das bases de dados fonte, existem métricas úteis para os relatórios, que são derivadas a partir de atributos presentes nas das tabelas de facto. Seria um desaproveitamento do espaço do DW calculá-las e armazená-las. Numa forma de economizar este espaço, o DW terá associadas views, que calculam estas medidas.

Os referidos atributos são os seguintes:

- Duração planeada
- Duração real
- Velocidade
- Viagem realizada
- Viagem completa
- Viagem com início pontual
- Viagem com duração esperada

Temos 3 atributos de medidas físicas, como a duração e velocidade, resultantes de fórmulas, calculados na view *DURATIONS_VIEW* apresentados na tabela 4.1, e quatro métricas booleanas, que identificam estados de cada uma das viagens, através do verificar da performance de acontecimentos, atributos temporais, a sua derivação está representada na tabela 4.2 de uma forma sintetizada e temos o código da view *DERIVATES_VIEW* na Listagem 4.1

	Formula
Duration_Planned	Arrival_Time_Planned - Departure_Time_Planned
Duration_Real	Arrival_Time_Real - Departure_Time_Real
Velocity	If (Duration_Real!= 0): Km_Real/Duration_Real

Tabela 4.1: Fórmula de cálculo de duration_planned, duration_real e velocity

	NULL	0
Held	-	Departure_Time_Real IS NULL and Arrival_Time_Real IS NULL
Concluded	Held=0	Arrival_Time_Real IS NULL
DepartureOnTime	Held=0 or Departure_Time_Planned IS NULL or Departure_Time_Real IS NULL	Departure_Time_Real - Departure_Time_Planned <60 segundos
ExpectedDuration	Held=0 or Duration_Planned IS NULL or Duration_Real IS NULL	Duration_Planned - Duration_Real <0 minutos

Tabela 4.2: Condições de atribuição do valor NULL e 0 aos atributos Held, Concluded, DepartureOnTime e ExpectedDuration

Listagem 4.1: Código de criação da DERIVATES_VIEW

```
1  CREATE OR ALTER VIEW DERIVATES_VIEW AS
2  SELECT
3  TimeID, PlateID, RouteID, Service_Date,
4  Duration_Planned, Duration_Real, Held,
5  CASE WHEN Held=0 THEN NULL
6       WHEN Arrival_Time_Real IS NULL THEN 0
7       ELSE 1 END as Concluded,
8  CASE WHEN (Held=0 or Departure_Time_Planned IS NULL
9       or Departure_Time_Real IS NULL) THEN NULL
10      WHEN DATEDIFF("s",Departure_Time_Planned,Departure_Time_Real) > 60 THEN 0
11      ELSE 1 END as DepartureOnTime,
12 CASE WHEN (Held=0 or Duration_Planned IS NULL or Duration_Real IS NULL)
13      THEN  NULL
14      WHEN Duration_Planned- Duration_Real < 0 THEN 0
15      ELSE 1 END as ExpectedDuration,
16 CASE WHEN (Duration_Real=0) THEN  NULL
17      ELSE Km_Real/CONVERT(NUMERIC(18, 2), Duration_Real/ 60
18      + (Duration_Real % 60) / 100.0) END as Velocity
19 FROM DURATIONS_VIEW;
```

É atribuído o valor **1** nas restantes situações. Numa linguagem corrente, uma viagem é realizada caso haja registo de partida ou chegada, concluída caso haja registo de chegada. Uma viagem com início pontual, assim o será caso parta dentro de 1 minuto do tempo de partida planeado e será considerado ter a duração esperada caso não ultrapasse 1 minuto da duração planeada.

O DW é materializado no SQL Server, com servidores localizados nas instalações da Tecmic. Para criar e manipular este DW foi utilizado o SQL Server Management Studio (SSMS).

4.3.4 Paralelismo BD e DW

Como para sumarizar o paralelismo entre os campos da BD fonte e o DW segue-se a tabela 4.3. À esquerda temos tabelas da BD e à direita as tabelas do DW que derivam das primeiras. O nome das tabelas tem uma cor de preenchimento, os campos não.

Departure	**Time**
hora_partida	Year
	Semester
	Quarter
	Month
	Day
Departure | Carreira	**Route**
route_id	Route_Nr
nr_carreira	Route_Street_Nr
nome_carreira	Route_Name
routevariantdirection_id	Route_Variant
sentido	Direction
PTPARAGEM	**Stop**
id_paragem	Stop_Nr
nr_ptparagem	Stop_Street_Nr
nome_paragem	Stop_Name
Departure	**Plate**
nr_chapa	Plate_Nr
nr_viagem	Trip_Nr
Condutor	**Driver**
nr_mec	Driver_Nr
nome	Driver_Name
data_nascimento	DateOfBirth
data_admissao	DateOfAdmission
Departure | EVENT_DAY	**Fact Trip**
hora_partida	Departure_Time_Planned
tripconclusiontime	Arrival_Time_Planned
effectivedeparturetime	Departure_Time_Real
effectiveconclusiontime	Arrival_Time_Planned
vehicle_Id	Vehicle_Id
data_servico	Service_Date
ev_tipo (contagem)	ExcessiveVelocity_Count
	ExcessiveAcceleration_Count
	ExcessiveBraking_Count
	ExcessiveLateralAcceleration_Count
	ExcessiveIdleTime_Count
	ExcessiveRPM_Count
plannedkm	Km_Planned
startvehiclekm	Km_Real
arrivalvehiclekm	
EVENT_DAY | STOP_INFO_DIA	**Fact Stop**
ev_time	Event_Time
tempo_parado	Stopped_Time

Tabela 4.3: Tabela de paralelismo entre os campos da base de dados fonte e o DW.

4.4 Processo ETL

Para desempenhar o processo ETL foi usado o SSIS como pacote no Visual Studio 2019. De uma forma simplificada temos como arquitetura essencial do SSIS os seguintes componentes [33]:

- Control Flow: É o centro das operações, onde são introduzidas, numa certa ordem, as tarefas a executar
- Data Flow: Componente onde os dados são retirados das fontes, se transformam e inserem no destino
- Event Handler: Proporciona uma maneira de tratar determinados eventos que ocorram

4.4.1 Control Flow

O processo de carregamento do schema base requer que se carreguem tabelas de facto e de dimensão. As tabelas de facto contêm *foreign keys* que apontam para as tabelas de dimensão. Desta forma, é importante que as tabelas de dimensão sejam preenchidas em primeiro lugar. Por uma questão de capacidade de manutenção, as dimensões são processadas de forma separada, havendo uma tarefa para cada uma delas. Uma única tarefa que atualizava múltiplas tabelas poderia ser difícil de gerir, à medida que o DW cresce. Foi então criado um fluxo que trata os dados que compõem, pela seguinte ordem, as tabelas de dimensão Time, Route, Plate, Stop e Driver, e por fim as tabelas de facto: FactTrip e FactStop. Como último elemento do control flow, temos uma tarefa de SQL, que insere a data atual numa tabela denominada por LastRun.

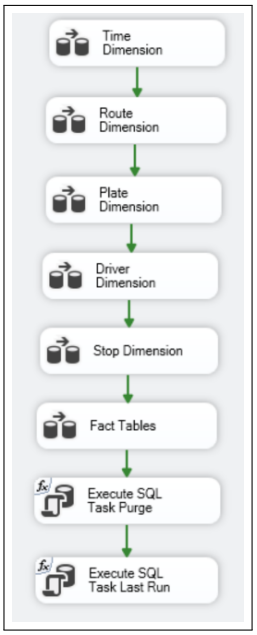

Figura 4.5: Control Flow do SSIS

4.4.2 Data Flow

4.4.2.A SQL Task Last Run

Começamos por introduzir a última tarefa do control flow, pois é fundamental para compreender algumas transformações feitas nas restantes tarefas em todo o processo. Falámos anteriormente no capítulo de criação de views na fonte, que apenas era necessário extrair da BD dados relativos a eventos que ocorreram após a última vez que o DW foi carregado. Para saber a data dessa ocorrência, é necessário que cada vez que o script corra com sucesso, esta seja registada. Foi criada no DW uma tabela denominada LastRun, com uma única coluna do tipo DATETIME, onde através de uma tarefa SQL de inserção é armazenada a data atual, no fim de um *job* de ETL bem-sucedido. A primeira tarefa do próximo *job*, será inserir esta data na, ou nas BD fonte.

4.4.2.B Dimensões

O preenchimento das tabelas de dimensão segue um procedimento base relativamente semelhante para todas elas. Como podemos ver na Figura 4.6, no passo *OLE DB Source*, são extraídos os dados da BD fonte através de uma view. Nesta view temos 2 fontes, os registos da atividade e de *LAST_RUN_EXTRACTION_VIEW*. Desta última é retirada a mais recente das datas: a mais recente execução do pacote ou menos 6 meses da data atual, utilizando a seguinte query 4.2.

Listagem 4.2: Query usada para obter a data a partir da qual vão ser extraídos registos

```
1 SELECT (CASE WHEN (SELECT MAX(Date) FROM LastRun) > DATEADD(month,-6, GETDATE())
2 THEN (SELECT MAX(Date) FROM LastRun)
3 ELSE DATEADD(month, -6,GETDATE()) end) AS date
```

Pelas mais variadas razões, o script de ETL poderá estar um extenso período sem correr, caso isto aconteça, apenas serão extraídos dados relativos aos mais recentes 6 meses. Dito de outra forma, um máximo de 6 meses de nova informação será armazenada no DW durante um carregamento.

Com esta data limite é imposta, na view usada em *OLE DB Source*, a condição data registo $>$ data limite, e apenas registos que a satisfaçam seguem em frente. Deste modo assegura-se que não há registos antigos, repetidos ou descartáveis a serem processados. São removidos os duplicados e é feito um lookup, que determina se já há instâncias semelhantes no DW. No fim de os dados estarem uniformizados e com a nomenclatura correta, são mapeados e inseridos nos campos respetivos do DW. Em situação de erro, serão tomadas precauções abordadas mais à frente.

Seguidamente são descritas algumas anotações e particularidades do modo de tratamento de cada uma das dimensões, não abordando, embora presentes, os procedimentos base descritos acima.

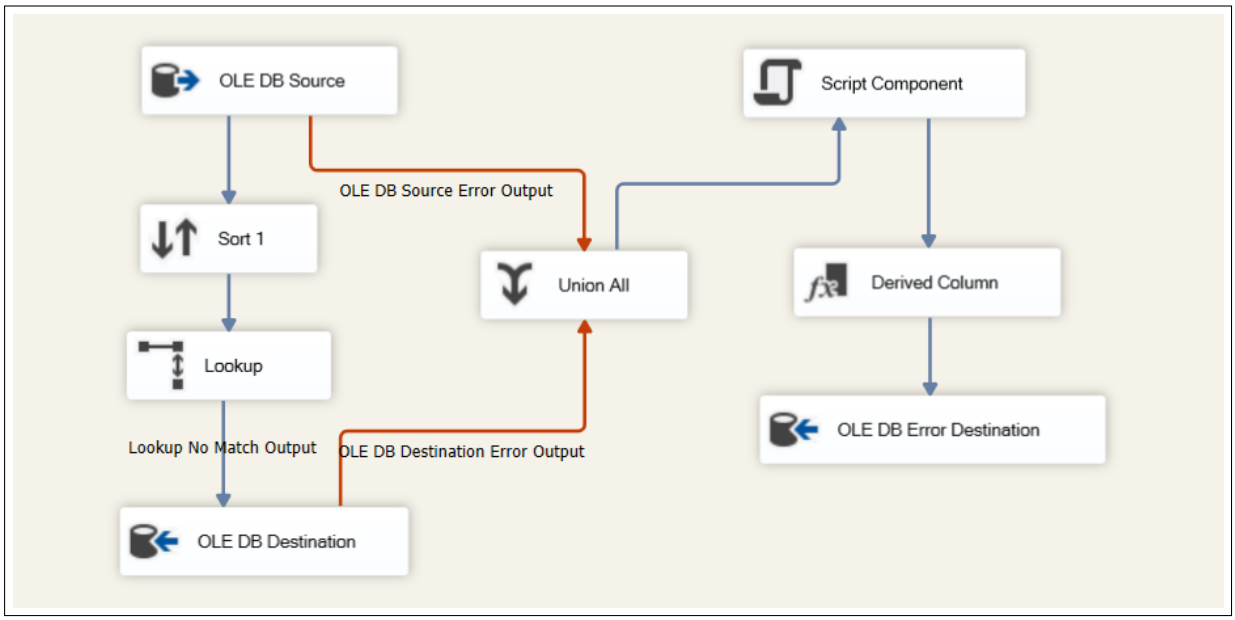

Figura 4.6: No SSIS, processo base para preencher uma tabela de dimensão

Time: Numa descrição mais detalhada, os dados da tabela de dimensão Time foram derivados do atributo HORA_PARTIDA. Neste estão contidos todos os dias em que ocorreu atividade. O ano, trimestre, mês e dia foram obtidos através de funções pré-existentes. O semestre calculado com uma operação ternária.

Route: Em Route, temos como já foi referido uma Slowly Changing Dimension. São consideradas business keys, Route_Nr, Route_Variant e Direction e changing attributes Route_Name e Route_Street_Nr.

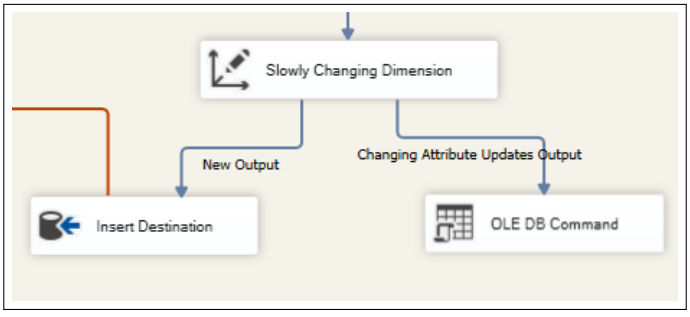

Figura 4.7: No SSIS, procedimento utilizado numa SCD

Plate: No lookup foram usados os atributos Plate_Nr e Trip_Nr.

Stop: A tabela Stop também é uma SCD, tem como business key Stop_Nr e changing attributes

Stop_Street_Nr e Stop_Name.

Driver: O preenchimento da tabela Driver segue um procedimento idêntico ao descrito como base. Foi considerado o Driver_Nr como atributo para o lookup.

4.4.2.C Factos

Nesta tarefa são preenchidas as tabelas FactTrip e FactStop. Não sendo viável demonstrar visualmente o script criado em SSIS, serão apresentados trechos do mesmo. Temos duas fontes de dados, uma com dados relativos a cada viagem e outra com dados relativos a eventos que ocorreram em cada viagem, duas tabelas facto e duas tabelas de destino para incoerências. Assim como nas tabelas de dimensão, todos os dados retirados das fontes estão sujeitos ao requisito de serem mais recentes do que a última vez que o *job* correu ou terem menos de 6 meses. Vamos ter o tratamento de dados de duas origens a ocorrer em paralelo, eventos retirados de *OLE DB Source Events* e viagens de *OLE DB Source Trips*.

De um dos lados, são retirados da fonte, pela EVENT_EXTRACTION_VIEW dados relacionados com eventos detetados pelas consolas de bordo, bem como dados que permitem identificar a viagem em que estes ocorreram. São feitos uma série de lookups entre as tabelas de dimensão já criadas e os dados da viagem para que possa ser feita uma correspondência entre estes e os campos ID dos atributos das tabelas de dimensão.

Depois desta correspondência ser estabelecida, surge uma tarefa de divisão condicional, onde cada um dos eventos será desassociado dos restantes, com recurso ao campo EV_TIPO. Os eventos de deteção de paragem em particular, um evento de código 5644, seguem para um lookup com a tabela de dimensão Stop e estando todos os componentes mapeados, são carregados na FactStop.

Se o evento em causa não for uma deteção de paragem, segue para as tarefas representadas na Figura 4.8. Com esta sequência de tarefas, é apurado o número de vezes que cada evento de excesso de velocidade, aceleração, travagens, aceleração lateral, tempo em ralenti ou rotações ocorre em cada viagem. O número de ocorrências de cada evento é computado através de uma função de agregação count, para cada agrupamento de TimeID, PlateID e RouteID, campos identificadores base de uma viagem.

Do outro lado, o percurso que os dados fazem ao sair de *OLE DB Departures Source*, é semelhante ao primeiro. É feita uma série de *lookups* para descobrir as *foreign keys* associadas à viagem, que a

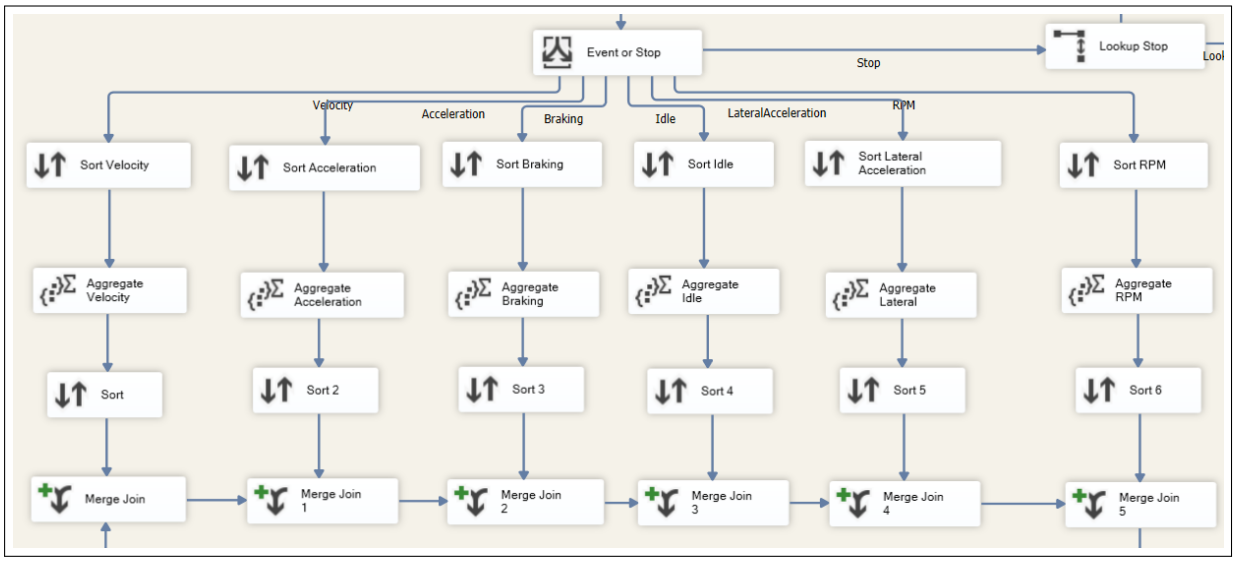

Figura 4.8: No SSIS, contagem do número de ocorrências de eventos e associação com a respetiva viagem

ela se agregam, são derivadas novas colunas e, entretanto, o número de ocorrências de cada evento de excesso ocorridas durante uma viagem, como podemos ver computado na figura 4.8 é finalmente associado a esta, numa série de *merge joins*. Paralelamente, são aplicadas proteções para captar erros e impostas restrições aos dados do fluxo, que quando não são satisfeitas, estes mesmos dados são considerados defeituosos e enviados para tabelas de erro, assuntos abordados posteriormente.

4.5 Tratamento de Erros e Incoerências

No processo ETL existem diversos pontos que são expostos a possíveis falhas, denunciadas pela imprevisibilidade dos dados, estas tanto se podem manifestar durante o próprio, em falhas no sistema, como apenas mais tarde, em dados aparentemente corretos, mas com falhas na sua lógica. Tendo isto em consideração, foi feita uma divisão destes problemas em 3 categorias:

- Incoerências
- Erros não fatais
- Erros fatais

Neste contexto, uma incoerência não põe em causa a continuidade do processo ETL, e se não for acautelada, pode nunca chegar a ser encontrada. Um erro não fatal, é apanhado pelo sistema, tratado e o processo continua. Um erro fatal, não é apanhado e acaba com o processo.

4.5.1 Incoerências

Pelas mais variadas razões, existem valores coletados na fonte que se verificam incoerentes com a realidade, como valores temporais discordantes ou valores negativos. Seria por tanto contraproducente colocar estes valores erróneos no DW, que viriam a deturpar os valores obtidos nos relatórios. Deste modo, os valores que não estão de acordo com parâmetros requeridos, serão colocados em tabelas à parte. A análise posterior destes valores pode dar à empresa indicações sobre possíveis erros que estão a ocorrer nos sistemas de coleção e nas bases de dados fonte.

Foram criadas duas tabelas para este tipo de incoerências, *IncoherenceTrip* e *IncoherenceEvent*, a primeira para colocar registos de viagens com valores incoerentes, e a segunda, de eventos incoerentes. A sua estrutura segue o esquema de uma *FactTrip* e *FactStop* alternativa, com as típicas métricas e para além disso, as colunas OriginalRowID para identificar a tabela de origem do registo com a incoerência, e ErrorDescription, onde está assinalado o problema.

Os valores definidos como limites para o erro foram discutidos com a equipa, abaixo segue a lista de onde se podem encontrar estas incoerências:

- Planned vs Real, caso a diferença entre a hora de partida ou chegada, planeada e real seja superior a 24 horas e vice-versa.
- Departure vs Arrival, caso a hora de partida seja posterior à chegada ou caso a hora de partida e chegada difiram por mais de 8 horas
- Event Count, caso alguma das contagens de eventos de excesso for superior a 100

- Km, caso os quilómetros planeados ou reais a percorrer na viagem seja superior a 500 km ou negativo
- Stop Time, caso o tempo de paragem do veículo seja negativo ou superior a 10 minutos
- Eventos sem viagem associada
- Registos de viagens sem campos de dimensão associados

Após estas incoerências serem detetadas, as viagens ou eventos a quais estes pertencem, seguem uma rota diferente das restantes, é dado um rótulo à incoerência e é inserida na respetiva tabela. Na figura 4.9, podemos observar o exemplo de registos associados a uma tabela *IncoherenceTrip*, onde temos a indicação da incoerência, a linha do registo na tabela na fonte, *departure* neste caso, e as métricas associadas à viagem, onde entre elas, aparecerá um valor incongruente, denunciado previamente pelo campo *IncoherenceDescription*.

	IncoherenceDescription	OriginalRowID	Departure_Time_Real	Arrival_Time_Real	Km_Real	Exces…
1	Planned vs Real	19	2021-08-11 13:24:10.000	2021-08-13 15:04:13.000	25	4
2	Km	14	2021-08-11 16:41:21.000	2021-08-11 17:19:43.000	-5	2
3	Departure vs Arrival	17	2021-08-11 15:45:30.000	2021-08-11 10:04:43.000	19	1
4	Event Count	25	2021-08-11 12:34:13.000	2021-08-11 13:03:25.000	17	149
5	Km	34	2021-08-11 14:50:24.000	2021-08-11 15:29:46.000	999	2

Figura 4.9: Parte da tabela IncoherenceTrip, com alguns registos de incoerências

4.5.2 Erros

Quando corremos o pacote SSIS, é possível que, pelas mais diversas razões, erros possam ocorrer. Desta forma é boa prática haver mecanismos de tratamento de erros no pacote. Dependente da gravidade, erros em SSIS podem ser classificados em três grupos, e cada um é capturado, tratado e reportado de uma forma diferente:

1. Informar e Passar — É tratado ao nível do componente
2. Informar e Falhar — A informação do erro passa para o Event Handler
3. Falhar — Pacote não é validado

4.5.2.A Informar e Passar

Em informar e passar temos erros que embora devam ser comunicados, não devem interromper o processo ETL, como por exemplo eventos de truncagem ou incoerência no tipo de dados. Ao serem tratados ao nível do componente, tornam-se erros não fatais, que são salvaguardados e o processo segue em frente intacto.

O fluxo de cada um dos componentes é normalmente dividido em dois possíveis outputs, o output normal e do erro. Quando ocorre um erro, o SSIS oferece 3 opções: *fail component*, *ignore error*, ou *redirect row*. Em locais do script mais suscetíveis ao erro é escolhida a opção *redirect row*, onde os registos da linha em que o erro ocorreu seguem um caminho alternativo. Com o propósito de establecer um destino para estes registos, foi criada no DW uma tabela de nome *ErrorHandler*, com os seguintes parâmetros que podemos observar na Figura 4.10.

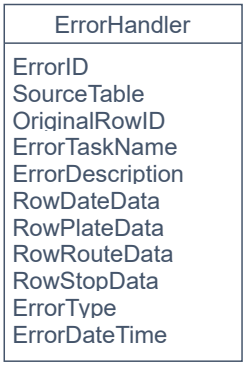

Figura 4.10: Campos da tabela ErrorHandler

Desta forma, são colocados na tabela os parâmetros identificativos da viagem ou evento juntamente com o erro, de modo a que seja possível encontrar a sua origem nas tabelas fonte, e possivelmente avaliar ou corrigir o problema associado. Temos SourceTable e OriginalRowID como forma de rastrear a origem do erro nas tabelas fonte. Em ErrorTaskName está o nome da tarefa do *control flow* do pacote SSIS onde ocorre o erro, e em ErrorDescription foi agregada a informação do passo onde ocorre o erro e uma descrição deste, que apareceria na consola, recolhidos através de um script. Temos também 4 colunas, de nomenclatura padrão "*Row* + Nome da dimensão de origem + *Data*", que aludem aos atributos base para identificar uma viagem ou evento, onde é colocada em *String* informação referente ao elemento, para um mais fácil rastreamento. Em ErrorType temos se o erro é fatal ou não. Neste tipo de erro, Informar e Passar, é registado como "non-fatal". Por fim a data em que ocorre o erro é colocada em ErrorDateTime, recolhida através de um getDate().

Esta tabela é preenchida com a sequência de passos, que pode ser vista na Figura 4.11, metadados do erro como a descrição do mesmo e a coluna onde ocorreu, são recolhidos através de um componente de script [34], exposto em 4.3. De seguida são derivadas novas colunas de campos como a tarefa onde ocorreu, tipo, tabela de origem do erro. Na Figura 4.12 podemos ver a disposição de registos na tabela *ErrorHandler*.

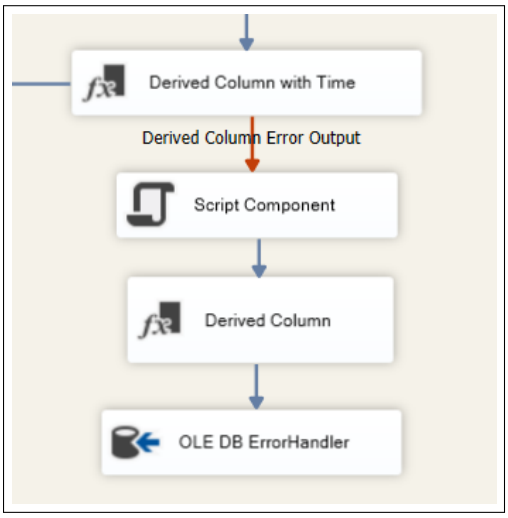

Figura 4.11: Procedimento no pacote SSIS em caso de erro

Listagem 4.3: Script em C# usado para recolher informação quanto ao erro

```
1   public override void Input0_ProcessInputRow(Input0Buffer Row)
2     {
3         Row.ErrorDescription =
4         this.ComponentMetaData.GetErrorDescription(Row.ErrorCode);
5         IDTSComponentMetaData130 componentMetaData =
6         this.ComponentMetaData as IDTSComponentMetaData130;
7         Row.ErrorColumnName =
8         componentMetaData.GetIdentificationStringByID(Row.ErrorColumn);
9     }
```

As capturas, vulgarmente conhecidas como catches, deste tipo de erros foram colocadas nos sítios considerados ser os mais vulneráveis a falhas: Extração e inserção de dados, *lookups* e cálculo de novas colunas.

4.5.2.B Informar e Falhar

Quando ocorre um erro que não está salvaguardado como os anteriores, e falha o pacote, não deixando o Data Flow terminar, entra em funcionamento o Event Handler.

O Event Handler é um componente do SSIS, que permite capturar e tratar eventos. Durante a execução de um pacote, cada tarefa levanta eventos à medida que corre, com o Event Handler podemos programar o tratamento de eventos específicos através da criação de workflows que são executados quando esses eventos relevantes são detetados.

Neste âmbito, após um evento de erro, este é direcionado para o Event Handler. Aqui, através de uma tarefa de SQL, o erro será registado e detalhado na tabela já anteriormente apresentada, *ErrorHandler*, desta vez tendo no campo ErrorType, o rótulo de falha "fatal". Tendo o erro uma origem desconhecida, não é possível preencher as colunas SourceTable e OriginalRowID.

	SourceTable	OriginalRowID	ErrorTaskName	ErrorDescription	ErrorType	ErrorDateTime
1	NULL	NULL	Fact Tables	The expression "(HORA_PARTIDA < DATEADD("...	Fatal	2021-10-06 15:40:50.107
2	NULL	NULL	Fact Tables	SSIS Error Code DTS_E_PROCESSINPUTFAILE...	Fatal	2021-10-06 15:43:53.673
3	Departure	23	Fact Tables	On step: Planned vs Real Check.Outputs[Valid Plan...	Non-Fatal	2021-10-06 15:46:00.730

Figura 4.12: Parte da tabela ErrorHandler, com registos de erro fatal e não fatal

4.5.2.C Falhar

Por alguma razão, como uma falha nos conectores, pode também dar-se o caso do pacote não chegar a ser validado. Quando o passo de validação falha, o pacote não consegue ser executado, e nem o Event Handler, nem o package logging, contidos no pacote, o conseguem reportar. Assim não haverá nenhum registo do erro nem do que o provocou, apenas falhará. Para alertar a pessoa responsável do sucedido, será enviado um email, cada vez que não há sucesso na execução do pacote, utilizando o SQL Server Agent, abordado no seguinte capítulo do agendamento.

4.6 Automatização

Neste capítulo aborda-se a questão da automatização do processo de carregamento do DW. Foi feito com recurso às funcionalidades do SSMS, tendo em primeiro lugar sido iniciados os serviços do SQL Server Agent. O SQL Server Agent é um componente do Microsoft SQL Server que agenda Jobs e manipula outras tarefas automatizadas. Em primeiro lugar foi necessário converter o pacote do SSIS, criado no Visual Studio, para deployment model, e importá-lo para um novo catálogo de Integration Services no SSMS, o default SSISDB.

É então programado um novo Job cujo propósito é correr um pacote SSIS. Para o programar temos as seguintes propriedades:

- General
- Steps
- Schedules
- Alerts
- Notifications
- Targets

Em *steps*, é definido que o objeto a ser executado é o pacote que foi importado previamente para o catálogo do SSMS. Em *schedules*, é agendado um horário para que o *step* seja executado na frequência pretendida, com as definições abaixo estipuladas. Foi decidido pela equipa que a frequência mais ajustada para o carregamento do DW seria semanal. Não obstante, esta periodicidade é facilmente reajustada caso se verifique necessário.

Em último lugar é agendado o envio de um mail caso o Job não seja executado com sucesso. É a única forma de feedback quando existe um erro do tipo 3, o último tipo de erro referido no capítulo anterior, quando falha a validação do pacote. É enviado de forma semelhante quando existe um erro de tipo 2, mas neste caso existem registos do erro na tabela *ErrorHandler*.

4.7 Relatórios

Foi instalado o PowerBI Desktop para se proceder à criação dos relatórios. Aquando da criação de um relatório em Power BI é imposta a questão da forma de acesso aos dados: Import ou DirectQuery. Em Import, as tabelas e colunas selecionadas são importadas, de uma forma comprimida, para o servidor do Power BI. Quando se interage com os relatórios, o Power BI utiliza os dados importados em cache, para ver alterações mais recentes, o conjunto de dados completo tem de ser atualizado para a versão mais recente, presente no DW. Com a modalidade DirectQuery nenhum dado é importado, à medida que se interage com os dados, o Power BI consulta a fonte dos dados diretamente, pelo que estão sempre presentes no relatório os dados mais recentes.

Embora aparentemente o DirectQuery pareça a opção mais desejável, a decisão teve de ser cuidadosamente ponderada, dado que teoricamente o DirectQuery tem um desempenho bastante inferior em termos de velocidade de pesquisa e algumas limitações na funcionalidade de criação de relatórios. Por outro lado, o modo Import tem a contrariedade de exigir mais memória, sendo os dados guardados novamente no próprio relatório, e ter um limite máximo de 1GB de tamanho do dataset para a versão sem custos. Este espaço seria estendido para 10 GB com a versão Premium. As diferenças de performance eram substanciais e foi decidido utilizar o modo de conexão Import.

Antes de começar a composição dos relatórios foram examinados os indicadores que se queriam apresentar e pensados os gráficos mais ajustados para o fazer, que conseguissem responder a todas as questões identificadas, bem como cada um deles se iria complementar. Foi também criado um variado conjunto de novas medidas auxiliares, em DAX, para a apresentação de medidas mais complexas nas visualizações dos relatórios.

Um relatório é composto por um conjunto de elementos visuais, que vão de gráficos a painéis de filtro, todos eles com interatividade entre eles.

De seguida apresentam-se três relatórios produzidos, com os indicadores a apresentar, os elementos visuais criados para tal e a demostração certos elementos de interatividade.

4.7.1 Relatório de Indicadores de Carreira

O relatório de Indicadores de Carreira apresentado na Secção 3.4 traduziu-se nos seguintes gráficos:

- Slicer Carreiras - Filtro das carreiras existentes
- Gauge - Gráficos de gauge que mostram a percentagem de viagens realizadas, completas, com partida pontual, e com a duração prevista
- Line chart - Viagens planeadas e realizadas por unidade de tempo, uma hierarquia temporal no eixo x
- Valores de duração planeada, duração real, desvio médio, quilómetros planeados, quilómetros realizados e velocidade média

A Figura 4.13 mostra o relatório em default com dados relativos a todas as carreiras.

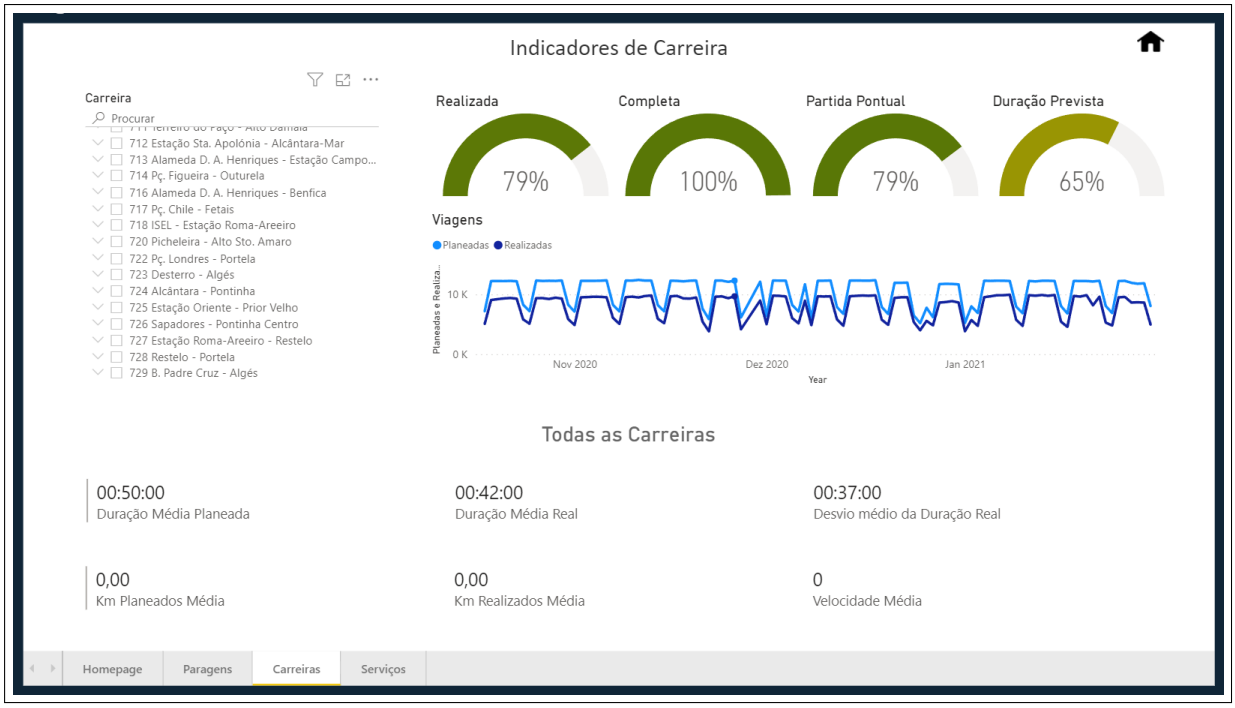

Figura 4.13: Relatório de Indicadores de Carreira no estado default

Alguns elementos do relatório são interativos, na Figura 4.14, foi selecionada a carreira 750 na variante ascendente 22343 no mês de novembro de 2020. Com esta seleção, os restantes gráficos passam a mostrar apenas indicadores relativos a estes elementos. A hierarquia temporal pode ser manipulada através de botões no canto superior direito do elemento visual ”viagens”.

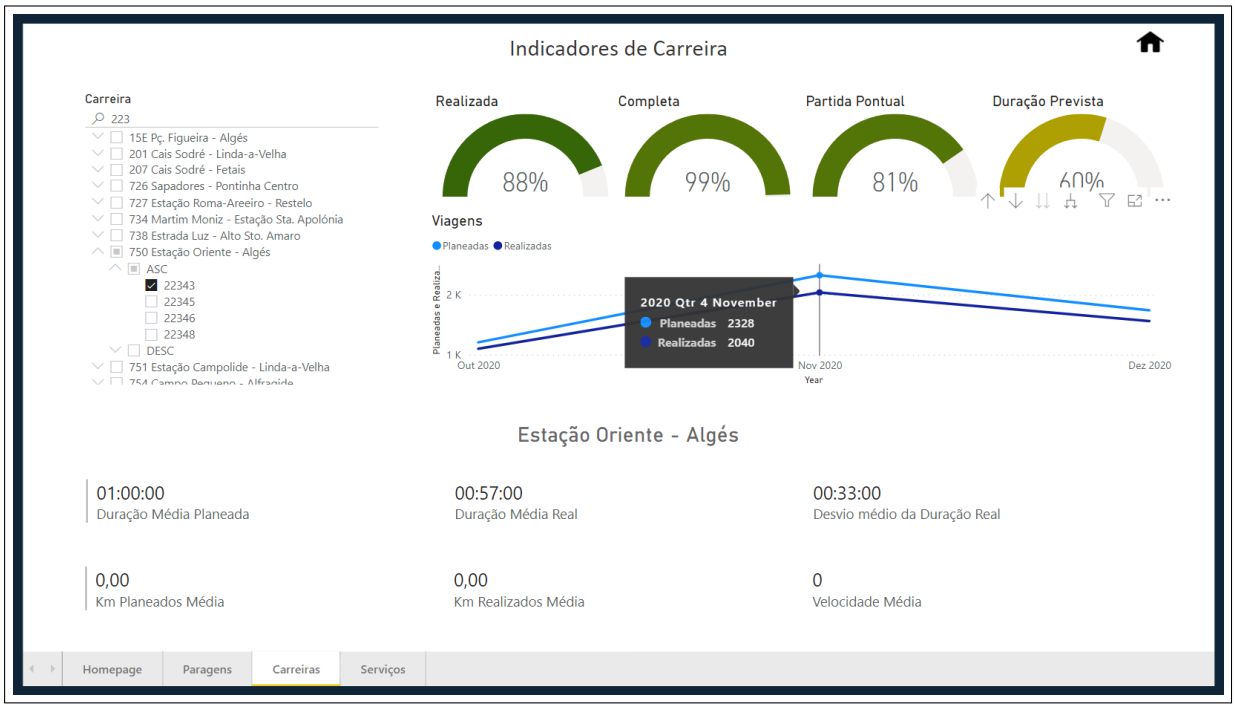

Figura 4.14: Relatório de Indicadores de Carreira com seleção de variante de carreira e data

4.7.2 Relatório de Indicadores de Pontos de Paragem

O relatório de Indicadores de Pontos de Paragem apresentado na Secção 3.4 traduziu-se nos seguintes gráficos:

- Slicer Data - Intervalo de tempo a ser observado
- Slicer Paragem - Lista das paragens existentes
- Frequência/Linha - A frequência de passagem, o tempo médio entre a passagem de dois veículos da mesma carreira pela paragem
- Passagens/Linha - A distribuição das carreiras dos veículos que passam pela paragem
- Passagens/Hora - A distribuição das horas em que os veículos que passam pela paragem
- Tempo Médio de Paragem – O tempo que os veículos demoram numa paragem de entrada e saída de passageiros.

Na Figura 4.15 é selecionada uma paragem específica e um intervalo de tempo, que traduz uma variação em todos os outros gráficos, que passam a mostrar apenas dados relativos a essa paragem.

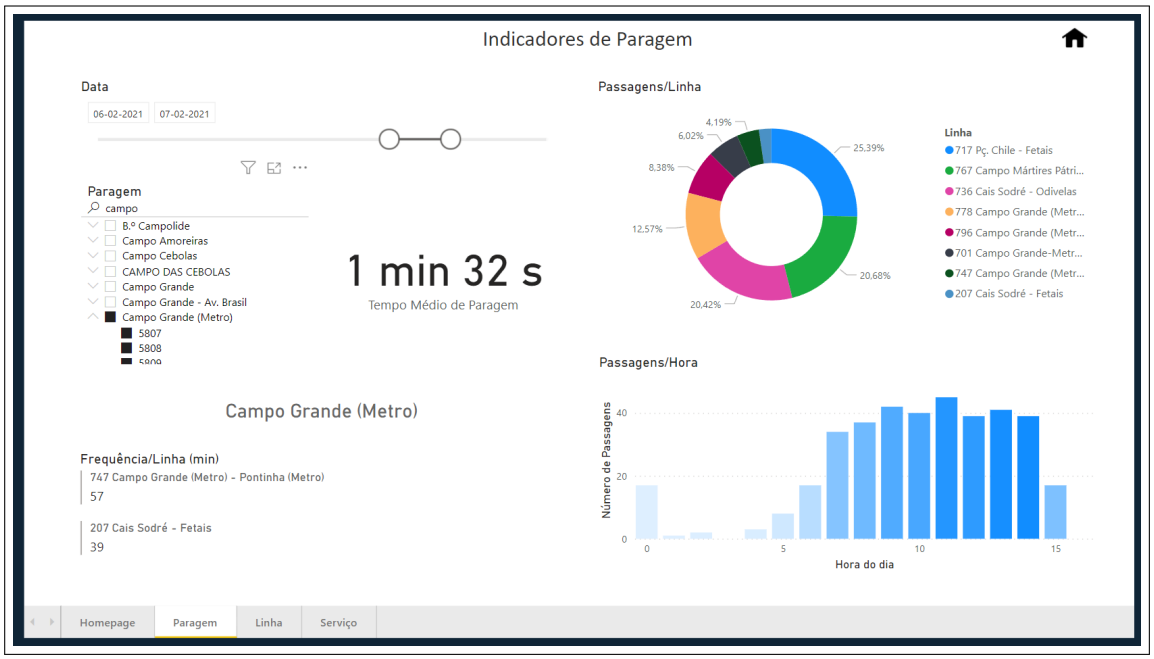

Figura 4.15: Relatório de Indicadores de Paragem com seleção de paragem

Na Figura 4.16 é selecionada uma hora específica no gráfico Passagens/Hora e observa-se a repercussão que esta tem nos restantes gráficos. Por exemplo, Frequência/Linha mostra agora o intervalo da passagem de veículos às 11 horas por cada carreira. Poderíamos ainda selecionar uma carreira específica para observar no contexto da paragem.

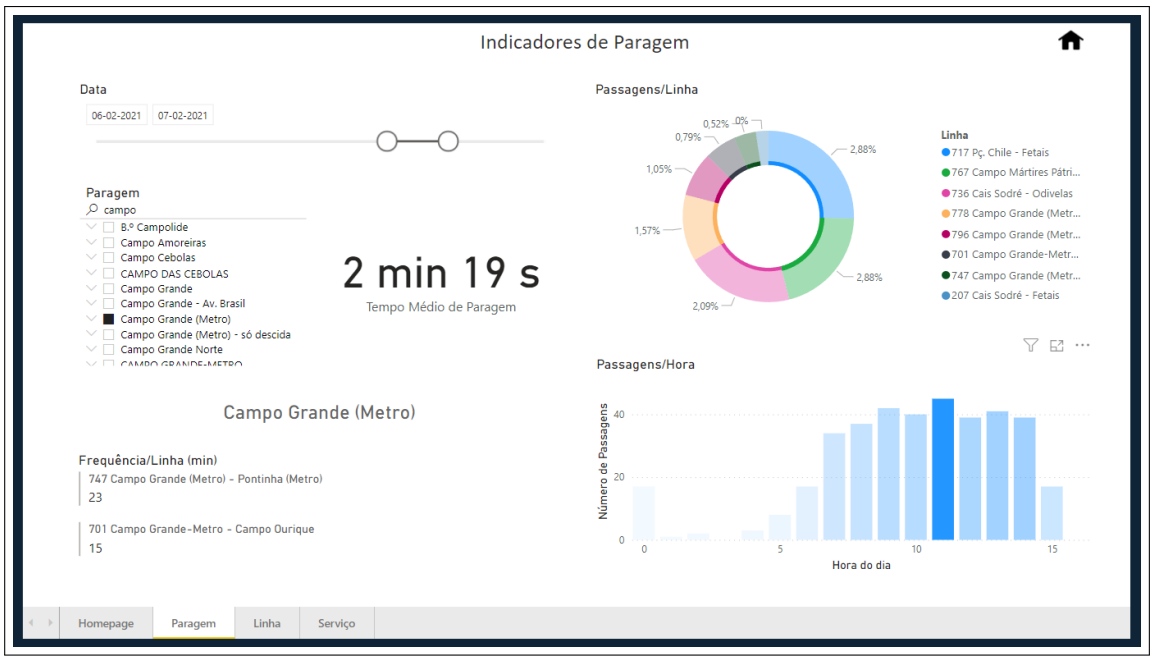

Figura 4.16: Relatório de Indicadores de Paragem com seleção de paragem e carreira

4.7.3 Relatório de Indicadores de Serviços

Serviços

Por uma questão de planeamento e organização, as viagens são agrupadas em serviços, que podem ser entendidos como o encadeamento lógico diário de múltiplas viagens, podendo estas pertencer à mesma carreira ou não. Ao iniciar funções, cada condutor terá um serviço a realizar, de ressalvar, no entanto, que este serviço pode ter uma permuta de condutor a meio.

Temos então como definição técnica de serviço, uma sucessão de viagens pertencentes à mesma chapa que decorreram na mesma data. Como o DW tem na sua estrutura básica o registo de viagens e suas métricas, foi necessário arranjar uma forma de as agrupar para poder observar e explorar serviços como um todo. Ainda no SSMS, foi criada uma nova view, que agrega os factos relativos às viagens que compõem cada serviço, como apresentado na imagem abaixo.

Listagem 4.4: Código de criação de SERVICES_VIEW

```
1  CREATE OR ALTER VIEW SERVICES_VIEW AS
2  SELECT Plate_Nr, Service_Date,
3       COUNT(b.Plate_Nr) as Planned_Trips,
4       SUM(Concluded) as Concluded_Trips,
5       SUM(Duration_Planned) as Duration_Planned_Services,
6       SUM(Duration_Real) as Duration_Real_Services,
7  CASE WHEN (count(b.Plate_Nr) - SUM(concluded)) = 0 THEN 1
8       ELSE 0 END as Complete_Services,
9  CASE WHEN sum(Concluded) is NULL THEN 0
10      WHEN (count(b.Plate_Nr) - SUM(concluded)) = COUNT(b.Plate_Nr) THEN 0
11      ELSE 1 END as Held_Services
12 FROM DERIVATES_VIEW as a, Plate as b WHERE a.PlateID = b.PlateID
13      GROUP BY service_date, b.Plate_Nr
```

Temos na definição de serviço realizado como um serviço com pelo menos uma viagem completa e um serviço completo como serviço com todas as viagens completas.

Chapa	Data	Viagens Planeadas	Viagens Completas	Serviço Realizado	Serviço Completo
10Z28E	**2021/02/21**	16	16	X	X
10Z28E	**2021/02/22**	16	12	X	
10Z28E	**2021/02/23**	16	0		

Tabela 4.4: Tabela resumo de serviços

Para apresentar dados referentes aos serviços foi necessário criar um conjunto de novas medidas que agrupam as métricas das viagens, utilizando a linguagem DAX.

O relatório de Indicadores de Serviços apresentado na Secção 3.4 traduziu-se nos seguintes gráficos:

- Filtro Chapa - Filtro das chapas existentes
- Gauge - Gráficos de gauge que mostram a percentagem de serviços que realizados e completos.
- Line chart Serviços - Serviços planeados e completos por unidade de tempo, tendo a hierarquia temporal no eixo x
- Valores respetivos de duração planeada e duração real

A Figura 4.17 mostra o relatório em default com dados relativos a todos os serviços.

Figura 4.17: Relatório de Indicadores de Serviços no estado default

É depois possível selecionar uma chapa específica e navegar a hierarquia temporal, passando os restantes gráficos a mostrar apenas dados relativos a essas seleções.

4.8 Integração na plataforma XTraN Passenger

Depois dos relatórios terem sido criados no Power BI Desktop, foram carregados no Report Server. A partir daí, todos os que sejam autorizados podem visualizar os relatórios, sem terem de ter uma conta própria de Power BI, estando ao mesmo tempo os relatórios protegidos pela firewall da empresa. O Power BI Report Server é uma solução *on-premises* para o Power BI, o servidor com os relatórios está instalado nos servidores da Tecmic, como era pretendido, ao invés de estarem alojados na cloud.

O próximo passo foi criar uma página web, na qual estão contidos os relatórios, aos quais é possível aceder remotamente. Através do Visual Studio foi criado um ficheiro em ASP.NET para se proceder a uma prova de conceito da integração de uma página, na qual estão contidos os relatórios, na plataforma web da Tecmic. Primeiramente, a solução encontrada para integrar os relatórios numa página web passa por uma iFrame, formada pelo endereço URL da localização dos relatórios, onde está contido o DNS do servidor onde estão alojados, e o sufixo '?rs:Embed=true'. Já implementada a página, consegue visualizar os relatórios, quem está ligado ao servidor por meio de uma VPN. Sendo desta forma possível aceder remotamente aos relatórios, a partir das múltiplas empresas que usufruem do serviço. Numa versão para além da prova de conceito, seriam necessárias as demais autenticações de utilizador na própria aplicação do XTraN Passenger.

Figura 4.18: Página inicial de acesso aos relatórios na aplicação web do XTraN Passenger

4.9 Purga de Dados

Foi implementado um mecanismo de purga de dados automatizado às tabelas FactTrip e FactStop, no qual registos com mais de 3 anos são eliminados, e nas tabelas auxiliares, como a IncoherentTrip, IncoherentEvent e ErrorHandler, com o limite de 1 ano. Embora este assunto tenha sido abordado, questões de purga de dados são contempladas nos contratos entre a Tecmic e as empresas que contratam os seus serviços. O tempo, número de anos usualmente, que os dados permanecerão armazenados nos servidores varia com este contracto, podendo inclusive estar dependente das leis do país onde está situada a empresa. Neste contrato poderão também vir definidas questões de duplicação dos dados ou a migração dos dados para um outro DW histórico, que não foram contempladas neste projeto.

5

Testes e Avaliação

Conteúdo

O sistema é composto por uma sequência de tecnologias e componentes, para sua avaliação, o fluxo de dados que passa por cada um deles terá de ser avaliado. Ao projeto realizado será de relevância testar o processo ETL, o DW, a integração da ferramenta BI e a produção de relatórios.

O processo de planeamento dos casos de teste começa logo na fase de construção do DW. Os casos de teste foram criados com base nos documentos de mapeamento da fonte-destino e nos requisitos. Mas em primeiro lugar, foi feita a verificação dos metadados, havendo a validação da estrutura da fonte e do destino no que diz respeito ao documento de mapeamento. Foi feita uma verificação do tipo de dados, do seu comprimento, das restrições impostas, e se as keys configuradas para serem geradas automaticamente eram criadas corretamente no DW.

Ao desenvolver um DW, será impossível testar todas as combinações de dados separadamente, desta forma, é importante determinar pontos estratégicos que terão de ser testados. De uma forma geral é norma começar com um pequeno conjunto de testes, geralmente erros mais graves irão aparecer e podem ser resolvidos rapidamente. Os seguintes testes foram realizados com recurso a automatização.

5.1 Testes Unitários

Para verificar a completude e qualidade dos dados, foram realizados uma série de testes unitários, que testam secções individuais do sistema e verificam se estão de acordo com o esperado. Foi usado tSQLt, uma framework open-source que permite executar testes unitários em bases de dados SQL Server. Os testes foram escritos na linguagem T-SQL, uma extensão de SQL. Depois, para a sua automatização, recorreu-se ao SQL Server Agent, onde se criou um job que permite correr todos os testes em sequência.

Podemos dividir estes testes em 5 partes, inserção básica de uma viagem nas diferentes tabelas, reação do sistema a dados duplicados e antigos, mudanças ao nível das SCDs e dados que ponham à prova as restrições aplicadas. Na tabela abaixo temos o resumo e resultado dos testes executados.

De seguida, na secção de código 5.1, temos um exemplo de um dos testes aplicados, ‘Inserção correta dos campos em Route’, um dos testes da secção ‘Dados Corretos’, que com o aparecimento de um novo registo de carreira, verifica se a inserção dos campos em Route está correta.

Testes	Resultado
Dados Corretos	
Inserção de dados corretos em Plate	✓
Inserção de dados corretos em Driver	✓
Inserção de dados corretos em Stop	✓
Inserção de dados corretos em Time	✓
Inserção de dados corretos em Route	✓
Inserção de dados corretos em FactTrip	✓
Inserção de dados corretos em FactStop	✓
Dados Duplicados	
Inserção de dados duplicados em Plate	✓
Inserção de dados duplicados em Driver	✓
Inserção de dados duplicados em Stop	✓
Inserção de dados duplicados em Time	✓
Inserção de dados duplicados em Route	✓
Dados Antigos	
Inserção de dados antigos em Plate	✓
Inserção de dados antigos em Time	✓
Inserção de dados antigos em Route	✓
Inserção de dados antigos em FactStop	✓
Inserção de dados antigos em FactTrip	✓
SCD	
Mudança em Stop	✓
Mudança em Route	✓
Dados Incorretos	
Tempo de partida superior ao de chegada	✓
Tempo de partida e chegada diferem mais de 8 horas	✓
Partida planeada e partida real diferem mais de 1 dia	✓
Quilómetros realizados ou planeados negativos e superiores a 500	✓
Contagem de evento de excesso de travagens superior a 100	✓

Tabela 5.1: Testes Unitários Realizados

Listagem 5.1: Modelo do teste unitário básico a Plate

```
1  CREATE OR ALTER PROCEDURE testBasic.[test Basic Plate]
2
3  AS
4  BEGIN
5  IF OBJECT_ID('actual') IS NOT NULL DROP TABLE actual;
6  IF OBJECT_ID('expected') IS NOT NULL DROP TABLE expected;
7
8
9      SELECT  PlateID, Plate_Nr, Trip_Nr
10     INTO actual
11     FROM Plate;
12
13
14     CREATE TABLE expected(
15     PlateID INT,
16     Plate_Nr NVARCHAR(10),
17     Trip_Nr INT
18     );
19
20
21     INSERT INTO expected (PlateID,Plate_Nr, Trip_Nr) VALUES (1, '6Z15E', 2);
22
23    EXEC tSQLt.AssertEqualsTable 'expected', 'actual';
24 END;
25
26 GO
```

5.2 Testes de Integração

Os testes de integração determinam se os diferentes componentes e módulos do sistema, desenvolvidos independentemente, funcionam corretamente quando interligados uns com os outros. Podemos também dizer, que verificam se um sistema composto por múltiplos módulos funciona corretamente [35].

Neste projeto, a condição para se verificar uma correta integração dos módulos, é que os registos presentes nas bases de dados fonte se convertam em informação correta quando apresentados nos relatórios.

O correto funcionamento dos relatórios foi testado nos próprios, sem recurso a testes codificados ou automatizados. De acordo com os dados existentes numa BD fonte, foram calculados os resultados que os relatórios deveriam apresentar. Os dados passam pelo processo ETL, pelo DW e só por fim são apresentados nos relatórios. A sua correção pressupõe a correta integração destes módulos.

Foram planificados testes com parte dos dados que tinham sido anteriormente criados para concretizar os testes unitários ao DW, com um reduzido número de registos para que os cálculos poderem ser efetuados.

Foi feito este processo para cada um dos relatórios, como apresentado nas seguintes tabelas. Temos os valores esperados com os relatórios em default, e a sua progressão à medida que os dados são observados de diferentes perspetivas.

Na tabela 5.2 temos os valores esperados para os indicadores de Paragem. Em primeiro lugar foi verificada a correção dos resultados que apareciam nos visuais do relatório no seu estado *default*. Após isto foi selecionada a a paragem "Santo Amaro", no dia "31/01/21"e os resultados dos visuais alteraram-se como de esperado. Por fim foi feita uma nova seleção, ainda no âmbito da paragem e data anteriormente definidas, agora para ver apenas a passagem da carreira 750.

Indicadores de Paragem	**Conteúdo Esperado**	**Resultado**
Geral: Todas as paragens e datas		
Passagens/Linha	15E: 25% 764: 45% 750: 30%	✓
Passagens/Hora	8: 5 | 9: 2 | 10: 2 | 11: 1 | 12: 1 | 15:1 | 16:1 | 17:2 | 18: 3 | 19:1 | 20:1	✓
Tempo Médio Paragem	0 min 51 segundos	✓
Frequência/Linha	15E: 13 min 750: 67 min 764: 46 min	✓
Paragem Santo Amaro a 31/01/21		
Passagens/Linha	750: 40% 764: 60%	✓
Passagens/Hora	8:1 | 9:1 | 10:1 | 18:1 | 20:1	✓
Tempo Médio Paragem	0 min 58 segundos	✓
Frequência/Linha	750: 61 min 764: 81 min	✓
750 na paragem Santo Amaro a 31/01/21		
Tempo Médio Paragem	1 min 25 segundos	✓

Tabela 5.2: Teste de Integração do relatório de indicadores de paragem

Na tabela 5.3 temos os valores esperados para os indicadores de Carreira. Como para o relatório ante-

rior, foram verificados os dados no relatório em *default*, em seguida foi selecionada carreira específica, e após isto uma variante dessa mesma carreira numa certa data, onde os resultados coincidiram com o esperado.

Indicadores de Carreira	**Conteúdo Esperado**	**Resultado**
Geral: Todas as Carreiras e datas		
Realizada	95%	✓
Completa	94%	✓
Partida Pontual	72%	✓
Duração Prevista	72%	✓
Duração Média Planeada | Real	00:31 | 00:30	✓
Desvio médio da duração real	00:04	✓
Km Planeados | Realizados	15.16 km | 13.95 km	✓
Velocidade Média	45 Km/h	✓
Carreira 750		
Realizada	100%	✓
Completa	100%	✓
Partida Pontual	80%	✓
Duração Prevista	100%	✓
Duração Média Planeada | Real	00:35 | 00:30	✓
Desvio médio da duração real	00:01	✓
Km Planeados | Realizados	10.7 km | 10 km	✓
Velocidade Média	33 Km/h	✓
Carreira 750 Variante 17400 a 13/03/21		
Realizada	100%	✓
Completa	100%	✓
Partida Pontual	33%	✓
Duração Prevista	100%	✓
Duração Média Planeada | Real	00:34 | 00:31	✓
Desvio médio da duração real	00:02	✓
Km Planeados | Realizados	10 km | 10 km	✓
Velocidade Média	32 Km/h	✓

Tabela 5.3: Teste de Integração do relatório de indicadores de carreira

O teste de verificação para o relatório de Indicadores de Serviços seguiu o mesmo procedimento dos anteriores. Como podemos ver na tabela 5.4 verificaram-se, incrementalmente, os dados apresentados no estado default, com a chapa ”3Z750”selecionada, e finalmente para o dia ”13/03/21”. Para todas estas seleções, verificou-se que os dados apresentados nos relatórios coincidiam com os dados calculados manualmente.

Indicadores de Serviços	**Conteúdo Esperado**	**Resultado**
Geral: Todas as chapas e datas		
Realizados	100 %	✓
Completos	75 %	✓
Duração Média Planeada | Real	02:31 | 02:07	✓
Chapa 3Z750		
Realizados	100 %	✓
Completos	100 %	✓
Duração Média Planeada | Real	02:55 | 02:31	✓
Chapa 3Z750 a 13/03/21		
Realizados	100 %	✓
Completos	100 %	✓
Duração Média Planeada | Real	03:29 | 03:05	✓

Tabela 5.4: Teste de Integração do relatório de indicadores de serviços

Após isto, foi também verificada a correta atualização dos dados apresentados ao haver mudanças nos dados do DW.

5.3 Avaliação

Como principal métrica de avaliação do projeto temos a performance. A parte do projeto em que os objetivos de desempenho são mais importantes, é na utilização dos relatórios. Para avaliar o desempenho dos relatórios foi utilizado o Performance Analyzer, componente do Power BI que regista a duração de cada ação que se executa no relatório. Regista o tempo que cada elemento visual demora a estar pronto, numa manipulação do relatório. Para efeitos de avaliação, foi considerada como duração do passo, o tempo que demorava até o último elemento visual estar carregado. De seguida temos a Tabela 5.5 que mostra como se comportaram os 3 relatórios na execução de passos.

O servidor usado é uma máquina virtual com o processador Intel(R) Xeon(R) E5645 @ 2.40GHz e 20GB de RAM. Os relatórios foram executados com 1, 2 e 10 milhões de registos de viagens na tabela FactTrip, para poder ser comparada a variabilidade do número de registos, com a performance. Para o cliente da Tecmic, a Carris, são realizadas cerca de 15 mil viagens diárias, segundo números das bases de dados fonte. Desta forma, 1 milhão de registos corresponderá a cerca de 2 meses de informação, 2 milhões a 4 meses, e 10 milhões, a quase 2 anos de informação relativa a viagens.

Para além disto, nos passos criados nos próprios relatórios foi escolhida a seleção de um elemento com muitas instâncias de viagens associadas, a carreira 15E e a chapa 1Z31B , e elementos com poucas, a carreira 73B e a chapa 22Z767. Cada um dos passos foi executado 10 vezes, sobre os quais foi feita a média, para aumento da precisão.

	1 M	**2 M**	**10 M**
Tarefa	**Duração (ms)**		
Selecionar Carreira 15E	206	216	226
Retirar Seleção	207	188	231
Selecionar Carreira 73B	233	236	238
Retirar Seleção	377	211	193
Selecionar Data 09/11/1010	176	144	270
Selecionar Chapa 1Z31B	197	182	165
Retirar Seleção	152	170	159
Selecionar Chapa 22Z767	181	179	239
Retirar Seleção	172	154	144
Selecionar Data 26/11/2020	133	146	197

Tabela 5.5: Avaliação da Performance dos relatórios com 1, 2 e 10 milhões de registos de viagens em *FactTrip*

Numa primeira abordagem os resultados mostraram-se inconclusivos não sendo observável a correlação esperada entre o tamanho do *dataset* e a duração do *feedback* às manipulações feitas. Numa análise mais detalhada, temos o tempo de compilação de cada elemento dividido nos seguintes: DAX, Visual e Others. Onde que DAX é o tempo de execução da *query* DAX associada ao passo, Visual é o tempo de renderização do elemento visual e Others, o tempo exigido pelo visual para preparar *queries*, aguardar pelo término de outros visuais ou realizar outro tipo de processamento em segundo plano. A duração do componente DAX é de facto mais elevada quando há um maior número de registos, mas a proporção do tempo ocupado neste componente em relação aos outros é tão reduzida, que não foi possível es-

tabelecer uma correlação entre o tamanho do *dataset* e a duração na tabela acima. Pelo que, para o utilizador, não serão aparentemente percetíveis mudanças do tamanho do *dataset*, para os valores avaliados. Como temos quase 2 anos de registos associados à coluna 10M, e a purga de dados na FactTrip é efectuada aos 3 anos, ressalvando que numa versão comercial este valor será definido por cada uma das empresas, as diferenças no tempo de *loading* devido à dimensão do dataset não deverão ser um fator de preocupação.

Em relação aos resultados gerais, são satisfatórios, pois para nenhuma das ações realizadas, a duração de resposta foi superior a 1 segundo, que como foi apontado na Secção 2.2 é o limite para que a linha de pensamento do utilizador permaneça ininterrupta. Desta forma, os objetivos de usabilidade foram cumpridos.

Quanto à avaliação do tempo de carga do DW, não foi efetuada, pois não se demonstrou relevante, já que o DW é apenas carregado uma vez por semana e o sistema BI é assíncrono em relação à execução das viagens.

Numa abordagem para descobrir como os restantes objetivos deste sistema foram cumpridos, tinham-se as seguintes métricas: custo e segurança. Tendo a Tecmic já acordos com a Micorsoft, incluindo os acordos e servidores com SQL Server, e o uso do Report Server, uma solução pouco divulgada pela Microsoft, os custos do projeto revelaram-se nulos, sem valor acrescentado aos habituais gastos da Tecmic, para uma versão de prova de conceito. O outro valor que se levantava era a segurança dos dados dos clientes. Foi possível construir um sistema em que os dados permanecem sempre dentro do domínio da Tecmic.

6

Conclusão

Conteúdo

Com este projeto será possível responder à necessidade do XTraN Passenger possuir um módulo que permita visualizar os dados armazenados nas suas bases de dados do SAEIP por novos prismas. E com estas novas perspetivas, auxiliar processos de tomada de decisão pelas empresas de transporte de passageiros que adquiram o produto à Tecmic. A solução passou por criar um modelo de Business Intelligence, dependente de um repositório histórico de dados, um Data Warehouse.

Foram primeiramente deliberadas as ferramentas a utilizar em cada uma das partes do trabalho. Por licenças prévias e compatibilidade da Tecmic com os produtos da Microsoft foi utilizado o SSIS para proceder ao ETL, o SQL Server como DW e o Power BI para as análises de Business Intelligence.

Como matéria-prima em bruto, tinham-se os dados da atividade dos veículos, recolhidos pelo hardware a bordo das frotas e guardados em bases de dados do SAEIP. Foram deliberados os campos relevantes da atividade para armazenar. Através da ferramenta SSIS, foi possível programar o procedimento de extração dos dados da fonte, transformação e adaptação dos mesmos às suas tabelas de destino e carregamento num DW em SQL Server *on-premises*, situado nas instalações da Tecmic. O DW tem como estrutura um *schema* fact constellation, com duas tabelas de factos, uma para viagens e outra para eventos de paragem, que para os dados e as suas relações foi verificado o mais satisfatório.

No trabalho feito com o SSIS, para além do natural procedimento de extração, transformação e carga dos dados, foi criado um conjunto de salvaguardas para incoerências e erros. Incoerências que ocorram nos dados a nível lógico são inseridas numa de duas tabelas de incoerências, ou para viagens ou para eventos. Erros que ocorram na execução do pacote de SSIS, fatais, ou não fatais, são colocados numa tabela para erros. Estes elementos defeituosos podem assim ser mais tarde analisados pela equipa e mais facilmente detectada a origem dos problemas. Com auxílio do SQL Server, este pacote de SSIS, será executado uma vez por semana, e há uma inserção de novos dados no DW.

Foi estabelecida uma conexão entre o DW e o serviço Power BI, onde foram criados relatórios de carreira, serviço e paragem, que permitem apresentar estes dados de uma forma elucidativa, são compostos por elementos visuais dinâmicos que permitem interagir e filtrar os dados de diferentes formas. Estes relatórios foram então alojados no Power BI Report Server, o seu acesso incorporado na aplicação web do XTraN Passenger, e por intermédio deste, acedido pelos computadores das empresas de transporte de passageiros.

Um dos maiores problemas encontrados foi a corrupção dos dados recolhidos pelos sistemas a bordo, com um elevado número de campos a NULL. Em alguns casos foi determinado inviável utilizar o atri-

buto, em outros, o atributo foi utilizado, devido à sua imprescindibilidade, isto pode levar a casos de valores inexatos nos relatórios face à realidade das viagens. Outro problema relacionado com as falhas na recolha de dados, é a associação entre viagens e eventos, certos eventos têm de ser descartados por não ser possível encontrar a viagem em que ocorreram. O contrário é ainda mais prevalente, muitas viagens não têm eventos associados, deixando uma pequena amostra servir como representação da realidade.

Na secção que fala sobre incoerências nos dados, 4.5.1, os dados ilógicos encontrados são armazenados em tabelas suplementares, para mais tarde serem observados. Não sendo inseridos em tabelas realmente pertencentes ao schema do DW, estes registos são perdidos para efeitos das análises aos relatórios, constituindo isto uma limitação do sistema.

Para versões em grande escala deste sistema, é estimado que o método de conexão nos termos presentes seja inviável, as soluções são o pagamento de taxas Premium à Microsoft, acrescendo os custos, ou mudar o tipo de conexão aos dados para *DirectQuery*, piorando o desempenho dos relatórios.

Com o desenvolvimento deste sistema será agora possível tirar partido dos dados recolhidos durante a atividade da frota, observando indicadores e estabelecendo relações causais, a título de exemplo, se temos um tempo de paragem excessivo para uma certa paragem poderá significar que algo: ou é uma paragem com muita procura e é preciso aumentar a frequência de passagem dos veículos, ou se se verificar que tal acontece para um certo veículo em específico, poderá significar que existe alguma problemática nos equipamentos de acesso. Se o desempenho do indicador *partida pontual* estiver anormalmente baixo numa dada época do ano, poderão ter de ser feitos ajustes no planeamento ou haver o aumento da oferta. Estas e muitas outras análises poderão ser feitas pelas empresas de transporte de passageiros, num intuito de melhorar a qualidade do serviço.

Com a realização deste projeto, é ambicionada uma maior competitividade do produto XTraN Passenger e derradeiramente a dos transportes de passageiros que o utilizam, face à concorrência direta e principalmente face a veículos particulares, contemplando todas as vantagens que daí advêm, seja comercial, para a Tecmic, seja da habitabilidade e qualidade de vida para uma cidade.

6.1 Trabalhos Futuros

Dentro do módulo concebido temos a sua entrada na última fase do desenvolvimento, o suporte, que decorre durante a utilização efetiva do sistema pelo cliente. Neste, poderão ser necessárias modificações

baseadas em pedidos por parte dos utilizadores, como novas funcionalidades ou relatórios. Para isto, poderão ser apenas necessárias mudanças ao nível dos relatórios, se forem unicamente necessários dados presentemente disponíveis no DW, ainda não usados, ou já utilizados em diferentes perspetivas ou podem ter de ser feitas alterações mais profundas, adicionando mais atributos desde as fontes. Conforme o tipo de cliente, pode também ser fundamental, mudar os valores de certas restrições impostas, tidas como incoerências, como o tempo de viagem ou o número de quilómetros percorridos, sendo que para este projeto foram tidas em conta viagens urbanas. Podem também ser identificados defeitos que imponham mudanças em qualquer uma das partes do sistema.

Quanto a conteúdos identificados desde já como importantes para aprimorar o módulo, temos a criação de um relatório que utilize os eventos de excesso cometidos nas viagens, a inserção de novos atributos relevantes à análise, não presentes no SAEIP utilizado, como afluência de passageiros ou vertentes monetárias do negócio e, encontrar um método para a avaliação dos condutores, para funcionar também como métrica na tabela de facto das viagens.

Outro trabalho futuro será, a realização de testes de usabilidade com utilizadores, já numa fase de incorporação com as empresas, de forma a ajustar e moldar os relatórios a cada uma das empresas que usufruem do XTraN Passenger.

Tem-se também a disponibilização de uma interface ao nível do DW, com os dados relativos a cada dia de exploração, em estruturas com todos os serviços e as viagens planeadas e realizadas, de preferência através de standards existentes para integração de dados de transportes como GTFS, NeTEx ou SIRI.

Bibliografia

[1] European Commission, “The Future Of Cities,” 2019, accessed 15-Out-2021. [Online]. Available: https://publications.jrc.ec.europa.eu/repository/handle/JRC116711

[2] Carris, “Relatório e Contas 2020 - CARRIS,” International Standard, 2020. [Online]. Available: https://www.carris.pt/media/wkidxkux/rc_carris2020_20210831.pdf

[3] Tecmic. (2020) XTraN Passenger. Accessed 25-Nov-2020. [Online]. Available: https://www.tecmic.com/solucoes-tecmic/gestao-de-transporte-de-passageiros/

[4] Q. Vo, J. Thomas, S. Cho, P. De, B. J. Choi, and L. Sael, “Next generation business intelligence and analytics: A survey,” 04 2017.

[5] Eckerson, W., *Performance Dasboards: Measuring, monitoring, and managing your business*. Wiley, 2011.

[6] M. Kasem and E. Hassanein, “Cloud business intelligence survey,” *International Journal of Computer Applications*, vol. 90, 02 2014.

[7] Cameron Graham. Why Companies Still Choose On Premise Business Intelligence Deployments. Accessed 15-Oct-2021. [Online]. Available: https://www.yellowfinbi.com/blog/2014/04/yfcommunitynews-why-companies-still-choose-on-premise-business-intelligence-deployments-160631

[8] Nielsen, J., *Usability engineering*. Boston: Academic Press, 1993.

[9] Microsoft. [Online]. Available: https://powerbi.microsoft.com/pt-pt/

[10] ——. (2021) Aplicar as noções básicas do DAX no Power BI Desktop. Accessed 16-Set-2021. [Online]. Available: https://docs.microsoft.com/pt-pt/power-bi/transform-model/desktop-quickstart-learn-dax-basics

[11] ——. Accessed 15-Dez-2020. [Online]. Available: https://docs.microsoft.com/pt-pt/power-bi/report-server/getting-around

[12] Tableau. Accessed 15-Oct-2021. [Online]. Available: https://www.tableau.com/

[13] A. Lopes, "Aplicação de Técnicas de Business Intelligence a Base de Dados Prosopográficas," Master's thesis, Universidade de Évora, 2017.

[14] Kimball, R., *The Data Warehouse Lifecycle Toolkit*. Wiley, 1998.

[15] Inmon, W.H., *Building the Data Warehouse*, 1992.

[16] Oracle. Data Warehousing Concepts. Accessed 17-Oct-2021. [Online]. Available: https://docs.oracle.com/cd/B10500_01/server.920/a96520/concept.htm#51078

[17] K. Hamstra, M., Zaharia, M., *Learning Spark: Lightning-Fast Big Data Analysis*, 2016.

[18] Microsoft. (2020) SQL Server Integration Services. Accessed 22-Jun-2021. [Online]. Available: https://docs.microsoft.com/pt-pt/sql/integration-services/sql-server-integration-services

[19] Microsoft Azure, "White paper - The Power BI Professional's Guide to Azure Synapse Analytics," 2020. [Online]. Available: https://azure.microsoft.com/en-us/resources/power-bi-professionals-guide-to-azure-synapse-analytics/

[20] Microsoft. (2021) What is SQL Server Management Studio (SSMS)? Accessed 19-Set-2021. [Online]. Available: https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms

[21] Friman, M., *Implementing quality improvements in public transport, Journal of Public Transportation*, 2004.

[22] B. Turnquist M., *Evaluating potential effectiveness of headway control strategies for transit systems*, 1980.

[23] Nakanishi, Y.J., *Bus performance indicators. On-time performance and service regularity", Transportation Research Record*, 1997.

[24] M. Eboli, L., "Performance indicators for an objective measure of public transport service quality." [Online]. Available: https://core.ac.uk/download/pdf/41174771.pdf

[25] Routematch. Business Intelligence: A Deep Dive. Accessed 17-Oct-2021. [Online]. Available: https://www.routematch.com/qa-business-intelligence/

[26] Rise of Big Data and Business Intelligence in Fleet Management. Accessed 17-Oct-2021. [Online]. Available: https://thelatiumgroup.com/blog/rise-of-big-data-and-business-intelligence-in-fleet-management/

[27] How Can Business Intelligence Solutions Benefit Companies in the Transport and Logistics Industry? Accessed 17-Oct-2021. [Online]. Available: https://www.infinitiresearch.com/thoughts/business-intelligence-solutions-benefits-transport-logistics-industry/?utm_source=BGWeek4&utm_medium=businesswireWeek4&utm_campaign=businesswireBGWeek4

[28] Digitalization for Public Transit. Accessed 17-Oct-2021. [Online]. Available: https://www.cosmoconsult.com/products/data-analytics/bi-industry-solutions/bi-branch-solution-public-transport/

[29] 4 moves towards data-driven public transport. Accessed 17-Oct-2021. [Online]. Available: https://zight.nl/solutions/

[30] Cygnet Infotech. Real-time Solution powered with Business Intelligence (BI) capabilities - For Effective Public Transport Management. Accessed 5-Jan-2021. [Online]. Available: https://www.cygnet-infotech.com/media/582851/real-time-solution-powered-with-business-intelligence.pdf

[31] Kris Wenzel. What is a Relational Database View? Accessed 26-Apr-2021. [Online]. Available: https://www.essentialsql.com/what-is-a-relational-database-view/

[32] Ben Snaidero. (2018) Surrogate Key vs Natural Key Differences and When to Use in SQL Server. Accessed 10-Jun-2021. [Online]. Available: https://www.mssqltips.com/sqlservertip/5431/surrogate-key-vs-natural-key-differences-and-when-to-use-in-sql-server/

[33] Richard Peterson. SSIS Tutorial for Beginners: What is, Architecture, Packages. Accessed 02-Jun-2021. [Online]. Available: https://www.guru99.com/ssis-tutorial.html

[34] Microsoft. Enhancing an Error Output with the Script Component. Accessed 21-Jun-2021. [Online]. Available: https://docs.microsoft.com/en-us/sql/integration-services/extending-packages-scripting-data-flow-script-component-examples/enhancing-an-error-output-with-the-script-component?view=sql-server-ver15

[35] Martin Fowler. (2018) Integration Test. Accessed 20-Set-2021. [Online]. Available: https://martinfowler.com/bliki/IntegrationTest.html

A

Views de Extração

Listagem A.1: Código de criação de LAST_RUN_EXTRACTION_VIEW

```
1 CREATE OR ALTER VIEW LAST_RUN_EXTRACTION_VIEW AS SELECT
2 (CASE WHEN (Select MAX(Date) from LastRun ) > DATEADD(month,-6, GETDATE())
3 then (Select MAX(Date) from LastRun) else DATEADD(MONTH, -6,GETDATE()) end)
4 as date  from LastRun;
```

Listagem A.2: Código de criação de TIME_EXTRACTION_VIEW

```
1 CREATE OR ALTER VIEW TIME_EXTRACTION_VIEW AS
2 SELECT id, hora_partida from Departure as a,
3 LAST_RUN_EXTRACTION_VIEW as c
4 where  a.HORA_PARTIDA > c.date
```

Listagem A.3: Código de criação de ROUTE_EXTRACTION_VIEW

```
1 CREATE OR ALTER VIEW ROUTE_EXTRACTION_VIEW AS
2 SELECT a.ID, HORA_PARTIDA, DATA_SERVICO, ROUTE_ID,
3 ROUTEVARIANTDIRECTION_ID, SENTIDO,
4 NR_CARREIRA, NOME_CARREIRA from Departure as a, Carreira as b,
```

```
5  LAST_RUN_EXTRACTION_VIEW as c where a.ROUTE_ID = b.ID
6  and a.DATA_SERVICO > c.date
```

Listagem A.4: Código de criação de PLATE_EXTRACTION_VIEW

```
1  CREATE OR ALTER VIEW PLATE_EXTRACTION_VIEW AS
2  SELECT ID, DATA_SERVICO,NR_CHAPA, NR_VIAGEM from Departure as a,
3  LAST_RUN_EXTRACTION_VIEW as c where  a.DATA_SERVICO > c.date
```

Listagem A.5: Código de criação de DRIVER_EXTRACTION_VIEW

```
1  CREATE OR ALTER VIEW DRIVER_EXTRACTION_VIEW AS
2  SELECT  ID, NR_MEC, nome, data_admissao, data_nascimento
3  FROM CONDUTOR
```

Listagem A.6: Código de criação de STOP_EXTRACTION_VIEW

```
1  CREATE OR ALTER VIEW STOP_EXTRACTION_VIEW AS
2  SELECT  ID_PARAGEM, NR_PTPARAGEM, NOME_PARAGEM from PTPARAGEM
```

Listagem A.7: Código de criação de TRIP_EXTRACTION_VIEW

```
1  CREATE OR ALTER VIEW TRIP_EXTRACTION_VIEW AS
2  SELECT a.ID, EffectiveDepartureTime, EffectiveConclusionTime, HORA_PARTIDA,
3  TripConclusionTime, DATA_SERVICO, NR_CHAPA, NR_VIAGEM, ROUTE_ID,
4  NR_CARREIRA, ROUTEVARIANTDIRECTION_ID, SENTIDO, StartVehicleKm,
5  ArrivalVehicleKm, PlannedKm, Vehicle_Id,NR_MEC from Departure as a
6  LEFT JOIN Carreira as b
7  ON a.ROUTE_ID = b.ID LEFT JOIN Condutor as c ON a.Driver_Id=c.ID,
8  LAST_RUN_EXTRACTION_VIEW as d where  a.DATA_SERVICO > d.date
```

Listagem A.8: Código de criação de EVENT_EXTRACTION_VIEW

```
1  CREATE OR ALTER VIEW EVENT_EXTRACTION_VIEW AS
2  (select  a.ID, a.NR_CHAPA, a.NR_CARREIRA, a.SENTIDO, a.ID_CVS,
3  a.NR_VIAGEM, a.NR_PARAGEM, a.EV_TIPO, a.EV_TIME,
4  Null as TEMPO_PARADO
5  from EVENT_DAY as a, LAST_RUN_EXTRACTION_VIEW as c
6  where   a.EV_TIME > c.date
7  and (a.EV_TIPO=5646 or a.EV_TIPO=65536 or a.EV_TIPO=65537 or
8  a.EV_TIPO=65538 or a.EV_TIPO=65540 or a.EV_TIPO=65543))
9  UNION
10 (select a.ID, a.NR_CHAPA, a.NR_CARREIRA, a.SENTIDO, a.ID_CVS,
11 a.NR_VIAGEM, a.NR_PARAGEM, a.EV_TIPO, a.EV_TIME, b.TEMPO_PARADO
12 from EVENT_DAY left join STOP_INFO_DIA as b
13 on a.NR_CARREIRA=b.NR_CARREIRA
14 and a.ID_CVS=b.ID_CVS and a.NR_CHAPA=b.NR_CHAPA and
15 a.NR_VIAGEM=b.NR_VIAGEM and a.NR_CARREIRA=b.NR_CARREIRA and
16 a.NR_PARAGEM=b.NR_PARAGEM and a.EV_TIME=b.DATA_HORA,
17 LAST_RUN_EXTRACTION_VIEW as c where
18 a.EV_TIPO=5644 and a.EV_TIME> c.date )
```