INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Criação de um processo de Business Intelligence numa empresa de comércio eletrónico

JOÃO CARLOS CORREIA DE OLIVEIRA
Outubro de 2021

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Criação de um processo de *Business Intelligence* numa empresa de comércio eletrónico

João Carlos Correia de Oliveira

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas de Informação de Conhecimento

Orientador: Paulo Oliveira

Júri:

Presidente:

Vogais:

Porto, outubro de 2020

Resumo

Uma empresa que fornece um serviço de sugestões de conjuntos de produtos e dos respetivos acessórios a outras empresas de comércio eletrónico, revelou um potencial crescimento nos últimos dois/três anos. Consequentemente a este crescimento, e com a grande quantidade de dados que começaram a ser produzidos, surgiu a necessidade de se obter indicadores e métricas dos mesmos. Com a existência destes indicadores será possível fazer análises transversais às vendas dos clientes, o que facilita aos gestores da empresa tomar decisões imediatas e estratégicas.

Desenvolveu-se um sistema de informação de Business Intelligence de forma ir ao encontro destas necessidades e, naturalmente, centralizar a informação alusiva às vendas. Desta forma, o sistema encontra-se preparado e datado de potencialidades que permitem solucionar determinadas questões que possam surgir no futuro, sem que tenham que ser feitas grandes alterações à estrutura do mesmo, como por exemplo o surgimento de novas fontes de dados. Tendo em vista o fácil acompanhamento dos indicadores implementados por parte dos gestores foi disponibilizado um conjunto de relatórios, nos quais a informação pode ser filtrada e se encontra sob a forma de dashboards.

Por fim, como forma de validar o sistema desenvolvido, foi elaborado inquéritos de satisfação a profissionais da empresa, onde se avalia a usabilidade e a utilidade do mesmo. Os inquiridos demonstraram opiniões bastante positivas às afirmações que lhes foram colocadas acerca da aplicação desenvolvida. De acordo com os resultados obtidos, o desenvolvimento da aplicação atingiu os objetivos definidos e mostrou utilidade para os utilizadores finais.

Palavras-chave: Business Intelligence, indicadores, relatórios, dashboards, vendas, comércio eletrónico, Big Query, Google Cloud Storage

Abstract

A company, that offers a service of bundles of products and their accessories to other e-commerce companies, had a great growth in recent years. With this growth and with the large amount of data produced, the need arose to get indicators and metrics from them. Having these indicators will make it possible to carry out cross-sectional analyzes of customer sales, which will make it easier for the company's managers to make strategic decisions.

A Business Intelligence information system was developed to solve these needs and where information about sales is centralized. In this way, the system is prepared for future needs, without having to make major changes to its structure. For managers, to monitor the implemented indicators, a set of reports were available, in which the information can be filtered and found in the dashboards.

Finally, as a way of validating the developed system, satisfaction surveys were drawn up among the company's professionals, which assess its usability and usefulness. Respondents showed very positive evaluations of the statements that were made about the developed application. According to the results obtained, the development of the application was achieved according to the objectives to be proposed and it shown utility for the end user.

Keywords: Business Intelligence, indicators, reports, dashboard, sales, e-commerce, Big Query, Google Cloud Storage

Agradecimentos

A escrita desta dissertação não teria sido possível sem o apoio da minha namorada, sendo ela a principal responsável por eu não desistir.

Aos meus pais e irmã por me acompanharem em todo o meu percurso académico.

Agraço ainda ao meu Orientador, professor Paulo Oliveira, pelo apoio, disponibilidade, pelos ensinamentos e pelos conselhos ao longo desta dissertação.

E por último, agradeço também aos meus colegas de mestrado com os quais tive oportunidade de trabalhar e colegas de trabalho pela partilha de conhecimento constante.

Índice

Índice de Figuras

Índice de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

BI	*Business Intelligence*
AD	Armazém de dados
ETL	*Extraction, Transformation and Load*
SCD	*Slowly Changing Dimensions*
SAD	Sistemas de Apoio à Dimensão
DWBA	*Data Warehouse Bus Architecture*
CIF	*Corporate Information Factory*
OLAP	*Online Analytical Processing*
ROLAP	*Relationanal Online Analytical Processing*
MOLAP	Multidimensional *Online Analytical Processing*
HOLAP	*Hybrid Online Analytical Processing*
OLTP	*Online Transactional Processing*
SGBD	Sistema de Gestão de Base de Dados
GCP	*Google Cloud Platform*
FFE	*Fuzzy Front-End*
SWOT	*Strenghts, Weaknesses, Opportunities, Threats*

1 Introdução

Esta dissertação foi desenvolvida no âmbito do mestrado em Engenharia Informática, do ramo Sistemas de Informação e Conhecimento. Encontra-se dividida em cinco capítulos: Introdução; Estado da Arte; Análise de valor; Design da Solução; Avaliação da Solução.

Relativamente à Introdução, o primeiro capítulo deste documento, será exposto o enquadramento (1.1), que justifica a opção da investigação tomada, será lançado o problema (1.2) e os objetivos (1.3) que se pretendem alcançar no final do projeto.

No segundo capítulo, o Estado da Arte, pode-se verificar seis subcapítulos, onde serão expostas as várias soluções e ferramentas existentes que podem ser aplicadas no problema levantado. São eles: Tomada de Decisão (2.1); Business Intelligence (2.2); Armazém de Dados (2.3); *Online Analytical Processing*, OLAP (2.4); Ferramentas (2.5); Conclusões (2.6).

De seguida, expõe-se o terceiro capítulo, Análise de Valor, que se resume às oportunidades identificadas e analisadas que acrescentarão valor à empresa. O mesmo subdivide-se em: Processo Fuzzy Front-End (3.1), Modelo de CANVAS (3.2) e Cadeia de valor de Porter (3.3).

Posto isto, apresenta-se o quarto capítulo, Análise e Design, em que se definem os requisitos e desenha-se uma proposta de solução. O mesmo é composto por: Identificação de Atores (4.1); Requisitos (4.2); Arquitetura Proposta (4.3).

Posteriormente, expõe-se o quinto capítulo, Implementação da Solução, em que se apresenta como foi implementado o sistema. Este é composto por: Extração (5.1); Transformação (5.2); Carregamento (5.3); Relatórios (5.4); Testes (5.5); Implantação da Solução (5.6).

No sexto capítulo é apresentada a Avaliação da Solução, onde é apresentada a Metodologia da Avaliação (6.1) e é feita uma Análise do Resultado dos Inquéritos (6.2)

Por fim, no sétimo capítulo, Conclusão, são apresentados os Objetivos Alcançados (7.1) os Problemas/Limitações (7.2) e o Trabalho Futuro (7.3).

1.1 Enquadramento

Para clarificar o desenvolvimento deste projeto, é necessário expor o contexto onde será aplicada esta investigação, para se perceber a pertinência e urgência da procura da solução para esta problemática. Assim sendo, para o desenvolvimento deste documento, pretende-se recorrer a dados reais de uma empresa britânica, atualmente sediada em Portugal. Estes dados que serão recolhidos ao longo da investigação, servirão como base de análise e de reflexão, de modo a construir um processo que permita melhorar a produtividade e o lucro desta empresa. Desta forma, para que isto seja possível, será necessário melhorar a tomada de decisão sendo um dos aspetos centrais e mais preocupantes para este aperfeiçoamento.

A mesma terá sido fundada em 2012 e revela um modelo de negócio que se foca, essencialmente*, Business to Business*, sendo que apresenta um leque de clientes dispersos por mais de trinta países. Revela-se fundamental evidenciar que a sua missão é ajudar os clientes a aumentarem o lucro financeiro, fornecendo-lhes serviços variados.

Grande parte do lucro de diversas empresas de comércio eletrónico é obtido através da venda de acessórios ou de serviços para produtos. Desta forma, a empresa criou um sistema para recomendar acessórios de uma forma relevante, para que o utilizador da sua plataforma realize as suas compras e acrescente os mesmos, antes de finalizar a sua compra, recomendado pela plataforma. Isto permite um aumento da margem de lucro dos vendedores. Na Figura 1, consegue-se observar um exemplo de um conjunto de produto mais o seu acessório que é recomendado pela empresa

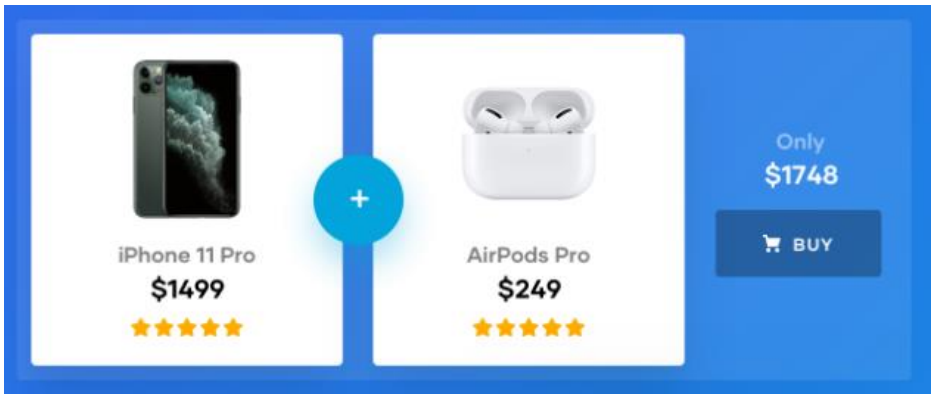

Figura 1 - Exemplo de conjunto de produto mais acessório

No ano de 2019 foram feitas mais de três mil milhões de recomendações e em 2020 o número de clientes tem aumentado. Importa, também, salientar que os líderes em vendas a nível mundial de artigos de beleza, desporto e eletrónica na Europa, pertencem ao leque de clientes da empresa.

Com este crescente de volume de transações, torna-se essencial a criação de uma base que permita armazenar todas as informações relativas às vendas, onde seja, também, possível analisar as mesmas. Para isso, e com o exponencial aumento das vendas, tal como foi referido,

é necessário agregar esta informação num único local, mantendo o seu histórico e criando um armazém de dados.

1.2 Problema

Sempre que há uma interação no site dos clientes desta empresa, esta recebe a informação alusiva a essa interação. No entanto, para a empresa conseguir cobrar uma taxa sobre a venda dos produtos, terá de ocorrer uma das seguintes interações:

- no ato da compra de artigos, deverá estar incluído pelo menos um produto recomendado pela empresa;
- no caso de o utilizador comprar apenas um produto, o mesmo deverá de ter sido sugerido, previamente, pela empresa.

Todos os meses, para a empresa obter o relatório de vendas do mês transato tem de executar diversos processos. Estes são intrínsecos a cada cliente e agregam todas a interações dos seus utilizadores para determinar quais são as vendas que irão ser taxadas. A grande limitação destes processos é estarem sediados numa função *Lambda* que se a execução do processo não terminar em quinze minutos é cancelado (Amazon, 2020).

Com o aumento do número de clientes e estes a aumentarem as suas vendas a geração de relatórios torna-se cada vez mais lenta, fazendo com que o processo comece a tornar-se falível, tendo de se gerar novos relatórios, e os mesmos demoram o seu tempo, tempo este dispensável e inútil. Para além disso, estes relatórios sobrecarregam as bases de dados que se querem operacionais.

Mais ainda, com o aumento significativo do número de clientes e a quantidade de vendas de cada cliente, o processo não está preparado para ser escalável, ou seja, está a atingir os limites de capacidade de processamento, o que poderá causar falhas nos dados dos relatórios. Embora, se vá conseguindo gerar os relatórios de vendas dos clientes do mês passado, já não é possível obter os relatórios de meses anteriores, o que é uma grande limitação deste processo.

Os relatórios são gerados para um documento EXCEL, que por si só tem uma limitação de 1,048,576 linhas, segundo a Microsoft (Microsoft, 2021b).

Este relatório contém dados e um conjunto de gráficos que são analisados de forma não fundamentada, sendo um processo praticamente manual e maioritariamente baseado na intuição e no conhecimento da pessoa encarregue. Torna este processo vulnerável a falhas humanas em praticamente todas as etapas.

Visto que as base de dados operacionais são otimizadas para dar suporte às operações do dia-a-dia, quando se tenta executar *queries* complexas a performance torna-se inaceitável, segundo Chaudhuri e Dayal (Chaudhuri e Dayal, 1997). Além disso, também referem que os sistemas de apoio à decisão necessitam de dados que possa faltar em base de dados operacionais, como

por exemplo para se perceber tendências ou realizar previsões são necessários dados históricos, enquanto as base de dados operacionais só armazenam, tipicamente, os dados atuais (Chaudhuri e Dayal, 1997).

Uma vez que não existe um processo estruturado de análise de dados e o processo de geração de relatórios está cada vez mais falível, o objetivo desta dissertação é apresentar as vantagens de criar um Sistema de Apoio à Decisão (SAD) e estruturar o processo de análise de dados de forma que se consiga fazer uma análise mais objetiva quais serão os melhores futuros clientes, quais são as categorias com melhor desempenho e quais são os melhores possíveis descontos. Facilitando, assim, as decisões táticas e estratégicas.

1.3 Objetivos

Tal como foi mencionado na Secção 1.2, a empresa tem uma limitação no que toca à análise de dados. Tendo em consideração este aspeto e servindo o mesmo como ponto de partida para esta dissertação, torna-se essencial referir o objetivo central que dará contorno a toda a investigação. Assim sendo, traçou-se o objetivo de tornar o processo de análise de dados simples e rápido.

No sentido de:

- Otimizar o processamento da análise de dados
- Precisar as métricas dos conjuntos de dados obtidos
- Tornar eficiente a obtenção de dados para os relatórios
- Consultar dados históricos de forma rápida
- Tornar os dados coerentes
- Reduzir os custos inerentes à análise de dados

Posto isto, é essencial traçar alguns objetivos específicos que vão ao encontro do referido anteriormente. Assim sendo, os mesmos definem-se em:

Otimização de recursos: como observado na Secção 1.2, a empresa vê-se perante um problema de ter os recursos limitado na geração de relatórios. Com uma solução de BI, este processo irá ser facilitado e possibilitará a consulta de informação de uma forma mais simples e versátil.

Poupança nos custos: Como consequência da otimização de recursos, este trará poupanças para a empresa. Também a empresa poupará dinheiro com os recursos humanos visto que estes não terão de perder mais tempo com a criação dos relatórios. Por último, uma solução de BI mostrará uma visão mais detalhada onde se poderá analisar os custos podendo otimizá-los.

Única versão da verdade: tendo diversas bases de dados operacionais, torna-se benéfico ter um sistema único como fonte de verdade para todos os departamentos terem a mesma visão. Tendo múltiplos Excels estão sempre sujeitos a modificações.

Único responsável pela informação: Num projeto de BI é definida uma pessoa ou um departamento que será o único ponto de contacto para se obter algum tipo de informação, deixando assim de se ficar dependente de diferentes fontes.

Análises sob demanda: Com a solução de BI qualquer utilizador poderá gerar os relatórios, sendo que um dos pré-requisito será a informação estar disponível sempre que necessária.

2 Estado de Arte

Neste capítulo abordar-se-á os diferentes tipos de arquiteturas e tecnologias para a construção de um armazém de dados. Esta abordagem será feita através de uma comparação constante entre arquiteturas possíveis para a criação do armazém de dados e as ferramentas que serão necessárias para auxiliar esse processo. No final, perceber-se-á qual as opções que estarão mais enquadradas e próximas às necessidades da empresa para resolver este problema.

2.1 Tomada de Decisão

Quando se fala em tomada de decisão, entende-se que é feita uma escolha num leque variado de alternativas, considerando que uma dessas alternativas é possível que seja não ter qualquer tipo de ação (Nutt, 2002).

Para que se tome uma decisão, seja ela de que natureza for, seja de cariz individual ou organizacional, é necessário que haja uma seleção das informações. Qualquer decisão que seja tomada terá efeito e consequências na vida dos intervenientes e poderá também alterar o rumo de uma organização. Neste sentido, torna-se essencial aumentar a eficácia na tomada de decisões, uma vez que traz implicações no futuro das empresas e permite maximizar a produtividade do trabalho (Ireland & Miller, 2004).

Em qualquer organização são tomadas várias decisões, que são feitas nos diversos setores das empresas. Algumas são mais frequentes e de rotina, são chamadas de decisões **estruturadas**. Tendo em vista isto, independentemente do impacto que as decisões, por mais pequenas que sejam, acrescentem à empresa, na sua tentativa de melhoria, é importante pensar a longo prazo e perceber que é o seu somatório final que poderá trazer um impacto positivo a qualquer organização.

Por outro lado, existem decisões únicas e importantes que exigem recolha de informação, reflexão e ponderação das diferentes alternativas. Estas exigem mais tempo e são consideradas decisões **não estruturadas**.

As decisões podem ser classificadas em três diferentes categorias baseadas no nível onde ocorrem:

Decisões estratégicas que afetam o rumo de uma organização por um grande período, como por exemplo, a decisão de se juntar a outra organização ou desenvolver um novo produto.

Decisões táticas afetam apenas parte da organização, habitualmente um único departamento, têm um intervalo de tempo limitado, tipicamente um ano, por exemplo, definir o plano de produção para o próximo mês.

Decisões operacionais são específicas de cada atividade dentro da organização, tendo pouco impacto no futuro, por exemplo, a gestão de encomendas.

Para facilitar a compreensão do que foi mencionado, pode-se observar a Figura 2, onde está sintetizado os vários tipos de decisões.

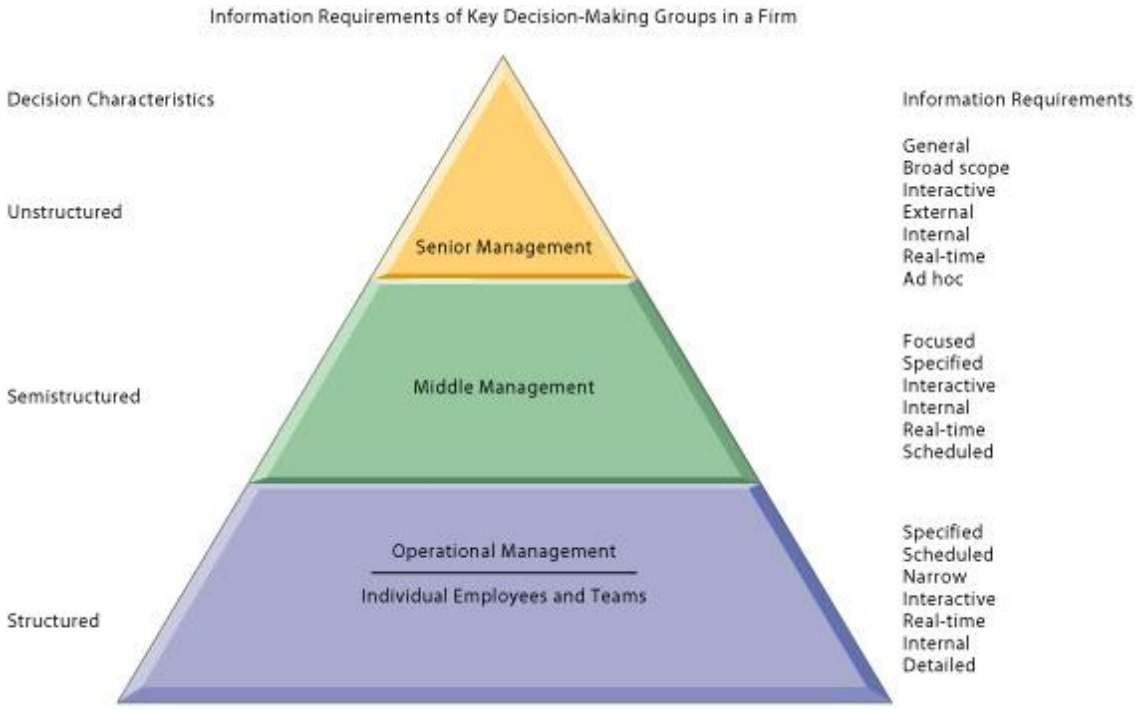

Figura 2 - Grupos de tomada de Decisão (FEUP, 2021)

Para auxiliar os diferentes tipos e níveis de decisão, existem quatro diferentes géneros de sistemas: Management Information systems, decision-support systems; executive support systems ; group decision-support systems (FEUP, 2021). Assim sendo, importa clarificar que são estes a base para a criação de um processo de BI.

2.2 Business Intelligence

O termo *Business Intelligence* (BI) já tem mais de 150 anos e foi introduzido por Richard Millar Devens *em* Cyclopædia of Commercial and Business Anecdotes.

Atualmente, existem várias definições para o termo BI. Nogués e Valladares (Nogués e Valladares, 2017) definem BI como sendo um conjunto de ferramentas e processos que ajuda a tomar decisões baseadas em dados precisos, poupando tempo e esforço. O site *tableu.com* (Tableau, 2020) afirma que BI combina análise de negócio, exploração de dados, visualização de dados, ferramentas de dados, infraestrutura e as melhores práticas que ajuda organizações a tomar decisões baseadas em dados. O site *oracle.com* (Oracle, 2020) refere que um solução de BI é uma combinação de estratégia e tecnologia para recolher, analisar e interpretar dados de fontes internas ou externas, com a finalidades de facultar informação sobre o passado, presente e futuro do tema examinado.

Em todas as definições referem que BI auxiliam na tomada de decisão através do processamento de dados, criando informação útil para o utilizador. Tal como foi referido no capítulo da tomada de decisão, o sistema de apoio à decisão (SAD) serve como base para o processo de BI, sendo que este exige um armazém de dados onde ficará guardada toda a sua informação.

2.3 Armazém de dados

Neste subcapítulo serão apresentadas definições, nomeadamente, o que é um Armazém de Dados, a explicação de um processo ETL, definição de tabelas de Facto e de Dimensão, do Modelo Dimensional e OLAP. Para além disso, serão apresentadas as diferentes arquiteturas para a construção de um Armazém de Dados.

2.3.1 Definição

Armazém de Dados (AD), também conhecido por *Data Warehouse,* é a base de todo o processo de BI pois este é um repositório central que agrega todos os dados históricos e atuais dos diferentes sistemas de uma organização que já foram previamente tratados. Segundo Inmon (Inmon, 2005) o AD "é orientado ao tema, não variável no tempo e uma coleção de dados não volátil que apoia o processo de tomada de decisão da gestão". Para a criação de um AD, os dados terão que atravessar por um processo de ETL, de forma a serem carregados no mesmo. Este é definido por tabelas de Facto, tabelas de Dimensão, tendo diferentes tipos de abordagem e de modelação dos dados. Para se manter o histórico dos dados, utiliza-se a estratégia Slow Changing Dimensions (Datawarehouse4u, 2021).

2.3.2 Processo ETL

Para a criação de um AD é necessário que os dados sejam tratados antes de serem carregados para o AD. Este tratamento de dados intitula-se processo ETL (extraction, transformation and load), que é uma *pipeline* de dados usada para recolher dados de diversas fontes, transformá-los de acordo com as regras de negócio e carregá-los na fonte de destino, como é representado na Figura 3. Este processo ocorre num motor especializado e envolve a utilização de base de dados temporária, onde irão ficar retidos os dados até serem transformados e por fim serem carregados para o AD. A parte da transformação engloba vários processos com os dados, tais como: agregação, limpeza, junção, validação e remoção de dados duplicados (ZoinerTejada, 2021).

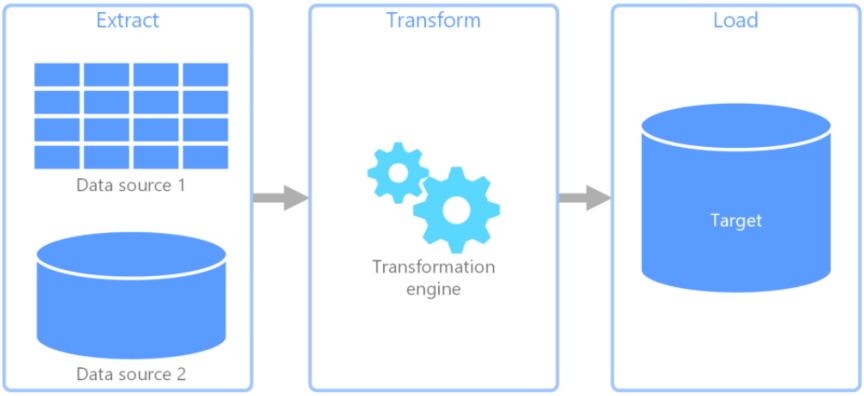

Figura 3 - Processo ETL (ZoinerTejada, 2021)

2.3.3 Tabelas de Facto e de Dimensão

As tabelas de factos contêm os dados que são a base da análise, por norma estes dados são numéricos de forma a poderem ser agregados e mensuráveis. Com a união de factos e com uma fórmula para resumir os dados, consegue-se retirar métricas, indicadores e/ou KPIs importantes para o negócio. Estes dados são também conhecidos por dados mensuráveis. Alguns exemplos de dados nas tabelas de factos são os valores das vendas, o inventário, o tempo gasto e os ganhos.

As tabelas de dimensões contêm dados descritivos, uma vez que são usados para descrever a forma como os dados irão ser agregados, e, também, a quantidade de dados a serem analisados. Os dados das tabelas de dimensões são, por exemplo: a data, o país, o produto ou qualquer campo que permita filtrar ou agregar a informação.

2.3.4 Data Mart

De forma a padronizar a análise de dados e permitir que a utilização de padrões seja simplificada, os AD, normalmente, são organizados em pequenas unidades orientadas para o problema, chamadas *Data Mart,* sendo este subconjunto de tabelas persistidas no mesmo AD. Cada um é dedicado a um problema específico, ajudando na modelação de um AD (Bonifati et al., 2001).

2.3.5 Modelo Dimensional

Na construção de um AD é necessário criar um modelo lógico que irá estabelecer a sua estrutura. Assim sendo, terá de se decidir quais são as tabelas que irão ser criadas e a sua relação, definindo, então, se a informação deve ficar normalizada ou desnormalizada. Segundo Nogués e Valladares (Nogués & Valladares, 2017) dever-se-á tentar encontrar uma solução que seja uma simbiose das duas soluções, não criando um modelo lógico, otimizando assim ao máximo as *queries.* Para a construção desse modelo lógico existente dois modelos que são os mais utilizados para a modelação dos AD, o esquema em estrela e o floco de neve.

2.3.5.1 Esquema em Estrela

O esquema em estrela é um tipo de modelo relacional altamente utilizado nos ADs, especialmente para *Data Marts* pequenos. Este esquema utiliza tabelas de Facto com a informação do negócio e tabelas de Dimensão que ficam relacionadas com as tabelas de Facto através de um campo chave (Nogués & Valladares, 2017). Na Figura 4 é apresentado um exemplo do esquema em estrela, onde se pode verificar esta relação entre a tabela de Facto e as diferentes tabelas de Dimensão.

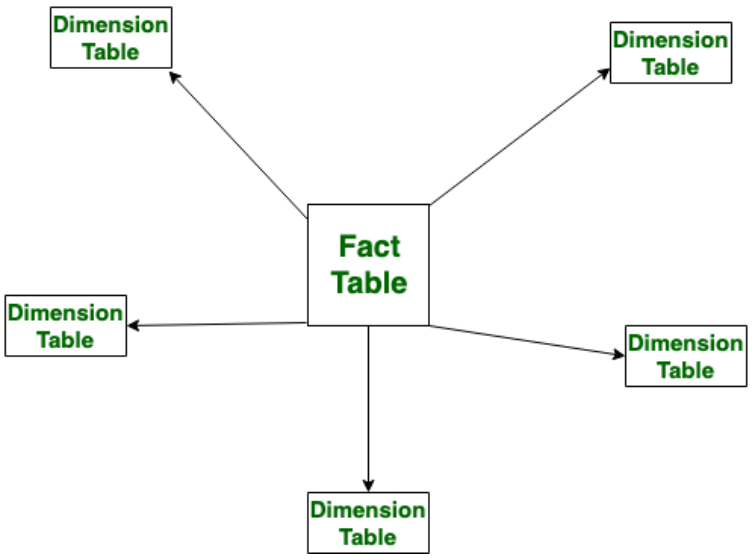

Figura 4 - Esquema em Estrela (GeeksforGeeks, 2019)

2.3.5.2 Esquema em Floco de Neve

O esquema floco de neve representa um modelo dimensional que é composto por uma tabela de Facto central e um conjunto de tabelas de Dimensão que podem ser repartidas em tabelas subdimensionais, como se pode verificar na Figura 5.

De acordo com Levene e Loizou (Levene e Loizou, 2003), os esquemas de estrela podem ser redefinidos em esquemas de floco de neve para oferecer suporte a atributos hierárquicos.

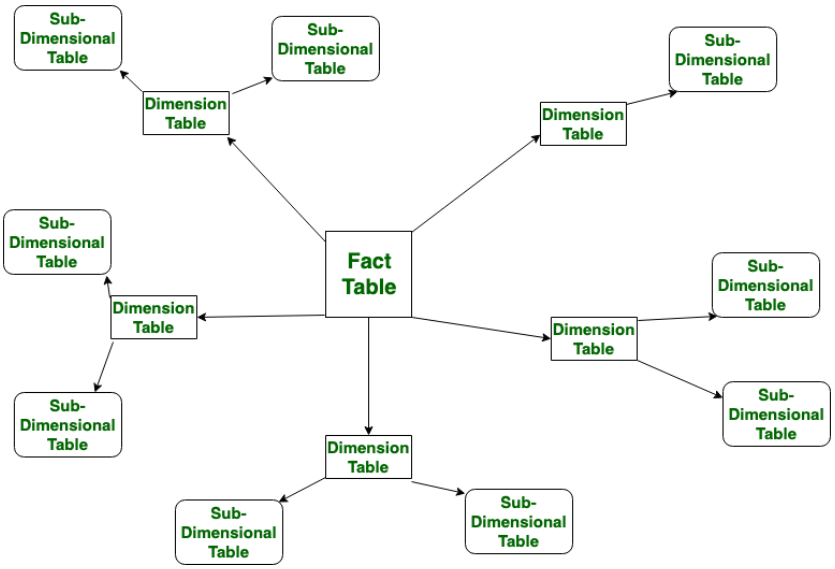

Figura 5 - Esquema Floco de Neve (GeeksforGeeks, 2019)

2.3.6 Slowly Changing Dimensions

Para se conseguir manter o histórico dos dados, por norma, utiliza-se uma estratégia de *Slowly Changing Dimensions (*SCD). Esta permite armazenar e gerir os dados históricos ao longo do tempo, sendo uma das tarefas mais críticas do processo ETL, uma vez que irá fazer com que os dados mantenham o seu histórico. Por sua vez, este é um dos objetivos principais do AD. Existem seis tipos de SCD:

- **Tipo 0 – Dimensão fixa:** Não são permitidas mudanças nas dimensões.
- **Tipo 1 – Sobrescrever:** Os dados novos sobrescrevem os dados já existentes perdendo assim os dados previamente armazenados, como se pode verificar na Figura 6.

CustD	CustName	CustType
1	Cust_1	Corporate

CustD	CustName	CustType
1	Cust_1	Retail

Figura 6 - SCD Tipo 1

- **Tipo 2 – Inserir novo registo/linha:** Nesta abordagem todas as modificações históricas são armazenadas no AD, adicionando uma nova entrada na base de dados com os dados atuais e atualizando a data final e o atributo ativo da última entrada desse registo, apresentada na Figura 7.

CustD	CustName	CustType	StarDate	EndDate	CurrFlag
1	Cust_1	Corporate	22-05-2018	31-12-9999	Y

CustD	CustName	CustType	StarDate	EndDate	CurrFlag
1	Cust_1	Corporate	22-05-2018	24-08-2018	N
2	Cust_1	Retail	25-08-2018	31-12-9999	Y

Figura 7 - SCD Tipo 2

- **Tipo 3 – Coluna do valor anterior:** Nesta abordagem só se mantem os dados históricos do registo anterior, tendo tipicamente duas colunas Anterior/Corrente, como é visível na Figura 8.

CustD	CustName	CurrentType	PreviousType
1	Cust_1	Corporate	Corporate

CustD	CustName	CurrentType	PreviousType
1	Cust_1	Retail	Corporate

Figura 8 - SCD Tipo 3

- **Tipo 4 – Tabela de dados históricos:** Mantém-se em separado uma tabela com todos os dados históricos. A tabela principal só irá guardar a informação mais recente. Pode-se observar na Figura 9 um exemplo da SCD tipo 4.

CustD	CustName	CustType
1	Cust_1	Corporate

CustD	CustName	CustType	StartDate	EndDate
1	Cust_1	Retail	01-01-2018	21-07-2018
1	Cust_1	Other	22-07-2018	17-05-2018
1	Cust_1	Corporate	18-05-2018	31-12-9999

Figura 9 - SCD Tipo 4

- **Tipo 6 – SCD híbrida:** é a combinação das SCDs do tipo 1, 2 ,3 (1+2+3 = 6). Esta mantem o atributo corrente, o atributo histórico, data de início, data de fim e o atributo a identificar se a linha se se encontra ativa, serve a Figura 10 para ilustrar o mencionado.

CustD	CustName	CustType	HistType	StartDate	EndDate	CurrFlag
1	Cust_1	Corporate	Retil	01-01-2018	21-07-2018	N
1	Cust_1	Corporate	Other	22-07-2018	17-05-2018	N
1	Cust_1	Corporate	Corporate	18-05-2018	31-12-9999	Y

Figura 10 - SCD Tipo 6

2.3.7 Arquitetura dos Armazéns de Dados

Na construção de um AD existem duas visões que se sobressaem neste campo, a de Bill Inmon (Inmon, 2005) e a de Ralph Kimball (Kimball, 2008).

A arquitetura Ralph Kimball, também conhecida por *Data Warehouse Bus Architecture* (DWBA), segue uma abordagem ascendente para a construção do AD definindo, em primeira instância, os Data Marts como requisito de negócio. As fontes de dados primárias são avaliadas e a ferramenta ETL extrai os dados, carregando-os numa base de dados relacional temporária de testes (staging area), por último, os dados são carregados num AD desnormalizado, já divididos em Data Marts, tal como se pode observar na Figura 11.

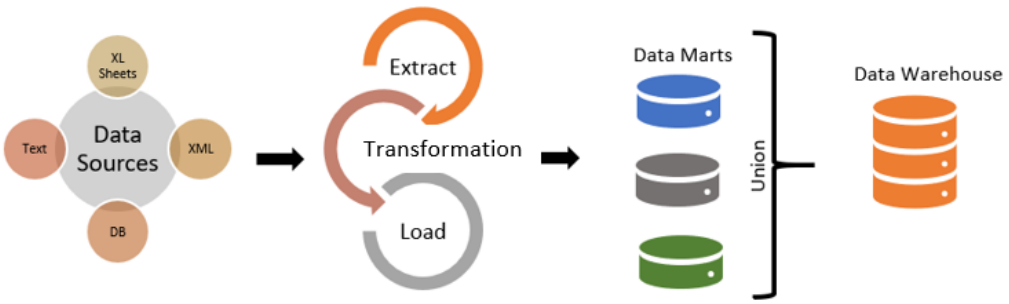

Figura 11 - Arquitetura DWBA (Ekanayake, 2021)

A arquitetura de Inmon, também conhecida por *Corporate Information Factory* (CIF), propõe a construção de um modelo lógico completo para cada entidade primária normalizado, evitando assim a redundância dos dados. Este sugere que os Data Marts sejam criados posteriormente derivado do AD principal, como se consegue perceber na Figura 12.

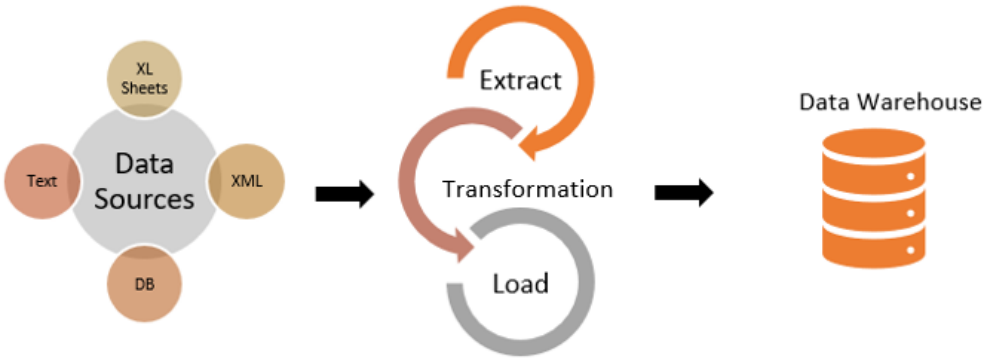

Figura 12 – Arquitetura CIF (Ekanayake, 2021)

Tanto a CIF de Bill Inmon (Inmon, 2005), como a DWBA de Ralph Kimball (Kimball, 2008) têm semelhanças e diferenças (Schouten, 2008). Uma das diferenças que mais se destaca entre as duas abordagens é que na CIF de Inmon sugere a criação num único AD normalizado. Pode ter *Data Marts*, mas estes são derivados do AD principal. A granularidade dos dados guardados deve ser o mais baixa possível para oferecer um maior nível de detalhe a todos os níveis, permitindo, assim, que caso os requisitos do negócio mudem drasticamente, o AD já estará preparado para esta mudança (Inmon, 2005). Por outro lado, Ralph Kimball, com a DWBA, acredita que o AD consiste num conjunto de múltiplos *Data Marts*. Os Data Marts apresentam uma constituição dimensional, com a granularidade que é suficiente e expetável para as necessidades do negócio (Kimball, 2008).

Quanto às semelhanças, as duas concordam com a necessidade de meta dados nas soluções de AD, que são a definição dos conceitos dos relatórios do AD. Inmon e Kimball também concordam que o AD deve ser construído através um processo iterativo (Schouten, 2008).

A arquitetura CIF normalmente é mais indicada para grandes empresas pois, por norma estas empresas tendem e ter bastantes unidades diferentes e dados de várias fontes. Neste caso, tendo dados de diversas fontes o processo de ETL irá ser algo complexo, mas agregará todos os dados num único AD, fazendo, assim, com que todos os colaboradores de uma empresa utilizem as mesmas unidades medida (Schouten, 2008).

A DWBA, em comparação com a arquitetura anterior, será uma solução melhor para empresas mais pequenas. Estas empresas tendem a ter requisitos mensuráveis mais previsíveis que não tendem a alterar com regularidade (Schouten, 2008). Como se pode verificar nesta secção, qualquer uma das duas soluções são opções viáveis, dependendo exclusivamente do contexto onde o AD irá ser construído.

2.4 OLAP

OLAP é o acrónimo para *Online Analytical Processing*, que segundo (IBM, 2021), é uma ferramenta para realizar análises multidimensionais, a velocidades elevadas num grande volume de dados num AD. Esta ferramenta divide os dados em diferentes dimensões, como por exemplo: numa empresa de vendas poderá existir diversas dimensões como localização (região, país, loja), tempo (dia, semana, mês, ano). Nos AD os dados estão organizados por conjuntos em tabelas, e, assim, OLAP extrai esses conjuntos de dados e reorganiza-os num formato multidimensional que permite o processamento muito mais rápido e análises mais perspicazes (IBM, 2021).

2.4.1 Cubo de OLAP

O cubo de OLAP é a base dos sistemas OLAP. É uma base de cariz multidimensional baseada em matrizes que permite processar e analisar várias dimensões de dados muito mais rápida e eficientemente do que uma base de dados relacional. O cubo de OLAP acrescenta camadas adicionais a uma tabela única, cada camada acrescentada inclui dimensões adicionais. Por exemplo, a camada superior do cubo organiza as vendas por região, as camadas adicionais poderiam ser país, cidade ou até por uma loja em específico, dependendo do nível de detalhe pretendido (IBM, 2021). A Figura 13 é uma representação de um cubo de OLAP.

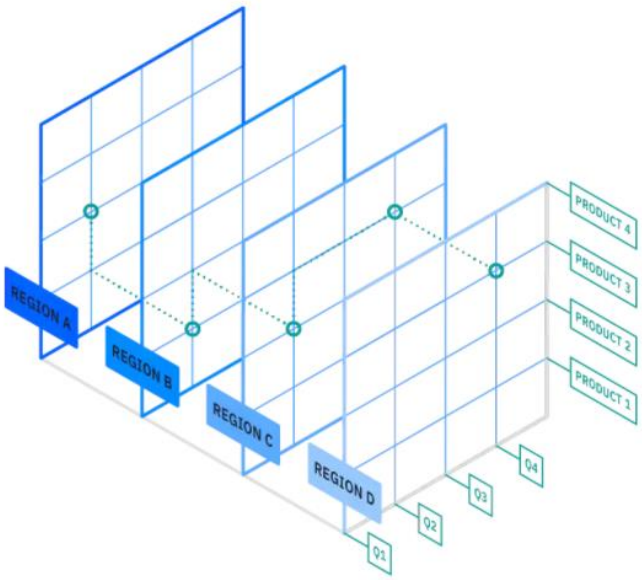

Figura 13 - Cubo de OLAP (IBM, 2021)

Os cubos de OLAP permitem quatro tipos básicos de análise de dados multidimensional:

- ***Drill-down*:** é a operação que converte dados com menos detalhe para dados com mais detalhe. Isto pode ser conseguido através de duas formas: adicionar uma dimensão ao

cubo ou descendo um grau da hierarquia da dimensão (Guru99, 2021), visível na Figura 14 que exemplifica esta operação.

- ***Roll-up*:** é a operação inversa do *drill-down,* ou seja, agrega os dados num cubo OLAP subindo uma hierarquia na dimensão ou reduzindo o número de dimensões, tal como se observa na Figura 15.
- ***Slice and Dice:*** A operação *slice* cria um subcubo escolhendo uma dimensão do cubo de OLAP principal (IBM, 2021), visível na Figura 16.
  A operação *dice* é muito idêntica com a operação *slice*, tendo como diferença a seleção das dimensões para a criação do subcubo, em que nesta operação são selecionadas 2 ou mais dimensões, a Figura 17 exemplifica esta operação.
- ***Pivot*:** A operação *pivot* permite rodar a visão do cubo sobre os seus eixos para apresentar uma nova representação dos dados, permitindo, assim, uma visão dinâmica e multidimensional dos dados, verifique-se a Figura 18 como exemplo.

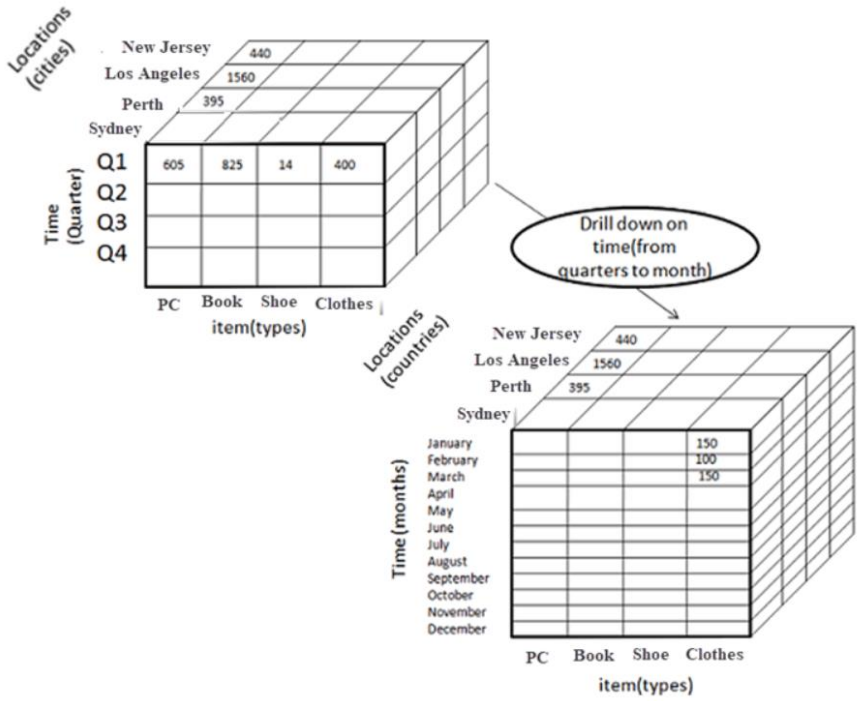

Figura 14 -Operação Drill-Down (Guru99, 2021)

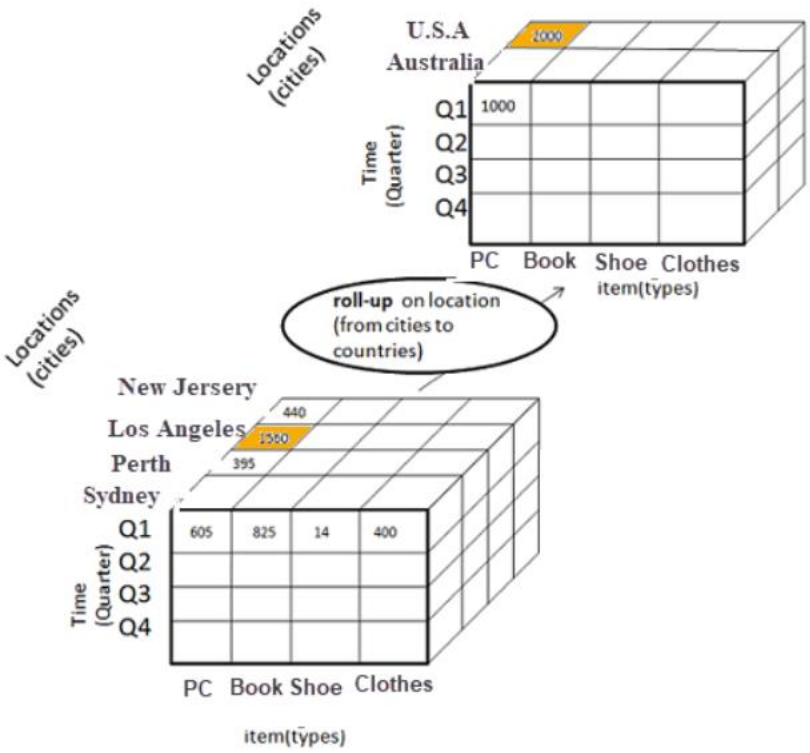

Figura 15 – Operação Roll-up (Guru99, 2021)

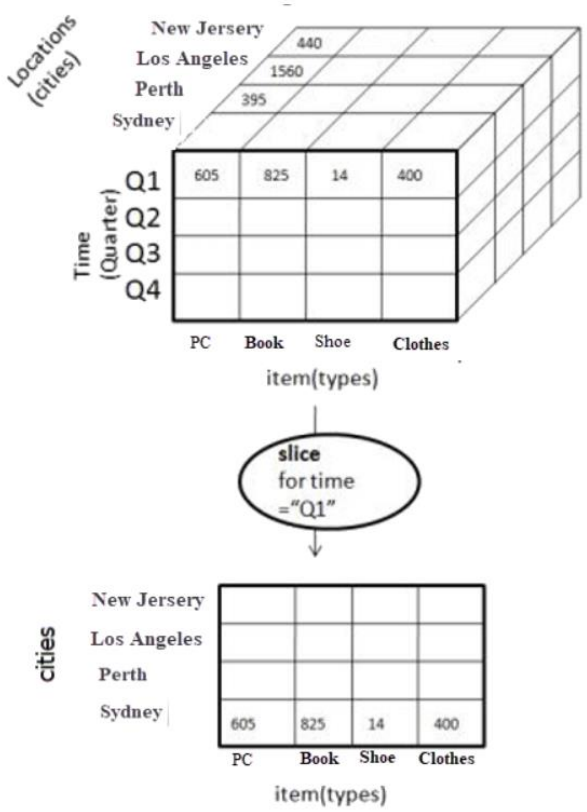

Figura 16 - Operação Slice (Guru99, 2021)

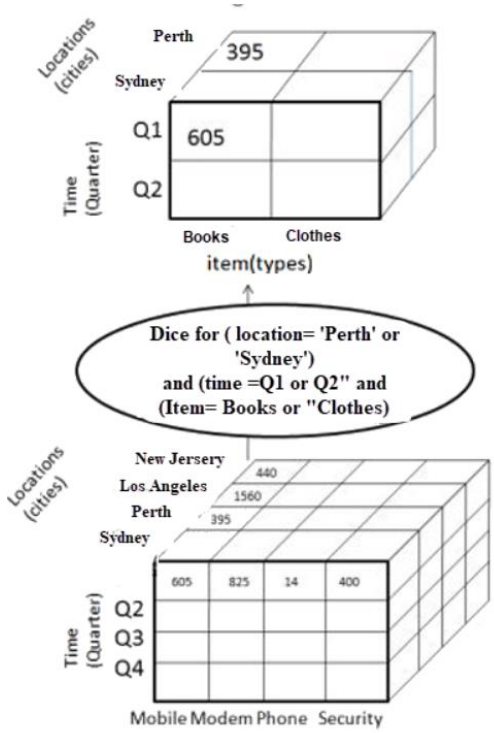

Figura 17 - Operação Dice (Guru99, 2021)

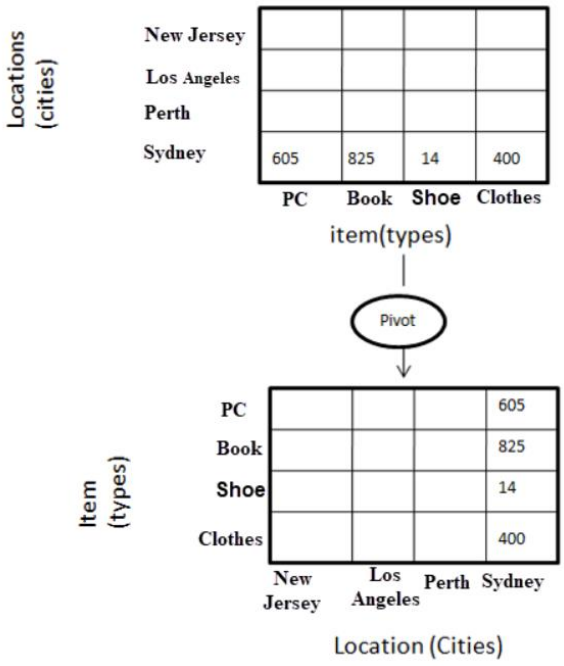

Figura 18 - Operação Pivot (Guru99, 2021)

2.4.2 ROLAP vs MOLAP vs HOLAP

ROLAP é o acrónimo para *Relational Online Analytical Processing,* esta armazena os dados em tabelas ou cubos numa estrutura relacional e devolve a informação sempre que o utilizador executa uma consulta de dados. Esta ferramenta acede aos dados numa base de dados relacional e geram consultas para calcular informações de acordo com o que é solicitado pelo utilizador. O principal problema com as implementações ROLAP tem que ver com o desempenho que é afetado por causa de operações de junção entre grandes tabelas (Key2Market, 2018).

MOLAP é o acrónimo para *Multidimensional Online Analytical Processing*, esta armazena os dados em matrizes multidimensionais otimizados. Esta ferramenta difere significativamente do ROLAP pois requere uma pré-computação dos dados e armazena a informação em cubos. Sempre que necessário, os dados são consultados rapidamente. As desvantagens do MOLAP predem-se com a exigência dos dados pré-computados e das suas limitações em trabalhar com dados demasiado complexos ou de uma elevada cardinalidade (Key2Market, 2018).

HOLAP é o acrónimo para *Hybrid Online Analytical Processing*, e é uma solução híbrida das apresentadas anteriormente, uma vez que permite armazenar parte dos dados em armazenamento MOLAP e outra parte dos dados em armazenamento ROLAP, tentando tirar partido das vantagens de cada uma das soluções. HOLAP exige que a maior quantidade dos dados esteja armazenada numa base de dados relacional, com o intuito de evitar os problemas dos dados desordenados, e um sistema multidimensional que guarda a informação que é requisitada com mais frequência pelos utilizadores. Caso essa informação não seja suficiente para resolver as consultas, o sistema terá que aceder aos dados que se encontram na base de dados relacional (Key2Market, 2018).

2.4.3 OLAP vs OLTP

OLTP é o acrónimo para *Online Transactional Processing* e foca-se, principalmente, na inserção, atualização e eliminação de registos. Consiste em transações curtas que são executadas em pequenos períodos. É um processamento de dados eficaz para executar muitas transações por segundo, ou seja, são ideais para a tarefas do dia a dia (Yıldırım, 2020).

OLAP em contraste com os processamentos de dados OLTP, tem como principal objetivo armazenar dados históricos e processar grandes conjuntos de dados que permitem responder a diversas perguntas sobre os dados, ao longo do tempo. Estas tipo de processamento são ideias para tomar decisões, pelo facto de se poder consultar dados históricos (Yıldırım, 2020).

A Tabela 1 - OLTP vs OLAP resume as principais diferenças entre as duas bases de dados.

Tabela 1 - OLTP vs OLAP (Yıldırım, 2020)

	OLTP	**OLAP**
Função	Operações do Dia-a-dia	Suporte de decisões
Design da base de dados	Orientado a aplicação	Orientado ao tema
Dados	Correntes, detalhes atualizados, isolados	Históricos, resumidos de forma multidimensional, consolidados
Uso	Repetidamente	Sempre que necessário
Unidades de trabalho	Curta, transações simples	Consultas complexas
Tamanha da base de dados	Gigabytes	Terabytes
Métricas	Taxa de transações	Taxa de consultas e repostas

2.5 Ferramentas

No momento de construção de um sistema terá que, a certa altura, escolher qual a melhor ferramenta para construir o sistema. Este é um passo importante uma vez que, no caso de a mesma ser bem escolhida, isto simplificará os processos de construção, trará melhores funcionalidades e um melhor custo. Estas ferramentas no âmbito de BI, permitem recolher e analisar os dados, criando conhecimento nas organizações, sejam elas pequenas, médias ou grandes facilitando o seu crescimento.

2.5.1 Armazenamento de Dados na Cloud

Desta forma, a escolha da ferramenta será auxiliada pelos Quadrantes Mágicos da Gartner.

> "Quadrantes Mágicos da Gartner são o culminar de uma pesquisa num mercado específico, oferecendo uma perspetiva em relação aos competidores do mercado. Ao aplicar um tratamento gráfico e um conjunto uniforme de critérios de avaliação, um Quadrante Mágico ajuda a determinar rapidamente como os fornecedores de tecnologia estão a executar as suas visões declaradas e o seu desempenho em função da visão de mercado da Gartner" (Gartner, 2021).

Um Quadrante Mágico oferece uma representação gráfica entre os concorrentes em que são posicionados em quatro diferentes tipos de fornecedores de tecnologia(Gartner, 2021):

- **Líderes:** executam bem a sua visão atual e estão bem posicionados para o futuro;
- **Visionários:** compreendem a direção do mercado ou têm uma visão que irá mudar as regras de mercado, mas ainda não a puseram em prática;

- **Concorrentes de nicho:** focam-se apenas num pequeno segmento de mercado específico;
- **Desafiadores:** excutam bem a sua visão e podem atá dominar o grande segmento, mas não compreendem a direção do mercado.

Antes de se iniciar a pesquisa sobre qual a ferramenta a utilizar para a construção do AD foi definido que esta devia de ser fácil de se gerir e manter e não devia trazer grandes custos à empresa. Por este motivo irá ser feita uma pesquisa de quais os Sistemas de Gestão de Base de Dados (SGBD) permitem a gestão de um armazém de dados na nuvem.

Gartner (Gartner, 2020), define o mercado de SGBD como sendo o de fornecedores que distribuem serviços em nuvens públicas ou privadas, que gerem dados lá armazenados. Os critérios de avaliação para a criação dos quadrantes mágicos de Gartner são divididos em dois grandes tópicos: capacidade para executar e a abrangência de visão (Gartner, 2020). Na Figura 19 encontra-se representado um quadrante mágico de Gartner aplicado a SGBD na nuvem.

Figura 19 - Quadrante Mágico aplicado a DBMS na nuvem(Gartner, 2020)

2.5.1.1 Google

A Google encontra-se localizada no Quadrante Mágico dos líderes. Apesar da mesma oferecer vários produtos, tais como: o Google Cloud Platform, Google Cloud, Cloud Spanner, Cloud Bigtable, BigQuery; esta dissertação pretende-se focar no seu AD que é o Big Query.

BigQuery é o armazém de dados totalmente gerido pela Google, da escala do petabytes, de baixo custo. Este não tem servidor, ou seja, as empresas não têm de se preocupar com a infraestrutura podendo, assim, focarem-se totalmente na análise de dados (G2, 2021). O BigQuery é uma plataforma de análise de dados utilizada por todo o tipo de empresas, desde as mais pequenas até às maiores, como por exemplo, PayPal, Twiter, UPS, entre outros (Google, 2021b). O BigQuery também disponibiliza uma API para uma melhor gestão de dados. Os seus preços são: 0.02$ por GB por mês de armazenamento, 0,01$ por inserção em *streaming,* $5 por TB nas consultas (Google, 2021e). A Figura 20 serve para ilustrar o que foi mencionado anteriormente.

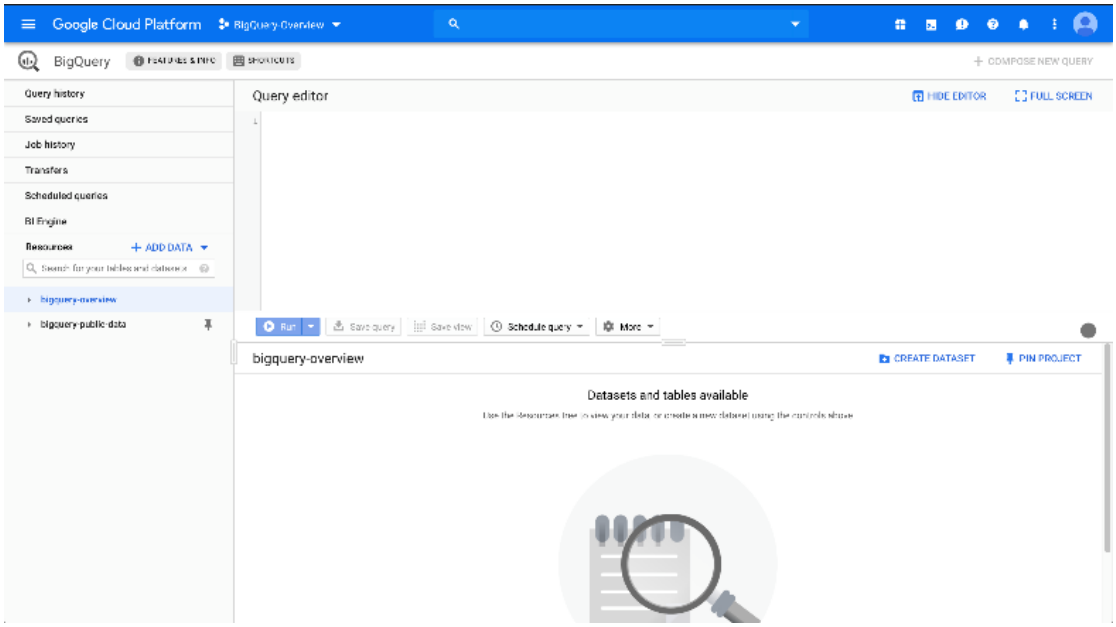

Figura 20 - Google BigQuery (Google, 2021a)

2.5.1.2 Amazon Web Services

De igual forma, salienta-se a Amazon Web Services que se localiza no Quadrante Mágico dos líderes. Independentemente desta empresa assegurar vários produtos, entre os quais: o Amazon *Relational Database Service*; Amazon Aurora e Amazon DynamoDB; nesta dissertação irá ser abordado o seu AD que é o Amazon Redshift.

O Amazon Redshift é um serviço de AD à escala dos petabytes, totalmente gerido na nuvem. É uma coleção de recursos computacionais chamados "Nós", que são organizados num grupo chamado *cluster*. Cada *cluster* tem um motor Amazon Redshift e contém uma ou mais base de dados (Amazon, 2021a). Alguns clientes que usam o Amazon Redshift são a Pfizer, o McDocnald's e a Fox (Amazon, 2021b). Para ilustrar o AD que foi referido, apresenta-se a Figura 21.

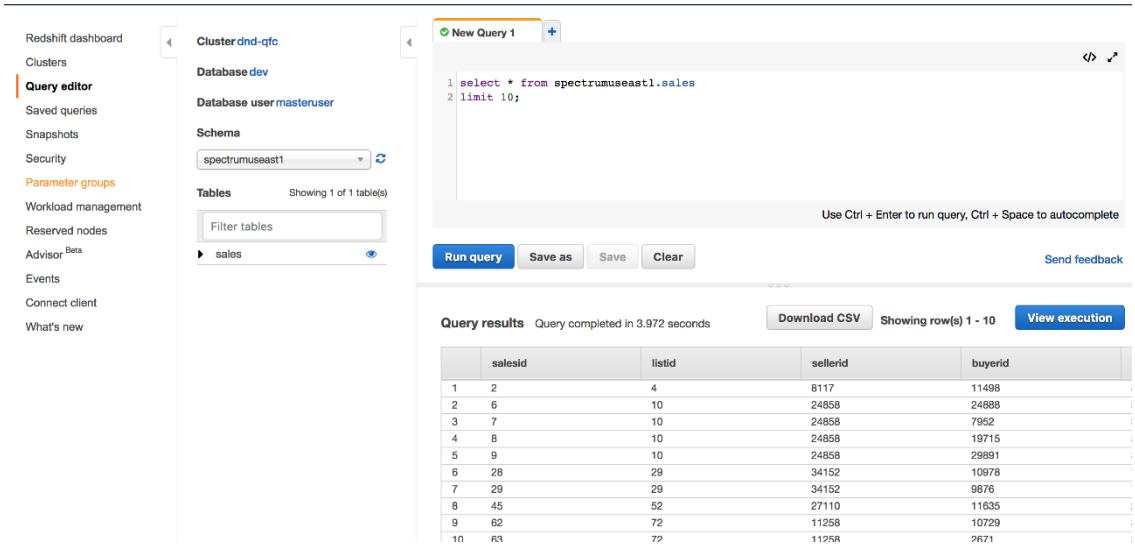

Figura 21 - Amazon Redshift (Amazon, 2021c)

Amazon Redshift tem duas modalidades de pagamento, pagar sobre demanda ou então reservar instâncias. Quando se reservam instâncias existem três tipos de modalidades de pagamento, não pagar adiantado: não se paga nada adiantado e compromete-se pagar mensalmente durante um ano; pagar parcialmente adiantado: paga-se uma porção da instância reservada adiantado e o restante no decorrer de um ou dos três anos de contrato; pagar tudo adiantado: paga-se adiantado todo o contrato da instância reservada, que pode ser de um ou três anos (Amazon, 2021c).

Existem diferentes tipos de máquinas que Amazon Redshift disponibiliza com diferentes características como se pode verificar na Tabela 2.

Tabela 2 - Características máquinas Amazon Redshift (Amazon, 2021c)

	VCPU	MEMORIA	ARMAZENAMENTO	I/O
DC2.LARGE	2	15 GB	0.16TB SSD	0.60 GB/s
DC2.8XLARGE	32	244 GB	2.56TB SSD	7.50 GB/s
DS2.XLARGE	4	31 GB	2TB HDD	0.40 GB/s
DS2.8XLARGE	36	244 GB	16TB HDD	3.30 GB/s
RA3.4XLARGE	12	96 GB	128TB RMS	2.00 GB/s
RA3.16XLARGE	48	384 GB	128TB RMS	8.00 GB/s

Na Tabela 3 serão apresentadas as diferentes modalidades de pagamento por máquina. Todos os preços das instâncias reservadas são apresentados se for executado o pagamento total adiantado para simplificar a comparação dos dados na tabela.

Tabela 3 - Preçário Amazon Redshift

MÁQUINAS	SOBRE DEMANDA/H	INSTÂNCIA RESERVADA 1 ANO/H	INSTÂNCIA RESERVADA 3 ANOS/H
DC2.LARGE	$0.324	$0.208	$0.121
DC2.8XLARGE	$6.048	$4.100	$1.940
DS2.XLARGE	$1.026	$0.664	$0.286
DS2.8XLARGE	$8.208	$5.310	$2.280
RA3.4XLARGE	$3.894	$2.570	$1.460
RA3.16XLARGE	$15.578	$10.300	$5.840

2.5.1.3 Microsoft

A Microsft surge no Quadrante Mágico dos líderes. A mesma apresenta vários produtos dos quais são exemplos: Azure Synapse Analytics, Azure SQL Database, Azure SQL Managed Instance, Azure Cosmos DB, Azure HDInsight e Azure Database; apesar disto, esta dissertação pretende-se focar no seu AD que é o Azure Synapse Analytics.

Azure Synapse Analytics é um serviço de análise ilimitada que traz em conjunto, a integração de dados, os armazéns de dados e análise de *Big Data*. Também dá a liberdade de escolher se pretende-se ter os dados numa solução sem servidor ou com os recursos à escala do projeto do utilizador (Microsoft, 2021a). Este tem como clientes a Walgreen, a Co-op e a Neogrid. Na Figura 22 pode-se observar a interface gráfica do Azure Synapse Analytics.

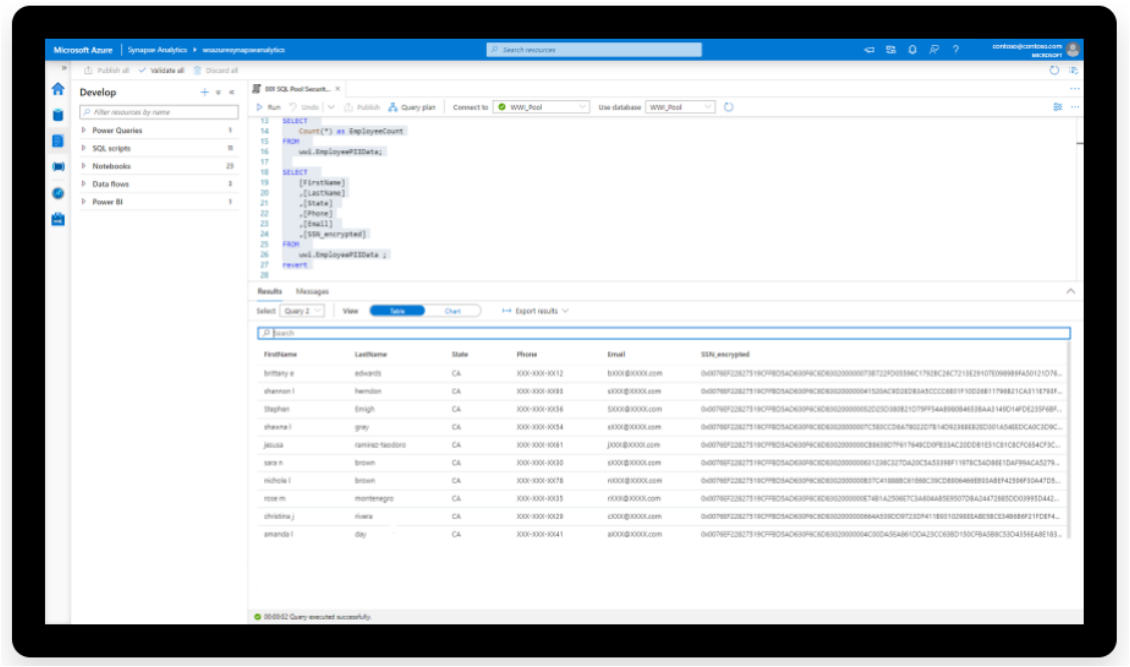

Figura 22 - Azure Synapse Analytics (Microsoft, 2021a)

O plano de preços deste serviço é composto por dois níveis de recursos: computação e armazenamento. No que toca ao armazenamento este é pago a $5 por TB de dados processados. A computação é medida por Data Warehouse Units (DWU) que foram projetadas para cargas de trabalho intensivas com maior desempenho das consulta e necessidades de escalabilidade computacional. As taxas de computação vão desde as 100DWU, que custam $1.20/hora até às 30000DWU que custam $360/hora.

O armazenamento de dados é cobrado à taxa de $23 por TB dos dados processados ($0,04/1 TB/hora). O armazenamento de dados inclui o tamanho do seu armazém de dados e 7 dias de armazenamento incremental. Também é cobrada uma taxa para a redundância geográfica para a recuperação em caso de desastre. Essa taxa começa nos $0.057/GB/mês.

2.5.1.4 Oracle

A Oracle localiza-se no Quadrante Mágico dos líderes. Embora ofereça vários produtos: Autonomous Transaction Processing e Autonomous Data Warehouse; esta dissertação pretende focar-se no seu AD que é o Autonomous Data Warehouse.

Autonomous Data Warehouse é um serviço sediado na nuvem de armazéns de dados que elimina praticamente todas as complexidades de operar um armazém de dados. A gestão do armazém de dados é automatizada para as tarefas de configuração, segurança, provisionamento e escalabilidade, removendo grande parte das tarefas manuais e complexas que podem introduzir erros humanos (Oracle, 2021). A Figura 23 apresenta a interface gráfica do AD referido.

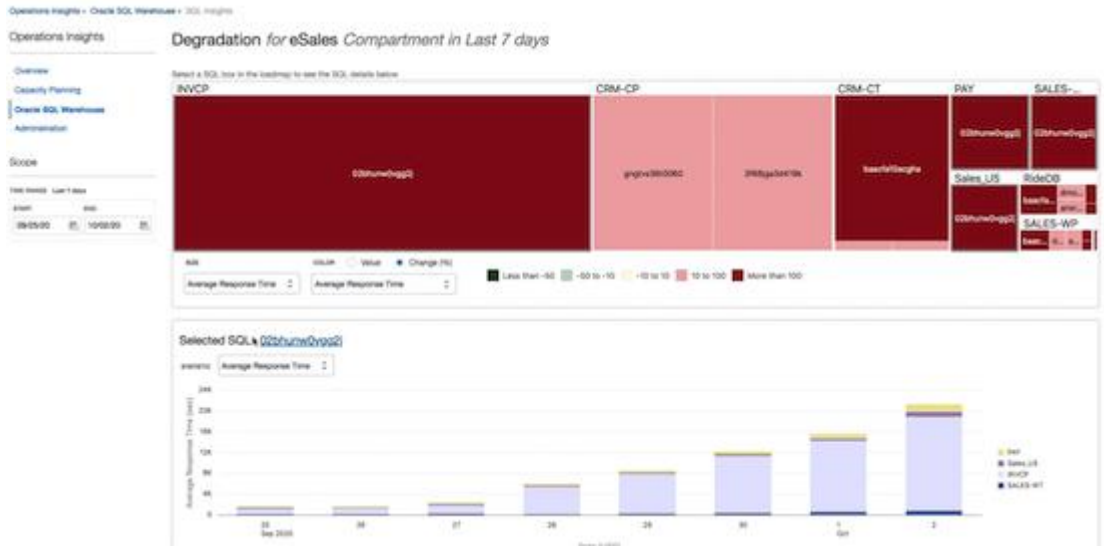

Figura 23 – Oracle Autonomous Data Warehouse (Oracle, 2021)

Este serviço oferece três tipos de implantação:

- Infraestrutura partilhada (nuvem publica);
- Infraestrutura dedicada (nuvem publica);
- Infraestrutura Cloud@Customer (centro de dados do cliente).

O preço da implantação numa infraestrutura partilhada é de $ 1.3441 por OCPU[1] por hora mais $118.40 por TB por mês. Já na infraestrutura dedicada o preço varia no tipo de máquinas onde vai estar alocado o armazém de dados que vai desde $14.5162 por hora num quarto de prateleira X8, até $86.0215 por hora numa prateleira completa X7. Para se perceber melhor o mencionado, passa-se a explicar o conceito de prateleira, sendo que é o local onde está sediado o sistema que contem a base de dados.

2.5.1.5 Comparação entre tecnologias

Como se pode verificar, existem diversas ferramentas que permitem a criação e a gestão de um AD na nuvem de forma simplificada.

Devido à existência da possibilidade de a empresa querer migrar a infraestrutura para a Google, do BigQuery ter uma boa gestão das *cached queries*[2], ser mais barato e já existir conhecimento na empresa de como utilizar esta ferramenta, foi decidido que se iria utilizar o BigQuery como AD. Posto isto, todas a ferramentas que irão ser referidas posteriormente, tanto para a criação do processo ETL, bem como para a criação dos relatórios e dos *dashboards*, terão de ter integração como o BigQuery.

[1] Oracle CPU

[2] Consultas que ficam guardadas em memória caso seja uma operação frequentemente repetida

2.5.2 Ferramentas ETL

De acordo com a subsecção anterior a ferramentas a utilizar terão que oferecer a possibilidade de integração com o BigQuery e segundo Hevo (Hevo, 2021a), existem diversas ferramentas para desenvolver um processo ETL para o BigQuery, como por exemplo: Google Cloud Platform Data Flow, Hevo Data, Talend.

2.5.2.1 Google Cloud Platform Data Flow

Google Cloud Platform (GCP) Data Flow é um sistema interno da Google sediado na nuvem com capacidade para processar dados em tempo real ou em pedaços. Este consegue auto escalar automaticamente de forma a detetar o número de trabalhadores necessários dependendo do volume dados de cada tarefa. Também oferece um conjunto de ferramentas para desenvolver a *pipeline* através do Apache Beam (Google, 2021c). Na Figura 24 pode-se verificar um exemplo de uma *pipeline* em GCP Data Flow.

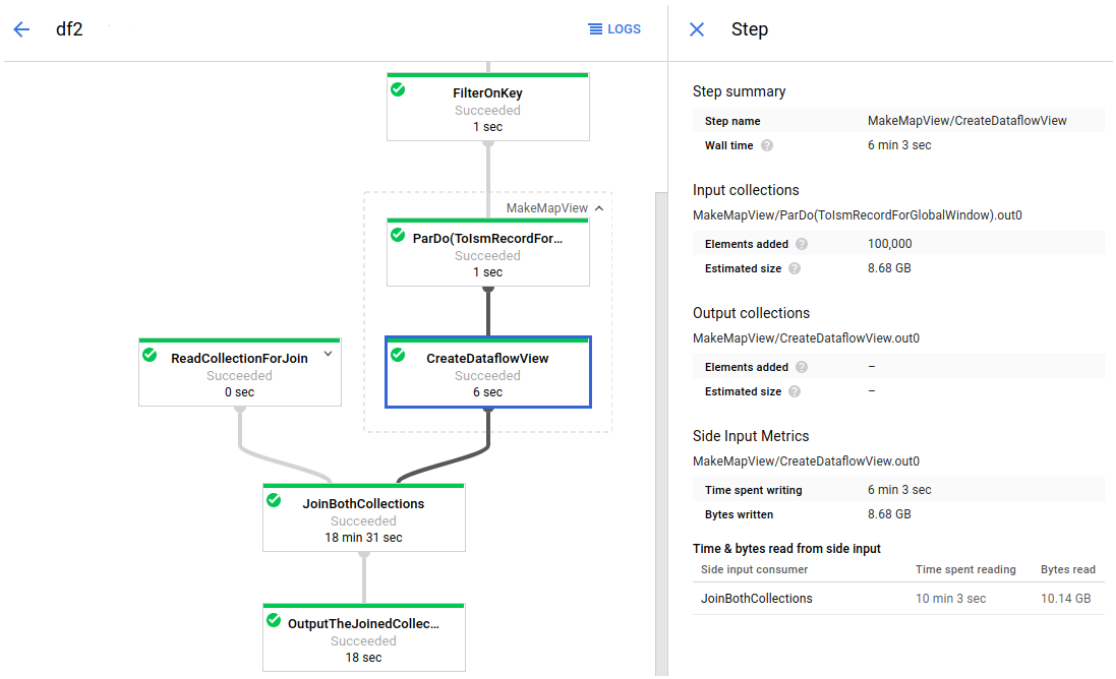

Figura 24 – Processo ETL através do GCP DataFlow (Google, 2021c)

O GCP tem um preçário diferente pelas diversas regiões, como é pretendido criar um AD na Europa irá ser observado os preços da região da Bélgica (europe-west-1). Os preços são diferentes pelo tipo de processamento que são em *batchs* e em *streaming,* e pelos recursos que cada processo consome, como vCPU, memória e armazenamento. A Tabela 4 - Preçário GCP DataFlow apresenta os preços de cada um do tipo de processamento (Google, 2021d).

Tabela 4 - Preçário GCP DataFlow

Tipo de processo	vCPU/h	Memoria/GB/h	Dados processados / GB
Batch	$0.059	$0.004172	$0.011
Streaming	$0.072	$0.004172	$0.018

2.5.2.2 Hevo Data

Hevo Data é uma pipeline de dados sem código. Suporta a integração de dados pré-construída para mais de cem fontes, incluindo o Google Big Query. Não requer gestão e manutenção de infraestrutura, migra os dados em tempo real, deteta os esquema dos dados que está a receber e mapeia-os para o esquema destino (Hevo, 2021c). A Deliverr, a Meesho e a Groww são alguns exemplos de clientes da Hevo Data (Hevo, 2021b). Como exemplo do processo ETL através da Hevo Data é apresentada a Figura 25.

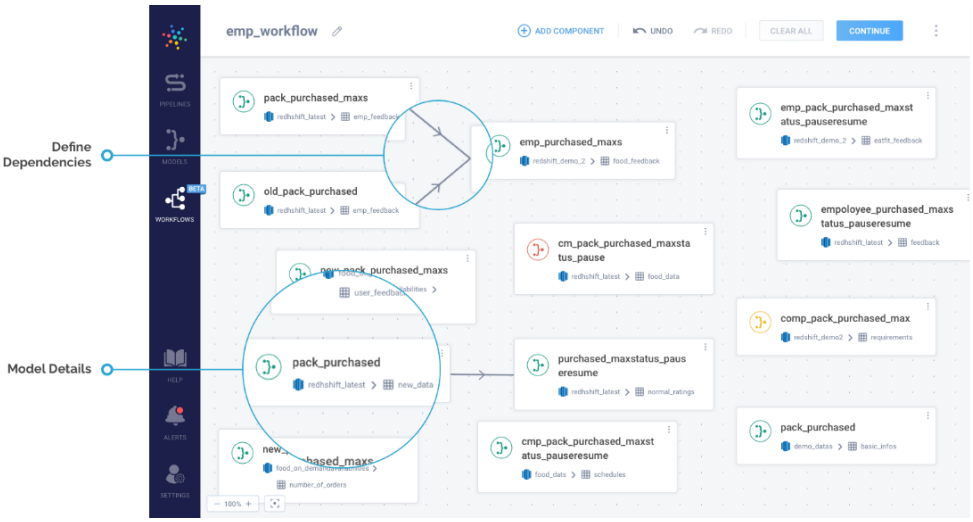

Figura 25- Processo ETL através do Hevo Data (Hevo, 2021c)

A Hevo tem três planos de preços, o Basic, o Starter e o Business. O Basic custa $249 por mês e inclui duas fontes de dados para a integração e 10 modelos de dados, podendo-se processar vinte milhões de eventos. O custo do Starter pode ir desde $399 por mês que permite o processamento de cinquenta milhões de eventos, até $799 por mês que permite o processamento de trezentos milhões de eventos. Este inclui também cinco fontes de dados para integração e vinte e cinco modelos de dados. Por último, existe o plano Business que pode custar entre $799 por mês, que permite processar quatrocentos milhões de eventos, até $1599 por mês e permite processar mil milhões de eventos. Este oferece ainda a integração até dez fontes de dados e cinquenta modelos de dados (Hevo, 2021d).

2.5.2.3 Talend

A Talend é uma ferramenta para criar o processo de ETL baseado em arrastar e largar componentes de uma palete. Esta palete permite integrações, a partir de diversas fontes de dados e permite também executar várias transformações sobre os dados para estes ficarem mais limpos, sendo possível observar um exemplo de interface gráfica desta ferramenta, através da Figura 26 - Processo ETL através do Talend (Talend, 2021b). Alguns dos clientes da Talend são a Toyota, a Loreal e o Domino's (Talend, 2021b).

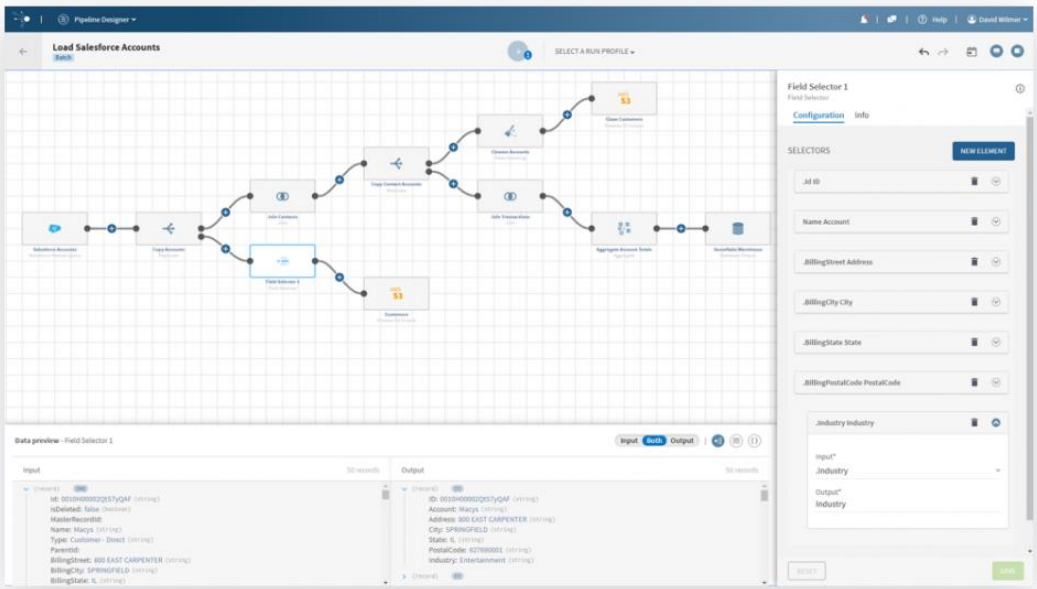

Figura 26 - Processo ETL através do Talend (Talend, 2021b)

A Talend tem diferentes planos de pagamento. O Talend Open Source que é gratuito para todos os clientes, o Stitch Data Loader que custa entre $100 a $1000 por mês, o Talend Pipeline Designer que tem um custo baseado no uso por hora, o Talend Data Cloud Integration que custa $1,170 por mês por utilizador ou se for pago anualmente custa $1200 por utilizador, por último existe o plano Talend Data Fabric que inclui todas as ferramentas da Talend, mas o preço só é revelado mediante o pedido de um orçamento (Talend, 2021a).

2.5.2.4 Comparação Ferramentas ETL

Os critérios para a escolha da ferramenta para o processo ETL são: permitir o processamento em lotes, carregamento dos dados para o BigQuery, conexão com o dynamoDB e o mongoDB e o preço. Por último, um dos critérios de escolha da ferramenta para executar o processo ETL é a permissão de construir o processo através de código, pois dará uma maior flexibilidade, uma maior portabilidade e será mais fácil manter. De seguida, apresenta-se a Tabela 5.

Tabela 5 - Tabela Comparativa entre a ferramentas de ETL

Funcionalidades	CGP DataFlow	Hevo Data	Talend
Processamento em Batch	☒	☒	☒
Processamento em Streaming	☒	☒	☒
ETL através de código	☒	☐	☐
Carregamento para o Big Query	☒	☒	☒
Integração com o DynamoDB	☒	☒	☒
Conexão com o MongoDB	☒	☒	☒
Integração com o AWS S3	☒	☒	☒

Posto isto, verificou-se que existem diversas ferramentas para a criação de um processo ETL. Efetivamente, consegue-se salientar as vantagens e desvantagens de cada uma destas ferramentas, no entanto, as mesmas poderão ser adequadas e poder-se-á tirar o melhor proveito das suas funcionalidades para as necessidades de cada empresa.

2.5.3 Ferramentas OLAP

Tendo em conta o que foi mencionado até então, surge neste trabalho a abordagem a algumas ferramentas de apresentações de gráficos e relatórios, possibilitando uma análise exploratória e aprofundada dos dados.
Para a escolha da ferramenta de relatórios e de apresentação de gráficos para analisar os dados no armazém de dados, teve de se realizar uma comparação entre ferramentas que tivessem integração com o BigQuery. Segundo Tuan Nguyen em (Tuan, 2020), as principais ferramentas para relatórios e apresentação de gráficos com integração com o BigQuery, são o: o Google Data Studio, Power BI, Looker e Tableau.

2.5.3.1 Google Data Studio

Google Data Studio é uma ferramenta gratuita de apresentação de gráficos do ecossistema da Google, o que facilita a sua integração com o BigQuery. Este sistema transforma os dados em relatórios e painéis informativos, fáceis de ler e personalizáveis, que é visível Figura 27. No editor de relatórios pode-se arrastar e soltar as componentes para (Google, 2021f):

- Criar gráficos de linhas, barras, piza, mapas geográficos, gráficos de áreas e bolhas, tabelas de dados paginadas, tabelas dinâmicas, entre outros;
- Incluir links e imagens clicáveis para criar catálogos;

- Adicionar anotações com texto e imagens;
- Aplicar estilos e temas de cores;

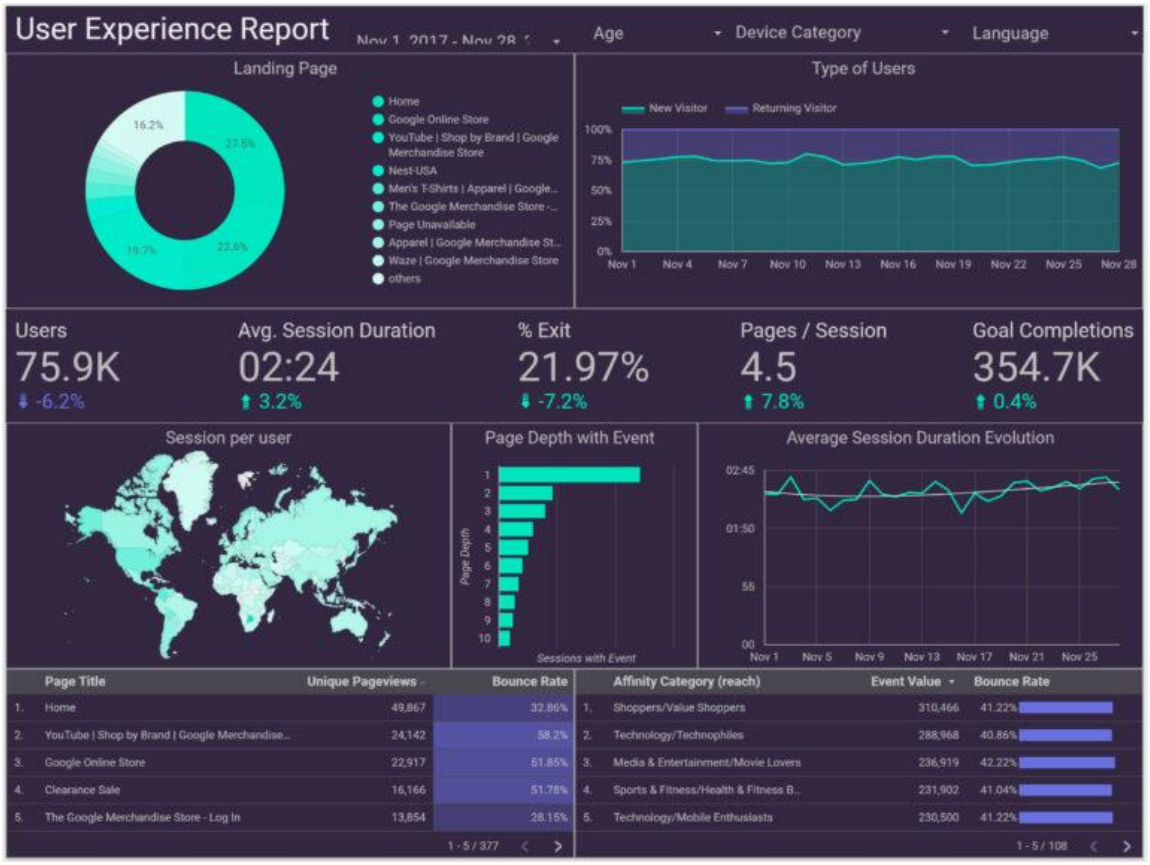

Figura 27 - Google Data Studio (Google, 2021f)

2.5.3.2 PowerBI

PowerBI é uma ferramenta de relatórios e apresentação de gráfica dos dados, que pertence à Microsoft. É a ideal para empresas que adotaram o ecossistema da Microsoft. É uma ferramenta que tem uma interface gráfica amigável para o utilizador, suporta a ligação com centenas de fontes de dados, que vão desde serviços na nuvem como o Big Query, até ficheiros locais como Excel. Esta oferece uma versão para o computador e uma versão móvel para Android e IOS. Esta ferramenta só funciona em computadores com o sistema operativo Windows, não suportando outros sistemas operativos como Linux ou Mac (Microsoft, 2021d). Na Figura 28 é apresentado um exemplo da interface gráfica da ferramenta.

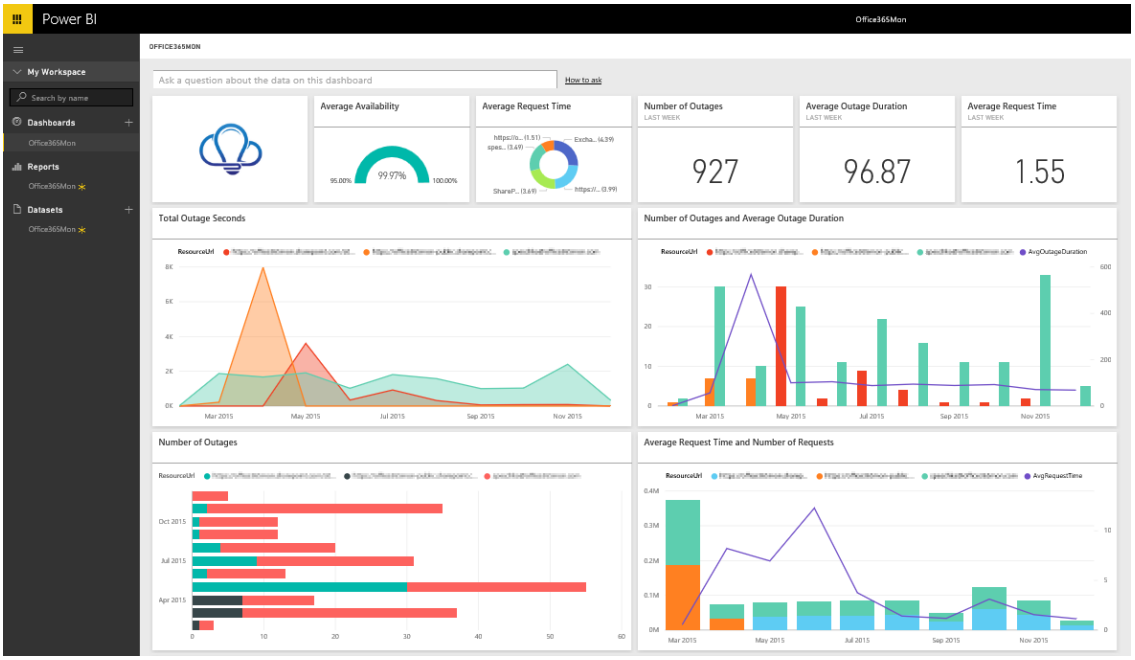

Figura 28 – PowerBI (Microsoft, 2021d)

PowerBI tem três planos de pagamento, versão gratuita, Pro e Premium. A versão gratuita pode ser descarregada para o computador, só se pode visualizar os dados nos locais onde a ferramenta está instalada e não permite a partilha de relatórios ou gráficos. A versão Pro, pode ser acedida na nuvem, permite a partilha de relatórios e gráficos entre utilizadores com a versão Pro, é totalmente gerida pela Microsoft e tem um preço de 9.99$ por mês por utilizador. A versão Premium acrescenta mais recursos de forma a se conseguir fazer analises mais detalhadas, oferece uma nuvem com recursos de computação e armazenamento dedicado e permite a partilha de conteúdo do Power BI com qualquer utilizador. Tem um preço de $4,995 por mês por nuvem dedicada (Microsoft, 2021c).

2.5.3.3 Looker

É um produto da Google, que é uma ferramenta de BI poderosa que fornece uma abordagem intuitiva para a análise e exploração de dados em tempo real. Looker corre inteiramente num explorador de internet não precisando assim de se instalar nenhum programa, o que torna mais fácil a entrega de dados e a colaboração entre utilizadores. Permite a automatização de relatórios, podendo agendar o envio de emails diários, semanais ou mensais com os relatórios ou até alertar de possíveis anomalias com os dados. Também possibilita a integração com o Github, facilitando assim o versionamento e possibilidade de vários utilizadores trabalharem sobre os mesmos relatórios (Looker, 2021). A Figura 29 permite perceber o que foi referido.

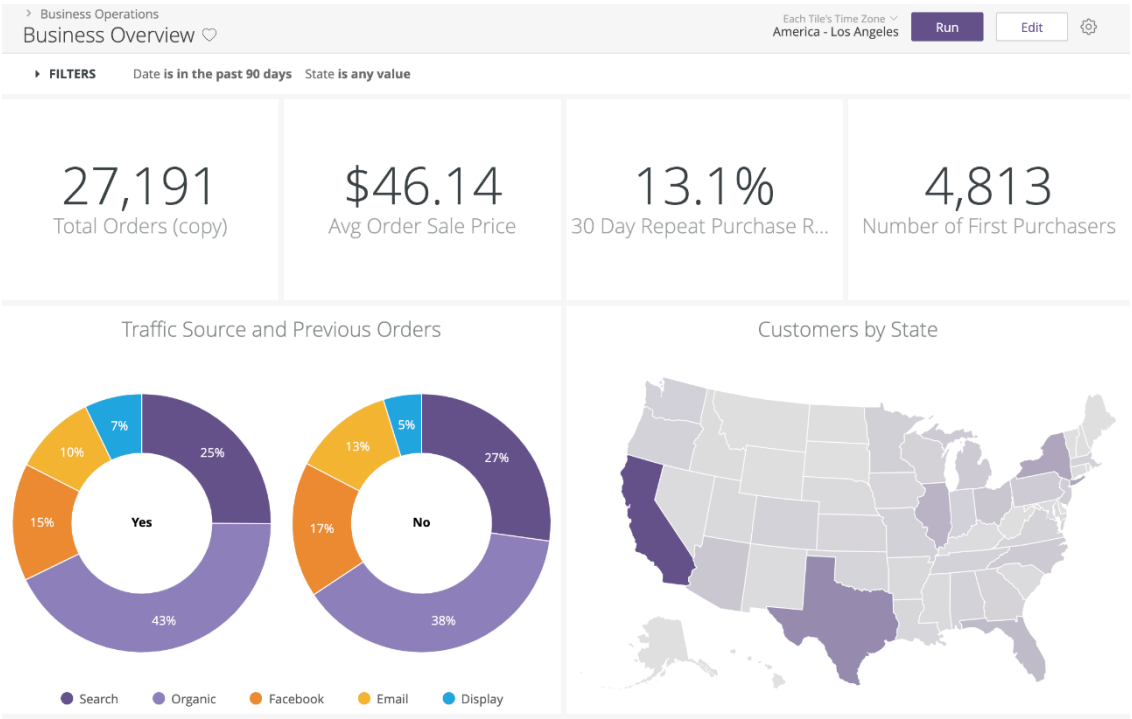

Figura 29 – Looker (Looker, 2021)

O preçário de utilização do Looker só está disponível sobre pedido de orçamento.

2.5.3.4 Tableau

Tableau é uma ferramenta poderosa para a visualização de dados usada na indústria de BI. Esta ajuda na simplificação de dados "crus" para um formato mais compreensível. Tableau ajuda a criar os dados que possam ser compreendidos pelos utilizadores de uma organização entre diferentes departamentos, tal como se vê na Figura 30. Tableau oferece vários produtos (Tableau, 2021a):

- **Tableau Prep:** oferece uma forma visual e direta para combinar, modelar e limpar os dados.
- **Tableau Desktop:** um programa que oferece uma interface intuitiva que encoraja a curiosidade, criatividade e tomada de decisão baseada em dados;
- **Tableau Server e Tableau Online:** permite estender os dados por toda a organização através de estes ficarem alocados numa nuvem;
- **Tableau Mobile:** permite ter a aplicação do Tableau no telemóvel Android ou iOS;

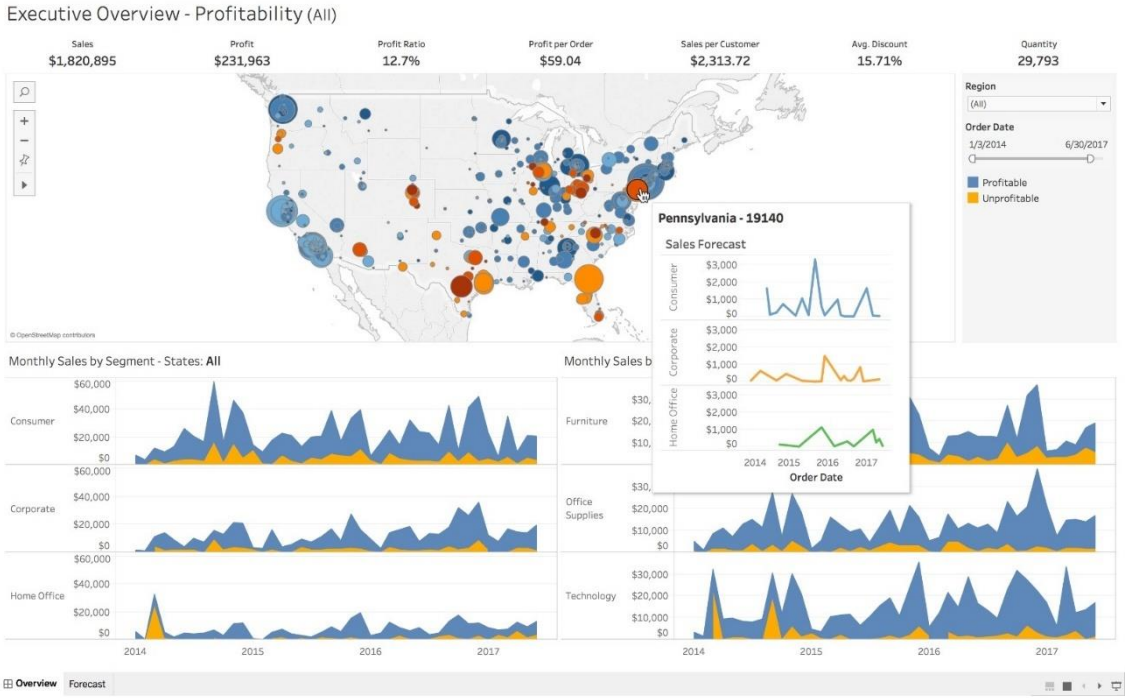

Figura 30 - Tableau

O Tableau oferece diferentes planos, o Tableau Viewer que oferece uma licença do Tableau Server e permite visualizar e interagir com os gráficos, tem um custo de $12 por utilizador por mês. O Tableau Explorer oferece uma licença do Tableau Server e permite o acesso dos recursos de análise de forma a obter as respostas pretendidas de forma mais rápida, este plano tem um custo de $35 por utilizador por mês. Por último, existe o Tableau Creator que oferece todos os recursos referidos anteriormente e tem um custo de $70 por utilizador por mês (Tableau, 2021b).

2.5.3.5 Comparação Ferrramentas OLAP

De seguida, apresentar-se-á uma comparação com as características que para a empresa mais importa neste tipo de ferramentas, que são: integração com o Big Query, suporte de partilha dos gráficos e relatórios entre as várias pessoas da empresa, construção de gráficos através de arrastar e soltar os recursos existentes, apresentação dos gráficos em tempo real, ser compatível com os diferentes sistemas operativos e o preço.

Tabela 6 - Tabela comparativa de ferramentas de relatórios e gráficos

Funcionalidades	Data Studio	PowerBI	Looker	Tableau
Integração Big Query	☒	☒	☒	☒
Partilha de relatórios e gráficos	☒	☒	☒	☒
Função de arrastar e largar recursos	☒	☒	☐	☒
Apresentação de relatórios e relatórios em tempo real	☒	☒	☒	☒
Integração com diferentes sistemas operativos	☒	☐	☒	☒
Preço	Gratuito	$0-$4,995/mês	Sob orçamento	$12–$70/mês

Nesta tabela, Tabela 6, é possível verificar que as quatro ferramentas têm integração com o Big Query, todas dão para a partilha de relatórios e gráficos, apenas Looker não dá para a função de arrastar e largar recursos, todas permitem a apresentação de relatórios e relatórios em tempo real, na integração com diferentes sistemas operativos é excluída a PowerBI, por fim, todas elas apresentam um custo associado, exceto o Data Studio.

2.6 Conclusão

Tendo em vista todas as secções desenvolvidas anteriormente, e após a sua análise, torna-se essencial evidenciar que existem diferentes arquiteturas e tecnologias para a construção de um processo de BI. Verificou-se que a criação deste processo tem de ser realizada em três etapas tal como foram exploradas anteriormente: construção do processo ETL, construção do armazém de dados, criação dos diversos relatórios e dashboards. Para além disto, também se constatou que para cada uma das etapas existem diferentes tecnologias, cada uma delas com as suas particularidades e benefícios, que dependem e variam consoante o contexto em que são aplicadas.

Como já foi referido anteriormente, para este projeto foi escolhido o armazém de dados da Google, o Big Query, tendo sido feita esta escolha com base no seu valor monetário, na sua possibilidade de *cache* de *queries, ainda,* a existência da possibilidade de a empresa migrar toda a infraestrutura para os serviços da Google e, também, a existência de algum conhecimento, dentro da empresa, de trabalhar com o Big Query. Posto isto, a ferramenta escolhida para o processo de ETL foi o Google DataFlow, uma vez que oferece uma grande flexibilidade de

personalização de componentes via código, e, também, o facto de a mesma se encontrar presente no ecossistema da Google e os seus custos serem bastante reduzidos. E, por último, a ferramenta escolhida de apresentação de gráficos e relatórios foi o Google Data Studio, também pelo facto de pertencer ao ecossistema da Google, ter a integração automática com o Big Query e por ser gratuita.

3 Análise de Valor

Este capítulo tem como objetivo apresentar a análise de valor. Nas secções seguintes irão ser descritas as oportunidades identificadas e analisadas que irão acrescentar valor à empresa, bem como o produto que irá ser desenvolvido.

3.1 Processo Fuzzy Front-End

No processo de inovação e proposta de valor, uma das técnicas utilizadas é o Fuzzy Front-End (FFE). É geralmente considerado como uma das maiores oportunidades para melhorar o processo global de inovação. FFE é distribuído por cinco elementos de forma a suportar a inovação de forma sustentável, como identificação de oportunidade, geração e enriquecimento da ideia, seleção da Ideia e definição do conceito (Belliveau et al., 2002).

3.1.1 Identificação da oportunidade

A oportunidade surge devido à necessidade de recolher métricas dos serviços que a empresa oferece, de forma que se consiga identificar quais são os pontos passiveis de melhoria no processo e de forma a conseguir apresentar a possíveis clientes indicadores, que apresentem um grau de confiança elevado, o desempenho dos serviços oferecidos pela empresa.

3.1.2 Análise de oportunidade

Com o objetivo de sustentar a oportunidade referida na secção anterior, é necessário analisar a mesma, de maneira a perceber quais são as vantagens e desvantagens desta solução. Para realizar um diagnóstico estratégico irá ser realizada uma análise SWOT. As letras SWOT

referem-se a Strenghts (pontos fortes), Weaknesses (pontos fracos), Opportunities (oportunidades) e Threats (ameaças). Os pontos fortes e pontes fracos dizem respeito a fatores internos que estão sob o controlo da empresa, já as oportunidades e ameaças correspondem a fatores externo que são extrínsecos ao controlo da empresa (Economias, 2021). Na Figura 31 irá ser realizada uma análise SWOT relativamente a solução apresentada.

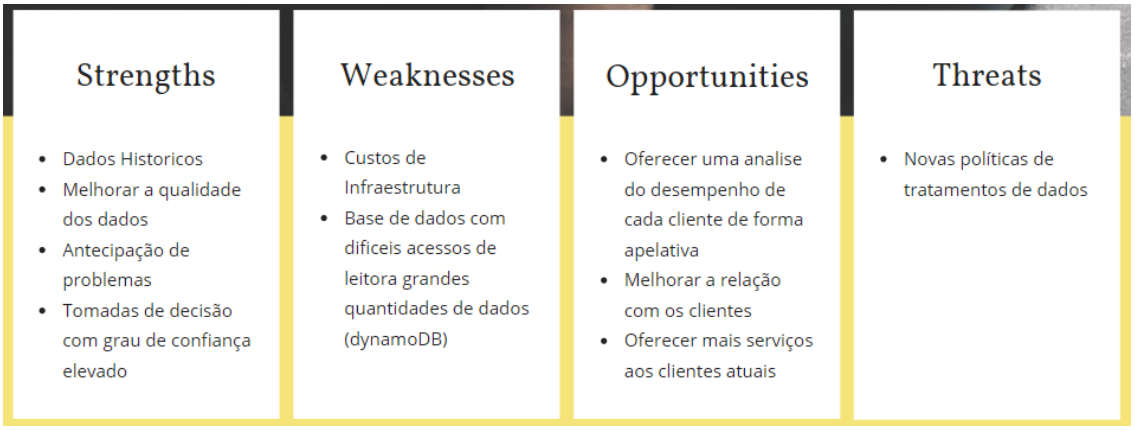

Figura 31 - Análise SWOT

3.1.3 Geração e Enriquecimento da Ideia

De forma a tornar possível a tomada de decisão utilizando dados, é necessário ter dados que apresentem um grau de qualidade elevado para que essas decisões possam ter um grau de confiança elevado. Efetivamente, para se concretizar este objetivo, surgiu a ideia de se implementar um sistema que com ferramentas de BI torne os dados coesos e confiáveis.

3.1.4 Seleção da Ideia

A ideia apresentada no ponto anterior integra uma solução de BI. A partir de tecnologias apropriadas, será construído um sistema de recolha, tratamento e carregamento de dados para uma fonte de informação com dados históricos imutáveis onde os mesmos serão apresentados e de forma detalhada e sumaria, através de tabelas, gráficos e relatórios.

3.1.5 Definição de Conceito

A solução apresentada anteriormente irá expor a informação recolhida e armazenada no AD de forma sucinta, de fácil interpretação, oferecendo indicadores e dashboards relevantes para o desempenho do negócio.

3.2 Modelo de Negócio CANVAS

O modelo de negócio CANVAS irá ser usado nesta secção com o objetivo de dar outra perspetiva sobre a solução a ser desenvolvida.

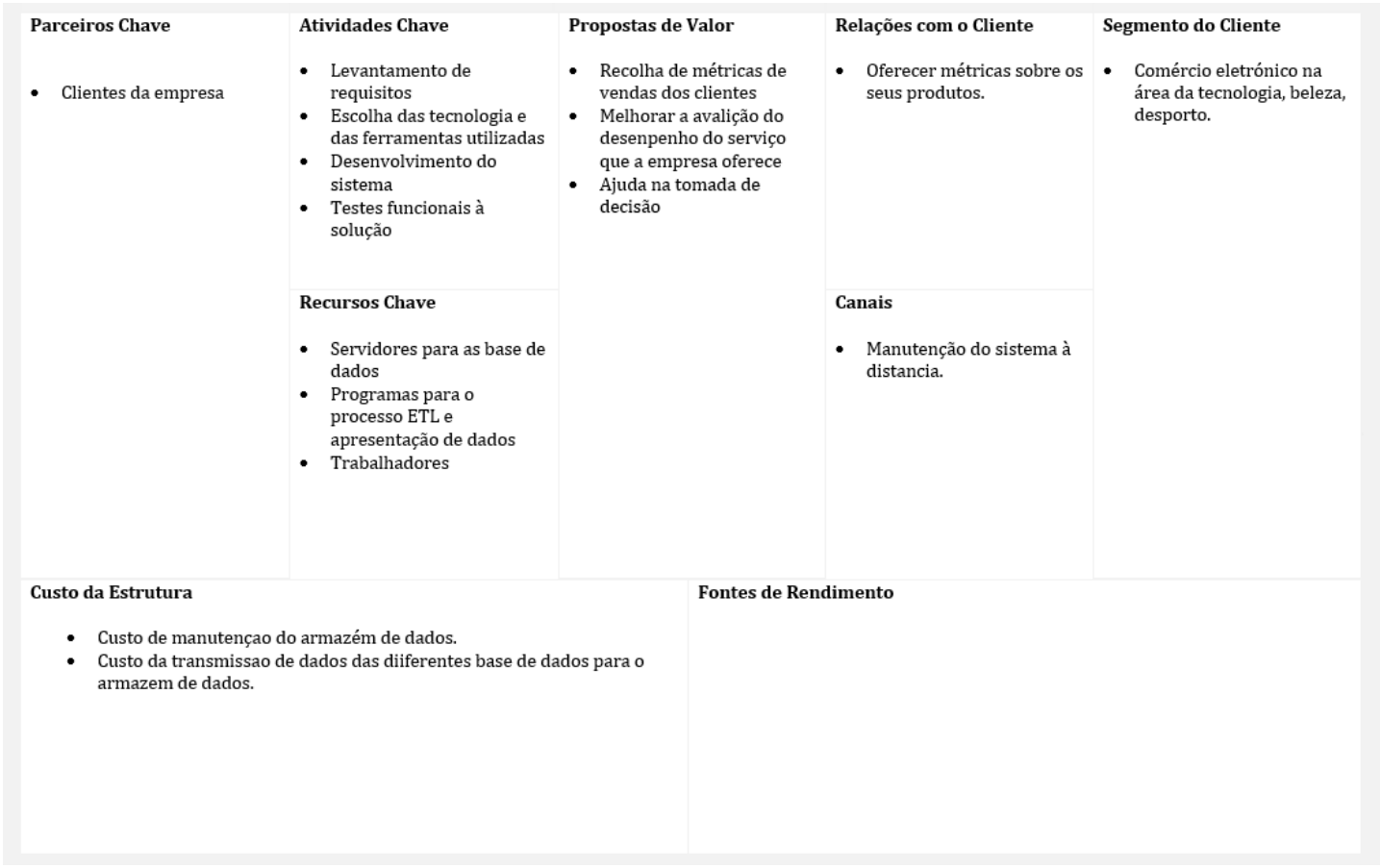

Figura 32 - Modelo de CANVAS

Para ajudar na compreensão da Figura 32, o texto seguinte irá explicar com maior detalhe cada um dos pontos apresentados no modelo de negócio CANVAS.

Parceiros Chave

Este ponto refere-se a cada um dos clientes da empresa, pois sem estes não existiriam dados relevantes para a criação de um processo de BI.

Atividades Chave

Para a criação da solução é necessário passar pelas seguintes atividades: Levantamento de requisitos, escolha das tecnologias e das ferramentas utilizadas, desenvolvimento do sistema, testes funcionais à solução.

Recursos Chave

Para se conseguir produzir a solução expectável é necessário um servidor onde irá ficar sediado o armazém de dados, os programas para a construção do processo ETL, os programas para apresentação dos dados em gráficos e relatórios e por fim os funcionários, que vão desde os programadores às outras equipas para perceber quais são todos os requisitos do negócio.

Proposta de Valor

É o "valor" ou benefício que a solução entregará ao cliente, que vai desde a melhoria e consolidação dos dados para as tomadas de decisão como possibilitará no futuro ser vendido aos clientes de forma a estes poderem ter acesso a métricas das suas vendas relevantes.

Relação com o cliente

Oferecendo aos clientes métricas sobre os seus sistemas, isto trará uma retenção adicional dos clientes, pelo facto de ser mais um serviço que estes podem usufruir.

Canais

São as formas de comunicação e de distribuição pelos clientes e pela organização, que neste caso em específico é remotamente.

Custo da Estrutura

São os custos que derivam do desenvolvimento da solução e da sua manutenção, que representa os custos da manutenção do armazém de dados e da transmissão da informação das bases de dados operacionais para o armazém de dados.

Fontes de Rendimento

Neste caso, esta solução não trará fontes de rendimentos diretas, ou seja, ajudará a empresa a tomar melhores decisões no ponto de vista estratégico o que a ajudará a crescer, melhorando assim as suas fontes de rendimento.

Segmentação do Cliente

São clientes que estão dispostos a pagar pelo serviço oferecido, que nesta solução abrange cliente do comércio eletrónico da área das tecnologias, beleza e desporto.

3.3 Cadeia de valor de Porter

Cadeia de valor pode ser definida como um conjunto de atividades que uma organização tem de realizar para criar valor para os seus clientes.

Em (Porter, 1998), Porter propôs a criação de uma cadeia de valor em como os recursos de entrada se tornam em recursos de saída que sejam comprados pelos consumidores. Posto isto, Porter concebe uma cadeia de atividades comuns a todos os negócios que pode ser dividida em dois setores, atividades primárias e atividades de suporte como é apresentado na Figura 33.

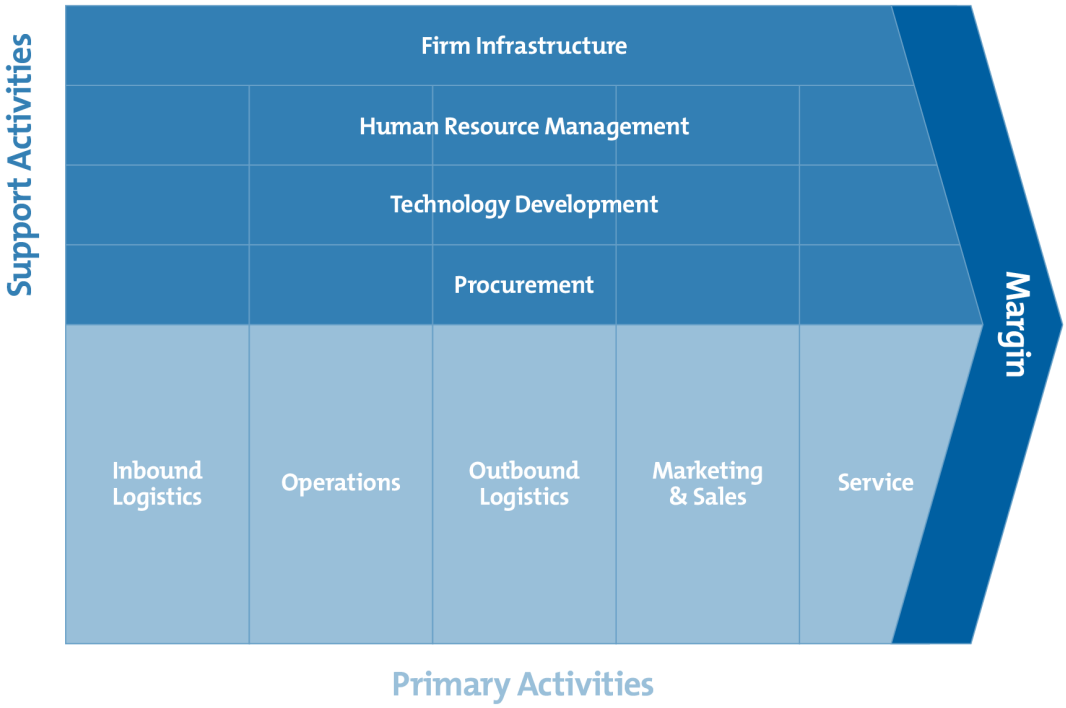

Figura 33 -Cadeia de Valor de Porter (Porter, 2021)

As atividades primárias consistem em:

Logística de entrada: são todos os processos de entrada de produtos que neste caso se pode entender como os dados a serem tratados.

Operações: são os processos que transformam entradas em saídas, no contexto desta dissertação pode-se entender como o processo de ETL.

Logística de saída: são atividades que permitem a entrada do produto ou serviço ao cliente, que neste contexto pode-se identificar como os relatórios e os gráficos finais.

Marketing e vendas: são os processos que incentivam o cliente a comprar o produto ou serviço, que se pode refletir como a disponibilização para os clientes das suas métricas de vendas.

Serviço: são atividades que mantêm o valor do produto ou serviço, que se enquadra na manutenção do armazém de dados bem como na geração de novos relatórios com diferentes requisitos.

Já as atividades de suporte podem ser divididas em quatro diferentes áreas:

Aquisição: recursos que necessitam de ser comprados, como as licenças dos programas a serem utilizados e os servidores para alojar o armazém de dados, para o desenvolvimento da solução.

Gestão dos recursos humanos: esta área representa como é que a empresa contrata, recruta treina e motiva os seus colaboradores, podendo proporcionar eventos, desafios e convívios entre os funcionários da empresa.

Infraestrutura: são os sistemas que suportam a empresa e que permitem a sua atividade diária, definindo uma estrutura organizacional com diferentes departamentos, como, por exemplo, recursos humanos, financeiro, administrativo e tecnológico.

Com a ajuda da cadeia de valor de Porter consegue-se perceber que existe potencial e valor no produto a ser desenvolvido e que este de adaptará ao contexto real de uma organização.

4 Análise e design

Como já foi referido nas secções anteriores, a empresa para a qual foi criada uma solução, oferece um serviço a outras empresas no ramo do comércio eletrónico. A mesma recolhe todas as interações nos sites dos seus clientes e, à medida que o número de clientes vai aumentando, e consecutivamente as suas vendas, é inevitável que a informação recolhida seja cada vez maior. Posto isto, é necessário que esses dados sejam apresentados de forma clara e intuitiva para efetivar e criar um processo para guiar a tomada de decisão. Surge então, a proposta de unificar os dados e tratá-los para que estes se encontrem num único local, de maneira que as decisões sejam tomadas baseadas nesta informação, apresentando a possibilidade de gerar relatórios e dashboards de forma flexíveis, ou seja, podendo escolher que variáveis visualizar e comparar.

Neste capítulo irá ser apresentado o design da proposta da solução e os requisitos para a construção da mesma.

4.1 Identificação dos atores

Os proponentes para a criação desta solução são os gestores da empresa e as equipas de categorias. . As equipas de categorias são responsáveis por catalogar os produtos dos clientes e criar as regras irão ser usadas para criar os conjuntos que a empresa fornece aos seus clientes.

Os gestores da empresa utilizarão a solução para conseguirem tomar decisões estratégicas, como por exemplo observar qual é o ramo de comércio eletrónico em que o serviço tem um melhor desempenho para averiguar quais os setores do mercado a incidir.

As equipas de categorias irão usufruir desta solução para tomar decisões operacionais, tais como analisar os relatórios e perceber quais são os conjuntos de categorias que têm um melhor desempenho, o que facilitará o seu trabalho no momento de criar as regras.

4.2 Requisitos

Neste capítulo serão apresentados os requisitos funcionais e não funcionais com o intuito de esclarecer as especificações do cliente/empresa e criar uma solução conforme as necessidades e os critérios pretendidos. Usou-se o sistema FURPS+ (IBM, 2020) para definir os requisitos deste capítulo, sendo que cada letra da sigla FURPS+ representa um tipo de requisito.

4.2.1 Requisitos Funcionais

Os requisitos funcionais, que estão representados pela letra F (Functionality) do sistema FURPS+ são a principal ferramenta de comunicação para o cliente apresentar os seus requisitos à equipa de projeto. Estes ajudam a equipa de projeto a manter-se na direção correta.

Para este projeto foram definidos os seguintes requisitos funcionais em formato de User Stories:

- Como gestor da empresa pretendo consultar a informação de vendas a qualquer altura.
- Como gestor da empresa pretendo obter o relatório de vendas em bruto de maneira que seja possível visualizar a informação de forma detalhada.
- Como gestor da empresa pretendo obter o relatório de vendas agregadas que me permita consultar de forma rápida os valores totais de vendas.
- Como gestor da empresa pretendo consultar os Dashboards para conseguir analisar os indicadores de desempenho relevantes ao negócio.
- Como membro da equipa de categorias pretendo obter o relatório de desempenho de categorias por forma a conseguir analisar o desempenho das categorias oferecidas pela empresa.
- Como membro da equipa de categorias pretendo obter o relatório de categorias cruzadas para comparar o desempenho das categorias vendidas pelo cliente com as da empresa de forma a criar a possibilidade de estabelecer novos conjuntos.

O relatório de desempenho de categorias tem a informação da venda dos produtos que são incluídos no serviço da empresa para perceber se os conjuntos sugeridos têm o desempenho esperado. O relatório das categorias cruzadas apresenta quais foram as combinações de vendas com melhor desempenho independentemente se as vendas foram realizadas pelo serviço oferecido ou através do cliente. Possibilitando, assim, a análise de possíveis futuros conjuntos. Estes relatórios são muito uteis para a equipa de categorias pois irá permitir quais são as categorias que funcionam em conjunto.

Já os gestores, pretendem analisar os relatórios de vendas em bruto e agregadas, como consultar os dashboards para conseguir obter indicadores de desempenho relevantes para o negócio.

4.2.2 Requisitos Não Funcionais

Os Requisitos não funcionais são as características que o sistema deve adquirir ao executar as suas funcionalidades. Também é utilizado o sistema FURPS+ para definir este tipo de requisitos.

O sistema deve processar os dados aproximando-os à realidade, ou seja, ao tempo em que são gerados, pois pretende-se que os gestores possam obter os dados em qualquer momento, mesmo em reuniões com clientes. Este requisito enquadra-se na letra P (*Performance),* o que significa que o sistema tem que ter um bom desempenho.

O sistema deve ser fácil de se usar, pois os atores não possuem conhecimentos técnicos e terão de conseguir utilizar o sistema com a mesma facilidade de alguém que tenha conhecimentos técnicos. Este requisito enquadra-se na letra U (*Usability),* o que significa que o sistema tem que ser fácil de se usar.

Os dados devem ser íntegros e fiáveis de forma a possibilitarem a tomada de decisão a partir dos mesmos, caso contrário estas não serão tão fiáveis. Este requisito enquadra-se na letra R (*Realibility*), o que se traduz que o sistema tem que ser fiável.

Por último, o sistema deve ser modular permitindo assim que novas funcionalidades sejam implementas com pouco esforço, ficando este requisito enquadrado com a letra S (*Supportability*).

4.3 Arquitetura Proposta

Com base nas arquiteturas propostas na Secção 2.3.7, foi selecionada a abordagem de Ralph Kimball para a concretização da solução proposta. O motivo pelo qual se optou por esta abordagem teve que ver com a dimensão da empresa, sendo ela relativamente pequena, e ainda pela urgência da obtenção da solução. Isto, vai ao encontro do que foi citado relativamente às particularidades de cada uma das propostas apresentadas. Tendo em conta o mencionado, será criado um armazém de dados subdivido em três tabelas: produtos, vendas e tracking.

Tendo em vista a implementação desta abordagem, é necessário salientar que a mesma concretiza um dos requisitos não funcionais: a modularidade. Com isto, torna-se possível a introdução de novas áreas, no futuro, tendo por base as necessidades do cliente ou da empresa, não imiscuindo com a solução existente.

Acrescenta-se ainda a possibilidade de que, querendo ou não recorrer a alguma alteração, quer para adicionar ou remover funcionalidades, ou até para manutenção, é possível fazer de forma rápida e sem causar transtornos.

O processo consiste em recolher os dados de diversas fontes, guardá-los numa base de dados de preparação (*staging),* transformá-los e carregá-los no armazém de dados. Tendo sido isto realizado, será utilizada uma ferramenta para se construir *Dashboards* e será criada uma API

que irá ser o ponto de ligação de outras aplicações ao AD. Esta, por exemplo, será a responsável por fornecer toda a informação já tratada para se criar os relatórios necessários. Na Figura 34, pode-se observar a arquitetura da solução proposta que será descrita nos capítulos seguintes.

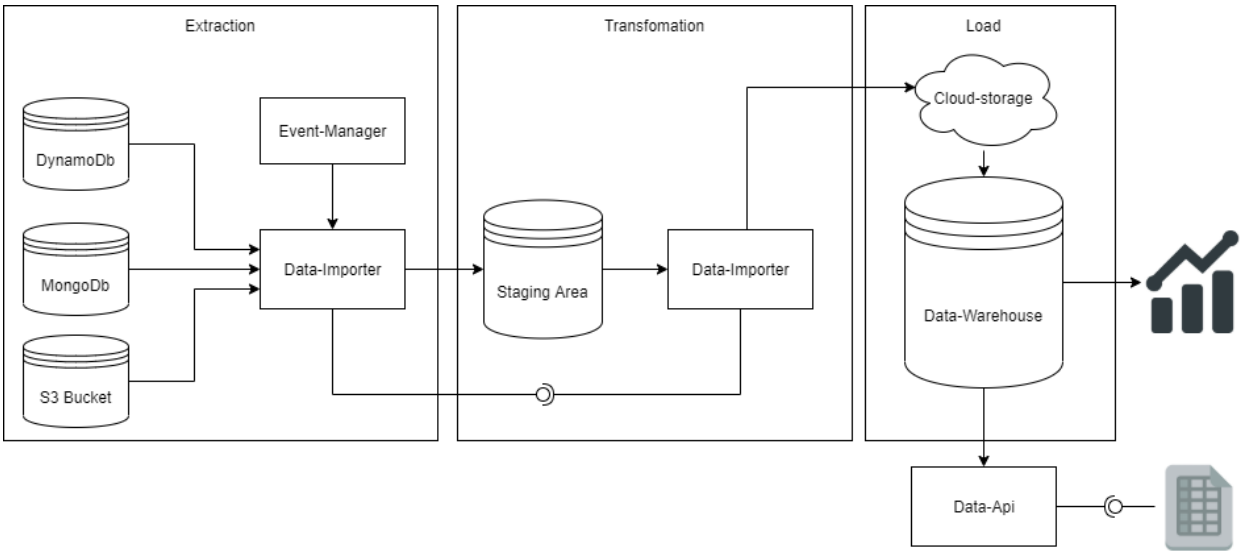

Figura 34 - Arquitetura Proposta

4.3.1 Fontes de dados

Os dados para desenvolver a solução proposta encontram-se dispersos por diferentes bases de dados de diferentes tipos. Os dados das vendas e de todas as interações de utilizadores nos sites dos clientes são armazenados na base de dados DynamoDB. Já os produtos estão armazenados numa base de dados MongoDB. Por fim, os descontos são armazenados em ficheiros S3. Devido a esta heterogeneidade dos dados surgiu a necessidade de criar uma solução na qual os dados estivessem todos centralizados, de forma a facilitar o processo de consulta de métricas.

4.3.2 Staging Area

A Staging Area ou repositório de preparação, é uma base de dados relacional onde os dados ficam armazenados até serem carregados para o AD. Nesta fase, os dados são agregados todos numa única base de dados, realiza-se as transformações pretendidas de forma que os dados fiquem prontos para serem carregados no AD. No final do carregamento, os dados são apagados de forma a libertar espaço da base de dados.

Para se conseguir carregar os dados na S*taging Area* criou-se uma aplicação que gere esta extração de dados das diversas fontes existentes e carrega-os na *Staging Area.* Esta aplicação chama-se *Data-Importer.* Toda a gestão da extração de dados é realizada por este projeto, que durante intervalos de tempo recorrentes de dez minutos, faz pesquisas por dados novos nas fontes de dados e, caso encontre informação nova, importa-a para a *Staging Area*. A Figura 35 apresenta como se encontra estruturado o Data-Importer.

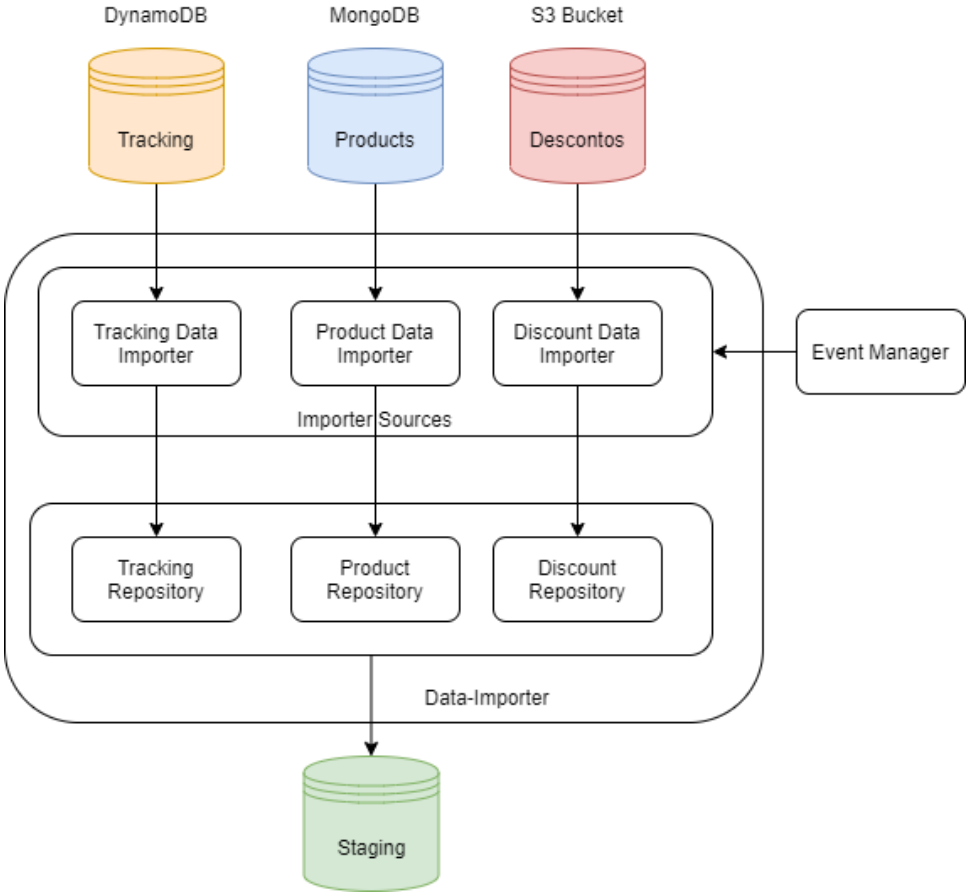

Figura 35 - Design Data-Importer

Foram criados três conetores para aceder às três fontes de dados que utilizam tecnologia diferente. Esses estão incluídos nas *importer-sources,* e ficaram responsáveis por lidar com todos os detalhes das conexões da base de dados e importar os dados para aplicação. Já os repositórios contêm um esquema de como a informação fica guardada na base de dados de preparação e são responsáveis de guardar a informação recebida pelas *importer-sources* nessa mesma base de dados. Também existe um gestor de eventos que é responsável por despoletar uma nova execução para se importar novos dados.

Esta base de dados está dividida em três principais tabelas que contêm os dados das respetivas fontes. Uma tabela de t*racking* chamada "tracking_data", outra de produtos intitulada de "produtcts_data" e, por fim, uma tabela de descontos denominada por "*discount_data*". Dado que todas as bases de dados fonte são não relacionais, definiu-se um esquema na tabela

relacional situada na Staging Area com os campos obrigatórios em todos os registos. Tendo em vista o que foi referido anteriormente, delineou-se os seguintes esquemas para as tabelas: tracking_data e products_data, estando os seus esquemas representados na Figura 36.

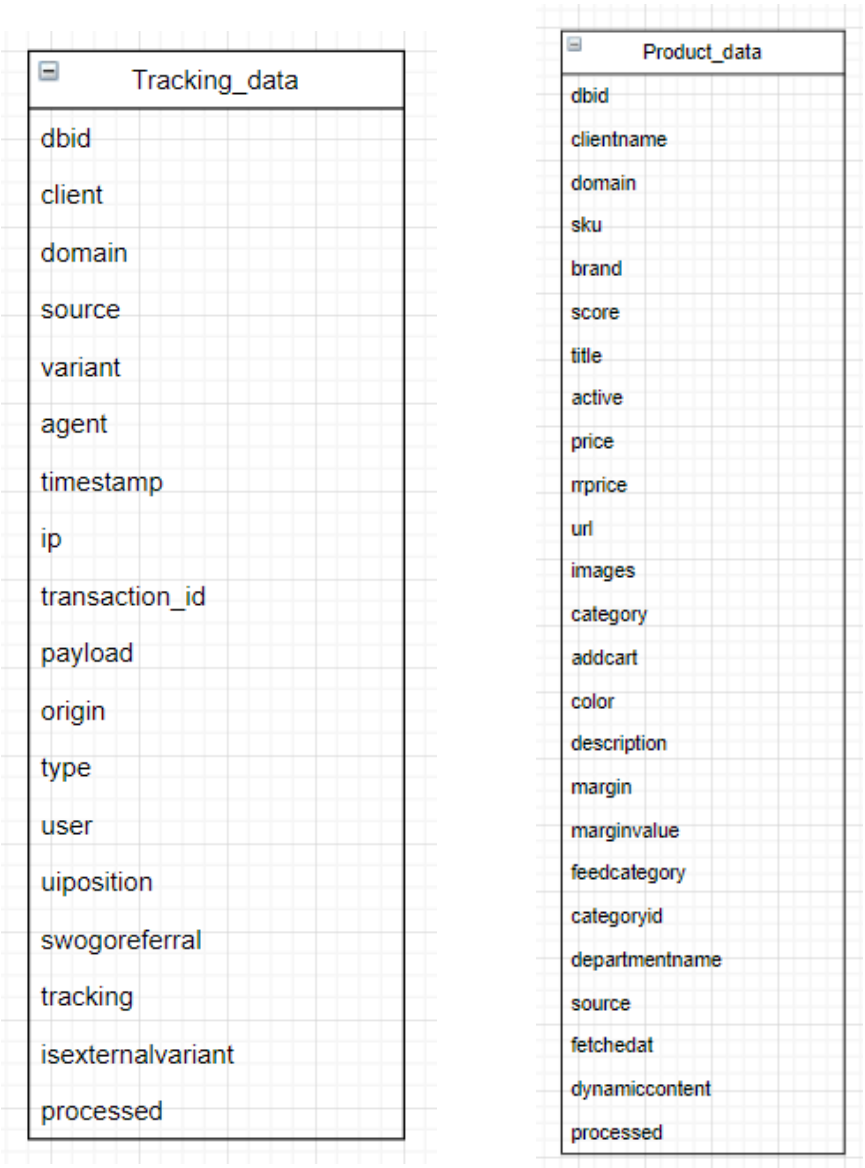

Figura 36 - Esquema das tabelas "tracking_data" e “product_data”

4.3.3 Transformação e Carregamento dos dados

Para se preparar os dados que se encontram na S*taging Area* é necessário transformar os dados antes de carregá-los no AD. Desta forma, criou-se um projeto que deteta a entrada de novos dados na S*taging Area* e prepara-os para o AD, este projeto é chamado de *Data-Exporter.* As suas responsabilidades são recolher os novos dados, transformá-los, carregá-los no AD e, por fim, apagá-los da S*taging Area.*

O Data-Exporter coloca a informação de cada uma das tabelas, previamente preparada, em datasets que auxiliam a construção de tabelas e gráficos, e ajudam a analise da informação. Desta forma, os datasets serão três: sales, products e tracking. Para se conseguir carregá-los tem que se criar três *pipelines,* uma para cada dataset.

A Figura 37 apresenta um esquema de como será realizado todo o processo de transformação e carregamento. Depois da extração de dados pelo Data-Importer, este chama um endpoint no Data-Exporter para que, o mesmo, dê início ao processo de transformação. Quando os dados estiverem prontos para serem carregados, estes são colocados no serviço da nuvem para que seja criado um processo que os carrega para o AD.

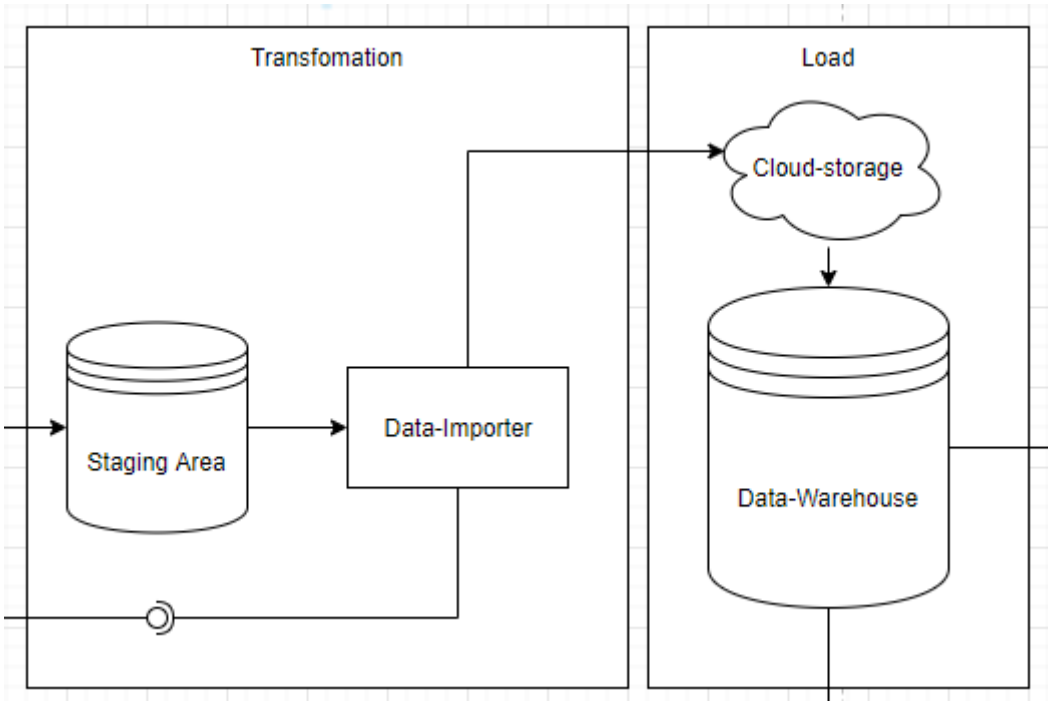

Figura 37 – Transformação e carregamento

4.3.4 Modelação do Armazém de Dados

Na execução de um modelo dimensional é necessário cumprir um conjunto de etapas que são essenciais para a definição do modelo de dados. Primeira etapa, resume-se na definição das áreas de negócio que o armazém de dados guarda. Segunda etapa, selecionar a granularidade

de detalhe dos dados, as tabelas de factos e de dimensões. E, por fim, eleger os atributos que necessitam de manter o histórico. Efetivamente, isto vai ao encontro da metodologia de Kimball aplicada ao AD que segue a arquitetura Data Warehouse Bus Architecture.

Como a principal área de negócio a analisar são as vendas, será criada uma tabela de factos de vendas que com as dimensões de produtos, tracking e descontos, como é possível observar na Figura 38.

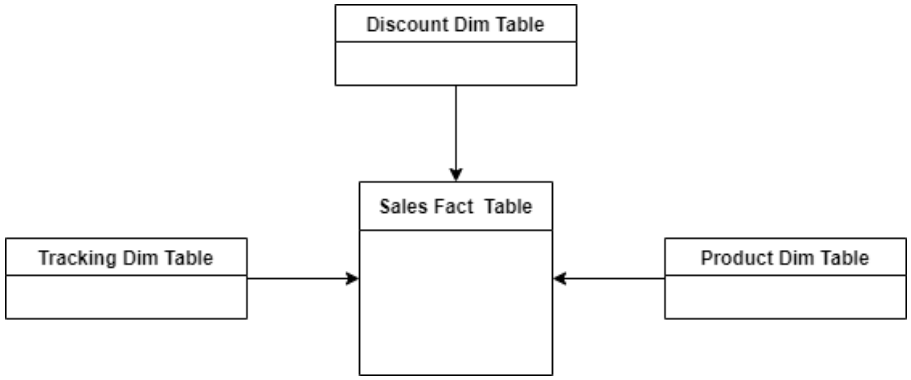

Figura 38 - Esquema Modelação AD

Na Figura 39 pode-se observar a tabela de factos de vendas onde contém toda a informação necessária para uma venda. Também foi adicionada alguma informação dos produtos e de tracking de forma a reduzir o número de *joins* sempre que se fizer alguma consulta nesta tabela.

Field name	Type	Mode
id	STRING	REQUIRED
isPotentialSwogoSale	BOOLEAN	REQUIRED
isSwogoSale	BOOLEAN	REQUIRED
clientName	STRING	REQUIRED
domain	STRING	REQUIRED
variant	STRING	REQUIRED
userID	STRING	REQUIRED
▼ accessories	RECORD	REPEATED
sku	STRING	REQUIRED
title	STRING	NULLABLE
price	FLOAT	REQUIRED
discountID	STRING	REPEATED
bundleDiscountID	STRING	REPEATED
▼ products	RECORD	REPEATED
sku	STRING	REQUIRED
title	STRING	NULLABLE
price	FLOAT	REQUIRED
type	STRING	REQUIRED
location	STRING	NULLABLE
origin	STRING	REPEATED
uiPosition	STRING	REPEATED
discountID	STRING	REPEATED
bundleDiscountID	STRING	REPEATED
▼ trackingEvents	RECORD	REPEATED
dbid	INTEGER	REQUIRED
type	STRING	REQUIRED
hasIssues	BOOLEAN	REQUIRED
isReferral	BOOLEAN	REQUIRED
isBundleSale	BOOLEAN	REQUIRED
isNonSwogo	BOOLEAN	REQUIRED
timestamp	TIMESTAMP	REQUIRED

Figura 39 - Esquema tabela de factos vendas

Para complementar a informação dos produtos inexistente na tabela de vendas também será necessária a criação de uma tabela dimensão de produtos. Cujo seu esquema se encontra representado na Figura 40.

Field name	Type	Mode
dbid	INTEGER	NULLABLE
clientName	STRING	REQUIRED
domain	STRING	REQUIRED
sku	STRING	REQUIRED
brand	STRING	NULLABLE
score	FLOAT	NULLABLE
title	STRING	NULLABLE
active	INTEGER	NULLABLE
price	FLOAT	NULLABLE
rrPrice	FLOAT	NULLABLE
pageUrl	STRING	NULLABLE
description	STRING	NULLABLE
images	STRING	REPEATED
category	STRING	REPEATED
margin	STRING	NULLABLE
marginValue	FLOAT	NULLABLE
departmentName	STRING	NULLABLE
feedCategory	STRING	NULLABLE
categoryID	STRING	NULLABLE
addCart	STRING	NULLABLE
color	STRING	REPEATED
source	STRING	NULLABLE
dynamicContent	STRING	NULLABLE
fetchedAt	TIMESTAMP	NULLABLE

Figura 40 - Esquema da dimensão produtos

Também terá sido criada uma dimensão de *tracking* para completar a informação na tabela de vendas ou então para fazer análises a utilizadores que não chegaram a comprar, mas que navegaram pelas páginas dos clientes, que o seu esquema se encontra representado na Figura 41.

Field name	Type	Mode
dbid	INTEGER	REQUIRED
clientName	STRING	REQUIRED
domain	STRING	REQUIRED
source	STRING	REQUIRED
variant	STRING	REQUIRED
timestamp	TIMESTAMP	REQUIRED
transactionId	STRING	REQUIRED
payload	STRING	REQUIRED
origin	STRING	REQUIRED
type	STRING	REQUIRED
user	STRING	REQUIRED
uiPosition	STRING	NULLABLE
swogoReferral	INTEGER	REQUIRED
isBeingTracked	STRING	NULLABLE

Figura 41 - Esquema da dimensão tracking

Por último, foi construída uma tabela de descontos com a estrutura representada na Figura 42. Esta dimensão não foi implementa no contexto desta dissertação.

Field name	Type	Mode
dbid	INTEGER	REQUIRED
clientName	STRING	REQUIRED
domain	STRING	REQUIRED
discountId	STRING	REQUIRED
message	STRING	REQUIRED
priority	INTEGER	REQUIRED
discountSelector	INTEGER	REQUIRED
startDate	TIMESTAMP	REQUIRED
endDate	TIMESTAMP	REQUIRED
isStrict	BOOLEAN	REQUIRED
applyTo	STRING	REQUIRED
applyStrategy	STRING	NULLABLE
▼ tags	RECORD	REPEATED
field	STRING	REQUIRED
type	STRING	REQUIRED
value	STRING	REQUIRED
hostSelector	STRING	REQUIRED
accessoriesSelector	STRING	REQUIRED
timestamp	TIMESTAMP	REQUIRED

Figura 42 - Esquema da dimensão descontos

5 Implementação da Solução

Neste capítulo apresentar-se-ão todas a etapas da implementação. Neste sentido, recorreu-se à construção de um subcapítulo para cada passo do processo ETL, onde são apresentados e explicados os grafos que representam o fluxo dos dados, bem como os excertos de código, de forma a apresentar partes do que foi desenvolvido.

Por fim, é explicado de que forma foram criados os relatórios que são necessários para a empresa.

5.1 Extração

Como já foi referido, existem três fontes diferentes de dados de onde é extraída informação.

Os eventos de *tracking* estão guardados numa base de dados DynamoDB. Cada cliente tem a sua base de dados e o nome da mesma começa sempre por "sales_" seguido do nome do cliente, "shop", e o seu domínio, "pt", seguindo este exemplo, o nome completo da base dados fica "sales_shoppt". Todos os registos obedecem à mesma estrutura, tendo sempre os seguintes atributos:

- Variant – indentificador que distingue qual é a implementação que o evento de tracking se refere;
- Timestamp - momento em milissegundos em que o evento foi despoletado
- Agent – informação do navegador utilizado;
- Origin - tem a informação de onde foi despoletado o evento, na página de detalhes de um produto, no carrinho ou na página de pagamento;

- Payload - contém toda a informação dos conjuntos que despoletaram este evento;
- SwogoReferral - revela se o evento veio com origem nos conjuntos de produtos ou através de uma referência de um produto;
- TransactionId - contém partes do IP encriptado do utilizador final;
- Type - informa qual é o tipo do evento que foi lançado;
- User - contém um ID do utilizador lançou o evento;

Na Figura 43 consegue-se observar um exemplo de um registo de uma tabela de tracking.

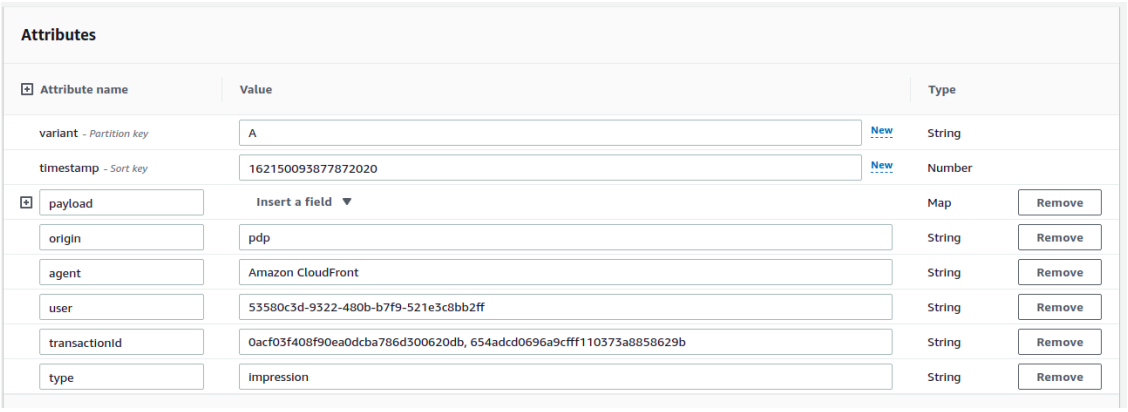

Figura 43 - Registo da tabela de tracking

O DynamoDB obriga a particionar a tabela por um atributo e ordenar por outro, desta forma as tabelas encontram-se particionadas pelo atributo *"variant"* e utilizam o atributo "timestamp" como forma de ordenação. As tabelas também se encontram configuradas com o modo de capacidade de leitura e escrita provisionado, como forma de conter os custos de tabela. Deste modo, obriga a definir quantas unidades de leitura e unidades de escrita são necessárias para ler a quantidade de dados que se encontram armazenados. Quanto maior for a quantidade, maior no número de unidades de leitura terão que ser disponibilizadas para o *Data-Importer* conseguir extrair a informação que necessita. Dado que, por vezes, o sistema não consegue extrair todos os dados, foi criada uma tabela para registar qual o timestamp do último registo que foi extraído. Este processo é executado a cada dez minutos, iniciando por saber qual foi o timestamp do último registo consumido. No final do processo de importação a tabela de meta dados do *tracking* é atualizada, indicando quantas unidades de capacidade de leitura terminou a tabela no final da importação, se já foi executada a importação dos dados completa e qual o tempo que os registos vão permanecer guardados na base de dados.

Já os produtos encontram-se guardados numa base de dados MongoDB e, dado que esta tecnologia não apresenta nenhum entrave no que toca à leitura de dados, estes são extraídos para a base de dados temporária. A extração é incremental e o Data-Importer consulta as datas de alteração dos produtos e extrai todos os produtos que tenham uma data de alteração mais recente que a data que foi consultada, no fim, é guardada a data mais recente dos produtos para ser utilizada na próxima consulta.

Para gerir a execução da importação dos dados é utilizado um gestor de eventos, este é o responsável por saber executar novos processos de importação, saber quais é que estão a ser executados naquele momento e disponibilizar um histórico de todos os processos que foram executados. Este gestor de eventos utilizada uma tecnologia intitulada de celery[3]. Na Figura 44 é apresentado um bloco do código fonte que inicia a execução dos agendamentos. Este cria um processo que fica sempre ativo e que aguarda que uma tarefa esteja concluída para iniciar uma próxima.

```
if __name__ == "__main__":
    logger.info("Setting up base schedules...")
    setup_all_schedules()
    logger.info("Set up complete. Scheduler main loop initialized.")
    while True:
        logger.info("Executing pending tasks...")
        schedule.run_pending()
        logger.info("Pending tasks execution completed.")
        date = schedule.next_run()
        current = datetime.datetime.now().timestamp()  # seconds, with .microseconds on decimals
        sleeping_time = 1
        if date is not None:
            next = date.timestamp()  # seconds, with .microseconds on decimals
            sleeping_time = next - current
        sleeping_time = max(sleeping_time, 0.1)
        logger.info("Main loop will wait for the next task and sleep for %.4f seconds." % sleeping_time)
        time.sleep(sleeping_time)
```

Figura 44 - Bloco de código que inicia o celery

Na Figura 45 é apresentado um bloco de código que mostra como é iniciado o processo de importação dos eventos de *tracking.*

```
@celery_app.task(bind=True)
def import_tracking_data(self, sales_table_name, pipeline_name):
    # TO-DO: if it makes sense, move "valid_client_names" fetching to a specific method and invoke here
    # Get the list of valid client names
    mongo_list = product_data_importer.list_clients_databases()
    valid_client_names = {}
    for elem in mongo_list:
        lelem = elem.lower()
        if lelem not in valid_client_names:
            valid_client_names[lelem] = elem

    key = f"{TRACKING_TASKS_QUEUE}_tracking_{sales_table_name}"

    try:
        until_timestamp = TRACKING_MAX_TIMESTAMP if TRACKING_MAX_TIMESTAMP > 0 else None
        with TaskLockManager(key=key, timeout=TRACKING_TASK_TIMEOUT) as lock:
            if lock:
                tracking_data_importer.fetch_tracking_data_for_table(sales_table_name, valid_client_names,
                                                                     pipeline_name, until_timestamp=until_timestamp)
                return f"Completed '{sales_table_name}' tracking"
            else:
                print(f"Could not get lock for {sales_table_name}")
                return f"Skipped: lock-acquisition-failed[{key}]"
    except Exception as e:
        db_sys.sess.rollback()
        raise e
    finally:
```

Figura 45 - Bloco de código que inicia importação de eventos de tracking

[3] Celery é um mecanismo de fila de tarefas para distribuir o trabalho por diferentes processos ou máquinas (Celery, 2021)

5.2 Transformação

Após os dados se encontrarem na base de dados de preparação, o Data-Importer chama um endpoint no Data-Exporter para este iniciar o processo de transformação e carregamento dos dados para o Big Query. O Data-Exporter foi concebido utilizando Java de forma a suportar a tecnologia Apache Beam, que é a maneira que é utilizada para criar processos no Dataflow. Quando o endpoint é chamado, este inicia o primeiro processo no Dataflow de transformação dos dados, que no caso dos dados de tracking, este processo lê os registos da base de dados, transforma-os em ficheiros no formato AVRO[4] e coloca-os no Google Cloud Storage (GCS), como pode ser observado na Figura 46.

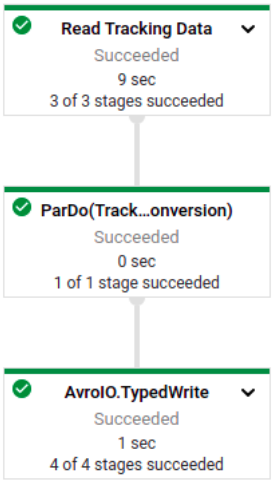

Figura 46 - Grafo das transformações de tracking

No que toca às vendas, estas podem ser compostas por quatro tipos de eventos de *tracking: impressions, bundleList, basketList e payment.* Cada um destes eventos contém a informação respetiva aos produtos que foram carregados na página do cliente e na qual o utilizador teve uma interação com os mesmos. A combinação destes eventos são a forma da empresa detetar as vendas realizadas pelos clientes através dos seus *bundles*. Uma *impression* é lançada sempre que os *bundles* da empresa são apresentados ao cliente. A base de dados recebe um bundleList sempre que o utilizador pressiona o botão de adicionar ao carrinho que existe nos *bundles.* O evento de basketList acontece quando a página do carrinho do cliente é carregada. Por fim, o evento *payment* é lançado quando o utilizador chega à página de confirmação de pagamento.

Para se criar uma venda corre um processo no dataflow que vai agrupar estes eventos por utilizador, nome do cliente, domínio e variante, e verificar se existe algum payment no conjunto de dados, caso haja, todos os eventos do utilizador serão processados de forma a gerar todos

[4] AVRO é um formato de ficheiro comprimido que permite definir uma estrutura. Este é o tipo de ficheiro recomendado pela Google, pois o Big Query consegue ler vários ficheiros paralelamente (Google, 2021)

os dados necessários para criar um registo de venda. No fim de transformar os eventos de tracking numa venda é extraído o número da encomenda para ser colocado numa tabela à parte, chamada sales_order, pois este tem que ser apagado ao fim de um ano. Caso exista algum erro com a venda, como por exemplo, a falta de um campo obrigatório, esta venda será guardada noutra tabela intitulada por sales_errors. Todas estas vendas no final da transformação são guardadas no GCS em ficheiros do tipo AVRO, para poderem ser carregadas em lotes de forma paralela. Na Figura 47 é possível visualizar um grafo com as transformações referidas.

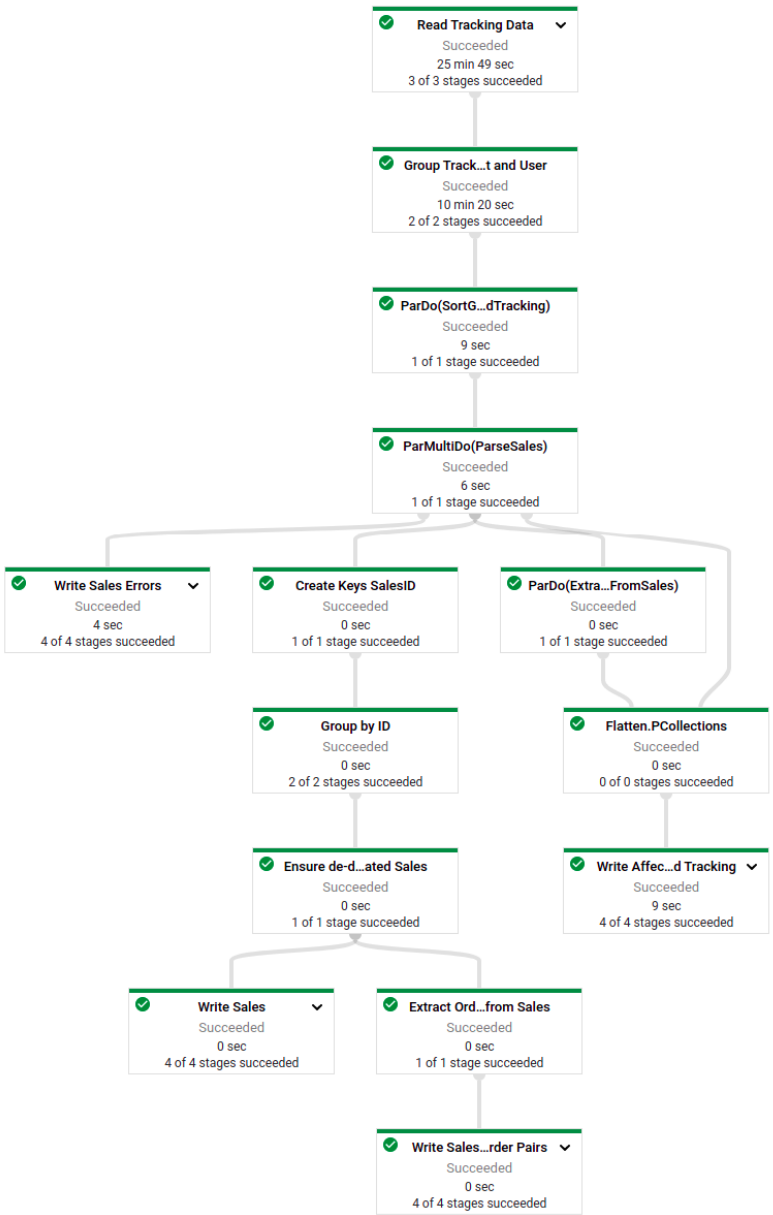

Figura 47 - Grafo transformações vendas

No que toca às transformações executadas aos produtos este agrupa os produtos por cliente e domínio, verifica a unicidade do campo SKU por cliente e, também os converte em ficheiros do tipo AVRO, como se pode verificar na Figura 48.

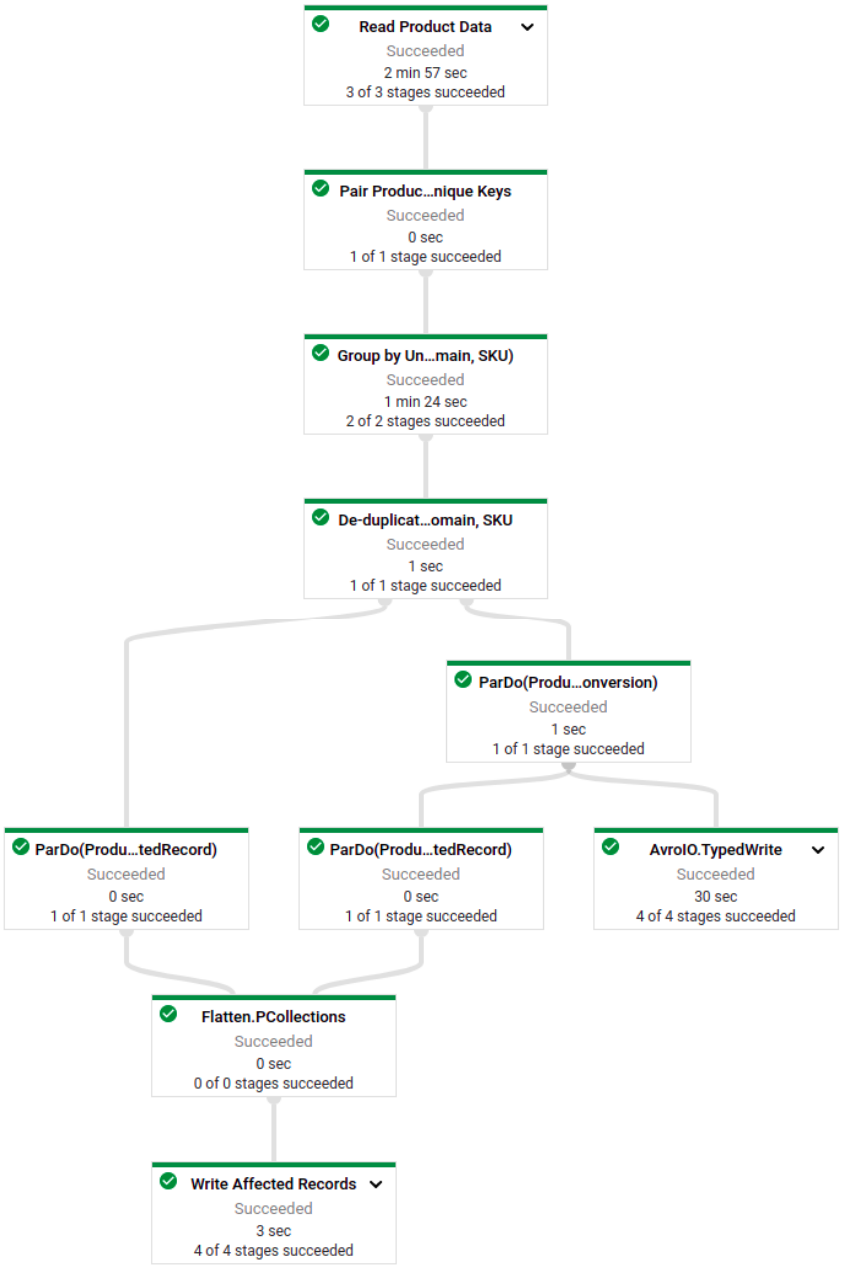

Figura 48 - Grafo transformações produtos

5.3 Carregamento

O processo de carregamento de cada umas das pipelines é executado sempre após o término do processo de transformação. Este processo tem como principal função carregar os ficheiros AVRO, previamente criados no GCS pelos processos de transformação, no BigQuery e, posteriormente, apagá-los e tem sempre o nome da pipeline adicionando o sufixo *loadclean.*

O carregamento dos dados para todas as tabelas é feito sempre para uma tabela stash. Esta é uma tabela auxiliar, que permite que os dados sejam para lá carregados antes de serem carregados para a tabela principal e, assim, garante-se que não exista nenhum problema com o carregamento dos dados evitando-se dados corrompidos. Caso seja detetado algum erro no carregamento é executado o processo de *rollback*, que atualiza na base de dados de preparação os registos que tinham sido consumidos pela pipeline e coloca o atributo *processed* a falso, para que numa nova iteração da *pipeline* ela os detete como novos registos. Na Figura 49 pode-se ver o código fonte deste processo de *rollback.*

```
@ProcessElement
public void processElement(ProcessContext ctx) throws Exception {
    List<Long> input = ctx.element();

    //Skip if there are no ids to update:
    if (!input.isEmpty()) {

        Connection con = connectionFactory.getConnection();
        con.setAutoCommit(false);
        try {

            int chunkSize = 100000;
            List<String> queryValues = Utils.limitQueryListValues(input, chunkSize);
            for (int i = 0; i < queryValues.size(); i++) {
                String queryVal = queryValues.get(i);
                //Auto-commit is true by default.
                String queryStatement =
                        String.format( s: "UPDATE %s SET processed = false WHERE dbid in %s", tableName, queryVal);
                Logger.getLogger(LOGGER_NAME).info(String.format( s: "Executing query %d of %d (%s)", i,
                        queryValues.size(), queryStatement));
                PreparedStatement stat = con.prepareStatement(queryStatement);
                stat.execute();
            }
            con.commit();
            con.close();
        } catch (SQLException ex) {
            con.rollback();
            con.close();
            Logger.getLogger(LOGGER_NAME).log(Level.SEVERE, String.format( s: "unable to rollback \"signaled records as updated\"" +
                    " in temporary batch database: %s", ex.getMessage()));
            throw ex;
        }
    } else {
        Logger.getLogger(LOGGER_NAME).log(Level.WARNING, s: "No rows will be updated, as there are no ids.");
    }
}
```

Figura 49 - Bloco de código do processo de *rollback*

Caso o carregamento para a tabela *stash* seja executado com sucesso, de seguida é executado o processo de transferências dos dados para a tabela principal, que anexa os dados existentes na tabela *stash* à tabela principal. Posteriormente, são apagados todos os ficheiros avro do GCS que estão presentes na pasta relativa à *pipeline* que executou. A Figura 50 representa a *pipeline de* produtos, sendo que esta se diferencia das outras duas no ponto de criar os dados com

*Slow*ly Changing Dimensions (SCD). Para os produtos necessitou-se de manter o histórico dos atributos *score*, *active, price, rrprice, margin, departmanteName, feedCategory, categoryID, addCart e category,* pois estes atributos podem ser alterados ao longo do tempo e desta forma, aplicou-se uma SCD do tipo 4 na qual a pipeline deteta se o produto já existe através dos atributos *clientName, sku e domain,* e, caso exista, copia os atributos do produto que se tem que manter histórico e cria uma nova entrada com esses atributos e coloca a data em que essa alteração foi realizada no campo *changed_at.*

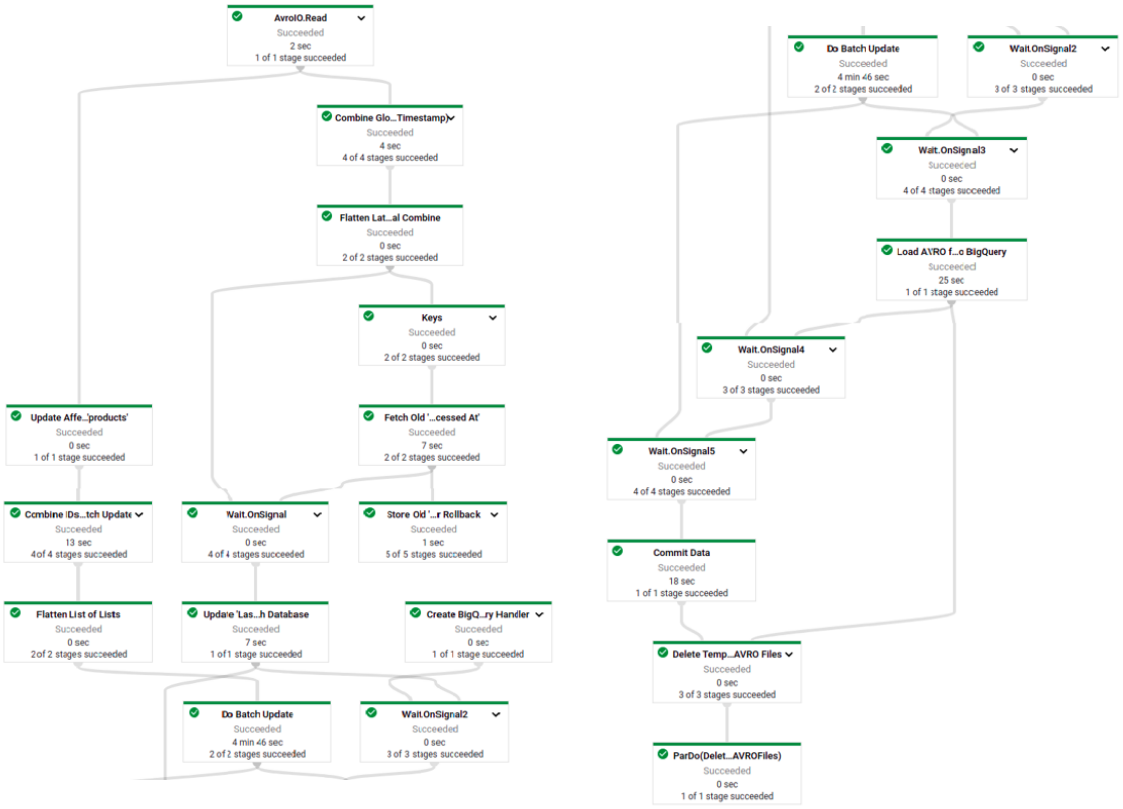

Figura 50 - Grafo de carregamento produtos

5.4 Relatórios

Atualmente existem quatro tipos de relatórios: *raw sales*, *sales, category performance e cross sell.* Estes são solicitados numa ferramenta interna e são enviados por email. Existem duas *lambdas* para gerar os relatórios atuais, uma que vai diariamente recolher os dados às tabelas do dynamoDB e processa as vendas e coloca-as em ficheiros no S3 bucket. A segunda lambda só é executada no momento em que o utilizador solicita os relatórios, esta recolhe a informação previamente colocada nos ficheiros no S3 e gera um relatório com essa informação.

Desta forma para se alterar o processo de geração de relatórios teve que se criar uma API chamada data-api, que é responsável por executar todas as consultas no BigQuery, ficando

também responsável de tratar os dados antes de os enviar. Por motivos de confidencialidade não foi possível apresentar exemplos dos relatórios criados.

5.4.1 Raw Sales

O relatório de *raw sales,* apresenta todas as vendas discriminadas de cada cliente. Com este relatório pode-se consultar que produtos foram vendidos em cada venda, se a venda foi feita através da empresa, entre outras informações.

Para obter a informação foi criado um *endpoint* do tipo GET. Ao fazer um pedido para o enderço '*/sales/raw'* com os parâmetros *clientName, clientDomain, startDate, endDate, rowCount, clientVariant e onlySwogoSales,* só o rowCount é que não é um parâmetro obrigatório. Um exemplo do endereço completo seria:

'/sales/raw?clientName=shop&clientDomain=Pt&startDate=2021-01-01&endDate=2021-01-21&rowCount=2000&clientVariant=A&onlySwogoSales=true'

A API quando recebe o pedido no sistema executa a consulta apresentada na Figura 51 e retorna informação em JSON com o formato indicado na Figura 52.

```
with sales as (select S.*,PROD
    from sales.main S,UNNEST(S.products) PROD
    where S.clientName = @clientName
        AND S.domain = @clientDomain
        AND (@allVariants OR S.variant = @clientVariant)
        AND S.timestamp BETWEEN @startDate AND @endDate
        AND (isBundleSale = @onlySwogoSales OR NOT(@onlySwogoSales))
        AND ((type <> 'clt' AND @onlySwogoSales) OR NOT(@onlySwogoSales))
),
prod as (select P.sku,P.clientName,P.domain,P.title,P.brand,P.pageUrl,P.category as category,round(P.rrPrice, 2) as rrPrice,P.margin,P.feedCategory as feedCategory,
        P.departmentName as departmentName, P.categoryID as categoryID
    from (select dbid, clientName, domain,title,sku,rrPrice,brand,pageUrl,categoryID,departmentName,feedCategory,category,margin
            from products.main
            where clientName = @clientName and domain = @clientDomain) as P)

select STRING(S.timestamp) as time,S.variant,S.id as tid,SO.orderID as transaction,if(S.PROD.type = 'acc', 1, 0) as swogoSale, IF(S.PROD.type = 'hst', 1, 0) AS host,
    ifnull(PFINAL.sku, S.PROD.sku) AS id,PFINAL.category AS category,ROUND(PROD.price, 2) AS price,PFINAL.rrPrice AS oldPrice,ifnull(PFINAL.margin, '') AS margin,
    ifnull(PFINAL.title, ifnull(PROD.title, '')) AS title,ifnull(PFINAL.brand, '') AS brand,
    if(CHAR_LENGTH(S.PROD.location) > 0,
        concat(REGEXP_REPLACE(REPLACE (S.PROD.location, "List", ""),"(basket(Mobile)?)(Hero|Popup)","\\1"), if(S.isReferral, '-link', '')),
        'not bundle'
    ) AS origin,
    ARRAY_LENGTH(S.accessories) AS numberSwogoItemsInTransaction,
    CONCAT(S.clientName, S.domain) AS source,
    PFINAL.pageUrl AS url,
    PFINAL.categoryID as categoryID,
    IF((PROD.discountID IS NOT NULL AND ARRAY_LENGTH(PROD.discountID) > 0),1,0
    ) AS swogoDiscounted,
    PFINAL.departmentName AS departmentName,
    PFINAL.feedCategory AS feedCategory
FROM sales as S
    left join sales.sales_orders SO on SO.saleID = S.id
    left join prod AS PFINAL ON PFINAL.sku = S.PROD.sku
    and S.clientName = PFINAL.clientName
    AND S.domain = PFINAL.domain
order by S.timestamp ASC;
```

Figura 51 - Consulta Raw Sales

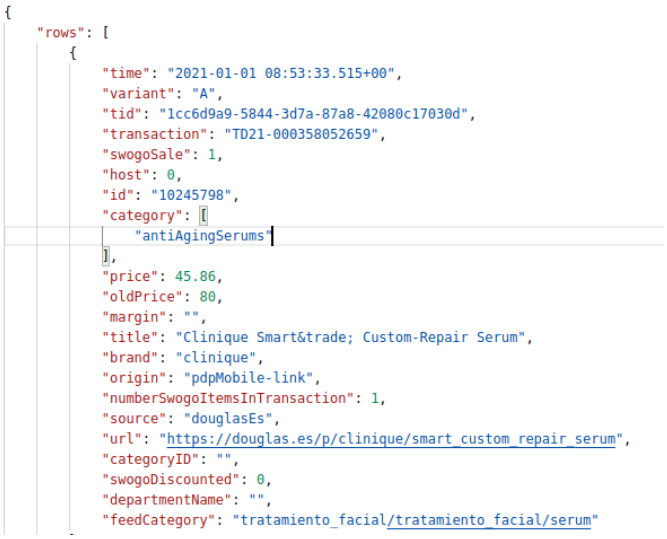

Figura 52 - JSON de retorno Raw Sales

5.4.2 Sales

O relatório de *sales* apresenta todas as categorias vendidas como produto principal por ordem de número de artigos vendidos e a sua receita, e, também, apresenta as quantidades e receitas das categorias vendidas como acessórios. Este também apresenta a receita e a quantidade total dos produtos que foram vendidos como produtos principais e também dos produtos que foram vendidos como acessórios dos produtos principais. Adicionalmente, apresenta uma contagem do número de conjuntos que foram adicionados ao carrinho, do número de produtos que foram adicionados ao carrinho que foram referidos nos conjuntos da empresa e por fim o número de encomendas feitas com acessórios dos conjuntos recomendados.

Dada a complexidade do relatório, foram criados dois *endpoints* do tipo GET, um para retornar os dados discriminados por categoria e outro para retornar os totais necessários para construir o relatório que foi previamente descrito, também retorna as vendas feitas só com um acessório, com dois acessórios e por aí em diante. O primeiro *endpoint* está presente com o enderço *'/sales'* com os parâmetros *clientName, clientDomain, startDate, endDate, rowCount, clientVariant e onlySwogoSales* só o rowCount é que não é um parâmetro obrigatório. Um exemplo do endereço completo seria:

'/sales/raw?clientName=shop&clientDomain=Pt&startDate=2021-01-01&endDate=2021-01-21&rowCount=2000&clientVariant=A&onlySwogoSales=true'

Este retorna os valores das categorias discriminados. No momento que este *endpoint* recebe um pedido no sistema executa a consulta e retorna informação em JSON com o formato indicado na Figura 53.

```
{
  "rows": [
    {
      "category": "antiAgingCreams",
      "totalCount": 261,
      "totalRevenue": 13223.36,
      "accessoryCount": 155,
      "accessoryRevenue": 7900.57,
      "avgAccessoryPrice": 50.97,
      "hostCount": 106,
      "hostRevenue": 5322.79,
      "avgHostPrice": 50.22
    },
    {
      "category": "facialMasks",
      "totalCount": 227,
      "totalRevenue": 1041.88,
      "accessoryCount": 164,
      "accessoryRevenue": 786.79,
      "avgAccessoryPrice": 4.8,
      "hostCount": 63,
      "hostRevenue": 255.09,
      "avgHostPrice": 4.05
    },
    {
      "category": "antiAgingEyeCreams",
      "totalCount": 205,
      "totalRevenue": 7181.19,
      "accessoryCount": 174,
      "accessoryRevenue": 6172.35,
      "avgAccessoryPrice": 35.47,
      "hostCount": 31,
      "hostRevenue": 1008.84,
      "avgHostPrice": 32.54
    },
```


Figura 53 – JSON resposta Sales

Já o segundo endpoint, que irá retornar os valores dos totais por produtos principais e acessórios, o seu endereço é '*/sales/metrics*' que retornará um JSON representado na Figura 54.


```
  "rows": [
    {
      "totals": {
        "AOV": 53.72,
        "AOS": 2.54,
        "bundlesValue": 21354440.95,
        "bundlesSold": 397539,
        "itemsSold": 1011495,
        "swogoSalesCount": 13871,
        "accessoryTotalCount": 19019,
        "accessoryTotalRevenue": 451055.72,
        "hostTotalCount": 17810,
        "hostTotalRevenue": 452517.18,
        "bundlesATC": 75961,
        "referralATC": 356900
      },
      "addons": [
        {
          "addonName": "0 Add-on",
          "addonOrders": 383668,
          "addonPercentage": 96.51
        },
        {
          "addonName": "1 Add-on",
          "addonOrders": 10482,
          "addonPercentage": 2.64
        },
        {
          "addonName": "2 Add-on",
          "addonOrders": 2381,
          "addonPercentage": 0.6
        },
```


Figura 54 - JSON resposta Sales Metrics

5.4.3 Category Performance

O objetivo do relatório de *category performance* é apresentar todas as categorias que foram vendidas em conjuntos e as suas quantidades, ou seja, apresenta sempre a categoria principal com os seus acessórios e os dados relativos a essas vendas. Criou-se um *endpoint* do tipo POST, pois este terá que receber uma estrutura de dados com as categorias principais e seus acessórios. O endereço é '/sales/category-performance' e recebe no corpo do pedido um JSON com a estrutura exemplo representada na Figura 55.


```
{ "matchingConfig": [
        {"host":"womanShampoos1", "accessories": ["womanHairConditioners", "womanHairMasks", "hairTreatmentOils", "hairTreatments", "hairProtectors"]},
        {"host": "stickBlush", "accessories": ["powderFoundation", "liquidFoundation", "lipsticks, womanHairConditioners"]},
        {"host": "facialCleansers","accessories": ["facialScrubs", "makeupRemovers", "facialMoisturizers", "antiAgingCreams", "facialMasks"]}
    ],
  "clientName":"shop",
  "clientDomain": "Es",
  "startDate": "2020-01-01",
  "endDate":"2021-01-20",
  "rowCount": 10000,
  "clientVariant": "A",
  "withAccessoryData": true
}
```


Figura 55 - JSON corpo pedido Category Performance

Dado não ser possível obter a informação estruturada com uma consulta ao Big Query, criou-se um agregador que faz várias consultas, e consolida os dados do lado da API, criando uma estrutura de dados igual à representada na Figura 56 para devolver como resposta.


```
"rows": [
    {
        "hostCategoryName": "womanShampoos1",
        "totalAttachedAccessoryCount": 0,
        "clientAttachedAccessoryCount": 0,
        "swogoAttachedAccessoryCount": 0,
        "totalAttachedAccessoryRevenue": 0,
        "clientAttachedAccessoryRevenue": 0,
        "swogoAttachedAccessoryRevenue": 0,
        "swogoHostCount": 0,
        "swogoHostRevenue": 0,
        "clientHostCount": 0,
        "clientHostRevenue": 0,
        "totalHostCount": 0,
        "totalHostRevenue": 0,
        "accessoryCategories": [
            {
                "categoryName": "womanHairConditioners",
                "totalAttachedAccessoryCount": 0,
                "swogoAttachedAccessoryCount": 0,
                "clientAttachedAccessoryCount": 0,
                "totalAttachedAccessoryRevenue": 0,
                "swogoAttachedAccessoryRevenue": 0,
                "clientAttachedAccessoryRevenue": 0,
                "clientAccessoryCountPercentage": 0,
                "swogoAccessoryCountPercentage": 0,
                "clientAccessoryRevenuePercentage": 0,
                "swogoAccessoryRevenuePercentage": 0,
                "averageAccessoryPrice": 0,
                "swogoAverageAccessoryPrice": 0,
                "clientAverageAccessoryPrice": 0,
                "averageAccessoryPriceOfHostPricePercentage": 0,
                "swogoAverageAccessoryPriceOfHostPricePercentage": 0,
                "clientAverageAccessoryPriceOfHostPricePercentage": 0,
```


Figura 56 - JSON resposta Category Performance

5.4.4 Cross Selling

O relatório de cross selling retorna diversas informações sobre as categorias de produtos que foram vendidas em conjunto, o que dará uma grande visibilidade de oportunidades de criar conjuntos que possam estar a ser perdidas pela empresa. Para retornar informação necessária para se criar este relatório criou-se um *endpoint* do tipo GET que recebe como parâmetros *clientName, clientDomain, startDate, endDate, rowCount, clientVariant, onlySwogoSales, clientVariant* e *aggregationNumber.* Este retorna um JSON com a estrutura que está exemplificada na


```
{
    "rows": [
        {
            "category": "womanFragrances",
            "categoryAggregation": "womanFragrances",
            "amount": 2517,
            "sumCategoryPrice": 101962.55,
            "sumCategoryAggregationPrice": 98206.48
        },
        {
            "category": "facialMoisturizers",
            "categoryAggregation": "facialMoisturizers",
            "amount": 1980,
            "sumCategoryPrice": 22587.62,
            "sumCategoryAggregationPrice": 24446.21
        },
        {
            "category": "manFragrances",
            "categoryAggregation": "manFragrances",
            "amount": 1440,
            "sumCategoryPrice": 60838.07,
            "sumCategoryAggregationPrice": 59040.04
        },
        {
            "category": "manFragrances",
            "categoryAggregation": "womanFragrances",
            "amount": 1132,
            "sumCategoryPrice": 51785.09,
            "sumCategoryAggregationPrice": 50914.15
        },
        {
            "category": "womanFragrances",
            "categoryAggregation": "manFragrances",
            "amount": 1132,
            "sumCategoryPrice": 50914.15,
```


Figura 57 - JSON de resposta cross selling

Devido à complexa estrutura que o JSON de reposta apresenta, este relatório também seguiu a mesma abordagem que o relatório de Category Performance, criando o agregador e de forma iterativa executa consultas no Big Query até ter todos os dados necessários.

5.5 Testes

Ao longo do desenvolvimento do projeto foram criados diversos tipos de testes, de forma a garantir o correto funcionamento da aplicação desenvolvida. Desta forma foram criados testes unitários para garantir que os métodos retornam o que é pretendido, testes funcionais para garantir que todo o fluxo dos dados na aplicação é o correto e testes de integração para garantir que as diversas componentes dos sistemas comunicam corretamente entre si.

Sendo assim, os próximos subcapítulos irão detalhar que testes foram criados para cada uma das aplicações desenvolvidas: *Data-Importer, Data-Exporter e Data-Api.*

5.5.1 Data-importer

Dado que nesta aplicação as operações são basicamente de extração e importação, o foco neste projeto foi criar um grande número de testes de integração. Os testes unitários foram criados para validar lógica, como é possível observar na Figura 58 que valida todas as restrições impostas para o SKU de um produto.

```
def test_is_valid_sku(self):
    self.assertTrue(is_valid_sku("1230123"))
    self.assertTrue(is_valid_sku("SL1230123"))
    self.assertTrue(is_valid_sku("MM_1230123"))
    self.assertTrue(is_valid_sku("MY SPECIAL PRODUCT ID"))
    with self.assertRaises(Exception):
        is_valid_sku(None)
    with self.assertRaises(Exception):
        is_valid_sku("Error")
    with self.assertRaises(Exception):
        is_valid_sku("SL_ERROR")
    with self.assertRaises(Exception):
        is_valid_sku("{'error':'Request returned status code 401'}")
```

Figura 58 – Teste unitário Data-Importer

Já os testes de integração vão inserir dados de teste numa base de dados local com as mesmas configurações que o DynamoDB e o MongoDB e validar se os dados que são inseridos na base de dados de preparação, também com uma imagem local, são os pretendidos. Na Figura 59 encontra-se um exemplo de um teste de integração que insere dados de *tracking* e verifica se ficaram guardados na base de dados local e posteriormente testa se os dados foram apagados.

```
def test_delete_obsolete_tracking(self):
    table_name = "sales_dummyptlol"
    expected_objects = 1000
    for i in range(expected_objects):
        sess.add(TrackingDataSales(dbid=str(uuid.uuid4()), client_name="dummy", domain="PtLol", timestamp=0,
                                   processed=True))
    metadata, exists = get_or_default_metadata(table_name, PIPELINE)
    sess.add(PipelineProcessMetadata(pipeline_name=PIPELINE, data_table="tracking_data_sales", source=table_name,
                                     last_processed_at=time.time() * 1000))
    sess.add(metadata)
    sess.commit()
    self.assertEqual(expected_objects, len(sess.query(TrackingDataSales).all()))
    delete_obsolete_tracking()
    count = len(sess.query(TrackingDataSales).all())
    self.assertEqual(0, count, f"Delete obsolete tracking data should have deleted all objects, but"
                               f" data base still has {count}")
```

Figura 59 - Teste de integração do Data-Importer

5.5.2 Data-Exporter

No Data-Exporter, de igual forma, foram criados testes unitários para validar a lógica dos métodos e também foram criados bastantes testes funcionais de forma a garantir o correto fluxo de dados. Na Figura 60está representado um exemplo de um teste unitário nesta aplicação.

```
@Test
public void generateUniqueID_EqualID() throws IOException, InterruptedException {
    PayBlock payBlock = new PayBlock();
    payBlock.clientName = "test";
    payBlock.domain = "pt";
    payBlock.orderID = "123";
    payBlock.userID = "anon.123";
    payBlock.variant = "A";
    payBlock.lastPaymentTimestamp = BigDecimal.valueOf(1234567);
    Sale s1 = new Sale(payBlock);
    Thread.sleep( l: 500);
    //No matter how long it takes, it should always generate the same id from the same data.
    Sale s2 = new Sale(payBlock);
    assertEquals(s1.id, s2.id);
}
```

Figura 60 - Teste unitário Data-Exporter

Os testes funcionais desta aplicação são executados utilizando uma base de dados a correr no ambiente de staging, que simula a base de dados de preparação, inserindo lá dados através de scripts SQL com diversos cenários de testes. Após essa inserção é executado o teste e este corre através do dataflow em produção que o resultado é inserido num dataset no Big Query chamado deploy_test. Tem exatamente a mesma estrutura que os datasets que são utilizados em produção. No final do teste é executada uma consulta no Big Query para verificar se os dados foram inseridos corretamente.

5.5.3 Data-api

Na data-api, para garantir que os dados são retornados corretamente foram criados bastantes testes unitários, devido ao facto de ter muita lógica de negócio associada e conseguiu-se uma cobertura de 100%, como se pode observar na

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	100	100	100	100	
src	100	100	100	100	
app.controller.ts	100	100	100	100	
app.service.ts	100	100	100	100	
src/bigquery	100	100	100	100	
bigquery.service.ts	100	100	100	100	
utils.ts	100	100	100	100	
src/sales	100	100	100	100	
queries.ts	100	100	100	100	
sales.controller.ts	100	100	100	100	
sales.service.ts	100	100	100	100	
src/sales/agregator	100	100	100	100	
base.aggregator.ts	100	100	100	100	
category.performance.aggregator.ts	100	100	100	100	
cross.sell.aggregator.ts	100	100	100	100	

Figura 61 - Cobertura teste unitários Data-Api

Também foram criados testes de integração para garantir que a informação que é retornada dos *enpoints* é a correta e que também apresenta uma estrutura correta.

5.6 Implantação da solução

De forma a possibilitar que os o sistema corra num ambiente em produção, necessitou-se de criar uma estratégia de implantação da solução. Deste modo, os projetos foram desenvolvidos em contentores de Docker.

As soluções apresentadas, após estar concretizadas e testadas, foram colocadas numa máquina virtual disponibilizada na nuvem pela AWS. Esta máquina é completamente gerida pela plataforma, na qual só é necessário escolher o processador e armazenamento, rede e sistema operativo (AWS, 2021). Este serviço facilita os ajustes de configurações conforme o que vai sendo necessário, como por exemplo aumentar ou diminuir memória.

Já a base de dados de preparação ficou hospedada numa Amazon Relational Database Service. Este serviço é muito idêntico ao anterior, mas é mais apropriado para bases de dados, permitindo também a aplicação de backups (AWS, 2021).

Posto o que foi citado anteriormente, os contentores de Docker das aplicações, contêm as informações que são necessárias para a aplicação estar a ser executada como drivers e variáveis de ambiente. A Figura 62 apresenta um exemplo da imagem de Docker que permite os contentores sere executados.

```yaml
#docker-compose.yml
version: '3'

services:
  database:
    image: "postgres"
    environment:
      POSTGRES_USER:
      POSTGRES_PASSWORD:
      POSTGRES_DB:
    ports:
      - "5432:5432"
    volumes:
      - database-data:/var/lib/postgresql/data/ # persist data even if the container is shut down
  celery-broker:
    image: "redis"
    ports:
      - "6379:6379"
  flower:
    image: mher/flower:0.9.5
    command: [ "flower", "--broker=redis://celery-broker" ]
    ports:
      - 5555:5555
    depends_on:
      - celery-broker
  mongodatabase:
    image: mongo:4.0
    ports:
      - 27017:27017
    restart: unless-stopped

volumes:
  database-data: # named volumes can be managed easier using docker-compose
```

Figura 62 - Imagem de Docker do Data-Importer

Na Figura 63 pode-se observar todos os contentores de Docker que estão a ser executados na máquina EC2.

```
CONTAINER ID        IMAGE                              COMMAND                  CREATED             STATUS
1adcfcf67bac        data-pipeline_data-importer        "./start_scheduler.sh"   14 minutes ago      Up 14 minutes
a601849dc084        data-pipeline_flower               "flower /bin/sh -c '…"   14 minutes ago      Up 14 minutes
f41c3ca53039        data-pipeline_celery-worker        "./start_worker.sh"      14 minutes ago      Up 14 minutes
4fd80fe68286        data-pipeline_data-exporter        "/usr/local/bin/mvn-…"   14 minutes ago      Up 14 minutes
```

Figura 63 - Contentores de Docker em produção

6 Avaliação da Solução

Este capítulo tem o objetivo de avaliar o sistema de Business Intelligence que se dedica à análise de dados provenientes dos clientes da empresa. Para tal, recorre-se a um conjunto de grandezas para avaliar a viabilidade da solução implementada. A solução foi avaliada através de inquéritos de satisfação, onde requisitos como a usabilidade e desempenho são analisados.

6.1 Metodologia de Avaliação

A solução desenvolvida, foi avaliada com recurso a dois inquéritos de satisfação, realizados a onze colaboradores da empresa, os que trabalham diretamente com a aplicação desenvolvida. O objetivo é saber a opinião de quem utiliza a ferramenta diariamente de forma a perceber quais os pontos de melhoria. Melhor do que ninguém, os profissionais olham para os relatórios e analisam métricas para apresentarem aos seus clientes, conseguem ter uma maior capacidade analítica acerca da utilidade deste tipo de aplicação, dado as circunstâncias que enfrentam no dia a dia. O inquérito foi disponibilizado e realizado online, dado ser uma forma rápida de fazer chegar os mesmos aos profissionais e também se torna intuitiva a forma como a informação é apresentada. Aliás, é uma forma de aproveitar o potencial de ferramentas online que permitem de imediato obter a avaliação acerca das respostas obtidas. No inquérito de satisfação foi utilizado um conjunto de afirmações as quais foram classificadas, segundo a escala Likert, para especificar o nível de concordância com as mesmas. A escala de Likert mede a satisfação do utilizador segundo níveis de classificação. Na solução implementada foram aplicados cinco níveis de satisfação, de forma a ser possível medir o comportamento dos utilizadores relativamente à concordância com as afirmações, a avaliação do serviço prestado e a frequência de utilização. Nos inquéritos de satisfação foi utilizada a escala que se encontra na Tabela 7.

Tabela 7 - Escala de Likert

Escala	Descrição
1	Discordo totalmente
2	Discordo
3	Não concordo, nem discordo
4	Concordo
5	Concordo totalmente

Na Figura 64 encontra-se representado o inquérito de satisfação que foi respondido pelos profissionais da empresa, que é constituído por seis questões.

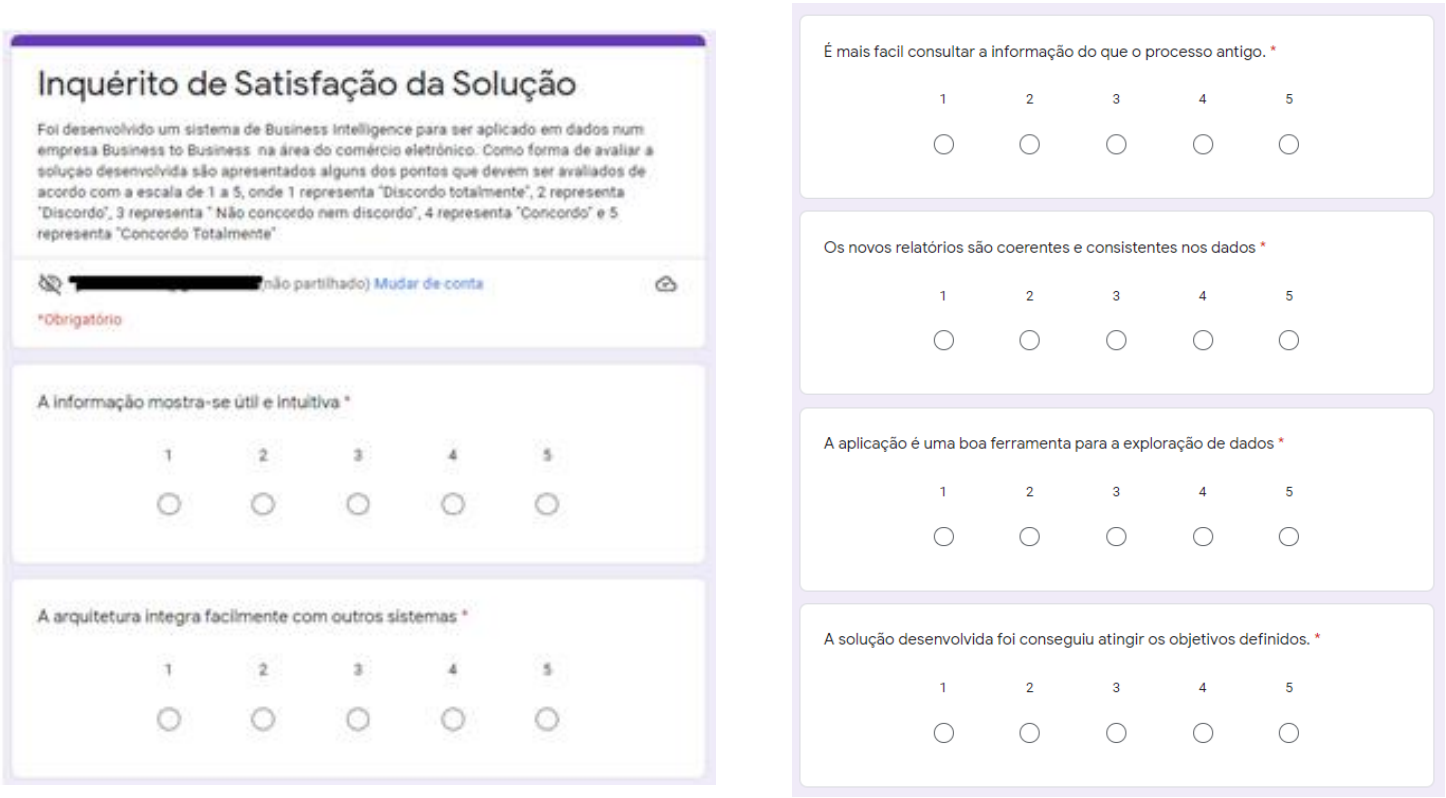

Figura 64 - Inquérito de Satisfação da Solução

6.2 Análise do Resultado dos Inquéritos

O inquérito de satisfação da solução contou só com a participação de um total de onze profissionais, devido ao número reduzido de pessoas que utilizam a solução desenvolvida. De acordo com as opiniões obtidas, conseguiu-se resultados bastantes positivos. A Tabela 8 apresenta de forma resumida o resultado para a frequência das respostas a cada questão.

Tabela 8 - Frequência de Respostas ao Inquérito

	1	2	3	4	5
R1	0% (0)	0% (0)	0% (0)	18,2% (2)	81,8% (9)
R2	0% (0)	0% (0)	0% (0)	36,4% (4)	63,6% (7)
R3	0% (0)	0% (0)	0% (0)	27,3% (3)	72,7% (8)
R4	0% (0)	0% (0)	0% (0)	36,4% (4)	63,6% (7)
R5	0% (0)	0% (0)	0% (0)	27,3% (3)	72,7% (8)
R6	0% (0)	0% (0)	0% (0)	45,5% (5)	54,5% (6)

Segundo os resultados obtidos, na primeira questão dois participantes selecionaram a opção 4 (Concordo) e outros 6 selecionaram a opção 5 (Concordo totalmente). Dado estas respostas, pode se concluir que a informação resultante da solução criada mostra-se útil e intuitiva.

Na segunda questão existiram quatro participantes que escolheram a opção 4 (Concordo) e outros 7 que optaram pela opção 5 (Concordo totalmente). Posto isto, pode se verificar que foi criada uma arquitetura para a solução que pode ser facilmente integrada com outros sistemas.

A terceira e a quinta questão foram respondidas com uma igual frequência, onde três profissionais selecionaram a opção 4 (Concordo) e sete escolheram a opção 5 (Concordo totalmente). Através destas respostas, conclui-se que a solução facilitou o acesso à informação, comparativamente o processo previamente existente e os inquiridos sentiram-se confortáveis com a nova ferramenta de exploração de dados.

Na quarta questão foi selecionada a opção 4 (Concordo) por quatro inquiridos e a opção 5 (Concordo totalmente) por sete inquiridos. Desta forma conclui-se que os dados apresentados nos relatórios são coerentes e consistentes.

Por último, na sexta questão a opção 4 (Concordo) foi selecionada por cinco inquiridos e a opção 5 (Concordo totalmente) foi escolhida por seis inquirido, provando assim, que, segundo os inquiridos os objetivos foram concretizados.

Com este inquérito conseguiu-se concluir que o grau de satisfação dos profissionais que trabalham com a solução desenvolvida é elevado.

7 Conclusão

Neste capítulo são descritas todas as conclusões obtidas ao longo desta dissertação. Para além dos objetivos alcançados, são mencionados os problemas/limitações que existiram ao longo da mesma e por fim, é apresentado o trabalho futuro que ainda se poderia desenvolver, tendo por base o contexto desta dissertação.

7.1 Objetivos alcançados

Esta dissertação teve como objetivo o desenvolvimento de um sistema de BI que permitisse a gestão de informação numa empresa de comércio eletrónico que fosse capaz de fornecer toda a informação necessária aos gestores, de forma a apoiar no processo de tomada de decisão relativamente ao tema em questão.

Ao longo desta dissertação, foi contextualizado o problema, apresentou-se a análise de valor, foram abordados conceitos e explorados a aplicabilidade de sistemas de BI, tendo em consideração o tema em estudo.

Para além disso, foram analisadas um conjunto de ferramentas de ETL e de BI, de forma a verificar quais as que se adequavam melhor às necessidades pretendidas. As ferramentas selecionadas foram o DataFlow e o BigQuery da Google.

Foi criado sistema de BI que possui uma arquitetura centralizada, de forma a estar preparado para situações futuras que possam advir (Ex: outras fontes de dados), disponibilizando toda a informação segundo um único padrão e capaz de armazenar histórico.

Foram criadas diversas análises às vendas dos clientes para perceber o seu desempenho e também foram criados relatórios tanto para fornecer aos clientes, como para fazer análises internas.

Por fim, como forma de validar se os requisitos funcionais mencionados nesta dissertação foram cumpridos, foram realizados inquéritos de satisfação. Os formulários disponibilizados tinham como objetivo obter a opinião relativamente há usabilidade do sistema, ao grau de satisfação relativo ao tempo de resposta, ao cumprimento dos requisitos e à importância dos indicadores implementados. Através dos inquéritos de satisfação efetuados aos profissionais, todos mostram opiniões bastante positivas, que refletem que os objetivos a atingir com a aplicação foram conseguidos.

7.2 Problemas/Limitações

Ao longo da criação da solução apareceram alguns problemas e pontos a melhorar que serão considerados para os trabalhos futuros.

Devido à limitação do tempo não foi possível a criação de uma pipeline de descontos, que terá que ser elaborada numa fase posterior.

Também se detetou que em muitas análises feitas sobre a tabela de vendas era necessária a informação da categoria do produto, o que leva a executar sempre *joins* com a tabela de produtos e que torna essa consulta mais lenta e dispendiosa.

Por fim, detetou-se que na base de dados de preparação, a tabela de *tracking* fica com as consultas muito lentas, devido a existir mais do que uma pipeline a fazer consultas e inserções ao mesmo tempo.

7.3 Trabalho futuro

Dado a este projeto não estar fechado, existem vários pontos que podem ser feitos e melhorados futuramente.

Como foi mencionado no capítulo anterior, poderá alterar-se a pipeline para que no momento que é criada uma venda, esta ir recolher os dados das categorias dos produtos associados a essa venda.

Também foi mencionado que as pipelines que utilizam os dados de tracking estariam lentas devido ao facto de lerem e escreverem para lá todas ao mesmo tempo, futuramente, poderá ser criada uma tabela na base de dados de preparação para cada uma das *pipelines (tracking,* vendas).
Por último, dados que os dados se encontram todos num repositório, será possível aplicar-se modelos de *machine learning* sobre os dados. Com isto consegue-se criar conjuntos que

tenham mais vendas, reduzindo, assim, o esforço humano que é necessário para se criar as regras que são utilizadas no momento da criação de conjuntos.

Referências

Amazon. (2020, Dezembro 18). *AWS Lambda enables functions that can run up to 15 minutes*. Amazon Web Services, Inc. https://aws.amazon.com/about-aws/whats-new/2018/10/aws-lambda-supports-functions-that-can-run-up-to-15-minutes/

Amazon. (2021a, Fevereiro 2). *Deploy a Data Warehouse on AWS*. Amazon Web Services, Inc. https://aws.amazon.com/getting-started/hands-on/deploy-data-warehouse/

Amazon. (2021b, Fevereiro 7). *Amazon Redshift Customers—Cloud Data Warehouse—Amazon Web Services*. Amazon Web Services, Inc. https://aws.amazon.com/redshift/customer-success/

Amazon. (2021c, Fevereiro 13). *Amazon Redshift Pricing—Cloud Data Warehouse—Amazon Web Services*. Amazon Web Services, Inc. https://aws.amazon.com/redshift/pricing/

AWS. (2021). *AWS RDS (Relational Database Service)—Amazon Web Services*. Amazon Web Services, Inc. https://aws.amazon.com/pt/rds/

AWS. (2021). *Elastic Compute Cloud—Amazon EC2—AWS*. Amazon Web Services, Inc. https://aws.amazon.com/pt/ec2/

Belliveau, P., Griffin, A., Somermeyer, S., & Product Development & Management Association (Eds.). (2002). *The PDMA toolbook for new product development*. John Wiley & Sons, Inc.

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing Data Marts for Data Warehouses. *ACM Transactions on Software Engineering and Methodology*, *10*(4), 32.

Celery. (2021). *Introduction to Celery—Celery 5.1.2 documentation*. https://docs.celeryproject.org/en/stable/getting-started/introduction.html

Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, *26*(1), 65–74. https://doi.org/10.1145/248603.248616

Datawarehouse4u. (2021, Março 7). *What are Slowly Changing Dimensions?* Xplenty. https://www.datawarehouse4u.info/SCD-Slowly-Changing-Dimensions.html

Economias. (2021, Fevereiro 22). *Análise SWOT: O que é e para que serve?* Economias. http://www.economias.pt/analise-swot-o-que-e-e-para-que-serve/

Ekanayake, J. (2021, Fevereiro 7). *Inmon vs Kimball—The great data warehousing debate*. Medium. https://medium.com/cloudzone/inmon-vs-kimball-the-great-data-warehousing-debate-78c57f0b5e0e

FEUP. (2021, Fevereiro 22). *Chapter 13*. https://paginas.fe.up.pt/~acbrito/laudon/ch13/chpt13-1bullettext.htm

G2. (2021, Fevereiro 2). *The G2 on Google BigQuery*. G2. https://www.g2.com/products/google-bigquery/reviews

Gartner. (2020, Novembro 23). *Magic Quadrant for Cloud Database Management Systems*. https://www.gartner.com/doc/reprints?id=1-24BO6U2T&ct=201006&st=sb

Gartner. (2021, Janeiro 31). *Magic Quadrant Research Methodology*. Gartner. https://www.gartner.com/en/research/methodologies/magic-quadrants-research

GeeksforGeeks. (2019, Maio 27). Difference between Star Schema and Snowflake Schema. *GeeksforGeeks*. https://www.geeksforgeeks.org/difference-between-star-schema-and-snowflake-schema/

Google. (2021). *Loading Avro data from Cloud Storage | BigQuery*. Google Cloud. https://cloud.google.com/bigquery/docs/loading-data-cloud-storage-avro

Google. (2021a, Fevereiro 2). *BigQuery: Cloud Data Warehouse*. Google Cloud. https://cloud.google.com/bigquery

Google. (2021b, Fevereiro 2). *Customers | Google Cloud*. https://cloud.google.com/customers

Google. (2021c, Fevereiro 2). *Dataflow*. Google Cloud. https://cloud.google.com/dataflow

Google. (2021d, Fevereiro 2). *Dataflow pricing*. Google Cloud. https://cloud.google.com/dataflow/pricing

Google. (2021e, Fevereiro 2). *Pricing | BigQuery*. Google Cloud. https://cloud.google.com/bigquery/pricing

Google. (2021f, Março 7). *Conheça o Data Studio—Ajuda do Data Studio*. https://support.google.com/datastudio/answer/6283323?hl=pt-BR

Guru99. (2021, Janeiro 24). *What is OLAP? Cube, Operations & Types in Data Warehouse*. https://www.guru99.com/online-analytical-processing.html#4

Hevo. (2021a, Fevereiro 2). 7 Best BigQuery ETL Tools. *Learn | Hevo*. https://hevodata.com/learn/bigquery-etl-tools/

Hevo. (2021b, Fevereiro 6). *Our Customers | Hevo*. https://hevodata.com/customers/

Hevo. (2021c, Fevereiro 6). *Platform | Hevo*. https://hevodata.com/platform/

Hevo. (2021d, Fevereiro 6). *Pricing | Hevo*. https://hevodata.com/pricing/

IBM. (2020, Novembro 12). *Capturing Architectural Requirements*. https://web.archive.org/web/20201112020231/http://www.ibm.com/developerworks/rational/library/4706.html#N100A7

IBM. (2021, Janeiro 6). *What is OLAP?* https://www.ibm.com/cloud/learn/olap

Inmon, W. H. (2005). *Building the data warehouse* (4th ed). Wiley.

Ireland, R. D., & Miller, C. C. (2004). Decision-making and firm success. *Academy of Management Perspectives*, *18*(4), 8–12. https://doi.org/10.5465/ame.2004.15268665

Key2Market. (2018, Março 14). *ROLAP, MOLAP and HOLAP*. Medium. https://medium.com/@key2market/rolap-molap-and-holap-5993ac58790c

Kimball, R. (2008). *The data warehouse lifecycle toolkit by Ralph Kimball*.

Levene, M., & Loizou, G. (2003). Why is the snowflake schema a good data warehouse design? *Information Systems*, *28*(3), 225–240. https://doi.org/10.1016/S0306-4379(02)00021-2

Looker. (2021, Março 7). *Business Intelligence*. Looker. https://looker.com/product/business-intelligence

Microsoft. (2021a, Fevereiro 13). *Azure Synapse Analytics | Microsoft Azure*. https://azure.microsoft.com/en-us/services/synapse-analytics/

Microsoft. (2021b, Fevereiro 22). *Excel specifications and limits*. https://support.microsoft.com/en-us/office/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3

Microsoft. (2021c, Março 7). *Comparação de Preços e Produtos | Microsoft Power BI*. https://powerbi.microsoft.com/pt-pt/pricing/

Microsoft. (2021d, Março 7). *O que é o Power BI | Microsoft Power BI*. https://powerbi.microsoft.com/pt-pt/what-is-power-bi/

Nogués, A., & Valladares, J. (2017). *Business Intelligence Tools for Small Companies*. Apress. https://doi.org/10.1007/978-1-4842-2568-4

Nutt, P. C. (2002). Making Strategic Choices. *Journal of Management Studies*, *39*(1), 67–96. https://doi.org/10.1111/1467-6486.00283

Oracle. (2020, Dezembro 12). *What does Business Intelligence mean to you?* https://www.oracle.com/what-is-business-intelligence.html

Oracle. (2021, Fevereiro 13). *Automate Your Data Warehouse*. https://www.oracle.com/pt/autonomous-database/autonomous-data-warehouse/

Porter. (2021, Fevereiro 6). *Porter's Value Chain: Understanding How Value is Created Within Organizations*. http://www.mindtools.com/pages/article/newSTR_66.htm

Porter, M. E. (1998). *Competitive strategy: Techniques for analyzing industries and competitors: with a new introduction* (1st Free Press ed). Free Press.

Schouten, E. (2008, Maio 14). The theory of Data Warehousing. *Edwin Schouten*. https://edwinschouten.nl/2008/05/14/the-theory-of-data-warehousing/

Tableau. (2020, Dezembro 12). *What is business intelligence? Your guide to BI and why it matters*. Tableau. https://www.tableau.com/learn/articles/business-intelligence

Tableau. (2021a, Fevereiro 20). *Our Products*. https://www.tableau.com/products

Tableau. (2021b, Fevereiro 20). *Pricing for data people*. Tableau. https://www.tableau.com/pricing/teams-orgs

Talend. (2021a, Fevereiro 6). *Talend Pricing Model: Learn about Talend License Cost Details*. Talend Real-Time Open Source Data Integration Software. https://www.talend.com/products/pricing-model/

Talend. (2021b, Fevereiro 6). *Talend—A Cloud Data Integration Leader (modern ETL)*. Talend Real-Time Open Source Data Integration Software. https://www.talend.com/

Tuan, N. (2020, Outubro 19). *5 BigQuery BI & Reporting Tools that levels up your data team's performance*. The Holistics Blog. https://www.holistics.io/blog/bigquery-bi-and-reporting-tools-that-level-up-your-data-team-performance/

Yıldırım, A. (2020, Junho 15). *What's the difference between OLTP and OLAP?* Medium. https://medium.com/@yildirimabdrhm/whats-the-difference-between-oltp-and-olap-bdcafdffb1c3

ZoinerTejada. (2021, Janeiro 10). *Extração, transformação e carregamento (ETL)—Azure Architecture Center*. https://docs.microsoft.com/pt-pt/azure/architecture/data-guide/relational-data/etl