FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Integração Avançada de Dados: Uma Abordagem para Gestão Centralizada e Análise Estratégica

Diogo Alexandre da Costa Melo Moreira da Fonte

Mestrado em Engenharia Informática e Computação

Orientador: Prof. Mariana Curado Malta

17 de julho de 2025

Integração Avançada de Dados: Uma Abordagem para Gestão Centralizada e Análise Estratégica

Diogo Alexandre da Costa Melo Moreira da Fonte

Mestrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Prof. Gil Gonçalves

Arguente: Prof. Agostinho Pinto

Vogal: Prof. Mariana Malta

17 de julho de 2025

Resumo

O presente documento apresenta um projeto no âmbito de uma Dissertação de Mestrado em Engenharia Informática e Computação. O objetivo do projeto é o desenvolvimento de uma solução tecnológica centrada na integração avançada de dados, que otimiza a gestão da informação e apoia a análise estratégica, numa organização do setor das comunicações eletrónicas e serviços postais. O projeto foi realizado em colaboração com a empresa ARMIS e consistiu na construção de uma plataforma centralizada em *cloud*, suportada por Microsoft Fabric, que integra componentes de Engenharia de Dados e *Business Intelligence*.

A solução implementada incluiu a modelação de um armazém de dados analítico, estruturado em *star schema*, e o desenvolvimento de *data pipelines* de integração baseadas no paradigma ELT, que automatizam o carregamento, transformação e controlo dos dados. A camada de dados foi organizada segundo a arquitetura *Medallion* (bronze, prata e ouro), utilizando *notebooks*, *data pipelines* e mecanismos de alerta para garantir a rastreabilidade e fiabilidade dos processos. Para a componente analítica, foi desenvolvido um relatório Power BI composto por um *dashboard* interativo e uma infografia estática (ambos multilíngues), permitindo uma análise visual clara, consistente e acessível.

Os resultados obtidos demonstraram a eficácia e escalabilidade da solução proposta, sendo validada em ambiente real com dados confidenciais. A centralização da plataforma em ambiente *cloud* revelou-se um fator diferenciador, permitindo uma gestão facilitada dos artefactos e promovendo maior eficiência, manutenção e evolução contínua do sistema. A solução desenvolvida contribui ainda para os Objetivos de Desenvolvimento Sustentável (ODS) 8 e 9 das Nações Unidas, ao promover o crescimento económico sustentado e a inovação tecnológica na gestão de dados organizacionais.

Como trabalho futuro, está prevista a expansão do modelo de dados, a criação de novos relatórios temáticos, a integração de módulos funcionais adicionais com requisitos específicos, e o desenvolvimento de funcionalidades avançadas, como painéis de monitorização e componentes de análise preditiva. Estas evoluções visam reforçar a adaptabilidade, o valor analítico e o impacto estratégico da solução implementada.

***Keywords*:** Integração de Dados, *Business Intelligence*, Modelos Dimensionais de Dados, *Extract Load and Transform*, ELT, *Data Lakehouse*, *Cloud Technology*.

Abstract

This document presents a project developed within the scope of a Master's Dissertation in Informatics and Computing Engineering. The goal of the project is the development of a technological solution focused on advanced data integration, aimed at optimizing information management and supporting strategic analysis within an organization from the electronic communications and postal services sector. The project was carried out in collaboration with the company ARMIS and involved the construction of a centralized cloud-based platform, supported by Microsoft Fabric, integrating components of Data Engineering and Business Intelligence.

The implemented solution included the modeling of an analytical data warehouse structured in a star schema, and the development of integration data pipelines based on the ELT paradigm, automating data loading, transformation, and control. The data layer was organized according to the Medallion architecture (bronze, silver, and gold), using notebooks, data pipelines, and alert mechanisms to ensure process traceability and reliability. For the analytical component, a Power BI report was developed, consisting of an interactive dashboard and a static infographic (both multilingual), enabling clear, consistent, and accessible visual analysis.

The results demonstrated the effectiveness and scalability of the proposed solution, which was validated in a real-world environment using confidential data. The centralization of the platform in the cloud proved to be a differentiating factor, enabling simplified management of artifacts and promoting greater efficiency, maintainability, and continuous evolution of the system. The developed solution also contributes to the United Nations Sustainable Development Goals (SDGs) 8 and 9, by fostering sustained economic growth and technological innovation in organizational data management.

Future work includes expanding the data model, creating new thematic reports, integrating additional functional modules with specific requirements, and developing advanced features such as monitoring dashboards and predictive analytics components. These enhancements aim to further strengthen the adaptability, analytical value, and strategic impact of the implemented solution.

Keywords: Data Integration, Business Intelligence, Dimensional Data Models, Extract Load and Transform, ELT, Data Lakehouse, Cloud Technology.

Objetivos de Desenvolvimento Sustentável das Nações Unidas

Os Objetivos de Desenvolvimento Sustentável (ODS) das Nações Unidas constituem um plano global para alcançar um futuro mais justo, sustentável e próspero para todos. Os 17 objetivos incluídos abordam os maiores desafios sociais, económicos e ambientais da atualidade, incentivando ações em diversas áreas como a erradicação da pobreza, igualdade de género, acesso a serviços essenciais e promoção da inovação.

O projeto desenvolvido nesta dissertação contribui de forma direta para os seguintes ODS:

ODS 8 Promover um crescimento económico sustentado, inclusivo e sustentável, o emprego pleno e produtivo, e trabalho digno para todos.

ODS 9 Construir infraestruturas resilientes, promover a industrialização inclusiva e sustentável, e fomentar a inovação.

ODS	**Meta**	**Contributo**	**Indicadores e Métricas de Desempenho**
8	8.2	O projeto contribui para uma maior eficiência no setor dos serviços postais e das comunicações eletrónicas, apoiando o crescimento económico sustentado através da digitalização e otimização de processos.	Redução do tempo de obtenção de relatórios analíticos. Aumento da produtividade das equipas de análise.
	8.3	A plataforma promove práticas de inovação ao facilitar o acesso a dados integrados, permitindo decisões baseadas em evidência e a criação de novos serviços orientados por dados.	Número de decisões estratégicas suportadas por integração de dados.
9	9.1	A infraestrutura de dados construída com tecnologias modernas, como o Microsoft Fabric, suportada por *pipelines* automatizadas, apoia uma base digital resiliente e escalável.	Volume de dados processados. Número de *pipelines* executadas com sucesso.
	9.5	O uso de práticas de Engenharia de Dados e *Business Intelligence*, transformação automatizada e integração contínua promove a inovação tecnológica e o desenvolvimento de capacidades analíticas.	Número de *notebooks* utilizados. Taxa de automatização de tarefas repetitivas.

Agradecimentos

A realização desta dissertação marcou uma etapa significativa na minha formação académica e pessoal, sendo o culminar de um percurso de aprendizagem e crescimento. Este trabalho não teria sido possível sem o contributo, apoio e incentivo de várias pessoas, a quem expresso, com enorme gratidão, o meu sincero agradecimento.

À Professora Mariana Curado Malta, minha orientadora académica, agradeço profundamente a orientação rigorosa, a disponibilidade constante, os ensinamentos partilhados e a confiança demonstrada ao longo de todo o projeto. O seu acompanhamento foi essencial para a concretização deste trabalho com qualidade e ambição.

Ao Engenheiro Milton Nunes, orientador na empresa ARMIS, agradeço o apoio técnico, a partilha de conhecimentos práticos e a confiança no meu trabalho. A colaboração próxima com a equipa da ARMIS permitiu-me desenvolver competências cruciais e compreender melhor os desafios reais do mundo profissional.

À minha namorada, Ana Martins, um agradecimento especial pelo amor, paciência e apoio incondicional nos momentos de maior exigência. A tua presença foi um pilar fundamental ao longo desta jornada.

À minha mãe, Maria Alice Fonte, ao meu pai, Manuel Fonte, e ao meu irmão, Tiago Fonte, agradeço o apoio constante, os valores transmitidos e o incentivo permanente para seguir em frente. Sem o vosso suporte e confiança, este caminho não teria sido possível.

A todos os colegas, professores e amigos que, de forma direta ou indireta, contribuíram para o meu percurso académico e para a realização deste trabalho, deixo também uma palavra de reconhecimento e gratidão.

Por fim, à Faculdade de Engenharia da Universidade do Porto, pelo ambiente académico de excelência, e à ARMIS, pela oportunidade de desenvolver este projeto num contexto profissional real, deixo o meu sincero agradecimento.

Diogo Alexandre da Costa Melo Moreira da Fonte

"Success is the sum of small efforts,
repeated day in and day out."

Robert Collier

Índice

Lista de Figuras

Lista de Tabelas

Lista de Excertos de Código

Abreviaturas e Símbolos

BI	*Business Intelligence*
DAX	*Data Analysis Expressions*
DW	*Data Warehouse*
ELT	*Extract, Load, Transform*
ETL	*Extract, Transform, Load*
IA	Inteligência Artificial
JSON	*JavaScript Object Notation*
KPI	*Key Performance Indicator*
ML	*Machine Learning*
OLAP	*Online Analytical Processing*
RTDW	*Real Time Data Warehouse*
XML	*Extensible Markup Language*

Capítulo 1

Introdução

Este documento apresenta o trabalho de dissertação realizado no âmbito do Mestrado em Engenharia Informática e Computação (M.EIC), da Faculdade de Engenharia da Universidade do Porto (FEUP), no ano letivo 2024/2025. O projeto foi desenvolvido em colaboração com a empresa ARMIS Group[1] (doravante denominada por ARMIS).

Por meio desta dissertação, procura-se explorar e aplicar conceitos de *Business Intelligence* (BI), Engenharia de Dados e tecnologias emergentes, permitindo não apenas a implementação da solução proposta, mas também a análise comparativa de diferentes ferramentas e técnicas disponíveis no mercado. O trabalho pretende contribuir tanto para a evolução profissional do autor, quanto para a entrega de uma solução prática e eficiente.

Dado que o projeto foi desenvolvido no âmbito de um ambiente empresarial real e envolve um cliente da ARMIS, alguns dos artefactos apresentados ao longo do documento foram anonimizados ou apresentados com base em exemplificação técnica. Esta abordagem garante a confidencialidade da informação e assegura o cumprimento dos compromissos assumidos com as entidades envolvidas.

1.1 Contexto

A ARMIS tem como missão dedicar-se à inovação tecnológica, transformação digital e desenvolvimento de soluções à medida. A empresa é conhecida por desenvolver consultoria de soluções em empresas de grandes dimensões. A ARMIS possui escritórios em 7 referências geográficas: Porto e Lisboa (Portugal), São Paulo e São José dos Campos (Brasil - através da ARMIS Brasil), Dubai (Emirados Árabes Unidos - através da parceria SHIFT+ARMIS), Miami e Nova Iorque (Estados Unidos da América - através da NOVARMIS).

Esta dissertação tem como objetivo implementar uma solução tecnológica para otimizar a gestão de dados provenientes de diversas fontes de informação, eliminando os desafios relacionados à dispersão e descentralização de dados enfrentados por uma das empresas clientes da ARMIS. Para isso, será desenvolvida uma plataforma centralizada que integre, automatize e disponibilize

[1] ver https://www.armis.pt/armis/sobre-nos/ - acedido em 02/10/2024

informações críticas, permitindo maior eficiência operacional e suporte à tomada de decisão, eliminando silos de informação.

Será necessário:

- Integrar diversas fontes de dados;
- Automatizar os processos de preparação e transformação de dados e definir *schedules*, para serem executados de forma automática;
- Melhorar o acesso à informação;
- Disponibilizar relatórios de análise das várias fontes de informação, de forma a auxiliar as tomadas de decisão dos colaboradores.

1.2 Motivação

A ARMIS possui vários projetos inseridos em diversas áreas de informática. No contexto de análise e engenharia de dados, foi encontrado um problema de consolidação de dados num cliente da empresa. Este problema é uma oportunidade para o desenvolvimento pessoal e científico do autor, com o objetivo de desenvolver uma solução que o resolva.

O projeto oferece um campo fértil para explorar e comparar diversas tecnologias modernas, contribuindo para a especificação e desenvolvimento da solução final. Esta análise permitirá estudar e avaliar diferentes abordagens em termos de desenvolvimento, funcionalidades e desempenho. Além disso, o tema possibilita uma investigação de conceitos e técnicas nas áreas de BI e Engenharia de Dados, permitindo, ao autor, expandir o conhecimento e aplicar boas práticas do setor.

Ao longo deste trabalho, espera-se adquirir experiência prática com tecnologias que ainda não foram exploradas, comparando-as com as ferramentas que já são conhecidas, na área de engenharia e análise de dados. É pretendido ainda aprofundar as técnicas mais importantes nestes domínios, consolidando as competências necessárias para enfrentar os desafios técnicos e estratégicos do mercado atual.

1.3 Problema

A análise de dados no cliente da ARMIS enfrenta atualmente desafios significativos, devido à falta de centralização e à ineficiência dos processos utilizados. As informações provenientes de diversas fontes são frequentemente geridas de forma descentralizada, recorrendo a ficheiros individuais que são enviados por *e-mail* ou descarregados manualmente de plataformas distintas. Esta abordagem fragmentada resulta em dados dispersos entre múltiplas unidades de negócio, armazenados em formatos variados e partilhados de forma inconsistente. Com frequência, esta disseminação ocorre por meio de *e-mail* ou pastas de rede com acesso restrito a determinados colaboradores.

Este método de gestão de dados não apenas consome uma quantidade significativa de tempo para recolha e consolidação da informação, como também prejudica a eficiência operacional. Os colaboradores enfrentam dificuldades em obter os dados necessários para análises críticas, atrasando decisões estratégicas que deveriam ser suportadas por informações atualizadas e precisas. Além disso, a descentralização aumenta o risco de perda de informação, seja por falhas humanas ou técnicas nos canais de comunicação, comprometendo a integridade e a acessibilidade dos dados.

Estes desafios evidenciam a necessidade de implementar uma solução centralizada e automatizada, que elimine os silos de informação, otimize o acesso aos dados e facilite a análise e partilha de informação de forma segura, eficiente e confiável. Tal solução não apenas melhorará a eficiência interna, como também fortalecerá a base para tomadas de decisão mais rápidas e fundamentadas.

1.4 Perguntas de Investigação

P1 - Qual a melhor tecnologia para desenvolver a solução necessária?
P2 - Quais as técnicas/ferramentas a utilizar em cada etapa ou processo de desenvolvimento da solução?

1.5 Objetivos

O projeto tem como principal objetivo a implementação de uma base de dados analítica focada no mercado das comunicações eletrónicas, bem como a criação de um processo que acompanhe todo o ciclo de vida dos dados, desde as fontes até à sua disponibilização para análise e *reporting*.

Os objetivos secundários são:

- Centralizar a gestão da informação, eliminando silos de informação e de sistemas.
- Disponibilizar informação única, corporativa e colaborativa, contribuindo para uma difusão alargada e equitativa da informação.
- Aumentar a rapidez e garantir um acesso estruturado à informação, pelas unidades orgânicas do cliente e do público em geral.
- Auxiliar a tomada de decisão através de análises sistematizadas, promovendo melhor conhecimento do sector.
- Assegurar o acesso a soluções aplicacionais de configuração intuitiva e aplicabilidade abrangente.

1.6 Abordagem Metodológica

Esta secção descreve a abordagem adotada para resolver o problema identificado. A solução proposta baseia-se na integração de tecnologias avançadas de BI e Engenharia de Dados, com o objetivo de desenvolver uma plataforma centralizada para gestão e análise de dados. Serão detalhados os métodos e processos utilizados no desenvolvimento da solução, o desenho experimental para validação e os resultados esperados.

1.6.1 Solução

A solução proposta consiste em duas componentes principais:

- ***Data Warehousing*** - Implementação de um armazém de dados para centralizar, integrar e armazenar informações provenientes de diversas fontes organizacionais. Este armazém será modelado com base no modelo dimensional (*star schema*) criado, de modo a otimizar o desempenho em consultas analíticas e a facilitar a exploração de dados. Esta componente será implementada num *workspace* de Microsoft Fabric[2], utilizando diversas ferramentas disponíveis nesta tecnologia, como por exemplo, *python notebooks* e *data pipelines*.
- **Ferramentas de Análise Dinâmica** - Desenvolvimento e utilização de relatórios Microsoft Power BI[3], com o objetivo de criar visualizações interativas e personalizadas. Estes relatórios permitirão que os utilizadores finais explorem os dados de forma intuitiva, auxiliando na tomada de decisão baseada em factos. Esta componente de análise será também inserida no próprio *workspace* do Microsoft Fabric, mencionado na componente anterior.

1.6.2 Etapas

As etapas principais, constituintes da abordagem proposta, necessárias para o desenvolvimento da solução ao problema, são:

- Extração, transformação e carregamento dos dados provenientes de diversas fontes.
- Modelagem dimensional do armazém de dados.
- Desenvolvimento de relatórios analíticos, em Power BI.
- Processo experimental e avaliação da solução, com recurso a validações e testes, em ambiente de produção.

[2] ver https://www.microsoft.com/pt-pt/microsoft-fabric - acedido em 11/01/2025
[3] ver https://www.microsoft.com/pt-pt/power-platform/products/power-bi?market=pt - acedido em 11/01/2025

1.6.3 Processo Experimental e Avaliação

A avaliação da solução proposta foi conduzida com base em três parâmetros, definidos com o objetivo de analisar a robustez/desempenho, a escalabilidade e a utilidade prática da solução desenvolvida. Cada parâmetro está associado a aspetos específicos do sistema e é medido através de critérios objetivos.

O parâmetro 'Robustez-Desempenho' é avaliado através dos seguintes critérios:

- **Critério 1 - Qualidade do Modelo Dimensional** - Garantia de que o modelo dimensional está implementado de forma correta. A avaliação deste critério é realizada a partir da aferição da consistência e precisão dos dados no armazém de dados, através da verificação das relações entre tabelas e da integridade dos dados.

- **Critério 2 - Performance do Processo de Integração de Dados** - Garantia de que o processo de integração de dados é eficiente e eficaz. Para avaliar este critério é realizada a medição do tempo de execução, e a partir dos resultados da medição, analisada a eficiência dos processos de integração de dados.

O parâmetro 'Escalabilidade' é avaliado através do seguinte critério:

- **Critério 3 - Cobertura e Escalabilidade** - Garantia de que a solução atende a todas as áreas organizacionais relevantes, com capacidade para expandir à medida que novos requisitos surgirem. Para avaliar este critério são feitas experiências que consideram evoluções da versão inicial da solução.

Por fim, o parâmetro 'Utilidade Prática' é avaliado através do seguinte critério:

- **Critério 4 - Utilização do Relatório** - Garantia de que a solução atende às necessidades dos utilizadores finais, de acordo com a especificação. Para avaliar este critério é utilizada a técnica de observação da utilização do relatório analítico por utilizadores finais e a recolha de *feedback*, para validação das funcionalidades implementadas e da informação apresentada.

1.6.4 Resultados Esperados e Relevância

Com esta abordagem, espera-se alcançar os seguintes resultados:

- Centralização e integração de dados - Um armazém de dados unificado que permita eliminar silos de informação.

- Otimização do tempo de análise - Redução do tempo necessário para aceder e processar informações.

- Melhoria no suporte à tomada de decisões - Relatórios dinâmicos e visuais claros que possibilitem análises rápidas e fundamentadas.

- Relevância organizacional - Um impacto positivo na eficiência operacional e no suporte às decisões estratégicas. A relevância deste projeto reside na sua aplicabilidade prática, podendo servir como modelo replicável, para outras organizações, que enfrentem desafios semelhantes de gestão de dados.

1.7 Considerações Finais

A dissertação parte de um problema real de dispersão e dificuldade de acesso à informação, propondo uma solução tecnológica centrada na integração e análise avançada de dados. Através da construção de uma plataforma analítica centralizada, baseada em tecnologias modernas de Engenharia de Dados e *Business Intelligence*, pretende-se melhorar significativamente a eficiência na gestão e utilização da informação.

O projeto não só responde a necessidades específicas de uma organização parceira, como também constitui um caso aplicável a contextos semelhantes noutras entidades. A abordagem proposta aposta na escalabilidade, automação e fiabilidade dos processos de integração e análise de dados.

Nos capítulos seguintes, serão aprofundadas as revisões teóricas e tecnológicas que fundamentam a escolha da solução, seguidas da análise e especificação da mesma, culminando com a implementação, resultados e avaliação final.

Capítulo 2

Revisão de Literatura

Esta dissertação insere-se no contexto de Engenharia de Dados e *Business Intelligence* (BI), que são áreas que têm evoluído bastante devido à crescente importância dos dados e da sua análise. Com o aumento exponencial de dados gerados pelas organizações e a crescente demanda por soluções analíticas robustas, estes contextos de trabalho têm-se consolidado como áreas cruciais para a tomada de decisões baseadas em dados. Nesta revisão, serão discutidos os conceitos-chave relacionados com a Engenharia de Dados e *Business Intelligence*, bem como as abordagens teóricas e metodológicas adotadas por diversos autores.

2.1 Metodologia da revisão

Nesta secção é descrita, com detalhe, a execução de todos os passos da revisão de literatura executada para esta dissertação. A primeira subsecção explica o tipo de revisão utilizado e nas subsecções seguintes são explicados todos os passos que foram efetuados. Na última subsecção (2.1.5.2) são descritos os resultados finais desta revisão de literatura.

2.1.1 Tipo de Revisão

O tipo de revisão escolhido para esta dissertação foi a *Systematized Review*. Com este tipo de revisão é possível utilizar processos sistemáticos, como definição de *queries* e de critérios de inclusão, sem a necessidade de cumprir com os padrões rigorosos de uma *Full Systematic Review*. Este tipo de revisão envolve uma pesquisa abrangente e pode incluir uma avaliação formal ou informal da qualidade dos resultados, dependendo das necessidades. No caso desta dissertação, será utilizada uma avaliação informal da qualidade dos resultados, devido às restrições de tempo disponível para a elaboração deste trabalho.

Este método é o adequado para o projeto em questão, já que permite a pesquisa e seleção abrangente de publicações relevantes, com base em critérios de inclusão e verificação do conteúdo. A verificação do conteúdo das publicações resultantes é baseada na leitura dos títulos e dos resumos de cada uma. Ao aplicar técnicas sistemáticas de pesquisa e inclusão, o tipo de revisão escolhido permite uma síntese organizada de evidências e perceções, que podem apoiar as nossas

conclusões. Esta escolha também permite a eficiência em termos temporais, tornando-se prática e viável, dentro das limitações de uma dissertação de mestrado.

Uma *Systematized Literature Review* deve seguir alguns passos essenciais, que no caso são similares a uma *Systematic Literature Review*, mas mais flexíveis principalmente na pesquisa, avaliação e nas diretrizes de comunicação[1]. Os principais passos deste método de revisão de literatura são os seguintes[2]:

1. Enunciar a questão de investigação
2. Definição dos critérios de inclusão/exclusão
3. Seleção das bases de dados a pesquisar
4. Efetuar as pesquisas
5. Seleção dos estudos e avaliação da qualidade

O primeiro passo pode ser dividido em duas partes: escrita das perguntas de investigação e formulação das *queries* de pesquisa. Isto porque as *queries* são formuladas através da definição do nosso objetivo de investigação/pesquisa. A primeira parte foi descrita na secção 1.4 do capítulo 1.

2.1.2 Definição de Queries de Pesquisa

Foram definidas três *queries* considerando as áreas de investigação associadas a esta dissertação e os conceitos a serem explorados, no capítulo de revisão de literatura e estado da arte. As áreas de investigação em que esta dissertação se encontra inserida são *Business Intelligence*, *Data Engineering* e *Data Analysis*. As *query strings* resultantes são apresentadas na tabela 2.1.

Tabela 2.1: *Query strings*

Query
QS1 - ("business intelligence"OR BI) AND "BI tools"AND dashboards
QS2 - "data engineering"AND (ELT OR ETL) AND "business intelligence"
QS3 - "data modeling"and "data engineering"
QS4 - "data engineering"AND "data integration"AND "data analysis"

2.1.3 Definição de Critérios de Inclusão

Foram definidos critérios de inclusão para selecionar as publicações obtidas nas pesquisas efetuadas nos motores de busca. Na tabela 2.2, são apresentados os critérios de inclusão definidos e a respetiva justificação da escolha de cada um deles.

[1] ver https://mcw.libguides.com/SystematizedReviews - acedido em 30/10/2024
[2] ver https://mcw.libguides.com/SystematizedReviews/rightreview - acedido em 30/10/2024

Tabela 2.2: Critérios de inclusão

Número	Critério de Inclusão	Justificação
1	Acesso ao texto completo.	Sem acesso ao texto completo, não possuímos possibilidade de aceder à informação completa da publicação, podendo espoletar a falta de conceitos intermédios.
2	Publicação em inglês ou português.	Os idiomas compreendidos pelos investigadores são somente o inglês e português.
3	Contexto de Engenharia de Dados e/ou *Business Intelligence* e/ou Análise de Dados.	A publicação necessita abordar as áreas de estudo referidas, de forma a suportar a pesquisa dos conteúdos necessários, para a realização da revisão de literatura e estado de arte.
4	Somente publicações de trabalhos terminados.	Para a pesquisa ser suportada por informação de elaboração terminada e validada, necessitamos de publicações já finalizadas.
5	Artigo de revista, de conferência ou capítulo de livro, ou livro.	São os tipos de publicações mais fiáveis.

2.1.4 Seleção de Motores de Pesquisa

Para realizar as pesquisas para a revisão de literatura, foram selecionados 3 motores de pesquisa, para possuirmos variadas fontes de publicações, esperando atingir um alcance mais abrangente. Os motores selecionados e utilizados nesta revisão foram os seguintes:

- ACM Digital Library (`https://dl.acm.org` [3])
- Scopus (`https://www.scopus.com` [4])
- IEEE Xplore (`https://ieeexplore.ieee.org/Xplore/home.jsp` [5])

2.1.5 Pesquisa e Seleção de Publicações

Após a definição das *queries* de pesquisa, dos critérios de inclusão e dos motores de pesquisa, passamos a descrever o processo de pesquisa e seleção da literatura. Este processo resultou na lista de publicações a ser usada como base para a escrita do contexto da dissertação e como suporte ao desenvolvimento do Estado da Arte.

[3] acedido em 15/10/2024
[4] acedido em 15/10/2024
[5] acedido em 15/10/2024

2.1.5.1 Processo

Na figura 2.1, é apresentado um diagrama com o processo de pesquisa, seleção, registo e análise, das publicações.

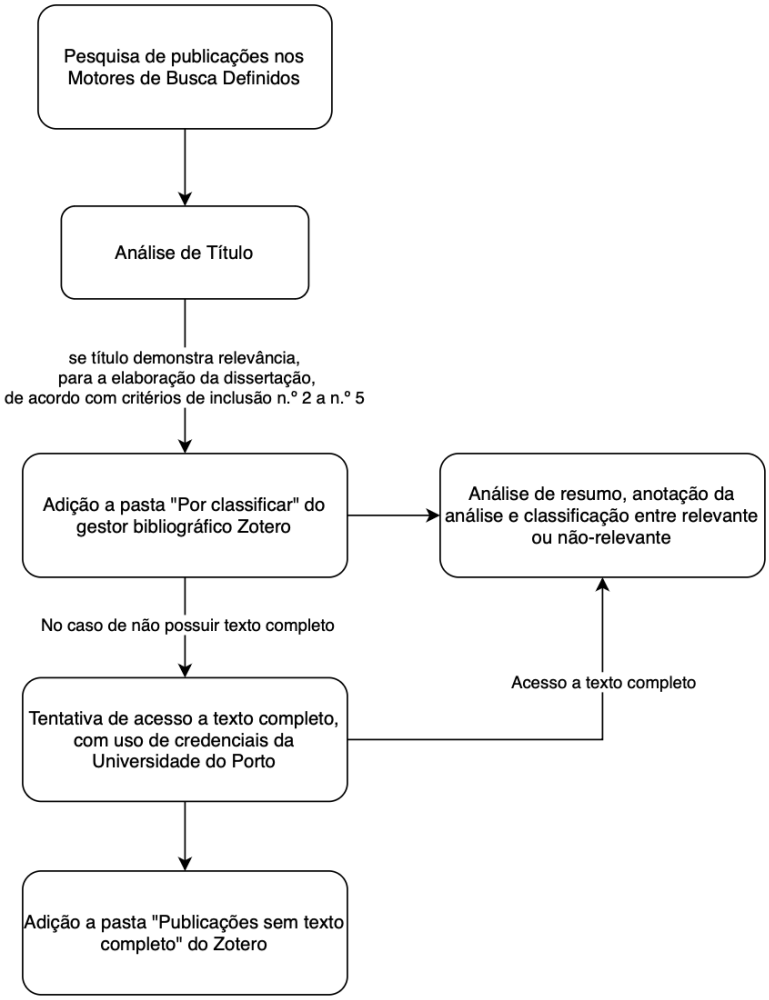

Figura 2.1: Processo de pesquisa e seleção de publicações

Foi utilizado o gestor bibliográfico Zotero[6] para organizar as publicações relevantes. Neste gestor, é possível organizar as publicações por pastas, adicionar etiquetas personalizadas e adicionar notas relevantes.

Este processo de pesquisa e seleção, começa com a análise dos títulos das publicações resultantes da pesquisa com a utilização das *queries* e posterior adição ao gestor bibliográfico Zotero (na pasta "Por classificar"), no caso de o título ser relevante para a elaboração da dissertação e a publicação cumprir com os critérios de inclusão n.º 2 a n.º 5. Após a inserção das publicações no Zotero, foi analisada a existência de acesso ao texto completo, de forma a cumprir com o critério de inclusão n.º 1. As publicações cujo texto completo não era acessível, mesmo com a utilização

[6] ver https://www.zotero.org - acedido em 27/10/2024

de credenciais da Universidade do Porto, foram colocadas numa pasta específica ("Publicações sem texto completo"), para posteriormente ser possível tentar contactar os primeiros autores de cada uma. Por fim, foi feita uma análise de todas as publicações que cumprem com os critérios de inclusão definidos; foram anotados pontos relevantes sobre cada uma e as publicações foram classificadas como relevantes ou não.

2.1.5.2 Registo dos Resultados

Com as *query strings* definidas e o processo de pesquisa e seleção estipulado, iniciámos a pesquisa, tal como definido. Na tabela 2.3 são apresentados os números de publicações resultantes, por cada *query*, em cada motor de pesquisa. No total, obtivemos 749 publicações, a partir dos resultados das *queries*.

Tabela 2.3: Publicações resultantes por *query string*

	QS1	QS2	QS3	QS4
Scopus	64	3	45	9
ACM	91	32	275	135
IEEE Xplorer	14	6	14	61

Contudo, após análise e seriação das publicações, tal como definido na secção 2.1.5.1, o resultado final foi de 40 publicações.

2.2 Conceitos Fundamentais

A evolução do tratamento e da análise de dados tem impulsionado o desenvolvimento de diversas abordagens, metodologias e tecnologias que permitem a extração de conhecimento e apoio à tomada de decisões. Esta secção apresenta os conceitos fundamentais que sustentam esta área, explorando definições, características e aplicações no âmbito dos sistemas de engenharia e análise de dados.

2.2.1 *Business Intelligence*

O conceito de *Business Intelligence* (BI) foi inicialmente proposto pelo Gartner Group[7], em 1989. O grupo apresentou BI, como um conjunto de conceitos e técnicas, para a elaboração de decisões empresariais, por meio de sistemas fundamentados em dados reais [13]. *Business Intelligence* integra metodologias, processos, arquiteturas e tecnologias, para converter dados brutos em informações valiosas, apoiando a tomada de decisões organizacionais. Por fornecer uma visão detalhada e prática dos dados, BI é fundamental para impulsionar o desempenho das empresas, pois ajuda a identificar novas oportunidades de mercado, a detetar potenciais riscos, a explorar perspetivas comerciais inovadoras e a otimizar os processos de tomada de decisões [12].

[7] ver https://www.gartner.com/en - acedido em 05/12/2024

Os dados servem de base para a informação, que, após análise, se transforma em conhecimento que suporta as escolhas e decisões dos utilizadores. No entanto, os dados por si só não são suficientes e, na maioria dos casos, devem ser processados para serem úteis [5]. As empresas estão cada vez mais interessadas em aceder a dados não estruturados e integrá-los com dados estruturados. No entanto, os dados não estruturados representam um desafio significativo, pois muitas vezes exigem bastante tempo para serem preparados e estruturados para análise. Os dados não estruturados representam entre 80 e 90 por cento dos dados gerados em todo o mundo, e a integração deste tipo de dados pode acrescentar um valor significativo às organizações. Além disso, o processamento e análise dos dados não estruturados, bem como a sua formatação e fusão com dados estruturados tradicionais, proporcionam uma visão mais completa do negócio aos utilizadores [6].

Apesar de os gestores e decisores confiarem fortemente na perícia e na intuição, muitas empresas constroem a sua estratégia com base em escolhas apoiadas em dados, em vez de confiarem apenas em intuições e/ou experiência. Os executivos podem utilizar os dados como um guia estratégico para identificar padrões que, de outro modo, poderiam não ser detetados. As empresas, com os recursos humanos com competências analíticas mais sofisticadas, têm um desempenho significativamente melhor do que as suas rivais [5].

As componentes fundamentais de BI são a análise descritiva (o que aconteceu) e a análise de diagnóstico (porque aconteceu). As duas principais componentes da análise avançada que constituem a base da análise empresarial são a previsão (o que é provável que aconteça) e a prescrição (o que deve ser feito) [5].

Outro aspeto essencial de BI é o seu fluxo de trabalho estruturado, que se organiza em cinco etapas principais: Coletar, Organizar, Analisar, Compartilhar e Controlar [26]. Este processo sistemático assegura a transformação eficiente dos dados em *insights* relevantes e está representado na figura 2.2. As etapas do fluxo de trabalho de BI são descritas da seguinte forma: [26]:

- Primeira etapa - Coletar: define-se o tipo de dados necessário e realiza-se a sua recolha, assegurando que os dados sejam apropriados para os objetivos da organização.
- Segunda etapa - Organizar: estrutura-se e classificam-se os dados recolhidos, facilitando as análises subsequentes.
- Terceira etapa - Analisar: organizam-se os dados e processam-se para identificar padrões, tendências e informações estratégicas.
- Quarta etapa - Compartilhar: disseminam-se as descobertas obtidas através de relatórios, *dashboards* ou outras ferramentas de comunicação.
- Quarta etapa - Controlar: aplicam-se e monitorizam-se ações baseadas nas análises, assegurando que os dados analisados resultem em melhorias concretas e sejam recorrentemente controlados, validando toda a sua integridade.

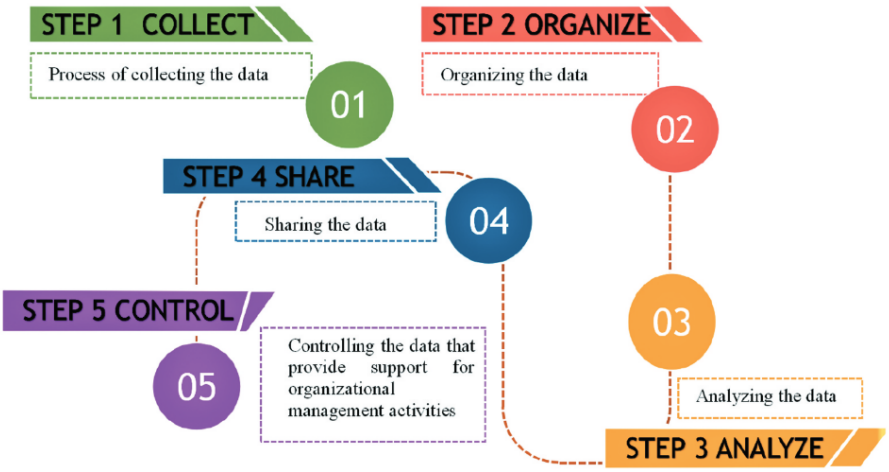

Figura 2.2: Fluxo de trabalho de *Business Intelligence* [26]

As soluções de BI combinam relatórios, análises, visualização e capacidades geoespaciais, para providenciar às organizações conhecimentos importantes que permitem decisões mais informadas, estratégicas e baseadas em factos [5]. Além disso, o campo de BI continua a evoluir, acompanhando tendências emergentes que visam enfrentar os desafios atuais. Entre estas tendências, destaca-se a automação de dados, que permite lidar de forma eficiente com os grandes volumes de informações geradas e coletadas diariamente. A governança de dados também se tornou uma prioridade, garantindo o cumprimento de normas e a utilização ética e segura dos dados. Por outro lado, a inteligência artificial (IA) explicável (*Explainable AI*) está a transformar BI, proporcionando maior clareza e melhorando a tomada de decisões, por meio de análises impulsionadas por IA. Por fim, a gestão da qualidade dos dados assegura a integridade, usabilidade e qualidade dos dados, e é um elemento crucial para organizações que dependem de informações confiáveis para as suas operações [26].

Arquitetura de *Business Intelligence*

O conceito de BI diz respeito a uma ferramenta digital que integra mecanismos para extrair, processar, armazenar e organizar grandes volumes de dados. Uma arquitetura de BI baseia-se em duas atividades principais (ver figura 2.3): *getting data in* e *getting data out*. A primeira é tradicionalmente conhecida como *data warehousing*, e envolve a movimentação de dados provenientes de diversas origens para um armazém de dados integrado, assegurando uma base consolidada para análises. Já a segunda abrange o acesso aos dados pelos utilizadores finais e aplicações analíticas, permitindo gerar *insights* e suporte às decisões organizacionais [8].

Na figura 2.3 é apresentada a proposta de arquitetura de um sistema de *Business Intelligence*, por Watson e Wixom [8], onde são referidas as várias camadas de um sistema de BI, que foram explicadas anteriormente.

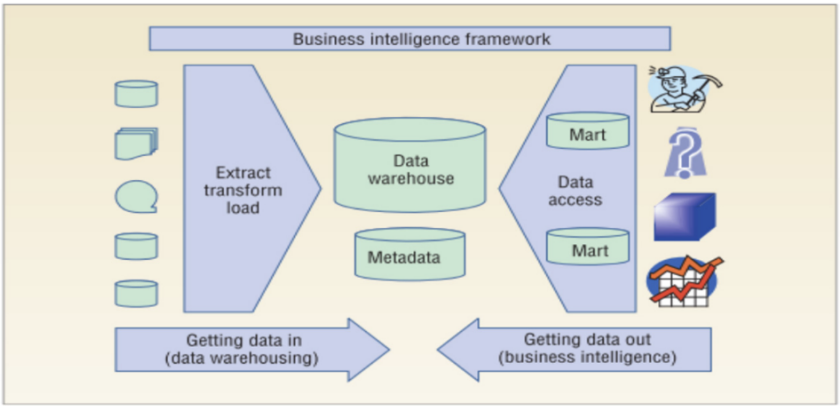

Figura 2.3: Proposta de arquitetura de BI, por Watson e Wixom [8]

2.2.2 Modelo de Dados Multidimensional

Um modelo de dados multidimensional é amplamente utilizado em sistemas de suporte à decisão, devido à sua capacidade de organizar e apresentar dados, de forma a facilitar análises complexas e detalhadas. Este modelo fundamenta-se em dois conceitos principais: dimensões e factos [7].

As tabelas de factos armazenam as métricas de interesse de negócio, como receita, preço e descontos, que constituem os valores numéricos analisados nas operações de negócio. As dimensões representam os atributos ou categorias, que podem ser utilizadas como filtros ou parâmetros na análise, como cliente, tempo e produto [7]. As dimensões fornecem o contexto necessário para interpretar os factos. Cada dimensão responde a perguntas fundamentais como "quem?", "o quê?", "onde?", "quando?", "porquê?"e "como?", permitindo uma análise aprofundada dos dados [8].

Os modelos multidimensionais podem ser classificados em duas categorias principais: estruturados, que incluem modelos relacionais e baseados em matrizes, e semiestruturados, como modelos baseados em XML. Dentro da categoria relacional, destacam-se o esquema em estrela (*Star Schema*) e os cubos de dados OLAP, que utilizam tabelas relacionais ou matrizes multidimensionais para organizar dimensões e factos. Nestes esquemas, a tabela de factos ocupa uma posição central, conectando-se a todas as tabelas de dimensões. Os valores armazenados na tabela de factos correspondem ao produto cartesiano das dimensões, permitindo uma análise detalhada das interações entre diferentes atributos do negócio [7].

2.2.3 Bases de Dados orientadas a Documentos

Tal como mencionado, no início da secção 2.2.1, os dados não estruturados têm vindo a crescer significativamente, e este aumento, aliado à complexidade do seu formato, apresenta novos desafios no contexto de *Business Intelligence* e *Big Data*. As bases de dados relacionais, amplamente utilizadas para armazenamento de dados estruturados, ao longo de décadas, já não se adequam

às exigências atuais, sobretudo devido à incapacidade de lidar com a velocidade, variedade e volume dos dados não estruturados. Para responder a estes novos desafios, surgem as bases de dados *Not-only-SQL* (NoSQL), que oferecem alternativas flexíveis e escaláveis às relacionais [6].

Entre os vários tipos de bases de dados NoSQL, destaca-se o modelo orientado a documentos, que é especialmente eficiente para lidar com dados semiestruturados. Este modelo organiza os dados no formato de documentos, encapsulando-os em estruturas como XML ou JSON. Ao contrário dos sistemas chave/valor simples, onde os dados são armazenados exclusivamente sob chaves primárias, as bases de dados orientadas a documentos permitem consultas em todos os componentes do esquema do documento. Essas bases de dados mantêm flexibilidade no armazenamento, uma vez que documentos com diferentes estruturas podem coexistir dentro da mesma coleção. Por esta razão, não existe necessidade de definir um esquema fixo para os dados, permitindo que cada documento funcione como uma unidade autónoma de informação e possibilitando o armazenamento de coleções heterogéneas [6].

Outro ponto relevante, nas bases de dados orientadas a documentos, é a capacidade de criação de índices, que incluem tanto os identificadores primários quanto as propriedades dos dados armazenados. Esta funcionalidade permite otimizar consultas específicas e facilita o acesso a informações em grande escala. As bases de dados orientadas a documentos mais populares, como Apache CouchDB, RavenDB e MongoDB, oferecem essa capacidade de consulta e recuperação eficiente, o que as torna uma escolha vantajosa para organizações que lidam com grandes volumes de dados não estruturados e que precisam de respostas rápidas. Em termos de desempenho, estas bases de dados são altamente otimizadas para operações de leitura, permitindo que conjuntos complexos de informações hierarquicamente estruturadas sejam recuperados rapidamente a partir de uma única chave [6].

2.2.4 Integração de Dados

A crescente necessidade de análises baseadas em dados atualizados ou em tempo real tem impulsionado a adoção de armazéns de dados em tempo real (RTDW). Estes sistemas representam um avanço significativo, mas também introduzem novos desafios na integração de dados. O primeiro desafio consiste em integrar dados em tempo real, o que envolve a execução dos processos de extração, transformação e carregamento (ETL) de forma contínua e eficiente. O segundo desafio está relacionado com a manutenção incremental em tempo real do modelo dimensional do armazém de dados. A manutenção incremental exige atualizações rápidas e precisas, sem comprometer o desempenho do sistema. Para extrair, transformar e carregar atualizações no armazém de dados, de forma instantânea, são fundamentais ferramentas de ETL em tempo real. Já no caso da manutenção incremental, apenas é necessário assegurar que os dados atualizados são propagados para o modelo multidimensional. No entanto, a escolha de um modelo estruturado ou semiestruturado influencia diretamente a eficácia e a latência deste processo, sendo um fator crítico na implementação de soluções RTDW [7].

2.2.5 Visualização de Dados

A visualização de dados, no contexto de *Business Intelligence*, desempenha um papel essencial ao traduzir informações complexas em representações visuais acessíveis para os utilizadores finais [8]. Também é fulcral na simplificação da análise de conjuntos de dados complexos, aliviando a carga cognitiva associada ao processamento de informações. Esta abordagem facilita a identificação de padrões e relações, tornando os dados mais acessíveis e memorizáveis [23]. Para além de acessível e percetível ao utilizador final, a visualização de dados deve ser interativa, de forma a conseguir tornar a experiência mais atrativa para o utilizador.

Utilizando as ferramentas de visualização de dados, os utilizadores podem monitorizar resultados em tempo real, avaliar o desempenho organizacional e identificar tendências ou problemas emergentes de forma proativa. Os *dashboards* sintetizam e agregam informações críticas, oferecendo uma visão integrada que facilita análises rápidas e baseadas em dados. A nível estratégico, as soluções de BI permitem a definição de metas e objetivos organizacionais (como indicadores-chave de desempenho (KPIs)), e a sua monitorização contínua, garantindo que a organização permaneça alinhada com os seus objetivos principais [8].

Um *dashboard*, que é normalmente uma combinação de vários visuais de BI, permite obter facilmente informações de várias fontes e é personalizável, apresentando dados gráficos e numéricos, num único ecrã. É possível obter conclusões a partir de dados tabulares utilizando recursos visuais como gráficos e diagramas, que não se conseguiriam obter de outra forma. No entanto, há que ter em conta que o sucesso dos *dashboards* depende da qualidade dos dados utilizados [5]. Os *dashboards* devem ser personalizados para atender às necessidades específicas de cada utilizador, uma vez que diferentes utilizadores necessitam de funcionalidades adaptadas às suas responsabilidades. É essencial que os KPIs relevantes sejam incorporados em cada *dashboard* para oferecer *insights* direcionados. Entre as principais vantagens dos *dashboards* estão a capacidade de processar e avaliar grandes volumes de dados, apresentar resultados em formatos de fácil compreensão, emitir notificações sobre métricas que ultrapassam limites predefinidos (prevenindo eventos adversos) e auxiliar na tomada de decisões mais eficazes e precisas. Além disso, os *dashboards* promovem uma gestão executiva mais orientada por dados, contribuindo para melhorias na eficiência e qualidade das operações das empresas [8].

Capítulo 3

Estado da Arte

O desenvolvimento de *Business Intelligence* (BI) e Engenharia de Dados tem sido fundamental para a transformação de dados em análises estratégicas, permitindo decisões mais informadas e ágeis nas organizações. A rápida evolução destas áreas, impulsionada pela crescente quantidade de dados e avanços nas tecnologias de armazenamento, processamento e análise, exige uma análise criteriosa das abordagens, metodologias e ferramentas atuais. Este capítulo aborda os principais conceitos e práticas de BI e Engenharia de Dados, com ênfase nas técnicas que suportam a construção de infraestruturas de dados eficazes e a criação de valor a partir dos dados.

3.1 Modelagem e Armazenamento de Dados

Em engenharia de dados e BI, a modelagem e armazenamento de dados constituem etapas fundamentais para estruturar, organizar e gerir grandes volumes de informação, de forma eficiente. Estes processos envolvem o desenvolvimento de modelos que traduzem a complexidade dos dados empresariais em estruturas lógicas e físicas, otimizando o armazenamento, a consulta e a manipulação dos dados. Atualmente, a escolha de abordagens adequadas para modelagem e armazenamento é essencial, já que os dados apresentam características diversas, incluindo diferentes formatos e níveis de estruturação. Nesta subsecção, são exploradas técnicas e tecnologias que suportam a modelagem de dados e o armazenamento adaptado às exigências atuais, abordando diferentes paradigmas.

3.1.1 *Data Warehouse* e *Data Mart*

Os conceitos de *Data Warehouse* e *Data Mart* (representados na figura 2.3) são fundamentais no contexto de BI, sendo ambos utilizados para armazenar e organizar dados de forma estruturada, com o objetivo de suportar análises e a tomada de decisões organizacionais. Embora estejam relacionados, possuem características e funções distintas.

Data Warehouse

O *data warehouse* (DW), ou armazém de dados, pode ser definido como um repositório centralizado de dados empresariais, projetado para integrar, consolidar e armazenar grandes volumes de dados provenientes de múltiplas fontes heterogéneas. Segundo Ralph Kimball, trata-se de um armazém de dados estruturado para analisar informações de todos os processos de negócio de uma organização, permitindo diferentes níveis de detalhe e suportando necessidades estratégicas de informação [3].

Estas bases de dados destinam-se a fornecer uma visão integrada e histórica dos dados, com foco em suportar a tomada de decisões. Diferentemente de uma base de dados transacional, um *data warehouse* é otimizado para consultas analíticas e relatórios, sendo considerado um conjunto de dados temáticos, que se alteram ao longo do tempo, mas não são voláteis [3].

Além disso, um *data warehouse* também pode ser descrito como um repositório estratégico ou multidimensional, projetado para consolidar informações relevantes para a organização num único modelo de negócio [3].

Data Mart

Um *data mart*, por sua vez, é um subconjunto especializado de um *data warehouse*. O principal objetivo é identificar e armazenar um conjunto mais específico de dados, para atender às necessidades analíticas de um departamento ou unidade de negócio específica [26].

De forma simplificada, um *data mart* pode ser considerado uma base de dados departamental, projetada para suportar análises mais direcionadas e específicas. Por exemplo, pode ser criado para armazenar e analisar dados relacionados exclusivamente ao desempenho de vendas ou às operações de *marketing* de uma organização. Este foco departamental possibilita consultas mais rápidas e direcionadas, em comparação ao *data warehouse* global [3].

Enquanto o *data warehouse* oferece uma visão ampla e integrada da organização, o *data mart* concentra-se numa área específica, sendo um derivado do *data warehouse* e otimizando os recursos de armazenamento e análise para fins particulares [3].

3.1.2 *Data Lakes*

Os *data lakes* têm surgido como uma solução para lidar com grandes volumes de dados, sem a necessidade de uma compreensão completa ou prévia da sua estrutura. Ao contrário dos armazéns de dados tradicionais, que exigem transformação e organização dos dados antes da ingestão, os *data lakes* permitem o armazenamento de dados em bruto, preservando-os na sua forma original. Essa abordagem não só reduz o esforço e o tempo de preparação dos dados, como também oferece flexibilidade analítica, uma vez que dados originalmente considerados "sem importância"podem ser reutilizados para novas necessidades. Além disso, ao manterem os dados iniciais, os *data lakes* fornecem uma “verdade” fundamental para as organizações, permitindo-lhes explorar o potencial de dados não estruturados e realizar análises mais profundas e diversas [12].

Uma das principais finalidades dos *data lakes* é funcionar como uma plataforma central para várias atividades analíticas e de preparação de dados, servindo diversos propósitos numa arquitetura de dados. Por exemplo, podem atuar como áreas de preparação ou fontes para armazéns de dados, onde a integração e limpeza de dados ocorrem antes de serem carregados num repositório mais estruturado. Além disso, servem como um espaço de experiências/testes, possibilitando a exploração de dados em estado bruto sem limitações estruturais. Outra finalidade dos *data lakes* é servirem como uma fonte direta para serviços *online* de BI, permitindo o acesso rápido e exploratório aos dados por vários utilizadores na organização [12].

Os *data lakes* possuem, no entanto, desafios próprios, especialmente em áreas como gestão, governança e recuperação de dados. A falta de organização prévia nos dados requer uma gestão cuidadosa para evitar que o *data lake* se transforme num *data swamp*, um *data lake* deteriorado e impossível de gerir onde dados se acumulam sem controlo ou organização adequada. A governança de dados também se revela crítica, já que as permissões e o controlo de acesso tornam-se complexos numa estrutura tão diversificada e flexível. Além disso, a utilização analítica dos dados em bruto exige competências técnicas avançadas, pois a compreensão e preparação de dados de formatos diversos exigem conhecimentos específicos. Para muitas organizações, o equilíbrio entre a flexibilidade e o rigor da governança de dados torna-se um fator essencial para o sucesso na implementação de soluções com *data lakes*, fazendo com que a sua utilização efetiva dependa de práticas de gestão rigorosas e de profissionais qualificados [12].

3.1.3 *Data Lakehouses*

A crescente complexidade e volume dos dados nas organizações modernas exige arquiteturas que consigam garantir não só a escalabilidade e flexibilidade, mas também a fiabilidade, a governança e a qualidade dos dados. Neste contexto, a arquitetura *data lakehouse*, ou simplesmente *lakehouse*, apresenta-se como uma solução inovadora e altamente eficaz, ao combinar os pontos fortes dos *data lakes* e dos *data warehouses* numa única plataforma unificada [1].

Tradicionalmente, os *data warehouses* são utilizados para armazenar dados estruturados e otimizados para consultas SQL e análises de BI, enquanto os *data lakes* se destacam por armazenar grandes volumes de dados — estruturados, semiestruturados e não estruturados — para fins como *machine learning*, análises preditivas ou processamento de *Big Data*. No entanto, a coexistência destas duas abordagens implicava a gestão de silos de dados separados, transferências constantes entre sistemas, processos complexos de ETL (*Extract*, *Transform*, *Load*), custos adicionais e riscos acrescidos de inconsistência e duplicação de dados [1].

A arquitetura *data lakehouse* surgiu como resposta direta a estes desafios, propondo-se a eliminar os silos de dados e a consolidar o armazenamento e processamento numa única infraestrutura. Este modelo permite armazenar dados brutos em armazenamento de baixo custo, como num *data lake*, ao mesmo tempo que oferece funcionalidades tradicionalmente associadas a *data warehouses*, como suporte a transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade), esquemas estruturados, controlo de qualidade e governança de dados [1].

Entre os principais benefícios desta abordagem destacam-se [1]:

- Arquitetura simplificada, ao consolidar os dados num único repositório.
- Redução de custos, ao evitar a duplicação de armazenamento e diminuir a complexidade dos processos ETL.
- Maior fiabilidade e consistência dos dados, devido à menor necessidade de movimentação e transformação entre sistemas.
- Melhoria na governança e segurança, facilitada por um ponto central de controlo.
- Flexibilidade para múltiplos casos de uso, incluindo análises em tempo real, *machine learning*, BI e ciência de dados.
- Alta escalabilidade, possibilitada pela separação entre armazenamento e computação, especialmente em ambientes *cloud*.

Embora se trate de uma arquitetura relativamente recente, com práticas ainda em consolidação, os benefícios operacionais e estratégicos da adoção de um *data lakehouse* superam largamente os desafios iniciais [1].

3.1.4 Estruturação de Dados

A estruturação de dados pode ser feita de duas maneiras: desnormalizada e normalizada. A compreensão dos conceitos de desnormalização e normalização é essencial para interpretar as características e a forma como os dados são organizados em modelos multidimensionais. Estes dois conceitos descrevem métodos distintos de armazenar dados, cada um com as suas vantagens e aplicações específicas [17].

3.1.4.1 Desnormalização

A desnormalização é amplamente utilizada no contexto de armazéns de dados e modelos *star schema*. Este método organiza os dados de forma a incluir não apenas as chaves primárias que identificam os registos, mas também informações adicionais relacionadas [17].

Por exemplo, numa tabela de vendas desnormalizada, além da chave para um certo produto (ProductKey), são armazenados detalhes do produto, como nome, categoria e cor. Essa redundância simplifica as consultas e elimina a necessidade de *joins* frequentes, otimizando o desempenho em cenários analíticos [17].

Na figura 3.1, observa-se um exemplo de tabela desnormalizada, onde tanto a ProductKey quanto as informações associadas aos produtos são armazenadas na mesma tabela [17]. Estas colunas, que representam informações associadas aos produtos, também poderiam ser chaves para dimensões, mas continuaríamos com uma tabela desnormalizada.

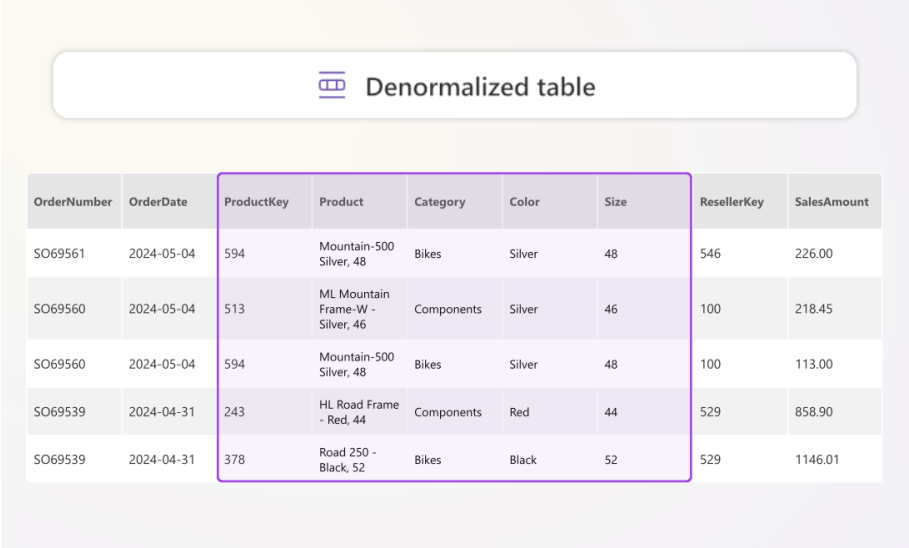

Figura 3.1: Exemplo de desnormalização [17]

3.1.4.2 Normalização

A normalização foca na eliminação de redundâncias e na melhoria da consistência dos dados, armazenando-os em tabelas separadas relacionadas por chaves primárias e estrangeiras. Esta abordagem é mais comum em bases de dados operacionais, mas também desempenha um papel em armazéns de dados onde a integridade dos dados é crítica [17].

Por exemplo, numa tabela de vendas normalizada, apenas as chaves dos produtos, como a ProductKey, são registadas. As informações dos produtos (nome, categoria, cor, etc.) são mantidas numa tabela separada e associadas por meio de relações [17].

Na figura 3.2, é possível visualizar uma tabela normalizada, onde a chave ProductKey apenas referencia os produtos, sem incluir informações adicionais [17]. A própria dimensão de produto, é que torna possível as relações com as restantes informações.

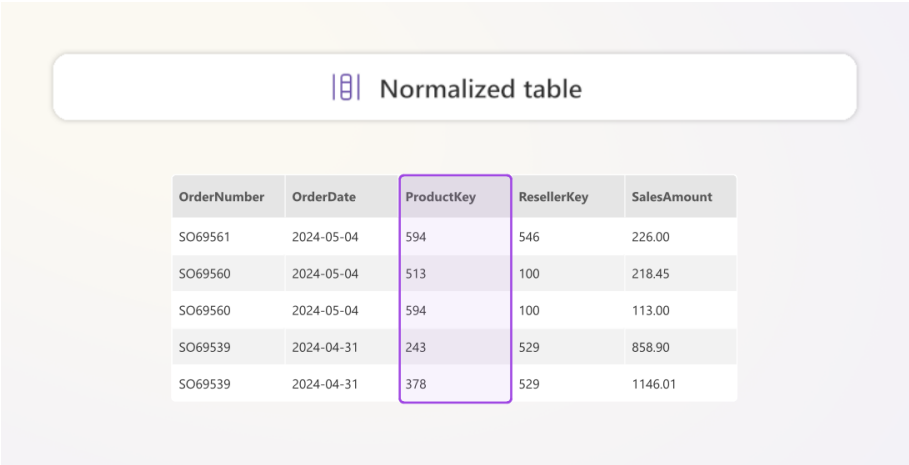

Figura 3.2: Exemplo de normalização [17]

3.1.4.3 Considerações

A desnormalização é frequentemente escolhida em armazéns de dados devido à sua capacidade de otimizar a leitura e simplificar as consultas analíticas, como no *star schema*. Em contrapartida, a normalização reduz a redundância e garante maior consistência nos dados, sendo ideal para sistemas onde a integridade dos dados é prioritária, como, por exemplo, em modelos *snowflake schema*. Ambas as abordagens são complementares e devem ser utilizadas conforme o contexto e os objetivos específicos do sistema [17].

3.1.5 Modelos de Dados

A conceção de um sistema de *Data Warehouse* (DW) eficaz exige a adoção de um modelo que organize os dados de maneira estruturada, com o objetivo de atender tanto às necessidades operacionais como às analíticas, da organização. Foram propostos vários modelos de dados, ao longo do tempo, destacando-se os desenvolvidos por Bill Inmon e Ralph Kimball, que abordam de formas distintas a estruturação, armazenamento e utilização de dados.

Esta secção descreve os dois modelos mais reconhecidos e utilizados: o modelo de Inmon e o modelo de Kimball. Em seguida, realiza-se uma breve comparação entre ambas as abordagens, destacando as principais características e diferenças.

3.1.5.1 Modelo de Inmon

Proposto por Bill Inmon, na década de 1990, o modelo de dados de Inmon adota uma abordagem "orientada para os dados", priorizando a construção de um *data warehouse* centralizado que armazena todas as informações da organização, independentemente das necessidades específicas dos utilizadores no momento [28].

Na figura 3.3, é apresentado um diagrama com a arquitetura do modelo de Inmon. Este modelo organiza os dados em quatro níveis principais [28]:

1. **Operacional**: Contém os dados transacionais usados nas operações diárias da organização.
2. **Atómico**: Representa o próprio *data warehouse*, no qual os dados são armazenados de forma integrada e consolidada.
3. **Departamental**: refere-se aos *data marts*, que oferecem uma visão orientada para temas específicos e são derivados do *data warehouse*.
4. **Individual**: engloba os relatórios, análises OLAP, *dashboards* e outras ferramentas que facilitam a exploração dos dados, pelos utilizadores finais.

A abordagem de Inmon utiliza processos de *Extract, Transform, Load* (ETL), para transformar os dados operacionais em informações integradas e consistentes dentro do armazém atómico. Os *data marts* são derivados diretamente do *data warehouse*, garantindo que os dados departamentais sejam sempre consistentes [28].

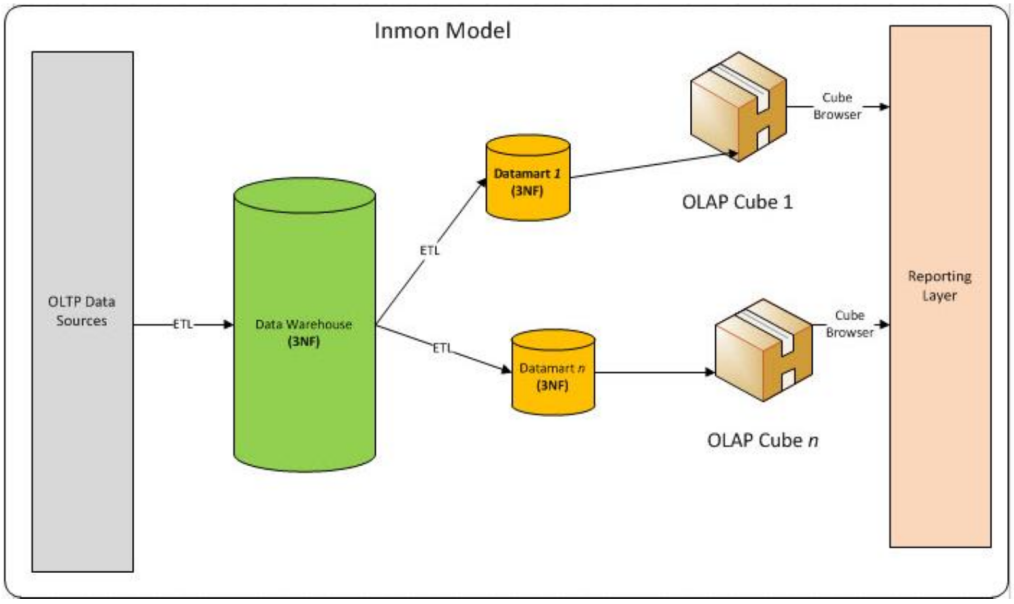

Figura 3.3: Modelo de Inmon [4]

Inmon defende que o *data warehouse* e os *data marts* devem ser armazenados de forma independente, mas interligados, com o *data warehouse* com a escalabilidade e rastreabilidade como prioridades, e os *data marts* orientados para o desempenho analítico e os requisitos específicos dos utilizadores [28].

3.1.5.2 Modelo de Kimball

A abordagem proposta por Ralph Kimball também surgiu na década de 1990 e é conhecida como "orientada para os requisitos do utilizador", uma vez que envolve os utilizadores finais desde as fases iniciais do projeto. Kimball prioriza a modelagem dimensional, utilizando dimensões conformadas para organizar os dados de maneira que atendam diretamente às necessidades analíticas [28]. Uma dimensão conformada, é uma dimensão que tem o mesmo significado, granularidade/-valores em diversas tabelas de factos [2].

Diferentemente da proposta de Inmon, Kimball considera o *data warehouse* como um conjunto de *data marts* integrados e consistentes, baseados em dimensões partilhadas. Esta visão multidimensional destaca os temas analisados, permitindo aos utilizadores explorar os dados sob diferentes perspetivas e obter uma visão aprofundada de cada área de interesse [28]. Na figura 3.4, é apresentado um diagrama que representa a arquitetura do modelo de Kimball.

A arquitetura do modelo de Kimball é desenhada para ser mais ágil, focando nas necessidades específicas de análise. Isto torna a implementação do modelo mais direta, embora a visão integrada dependa do rigor na definição e partilha da solução [28].

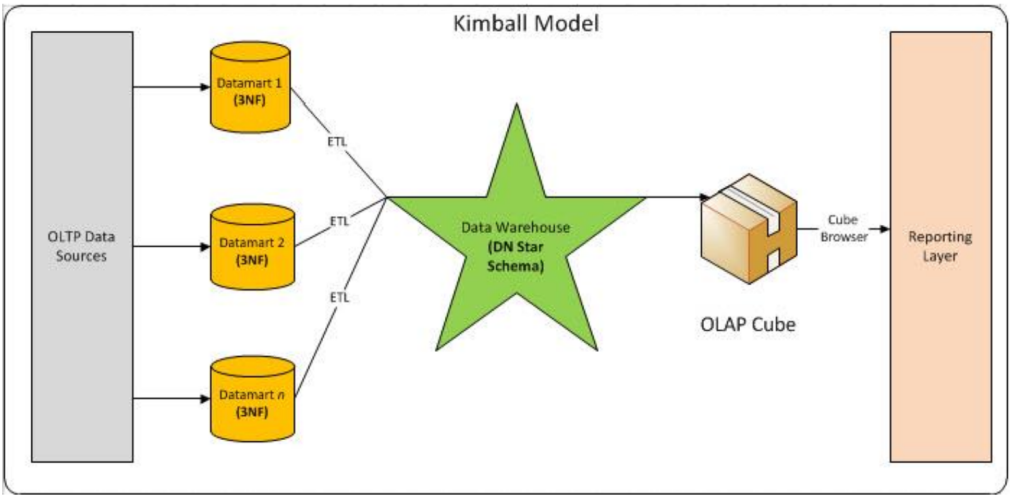

Figura 3.4: Modelo de Kimball [4]

3.1.5.3 Comparação

As abordagens de Inmon e Kimball apresentam diferenças marcantes na forma como estruturam e gerem o *data warehouse*, mas ambas compartilham o objetivo de consolidar e organizar os dados empresariais para apoiar a tomada de decisões. Apesar das suas diferenças, ambas utilizam processos de ETL, para integrar dados de múltiplas fontes, assegurando a consistência e a qualidade da informação armazenada [28].

Uma análise comparativa de ambas as abordagens revela diferenças em critérios filosóficos, metodológicos e técnicos, que são determinantes na escolha do modelo mais adequado para cada organização. A tabela 3.1 apresenta os principais pontos de distinção entre os dois modelos [28].

A abordagem de Inmon é mais indicada, quando os requisitos de análise não estão definidos ou quando é necessário um sistema robusto e escalável, para suportar múltiplas ferramentas de BI. Por outro lado, a abordagem de Kimball destaca-se pela sua simplicidade e alto desempenho em consultas, sendo ideal para organizações que possuem requisitos claros e bem definidos. Embora nenhuma das abordagens seja universalmente superior, a escolha entre elas depende das necessidades específicas de cada organização, bem como da sua maturidade em termos de gestão de dados e recursos disponíveis. Assim, a decisão deve ser tomada com base num entendimento aprofundado dos objetivos de negócio e da natureza das fontes de dados [28].

3.1.6 Esquemas de Dados

A organização lógica dos dados, num *data warehouse*, é um fator determinante para a sua eficiência e eficácia. Os esquemas de dados oferecem modelos estruturais, para organizar e representar a informação, possibilitando a realização de consultas analíticas rápidas e eficazes. Entre os esquemas mais utilizados destacam-se o *star schema* e o *snowflake schema*, cada um com características próprias que os tornam mais adequados a diferentes necessidades e contextos. Esta secção

Tabela 3.1: Comparação entre os modelos de Inmon e Kimball

Critério	**Modelo de Inmon**	**Modelo de Kimball**
Filosofia	Orientado para os dados, com foco no *data warehouse*.	Orientado para os requisitos dos utilizadores.
Estrutura	*Data warehouse* central, que fornece os dados para os *data marts* departamentais.	*Data warehouse* como um conjunto de *data marts* integrados.
Complexidade	Elevada, devido à modelagem detalhada e centralizada.	Moderada, com foco na simplicidade e rapidez.
Custos iniciais	Altos, mas com custos subsequentes reduzidos.	Relativamente baixos, mas custos equilibrados em cada etapa.
Tempo de implementação	Longo, devido à abordagem centralizada.	Curto, devido à implementação incremental.
Envolvimento dos utilizadores	Limitado nas fases iniciais.	Forte participação dos utilizadores finais.
Desempenho das consultas	Mais lento, devido à estrutura em normalizada.	Alto, devido à modelagem dimensional.
Flexibilidade	Maior estabilidade em sistemas com fontes de dados fixas.	Mais adequado para sistemas com requisitos bem definidos e estáveis.

explora as especificidades de cada modelo, incluindo os seus benefícios e limitações, e apresenta exemplos práticos para ilustrar as suas aplicações.

3.1.6.1 *Star Schema*

O *star schema* é um dos modelos mais utilizados em *Business Intelligence*. Está estruturado em torno de uma tabela central de factos que está conectada a diversas tabelas de dimensão [8]. Neste modelo, as tabelas de dimensão possuem uma estrutura desnormalizada, o que facilita a navegação e melhora o desempenho das consultas analíticas. Essa simplicidade estrutural é especialmente vantajosa em ferramentas OLAP, que se beneficiam de um modelo de dados que permite consultas rápidas e eficientes.

Na figura 3.5, é apresentada a estrutura pretendida num modelo *star schema*. É possível observar a tabela central de factos, com as várias tabelas de dimensões ao seu redor. A relação entre dimensão e facto é sempre de 1 para muitos (incluindo o 0).

O *star schema* é amplamente reconhecido pela sua facilidade de compreensão, tanto para analistas técnicos como para utilizadores de negócio, o que o torna uma escolha preferida em muitos projetos de BI. Outra característica importante deste modelo é a sua flexibilidade. Ao concentrar-se num único processo de negócio por tabela de factos, é possível criar um modelo modular que pode ser expandido para atender a diferentes necessidades analíticas (adicionando novas tabelas de factos e dimensões). No entanto, essa estrutura também apresenta desafios, como o potencial crescimento excessivo de dimensões ou a necessidade de garantir a consistência dos

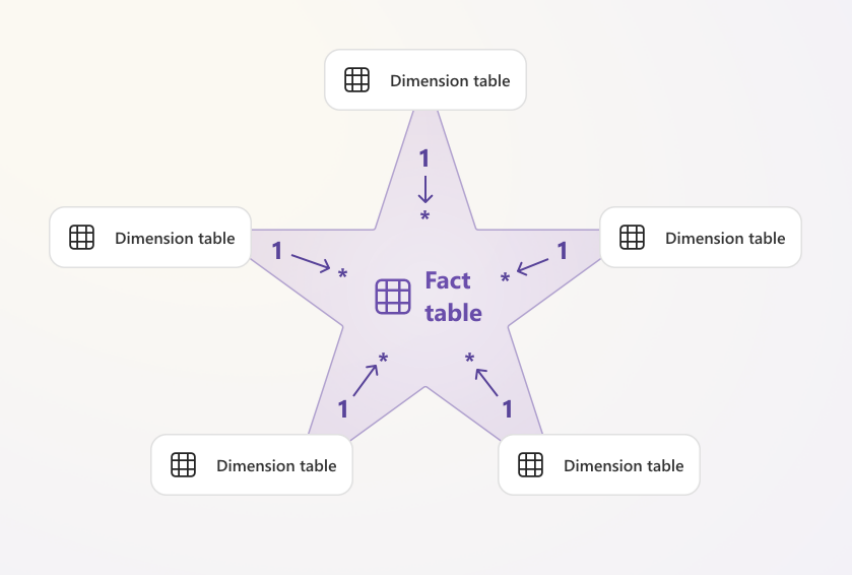

Figura 3.5: Modelo *star schema* [17]

dados ao longo de múltiplas tabelas de dimensão, especialmente em cenários de dados complexos e dinâmicos.

Na figura 3.6, é apresentado um exemplo de *star schema*, no qual existe uma tabela de factos central de vendas e 5 dimensões (Produto, Data, Revendedor, Funcionário e Território de Vendas), que categorizam os dados inseridos na tabela de factos.

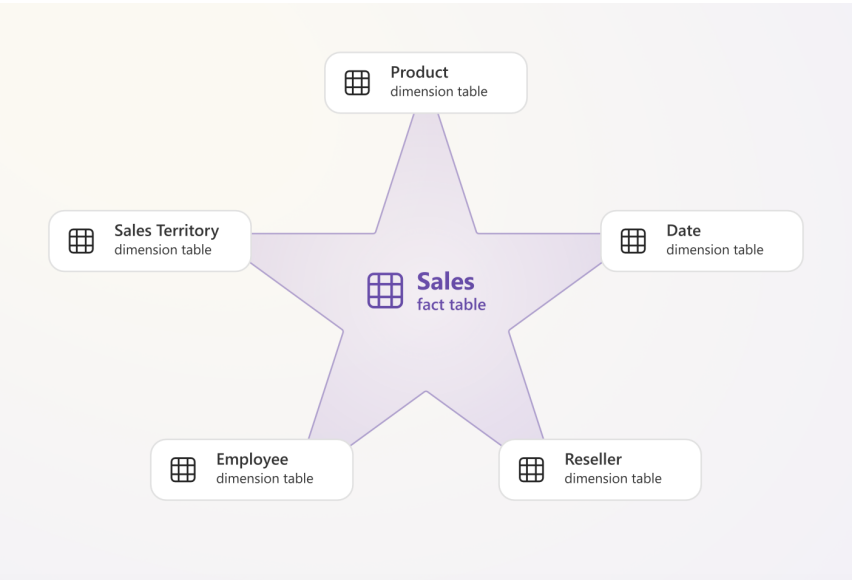

Figura 3.6: Exemplo de *star schema*, num contexto de vendas [17]

3.1.6.2 *Snowflake Schema*

O *snowflake schema* é uma extensão do *star schema*, que adota uma abordagem mais normalizada para a estrutura das tabelas de dimensão. Neste modelo, as tabelas de dimensão são divididas em várias tabelas relacionadas, organizadas de forma hierárquica. Por exemplo, num cenário de vendas, a dimensão de produtos pode ser subdividida em tabelas para categorias de produto, subcategorias e os próprios produtos. Essa estrutura cria uma forma que se assemelha a um floco de neve, daí o nome *snowflake*[17].

Esta normalização permite reduzir a redundância nos dados, já que informações comuns (como o nome de uma categoria de produto) são armazenadas uma única vez, em vez de serem repetidas em cada linha da tabela de dimensão. No entanto, esta abordagem pode ter implicações de desempenho e complexidade, em relação ao *star schema*.

Vantagens [17]

- Redução de redundância: a normalização minimiza o armazenamento redundante, o que pode ser benéfico em cenários onde o espaço é uma preocupação crítica.
- Facilidade de atualização: alterações em informações comuns, como a alteração do nome de uma categoria, precisam ser feitas apenas numa única tabela, garantindo maior consistência.
- Organização hierárquica: modelos normalizados podem refletir de forma mais natural as hierarquias do mundo real, como categorias e subcategorias de produtos.

Desvantagens [17]

- Complexidade aumentada: consultas que envolvem várias tabelas podem ser mais difíceis de criar e interpretar, especialmente para utilizadores menos técnicos.
- Impacto no desempenho: a execução de consultas pode ser mais lenta devido ao maior número de *joins* necessários entre as tabelas.
- Menor intuitividade: a multiplicidade de tabelas no modelo pode ser confusa para utilizadores de negócio, que não têm familiaridade com estruturas de dados normalizadas.

Na figura 3.7 é apresentado um exemplo de *snowflake schema*, no qual uma dimensão de produtos é subdividida em tabelas para categorias, subcategorias e os próprios produtos, num contexto de vendas.

3.2 Preparação de Dados

Atualmente, os dados que circulam dentro das organizações apresentam um nível de granularidade mais elevado e são gerados em volumes significativamente maiores do que no passado. O processo de preparação de dados desempenha um papel crucial ao integrar esses dados e assegurar

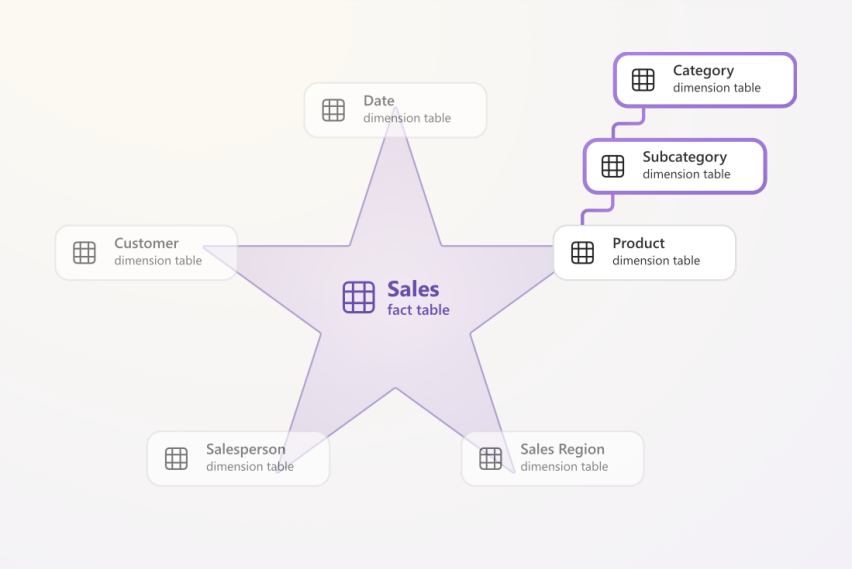

Figura 3.7: Exemplo de *snowflake schema*, num contexto de vendas [17]

a sua qualidade e relevância para a organização [8]. O processo de armazenamento de dados envolve a transferência de informações a partir de diversos sistemas de origem para um repositório central, com o objetivo de facilitar a análise e o suporte à decisão. Os dados extraídos passam inicialmente por uma área de preparação temporária (normalmente representada por uma base de dados (ou apenas um *schema*) distinta e denominada por *staging*), onde são organizados e tratados antes de seguirem para o armazenamento definitivo. A etapa de transformação, fundamental para a qualidade dos dados, aplica um conjunto de regras de negócios que converte esses dados em formatos consistentes e padronizados [12].

Um armazém de dados (*data warehouse*) é um repositório central onde dados extraídos de várias fontes são armazenados e organizados para facilitar relatórios e análises. Normalmente, os dados num *data warehouse* estão num formato desnormalizado, o que simplifica a sua consulta e análise em comparação com os dados nos sistemas de origem. Existem diferentes métodos de processamento de dados num armazém de dados. O mais comum é o ETL (*Extract*, *Transform*, *Load*), que consiste em extrair dados dos sistemas de origem, transformá-los para um formato adequado e, por fim, carregá-los para o armazém. Outro método, conhecido como ELT (*Extract*, *Load*, *Transform*), envolve carregar os dados diretamente no armazém antes de realizar a transformação. Ambos os métodos são fundamentais para preparar os dados para análises, garantindo a sua consistência e relevância para as necessidades de negócio [25].

Nas próximas secções, são abordados os métodos de ETL e ELT, discutindo as suas características, vantagens e limitações. Adicionalmente, é abordado o conceito de Reverse ETL, que é uma prática recente que visa exportar dados do armazém para sistemas operacionais, fechando o ciclo de dados e tornando-os acessíveis nos sistemas de origem.

3.2.1 ETL

O método ETL (*Extract*, *Transform*, *Load*) é um paradigma clássico para a preparação e integração de dados em sistemas de armazenamento de dados, e tem sido amplamente utilizado em infraestruturas de centralização de dados. Mesmo com a crescente adoção de tecnologias em *cloud*, o ETL continua a ser uma preferência para várias soluções de grandes empresas, especialmente em ambientes onde as infraestruturas existentes favorecem o uso deste método [25]. Este método combina três funções essenciais para preparar e gerir os dados de forma eficaz [25]:

- **Extração** (*Extraction*): Nesta fase inicial, os dados são extraídos de uma fonte de dados de origem, de modo a recolher a informação relevante, com o mínimo de impacto possível sobre o sistema de origem. O objetivo é maximizar a quantidade de informação capturada, sem afetar negativamente o desempenho do sistema de origem, em termos de tempo de resposta ou tempo de execução.

- **Transformação** (*Transformation*): Após a extração, os dados passam por um processo de filtragem e limpeza, onde são preparados para integração no armazém definido. Nesta etapa, os dados podem ser combinados com outras fontes, ajustados para eliminar duplicados, validados e convertidos para garantir consistência. A transformação também pode envolver a normalização, reestruturação e a verificação de consistência dos dados, garantindo que estejam prontos para análise.

- **Carregamento** (*Load*): A fase final envolve o armazenamento dos dados transformados no *data warehouse*. Os dados são adicionados na base de dados-alvo através de processos de inserção, frequentemente usando comandos SQL para adicionar os registos transformados como novas colunas ou entradas. Também podem ser utilizadas *stored procedures* para garantir operações atómicas, garantindo consistência na base de dados, pois evita a inserção parcial de dados.

O ETL organiza as tarefas de integração em três fases distintas, e desta forma permite uma estrutura robusta e controlada para a movimentação de dados, sendo uma abordagem eficaz para cenários com requisitos rigorosos de controle de qualidade e integridade dos dados.

3.2.2 ELT

O método ELT (*Extract*, *Load*, *Transform*) é uma alternativa ao ETL, que também visa a integração e preparação de dados para análise, mas com uma abordagem que aproveita melhor as infraestruturas de armazenamento em *cloud*. Assim como no ETL, o ELT começa com a extração de dados das fontes de origem. Contudo, no ELT, a fase de carregamento ocorre antes da transformação, o que permite que os dados extraídos sejam carregados rapidamente no armazém, sem serem modificados. Inicialmente, os dados carregados permanecem em formato bruto, ou seja, mantêm os mesmos nomes e estruturas dos segmentos das fontes de dados de origem, e não contêm novos campos ou informações adicionais [25].

Após o carregamento, a transformação é realizada diretamente no sistema de armazenamento, geralmente utilizando a capacidade computacional das tecnologias em *cloud* e controladores SQL locais. Com isso, as mudanças e requisitos de negócio são aplicados em tempo real, permitindo uma preparação rápida e eficiente dos dados, sem a necessidade de sobrecarregar o processo de integração. Esta abordagem é especialmente útil para lidar com grandes volumes de dados, uma vez que permite filtrar informações desnecessárias na fonte e otimizar o armazenamento ao eliminar linhas e colunas redundantes [25].

O ELT tira partido das tecnologias em *cloud*, que têm capacidade para processar e armazenar grandes volumes de dados de forma flexível e escalável. Esta abordagem reduz o custo e o trabalho necessário para manter os dados e facilita a análise de grandes conjuntos de dados, com menor manutenção. Com os avanços nas ferramentas nativas de integração de dados para alguns tipos de soluções (Hadoop e NoSQL), espera-se que o alcance do ELT continue a expandir-se [25].

3.2.3 Reverse ETL

O Reverse ETL é um método oposto ao ETL/ELT, onde o armazém de dados atua como fonte, em vez de ser o destino. Neste método, os dados armazenados no *data warehouse* são extraídos e transformados para atender às exigências dos sistemas de destino, antes de serem integrados em aplicações de uso direto por vários departamentos/equipas. Ao enviar dados de volta para sistemas operacionais, o Reverse ETL torna esses dados prontos para uso direto em ferramentas de trabalho, permitindo que os colaboradores os utilizem em tempo real para apoiar decisões e ações empresariais nas suas atividades diárias. Esta abordagem não substitui os métodos ETL ou ELT, mas complementa-os, ao fornecer um fluxo adicional de dados que permite alinhar diferentes áreas da empresa com as informações extraídas do armazém de dados [25].

Com o surgimento de uma nova geração de arquiteturas de dados, as empresas estão a reconhecer a importância de tornar os dados acessíveis a todas as equipas de forma integrada, em vez de os manter isolados, em silos centralizados. O Reverse ETL completa o ciclo de análise operacional, permitindo que os dados sejam integrados diretamente nas operações de diferentes áreas da empresa, tornando-se uma parte essencial dos métodos/tecnologias a adotar numa solução de dados moderna [25].

3.2.4 Tipos de Carregamento de Dados

Os tipos de carregamento utilizados no processo ETL desempenham um papel crucial na eficiência e desempenho de um armazém de dados, especialmente no contexto de estruturas que exigem elevados volumes de dados. O sistema ETL é tradicionalmente um dos componentes que consome mais recursos, em termos de tempo de processamento e esforço de implementação. Geralmente, as tarefas de carregamento podem ser classificadas em três categorias principais: o carregamento inicial, que é realizado quando o armazém de dados é preenchido pela primeira vez com um volume

exaustivo de dados; o carregamento completo (*full-load*), que substitui inteiramente os dados existentes no repositório; e o carregamento incremental (*partial-load*), que atualiza apenas os dados alterados ou recém-inseridos [7].

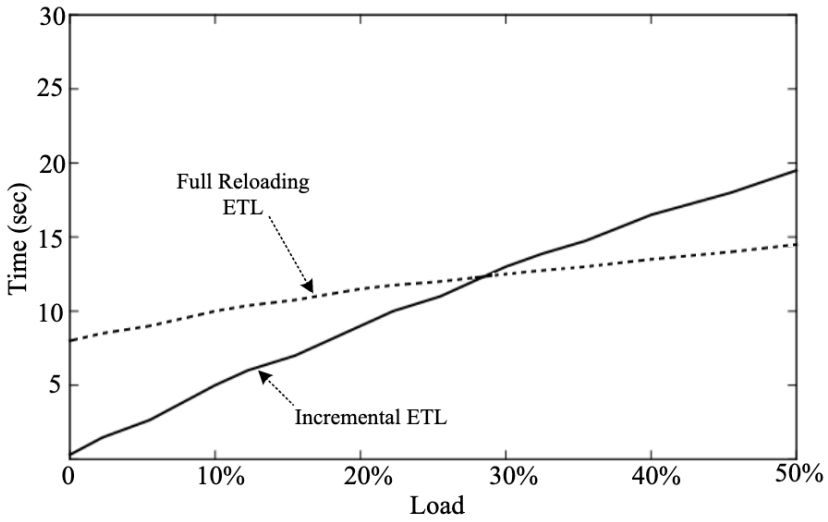

Figura 3.8: Comparação entre tipos de carregamento *Full-load* e *Partial-load* [7]

Observando a figura 3.8, conseguimos comparar o desempenho entre carregamento completo e carregamento incremental. Conseguimos perceber que em modelos multidimensionais relacionais (DMM), o carregamento incremental apresenta um desempenho superior, especialmente em cenários onde o volume de novas inserções é baixo. Dados experimentais ilustram que, enquanto o carregamento incremental continua eficiente mesmo com aumentos moderados na percentagem de novas inserções, o carregamento completo torna-se competitivo apenas quando o volume de inserções ultrapassa um limiar de aproximadamente 28%. Neste ponto, o desempenho entre as duas abordagens tende a inverter-se e o carregamento completo demonstra vantagens devido à maior proporção de dados novos em relação aos existentes [7].

Assim, a escolha entre carregamento completo e incremental deve considerar cuidadosamente o contexto operacional, o volume de dados e a frequência de atualizações necessárias, maximizando a eficiência do processo de integração de dados no armazém.

3.3 *Online Analytical Processing* (OLAP)

Os sistemas de *Online Analytical Processing* (OLAP) são projetados para fornecer uma visão orientada para a atividade dos dados, permitindo que gestores e analistas naveguem, explorem e analisem grandes volumes de informação de forma eficiente. Por serem baseados num modelo multidimensional, os sistemas OLAP facilitam a identificação de padrões e tendências [29].

Uma das principais características de OLAP é a utilização de cubos de dados, em vez das tradicionais tabelas. Estes cubos permitem uma organização hierárquica das informações, facilitando a navegação e a análise em diferentes níveis de granularidade. Nestes cubos existem dois

tipos principais de dados: métricas e dimensões. As métricas são os dados numéricos que são analisados, como totais, médias e quantidades, ou seja, resumidos. As dimensões são as categorias pelas quais as métricas são organizadas, como tempo, local ou produto, permitindo a análise dos dados sob diferentes perspetivas e níveis de detalhe. Por exemplo, ao analisar vendas, com OLAP pode-se facilmente mostrar totais de vendas por região ou país, além de permitir, com mais detalhe, visualizar quais lojas apresentam desempenho superior ou inferior [15].

3.3.1 Cubo OLAP

O cubo OLAP é o núcleo dos sistemas OLAP, sendo uma estrutura multidimensional, que permite processar e analisar dados, tendo em conta várias dimensões, de forma mais rápida e eficiente do que os armazéns de dados. Enquanto as tabelas de armazéns de dados são estruturadas de forma bidimensional, com linhas e colunas, armazenando registos individuais, o cubo OLAP expande esse conceito, permitindo a análise de múltiplas dimensões simultaneamente [10].

Numa tabela relacional, os factos ou dados de interesse (como totais de vendas) estão localizados na interseção de duas dimensões (por exemplo, região e vendas). No entanto, à medida que o volume de dados aumenta, as consultas em tabelas relacionais tornam-se mais lentas, e a reorganização dos resultados, para analisar diferentes dimensões, exige maior esforço. O cubo OLAP resolve esse problema, ao organizar os dados em camadas adicionais, cada uma representando um nível superior ou diferente da hierarquia das dimensões [10].

Por exemplo, se considerarmos uma dimensão localização, a camada superior pode organizar as vendas por região, e camadas adicionais podem detalhar os dados por país, distrito, cidade e até loja específica. Esse modelo permite que o cubo OLAP seja flexível e adaptável, podendo incluir as camadas que forem necessárias para a análise. Quando o cubo contém mais de três dimensões, é denominado hipercubo. Além disso, o cubo pode ter sub-cubos dentro das camadas, permitindo uma análise detalhada e eficiente [10]. Na figura 3.9 é apresentado um exemplo de um cubo OLAP, que vai de encontro aos temas que foram mencionados nesta secção.

Existem várias operações que podem ser realizadas num cubo OLAP, para analisar dados com diferentes perspetivas. As quatro principais operações são [10]:

- ***Drill-down*** - Esta operação permite aprofundar a análise, convertendo dados, de menor nível de detalhe, para dados mais específicos. Isso pode ser feito movendo-se para baixo, na hierarquia de conceitos de uma dimensão, ou adicionando uma nova dimensão ao cubo. Por exemplo, se estamos a visualizar os dados de vendas por trimestre fiscal, pode-se realizar um *drill-down* para ver as vendas detalhadas por mês, explorando a dimensão tempo, num nível mais baixo.

- ***Roll-up*** - O *roll-up* é a operação oposta ao *drill-down*, pois agrega dados dentro de um cubo OLAP, movendo-se para cima na hierarquia de conceitos ou reduzindo o número de dimensões. Por exemplo, se os dados de vendas estiverem organizados por cidade, ao realizar um *roll-up*, podemos agregar essas informações e visualizá-las por país, tendo uma visão mais ampla e consolidada dos dados.

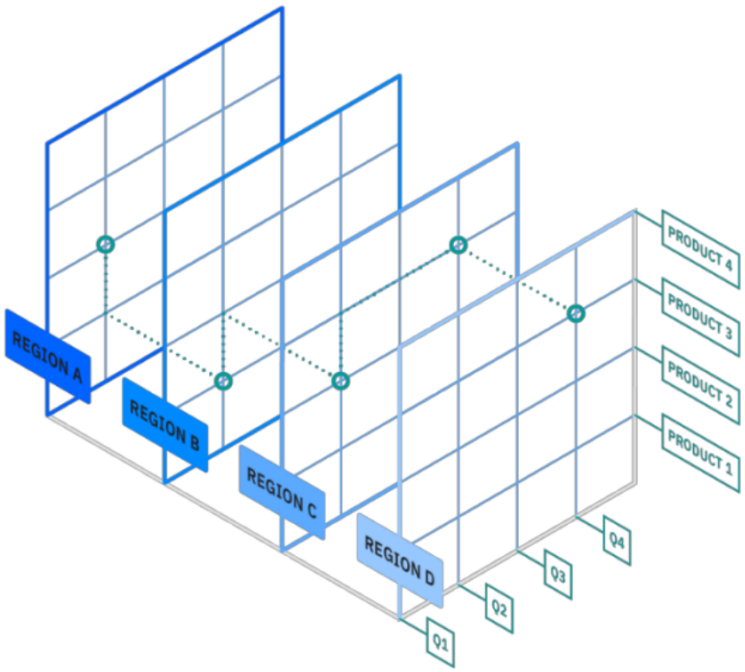

Figura 3.9: Exemplo de cubo OLAP [10]

- ***Slice* e *Dice*** - A primeira operação cria um subcubo, selecionando uma única dimensão do cubo principal, permitindo isolar um conjunto específico de dados. Por exemplo, ao realizar esta operação, conseguimos destacar todos os dados de vendas do primeiro trimestre fiscal de uma organização. Já a operação *dice* envolve a seleção de várias dimensões ao mesmo tempo, criando um subcubo mais complexo. Por exemplo, ao realizar este tipo de operação, podemos filtrar as vendas no primeiro trimestre fiscal e em países específicos, como Portugal e Espanha.

- ***Pivot***: A operação *pivot* permite alterar a visão do cubo, rodando o cubo sobre os seus eixo, sendo assim possível visualizar uma nova representação dos dados. Esta operação permite uma visualização dinâmica e multidimensional, permitindo que os utilizadores analisem os dados de diferentes ângulos.

Em suma, OLAP extrai os dados provenientes de um armazém de dados e reorganiza-os de forma multidimensional, resumindo todo o tipo de factos (valores numéricos), por dimensões, fazendo com que o processamento dos dados seja mais rápido e a análise seja mais eficiente [10].

3.3.2 Tipos de OLAP

O *Online Analytical Processing* pode ser implementado de diferentes formas, dependendo das necessidades específicas de armazenamento e análise de dados. Os principais tipos de OLAP são o **MOLAP** (OLAP Multidimensional), o **ROLAP** (OLAP Relacional) e o **HOLAP** (OLAP

Híbrido), cada um com características distintas, que os tornam mais adequados para diferentes cenários [10].

O MOLAP é o tipo mais comum e amplamente utilizado e funciona diretamente com cubos OLAP multidimensionais, providenciando análises rápidas e eficientes. A principal vantagem deste tipo é o desempenho, porque os dados são pré-processados e organizados em cubos. No entanto, esta abordagem é menos eficiente se tiver que lidar com conjuntos de dados muito grandes, devido às limitações de armazenamento dos cubos multidimensionais [10].

O ROLAP utiliza diretamente as tabelas relacionais para análise de dados multidimensionais, sem os reorganizar num cubo OLAP. Apesar da sua viabilidade para lidar com grandes volumes de dados, o seu desempenho pode ser prejudicado, principalmente em consultas mais complexas. Com ROLAP, as visualizações geradas são estáticas e não permitem a flexibilidade de manipulação multidimensional, que o MOLAP oferece, com as operações anteriormente mencionadas. Esta abordagem é recomendada em cenários onde o desempenho do sistema é menos importante do que o acesso a grandes quantidades de dados [10].

O HOLAP é uma solução híbrida das anteriores, em que os dados detalhados permanecem em tabelas relacionais, enquanto os cubos OLAP são utilizados para armazenar agregações e realizar processamentos. O HOLAP permite uma análise eficiente, oferecendo escalabilidade ao aceder a grandes volumes de dados e flexibilidade para realizar consultas detalhadas nos cubos. Contudo, esta solução implica uma arquitetura mais complexa, com maiores custos de manutenção, e não possui o mesmo desempenho, que MOLAP, no acesso direto a dados relacionais [10].

Na tabela 3.2, são resumidas as principais diferenças entre os três tipos de OLAP. Estas centram-se no desempenho necessário das consultas ao sistema e no volume de dados que é necessário armazenar.

Tabela 3.2: Comparação entre MOLAP, ROLAP e HOLAP

Critério	**MOLAP**	**ROLAP**	**HOLAP**
Desempenho	Muito eficiente nas consultas necessárias e na manipulação multidimensional.	Consultas pouco eficientes e falta de flexibilidade de manipulação multidimensional.	Análise eficiente e flexibilidade nas consultas, mas com mais custos de manutenção e menos desempenho que MOLAP.
Volume de dados	Possui limitações de armazenamento dos cubos multidimensionais.	Consegue lidar com grandes volumes de dados.	Escalável no acesso a grandes volumes de dados.

3.4 Tecnologias

A escolha das tecnologias adequadas é um fator determinante para o sucesso de qualquer projeto de *Business Intelligence*. As ferramentas e plataformas utilizadas desempenham um papel essencial em todas as etapas do processo, desde a extração, transformação e carregamento (ETL) dos dados até à sua análise e visualização. Nesta secção, serão exploradas as principais tecnologias aplicadas ao desenvolvimento e gestão de armazéns de dados, destacando as suas funcionalidades, vantagens e adequação a diferentes cenários empresariais.

3.4.1 Microsoft

Microsoft é uma das líderes globais no desenvolvimento de soluções tecnológicas avançadas, oferecendo uma ampla gama de ferramentas voltadas para a análise e gestão de dados, além de soluções de produtividade e colaboração em *cloud*. As tecnologias da Microsoft têm sido amplamente adotadas por organizações em diversos setores, permitindo a centralização de processos de dados, a criação de modelos analíticos e a implementação de soluções inteligentes, para otimização das decisões empresariais. Entre as principais ofertas, destacam-se o Power BI e o Microsoft Fabric, que fornecem ferramentas poderosas para a análise e integração de dados em tempo real. Ambas as plataformas fazem parte do ecossistema Microsoft, oferecendo soluções flexíveis e escaláveis. Esta secção explora as funcionalidades, vantagens e principais características destas ferramentas.

3.4.1.1 Power BI

O Microsoft Power BI é uma ferramenta de análise e visualização de dados em *cloud*, amplamente utilizada para monitorizar o desempenho organizacional e explorar dados, permitindo a obtenção de *insights* detalhados ou de alto nível. Esta plataforma integra um conjunto de funcionalidades que incluem a análise de dados, a conexão com diversas fontes de dados, bem como a criação de relatórios, gráficos, visualizações e *dashboards*. Estas capacidades são utilizadas para analisar indicadores de desempenho (KPIs) e métricas essenciais para o crescimento das organizações e para apoiar os processos de tomada de decisão [8].

Entre as principais vantagens do Power BI, destacam-se o seu baixo custo e facilidade de utilização, bem como atualizações frequentes que melhoram continuamente as funcionalidades. A ferramenta disponibiliza uma ampla diversidade de gráficos e visualizações pré-definidos, com *designs* atrativos que facilitam a adoção por utilizadores inexperientes. Além disso, permite a integração com várias fontes de dados, simplificando a extração e transformação de dados (ou simplesmente obtenção de dados da fonte definida). O Power BI também suporta fórmulas e expressões DAX (*Data Analysis Expressions*), possibilitando análises mais complexas, e oferece integração em *cloud*, promovendo uma utilização eficiente e colaborativa [8].

Outra característica notável do Power BI é a sua excelente integração com o ecossistema do Microsoft Office, o que aumenta a sua funcionalidade para utilizadores habituados às ferramentas

da Microsoft. No entanto, a ferramenta de desenvolvimento apresenta uma limitação relevante: está disponível apenas para o sistema operativo Windows, o que pode restringir a sua adoção em ambientes empresariais que utilizam outras plataformas [8].

A ferramenta de desenvolvimento do Power BI denomina-se Power BI Desktop e é a ferramenta mais amplamente utilizada, ocupando a posição de primeiro líder no Quadrante de Gartner[1] de *Analytics and Business Intelligence Platforms*[2] [8]. A edição de 2024 do quadrante é apresentada na figura 3.10.

Figura 3.10: *Gartner Magic Quadrant for Analytics and Business Intelligence Platforms 2024* [18]

Fluxo de Trabalho do Power BI

O fluxo de trabalho mais comum no Power BI começa com a conexão às fontes de dados, previamente definidas, no Power BI Desktop e criando um relatório (ficheiro .pbix) [22]. Um relatório pode ser composto por uma ou mais páginas/*dashboards*. Após a implementação do relatório, publica-se o mesmo relatório no Power BI Service e, após isso, o relatório está disponível para ser compartilhado com os utilizadores de negócio, para poderem visualizar e interagir com os relatórios no Power BI Service e em dispositivos móveis [22]. O Power BI Service[3] é a plataforma *online* do Power BI, onde são armazenados os modelos de dados (denominados por Modelos Semânticos, neste contexto) e os próprios relatórios Power BI. Na figura 3.11 é apresentado um esquema com o fluxo de trabalho descrito.

[1] ver https://www.gartner.com/en/research/methodologies/magic-quadrants-research - acedido em 05/12/2024
[2] ver https://www.gartner.com/en/documents/4247699 - acedido em 05/12/2024
[3] ver https://app.powerbi.com - acedido em 07-12-2024

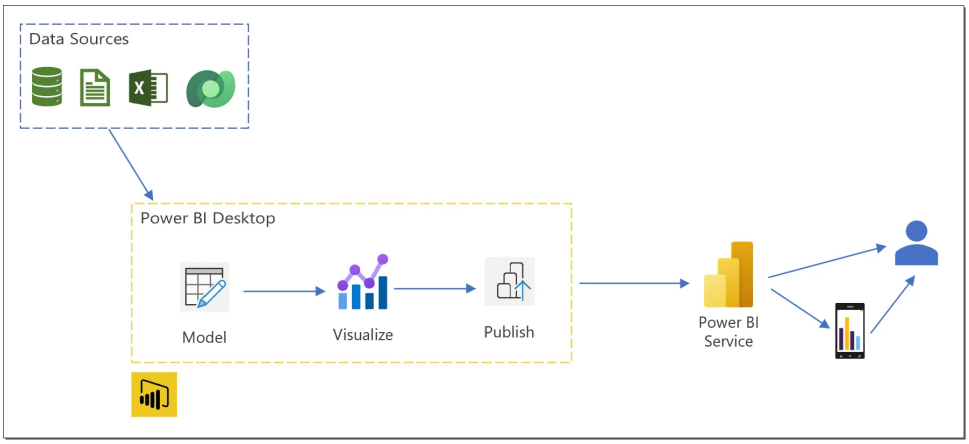

Figura 3.11: Fluxo de trabalho do Power BI [22]

Criação de Modelo Analítico no Power BI Desktop

Após obtermos os dados no Power BI, este permite-nos criar um modelo, adicionando as relações que pretendemos. Deste modo, conseguimos configurar o *dashboard*, para as tabelas de dimensões poderem filtrar as tabelas de factos. O funcionamento de qualquer filtro (a partir de conteúdos de dimensões), no Power BI, necessita da correta configuração da relação com a tabela facto. Na figura 3.12 conseguimos observar um exemplo de um modelo de dados (no caso *star schema*), configurado no Power BI, neste caso abordando um tema de vendas.

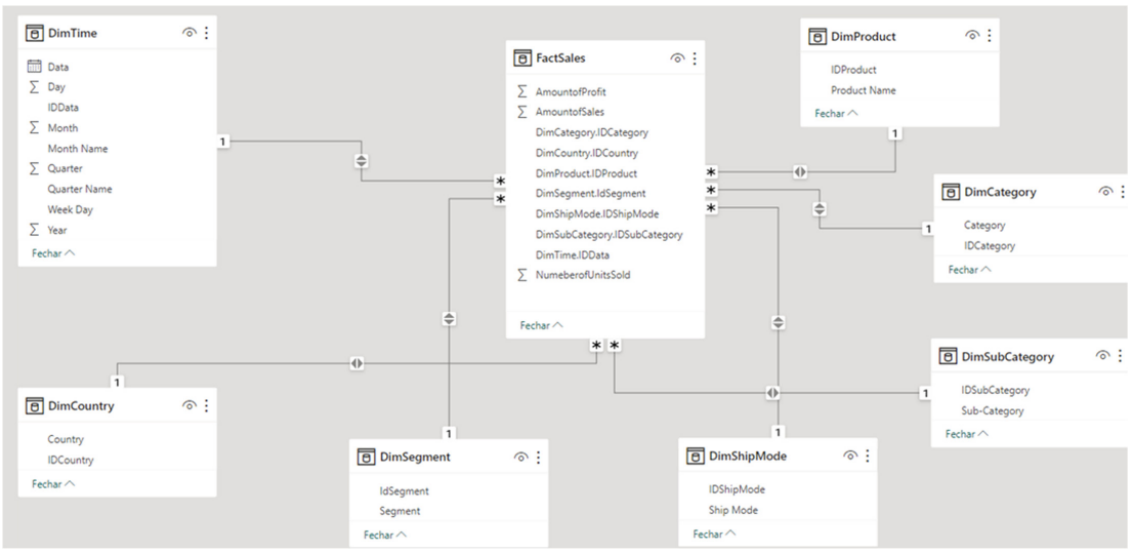

Figura 3.12: Exemplo de modelo *star schema*, configurado em Power BI [8]

Power BI *Dashboards*

Na figura 3.13, podemos observar um exemplo de um *dashboard*, em Power BI, com vários tipos de visuais, incluindo cartões no topo, com os respetivos KPIs. Também são incluídos dois

filtros de seleção (no formato de botões): um para o ano e outro para a categoria de produto.

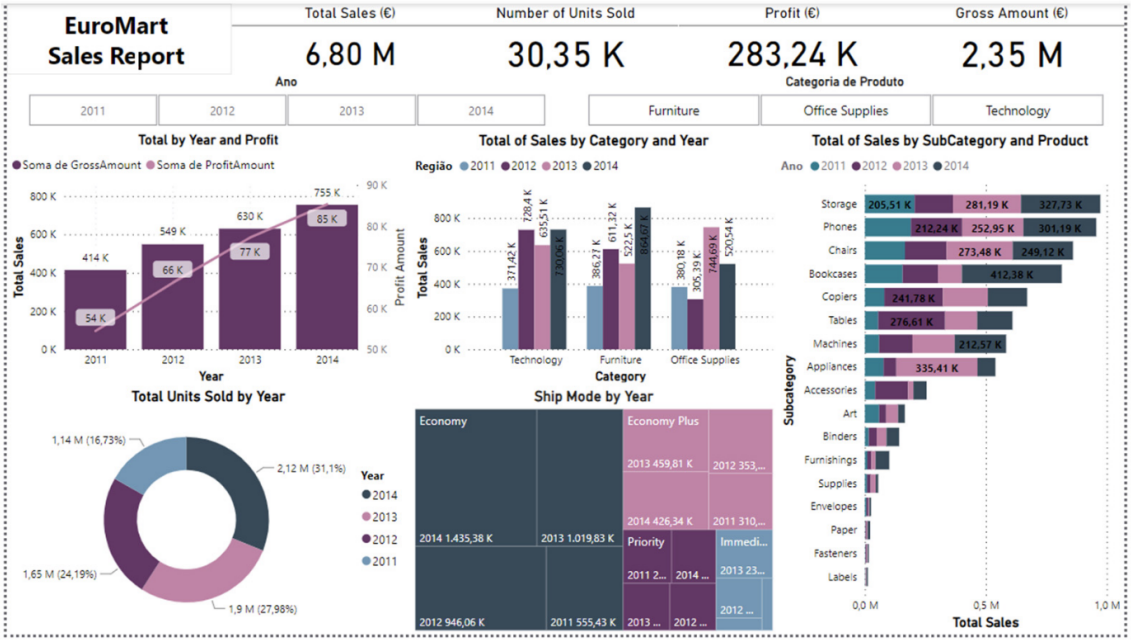

Figura 3.13: Exemplo de *dashboard* Power BI - 1 [8]

Na figura 3.14, é apresentado um segundo exemplo de um *dashboard*, em Power BI, com visuais distintos dos anteriores, sendo um deles um mapa, que neste caso apresenta a distribuição de vendas, por ano, em cada país. Neste *dashboard* são incluídos vários filtros que, neste caso, são *dropdowns*.

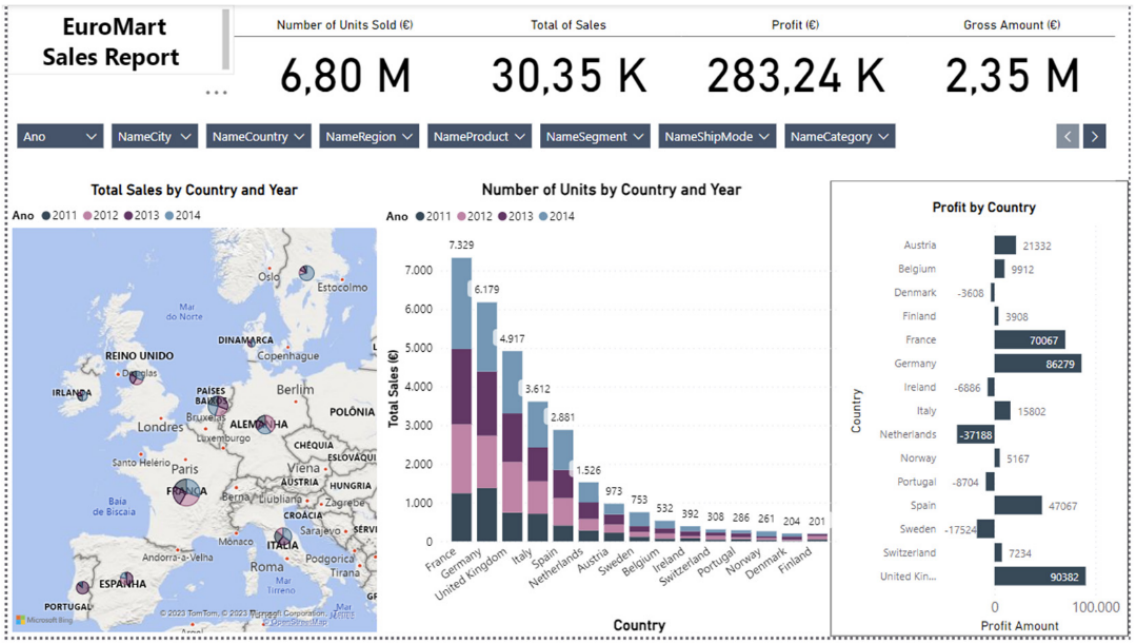

Figura 3.14: Exemplo de *dashboard* Power BI - 2 [8]

Os autores de [23] utilizaram o Power BI para transformar os dados originais, de forma a conseguirem estruturar o seu modelo, para posteriormente construírem as análises e visuais pretendidos. Nas limitações do trabalho deles, foi mencionado que uma delas foi, como limitação financeira, a extração dos dados, a partir das 4 fontes distintas, que pretendiam. Isto deve-se à ferramenta de transformação dos dados do Power BI (denominada Power Query) não ser eficiente na transformação de dados, com operações essenciais, como *joins*, por exemplo.

3.4.1.2 Fabric

O Microsoft Fabric é uma plataforma de engenharia e análise de dados, de ponta a ponta, em *cloud*, desenvolvida para oferecer uma solução unificada para empresas que necessitam de ferramentas integradas de movimentação de dados, processamento, ingestão, transformação, monitoramento de eventos em tempo real e criação de relatórios. Esta plataforma é projetada para simplificar e centralizar as diversas tarefas relacionadas à engenharia e análise de dados, oferecendo uma vasta gama de serviços destas áreas da Microsoft, como, por exemplo, Data Factory, Fabric Data Science, Fabric Data Warehouse, Real-Time Intelligence e bases de dados em *cloud* [16].

Uma das grandes vantagens do Fabric é sua integração num único ambiente, evitando a necessidade de combinar serviços de diferentes fornecedores. Em vez disso, a plataforma proporciona uma solução integrada e de fácil utilização, operando no modelo SaaS (*Software as a Service*), que simplifica a implementação e utilização das ferramentas. Além disso, permite a centralização do armazenamento de dados através do OneLake, um *data lake* unificado, que facilita a gestão e o acesso aos dados, mantendo-os no seu local original [16].

O Fabric também incorpora recursos avançados de Inteligência Artificial, com o Azure AI Studio, que facilita a criação e a implementação de modelos de IA e ML. Esta integração é fundamental para a automação de tarefas e para a criação de *insights* a partir de grandes volumes de dados. Além disso, o suporte ao Microsoft Copilot[4] oferece recursos orientados por IA, que ajudam os utilizadores com sugestões inteligentes e automatização de tarefas, melhorando a produtividade e a eficiência operacional [16].

Em resumo, o Microsoft Fabric é uma plataforma de dados poderosa e flexível, projetada para fornecer uma solução unificada em *cloud*, que combina várias funcionalidades avançadas de engenharia de dados, análise e IA, tudo num ambiente SaaS simplificado. Este conjunto de recursos torna a plataforma uma excelente opção para empresas que procuram uma solução para otimizar a gestão de dados e transformar grandes volumes de informação em *insights* acionáveis e estratégicos.

[4] ver https://www.microsoft.com/pt-pt/microsoft-copilot/personal-ai-assistant - acedido em 12/12/2024

3.4.2 Qlik

Qlik é um dos líders no Quadrante de Gartner[5] de *Analytics and Business Intelligence Platforms*[6], há 14 anos seguidos [19]. Podemos ver a sua posição na edição de 2024, do quadrante mencionado, na figura 3.10. Iremos abordar as duas tecnologias mais conhecidas da empresa: QlikView e Qlik Sense.

3.4.2.1 QlikView

QlikView é uma ferramenta que ajuda as empresas a compreender melhor os seus dados, permitindo análises rápidas e eficazes. Foi pioneira no conceito de "*in-memory data discovery*", que permite processar e analisar grandes volumes de dados diretamente na memória do computador. Esta abordagem elimina a necessidade de acessos frequentes ao disco, acelerando consideravelmente os tempos de resposta e otimizando a experiência dos utilizadores [27].

Uma das características distintivas do QlikView é o modelo de dados associativo, que permite aos utilizadores explorar as conexões entre diferentes elementos de dados de forma intuitiva, como se estivessem a montar um quebra-cabeças e a observar como as peças se interligam. Além disso, a ferramenta permite criar gráficos e relatórios interativos, que oferecem detalhes adicionais quando clicados, facilitando a análise detalhada e a descoberta de informações relevantes [27].

Outra funcionalidade importante do QlikView é a análise *ad hoc*, que permite aos utilizadores realizarem perguntas específicas sobre os dados em tempo real, promovendo uma exploração flexível e orientada às necessidades do negócio. A ferramenta também suporta a integração com várias fontes de dados, possibilitando uma visão unificada e abrangente [27].

Embora seja uma solução robusta e poderosa, o QlikView é frequentemente preferido em cenários onde se deseja maior controlo, por parte dos analistas, na criação de relatórios estruturados e detalhados, sendo uma escolha ideal para empresas que valorizam análises bem definidas e pré-configuradas.

3.4.2.2 Qlik Sense

Qlik Sense[7] é uma aplicação avançada de visualização de dados, projetada para atender às necessidades analíticas de grupos, departamentos ou de organizações inteiras. Esta ferramenta destaca-se pela sua capacidade de gerir grandes volumes de dados, permitindo uma colaboração eficaz entre utilizadores, em qualquer dispositivo. A versatilidade da ferramenta permite a criação de relatórios dinâmicos e tabelas interativas, que facilitam a análise e a apresentação de informações de forma visual e acessível [9].

Entre as principais vantagens do Qlik Sense estão uma interface moderna e intuitiva, baseada em *drag-and-drop*, que facilita a criação de gráficos e *dashboards*, e a autonomia dos utilizadores finais na realização de análises personalizadas e criação de relatórios, o que elimina a dependência

[5] ver https://www.gartner.com/en/research/methodologies/magic-quadrants-research - acedido em 08/12/2024
[6] ver https://www.gartner.com/en/documents/4247699 - acedido em 08/12/2024
[7] ver https://www.qlik.com/pt-br/products/qlik-sense - acedido em 08/12/2024

de suporte técnico contínuo. Um destaque é a presença de ferramentas robustas de *data storytelling*, que transformam os *insights* em narrativas claras e visuais, promovendo uma comunicação eficaz das descobertas [27]. Além disso, a aplicação suporta modelagem de dados e cálculos complexos, tornando-se uma solução escalável e poderosa para organizações de diferentes dimensões. A ferramenta é também multiplataforma, adaptando-se automaticamente entre versões *desktop* e *mobile*, o que facilita a utilização em diferentes contextos e dispositivos [9].

Anteriormente, o Qlik Sense apresentava uma limitação em relação à análise preditiva, que só podia ser realizada através de integrações com *software* de terceiros, por meio de ligações API (*Application Programming Interface*). Este fator era um desafio para organizações que necessitam de funcionalidades nativas de previsão de tendências [9]. No entanto, o Qlik Sense já possui a funcionalidade de análise preditiva e avançada, a partir do Qlik AutoML. O Qlik AutoML permite que a equipa de análise crie modelos de *Machine Learning* e previsões a partir dos mesmos, sem necessidade de programação [20].

Em suma, o Qlik Sense é uma solução robusta, flexível e inovadora na área de *Business Intelligence*, com capacidades que permitem melhorar a comunicação, a colaboração dentro das equipas de trabalho e a previsão de tendências no negócio.

3.4.2.3 Comparação

Aqui são apresentados alguns pontos, em resumo, sobre as duas tecnologias abordadas anteriormente, de forma a comparar as duas soluções [27].

QlikView:

- É a solução tradicional da Qlik, lançada antes do Qlik Sense.
- Pioneiro na descoberta de dados em memória, com foco em análises detalhadas e guiadas.
- Foca-se em relatórios e análises detalhadas e guiadas, ou seja, as aplicações são geralmente pré-definidas por desenvolvedores ou analistas.
- Oferece flexibilidade na criação de *dashboards* e análises, mas requer maior conhecimento técnico para a configuração e criação.
- Possui um ambiente mais estático, voltado para análises predefinidas.
- Ideal para utilizadores que preferem controlar o desenvolvimento e a distribuição de relatórios estruturados.

Qlik Sense:

- É a evolução do QlikView, desenhada para ser mais moderna e intuitiva.
- Oferece maior autonomia aos utilizadores finais, com foco em *self-service*.

- Apresenta uma interface baseada em *drag-and-drop*, facilitando a criação de análises e *dashboards* interativos.
- Facilita a colaboração em equipa, com integração nativa em dispositivos móveis e suporte multiplataforma.
- Inclui funcionalidades avançadas, como Inteligência Artificial (IA) e *Machine Learning* (ML), para previsões e *insights* automatizados.
- Mais orientado para análises dinâmicas e exploratórias, com menor dependência de suporte técnico.

Em resumo, enquanto o QlikView é mais adequado para análises estruturadas e detalhadas com maior controlo técnico, o Qlik Sense destaca-se pela flexibilidade, facilidade de uso e recursos modernos, atendendo a um público mais amplo e diversificado. Na figura 3.15 é apresentado um exemplo de um *dashboard* em Qlik.

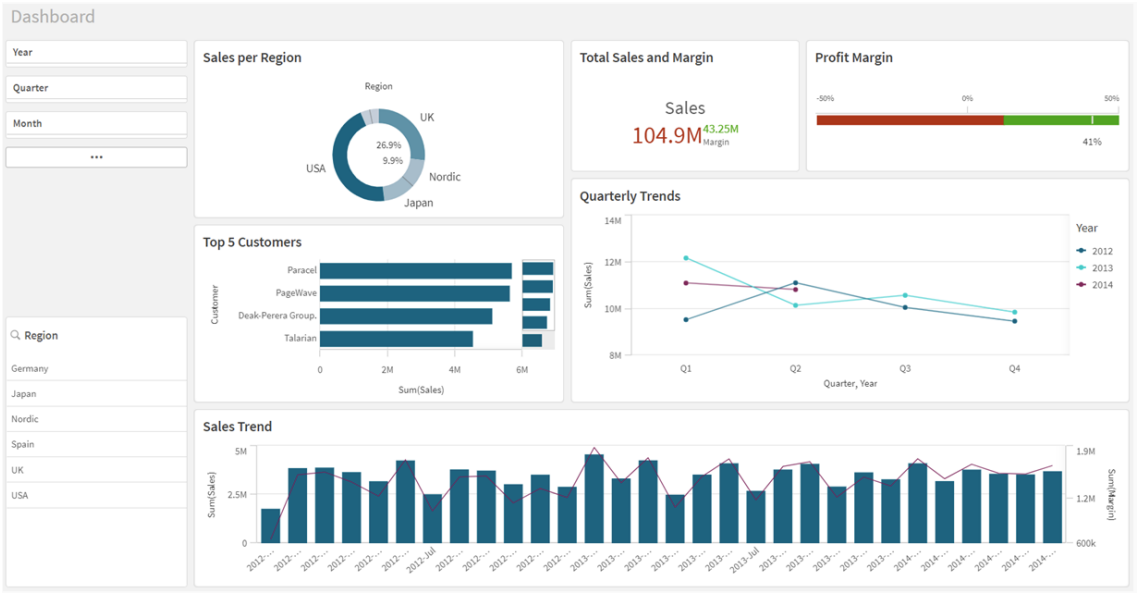

Figura 3.15: Exemplo de *dashboard* Qlik [21]

3.4.3 Tableau

Tableau é um sistema de *Business Intelligence* que auxilia as organizações na visualização de relatórios sobre o negócio. Os relatórios produzidos são de fácil compreensão e interligam diversas fontes. Tableau é versátil, intuitivo e confiável [26].

Uma das características distintivas do Tableau é o foco na simplicidade e usabilidade. A interface é intuitiva, pois funciona com movimentos de arrastar e largar, que permitem que qualquer utilizador, independentemente de conhecimentos técnicos, explore e analise dados sem necessidade de conhecimento técnico de informática. Esta abordagem simplifica o processo de análise

de dados, tornando-o acessível e rápido, sendo até 10 a 100 vezes mais rápido que sistemas tradicionais na integração e visualização de dados. Além disso, Tableau facilita o compartilhamento de informações através de navegadores *web* e dispositivos móveis, conectando-se a uma ampla variedade de fontes de dados [11].

O Tableau oferece uma gama de produtos adaptados a diferentes necessidades organizacionais, que estão representados na figura 3.16. Os produtos são [11]:

1. Tableau Desktop – Disponível em duas versões: a versão profissional, que suporta todas as fontes de dados e análise avançada, e uma versão individual, limitada a fontes como Excel e Access, sem suporte para servidores. Ambas as versões são populares em contextos profissionais e académicos.

2. Tableau Server – Uma solução flexível de *Business Intelligence* baseada na Internet. Permite criar visualizações no Tableau Desktop e distribuí-las dentro da organização, suportando múltiplos navegadores, dispositivos e plataformas.

3. Tableau Online – Uma versão simplificada do Tableau Server, focada na partilha de visualizações com colegas, parceiros e clientes, oferecendo 100 GB de armazenamento para licenças empresariais e facilitando respostas rápidas em qualquer lugar.

4. Tableau Public – Um recurso gratuito destinado à criação e partilha de histórias de dados *online*. É particularmente útil para profissionais que necessitam de disponibilizar conteúdos na Internet.

5. Tableau Reader – Uma aplicação gratuita que permite visualizar e interagir com visualizações criadas no Tableau Desktop, desde que as ligações necessárias tenham sido previamente configuradas.

Figura 3.16: Produtos Tableau [11]

Na figura 3.17, é apresentado um exemplo de um *dashboard* implementado em Tableau.

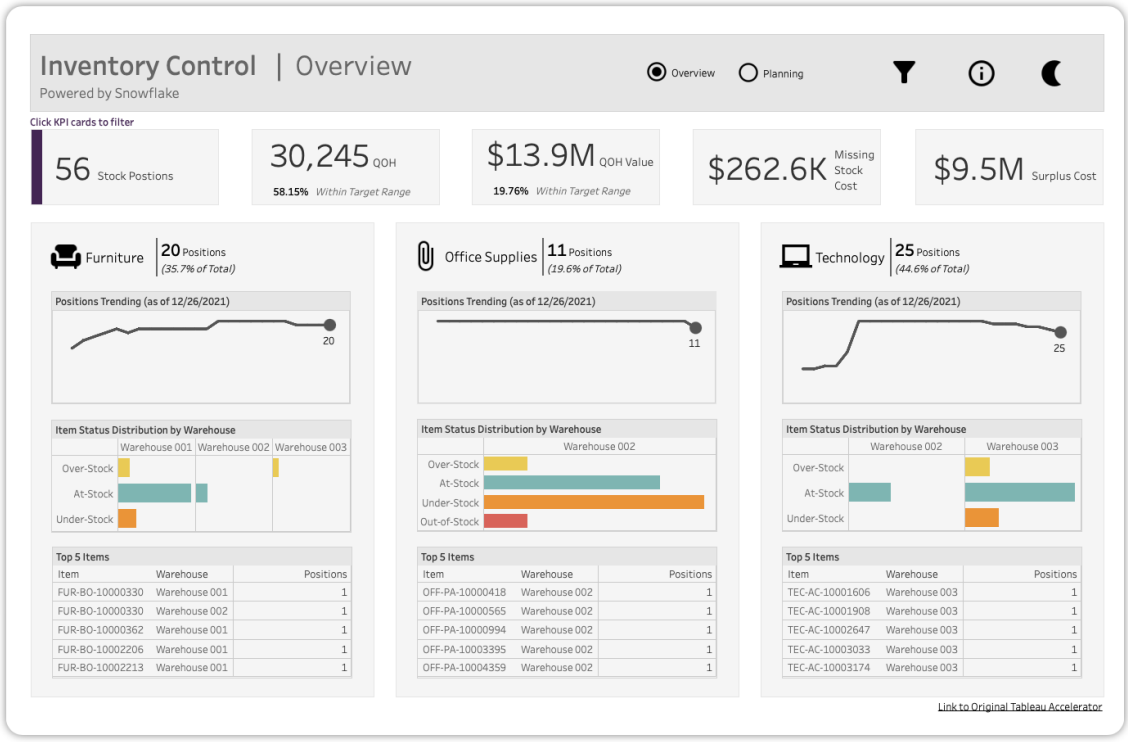

Figura 3.17: Exemplo de *dashboard* Tableau [14]

3.4.4 SAP

A SAP[8] é uma empresa alemã, que desenvolve software especializado para gestão de empresas. Esta empresa possui dois produtos para a construção/desenvolvimento de uma solução de *Business Intelligence* e estes são abordados nas duas subsecções seguintes (3.4.4.1 e 3.4.4.2).

3.4.4.1 SAP BW/4HANA

O SAP BW/4HANA é uma solução de armazenamento de dados, que fornece modelos pré-fabricados (blocos de construção) para padronizar a construção do *data warehouse* e apresenta uma modelagem de dados ágil e flexível. Esta tecnologia suporta uploads de *spreadsheets*, bem como outros tipos de uploads a partir de uma variedade de sistemas transacionais, bases de dados e ficheiros [5].

Para organizar e estruturar os objetos no SAP BW/4HANA, utiliza-se o conceito de *InfoArea*, que atua como um diretório onde se armazenam objetos personalizados específicos de cada projeto. No centro da modelagem de dados, encontram-se os *InfoObjects*, compostos principalmente por *Characteristics* (características) e *Key Figures* (indicadores-chave). As *Characteristics* representam dimensões de um modelo dimensional e permitem armazenar dados mestres e transacionais no *Data Warehouse*, de acordo com o modelo dimensional definido. Os *Attributes* das

[8] ver https://www.sap.com/portugal/index.html - acedido em 10/11/2024

Characteristics podem ser ativados como *Navigational Attributes*, permitindo a sua visualização e análise independente em qualquer consulta, o que facilita a modelagem e a personalização [5].

As *Key Figures* correspondem às métricas no modelo dimensional e fornecem os valores analisados nas consultas, como quantidades, valores monetários ou contagens. Além disso, o objeto *Advanced Data Store Object* (aDSO) desempenha um papel central na consolidação e armazenamento de dados, pois serve como destino para o carregamento de dados e proporciona uma interface que permite modificar o conteúdo de tabelas, sem comprometer a integridade dos dados [5].

Para unir dados de diferentes fontes, o *Composite Provider* permite realizar operações de união e junção entre os objetos *InfoProvider* do SAP BW (fontes de dados) e as *views* do SAP HANA, possibilitando consultas como se fossem objetos *InfoProvider*. No processo de ETL, a criação de um processo de transformação e de um *Data Transfer Process* (DTP) é necessária para mapear as *Characteristics* e *Key Figures* com as fontes de dados. O DTP espoleta a extração e carregamento dos dados para os *InfoProviders* [5].

No contexto da análise multidimensional, as *BW4 Queries* oferecem uma interface analítica robusta para consultas, adaptadas a diferentes ferramentas *frontend*, por meio de protocolos de integração específicos. Finalmente, os *Aggregation Levels* permitem a escrita e manutenção de dados transacionais diretamente no sistema OLAP do BW4, sendo utilizados como *InfoProviders* para modelar níveis que possibilitam alterações de dados de forma manual ou automática, conforme necessário para cumprir com os requisitos definidos [5].

3.4.4.2 SAP Lumira Designer

O SAP Lumira Designer é uma ferramenta de visualização de dados voltada para *Business Intelligence*, que permite a criação de aplicações analíticas interativas, desde *dashboards* de alto nível até análises OLAP detalhadas, acessíveis em qualquer dispositivo. Utilizando o modelo de dados armazenado no sistema SAP BW/4HANA (ver capítulo 3.4.4.1), o Lumira Designer facilita a visualização dos dados e promove a interatividade através de elementos de interface prontos para uso, como gráficos, tabelas cruzadas, mapas geográficos e componentes de filtro. A ferramenta também disponibiliza um conjunto abrangente de aplicações de análise, *templates* e exemplos para auxiliar na criação de conteúdos interativos [5].

No SAP Lumira Designer, um *Document* armazena toda a personalização de um projeto específico e pode ser guardado num servidor ou numa pasta local. As *Connections* permitem a integração com os sistemas de *back-end*, onde os dados e consultas fundamentais para o modelo de dados são definidos e armazenados. Para organizar os objetos de personalização, uma *Application* pode ser utilizada como um diretório para armazenar todos os objetos de um projeto [5].

As *Data Sources* representam o vínculo com o sistema de *Data Warehousing* do SAP BW/4HANA, onde o modelo de dados de origem é gerido. As consultas, que contêm os conjuntos de dados, filtros específicos a serem utilizados nos *dashboards* e funções analíticas do SAP Lumira, são criadas previamente no BW/4HANA e aplicadas nos objetos analíticos do Lumira Designer. Além disso, é

possível exibir e executar objetos de planeamento, como funções de cópia e distribuições, criadas no módulo de *Integrated Planning* do BW/4HANA [5].

Para organizar os componentes visuais, o objeto *Layout* permite estruturar *dashboards* e ferramentas analíticas, com separação por abas, para dividir diferentes aplicações analíticas conforme a exigência do projeto. Os filtros são essenciais em análises interativas, pois permitem que todos os componentes (visuais) dos *dashboards* sejam atualizados simultaneamente, com a seleção de algum valor, através da *interface* gráfica. Os filtros dinâmicos podem ser configurados com conhecimentos de *JavaScript* [5].

Na figura 3.18, é apresentado um exemplo de um *dashboard*, implementado no sistema SAP Lumira Designer.

Figura 3.18: Exemplo de *dashboard* SAP Lumira Designer [5]

3.5 Seleção de Tecnologias de BI

A escolha de uma ferramenta de *Business Intelligence* adequada para uma organização é um processo crítico que deve considerar múltiplos fatores, como as funcionalidades oferecidas, a arquitetura da solução, o custo associado e, principalmente, as necessidades específicas dos utilizadores. Entre as ferramentas analisadas, todas apresentam boas capacidades de gerar relatórios e apoiar decisões estratégicas, mas a melhor escolha dependerá do perfil e das exigências da organização [9].

Uma consideração essencial, ao selecionar uma ferramenta de BI, é entender as características dos utilizadores da organização, que podem ser classificados como produtores de informação (responsáveis por criar relatórios e análises) ou consumidores de informação (que utilizam os relatórios para tomar decisões). Estas diferenças devem guiar o processo de seleção, que pode seguir um caminho formal ou informal [9].

Processo Informal

O processo informal de seleção de uma ferramenta de BI é menos recomendado, pois geralmente resulta em decisões influenciadas por fatores externos, como estratégias de *marketing* das empresas fornecedoras, e não nas reais necessidades da organização. Este método pode levar a resultados inesperados, exigindo mudanças forçadas após alguns anos, causando perdas económicas e de tempo significativas [9].

Processo Formal

Por outro lado, um processo formal aumenta significativamente as probabilidades de escolher a ferramenta mais adequada ao negócio da empresa. Este tipo de seleção deve ser tratado como um projeto estruturado e incluir a criação de um comité de seleção, composto por membros de diferentes departamentos da organização. Este comité será responsável por [9]:

1. Identificar cenários e tipos de utilizadores, reconhecendo que diferentes utilizadores têm necessidades distintas de análises.
2. Definir os requisitos funcionais e técnicos para a solução de BI.
3. Analisar as funcionalidades oferecidas por cada ferramenta em relação aos requisitos.
4. Considerar o impacto da solução a curto e longo prazo.

Uma avaliação comparativa das tecnologias disponíveis permite compreender melhor as capacidades e limitações de cada solução. Ferramentas como Qlik Sense destacam-se pela sua escalabilidade e suporte a múltiplas fontes de dados; Power BI, pela integração com o ecossistema Microsoft e custo acessível; Tableau, pela experiência interativa e análises visuais intuitivas; e SAP, pela robustez em grandes organizações. Assim, a seleção da ferramenta deve alinhar-se aos objetivos estratégicos e operacionais da organização, garantindo a melhor relação custo-benefício [9].

3.6 Considerações Finais

Relembrando as perguntas de investigação definidas na secção 1.4, a pesquisa e a escrita da revisão de literatura e estado da arte foram executadas com o objetivo de perceber quais as melhores tecnologias no mercado, para a solução pretendida, e quais técnicas devem ser utilizadas nos vários processos/sistemas.

Tal como mencionado na secção 3.5, a escolha da(s) tecnologia(s) a utilizar, para desenvolver uma solução, deve ser um processo efetuado considerando diversos fatores. Desde a caracterização dos utilizadores finais, tipo de aplicação pretendida (*website*, aplicação *mobile*, *etc*) até à gestão de custos, por exemplo, todos estes aspetos devem ser bem considerados e é necessário um processo de reflexão e comparação das vantagens e desvantagens de cada tecnologia. Pelas razões

apresentadas, e tratando-se de um projeto definido pela ARMIS e pela empresa proponente, este processo de seleção será efetuado pelos responsáveis das duas empresas.

As técnicas utilizadas na solução também dependem das tecnologias definidas. Ao longo da escrita da revisão de literatura e do estado da arte, foram apresentados vários conceitos e técnicas globais necessárias para soluções de *Business Intelligence*. A pesquisa efetuada é útil para a especificação da arquitetura da solução e, posteriormente, para a sua implementação. Na especificação da arquitetura da solução, apresentada no capítulo seguinte, serão considerados todos os aspetos abordados na revisão de literatura e no estado da arte, com o objetivo de desenvolver uma solução baseada nas técnicas e tecnologias consideradas pela ARMIS como as mais apropriadas.

Capítulo 4

Análise e Especificação da Solução

Neste capítulo apresenta-se a solução a desenvolver, detalhando a sua estrutura, os componentes essenciais e as principais decisões arquitetónicas e tecnológicas. O objetivo é fornecer uma visão clara do contexto e da abordagem adotada, garantindo que a solução é eficaz, escalável e alinhada com as necessidades do projeto.

A especificação da solução inclui a descrição dos principais módulos, a organização do sistema e as estratégias utilizadas para a sua implementação. Desta forma, estabelece-se uma base sólida para a fase de desenvolvimento e integração.

Os requisitos do sistema/projeto foram definidos pelo diretor de projeto da ARMIS[1] e pelos gestores de projeto da parte do cliente.

O sistema foi projetado com base nas melhores práticas de utilização do ambiente em *cloud* da Microsoft, nomeadamente o Microsoft Fabric e o Microsoft Power BI, garantindo um produto final eficiente, flexível e de fácil manutenção.

4.1 Arquitetura do Sistema

A figura 4.1 apresenta o diagrama da arquitetura global do sistema a desenvolver. À esquerda estão representadas as fontes de dados. Estas são constituídas maioritariamente por ficheiros soltos, mas também por outras fontes, quer internas (do cliente) quer de dados públicos. Na secção central encontramos o armazenamento e transformação das várias etapas/formatos dos dados, seguindo uma arquitetura Medallion (ver 4.1.1. Por fim, à direita, é representada a camada final de análise de dados, constituída por um SQL *endpoint* (para possíveis consultas através de *querying* ou integração noutros sistemas) e os relatórios Power BI.

Para a implementação do sistema, irá ser utilizado o Microsoft Fabric. O Fabric possibilita a preparação, armazenamento, gestão e análise de dados, numa única plataforma e é abordado com mais pormenor na secção 3.4.1.2 desta dissertação. O *workspace* do Fabric irá alojar todos os componentes deste sistema, desde bases de dados, *scripts*, *notebooks*, fluxos de dados até relatórios

[1] Eng. Milton Nunes

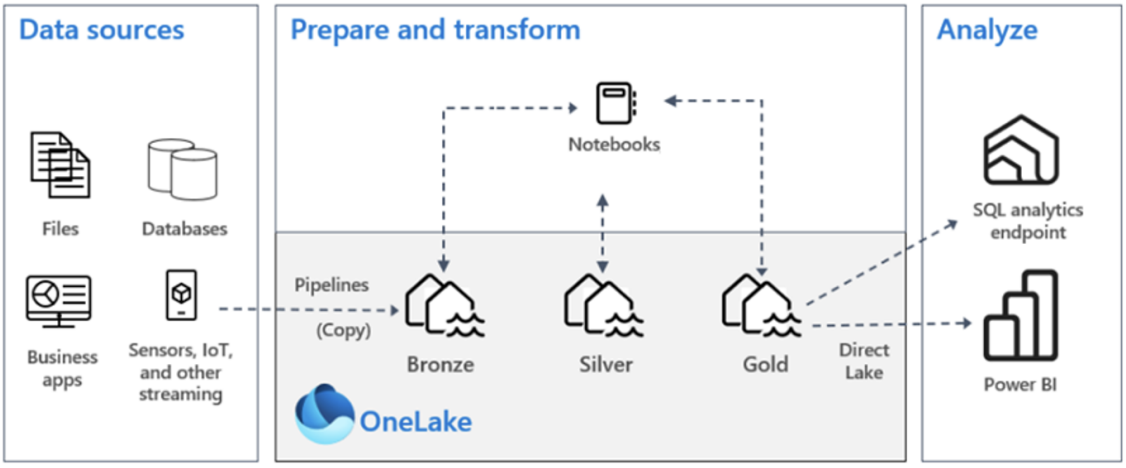

Figura 4.1: Arquitetura do sistema

Power BI, o que é uma vantagem em termos de organização da infraestrutura e componentes de desenvolvimento.

Para a utilização do Fabric, existe a necessidade de licenciamento, que depende da *performance* necessária/desejável e da quantidade/dimensão dos recursos. Considerando a solução a desenvolver, em termos de licenciamento da organização/projeto, foi recomendado ao cliente a capacidade F64[2], que inclui o Microsoft Copilot. Apesar de não existirem custos para utilizadores que apenas visualizam recursos, existe a necessidade de licenciamento Power BI Pro para os utilizadores que pretendam permissões de edição e de desenvolvimento[3].

4.1.1 Arquitetura *Medallion*

A arquitetura *Medallion* é um padrão de design amplamente utilizado em *data lakehouses*, com o objetivo de organizar os dados de forma estruturada e progressiva, garantindo uma melhoria contínua da qualidade e utilidade dos dados ao longo do seu ciclo de vida [24]. Esta arquitetura baseia-se na segmentação dos dados em três camadas (*Bronze*, *Silver* e *Gold*), tal como se pode observar na secção central da arquitetura do sistema, na figura 4.1.

A camada bronze é o ponto de entrada dos dados provenientes de sistemas externos, armazenados no seu formato original (*raw data*), complementados com metadados como data/hora de carregamento e identificadores de processo. Esta camada foca-se na captura de alterações (*Change Data Capture*), permitindo manter a linha temporal dos dados, garantir a rastreabilidade e facilitar eventuais reprocessamentos [24].

Na camada prata, os dados são limpos, validados e normalizados de forma mínima, aplicando transformações simples e sucintas para uniformizar formatos, remover inconsistências e garantir

[2] ver https://azure.microsoft.com/en-us/pricing/details/microsoft-fabric/#pricing - acedido em 08/04/2025. O custo mensal de infraestrutura está à data, entre 5.660,31€ e 9.518,80€, dependendo da forma de pagamento.

[3] ver https://www.microsoft.com/pt-pt/power-platform/products/power-bi/pricing?market=pt - acedido em 08/04/2025. O custo é de 13,10€, por mês, por utilizador.

qualidade. Esta camada oferece uma visão unificada dos principais conceitos e entidades do negócio, servindo de base para a análise empresarial. É nesta camada que se começa a consolidar os dados, mantendo a flexibilidade do modelo [24].

Por fim, a camada ouro é destinada à disponibilização dos dados prontos para consumo. Esta camada apresenta os dados já transformados, agregados e organizados segundo o modelo analítico definido. Aqui residem os *data marts*, orientados a relatórios e *dashboards*, com foco na performance de consulta [24].

A utilização da arquitetura *Medallion* neste sistema permite garantir uma organização clara e modular do fluxo de dados, facilitando a integridade, gestão e escalabilidade da plataforma. Este padrão suporta a evolução gradual da maturidade dos dados, desde a ingestão bruta até à disponibilização no formato final (otimizado para consulta), sendo ideal para ambientes com dados de múltiplas origens e naturezas distintas. Esta abordagem também se alinha com as boas práticas de engenharia de dados modernas, como o paradigma ELT, promovendo maior eficiência e controlo sobre o ciclo de vida dos dados.

4.1.2 Armazenamento de Dados

Em termos de armazenamento, são necessárias duas bases de dados: uma que guarda os dados analíticos e outra para guardar dados de controlo que gerem todos os processos do sistema. Foi decidido usar um *data lakehouse* para guardar os dados do modelo analítico e uma base de dados SQL *standard* para os dados de controlo.

A escolha de uma arquitetura *data lakehouse* para este projeto justifica-se pela sua capacidade de unir as vantagens dos *data lakes* e dos *data warehouses* numa única plataforma. Enquanto os *data warehouses* são otimizados para análises estruturadas e consultas SQL, e os *data lakes* permitem o armazenamento massivo de dados em bruto, um *data lakehouse* combina ambos os modelos, eliminando silos e simplificando a gestão dos dados. Esta abordagem permite armazenar dados estruturados e não estruturados de forma escalável e económica, assegurando simultaneamente integridade, gestão e desempenho analítico.

No *data lakehouse*, o armazenamento dos dados funciona com base num sistema de ficheiros. Estes ficheiros podem ser *delta tables*, *parquets* e outros, dependendo da fonte de dados. Na figura 4.2, é apresentada a estrutura de pastas do *data lakehouse* a implementar.

Na camada de bronze (primeira pasta), os ficheiros podem ser de vários formatos, pois são uma cópia exata dos ficheiros originais. Já na segunda camada (prata), os dados passam a ser armazenados em ficheiros *parquet*, pois apenas são preparados dados inicialmente em formatos estruturados, como Microsoft Excel e CSV, e apenas estes transitam para esta camada. Por último, na camada de ouro, são armazenados os dados na estrutura final, em *delta tables* que podem ser consultadas a partir do SQL *endpoint* disponível.

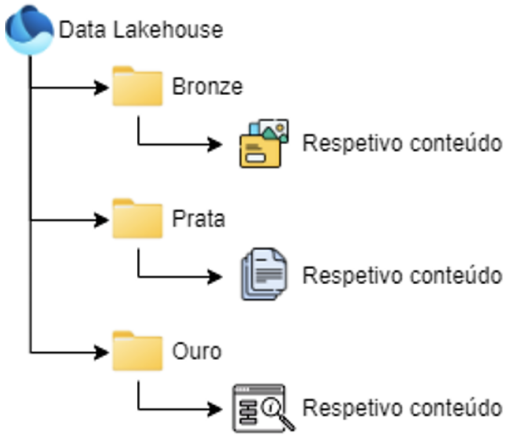

Figura 4.2: Sistema de ficheiros do *data lakehouse*

Parquet

A escolha do formato *parquet* para os ficheiros da camada prata prende-se com a sua eficiência no armazenamento e na leitura de grandes volumes de dados. *Parquet* é um formato colunar otimizado para consultas analíticas, permitindo leituras seletivas sem necessidade de percorrer o ficheiro completo. Ao contrário de ficheiros como CSV, que armazenam os dados linha a linha e exigem a leitura sequencial para aplicar filtros, os ficheiros *parquet* organizam os dados por colunas, permitindo que o motor de consulta aceda apenas às colunas relevantes e salte blocos de dados irrelevantes. Esta estrutura colunar também facilita a compressão e codificação eficientes, reduzindo significativamente o espaço em disco e o tempo de leitura. Este tipo de ficheiro não é ideal para operações frequentes de escrita ou atualização, pois exige a regravação do ficheiro. Contudo, *parquet* é bastante indicado para dados históricos ou preparados, como os presentes na camada prata, onde a prioridade é a performance de leitura em cenários de exploração e transformação de dados em larga escala.

Delta table

As *delta tables*, utilizadas na camada ouro, oferecem uma estrutura de armazenamento que alia a eficiência do formato *parquet* com capacidades transacionais típicas de bases de dados relacionais. Uma *delta table* não corresponde a um único ficheiro, mas sim a uma pasta que contém um ficheiro *parquet* e uma subpasta especial denominada _delta_log, tal como é demonstrado na figura 4.3. Esta pasta de *log* guarda ficheiros JSON com metadados que registam todas as operações realizadas sobre os dados, tais como inserções, atualizações ou remoções. Cada alteração na tabela gera um novo ficheiro de *log*, que descreve o ficheiro *parquet* afetado e o estado dos dados após a operação. Esta abordagem possibilita funcionalidades avançadas como a gestão de versões dos dados, *rollbacks*, e controlo de consistência em ambientes distribuídos. Assim, as *delta tables* são ideais para a camada final do modelo analítico, pois permitem consultas otimizadas através do

SQL *endpoint*, ao mesmo tempo que garantem integridade e rastreabilidade total sobre a evolução dos dados.

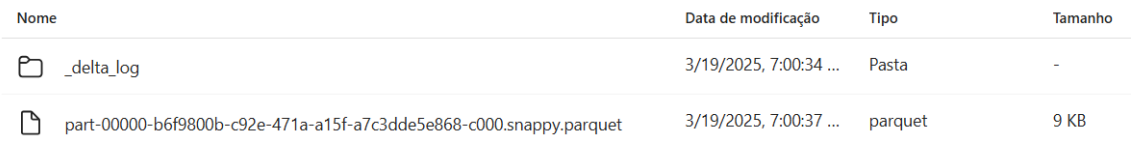

Figura 4.3: Exemplo de *delta table*

4.2 Processo de Transformação de Dados

Para este sistema, foi decidido seguir a abordagem ELT (*Extract*, *Load*, *Transform*). Com esta decisão, os dados são lidos e carregados imediatamente para a *lakehouse*, seguindo-se a leitura dos dados em bruto e a transformação dos mesmos.

A vantagem de escolher ETL seria a menor capacidade de armazenamento utilizada e, por isso, um menor custo de manutenção das bases de dados finais, porque não seria necessário guardar as versões originais dos dados. Contudo, a abordagem escolhida (ELT) permite obter um histórico de dados em bruto, que é bastante útil se, no futuro, for pretendido expandir o modelo de dados final ou adicionar mais funcionalidades, que necessitam de mais informações presentes nos dados originais. ELT também permite obter maior controlo sobre os dados não estruturados (como imagens ou vídeos), pois, na abordagem ETL, esses dados necessitam de ser ignorados, porque na fase de transformação só é possível manipular dados estruturados.

A abordagem ELT também foi escolhida devido à grande quantidade de fontes de informação que são necessárias consultar (os vários ficheiros soltos e outras fontes privadas e públicas) e que se encontram em diversos formatos em bruto (maioritariamente Microsoft Excel e CSV). A relevância de manter o histórico de ficheiros originais também foi uma das razões para a escolha desta abordagem, pois o modelo de dados final, em alguns casos, apenas necessita de parte dos dados em bruto. Desta forma, é possível evitar a perda de informação original, tornando possível uma futura expansão do modelo, tal como mencionado anteriormente.

4.3 Modelo de Dados

Para a modelagem de dados, foi decidido seguir uma estratégia de *star schema*, dado que este modelo é amplamente utilizado em sistemas de apoio à decisão, nomeadamente com ferramentas como o Power BI, pelas suas vantagens em termos de simplicidade, desempenho nas consultas e facilidade de compreensão por parte dos utilizadores de negócio. Este tipo de modelação facilita a construção de relatórios analíticos otimizados, com tempos de resposta rápidos mesmo sobre grandes volumes de dados, o que se alinha com os objetivos do projeto.

Dado o tema do projeto estar centrado nas comunicações eletrónicas e serviços postais, foi realizada uma análise aos ficheiros de origem para identificar os principais agrupamentos temáticos dos dados. A partir desta análise, foi possível identificar três grandes áreas de interesse para a empresa — acessos, clientes e tráfego — que deram origem às três tabelas factuais. Estes temas representam os processos principais a monitorizar e analisar, estando fortemente ligados aos indicadores de desempenho relevantes para o cliente.

De forma a complementar a informação factual, foram criadas as dimensões adequadas ao contexto necessário, permitindo enriquecer as análises com perspetivas temporais, geográficas, contratuais, entre outras.

Na figura 4.4 é apresentado o modelo de dados, constituído pelas três tabelas de factos e pelas dez dimensões, sem especificação dos campos constituintes de cada tabela. Todas as relações são direcionadas das factos para as dimensões, com cardinalidade 0..* para 0..1. Na secção 5.3 é apresentado o modelo de dados final, com todos os campos definidos.

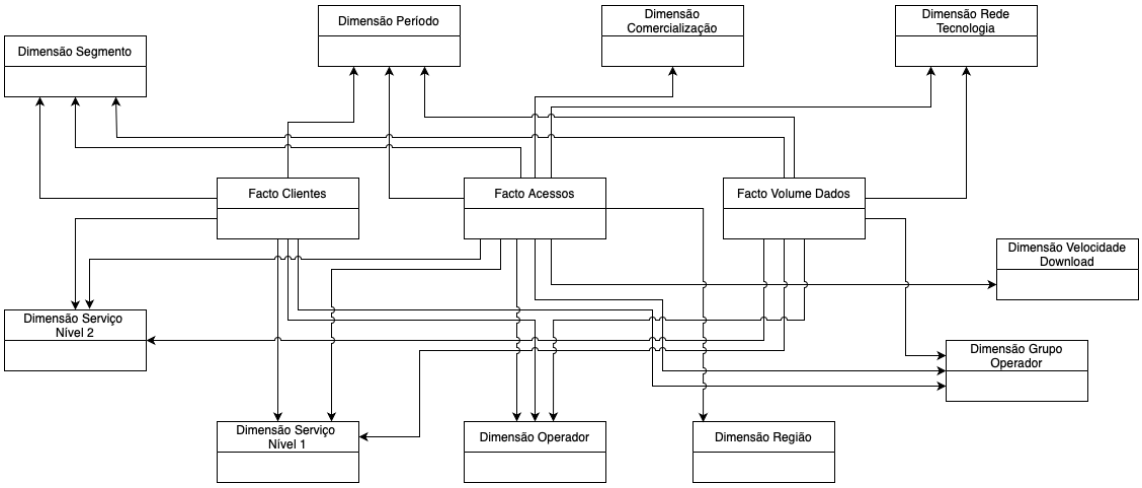

Figura 4.4: Especificação modelo de dados final

4.4 Relatório Analítico

É necessário o desenvolvimento de um relatório sobre o acesso à *Internet* em local fixo, composto por um *dashboard* e uma infografia, ambos em Power BI. O *dashboard* deve possuir várias páginas, para apresentar as várias análises do tema e deve ser interativo. A infografia consiste num *dashboard* estático (sem interações entre visuais) de uma única página, e que apresenta unicamente dados relativos ao último período disponível. O conceito de infografia garante que esta seja consultada após impressão, e para isso, é necessário ter em conta que o utilizador não tem acesso a ações interativas, como, por exemplo, *tooltips*. Os visuais implementados no *dashboard* devem ser reutilizados na infografia com o objetivo de manter os dois recursos consistentes e reduzir o tempo/custo de implementação.

As análises relevantes que devem estar obrigatoriamente presentes neste relatório (*dashboard* e infografia) são:

1. Acessos - valor e variação homóloga.
2. Penetração residencial - por 100 famílias.
3. Distribuição de acessos:
   (a) por segmento.
   (b) por rede/tecnologia.
   (c) por velocidade de download.
4. Tráfego - valor e variação homóloga.
5. Tráfego médio mensal - valor e variação homóloga.
6. Prestadores - valor e variação homóloga.
7. Quotas de acessos.

Será necessário existirem duas versões do relatório (portuguesa e inglesa), através das funcionalidades de tradução disponíveis, no Power BI. Na secção de implementação do relatório (5.5) é mencionado o detalhe destas funcionalidades, que são divididas em 2 partes principais[4]:

- Tradução ao nível de *labels*: todos os títulos e subtítulos necessitam de ser traduzidos automaticamente.
- Tradução ao nível dos dados: são necessárias duas colunas descritivas nas dimensões, uma por cada versão.

A equipa de *design* da ARMIS produziu alguns *mockups* com exemplos a seguir na implementação do *dashboard* e da infografia. Nas figuras 4.5, 4.6, 4.7 e 4.8 são apresentados alguns *mockups* resultantes, que devem servir como guia para a implementação de todos os relatórios.

[4] ver https://learn.microsoft.com/pt-pt/power-bi/guidance/multiple-language-translation - guia para implementação de traduções em Power BI - acedido em 08/04/2025

Figura 4.5: *Mockup* Power BI 1

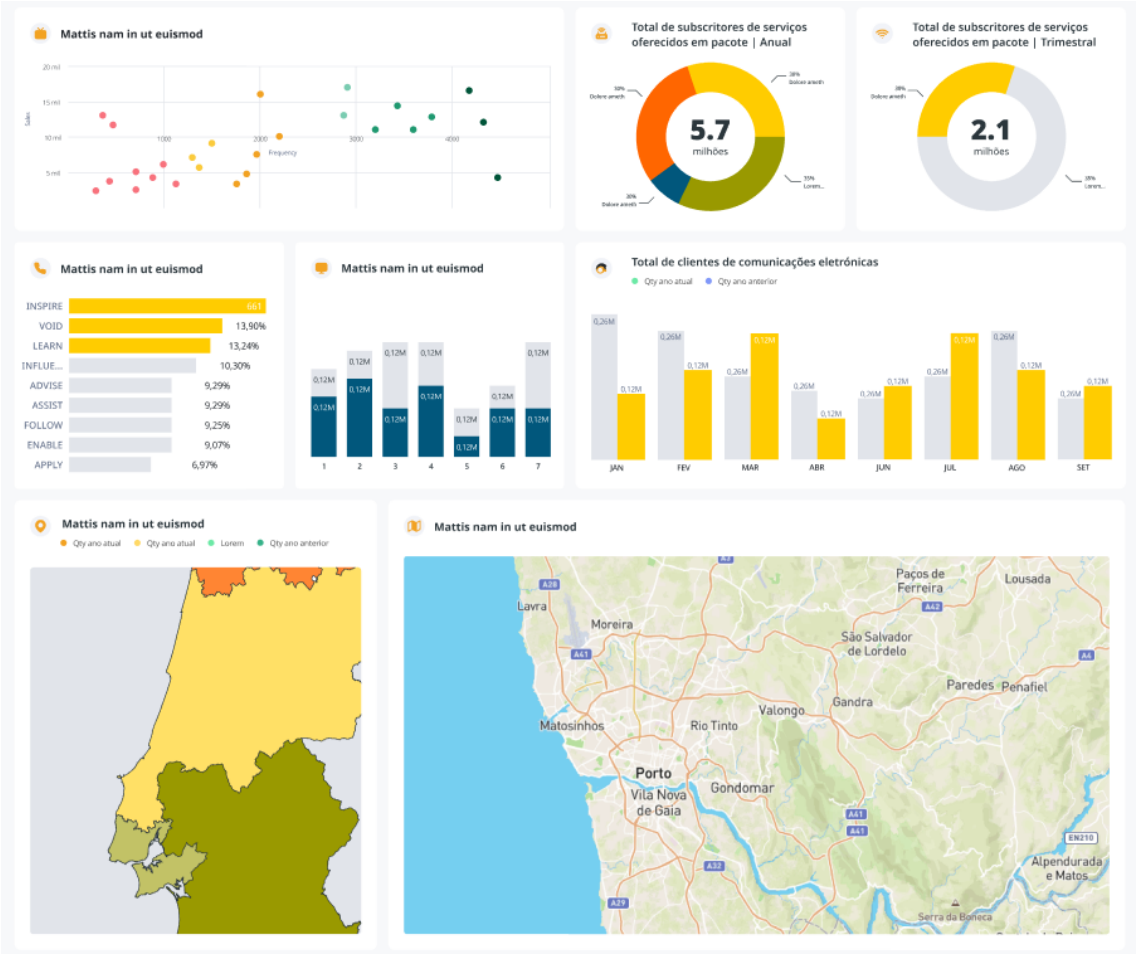

Figura 4.6: *Mockup* Power BI 2

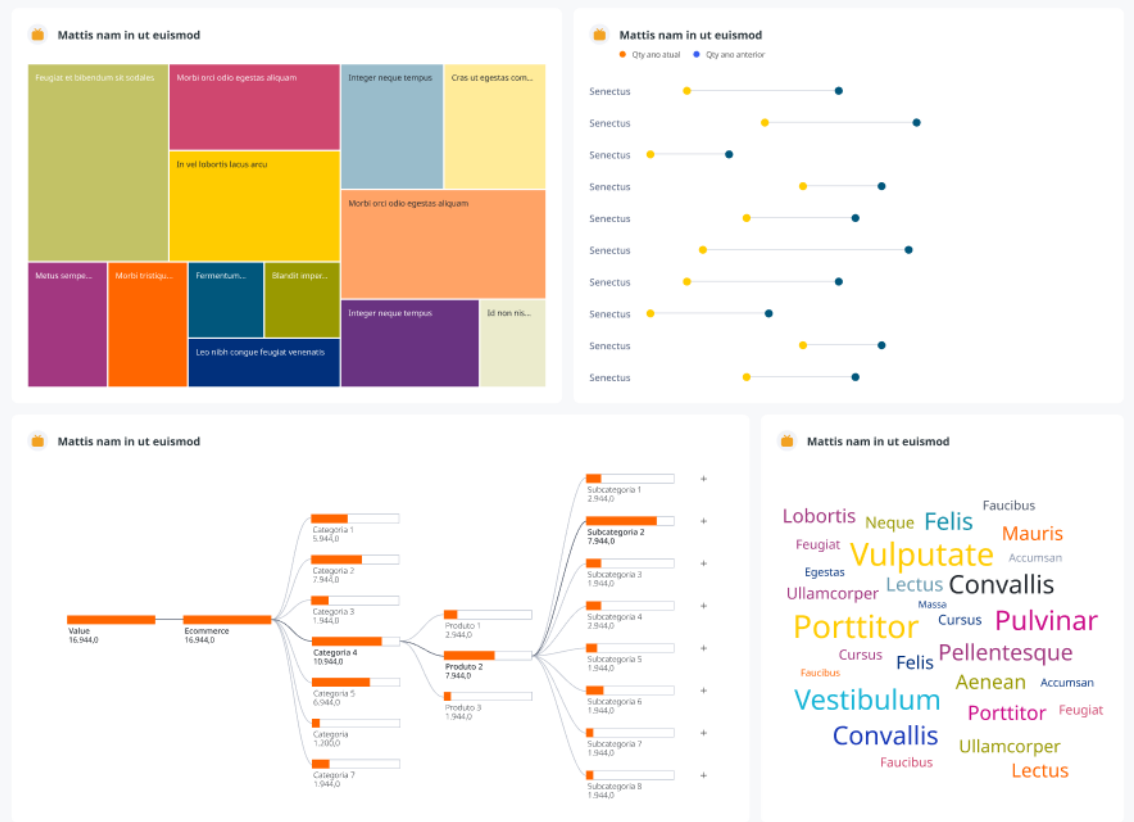

Figura 4.7: *Mockup* Power BI 3

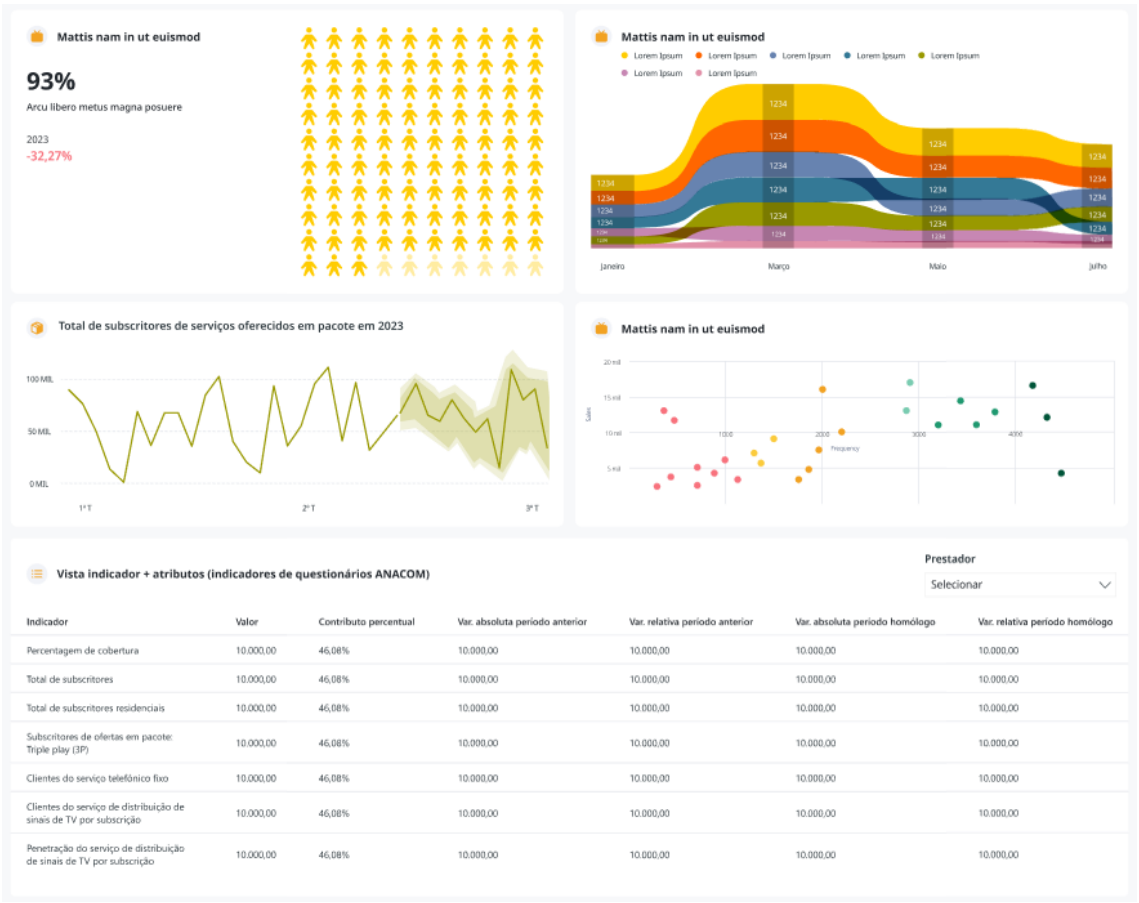

Figura 4.8: *Mockup* Power BI 4

Capítulo 5

Implementação e Resultados

Este capítulo apresenta a implementação da solução desenvolvida, bem como os principais resultados obtidos. São abordadas de forma sistemática todas as componentes do projeto: a modelagem do armazém de dados analítico em *data lakehouse*, o processo de integração e transformação de dados (ELT), e o desenvolvimento do relatório analítico Power BI.

Cada secção foca-se na explicação funcional dos artefactos produzidos, evidenciando as decisões tecnológicas e metodológicas subjacentes, bem como a execução técnica de cada parte do sistema.

Importa referir que, por se tratar de um projeto desenvolvido em contexto empresarial e direcionado a um cliente específico da ARMIS, parte dos artefactos apresentados neste capítulo foi anonimizada ou representada através de exemplos ilustrativos. Esta abordagem visa salvaguardar a confidencialidade da informação e assegurar o cumprimento dos compromissos de privacidade assumidos no âmbito do projeto.

5.1 *Workspace* Fabric

O *workspace* do Fabric é o espaço/plataforma onde todos os componentes do sistema são produzidos e/ou guardados. Na figura 5.1 encontramos a distribuição dos artefactos, por pastas, onde conseguimos visualizar as duas bases de dados (controlo e analítica), *pipelines*, *notebooks* e *dataflows*.

No próprio *workspace* encontra-se disponível um controlo de versões, através de integração com Git. As operações de sincronização são efetuadas através de uma interface *user-friendly*, tal como se pode observar na figura 5.2, com associação a um repositório Azure DevOps[1]. Existem alguns tipos de ficheiros para os quais ainda não existe a possibilidade de sincronizar com o Git, como, por exemplo, *dataflows*. Para estes casos, é necessária uma sincronização manual diretamente no repositório.

[1] ver https://azure.microsoft.com/en-us/products/devops/repos - acedido em 27/04/2025

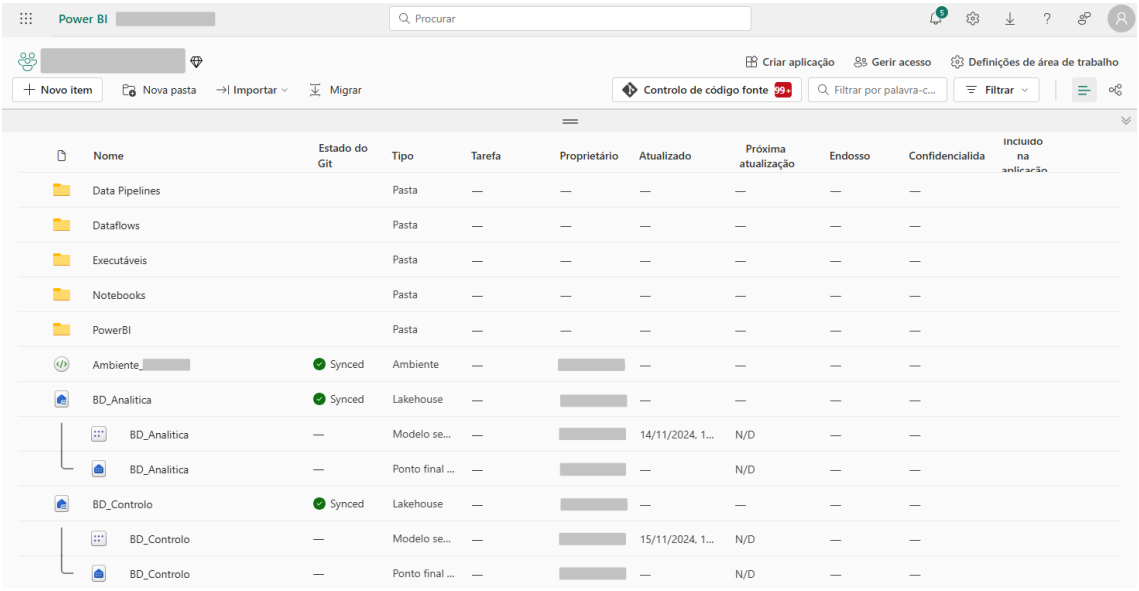

Figura 5.1: *Workspace* Fabric do projeto

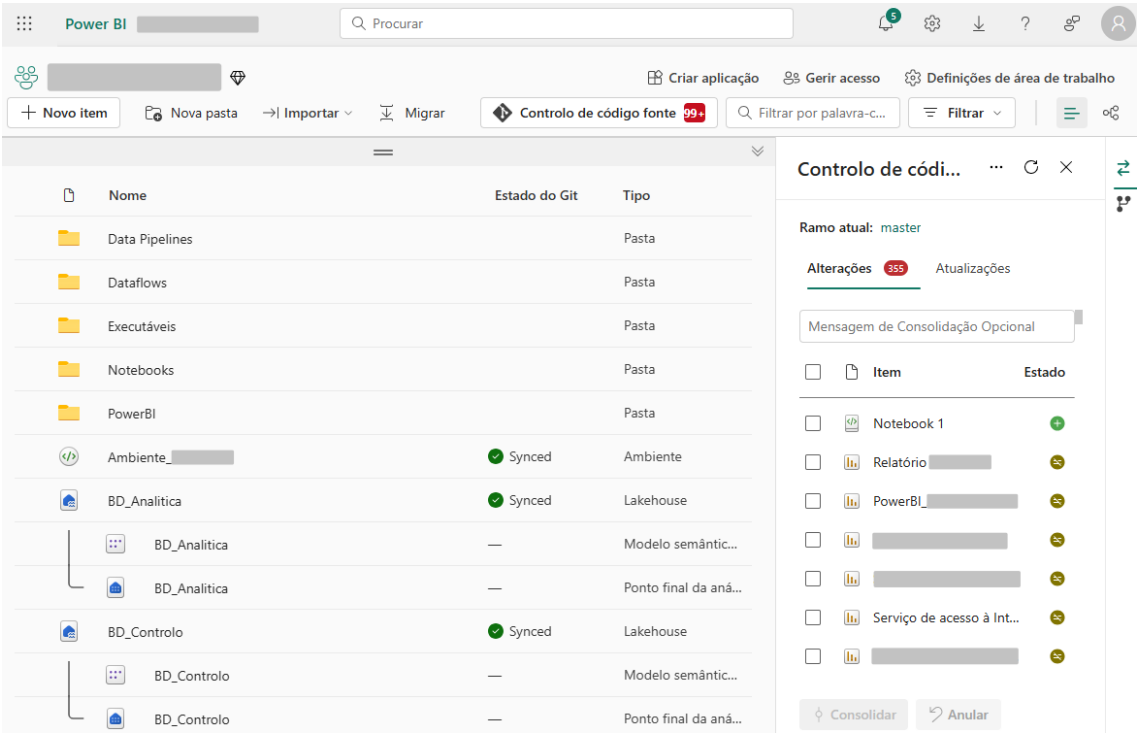

Figura 5.2: Sincronização Git disponível no *workspace* Fabric

5.2 Armazenamento

Tal como mencionado na secção 4.1.2 do capítulo anterior, são necessárias duas bases de dados. A primeira armazena todos os dados para análise de negócio e, para este caso, foi criado um *data lakehouse*, denominado BD_Analitica, que segue uma arquitetura *Medallion*, para armazenar as camadas de dados bronze, prata e ouro. Na figura 5.3, é apresentada a *lakehouse* inserida no *workspace* do Fabric.

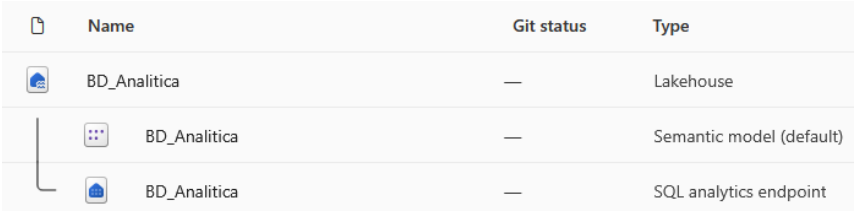

Figura 5.3: Base de dados analítica no *workspace* do Fabric

No caso da base de dados de controlo, denominada BD_Controlo, também foi decidido implementá-la através de um *data lakehouse*, por limitações do próprio Fabric. À data da implementação desta base de dados, não era possível criar bases de dados SQL no Fabric, o que levou a testes funcionais com uma *data warehouse*. No entanto, com esta decisão, surgiram erros em alguns processos, que conduziram ao teste com uma *data lakehouse*. Com esta troca de arquitetura, foi possível extinguir os erros nos processos e prosseguir com a restante implementação do sistema. Ao contrário da BD_Analitica, a BD_Controlo apenas possui as *delta tables* utilizadas para controlo e gestão de fluxos e execuções do sistema, através do SQL *endpoint*, e o sistema de ficheiros não é utilizado.

Na secção 4.1.2, do capítulo anterior, também foi estipulado que a BD_Analitica necessita de possuir um sistema de ficheiros que siga a arquitetura *Medallion*. O sistema de ficheiros desta *lakehouse* é apresentado na figura 5.4 e, tal como especificado, a BD_Analitica possui uma pasta por cada camada necessária (bronze, prata e ouro).

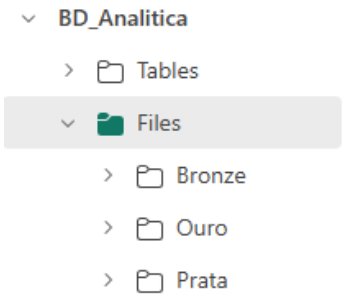

Figura 5.4: Sistema de ficheiros da base de dados analítica

A submissão de ficheiros na plataforma é efetuada de forma manual pelos colaboradores do cliente. Para serem processados, os ficheiros necessitam de ser submetidos numa pasta de um servidor local. De forma a esta pasta poder ser acedida no *workspace* do Fabric, para permitir a ingestão dos ficheiros para a camada bronze, é necessária uma "ponte"entre as duas entidades

(servidor e Fabric/Power BI Service). Para garantir este requisito, foi instalado e configurado um Power BI *On-premises Gateway*[2], que permite a ligação entre o servidor local e o Fabric.

5.3 Modelo de Dados

Seguindo o modelo de dados especificado na secção 4.3, apresenta-se nesta secção o modelo de dados final, agora detalhado com os atributos de cada tabela. Este modelo, baseado na estrutura de *star schema*, serve de suporte à base de dados analítica do projeto.

A estrutura é composta por três tabelas de factos (relativas a tráfego, acessos e clientes) e respetivas tabelas de dimensões, num total de 13 tabelas. As tabelas de factos representam eventos ou transações mensuráveis, enquanto as dimensões oferecem o contexto necessário à análise, possibilitando a segmentação e agregação dos dados.

No modelo de dados desenvolvido, optou-se por não incluir um identificador geral (ID) nas tabelas de factos, uma vez que, no contexto deste projeto, o principal objetivo é a análise e exploração de dados através de ferramentas como o Power BI. Nestes cenários analíticos, a identificação de cada registo é realizada com base na combinação das chaves das dimensões associadas e das respetivas métricas. Como não existem operações de atualização linha a linha, nem é necessário manter histórico detalhado a nível de registo, a inclusão de um ID adicional seria redundante e não traria benefícios práticos para a finalidade do sistema.

O modelo de dados final está representado na figura 5.5.

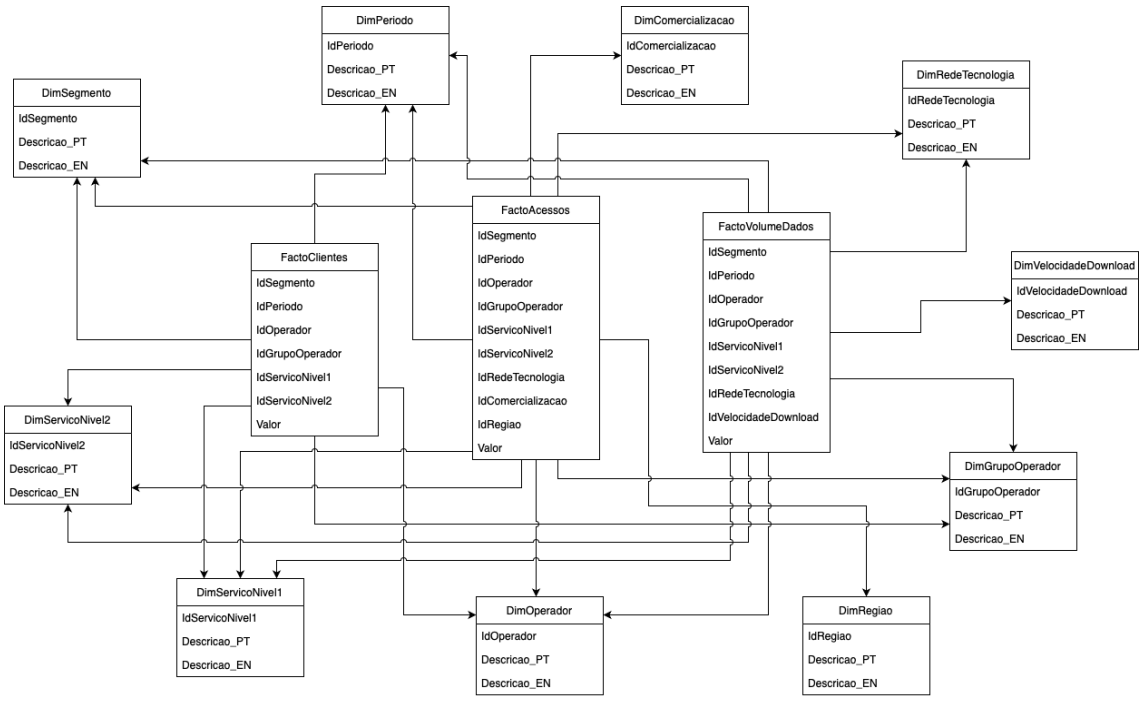

Figura 5.5: Modelo de dados da base de dados analítica

[2] ver https://medium.com/microsoft-power-bi/how-to-set-up-power-bi-on-premises-gateway-connections-254c81bc7457 - acedido em 20/05/2025

5.4 Processo de Transformação de Dados - ELT

O processo de integração de dados implementado segue a abordagem ELT (Extract, Load, Transform), que se adequa à arquitetura de *data lakehouse* adotada. Este processo foi dividido em múltiplas fases, refletidas nas três camadas do modelo: bronze, prata e ouro. As atividades de carregamento, limpeza e transformação de dados foram executadas através de *notebooks* desenvolvidos em Python. A orquestração das integrações foi realizada com recurso a *data pipelines*, que garantem a execução automatizada e cíclica das tarefas. Por fim, foram também implementados mecanismos de alerta, para garantir a monitorização e controlo do estado dos processos.

5.4.1 *Notebooks* de Preparação de Dados

Os *notebooks* Python no ambiente do Microsoft Fabric foram implementados com utilização de *dataframes* Spark e Pandas, dependendo do objetivo/necessidade de cada ação. Cada camada do modelo de dados tem *notebooks* dedicados, com transformações específicas consoante os objetivos de cada fase. Na figura 5.6, é apresentado o ambiente de desenvolvimento de *notebooks* no Microsoft Fabric. À esquerda, é possível observar a conexão direta à *lakehouse* BD_Analitica que foi configurada com o objetivo de facilitar o acesso à mesma. Com esta configuração, é possível aceder/mencionar a *lakehouse* no *notebook* apenas com a palavra "lakehouse", em vez da conexão completa à mesma.

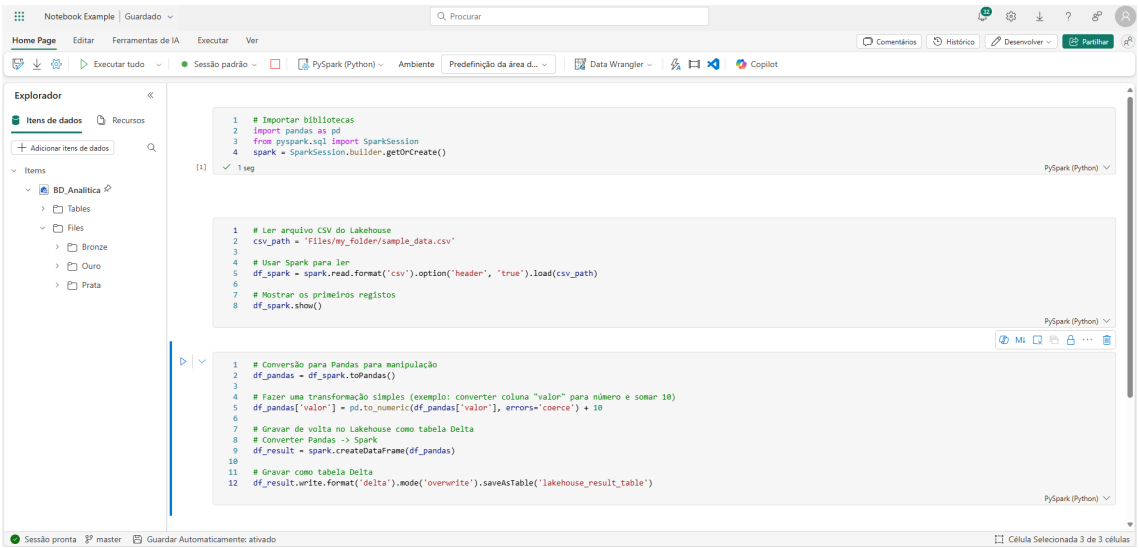

Figura 5.6: Ambiente de desenvolvimento de *notebook* no Microsoft Fabric

5.4.1.1 Camada Bronze

Na primeira camada, os dados são carregados tal como se encontram nos ficheiros originais, com o mínimo de transformação possível. O objetivo principal é garantir uma cópia fiável da origem, independentemente do formato (Excel, CSV, JSON, etc.). As operações realizadas incluem

a leitura dos ficheiros a partir da *lakehouse* e a escrita dos mesmos em formato *parquet*, preservando a estrutura original. No excerto de código 5.1, é apresentado um exemplo das operações necessárias e implementadas no *notebook* da camada bronze.

```
import os
import mssparkutils

# Definicao de caminhos
origem_dir = "Files/Bronze/Temporarios/Submissoes Questionarios/default"
bronze_dir = "Files/Bronze/Submissoes de ficheiros/Questionarios"

# Listagem dos ficheiros submetidos
ficheiros = [f.name for f in mssparkutils.fs.ls(origem_dir) if f.name.endswith(".
    xlsx")]

# Processamento e movimentacao dos ficheiros
for ficheiro in ficheiros:
    caminho_origem = f"{origem_dir}/{ficheiro}"
    caminho_destino = f"{bronze_dir}/{ficheiro}"

    # Validacao do nome
    if " " in ficheiro:
        print(f"Nome invalido: {ficheiro}")
        continue

    # Mover ficheiro para a camada bronze
    mssparkutils.fs.cp(caminho_origem, caminho_destino)
    print(f"Ficheiro {ficheiro} movido com sucesso.")

print("Processamento concluido.")
```

Excerto de Código 5.1: Exemplo *notebook* da camada bronze

5.4.1.2 Camada Prata

A camada de prata foca-se na limpeza e padronização dos dados. Dependendo da fonte de dados, são aplicadas transformações como:

- Normalização de formatos (data, inteiro, decimal, texto).
- Remoção de colunas ou linhas irrelevantes.
- Junções entre ficheiros de origem.
- Geração de colunas derivadas.

Esta camada tem a particularidade de possuir a necessidade de validar e calcular indicadores. As validações e cálculos dos indicadores têm que ser executados de acordo com as regras impostas

pelo cliente do projeto. As regras são armazenadas numa *delta table*, na BD_Controlo, para posteriormente serem utilizadas nos *notebooks* de transformação e carregamento.

Com isto, a fase de transformação e carregamento da camada prata foi dividida em 3 etapas:

1. Leitura e Formatação
2. Validação dos Indicadores
3. Cálculo dos Indicadores

Todos os dados nesta camada são armazenados em ficheiros *parquet*, que permitem consultas mais eficientes, tal como mencionado na secção 4.1.2. No excerto de código 5.2, é apresentado um exemplo de operações que são executadas no *notebook* da primeira etapa da camada prata. Aqui, o objetivo é carregar e limpar os ficheiros originais para o formato *parquet*, numa camada intermédia.

```
from pyspark.sql.types import StructType, StructField, StringType, IntegerType,
    DoubleType
from pyspark.sql.functions import col, when

# Definir schema e ler da camada bronze
schema = StructType([
    StructField("id", IntegerType(), True),
    StructField("nome", StringType(), True),
    StructField("idade", IntegerType(), True),
    StructField("salario", DoubleType(), True)
])

df_bronze = spark.read.format("csv") \
    .option("header", "true") \
    .schema(schema) \
    .load("/lakehouse/default/Files/Bronze/dados_funcionarios.csv")

# Limpeza e validacao simples
df_prata = df_bronze \
    .filter(col("idade").isNotNull()) \
    .withColumn("salario", when(col("salario").isNull(), 0).otherwise(col("salario"
    )))

# Gravar na camada prata
df_prata.write.mode("overwrite").parquet("/lakehouse/prata/dados_funcionarios_prata
    .parquet")

# Analisar resultado
display(df_prata)
```

Excerto de Código 5.2: Exemplo *notebook* da etapa 1 da camada prata

Na segunda etapa desta camada é necessário validar os indicadores, de acordo com as regras do cliente. As regras/fórmulas de validação são armazenadas na base de dados de controlo

(BD_Controlo). No excerto de código 5.3, é apresentado um exemplo de um *notebook* em que é efetuada a validação de alguns indicadores através de comparações dos resultados booleanos da avaliação de fórmulas fornecidas pelo cliente.

```
import pyspark
from pyspark.sql.functions import col

def traduzir_expressao(expressao, dicionario_dados, periodo):
    # Traduz uma expressao substituindo variaveis por valores reais
    # Substitui referencias a indicadores pelos valores
    for indicador, valor in dicionario_dados.items():
        expressao = expressao.replace(f'"{indicador}"', str(valor))

    # Substitui referencias de periodo
    expressao = expressao.replace('periodo()', f'"{periodo}"')

    return expressao

def validar_expressao(expressao, dicionario_dados, periodo):
    # Valida uma expressao de logica de negocio individual
    try:
        # Traduz a expressao com valores reais
        expressao_traduzida = traduzir_expressao(expressao, dicionario_dados,
    periodo)

        # Avalia a expressao
        resultado = eval(expressao_traduzida)
        return resultado
    except Exception:
        return False

def validar_indicadores(dados_staging, regras_validacao):
    # Processo principal de validacao de indicadores
    logs_validacao = []

    for regra in regras_validacao:
        # Aplica regra de validacao aos indicadores
        eh_valido = validar_expressao(
            regra['expressao'],
            dados_staging,
            regra['periodo']
        )

        # Regista resultado da validacao
        logs_validacao.append({
            'regra': regra['expressao'],
            'valido': eh_valido,
            'mensagem': regra.get('mensagem_erro', 'Validacao falhou')
        })
```

```
45
46      return {
47          'passou': all(log['valido'] for log in logs_validacao),
48          'logs': logs_validacao
49      }
50
51  def fluxo_trabalho_validacao(tabelas_staging, regras_validacao):
52      # Orquestra o processo completo de validacao
53      try:
54          # Carrega dados de staging
55          dados_staging = carregar_dados_staging(tabelas_staging)
56
57          # Valida indicadores
58          resultado_validacao = validar_indicadores(
59              dados_staging,
60              regras_validacao
61          )
62
63          return {
64              'sucesso': resultado_validacao['passou'],
65              'logs': resultado_validacao['logs']
66          }
67
68      except Exception as e:
69          return {
70              'sucesso': False,
71              'erro': str(e)
72          }
```

Excerto de Código 5.3: Exemplo *notebook* da etapa 2 da camada prata

Por fim, existe a etapa 3, que é responsável por calcular alguns indicadores pretendidos pelo cliente. O cálculo de cada indicador é efetuado com base na regra de criação indicada, previamente, pelo cliente e armazenada na BD_Controlo. No excerto de código 5.4, é apresentado um exemplo de um *notebook* responsável por esta etapa final da camada prata.

```
1  def _obter_operadora(self):
2      """
3      Recupera informacoes da operadora a partir do ID
4      """
5      query = "SELECT Designacao_Abreviado FROM BD_Controlo.dbo.Operadoras WHERE
       Id_Tabela = :id_operadora"
6      return self.spark.sql(query, args={"id_operadora": self.parametros['
       id_operadora']}).collect()[0].Designacao_Abreviado
7
8  def processar_questionario_q(self):
9      # Carrega formulas de calculo
10     df_formulas = self.spark.sql(
11         "SELECT * FROM BD_Controlo.dbo.MIO_RegrasCalculos WHERE Is_Ativo = 1"
12     )
```

```
13
14      # Carrega dados de staging
15      df_dados_completo = self._ler_tabela_staging()
16
17      for ordem in [1, 2, 3]:
18          df_formulas_etapa = df_formulas.filter(f"Ordem_Calculo = {ordem}")
19          # Valida e calcula indicadores
20          resultados = self._validar_calculos(
21              df_formulas_etapa,
22              df_dados_completo
23          )
24          # Atualiza dados com novos resultados
25          df_dados_completo = self._atualizar_dataframe(
26              df_dados_completo,
27              resultados
28          )
29
30      # Grava resultados finais
31      self._gravar_resultados(df_dados_completo)
32
33  def _validar_calculos(self, df_formulas, df_dados):
34      # Nucleo do processamento: valida e calcula indicadores
35      com base em regras definidas
36      resultados = []
37      for formula in df_formulas.collect():
38          # Interpreta e executa cada formula
39          resultado = self._interpretar_formula(
40              formula,
41              df_dados
42          )
43          resultados.append(resultado)
44      return resultados
45
46  def _interpretar_formula(self, formula, dados):
47      try:
48          # Logica de interpretacao e validacao de formulas
49          expressao_traduzida = self._traduzir_expressao(formula)
50          resultado = eval(expressao_traduzida)
51          return {
52              'indicador': formula.Cod_Indicador,
53              'valor': resultado
54          }
55      except Exception as e:
56          # Tratamento de erros de calculo
57          return {
58              'indicador': formula.Cod_Indicador,
59              'erro': str(e)
60          }
61
```

```
def main(spark, parametros):
    try:
        processador = CalculadorIndicadores(spark, parametros)
        processador.processar_questionario_q()
        return {"sucesso": True}
    except Exception as e:
        return {"sucesso": False, "erro": str(e)}

# Exemplo de chamada
resultado = main(spark, {
    'id_operadora': OPERADORA_ID,
    'periodo': PERIODO_REFERENCIA
})
```

Excerto de Código 5.4: Exemplo *notebook* da etapa 3 da camada prata

5.4.1.3 Camada Ouro

Na camada final, os dados são agregados e modelados de acordo com o modelo apresentado na secção 5.3. Nesta camada, os dados são inseridos em *delta tables*, tirando partido das funcionalidades acrescidas de versionamento e integração com *endpoints* SQL para consulta direta e eficiente.

Para a criação das tabelas de dimensões e factos foram necessários processos diferentes. No excerto de código 5.5, é apresentado um exemplo do *notebook* responsável pela transformação de dados das dimensões. Neste exemplo, para uma dimensão fictícia de produtos, são apresentados vários tipos de operações que foram utilizados na criação das tabelas de dimensão do modelo, dependendo da necessidade de cada uma.

```
from pyspark.sql.functions import (
    col, upper, trim, when, lit, length, substring, coalesce
)
from delta.tables import DeltaTable

# Leitura da dimensao da camada prata
df_prata_produtos = spark.read.format("delta").load("lakehouse/prata/source_file")

df_ouro_produtos = (
    df_prata_produtos
    # Normalizar nome do produto
    .withColumn("nome_produto", upper(trim(col("nome_produto"))))

    # Criar coluna de categoria padrao caso esteja nula
    .withColumn("categoria", coalesce(col("categoria"), lit("DESCONHECIDO")))

    # Criar coluna de tipo com base na descricao
    .withColumn(
        "tipo_produto",
        when(col("descricao").like("%luxo%"), "LUXO")
```

```
        .when(col("descricao").like("%economico%"), "ECONOMICO")
        .otherwise("PADRAO")
    )

    # Criar flag de produto ativo (1 ou 0)
    .withColumn(
        "flag_ativo",
        when(col("data_fim").isNull(), lit(1)).otherwise(lit(0))
    )

    # Criar codigo abreviado (primeiras 3 letras do nome + ID)
    .withColumn(
        "codigo_abreviado",
        upper(substring(col("nome_produto"), 1, 3)).cast("string")
        .concat(col("id_produto").cast("string"))
    )

    # Criar coluna de comprimento do nome
    .withColumn("tamanho_nome", length(col("nome_produto")))

    # Remover duplicados por ID
    .dropDuplicates(["id_produto"])
)

# Validacao final
df_ouro_produtos.select("id_produto", "nome_produto", "categoria", "tipo_produto",
    "flag_ativo", "codigo_abreviado").show(25, truncate=False)

# Escrita na camada ouro
df_ouro_produtos.write.format("delta") \
    .mode("overwrite") \
    .save("lakehouse/ouro/dim_produtos")
```

Excerto de Código 5.5: Exemplo de *notebook* de preparação das dimensões da camada ouro

Em contrapartida, no excerto 5.6, o código trata-se de um exemplo da preparação e carregamento de dados para as tabelas de factos. Neste caso, é necessário carregar as dimensões para ser possível associar os Ids de cada dimensão a cada linha da tabela facto. Para além dos identificadores das dimensões, é necessário carregar os restantes campos das tabelas, como, por exemplo, valor e unidade.

Em ambos os excertos, os dados da última camada prata (inicialmente em formato *parquet*) são transformados e inseridos no formato da camada ouro (*delta tables* de acordo com o modelo de dados final).

```
# Leitura das tabelas de dimensoes
dim_clientes = spark.read.format("delta").load("lakehouse/ouro/dim_clientes")
dim_produtos = spark.read.format("delta").load("lakehouse/ouro/dim_produtos")
dim_tempo = spark.read.format("delta").load("lakehouse/ouro/dim_tempo")
dim_tipo_doc = spark.read.format("delta").load("lakehouse/ouro/dim_tipo_documento")
```

```
dim_utilizador = spark.read.format("delta").load("lakehouse/ouro/dim_utilizador")

# Leitura do ficheiro
vendas = spark.read.parquet("lakehouse/prata/vendas")

# Juncao com dimensoes
df_facto_vendas = (
    vendas
    .join(dim_clientes, vendas.cliente == dim_clientes.cliente, "left")
    .join(dim_produtos, vendas.produto == dim_produtos.produto, "left")
    .join(dim_tempo, vendas.data == dim_tempo.data, "left")
    .join(dim_tipo_doc, vendas.tipo_doc == dim_tipo_doc.tipo_doc, "left")
    .join(dim_utilizador, vendas.utilizador == dim_utilizador.utilizador, "left")
    .select(
        "id_cliente",
        "id_produto",
        "id_tempo",
        "id_tipo_documento",
        "id_utilizador",
        "quantidade",
        "valor_liquido",
        "valor_bruto"
    )
)

# Escrita em delta table - camada ouro
df_facto_vendas.write.format("delta").mode("overwrite").save("lakehouse/ouro/
    facto_vendas")
```

Excerto de Código 5.6: Exemplo de *notebook* de preparação das factos da camada ouro

5.4.2 Controlo do Fluxo de Dados

O fluxo de carregamento e transformação de dados é controlado e gerido através de uma tabela da BD_Controlo, denominada ControloIntegracoes. É nesta tabela que são armazenados e consultados todos os detalhes relativos à integração de cada ficheiro de origem, submetido na plataforma.

Para cada submissão de um ficheiro, são guardados os seguintes campos:

- Id_Submissao (*varchar(160) not null*)
- Id_Estado (*smallint not null*)
- Id_Periodo (*varchar(40) not null*)
- Id_Operadora (*bigint not null*)
- Caminho_Ficheiro (*varchar(1200) not null*)

- Nome_Ficheiro (*varchar(800) not null*)
- Id_Execucao_Ingestao (*varchar(160) null*)
- Data_Ingestao (*datetime null*)
- Mensagem_Ingestao (*varchar(2000) null*)
- Id_Execucao_StagingPrata (*varchar(160) null*)
- Data_StagingPrata (*datetime null*)
- Mensagem_StagingPrata (*varchar(2000) null*)

Os campos Id_Execucao_Ingestao e Id_Execucao_StagingPrata são identificadores da execução dos *notebooks* da camada bronze e primeira etapa da camada prata, respetivamente. Estes dois Ids e o Id_Submissão são criados no formato GUID (*Global Unique Identifier*) para assegurar unicidade global, independência entre sistemas e rastreabilidade fiável das execuções[3]. A necessidade do formato GUID deve-se ao facto de os diversos Ids serem gerados em submissões de ficheiros, execuções de *notebooks* e *data pipelines*.

Nos campos Mensagem_Ingestao e Mensagem_StagingPrata são guardadas mensagens de erros que ocorreram na execução dos respetivos *notebooks*. Já nos campos Data_Ingestao e Data_StagingPrata são armazenadas as *timestamps* das execuções destes processos. O Id_Estado é um campo identificador da tabela de estados de processamento, que permite determinar a etapa de integração em que o ficheiro se encontra.

Na figura 5.7, é possível observar alguns registos guardados na tabela de controlo do fluxo de integração dos ficheiros.

	Id_Submissao	Id_Estado	Id_Periodo	Id_Operadora	Caminho_Ficheiro	Nome_Ficheiro	Id_Execucao_Ingestao	Data_Ingestao	Mensagem_Ingestao	Id_Execucao_StagingPrata	Data_StagingPrata	Mensagem_StagingPrata
1	oc5a15e9-c62...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:43....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:24:02....	NULL
2	6c0a117f-22e...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:05....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:39:56....	NULL
3	97f590d9-7db...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:08....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:39:42....	NULL
4	4d381207-74...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:45....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:23:48....	NULL
5	a781c9ec-a3...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:06....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:35:24....	NULL
6	70a841ee-ddf...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:44....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:23:19....	NULL
7	3ddca9b8-80...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:08....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:35:40....	NULL
8	15705862-56...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:46....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:23:35....	NULL
9	c06a5a44-73...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:04....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:35:09....	NULL
10	c348a5eb-76...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:42....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:23:03....	NULL
11	6f6b0a2e1-916...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:00....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:34:25....	NULL
12	e912a5e8-0a...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:38....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:22:15....	NULL
13	efd9a59b-d65...	5	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:02....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:36:55....	Colunas não estão conforme o pré-definido. Eram ...
14	ed86b01a-d1...	5	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:40....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:21:30....	Colunas não estão conforme o pré-definido. Eram ...
15	5c9633d3-e0...	6	2023 T1	464	Files/Bronze/Submissões de ficheiros/...		0bf7bcc5-07fa-4777-b93...	2024-10-17 17:08:23....	NULL	0bf7bcc5-07fa-4777-b93...	2024-10-17 17:29:18....	NULL
16	21689566-b9...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:00:58....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:34:56....	NULL
17	5d82f911-72f...	4	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:36....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:22:47....	NULL
18	adfc33a0-f52...	5	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:01:00....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:36:39....	Colunas não estão conforme o pré-definido. Eram ...
19	e5e27194-4c...	5	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		7d330664-abab-4acd-8e...	2025-05-16 16:14:39....	NULL	7d330664-abab-4acd-8e...	2025-05-16 16:21:44....	Colunas não estão conforme o pré-definido. Eram ...
20	0e852de7-0d...	5	2025 T1	688	Files/Bronze/Submissões de ficheiros/...		03d719a2-8ede-4d9b-8...	2025-05-12 14:00:58....	NULL	03d719a2-8ede-4d9b-8...	2025-05-12 14:33:38....	Colunas não estão conforme o pré-definido. Eram ...

Figura 5.7: Tabela de controlo do fluxo de integração dos ficheiros

Na figura 5.8 é apresentado o conteúdo da tabela de estados de processamento. São apresentados dois estados para a ingestão dos dados para a camada bronze, dez estados para o fluxo da camada prata e três estados para a passagem para a camada ouro. Os estados "*Staging%*"e "Pronto para validação"servem para a primeira etapa da camada prata. Os estados "Validação%"e "Cálculo%"são utilizados no controlo das duas etapas finais da camada prata: validação e cálculo de indicadores, respetivamente.

[3] ver https://stackoverflow.com/questions/371762/what-exactly-is-guid-why-and-where-i-should-use-it - acedido em 17/05/2025

	ID_Tabela	Descricao
1	1	Ingestão concluida com sucesso
2	2	Ingestão terminada com erros
3	3	Staging para prata em curso
4	4	Staging para prata concluido com sucesso
5	5	Staging para prata terminado com erro
6	6	Pronto para validação
7	7	Validação em curso
8	8	Validação concluida com sucesso
9	9	Validação concluida com erro
10	10	Cálculo dos indicadores em curso
11	11	Cálculo dos indicadores concluido com sucesso
12	12	Cálculo dos indicadores concluido com erro
13	13	Passagem para o modelo analitico em curso
14	14	Passagem para o modelo analitico concluido com sucesso
15	15	Passagem para o modelo analitico concluido com erros

Figura 5.8: Tabela de estados de processamento

5.4.3 Mecanismos de Alerta

Para garantir a monitorização eficaz do processo, foram implementados mecanismos de alerta, por *e-mail*, que notificam os responsáveis pelo sistema sobre o estado da execução das integrações.

Existem diferentes tipos de mensagens, com formatos distintos conforme o resultado:

- **Submissão bem-sucedida:** indica que os ficheiros foram recebidos e integrados com sucesso.
- **Erros na submissão:** reporta falhas como ficheiros mal posicionados, nomes inválidos, extensões incorretas ou diretórios inconsistentes.
- **Falta de dados:** deteta a ausência de informações obrigatórias, impedindo o avanço do processamento.
- **Falhas em validações ou cálculos:** indica erros nas validações de regras de negócio ou no processamento de dados para cálculo de indicadores.

Estes alertas são fundamentais para garantir uma operação contínua e controlada, que permita uma resposta rápida a eventuais falhas.

Para cada tipo de alerta, são guardados, numa tabela denominada InformacoesEnderecosAlertas, na base de dados de controlo, os endereços de *e-mail* dos destinatários que os devem receber. Esta tabela armazena o tipo de alerta, o endereço, um campo booleano (que define se a informação se encontra ativa) e duas *timestamps* (uma de ativação e outra de desativação).

No excerto de código 5.7, é apresentada uma versão exemplificativa do *notebook* responsável pelo funcionamento do mecanismo de gestão e envio de alertas. É possível observar o fluxo necessário para enviar uma mensagem de informação do estado dos ficheiros submetidos e integrados, durante uma determinada execução. O corpo do *e-mail* é criado consoante os ficheiros submetidos

e o resultado do processamento de cada um. A lista de destinatários é determinada com base numa consulta à tabela InformacoesEnderecosAlertas, abordada anteriormente.

```
from datetime import datetime
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
import smtplib
from pyspark.sql import SparkSession
from pyspark.sql.functions import col

spark = SparkSession.builder.getOrCreate()

# Parametros
id_execucao = "Execucao_123"
tipo_alerta = "Submissao"

# Acessos
tabela_enderecos = "BD_Controlo.InformacoesEnderecosAlertas"
tabela_logs_submissoes = "BD_Controlo.logs_submissoes"

# Obter lista de emails ativos para o tipo de alerta
def obter_destinatarios(tipo_alerta: str):
    df_emails = spark.read.format("delta").table(tabela_enderecos)
    emails = df_emails.filter((col("Is_Ativo") == True) & (col("Tipo_Alerta") ==
    tipo_alerta)) \
                      .select("Endereco").rdd.flatMap(lambda x: x).collect()
    return emails

# Gerar saudacao e corpo base
def gerar_corpo_base(id_execucao):
    hora = datetime.now().hour
    saudacao = "Bom dia" if hora < 12 else "Boa tarde"
    return f"""{saudacao},\nA execucao da pipeline com ID '{id_execucao}' foi
    concluida.\nSegue um resumo dos ficheiros submetidos:"""

# Acrescentar resumo de submissoes ao corpo do email
def adicionar_resumo_submissoes(corpo, id_execucao):
    df_logs = spark.read.format("delta").table(tabela_logs_submissoes)
    df_execucao = df_logs.filter(col("Id_Submissao") == id_execucao)

    if df_execucao.count() == 0:
        return corpo + "\n(Nenhuma submissao encontrada para esta execucao.)"

    for linha in df_execucao.select("Operadora", "Id_Periodo", "Ficheiro", "
    Estado_Execucao").collect():
        corpo += f"\n  - Operadora: {linha.Operadora}, Periodo: {linha.Id_Periodo},
     Ficheiro: {linha.Ficheiro}, Estado: {linha.Estado_Execucao}"

    return corpo

```

```
# Funcao de envio do email
def enviar_email(destinatarios, assunto, corpo):
    if not destinatarios:
        print("Sem destinatarios.")
        return
    msg = MIMEMultipart()
    msg['From'] = "exemplo@dominio.pt"
    msg['To'] = ", ".join(destinatarios)
    msg['Subject'] = assunto
    msg.attach(MIMEText(corpo, 'plain'))

    try:
        with smtplib.SMTP("smtp.servidor.pt", 587) as server:
            server.starttls()
            server.login("utilizador", "password")
            server.send_message(msg)
    except Exception as e:
        print("Erro ao enviar email:", e)


# Execucao principal
corpo_base = gerar_corpo_base(id_execucao)
corpo_completo = adicionar_resumo_submissoes(corpo_base, id_execucao)
corpo_completo += "\n\nCumprimentos,\nEquipa de Suporte"
destinatarios = obter_destinatarios(tipo_alerta)
assunto = f"Resumo da execucao {id_execucao}"

# Envio
enviar_email(destinatarios, assunto, corpo_completo)
```

Excerto de Código 5.7: Exemplo *notebook* de mecanismo de alerta

Para controlar o envio dos alertas necessários, foi criada uma tabela na base de dados de controlo denominada ControloEnvioAlertas. Esta tabela tem como objetivo guardar o histórico de alertas enviados ou por enviar. A tabela possui os seguintes campos:

- Id_Execucao_Pipeline (*varchar(160) not null*)
- Processo (*varchar(600) not null*)
- Id_Alerta (*varchar(160) not null*)
- Enviado (*bit not null*)
- Mensagem_Erro (*varchar(8000) null*)
- Data_Envio (*datetime not null*) - quando não enviado, esta data é preenchida na mesma com a data de tentativa de envio

5.4.4 *Data Pipelines*

As *data pipelines* permitem orquestrar todas as tarefas do processo de ELT deste sistema, garantindo que são executadas na sequência correta, com os devidos controlos e em horários definidos. Neste sistema, as *pipelines* têm como objetivos:

- Executar *notebooks* em sequência lógica,
- Verificar a presença de novos ficheiros.
- Controlar fluxos condicionais (ex: só avançar se o ficheiro existir).
- Realizar limpeza ou movimentação de ficheiros na *lakehouse*.

As *pipelines* suportam agendamentos e permitem personalização extensiva, incluindo a chamada de API's externas, lógica condicional e execução sequencial e/ou paralela de tarefas (exemplos: cópia de dados e execução de *notebooks*).

De forma a ser possível controlar as execuções de todas as *pipelines*, foi criada uma tabela de controlo, com o nome ControloExecucoesPipelines, para inserir os *logs* necessários. Esta tabela (ver figura 5.9) possui os seguintes campos:

- Id_Execucao (*varchar(160) not null*)
- Processo (*varchar(600) not null*)
- Em_Curso (*bit not null*)
- Sucesso (*bit not null*)
- Mensagem_Erro (*varchar(8000) null*)
- Data_Inicio_Execucao (*datetime not null*)
- Data_Fim_Execucao (*datetime null*)

	Id_Execucao	Processo	Em_Curso	Sucesso	Mensagem_Erro	Data_Inicio_Execucao	Data_Fim_Execucao
1	be065b43-e89...	Fluxo ...	0	1	NULL	2025-05-21 19:11:3...	2025-05-21 19:24:...
2	903bf136-7792...	Fluxo ...	0	1	NULL	2025-05-21 18:53:2...	2025-05-21 19:10:...
3	913bb95a-97a...	Fluxo ...	0	1	NULL	2025-05-21 18:46:5...	2025-05-21 18:52:...
4	27c843c2-e7d...	Fluxo ...	0	1	NULL	2025-05-21 17:38:1...	2025-05-21 18:00:...
5	ba4af72d-7990...	Fluxo ...	0	1	NULL	2025-05-21 15:11:0...	2025-05-21 16:49:...
6	5b79bfa7-7ba8...	Fluxo ...	0	1	NULL	2025-05-21 14:40:5...	2025-05-21 15:09:...
7	caee95b3-e47f...	Fluxo ...	0	1	NULL	2025-05-21 14:10:1...	2025-05-21 14:34:...
8	baf6b806-fa4f-...	Fluxo ...	0	1	NULL	2025-05-21 11:10:5...	2025-05-21 11:24:...
9	7112b73f-aa0c...	Fluxo ...	0	1	NULL	2025-05-21 10:51:5...	2025-05-21 11:08:...
10	96f49ed5-4026...	Fluxo ...	0	1	NULL	2025-05-21 09:20:0...	2025-05-21 09:48:...

Figura 5.9: Tabela de controlo de execuções de *data pipelines*

Na figura 5.10, são apresentadas as cinco *pipelines* que foram implementadas para processar todas as etapas do fluxo de carregamento e transformação de dados. As cinco utilizam atividades e comportamentos bastante similares, apenas com algumas diferenças entre as mesmas.

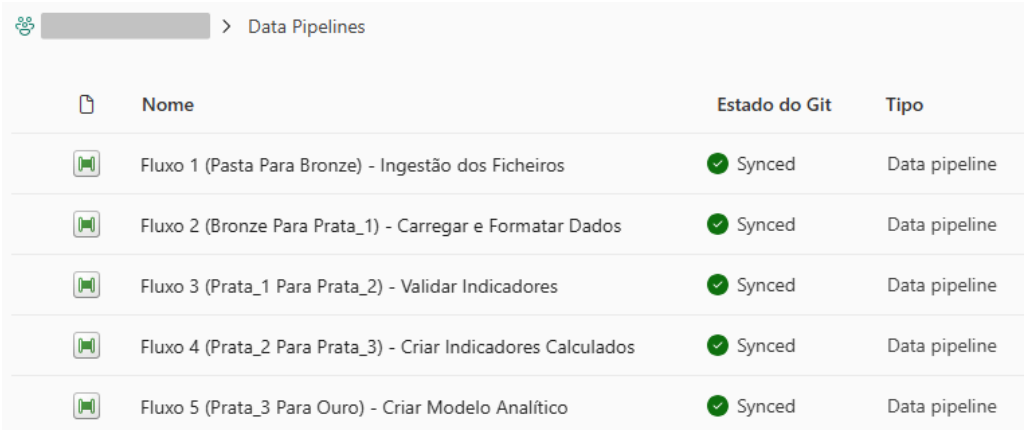

Figura 5.10: *Data pipelines* no *workspace* Fabric

5.4.4.1 Camada Bronze

O processo de integração de dados começa com a preparação da camada bronze. A *pipeline* da camada bronze começa por definir o Id de execução da *pipeline* e executa o *notebook* de carregamento da camada bronze. Após a execução, o resultado do mesmo é guardado numa variável, para poder ser utilizado no componente da condição lógica que verifica se ocorreu algum erro. No caso de ocorrer algum erro no *notebook*, é inserida uma mensagem de erro na tabela ControloExecucoesPipelines, abordada anteriormente. Após este passo, é executado o mecanismo de alerta, para informar os devidos contactos. Se o alerta for enviado com sucesso, é retornado sucesso no valor de retorno da *pipeline*. Caso contrário, é inserida uma mensagem de erro na tabela ControloEnvioAlertas.

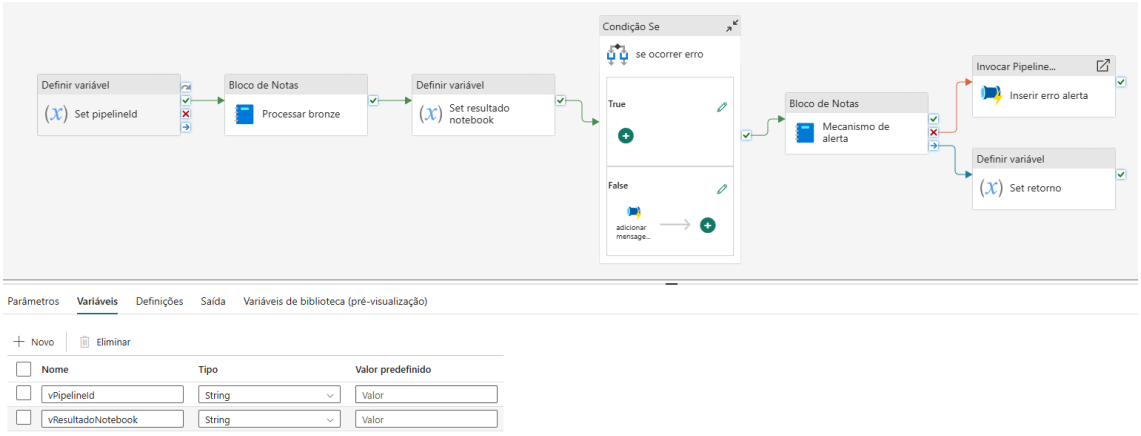

Figura 5.11: *Data pipeline* da camada bronze

5.4.4.2 Camada Prata

No caso da camada prata, existem três *pipelines*, uma por cada etapa, que são muito semelhantes. As diferenças são:

1. *Notebook* executado na segunda atividade.

2. Fluxo de controlo para execução do *notebook* de envio de alertas.

A primeira diferença é bastante direta e auto-explicativa, que é a alteração do *notebook* que necessita de ser executado em cada caso. Na primeira *pipeline*, é executado o *notebook* da etapa 1 da camada prata, na segunda o da etapa 2 e, por fim, na terceira o da etapa 3.

A segunda diferença acontece apenas na *pipeline* da última etapa da camada, porque nesta é sempre necessário executar o mecanismo de envio de alertas, pois trata-se da finalização da preparação e carregamento da camada prata. Nas duas primeiras *pipelines* não é o caso, pois são passos intermédios que constituem o processo de transformação e carregamento desta camada. Nestas duas *pipelines* apenas é executado o mecanismo de alertas, quando existem erros durante o processo.

Nas figuras 5.12, 5.13 e 5.14, são apresentadas as *data pipelines* das três etapas da camada prata. As atividades utilizadas nas implementações são as mesmas explicadas na camada bronze.

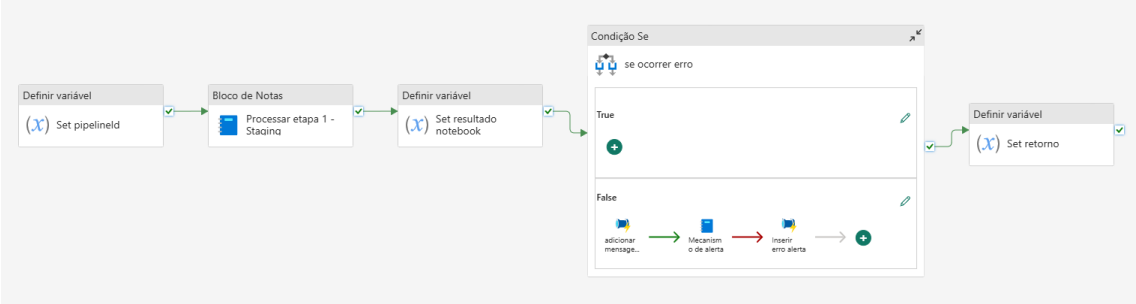

Figura 5.12: *Data pipeline* da etapa 1 da camada prata

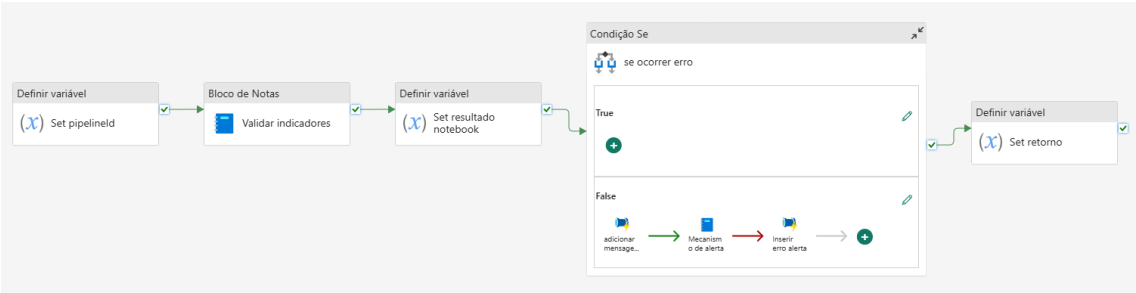

Figura 5.13: *Data pipeline* da etapa 2 da camada prata

5.4.4.3 Camada Ouro

A *data pipeline* responsável pela camada ouro segue a mesma arquitetura das *data pipelines* das camadas bronze e prata (ver figura 5.15). As atividades necessárias que foram implementadas são as seguintes:

- Gerar o Id de execução da *pipeline*.

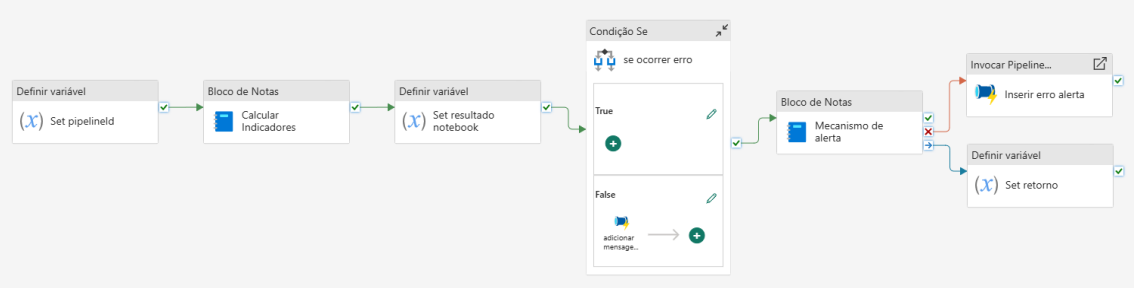

Figura 5.14: *Data pipeline* da etapa 3 da camada prata

- Executar o *notebook* de transformação e carregamento da camada ouro.
- Verificação do resultado da execução do *notebook*.
- Inserção de mensagem de erro, caso seja necessário.
- Execução do *notebook* de envio de alertas.
- Inserção de erro no envio do alerta, caso necessário.
- Definição do valor de retorno da *pipeline* - sucesso ou erro.

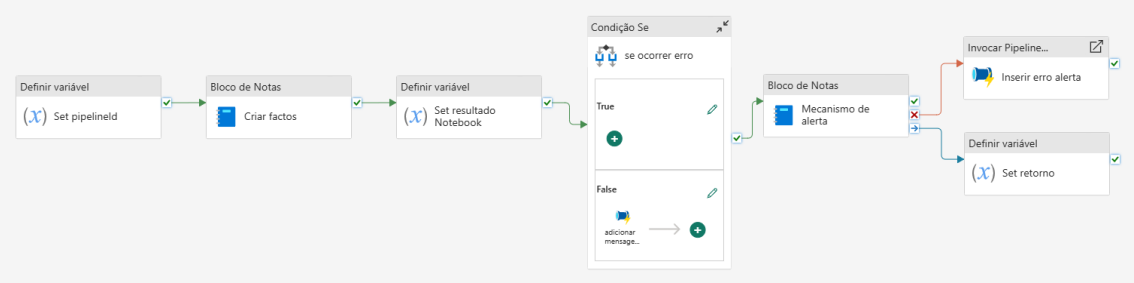

Figura 5.15: *Data pipeline* da camada ouro

Na *pipeline* da camada ouro apenas é executado o *notebook* da criação das tabelas factos, pois as dimensões são criadas no *deploy* da solução do sistema ou quando são necessárias alterações no conteúdo da mesma.

As dimensões são criadas com base em ficheiros armazenados numa pasta distinta das camadas da arquitetura, pois raramente necessitam de atualizações. No caso de ser necessário atualizar as dimensões, executa-se a pipeline "Fluxo - Criação de Dimensões"(ver figura 5.16), que insere os novos registos necessários ou atualiza os anteriores. A atualização dos registos pré-existentes é feita através da inserção dos novos dados e atualização dos metadados de controlo do registo anterior. Em cada dimensão, para além dos campos relativos à mesma, também são armazenados 3 campos de controlo, que permitem este funcionamento:

- Is_Ativo - booleano que define se o registo está ativo
- Data_Ativacao - *timestamp* de início do período ativo do registo

- Data_Desativacao - *timestamp* de fim do período ativo do registo

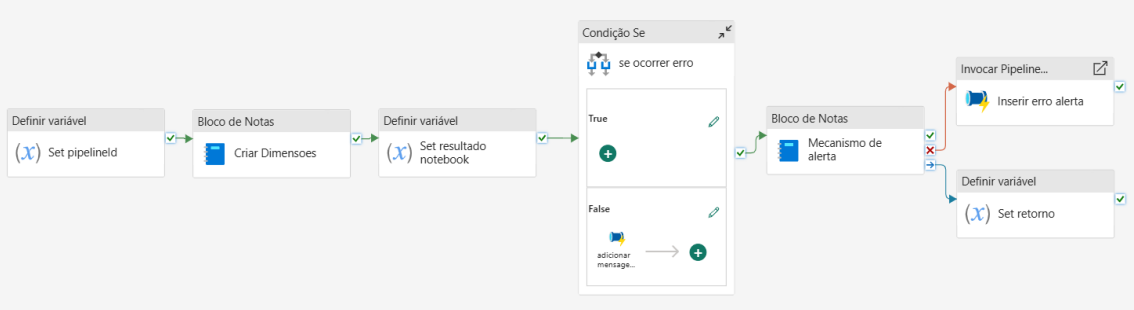

Figura 5.16: *Data pipeline* de criação de dimensões

5.4.4.4 Execução Global

De forma a culminar todos os processos anteriormente abordados, foi implementada uma *pipeline* que é responsável por executar as *pipelines* das camadas bronze, prata e ouro. A execução das mesmas necessita de seguir uma ordem sequencial, pois cada processo é dependente do anterior, com exceção do processo da camada bronze. Com esta *pipeline* de execução global, denominada "Fluxo Principal - Integração Ficheiros"e apresentada nas figuras 5.17 e 5.18, é possível executar a integração de dados completa dos ficheiros submetidos, com um único processo global e atómico.

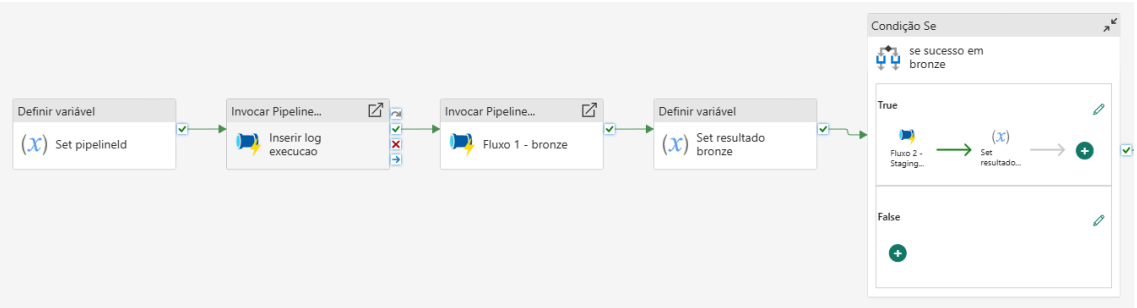

Figura 5.17: *Data pipeline* de execução global - parte 1

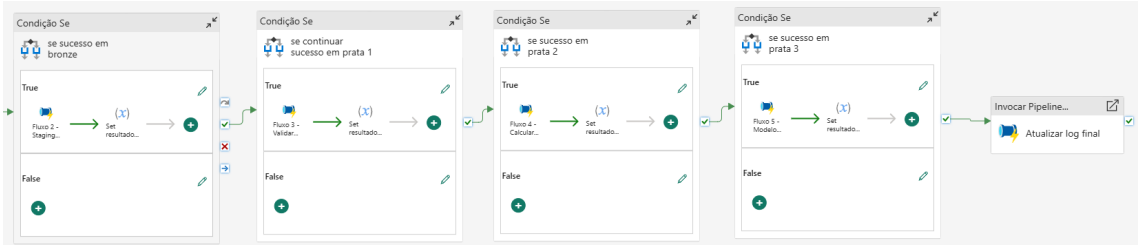

Figura 5.18: *Data pipeline* de execução global - parte 2

Foi configurado um *schedule* para esta *data pipeline* global ser executada de forma automática, todos os dias, às 01h00, como é possível observar na figura 5.19.

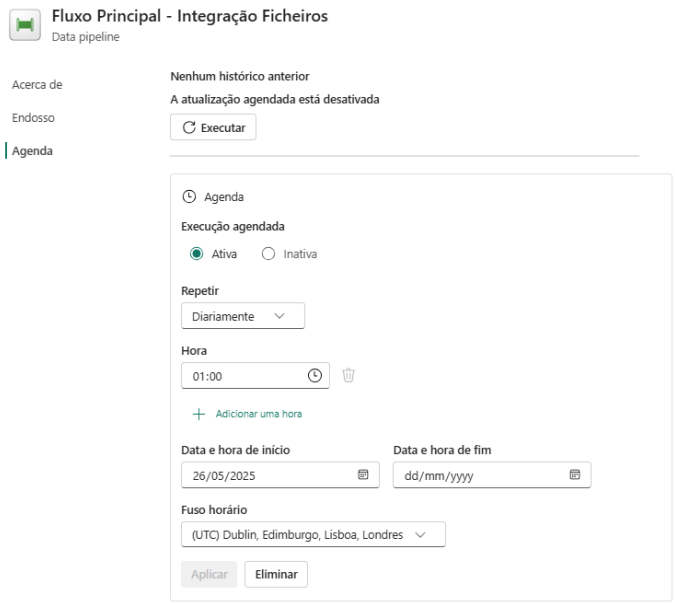

Figura 5.19: *Schedule* configurado para a *data pipeline* de execução global

Contudo, caso o utilizador pretenda integrar os dados submetidos num momento diferente do agendado, poderá executar manualmente o processo entrando na *pipeline* no *workspace* do Fabric, tal como demonstrado na figura 5.20.

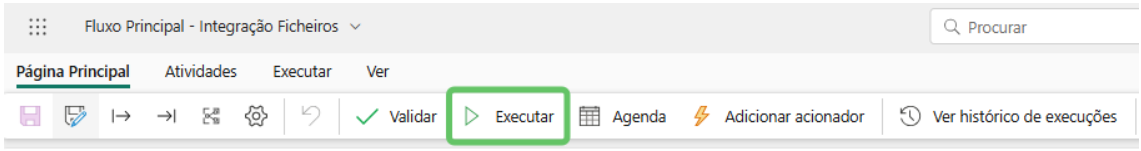

Figura 5.20: Execução manual da *data pipeline* de execução global

5.5 Relatório Power BI

Tal como definido na secção 4.4 do capítulo anterior, foi decidido implementar um relatório Power BI, constituído por um *dashboard* e uma infografia, sobre o tema de acesso à *Internet* em local fixo. As etapas de construção do relatório são:

1. Carregamento e configuração do modelo de dados.
2. Adição das funcionalidades de tradução.
3. Desenvolvimento das métricas e visuais necessários.
4. Desenvolvimento de *interface*, seguindo as maquetes definidas pela equipa de *design* da ARMIS.

5. Implementação da capa e respetivo menu.

6. Implementação do menu lateral.

7. Implementação da infografia.

5.5.1 Modelo de Dados

No momento de carregar as dimensões e factos no Power BI, também é necessário configurar o modelo de dados na própria aplicação. Todas as relações entre as tabelas de factos e de dimensões devem ser devidamente configuradas, com o objetivo de todas as cardinalidades e sentidos de filtro estarem de acordo com o pretendido. Na figura 5.21, é apresentado o modelo de dados já devidamente configurado. Pode-se observar que todas as relações possuem cardinalidade muitos para um e o modelo segue a estrutura especificada na secção 4.3.

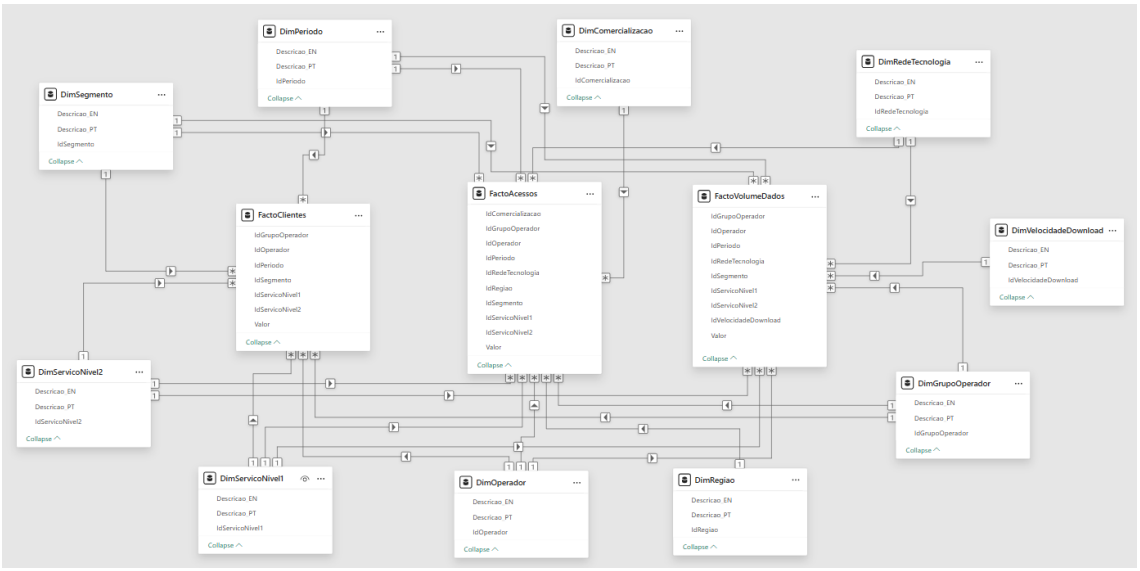

Figura 5.21: Modelo de dados do Power BI

5.5.2 Tradução

Uma das componentes importantes dos relatórios Power BI é a sua capacidade de traduções, ao nível das *labels* e dos dados.

Para garantir a acessibilidade do relatório a públicos diversificados, foi necessário implementar funcionalidades de tradução automática no Power BI. Estas funcionalidades permitem alternar entre diferentes idiomas sem necessidade de múltiplos ficheiros de relatório, simplificando a manutenção e assegurando uma experiência de utilização uniforme.

5.5.2.1 *Labels*

Nesta camada de tradução, os títulos, subtítulos, menus e demais elementos textuais da interface foram configurados com suporte a dois idiomas - português e inglês. Esta funcionalidade é suportada através da utilização da aplicação externa Translations Builder[4], que permite a alteração do texto a projetar para cada métrica e coluna, bem como adicionar *labels* independentes.

Na figura 5.22, são apresentados exemplos de traduções de métricas e na figura 5.23 exemplos da criação e tradução de *labels* independentes. Este último caso requer um passo extra, que é a criação de uma *Localized Labels Table* (ver figura 5.24) e, sempre que esta for alterada (inserção de uma *label* nova), é necessário gerar a mesma, para atualizar o modelo do Power BI.

Object Type	Property	Name	Portuguese [pt-PT]	English [en-US]
Measure	Caption	Measure[SomaTrafegoAnualCartao]	total	total
Measure	Caption	Measure[TrafegoMedioMensalTrimestralCartao]	mês / acesso	month / access
Measure	Caption	Measure[TrafegoMedioMensalAnualCartao]	mês / acesso	month / access
Measure	Caption	Measure[SomaClientesCartao]	clientes	clients

Figura 5.22: Exemplos de traduções de métricas

Object Type	Property	Name	Portuguese [pt-PT]	English [en-US]
Measure	Caption	Localized Labels[Taxa de penetração residencial]	Taxa de penetração residencial	Residential penetration rate
Measure	Caption	Localized Labels[Velocidade de acesso]	Velocidade de acesso	Access speed
Measure	Caption	Localized Labels[Segmento]	Segmento	Segment
Measure	Caption	Localized Labels[Tecnologia]	Tecnologia	Technology
Measure	Caption	Localized Labels[Tráfego]	Tráfego	Traffic
Measure	Caption	Localized Labels[Tráfego médio mensal por acesso]	Tráfego médio mensal por acesso	Average monthly traffic per access

Figura 5.23: Exemplos da criação e tradução de *labels*

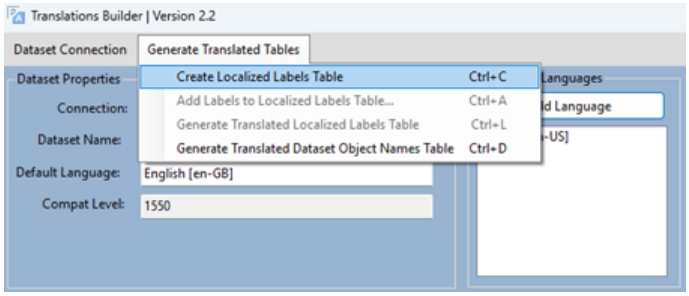

Figura 5.24: Criação da *Localized Labels Table*

5.5.2.2 Dados

Para permitir a tradução dos dados descritivos das dimensões (ex: nomes de regiões, segmentos ou serviços), foram utilizadas as colunas "Descricao_PT"e "Descricao_EN"presentes nas tabelas dimensionais. A funcionalidade que permite a tradução dinâmica destes dados consiste na utilização de *fields parameters*[5].

[4] ver https://github.com/PowerBiDevCamp/TranslationsBuilder - acedido em 27/05/2025

[5] ver https://learn.microsoft.com/en-us/power-bi/guidance/data-translation-implement-field (guia para criação e utilização de *fields parameters*) - acedido em 27/05/2025

Na figura 5.25, é apresentada a janela *pop-up* para criação inicial do *fields parameter* da dimensão de período, a título de exemplo. Inicialmente, o *fields parameter* é criado com a estrutura apresentada na figura 5.26. Contudo, é necessário alterar os campos "Descricao_PT"e "Descricao_EN"pelos nomes a serem apresentados em cada versão. Para além desta alteração, também é preciso adicionar uma nova coluna com valores "pt"e "en", para uso posterior. O resultado final do *field parameter* é apresentado na figura 5.27.

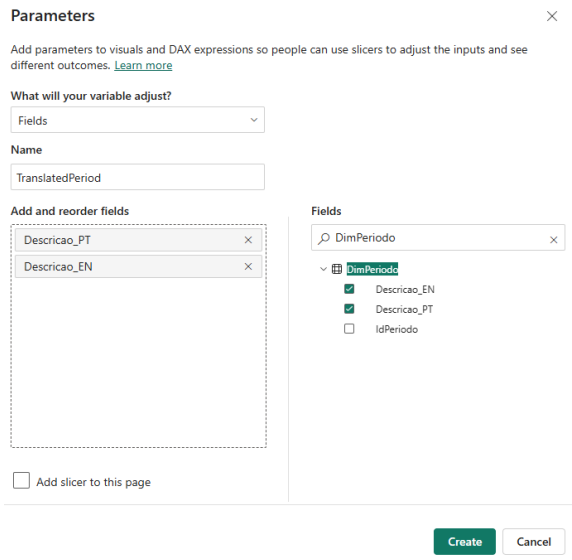

Figura 5.25: *Pop-up* para criação de *fields parameter*

```
TranslatedPeriod = {
    ("Descricao_PT", NAMEOF('DimPeriodo'[Descricao_PT]), 0),
    ("Descricao_EN", NAMEOF('DimPeriodo'[Descricao_EN]), 1)
}
```

Figura 5.26: *Fields parameter* inicial da dimensão período

```
TranslatedPeriod = {
    ("Período", NAMEOF('DimPeriodo'[Descricao_PT]), 0, "pt"),
    ("Period", NAMEOF('DimPeriodo'[Descricao_EN]), 1, "en")
}
```

Figura 5.27: *Fields parameter* final da dimensão período

Com o objetivo de controlar a escolha da apresentação do campo PT ou EN, de todos os *fiels parameters*, de forma simultânea, é necessária a criação de uma tabela central de línguas, que permita este controlo.

No Power Query[6] (funcionalidade de carregamento e transformação de dados do Power BI), foi criada a tabela Languages, apresentada na figura 5.28. O código utilizado na criação da tabela é apresentado na figura 5.28.

[6]ver https://learn.microsoft.com/en-us/power-query/power-query-what-is-power-query - acedido em 27/05/2025

	Language	LanguageId	DefaultCulture	SortOrder
1	Portuguese	pt	pt-PT	0
2	English	en	en-US	1

Figura 5.28: Tabela Languages do Power BI

Languages

```
let
  LanguagesTable = #table(type table [
    Language = text,
    LanguageId = text,
    DefaultCulture = text,
    SortOrder = number
  ], {
    {"Portuguese", "pt", "pt-PT", 0 },
    {"English", "en", "en-US", 1 }
  }),
  SortedRows = Table.Sort(LanguagesTable,{{"SortOrder", Order.Ascending}}),
  QueryOutput = Table.TransformColumnTypes(SortedRows,{{"SortOrder", Int64.Type}})
in
  QueryOutput
```

Figura 5.29: Código de criação da tabela Languages do Power BI

Por fim, na figura 5.30 é apresentado o modelo final, configurado no Power BI, com todas as relações entre os *fields parameters* e a tabela Languages, que controla os campos a serem apresentados nos visuais.

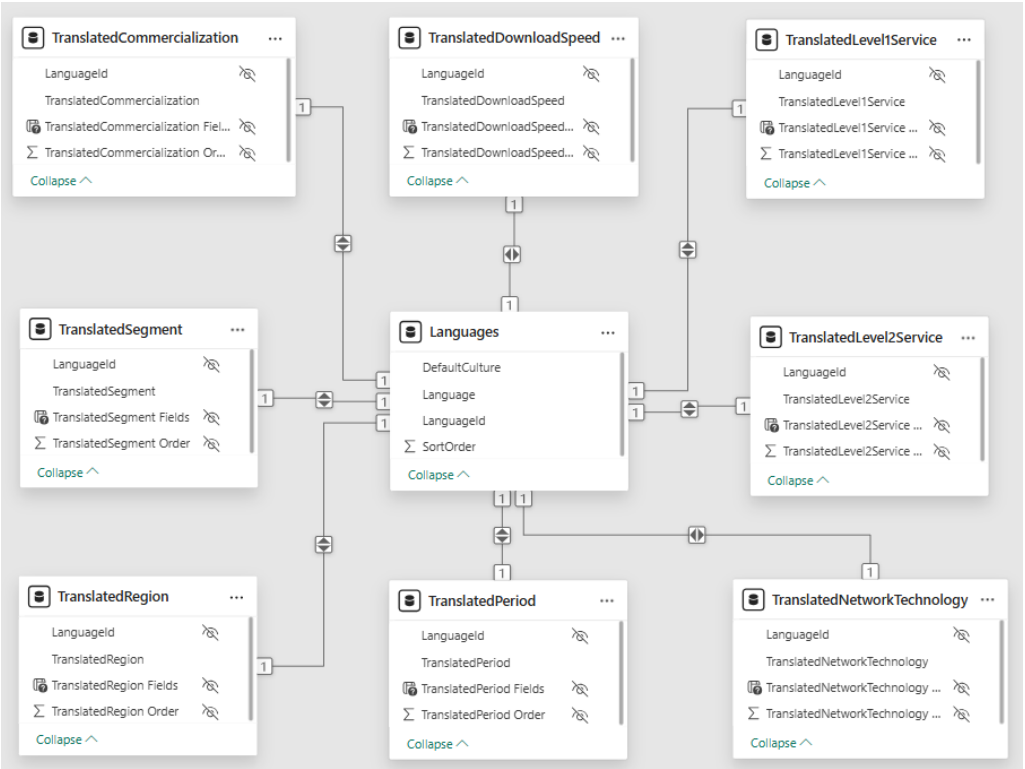

Figura 5.30: Modelo de tabelas de tradução no Power BI

5.5.2.3 Gestão de Versões

A gestão da versão linguística é feita de forma dinâmica, através do URL do relatório embutido num portal Web (sistema implementado por outra equipa da ARMIS, mas no contexto do mesmo projeto). Dependendo do idioma selecionado no portal (via *toggle*), são adicionadas duas *substrings* ao URL do relatório Power BI. A primeira define o idioma da interface (&language=pt-PT ou &language=en-US) que impacta todas as *labels* e a segunda filtra a tabela Languages que controla a exibição dos dados textuais provenientes dos *filters parameters* das dimensões (&filter=Languages/Language eq 'Portuguese' ou &filter=Languages/Language eq 'English'). Esta abordagem garante que a versão correta é carregada sem qualquer ação adicional por parte do utilizador final.

5.5.3 Métricas e Visuais

Para a construção do relatório, foram definidas diversas métricas e indicadores-chave de desempenho (KPIs), com o objetivo de permitir uma análise completa e fiável dos dados relativos ao acesso à *Internet* em local fixo. Estas métricas foram implementadas com recurso à linguagem DAX (*Data Analysis Expressions*), diretamente no Power BI.

As principais métricas desenvolvidas incluem:

- Número total de acessos: total de acessos agregados por período, com possibilidade de desagregação por tecnologia, segmento ou velocidade.
- Variação homóloga de acessos (ver excerto de código 5.8): cálculo da diferença percentual face ao mesmo período do ano anterior.
- Penetração residencial: número de acessos por 100 famílias, com base em dados populacionais de referência.
- Distribuição percentual: cálculo da percentagem de acessos por categoria (tecnologia, segmento, etc.).
- Tráfego total e médio mensal: análise da evolução do volume de tráfego agregado e por cliente.
- Número de prestadores ativos: total de prestadores registados no período analisado.
- Quota de mercado por prestador: percentagem de acessos por prestador.

```
1 VarHomologaAcessos =
2 VAR ID_Periodo_Atual = SELECTEDVALUE(DimPeriodo[Descricao_PT])
3
4 VAR ID_PerAnterior =
5     SWITCH(
6         TRUE(),
7         LEN(ID_Periodo_Atual) = 4, ID_Periodo_Atual - 1,                 // Anual
```

```
        LEN(ID_Periodo_Atual) = 6, BLANK(),                                    // Mensal
        LEN(ID_Periodo_Atual) = 7,                                             //
    Trimestral
            LEFT(ID_Periodo_Atual, 4) - 1 & RIGHT(ID_Periodo_Atual, 3)
    )

VAR ValorAtual = SUM(FactoAcessos[Valor])

VAR ValorAnterior =
    CALCULATE(
        SUM(FactoAcessos[Valor]),
        ALL(DimPeriodo),
        DimPeriodo[Descricao_PT] = ID_PerAnterior
    )

VAR Idioma = SELECTEDVALUE(Languages[LanguageId])

RETURN
    IF(
        ISBLANK(ValorAnterior) || ValorAnterior = 0,
        IF(Idioma = "pt", "n.d.", "n.a."),
        DIVIDE(ValorAtual, ValorAnterior, 0) - 1
    )
```

Excerto de Código 5.8: Métrica DAX para variação homóloga de acessos

Os visuais foram escolhidos de acordo com o tipo de informação a representar, privilegiando a clareza, a comparação visual e a capacidade de resposta a interações do utilizador. Entre os principais tipos de visual utilizados, destacam-se:

- Segmentações (*slicers*): permitem ao utilizador filtrar os dados por período, prestador, segmento ou outros atributos relevantes.
- Gráficos de barras horizontais e verticais: ideais para comparações entre categorias, como quota de mercado ou distribuição por segmento.
- Gráficos de linha: utilizados para representar evoluções temporais, como número de acessos ou tráfego ao longo do tempo.
- Gráficos de área: úteis para distribuições de métricas, tal como número de acessos.
- Mapas: aplicados para representar a distribuição geográfica de acessos a nível nacional.
- Cartões: utilizador para destacar valores absolutos e variações em destaque, como os KPIs principais.

A seleção dos visuais foi baseada nas boas práticas de visualização de dados, assegurando legibilidade, coerência com os objetivos analíticos e uma experiência de utilizador clara e eficiente.

5.5.4 *Dashboard*

O *dahsboard* resultante cumpre com a especificação apresentada na secção 4.4, tanto em métricas a calcular ou valores, como no *design* a seguir. Nas figuras 5.31, 5.32, 5.33, 5.34, 5.35 e 5.36 são apresentadas as páginas do *dashboard* implementado. Na figura 5.33, é apresentada a página de análise de acessos e clientes com o menu lateral aberto. A página de perfil de utilizador não foi implementada, por falta de especificação da parte do cliente e por falta de dados desta componente, no momento da implementação do sistema.

Figura 5.31: Capa do *dashboard*

Figura 5.32: Página de acessos e clientes do *dashboard*

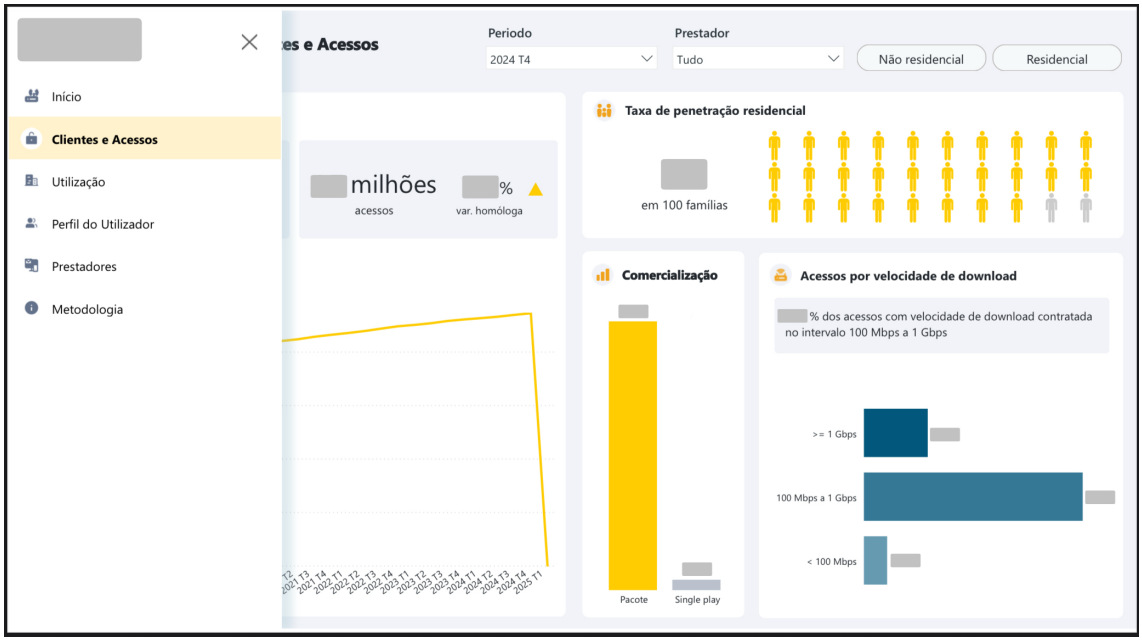

Figura 5.33: Página de acessos e clientes do *dashboard* com menu lateral aberto

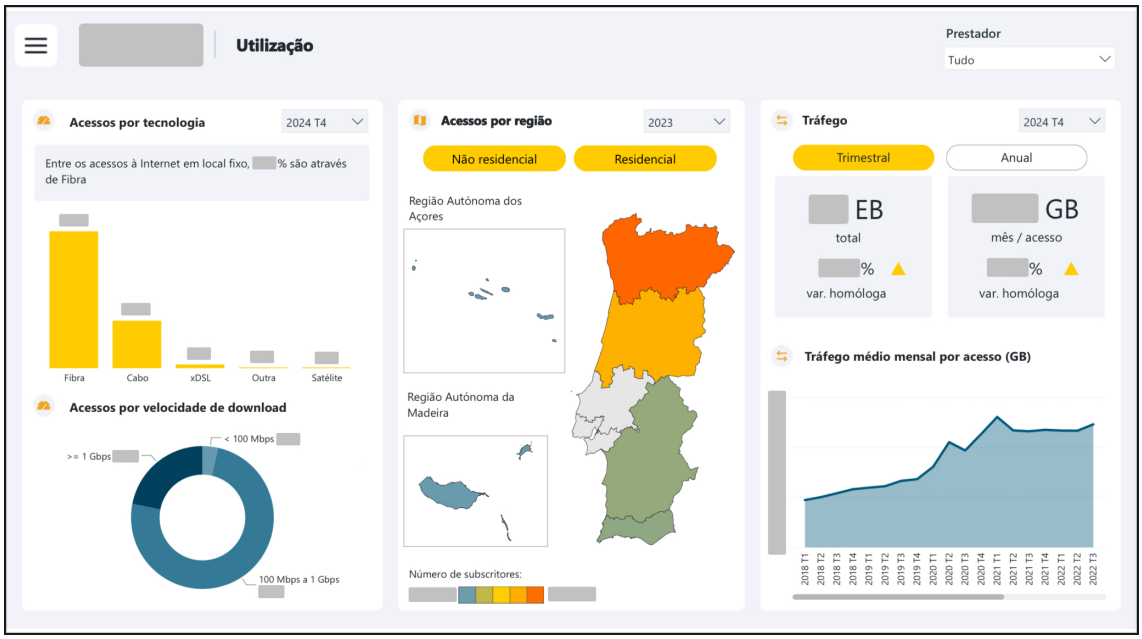

Figura 5.34: Página de utilização do *dashboard*

Figura 5.35: Página de prestadores do *dashboard*

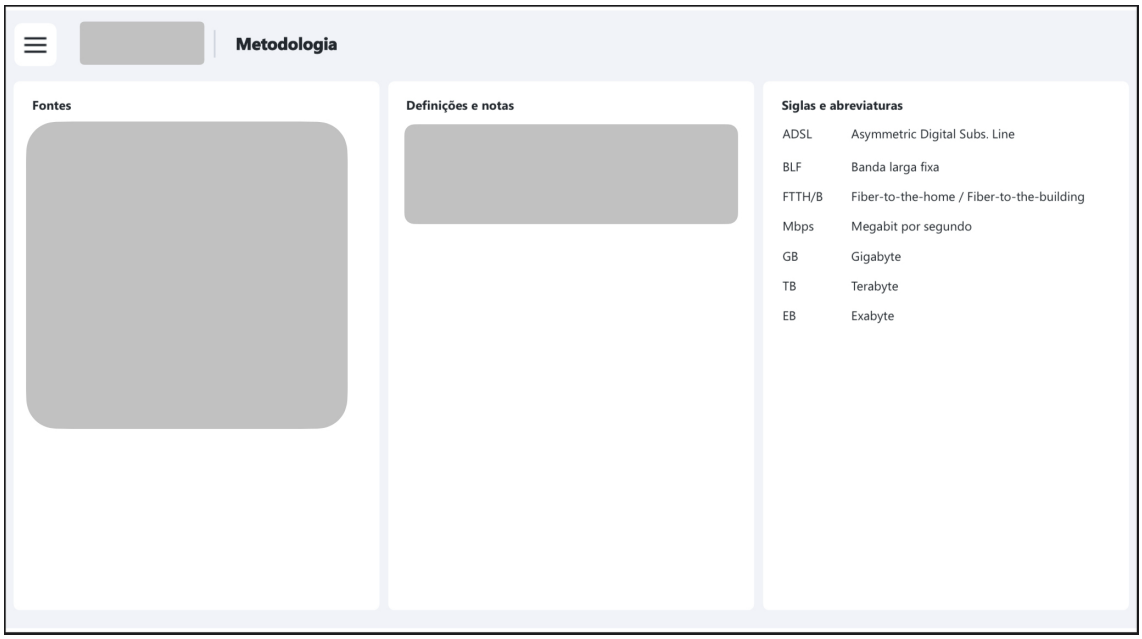

Figura 5.36: Página de metodologia do *dashboard*

5.5.5 Infografia

A infografia implementada reutiliza os visuais e análises do *dashboard* resultante. Em termos de *design*, também foram seguidos os *mockups* criados pela equipa de *design* da ARMIS, reutilizando o trabalho aplicado durante a implementação do *dashboard*. Na figura 5.37, é apresentada a infografia. Para demonstrar a implementação da tradução para inglês, na figura 5.38, é apresentada a versão inglesa da infografia.

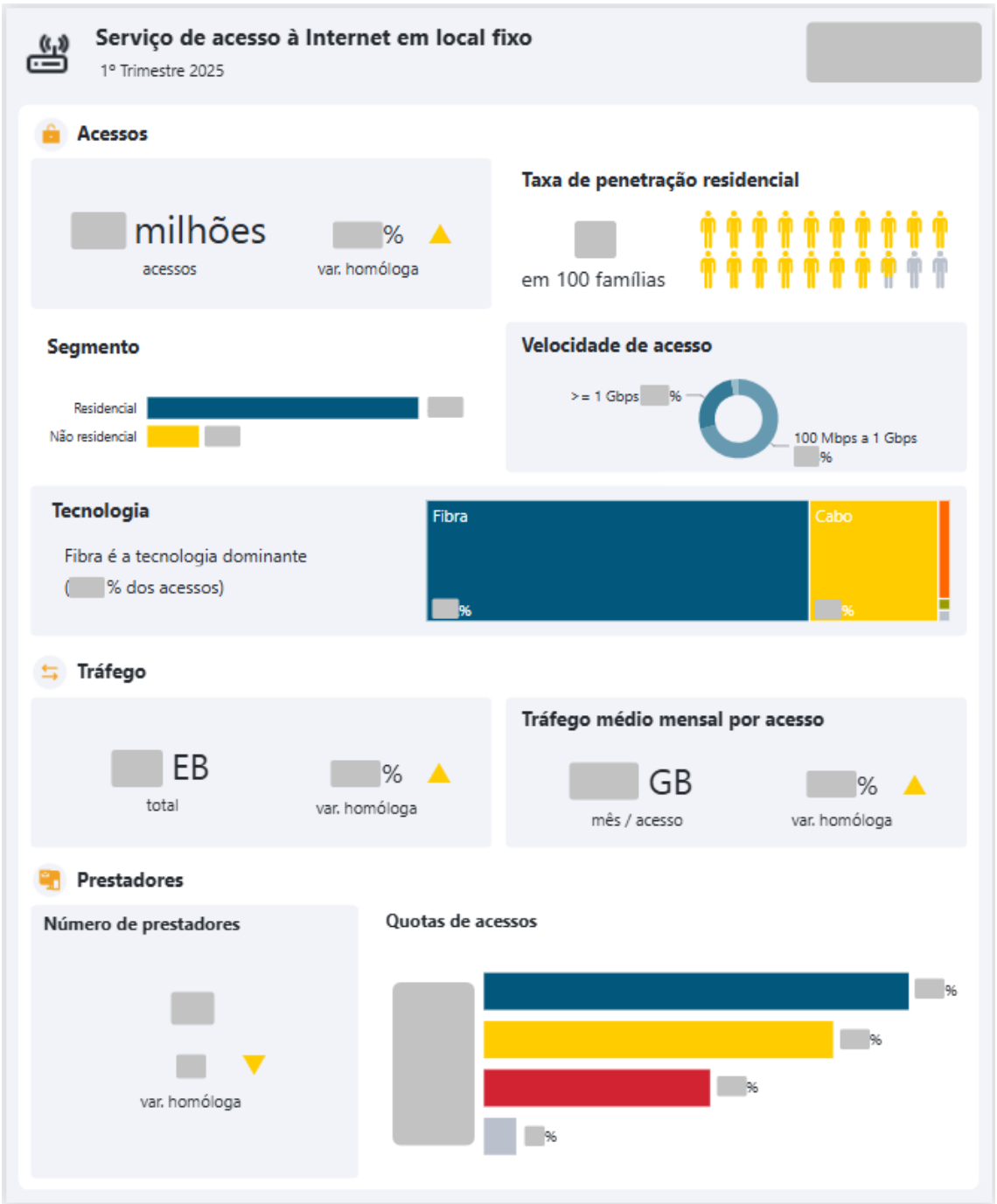

Figura 5.37: Infografia - versão PT

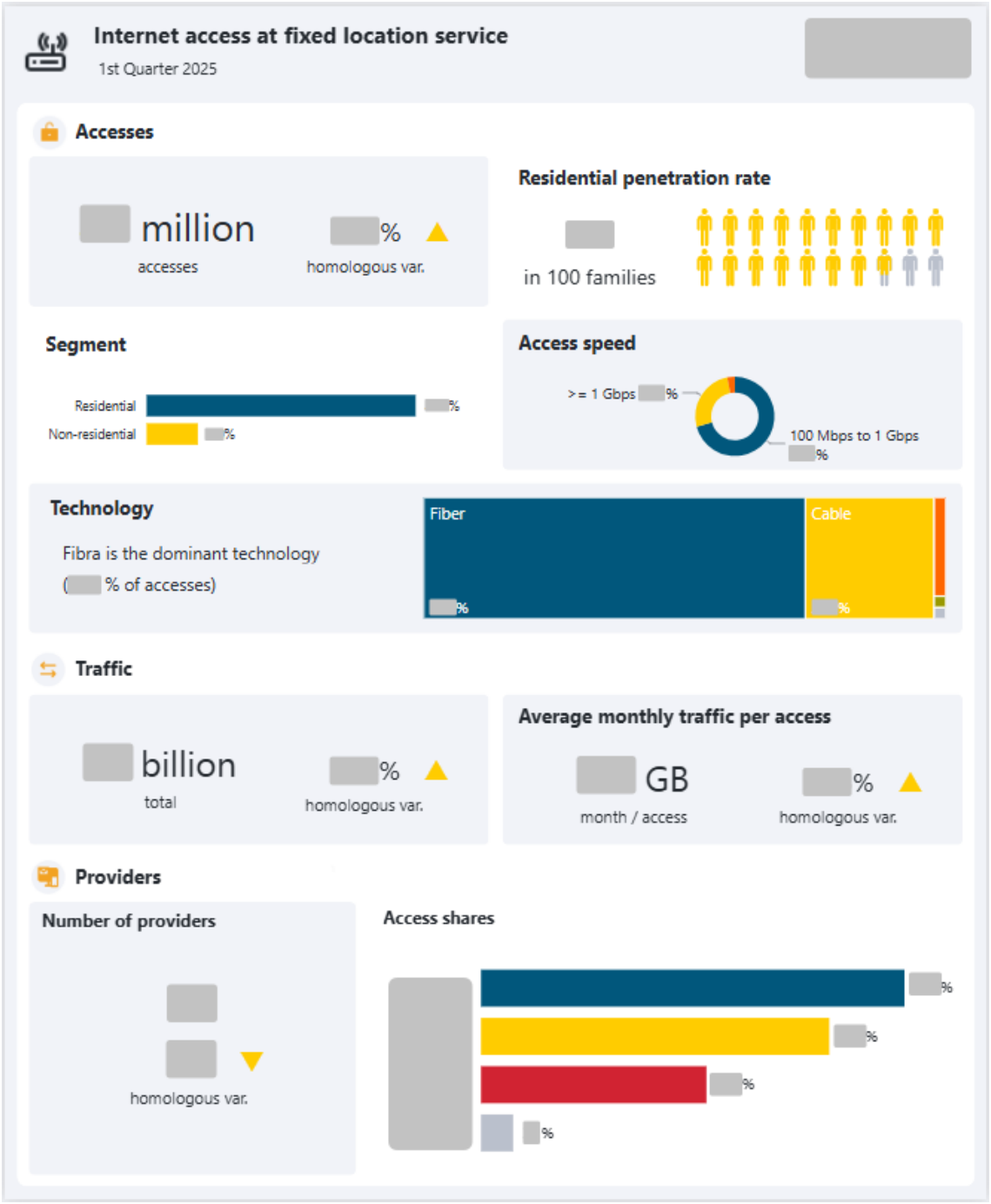

Figura 5.38: Infografia - versão EN

Capítulo 6

Avaliação da Solução

A avaliação da solução desenvolvida incide sobre três componentes principais: o modelo de dados do *data lakehouse*, o processo de transformação e integração de dados (ELT) e o relatório analítico, desenvolvido em Power BI. Em todas as fases do projeto, foram conduzidos testes funcionais e validações específicas com o intuito de garantir a robustez, a coerência e a eficácia de cada componente. Esta avaliação segue os critérios previamente definidos na secção 1.6.3, que abrangem a robustez/desempenho, a escalabilidade e a utilidade prática da solução desenvolvida.

6.1 Modelo de Dados do *Data Lakehouse*

A avaliação do modelo de dados incide sobre dois critérios distintos:

- **Critério 1 - Qualidade do Modelo Dimensional** - Associado ao parâmetro de 'Robustez-Desempenho'.
- **Critério 3 - Cobertura e Escalabilidade** - Relacionado com o parâmetro 'Escalabilidade'.

O Critério 1 visa garantir que o modelo foi corretamente implementado, apresentando dados consistentes, completos e com integridade referencial validada. Neste contexto, o Critério 3 pretende assegurar que a estrutura do modelo é suficientemente flexível e abrangente, permitindo a sua evolução futura com a integração de novos temas analíticos ou fontes de dados adicionais.

O modelo de dados implementado no *data lakehouse* foi concebido de acordo com os princípios do *star schema*, conforme detalhado na secção 4.3. A estrutura definida, composta por três tabelas de factos e dez dimensões, em estrela, revelou-se adequada, pois permite uma configuração rápida e eficiente das principais análises, requeridas pela organização cliente, especialmente no Power BI.

De forma a avaliar o modelo de dados, foram executadas diversas *queries*, no SQL Server Management Studio (SSMS)[1], através do *SQL endpoint* da *lakehouse*, disponibilizado pelo Fabric. Durante a avaliação, confirmou-se a integridade referencial entre factos e dimensões, a consistência dos dados armazenados e a adequação da granularidade definida em cada tabela. A validação

[1] ver https://learn.microsoft.com/en-us/ssms/sql-server-management-studio-ssms - acedido em 14/06/2025

foi efetuada através da comparação com dados de histórico carregados pelo cliente, o que permitiu verificar a veracidade dos dados no modelo e o sucesso do processo de integração. Esta análise permitiu validar o **Critério 1 - Qualidade do Modelo Dimensional**, garantindo que os dados se apresentavam completos, coerentes, precisos e alinhados com as necessidades analíticas da organização.

A estrutura modular do modelo permite extensibilidade futura, caso se verifique a necessidade de incluir novas fontes de dados ou expandir as análises existentes, adicionando novas tabelas de dimensão. Além disso, a organização por áreas temáticas (acessos, clientes e tráfego) facilitou a interpretação e exploração dos dados, por parte dos utilizadores do cliente, durante a fase de testes e validação da solução, aumentando a capacidade de resposta da solução a novas necessidades analíticas. Com isto, o **Critério 3 - Cobertura e Escalabilidade** foi validado.

6.2 Processo ELT (*Extract*, *Transform*, *Load*)

A avaliação do processo ELT incide sobre dois critérios:

- **Critério 2 - Performance do Processo de Integração de Dados** - Enquadrado no parâmetro 'Robustez-Desempenho'.
- **Critério 3 - Cobertura e Escalabilidade** - Associado ao parâmetro 'Escalabilidade'.

O Critério 2 visa aferir a eficiência dos processos de ingestão e transformação de dados, sendo avaliado com base nos tempos médios de execução observados e na estabilidade do sistema, durante as execuções das várias etapas. Neste caso, o Critério 3 procura verificar se a arquitetura e implementação do processo ELT garantem flexibilidade e capacidade de adaptação a novos requisitos, como a introdução de novas fontes de dados ou a execução simultânea de múltiplos fluxos em paralelo. Ambos os critérios são avaliados com base em testes experimentais realizados sobre diferentes cenários.

O processo de extração, carregamento e transformação de dados (ELT) foi estruturado com base na arquitetura *Medallion* (bronze, prata e ouro), e implementado com recurso a *notebooks* e orquestração por *data pipelines*.

Os *notebooks*, desenvolvidos para cada camada, executaram com sucesso todas as operações de ingestão, limpeza, transformação e modelação de dados. Foram utilizados *dataframes* Spark e Pandas, dependendo das operações a efetuar. O sucesso das operações aplicadas foi validado e verificado através de *querying*, analisando os dados armazenados, tal como mencionado na secção anterior.

As *data pipelines* desempenharam um papel fundamental na automatização da orquestração de tarefas. Foram testadas execuções em horários programados e observada a correta sequência e dependência entre atividades, através da exploração do histórico de execuções das *pipelines*. Confirmou-se que todas as dependências lógicas entre etapas, desde a camada bronze até à camada ouro, foram respeitadas e implementadas com sucesso. Adicionalmente, com recurso a *querying* à base de dados de controlo, verificou-se que os dados de controlo foram corretamente registados,

tanto em execuções bem-sucedidas como em situações de erro, assegurando a rastreabilidade e auditabilidade dos processos.

Durante a especificação da solução, não foram estabelecidos limites máximos de tempo para a execução global do processo de integração, nem para as suas etapas individuais. Esta decisão deve-se ao facto de o sistema ser executado de forma periódica e não requerer processamento em tempo real, pelo que a prioridade incide na fiabilidade, integridade e rastreabilidade dos dados. Ainda assim, os tempos médios obtidos durante os testes demonstraram ser bastante satisfatórios, tendo em conta o volume e a diversidade dos dados processados, bem como as transformações aplicadas, cumprindo o **Critério 2 - Performance do Processo de Integração de Dados**.

Para uma submissão e integração de um ficheiro, observaram-se os seguintes tempos médios de execução por camada:

- **Bronze** – 30 segundos
- **Prata** – 3 minutos (1 minuto por etapa)
- **Ouro** – 30 segundos

Adicionalmente, a possibilidade de execuções paralelas em cada camada permite manter a eficiência do processo, mesmo com múltiplos ficheiros em simultâneo, evitando o aumento proporcional dos tempos de execução e reforçando a escalabilidade da solução. Foi testada a integração de dez ficheiros em simultâneo, e os tempos médios de execução mantiveram-se iguais aos apresentados para a integração de um único ficheiro.

A arquitetura modular adotada, bem como a separação clara entre as diferentes camadas de transformação (bronze, prata e ouro), facilita a introdução de novas fontes de dados e a adaptação a novos requisitos. Caso seja necessário, podem ser criados módulos extra, compostos por *notebooks* e *data pipelines* independentes, como, por exemplo, para integrar algum tipo de informação externa, que necessite de processos distintos dos implementados. Desta forma, para além do bom desempenho, o processo ELT contribui também para o **Critério 3 - Cobertura e Escalabilidade**, assegurando que a solução poderá evoluir com facilidade para acomodar novas necessidades analíticas e operacionais.

Por fim, os mecanismos de alerta, implementados por envio de *e-mails* automáticos, demonstraram a sua utilidade na monitorização do processo. Foram testadas diferentes situações (execução com sucesso, falhas de dados, erro de carregamento) e, em todos os cenários, os alertas foram enviados no formato configurado e com a informação necessária dos eventos ocorridos, para apoio ao diagnóstico e resolução, se necessário.

6.3 Relatório Power BI

A avaliação do relatório analítico desenvolvido incide sobre o **Critério 4 - Utilização do Relatório**, enquadrado no parâmetro 'Utilidade Prática'. Este critério tem como objetivo analisar se a solução responde efetivamente às necessidades dos utilizadores finais, tanto ao nível funcional

como visual. A metodologia aplicada consistiu na observação da utilização do relatório Power BI por parte dos principais *stakeholders* da organização, complementada com a recolha de *feedback* orientado à experiência de utilização. Esta abordagem permitiu aferir a adequação da informação disponibilizada, a clareza da apresentação visual dos dados e a eficácia das funcionalidades implementadas, ao mesmo tempo que possibilitou identificar pontos de melhoria.

O relatório, descrito na secção 5.5, foi composto por um *dashboard* interativo e uma infografia estática, ambos desenvolvidos com base nos *mockups* fornecidos pela equipa de *design* da ARMIS. A estrutura e *design* do relatório respeitam os princípios de clareza visual, navegação eficiente e acessibilidade de conteúdos, definidos pela equipa.

O modelo de dados importado para o Power BI foi configurado corretamente, com todas as relações entre factos e dimensões verificadas, garantindo a coerência analítica e a propagação correta dos filtros entre visuais. O modelo configurado e validado é apresentado na secção 5.5.1.

As funcionalidades de tradução foram verificadas, com confirmação da comutação entre as versões portuguesa e inglesa, tanto ao nível das *labels* como dos dados descritivos, conforme descrito na secção 4.4.

Foram ainda avaliadas as métricas DAX desenvolvidas, nomeadamente os cálculos de variação homóloga, quotas de mercado e totais agregados, através da comparação com o cálculo manual das mesmas, com base em dados de histórico do cliente. As métricas comportaram-se corretamente em diferentes contextos e combinações de segmentações.

O relatório inclui um conjunto diversificado de visuais (gráficos de barras, linhas, cartões, *donuts*, entre outros) cumprindo as normas estipuladas pela equipa de *design* da ARMIS, quanto à interatividade, desempenho e interpretação visual.

A utilização do relatório, pelos *stakeholders* do cliente, foi acompanhada com o objetivo de avaliar o seu conteúdo e interatividade, resultando na recolha de sugestões de melhoria. Com base no *feedback* recebido, foram implementadas diversas correções e adicionadas novas funcionalidades. Um exemplo concreto é a introdução do grupo "Outros"nas quotas de mercado, que agrega os operadores (ou grupos de operadores) com uma quota inferior a 0,5%, no intervalo temporal e nas segmentações definidas para a análise.

Relativamente à infografia, confirmou-se a sua legibilidade em formato impresso e a clareza na apresentação dos dados do último período disponível. A reutilização dos visuais do *dashboard* garantiu consistência e eficiência no processo de implementação.

O **Critério 4 - Utilização do Relatório** foi, assim, comprovado, através da verificação da experiência de utilização e da adequação do relatório às necessidades analíticas da organização.

6.4 Conclusão da Avaliação

A avaliação global da solução desenvolvida permite concluir que os objetivos inicialmente definidos foram plenamente atingidos. A plataforma foi testada em ambiente real, com recurso a dados confidenciais e cenários representativos da atividade do cliente, tendo revelado eficácia e alinhamento com os requisitos do projeto.

O sistema demonstrou ser tecnicamente robusto, reutilizável e escalável, garantindo a estabilidade necessária para um funcionamento contínuo e eficiente. O modelo de dados, construído com base numa estrutura dimensional (*star schema*), revelou-se adequado às necessidades analíticas da organização, ao mesmo tempo que permite extensibilidade futura. O processo de integração, suportado por *notebooks* e *data pipelines* automatizadas, assegura a fiabilidade e rastreabilidade da integração de dados, com tempos de execução aceitáveis e gestão eficiente de recursos.

Do ponto de vista analítico, o relatório Power BI permitiu explorar os dados de forma clara e estruturada, com funcionalidades adequadas ao perfil dos utilizadores finais. A possibilidade de comutação entre versões multilíngues (portuguesa e inglesa), a variedade de visuais e a validação das métricas implementadas foram aspetos determinantes para a qualidade do relatório. O envolvimento dos utilizadores no processo de avaliação, através da recolha e integração de *feedback*, reforçou a utilidade prática da solução e a veracidade da informação apresentada.

Adicionalmente, a centralização da solução em ambiente *cloud*, aliada à automatização dos fluxos de integração e análise, constituiu um fator diferenciador. Esta abordagem facilitou a manutenção, a gestão dos artefactos e a evolução da plataforma, reduzindo a dependência de processos manuais e aumentando a eficiência global da gestão da informação.

Deste modo, e tendo por base os testes efetuados, validações técnicas e *feedback* dos utilizadores, conclui-se que a solução cumpre os quatro critérios definidos no plano metodológico:

- Critério 1 - Qualidade do Modelo Dimensional
- Critério 2 - Performance do Processo de Integração de Dados
- Critério 3 - Cobertura e Escalabilidade
- Critério 4 - Utilização do Relatório

A solução implementada representa, assim, uma ferramenta sólida e eficaz de suporte à gestão de informação e tomada de decisões, com margem clara para crescimento e adaptação a novas necessidades futuras.

Capítulo 7

Conclusões e Trabalho Futuro

Este capítulo apresenta uma reflexão final sobre o trabalho desenvolvido ao longo da dissertação, destacando os principais contributos, limitações encontradas e oportunidades futuras de evolução da solução proposta e desenvolvida.

7.1 Síntese do Trabalho Realizado

Ao longo desta dissertação, foi desenvolvida uma solução completa de integração, transformação e disponibilização de dados, com o objetivo de centralizar a informação proveniente de múltiplas fontes, otimizando o acesso e apoiando a tomada de decisão numa empresa cliente da ARMIS. A solução foi materializada através da construção de uma arquitetura *data lakehouse*, suportada por um modelo de dados dimensional e complementada por relatórios interativos em Power BI.

A componente técnica da solução envolveu a modelação de uma base de dados analítica segundo o paradigma *star schema*, a implementação de um processo de integração e transformação de dados (ELT), suportado por *notebooks* Python, *data pipelines* automatizadas e mecanismos de alerta. Por fim, foi desenvolvido um relatório Power BI multilíngue, composto por um *dashboard* interativo e uma infografia estática.

Todos os artefactos foram alvo de testes e validação rigorosa. Os resultados demonstram que a solução cumpre os objetivos propostos, apresentando elevados níveis de eficiência, escalabilidade e facilidade de manutenção.

7.2 Perguntas de Investigação

A dissertação procurou responder a duas perguntas de investigação principais: (P1) Qual a melhor tecnologia para desenvolver a solução necessária? e (P2) Quais as técnicas e ferramentas a utilizar em cada etapa ou processo de desenvolvimento da solução?

Quanto à primeira, a seleção tecnológica foi orientada pelos responsáveis da ARMIS e da entidade cliente, tendo sido fundamentada através de uma análise crítica dos requisitos, restrições

e objetivos específicos do projeto. A escolha recaiu sobre o ecossistema Microsoft Fabric, uma plataforma moderna, *cloud-first*, com capacidades integradas para Engenharia de Dados e *Business Intelligence*, destacando-se pela centralização, escalabilidade e facilidade de gestão.

Relativamente à segunda pergunta, a solução implementada recorreu a diversas técnicas e ferramentas adequadas a cada etapa do ciclo de vida dos dados. A ingestão e transformação de dados foram realizadas com recurso a *notebooks* Python, organizados segundo a arquitetura *Medallion*. A automatização do processo foi garantida por *data pipelines* com agendamentos e alertas. Por fim, a exploração dos dados foi suportada por um relatório Power BI com funcionalidades interativas e multilíngues, promovendo análises visuais acessíveis e informativas.

7.3 Contributos

A dissertação apresenta um contributo prático e relevante para a realidade empresarial, ao demonstrar como tecnologias modernas podem ser aplicadas de forma eficaz para resolver problemas reais de dispersão e fragmentação da informação. Entre os principais contributos, destaca-se a implementação de uma solução centralizada na *cloud*, que representa um avanço significativo face a abordagens tradicionais mais segmentadas e manuais.

Ao adotar uma arquitetura baseada em *data lakehouse* e em ferramentas *cloud-first*, como o Microsoft Fabric, foi possível consolidar num único espaço todas as etapas do ciclo de vida dos dados — desde a ingestão e transformação até à análise visual e disponibilização dos relatórios. Esta centralização facilita significativamente a gestão dos artefactos e a manutenção dos processos, reduzindo dependências entre sistemas e promovendo maior consistência, rastreabilidade e governança.

Além disso, a solução é facilmente escalável, adaptável a novas fontes de dados e requisitos analíticos, e garante maior segurança e controlo na manipulação da informação. A gestão em *cloud* também permite que equipas técnicas e funcionais acedam aos componentes do sistema de forma integrada, mesmo em ambientes distribuídos.

Outros contributos relevantes incluem:

- A implementação de um modelo de dados dimensional robusto, com base em *star schema*, capaz de suportar a análise multidimensional dos temas mais relevantes para o cliente.
- A automatização dos processos de integração e transformação de dados com recurso a *notebooks* e *data pipelines*, minimizando a intervenção manual.
- O desenvolvimento de um relatório Power BI multilíngue, com *dashboard* interativo e infografia estática, alinhado com as necessidades analíticas e de comunicação visual do cliente.
- A demonstração de boas práticas nas áreas de Engenharia de Dados e *Business Intelligence*, reforçadas pela utilização de uma arquitetura moderna, modular, segura e sustentável.

7.4 Limitações

Apesar dos resultados positivos, o projeto enfrentou algumas limitações:

- O tempo disponível para a realização do projeto foi limitado, o que condicionou a possibilidade de explorar todos os tópicos ou implementar funcionalidades adicionais sugeridas pelo cliente.
- Por se tratar de um projeto privado e sensível, vários detalhes e artefactos foram deliberadamente ocultados ou anonimizados, o que limitou a exposição completa de certos aspetos técnicos e de negócio na dissertação.
- Algumas fontes de dados relevantes para o cliente ainda não estavam plenamente integradas no momento da entrega, por se encontrarem fora do âmbito do projeto atual ou por dependerem de desenvolvimentos paralelos.

7.5 Trabalho Futuro

Existem várias oportunidades para o desenvolvimento e evolução da solução implementada:

- Expansão do Modelo de Dados: o modelo atual pode ser estendido com novas tabelas factuais e dimensionais, de forma a suportar novos tipos de análises, à medida que o cliente disponibilize novas fontes de dados.
- Criação de Novos Relatórios: para além do relatório sobre acessos à *Internet* em local fixo, estão previstos novos relatórios sobre temas como televisão, pacotes de serviços ou evolução de preços, utilizando a mesma base tecnológica e estrutura visual.
- Integração de Módulos Complementares: o cliente demonstrou interesse em integrar novos módulos com características específicas, que requerem adaptações ao modelo de dados e à lógica de integração atual.
- Aprimoramento de Monitorização e Alertas: poderá ser desenvolvido um painel de controlo central para acompanhar em tempo real o estado das execuções das *data pipelines* e os alertas gerados, melhorando o controlo operacional.
- Exploração de Técnicas Avançadas de Análise: futuras versões da solução poderão incorporar componentes de análise preditiva, com recurso a algoritmos de *machine learning*, aplicados aos dados integrados.

7.6 Considerações Finais

O projeto desenvolvido representa uma solução sólida, escalável e tecnicamente avançada para um problema real de negócio, demonstrando a aplicação prática dos conhecimentos adquiridos ao

longo do percurso académico. A dissertação reforça a importância da integração entre Engenharia de Dados e *Business Intelligence*, através da construção de uma plataforma centralizada, automatizada e orientada à análise estratégica.

A centralização da solução em ambiente *cloud* foi um fator diferenciador, permitindo maior controlo sobre a gestão dos artefactos, simplificando a manutenção dos processos e potenciando a escalabilidade e a evolução contínua do sistema. A adoção de ferramentas modernas e integradas revelou-se essencial para garantir uma solução sustentável, eficiente e adaptável às necessidades reais da organização.

Do ponto de vista pessoal e académico, o projeto constituiu uma experiência extremamente enriquecedora. O contacto com uma realidade empresarial concreta e a responsabilidade de desenvolver uma solução com aplicabilidade direta permitiram consolidar competências técnicas em áreas como modelação e integração de dados e visualização analítica. Adicionalmente, o projeto fomentou o desenvolvimento de capacidades de análise crítica, autonomia, organização e comunicação, essenciais para a vida profissional futura. Esta experiência reforçou também a importância do trabalho colaborativo e da adaptação a contextos exigentes e dinâmicos.

Simultaneamente, o trabalho desenvolvido gerou valor acrescentado tanto para a ARMIS, enquanto entidade promotora e parceira tecnológica, como para o cliente final, ao oferecer uma solução eficaz para a gestão e análise de grandes volumes de informação dispersa.

Referências

[1] O que é um data lakehouse e como funciona? | Google Cloud — cloud.google.com. https://cloud.google.com/discover/what-is-a-data-lakehouse?hl=pt-PT. [Acedido em 04-04-2025].

[2] O que é uma dimensão conformada? https://pt.linkedin.com/advice/3/what-conformed-dimension-skills-data-warehousing?lang=pt. [Acedido em 11-12-2024].

[3] Alexander Aguilar-Chávez, Jeshuá Banda-Barrientos, e Michael Cabanillas-Carbonell. Business intelligence, based on the ralph kimball methodology, for decision-making in general management. Em *2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, páginas 643–646, 2021.

[4] Benny Austin. Kimball and inmon dw models. https://bennyaustin.com/2010/05/02/kimball-and-inmon-dw-models. [Acedido em 11-12-2024].

[5] Sergi Batalla. From data to value: turn unstructured data into a dimensional data model using Data Warehouse (SAP BW/4HANA) and Business Intelligence (SAP Lumira designer) visualizations. Em *Proceedings of the 2024 10th International Conference on Computer Technology Applications*, ICCTA '24, páginas 103–108, New York, NY, USA, agosto 2024. Association for Computing Machinery.

[6] S. Bouaziz, A. Nabli, e F. Gargouri. Design a data warehouse schema from document-oriented database. *Procedia Computer Science*, 159:221–230, 2019.

[7] Farrah Farooq e Syed Mansoor Sarwar. Real-time data warehousing for business intelligence. Em *Proceedings of the 8th International Conference on Frontiers of Information Technology*, FIT '10, páginas 1–7, New York, NY, USA, dezembro 2010. Association for Computing Machinery.

[8] Célia Talma Gonçalves, Maria José Angélico Gonçalves, e Maria Inês Campante. Developing Integrated Performance Dashboards Visualisations Using Power BI as a Platform. *Information*, 14(11):614, novembro 2023. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[9] Teresa Guarda, Ana Carvaca, Ronald Gozabay, Mitzi Saquicela, e Helen Tomalá. Business Intelligence Analytics Tools. Em Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, e Chiara Garau, organizadores, *Computational Science and Its Applications – ICCSA 2022 Workshops*, páginas 352–362, Cham, 2022. Springer International Publishing.

[10] IBM. What is olap (online analytical processing)? https://www.ibm.com/topics/olap. [Acedido em 05-12-2024].

[11] V.S. Khatuwal e D. Puri. Business Intelligence Tools for Dashboard Development. Em *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, páginas 128–131, 2022.

[12] M.R. Llave. Data lakes in business intelligence: Reporting from the trenches. *Procedia Computer Science*, 138:516–524, 2018.

[13] Mehrdad Maghsoudi e Navid Nezafati. Navigating the acceptance of implementing business intelligence in organizations: A system dynamics approach. *Telematics and Informatics Reports*, 11:100070, setembro 2023.

[14] Megan Menth. Tableau Inventory Dashboard Example — phdata.io. https://www.phdata.io/blog/tableau-inventory-dashboard-example.

[15] Microsoft. Descrição geral do processamento analítico on-line (olap). https://support.microsoft.com/pt-pt/office/descrição-geral-do-processamento-analítico-online-olap-15d2cdde-f70b-4277-b009-ed732b75fdd6bmwhat_is_on_line_analytical_processing. [Acedido em 05-12-2024].

[16] Microsoft. O que é o Microsoft Fabric - Microsoft Fabric — learn.microsoft.com. https://learn.microsoft.com/pt-pt/fabric/get-started/microsoft-fabric-overview. [Acedido em 07-12-2024].

[17] Microsoft. Understand star schema and the importance for Power BI - Power BI — learn.microsoft.com. https://learn.microsoft.com/en-us/power-bi/guidance/star-schema. [Acedido em 05-12-2024].

[18] Oracle. Oracle named a leader in the 2024 gartner® magic quadrant™ for analytics and business intelligence platforms. https://www.oracle.com/pt/news/announcement/oracle-named-leader-in-2024-gartner-magic-quadrant-for-analytics-and-business-intelligence-platforms-2024-06-24/, 2024. [Acedido em 02-12-2024].

[19] Qlik. 2024 Gartner Magic Quadrant for BI and Analytics | Qlik — qlik.com. https://www.qlik.com/us/gartner-magic-quadrant-business-intelligence. [Acedido em 08-12-2024].

[20] Qlik. Qlik Sense | Modern Cloud Analytics — qlik.com. https://www.qlik.com/pt-br/products/qlik-sense. [Acedido em 08-12-2024].

[21] Qlik. The first sheet: Dashboard | Qlik Cloud Help. https://help.qlik.com/en-US/cloud-services/Subsystems/Hub/Content/Sense_Hub/Sheets/first-sheet-dashboard-cloud.htm. [Acedido em 12-12-2024].

[22] Dhyanendra Singh Rathore. Demystifying Power BI - Understanding Power BI Ecosystem and Landscape — medium.com. https://medium.com/microsoft-power-bi/demystifying-power-bi-understanding-power-bi-ecosystem-and-landscape-d61614458017. [Acedido em 07-12-2024].

[23] D. Ruqti, M. Alkhammash, e D. Al-Eisawi. Visual Data Analytics Dashboard for Insightful Traffic Monitoring: The Case of Jordan's Transportation Sector. *Lecture Notes in Networks and Systems*, 1082 LNNS:79–89, 2024.

[24] Jun Shan. Medallion Architecture: What, Why and How — junshan0. ttps://medium.com/@junshan0/medallion-architecture-what-why-and-how-ce07421ef06f. [Acedido em 04-04-2025].

[25] Bharat Singhal e Alok Aggarwal. ETL, ELT and Reverse ETL: A business case Study. Em *2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, páginas 1–4, dezembro 2022.

[26] Gautam Srivastava, Muneeswari S, Revathi Venkataraman, Kavitha V, e Parthiban N. A review of the state of the art in business intelligence software. *Enterprise Information Systems*, 16(1):1–28, janeiro 2022. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/17517575.2021.1872107.

[27] TheKnowledgeAcademy. Qlikview vs Qlik Sense: A Detailed Comparsion — theknowledgeacademy.com. https://www.theknowledgeacademy.com/blog/qlikview-vs-qlik-sense. [Acedido em 12-12-2024].

[28] Lamia Yessad e Aissa Labiod. Comparative study of data warehouses modeling approaches: Inmon, kimball and data vault. Em *2016 International Conference on System Reliability and Science (ICSRS)*, páginas 95–99, 2016.

[29] Wang Zhijuan, Wei Hongchang, e Wu Xuefang. A Data Warehouse Design Method. Em *2012 International Conference on Computer Science and Service System*, páginas 2063–2066, agosto 2012.