Desenvolvimento de um Sistema de *Business Intelligence* com um Algoritmo de Recomendações

Luís Freitas

UMinho | 2021

Universidade do Minho
Escola de Engenharia

Luís Pedro Novais Freitas

Desenvolvimento de um Sistema de *Business Intelligence* com um Algoritmo de Recomendações

Julho de 2021

Universidade do Minho
Escola de Engenharia

Luís Pedro Novais Freitas

Desenvolvimento de um Sistema de *Business Intelligence* com um Algoritmo de Recomendações

Dissertação de Mestrado

Mestrado em Engenharia de Sistemas

Trabalho realizado sob a orientação de

Prof. Doutor Paulo Jorge Freitas Oliveira Novais

Julho de 2021

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Agradecimentos

A dissertação é, muito provavelmente, o projeto mais solitário durante todo o percurso académico. De qualquer forma, para ser realizado é também necessário o apoio de quem nos é mais próximo e de entidades que o proporcionam.

Agradeço aos meus orientadores, Professor Paulo Novais e Doutor Amadeu, por promoverem a realização deste projeto, agradeço as suas sugestões e críticas construtivas.

Agradeço aos meus colegas de equipa da KSI por me ajudarem no desenvolvimento deste projeto e por me fazerem crescer em termos profissionais.

Gostava de fazer um agradecimento muito especial ao meu Pai e a minha Mãe que sempre me guiaram de forma a me tornarem num ser humano com valores e com objetivos, e essa é a base para que projetos como este possam ser realizados, com amor, com esforço e com dedicação. Também lhes agradeço por todo o apoio que me deram durante o meu percurso académico, felicitando-me no cumprimento dos meus deveres e aturando-me nos meus momentos mais difíceis sem nunca me virarem as costas. Agradeço também ao meu irmão, pelo seu carinho e pela verdadeira amizade que temos, também pela sua ambição e forma de ser que me motivaram na realização deste projeto.

Agradeço a minha namorada, que também é a minha melhor amiga, por todas as horas que se privou da minha companhia para que este projeto fosse uma realidade e por toda a ajuda que me deu, com a promessa de que o tempo perdido será compensado com muito amor e carinho.

Quero agradecer ao resto dos meus familiares e amigos por todas as horas que se privaram da minha companhia para que este projeto pode-se ser realizado.

Agradeço aos meus irmãos da Tuna pelos momentos de diversão e descontração proporcionados durante toda a minha vida académica e durante a realização da minha dissertação.

Por fim, agradeço à melhor, ao meu maior exemplo de força, de superação, agradeço por todo o amor que me dá, por tudo o que fez e continua a fazer por mim e por tudo o que me ensinou

Obrigado Avó.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Resumo

Desenvolvimento de um Sistema de Business Intelligence com um Algoritmo de Recomendações

O projeto de dissertação aborda a implementação de uma Solução de *Business Intelligence* e aplicação de algoritmos de recomendação num contexto empresarial.

Numa primeira fase foi elaborado o estudo da arte dos principais temas, os Sistemas *Business Intelligence* e os Sistemas de Recomendação. O levantamento de requisitos foi uma componente do projeto que serviu para definir os objetivos do desenvolvimento e perceber que problemas é que seriam resolvidos com as implementações. A análise da fonte de dados da organização foi também elaborada de forma a assegurar a informação necessária para o cumprimento dos objetivos.

A fase de desenvolvimento levou a cabo o desenho de um modelo dimensional para a implementação física de um *Data Warehouse*. A construção de uma pipeline ETL foi realizada de forma a armazenar os dados com conformação estruturada no *Data Warehouse*. O Sistema de *Data Warehousing* ficou completo depois de se programar um *job* do *SQL Server* para executar o processo ETL a uma hora estipulada todos os dias, de forma a refrescar os dados contidos na nova base de dados. Foi desenvolvida uma aplicação de monitorização das atualizações do *Data Warehouse*, de forma a que o gestor das bases de dados possa realizar auditorias e analisar estatísticas dos tempos do processo ETL, apenas acedendo à aplicação na sua versão *web* ou *mobile*.

Com os dados estruturados e armazenados no *Data Warehouse*, foi possível desenvolver um algoritmo de recomendações, filtrando desta forma, informações úteis para os utilizadores do sistema, e arrecadando novas oportunidades que são recomendadas por esta componente.

Com todo o processo de *back-end* criado, foi elaborada a fase de *front-end*. Para ser possível o acesso aos dados contidos no sistema de *Business Intelligence*, foram criados relatórios dinâmicos numa aplicação *web* para que os utilizadores consigam analisar as informações, oferecendo-lhes, desta forma, suporte nas tomadas de decisão.

Atualmente, o sistema encontra-se em fase de produção, dentro da organização, sendo que é constantemente necessária a sua manutenção para corrigir falhas que possam ocorrer.

Palavras-Chave: *Business Intelligence, Data Warehouse*, Sistemas de Recomendação, *Reporting*

Abstract

Development of a Business Intelligence System with a Recommendation Algorithm

This dissertation addresses the implementation of a Business Intelligence Solution and the application of recommendation algorithms in a business context.

In the first phase it was elaborated the study of the main themes, Business Intelligence Systems and Recommendation Systems. The requirements gathering was a component of the project that served to define the objectives of the development and to understand which problems would be solved with the implementations. The analysis of the organization's data source was also elaborated in order to ensure the necessary information for the fulfillment of the objectives.

The development phase carried out the design of a dimensional model for the physical implementation of a Data Warehouse. An ETL pipeline was built in order to store structured data in the Data Warehouse. The Data Warehousing System was completed after a SQL Server job was scheduled to run the ETL process at a stipulated time every day, in order to refresh the data contained in the new database. A Data Warehouse update monitoring application was developed, so that the database manager can perform audits and analyze statistics of the ETL process times, just by accessing the application in its web or mobile version.

With the data structured and stored in the Data Warehouse, it was possible to develop a recommendation algorithm, thus filtering useful information for the system users, and collecting new opportunities that are recommended by this component.

With all the back-end process created, the front-end phase was elaborated. To make possible the access to the data contained in the Business Intelligence system, dynamic reports were created in a web application so that the users can analyze the information, offering them, this way, support in the decision-making process.

Currently, the system is in the production phase, within the organization, and its maintenance is constantly needed to correct failures that may occur.

Keywords: Business Intelligence, Data Warehouse, Recomendation System, Reporting

Índice

Lista de Abreviaturas, Siglas e acrónimos

SAD – Sistemas de Apoio a Decisão

I&D – Investigação e Desenvolvimento

BI – *Business Intelligence*

KPI – Indicador Chave de Performance

IBM – *International Business Machines*

DW – *Data Warehouse*

OLTP – *Online Transaction Processing*

OLAP – *Online Analytical Processing*

ETL – Extração, Transformação e Carregamento

EDW – *Enterprise Data Warehouse*

ER – Entidade Relacionamento

SCD – *Slowly Changing Dimension*

SQL – *Structured Query Language*

DSA - *Data Staging Area*

MOLAP – *Multidimensional Online Analytical Processing*

ROLAP - *Relational Online Analytical Processing*

HOLAP – *Hybrid Online Analytical Processing*

KNN – K Vizinhos mais próximos

SVD – Decomposição em Valores Singulares

RMSE – Raíz do erro quadrático médio

SSIS – *Sql Server Integration Services*

NMF – *Non-negative matrix factorization*

SSRS - *Sql Server Reporting Services*

MDX – Multi Dimensional Expressions

Índice de Figuras

Lista de Tabela

1. Introdução

Esta dissertação foi desenvolvida no âmbito da aquisição do grau de mestre em Engenharia de Sistemas, perante um percurso profissional na empresa Kyaia Soluções Informáticas. Este capítulo dá início à dissertação e faz referência ao enquadramento e objetivos para a sua realização, descreve também a estrutura do documento e dos restantes capítulos.

1.1. Enquadramento

As tecnologias de informação têm tido um papel muito importante nas organizações. A informação organizacional é vista como um bem com cada vez mais importância devido à necessidade crescente de acesso a informação e a conhecimento. As organizações que contém sistemas de dados com informação de qualidade e com a possibilidade de serem acedidos no momento certo, levam seguramente vantagem competitiva sobre as outras.

O aumento da procura por estas vantagens competitivas leva cada vez mais as empresas a procurarem soluções tecnológicas que permitam o apoio nos processos de decisão. A informação gerada pelas aplicações informáticas disponibiliza aos gestores empresariais uma série de indicadores sobre o negócio para traçarem decisões futuras com o que aconteceu no passado. Neste momento, a falta de informação nas organizações já é vista como condutora para erros e perda de oportunidades. Cada vez mais há apostas nos sistemas de análise de dados para melhorar a qualidade da informação e do conhecimento empresarial.

A necessidade de as organizações lidarem com um grande conjunto de dados fez aparecer o conceito de *Business Intelligence*, terminologia utilizada para dar nome a sistemas que usam os dados numa organização para disponibilizar informação importante aos gestores durante o processo de tomada de decisão. Nos sistemas organizacionais encontram-se grandes conjuntos de informação de uma forma dispersa e repetida. Com o uso das tecnologias disponíveis num sistema de *Business Intelligence* é possível moldar e centralizar a informação dispersa e disponibiliza-la de uma forma automática.

Neste seguimento, concede-se o âmbito deste projeto. A empresa FOREVA, uma organização na área do retalho do comércio do calçado, reconhecida a nível nacional, sentiu a necessidade de disponibilizar informação, gerada pelos seus sistemas transacionais, aos seus gestores em tempo útil de forma a serem tomadas as melhores decisões e a facilitar um acompanhamento eficaz do negócio. Com

isto, deu-se inicio à implementação de uma solução de *Business Intelligence* capaz de satisfazer as necessidades analíticas que a organização vinha a solicitar, sendo possível eliminar tarefas diárias de organização de dados feitas por colaboradores, e automatizar um processo de pipelines condutoras de informação desde a entrada dos dados nos sistemas até à criação de conhecimento útil dos gestores.

1.2. Objetivos

O desenvolvimento de uma solução de *Business Intelligence* adaptado às necessidades da organização é o principal objetivo deste projeto, sendo este acompanhado por um sistema de recomendações específicas para o contexto do problema. As recomendações são um conteúdo que deve ser desenvolvido através da informação estruturada do Sistema de *Business Intelligence,* e, tem como critério central filtrar informações importantes de forma a fornecer alternativas aos utilizadores.

Sendo este um projeto de dissertação, teve como primeiro objetivo um estudo em relação ao estado da arte e metodologias que serão implementadas, e de que forma é que esta matéria têm influenciado o conhecimento organizacional.

O segundo objetivo diz respeito a análise dos requisitos e das fontes de dados da organização de forma a ser possível desenvolver uma arquitetura que seja capaz de resolver e cumprir os preceitos de um sistema de *Business Intelligence*.

O desenvolvimento da modelação dimensional e o desenvolvimento de um sistema de *Data Warehousing* revelaram-se como os objetivos seguintes, e posteriormente o desenvolvimento de algoritmos de recomendação.

Por fim, o último objetivo, foi a elaboração de uma estrutura *web* de *reporting* capaz de representar, de forma visual, a informação necessária aos gestores da empresa e cumprir os requisitos estipulados, finalizando, desta forma, o desenvolvimento do sistema.

1.3. Metodologia

A metodologia de investigação que foi utilizada neste projeto, tem nome de investigação-ação, pois durante o projeto existiu a mistura de ideologias e atividades, assim como, o desenvolvimento de soluções de forma a melhorar as capacidades de uma organização superando problemas que podem ser

encontrados de forma habitual. Esta metodologia leva a cabo a solução de problemas com base no estudo e análise de fontes de informação.

1.4. Estrutura da Dissertação

A dissertação é constituída por 5 capítulos, sendo que o presente capítulo descreve uma breve introdução ao projeto de forma a dar entendimento dos objetivos e fundamentos, assim como a estrutura de toda a dissertação.

O capítulo 2 aborda o Estado da Arte das áreas de *Business Intelligence* e Sistemas de Recomendação, tratando as suas principais metodologias e arquiteturas. Este capítulo estima algumas abordagens a áreas periféricas dos temas principais, como por exemplo, a importância da qualidade de dados, o conhecimento organizacional e o universo pluridisciplinar da ciência de dados.

No capítulo 3 é apresentada a empresa para qual a aplicação de *Business Intelligence* é desenvolvida e são explicados, detalhadamente, os objetivos que este novo produto tem de conseguir alcançar de forma a solucionar os problemas do cliente.

No capítulo 4 são apresentadas todas as fases de desenvolvimento do Sistema de *Business Intelligence* e como os requisitos apresentados no capítulo 3 vão sendo superados.

O capítulo 5 apresenta uma conclusão do projeto, analisando os resultados e as contribuições do seu desenvolvimento fazendo uma síntese geral de todo o conteúdo e os trabalhos futuros que poderão ser realizados.

2. Estado da Arte

No Capítulo atual são apresentados os principais conceitos e metodologias do projeto para fundamentar o trabalho desenvolvido.

2.1. A Ciência dos Dados e a Engenharia do Conhecimento nas Organizações

2.1.1. Conhecimento Organizacional

Em marcos históricos Platão defendia uma definição de conhecimento como sendo uma base em crenças verdadeiras e justificadas. Já Aristóteles, acreditava no conhecimento dividido em três áreas, a ciência, a prática e a técnica. O termo conhecimento faz parte do quotidiano, e, segundo Vasconcelos e Barão (2017), hoje em dia está associado ao saber, isto é, informação que se retém no dia-a-dia em processos de tentativa/erro para a formação de ideias, sendo que se aprende com suporte em premissas consideradas verdadeiras. Um requisito chave dentro de uma organização é a integridade e a classificação do conhecimento dentro das diferentes áreas, como por exemplo, a descrição de artigos, as capacidades associadas aos Recursos Humanos e a estrutura da própria empresa. A Gestão de Conhecimento Organizacional permite a eficiência no processo de trabalho e no dia-dia operacional da empresa, isto é, se um colaborador da organização precisar de uma informação para resolver um problema, quanto mais rápido ele obtiver essa informação mais rápido o problema poderá ser resolvido. Outro exemplo são os erros do passado, se existirem informações dos erros anteriores, provavelmente esses erros não vão voltar a acontecer.

O Conhecimento criado a partir da recolha de informações sobre o funcionamento da organização ajuda a criar uma distinção sobre a definição de dados e informação. Segundo Almeida (2007), os dados são entendidos como registos regulares alusivos a todo e qualquer acontecimento, objeto ou pessoa, e informação é quando os dados são, por exemplo, processados no sentido acumulativo ou comparativo, o que será vantajoso para futuras tomadas de decisões. Segundo Nonaka e Takeuchi (1997), quando as organizações do Japão enfrentam momentos de crise, focam-se em desenvolver métodos para criar novos conhecimentos evidenciando-se no entendimento do histórico para estimular novos processos nunca testados, e também desenvolver soluções a partir dos êxitos anteriores. Aqui encontra-se um

fundamento de como os dados históricos se podem tornar em informações úteis para subsidiar o conhecimento na tomada de decisão.

Os Sistemas de Gestão de Conhecimento têm como objetivo principal promover o crescimento do conhecimento dentro das organizações. Para estruturar grandes volumes de informação distribuídos dentro de uma organização, é necessário desenvolver metodologias capazes de criar, identificar e reutilizar os recursos de conhecimento existentes. Segundo Vasconcelos e Barão (2017) podemos interpretar a Gestão do Conhecimento como uma evolução natural dos Sistemas de Informação dentro das organizações. As tomadas de decisão organizacionais precisam, muitas das vezes, de serem efetuadas num ambiente diligente com suporte em dados retirados de múltiplas fontes de informação, assim, os grupos de trabalho nas organizações precisam de explorar e maximizar as melhores práticas para melhorar a eficiência do conhecimento organizacional.

Tal como o pensador Peter Drucker disse, "o que pode ser medido, pode ser melhorado", e dentro de uma organização, há metas e objetivos a cumprir, que devem ser calculados, como por exemplo a utilização da melhor maneira possível dos recursos e o cumprimento de todas as necessidades dos clientes. Dentro de um nível de procedimentos de avaliação geral se a organização se focar, por exemplo, nos departamentos de recursos humanos pode ter indicadores de performance como a taxa de trabalho extraordinário, idade média e média de horas de formação por trabalhador. Estes indicadores são conhecidos como indicadores de desempenho para monitorização organizacional (Caldeira, 2012).

A área da Ciência de Dados, juntamente com a Gestão de Conhecimento, tem ajudado na conceção de paradigmas para gerir e avaliar as competências organizacionais, através de projetos integrantes na área dos Sistemas de Informação para o desenvolvimento de software de gestão de capacidades organizacionais. (Vasconcelos e Barão, 2017).

2.1.2. A Análise de Dados nas Organizações

A análise de dados dentro das organizações permite a criação de várias vantagens competitivas. Para se resolver dificuldades utilizando a Ciência de dados, inicialmente deve-se descrever que tipos de problemas existem, depois de compreendidos estes problemas são identificadas as técnicas associadas à Ciência de dados que são adequadas para resolver o problema. Aplica-se as técnicas em conjunto com os algoritmos e avalia-se os resultados. Depois dos resultados serem avaliados e validados, são usados para melhorar o processo de tomada de decisão (Vasconcelos e Barão, 2017). É importante a cultura relacionada com a análise de dados que a organização deve ter de forma a ser possível pensar e atuar

contextualizando o problema e o processo de decisão. A principal ideologia da Ciência dos Dados é obter e extrair conhecimento dos dados organizacionais.

A união de áreas como as Ciências da Computação, as Ciências Matemáticas e as Estatísticas Computacionais permitem o desenvolvimento de tecnologias e ferramentas de software capazes de processar e analisar grandes volumes de dados fornecendo soluções de conhecimento. A Ciência de Dados é uma área que integra de forma coerente tecnologias de Gestão de Base de Dados, Matemática, Estatística e também tecnologias de *Data Mining* e *Machine Learning*. O processo de digitalização nas organizações é cada vez mais sistemático. Na era anterior, os dados eram analisados manualmente, mas agora os volumes de dados nas organizações crescem a um ritmo exponencial e a análise de dados manual será cada vez mais irrealizável. O decurso de digitalização das organizações serve de base à Ciência dos dados que vai tendo um papel fundamental no processo de análise de grandes volumes de dados, divulgando informação e conhecimento vantajoso para as organizações através de Sistemas de Apoio à Decisão. (Vasconcelos e Barão, 2017).

2.1.3. Sistemas de Apoio à Decisão

Os Sistemas de Apoio à Decisão têm princípio em áreas como a Gestão de Bases de Dados, a Gestão, a Contabilidade, a Inteligência Artificial, a Investigação Operacional e a Matemática Aplicada em que os factores importantes passam pelo armazenamento de dados e o desenvolvimento científico e tecnológico. Os SAD são Sistemas que, como o próprio nome indica, apoiam os decisores com informação vantajosa e pertinente através do armazenamento e gestão do conhecimento.

O nível de administração em que o decisor se encontra pode levar às diferentes caraterísticas das decisões. Estas podem ser estruturadas, estando ao nível de gestão operacional, como por exemplo, fazer inventários e emitir faturas, podem ser semiestruturadas, que corresponde ao nível de gestão tática e intermédia como a seleção de um plano de marketing, e por fim, não estruturadas, que corresponde à gestão estratégica de topo que são as definições de objetivos a longo prazo (Vasconcelos e Barão, 2017).

Figura 1 – Pirâmide Organizacional

A incerteza e a insegurança entre várias alternativas para decisões podem ser mitigadas ou até dissolvidas pelos Sistemas de Apoio à Decisão. O poder de manipular e processar grandes volumes de dados torna estes Sistemas capazes de efetuar análises e exibir relatórios que podem ir de um formato textual até a um formato com gráficos e imagens dinâmicas. Os SAD beneficiam de uma interface com o utilizador desenvolvida para que o utilizador faça pedidos e que o sistema devolva resultados. Para além disso são compostos por Bases de dados e pacotes de software que podem incluir modelos de análise.

As plataformas de SAD nas empresas são desenvolvidas de forma a relacionar a organização de dados e criação de *Data Warehouses* com a finalidade de gerar páginas de informação automatizadas para avaliar indicadores, criar modelos, previsões e desenvolver heurísticas de otimização que vão permitir às instituições ter vantagens competitivas de modo a tomarem decisões de forma mais eficiente em tempo útil.

2.1.4.O Universo Pluridisciplinar da Ciência dos Dados

O processo de extração de Conhecimento está inteiramente ligado ao gigantesco universo das ciências computacionais dos dados tendo como área essencial a gestão de bases de dados, sendo esta o núcleo impulsionador dos outros âmbitos ligados à categoria. A Matemática e a Estatística permitem a evolução de áreas como o *Machine Learning* e o *Data Mining* nas componentes de I&D das organizações. Estas matérias permitem descobrir padrões e antecipar comportamentos futuros através dos dados concebendo poderio na análise computacional e algorítmica, conduzindo o desenvolvimento de *Software* de inteligência artificial.

A área de Engenharia de Software é uma das mais importantes para o ramo da ciência dos dados. Os softwares de funcionamento empresarial têm uma ligação forte com a gestão de bases de dados e permitem o funcionamento de transações empresariais e o crescimento de dados para serem explorados nas disciplinas faladas anteriormente. A abordagem *Data Warehousing* é a área ligada à *Data Science* que permite a estruturação, armazenamento e organização dos dados para dar suporte às tomadas de decisão.

O grande objetivo que unifica todas estas áreas descritas em cima é a extração de conhecimento a partir de grandes volumes de dados (Vasconcelos e Barão, 2017).

2.1.5. Qualidade de Dados

Segundo o Neogrid (2019), *Data Quality* ou Qualidade de Dados é a componente analítica que tem como procedimento a avaliação da confiabilidade das informações que são utilizadas nas organizações. Para uma organização obter um bom indicador de qualidade de dados é preciso que os colaboradores cooperem conscientemente na hora de registar os dados não os tornando incorretos e imprecisos. A integração dos dados deve ser consistente e completamente aderente à estratégia do negócio não havendo repetição de dados e estes têm de estar disponíveis para serem trabalhados. A política de segurança de informação é também importante nas organizações devido à sensibilidade dos dados.

O ato de tomar uma decisão com dados sem integridade pode levar a um prejuízo enorme nas organizações sabendo que avançamos a cada dia que passa para uma cultura cada vez mais analítica. Todas as operações realizadas em ambiente empresarial devem ter em conta a qualidade dos dados para evitar processos deficientes e melhorar a cultura colaborativa evitando a falta de padronização entre

sistemas e trazendo às organizações a acessibilidade dos dados e a segurança para os gestores nas decisões estratégicas, táticas e operacionais.

O tópico seguinte fala sobre o desenvolvimento de soluções de *Business Intelligence* nas organizações e estes têm como objetivo o armazenamento estruturado dos dados para fins analíticos estando fortemente ligados aos procedimentos de *Data Quality*, pois é nestas laborações que são encontradas grande parte das distorções entre os dados nos sistemas empresariais. A etapa de melhoria da qualidade de dados é bastante importante para o sucesso de projetos ligados à *Data Science*, uma vez que permite à organização a redução do custo de correções de dados incoerentes nos sistemas transacionais.

2.2. Business Intelligence

2.2.1. O Processo de Tomada de Decisão, a Gestão Estratégica e o BI

As decisões organizacionais, normalmente, dizem respeito a planos estratégicos de alguma componente empresarial com o objetivo de tirar partido do investimento realizado. Segundo Vercellis (2009), o processo de tomada de decisão é a escolha pela qual se tenta preencher uma falha ou agarrar uma oportunidade nas condições de um determinado sistema e dar resposta a um problema. A natureza de uma decisão está dependente da estrutura da organização e das atitudes dos gestores que tomam decisões.

Numa era em que as organizações estão cada vez mais competitivas torna-se mais importante o elemento de gestão. Segundo Santos (2008), a atividade de gestão é um processo de planeamento, estruturação e incorporação de atividades organizacionais através dos recursos disponíveis para certificar o cumprimento eficiente das metas projetadas. Para se obter resultados positivos nesta componente, deve recorrer-se à formulação de estratégias empresariais planificadas sendo avaliados os progressos e o grau de sucesso através de indicadores chave de desempenho, mais conhecidos como KPI's.

Existem muitos indicadores de performance que podem ser utilizados para a gestão das áreas empresariais, sendo parte essencial das avaliações escolher os KPI's mais importantes para cada situação.

Assim, os KPI's acabam por ser um gerador de consenso dos elementos da organização porque são uma forma aceitável de quantificarem as performances e os resultados das metas da empresa.

Todos os KPI's são constituídos por um algoritmo, ou seja, por uma fórmula matemática que calcula o resultado do indicador. Segundo Jorge Caldeira (2020) para se encontrar um bom indicador é necessário visar algumas caraterísticas, como por exemplo a pertinência, ou seja, se um indicador não for desejado pelo gestor ele rapidamente vai deixar de o visualizar e perder o interesse. Outro factor importante e diretamente ligado com a qualidade de dados é a credibilidade do resultado, pois os dados que alimentam o algoritmo podem conter *bugs* e comprometer o indicador sendo, desta forma, importante fazer auditorias às fontes de dados para identificar possíveis erros. A simplicidade de interpretação é um dos factores mais importantes na escolha de um indicador, se for muito complexo pode gerar confusão na parte do destinatário e causar dificuldades na tomada de decisão. A possibilidade de os indicadores serem alimentados sem intervenção humana, ou seja, pelo computador, torna o processo mais ágil e faz com estes estejam sempre atualizados. Uma das formas de obter esta componente é criar *softwares* que permitam ao gestor aceder a relatórios com KPI's sempre atualizados, para isso recorre-se a soluções de *Business Intelligence*.

Segundo Magalhães (2017), em 1858, na obra *Cyclopaedia of comercial and business anecdotes,* é nomeada pela primeira vez, por Richard Devens, a expressão "*Business Intelligence*" para descrever o sucesso de negócio de um capitalista, Sir Henry Furnese. Cem anos mais tarde, o cientista da IBM Hans Luhn, considerado por muitos o edificador do *Business Intelligence*, referencia esta área num artigo chamado *A Business Intelligence System*.

O *Business Intelligence* é um procedimento que usa soluções tecnológicas para transformar dados em informação útil para as organizações de modo a contribuírem para o processo de tomada de decisão eficaz (Magalhães, 2017). É uma ferramenta de análise descritiva porque os relatórios são baseados no passado e no presente do negócio, ou seja, as informações nas soluções de BI não dizem diretamente ao decisor o que fazer, mas permitem que os gestores acompanhem o resultado das suas decisões e das tendências do mercado.

Os sistemas de BI são inicializados com uma infraestrutura que permite a integração dos dados das fontes num armazém de dados. Os dados ficam estruturados e concisos para serem disponibilizados em contexto histórico sobre a análise de negócio. As ferramentas de *Business Intelligence* acrescentam formas de análise avançada, como a análise estatística e o *Data Mining*, pois a sua flexibilidade em ligar-se a distintas fontes de dados faz com que estes sistemas favoreçam as conexões nas arquiteturas de *Big Data* (Magalhães, 2017).

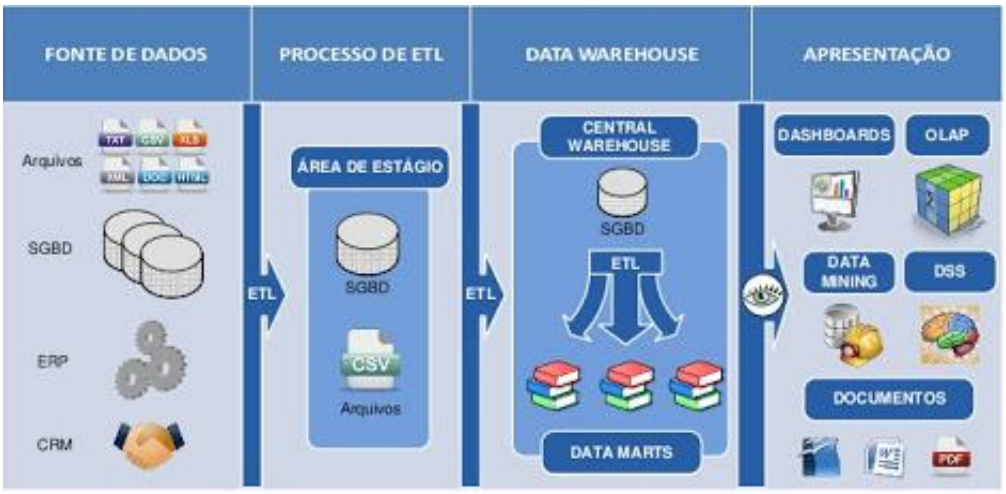


Figura 2 – Arquitetura de BI (Adaptado de Luiz Lorena, 2011)

A implementação de um sistema de BI pode contribuir para o crescimento de uma organização porque pode conceber o conhecimento necessário para boas escolhas futuras e colmatar as falhas do passado, fortalecendo a gestão estratégica e as decisões empresariais.

2.2.2. Sistema de *Data Warehousing*

Como já falado anteriormente no capítulo 2.1, a gestão de bases de dados nas organizações é a área que permite o bom funcionamento de todos os projetos ligados à ciência dos dados e às engenharias de *software*. Nas organizações muitas vezes surgem problemas ligados às proporções de dados inseridos nas bases de dados, dificultando ou até impossibilitando o uso dos mesmos para análise. Os sistemas de *Data Warehousing* permitem integrar os dados de forma estruturada e consistente para poderem ser disponibilizados aos decisores das empresas.

A definição de *Data Warehouse*, dada por Bill Imnon (2005), descreve um sistema de coleção de bases de dados desenhado e integrado para dar suporte à decisão, sendo cada agrupamento de dados não volátil e importante num determinado contexto.

Como é exibido na literatura (Magalhães, 2017), um *Data Warehouse* incorpora as seguintes características:

- Orientado ao Assunto/Contexto: O armazenamento e a organização dos dados num DW são elaborados de forma a permitir uma visualização concisa sobre um determinado assunto, possibilitando a flexibilidade da análise e excluindo os dados que não são aproveitados para o processo de suporte à decisão do alusivo contexto. Como o DW se centraliza em assuntos de

uma área peculiar da organização, não acata o armazenamento transacional e é apresentado como direcionado ao tema (ex: vendas, encomendas, stock, etc...).

- Integrado: O DW permite a integração de dados de várias fontes que muitas vezes são distintas, sendo os dados, dessas fontes diferenciadas, trabalhados e concatenados de forma a representarem uma estrutura única completamente integrada com toda a informação necessária para dar resposta a análises e a consultas.
- Não Volátil: Os dados de um DW não são modificados ou apagados, já nas bases de dados transacionais os dados são constantemente alterados. De forma a ser possível obter uma melhor perceção sobre este contexto imagine-se um círculo empresarial onde é introduzida uma encomenda na base de dados operacional de uma empresa, e posteriormente, no DW orientado ao respetivo assunto comercial. No caso de mais tarde esta encomenda ser cancelada, haverá uma alteração na base de dados transacional, sendo que o status da respetiva encomenda que se transformará em cancelamento. Contrariamente, no DW é introduzido um novo dado de cancelamento, não havendo qualquer alteração dos dados inseridos anteriormente.
- Variável no Tempo: O DW tem sempre a presença de um elemento de horizonte temporal ao contrário das bases de dados operacionais que podem até não abranger nenhum identificador temporal. Normalmente, o horizonte temporal de uma base de dados operacional poderá estar entre 60 a 90 dias, já num DW o período temporal pode possuir 5, 10 ou mais anos.

O *Data Warehouse* é representado como uma compilação de dados sólida que permite a criação de modelos para dar suporte às tomadas de decisão empresariais e o acesso a informação significativa para a gestão estratégica empresarial.

Geralmente, nas organizações, os *Data Warehouses* apresentam-se separados das bases de dados transacionais devido às diferentes finalidades dos sistemas (Dayal & Chaudhuri, 1997). As bases de dados operacionais permitem o funcionamento informático das operações dentro da empresa, inserindo, modificando e apagando dados conforme são registadas transações, estes sistemas são chamados de *On-Line Transaction Processing* (OLTP). Os *Data Warehouses* são focados nas componentes de processos analíticos e são conhecidos como *On-Line Analitic Processing* (OLAP).

2.2.3. Arquiteturas de Sistemas de *Data Warehousing*

Nos *Data Warehouses*, um dos passos mais relevantes de desenvolvimento é a decisão de como os sistemas se vão arquitetar. Estes sistemas podem ter uma componente de *Operational Data Store*, isto é, uma base de dados capaz de integrar dados de múltiplas fontes para depois serem tratados e serem enviados para o DW, ou podem ter múltiplos *Data Marts* que se referem a vários assuntos do negócio.

Como Magalhães (2017) explicou na sua obra, os sistemas de *Data Warehousing* são constituídos pelos seguintes níveis:

- *Data Source level*: São as fontes de dados que constituem os dados que serão usados para permanecerem no processo com o objetivo de serem carregados no *Data Warehousing*. Os dados podem ser provenientes de bases de dados OLTP, ficheiros de texto, ficheiros Excel, etc.;
- *Data Extraction level*: É o processo de filtragem de dados onde são importados os dados das bases de dados origem para o *Data Warehouse*.
- *Data Staging level*: Depois da saída da fonte de dados aqui situa-se o primeiro armazenamento, onde são implementados processos de tratamento de dados.
- *Data Storage level*: Os dados são armazenados nesta área depois dos procedimentos de extração e transformação.
- *Data Presentation level*: Ferramentas OLAP e aplicações de *front-end* são usadas para exibir informação, de variadas formas, aos utilizadores.

Normalmente, são aceites duas abordagens muito conhecidas na implementação de um sistema de *Data Warehousing*, sendo estas de dois tipos distintos, *bottom-up* e *top-down*.

A do primeiro tipo é uma abordagem definida por Ralph Kimball, e é constituída por 4 etapas. As fontes de dados operacionais, o *Data Staging*, o *Data Presentation* e as aplicações de *Business Intelligence*. As fontes de dados operacionais são sistemas que alimentam o DW e que o utilizador não tem qualquer controlo sobre elas sendo que as suas principais caraterísticas são conferir e executar as tarefas fundamentais dos negócios da organização e a sua velocidade de processamento é rápida no que diz respeito a inserir, apagar e alterar dados, mas não tão rápida no processo de consulta, podendo estas, ser bases de dados relacionais, ficheiros Excel, ficheiros de texto, entre outros.

A área de estágio é relativamente importante pois adota uma área de armazenamento de dados e um sistema de Extração, Transformação e Carregamento de dados (*ETL System*), sendo que esta área

não faculta nenhum instrumento de consulta ou visualização. O processo ETL começa pela extração dos dados para estes poderem sofrer uma série de transformações e modificações de forma a que no final sejam carregados na área de apresentação com uma forma estruturada.

A área de apresentação é a área onde os dados estão organizados e estruturados nos *Data Marts* que dizem respeito às áreas de negócio, e podem ser acedidos por ferramentas de *Business Intelligence*.

As aplicações de *Business Intelligence* são a fase final desta arquitetura e são aplicações que são responsáveis por gerar relatórios e fornecer informação importante para a tomada de decisão empresarial (Magalhães, 2017).

Como se pode perceber, esta arquitetura, também conhecida como *Dimensional Data Warehouse*, para além de facilitar o acesso aos dados dos sistemas, dá mais pertinência aos aspetos de negócio e começa por criar os *Data Marts* que ao longo do tempo vão completando o *Data Warehouse*, daí o tipo *bottom-up* (Magalhães, 2017).

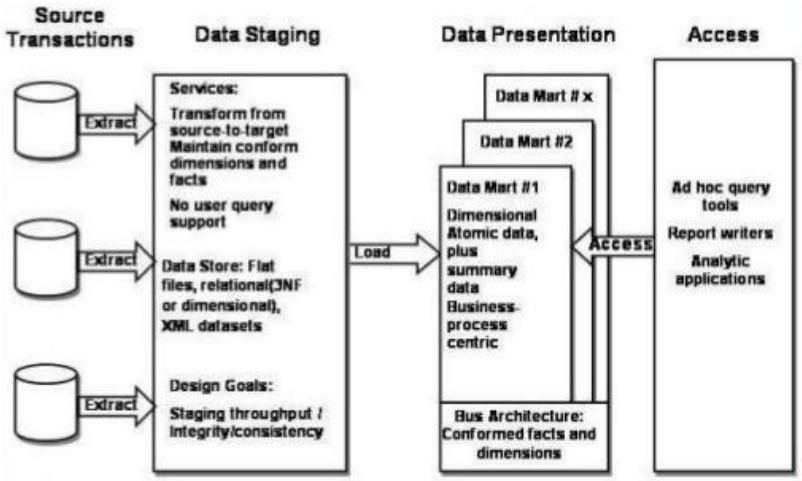


Figura 3 – Arquitetura de Kimball (adaptada de Kimball e Ross, 2003)

A outra arquitetura, do tipo *top-down*, foi idealizada por Bill Inmon e é centralizada nos departamentos empresariais, isto é, cada *Data Mart* armazena os dados de um determinado departamento dentro da empresa. Nesta arquitetura, quando os dados saem das fontes de dados, são armazenados num *Entreprise Data Warehouse* durante o processo ETL e depois são armazenados nos respetivos *Data Marts* para finalmente servirem as aplicações de *Business Intelligence*. Uma grande particularidade nesta arquitetura é que a componente de *Data Presentation* também abrange o EDW, ou seja, podem ser efetuadas consultas no *Data Warehousing* empresarial e nos *Data Marts* (Magalhães, 2017).

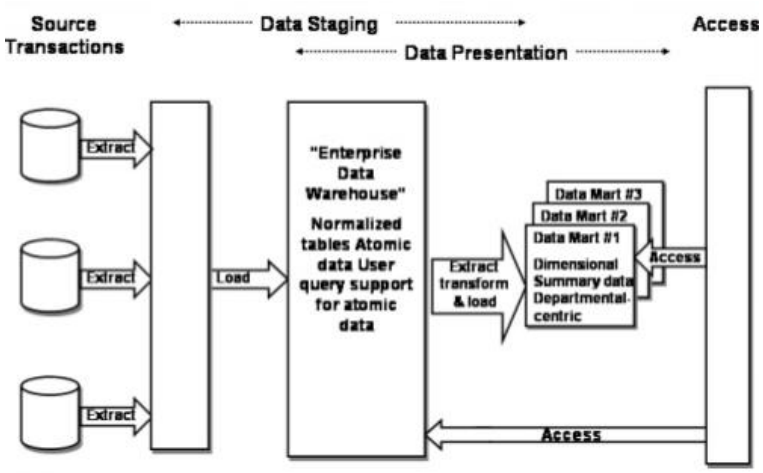


Figura 4 – Arquitetura de Inmon (adaptada de Inmon, 2002)

Uma das grandes preocupações empresariais no que diz respeito à área de BI é saber que tipo de arquitetura é mais benéfica para construir um sistema de *Data Warehouse* e é importante estabelecer a diferença entre as arquiteturas faladas anteriormente definindo o que se perde e o que se ganha na escolha de cada uma.

A laboração da arquitetura de Kimball tem como objetivo implementar *Data Marts* dimensionais de acordo com as necessidades de cada unidade de negócio, tendo em vista a unificação dos mesmos. É uma abordagem mais rápida, os resultados são maís detalhados, mas como o desenvolvimento dos *Data Marts* é feito consoante a necessidade das unidades de negócio, é difícil unificá-los devido às particularidades específicas dos sistemas de dados em cada processo de negócio. É uma abordagem mais útil para decisões táticas e as equipas de desenvolvimento não necessitam de ser muito grandes.

A abordagem de Inmon baseia-se no desenvolvimento de um *Data Warehouse* empresarial utilizando o modelo ER (Entidade Relacionamento), e depois na criação de *Data Marts* com modelos dimensionais, esta abordagem precisa de mais tempo para o desenvolvimento e tem uma elevada complexidade pois o EDW tem de conseguir abranger dados de todos os departamentos da empresa, a equipa tem de ser grande e com especialistas na área, o foco são as decisões estratégicas (Ariayachandraa & Watson, 2011).

2.2.4. Modelos e Dados Dimensionais

O desenho dimensional é fundamental no desenvolvimento do *Data Warehousing* porque este precisa de estar estruturado para sustentar eficientemente a análise de dados e o desenvolvimento de

relatórios. O modelo dimensional (OLAP) é constituído por tabelas dimensão e tabelas de facto e as relações entre elas têm de permitir a consulta dos dados de forma eficaz e célere. Ao contrário dos sistemas OLAP, os sistemas OLTP usam o modelo transacional que é útil para processamento de dados operacionais, mas é inadequado para modelos de *Data Warehouse*. Nos modelos de *Data Warehouse*, o objetivo é a execução de consultas e carregamento de dados, e por isso utiliza-se a modelação dimensional (Magalhães, 2017).

Como foi possível verificar no parágrafo anterior, a modelação dimensional apresenta como constituintes as tabelas de facto e as dimensões, mas não são os únicos elementos, existem também os atributos e as hierarquias. As dimensões são os grupos de dados que representam os processos de negócio das tabelas de facto, ou seja, as variáveis de análise, como por exemplo a dimensão produto, dimensão data ou dimensão cliente. Cada dimensão apresenta o seu conjunto único de atributos e pode usar-se como exemplo a dimensão cliente, onde alguns dos seus atributos podem ser nome, morada e idade sendo que os níveis de relação dos atributos são indicados pelas hierarquias. Usando novamente o exemplo da dimensão cliente podemos ter como hierarquia país-cidade-distrito-morada, as hierarquias determinam como os factos podem ser agregados, estruturando a análise dos dados. Em cada dimensão existe um atributo que é a chave primária e pode existir um atributo que é a chave substituta. A chave primária é única que liga os dados das fontes às dimensões, mas no contexto de DW pode usar-se a chave substituta, que também é única e não tem conjuntura semântica, ou seja pode ser utilizada como chave primária para ligar as tabelas de facto às dimensões, tornando, desta forma, a chave primária da fonte de dados como chave candidata no DW. Portanto, nesta abordagem, fica a chave substituta (*Surrogate Key*) como chave primária (*Primary Key*) para controlar os registos únicos da base de dados, e a chave primária da fonte de dados passa a chave de negócio no DW para ser exposta e manipulada pelo utilizador já que é exclusiva e contem o contexto de negócio (Magalhães, 2017).

Um dos fatores mais importantes nas dimensões é que estas podem variar ao longo do tempo, ou seja, sofrem mudanças físicas de uma forma imprevisível que tanto podem ser atualizações dos dados como correções dos dados (Rainardi, 2008). Se se pretender fazer alterações numa tabela dimensão sem modificar as relações entre as dimensões e as tabelas de facto e mantendo o registo histórico, recorre-se a estratégia de *Slowly Changing Dimensions* (SCD) que são as dimensões com variação lenta (Santos & Ramos, 2009). As SCD são especificadas na construção do processo ETL e as mais conhecidas na literatura, podem apresentar vários tipos (Magalhães, 2017):

- Tipo 0: Não é executado nenhuma alteração nos atributos da Dimensão;
- Tipo 1: Na atualização, os valores antigos são substituídos pelos valores mais recentes;

- Tipo 2: Gera-se um novo registo (uma nova linha) na tabela dimensão;
- Tipo 3: O novo registo é colocado na coluna de "registo atual" e o registo antigo na coluna de "registo anterior" adicionadas na tabela;
- Tipo 4: Cria-se uma tabela que guarda o histórico de alterações;

O Tipo 2, que reverencia a caraterística anunciada por Inmon (2005) de o DW ser não-volátil, é o mais utilizado nestes sistemas pois é meditado como o mais eficiente permitindo a criação de um novo registo e desta forma manter todo o histórico sendo também um dos mais complexos de implementar porque é preciso usar uma estrutura que permita evidenciar facilmente o registo antigo e o registo recente (Nguyen, Tjoa, Nemec, & Windisch, 2006). Uma das técnicas mais usadas (Magalhães, 2017) para corresponder ao Tipo 2 é incluir 3 colunas na dimensão:

- DataInicial: registo da data/hora da inserção na linha;
- DataFinal: Registo da data/hora em que a linha se tornou desatualizada, se a linha ainda está atualizada então este campo aparece sem nenhum registo;
- StatusAtual: Registo da indicação da linha mais recente;

É possível usar apenas os campos DataInicial e DataFinal ou usar apenas o campo StatusAtual, mas esta última faz perder a perspetiva temporal dos registos (Magalhães, 2017).
Em relação ao Tipo 1 e ao Tipo 3, no primeiro temos a técnica de atualização mais fácil porque é só substituir os registos na atualização do DW mas, por outro lado, os registos atualizados não são guardados, então o uso do Tipo 1 deve ser em registos em que não faz sentido guardar as alterações históricas. No que diz respeito ao Tipo 3, também de fácil implementação, como só é guardada a informação do registo atual e do registo anterior, só parte do histórico é que é guardado nas atualizações (Nguyen, Tjoa, Nemec, & Windisch, 2006).

Em relação às tabelas de facto, estas acomodam-se a um específico assunto que entra nos objetivos das análises que são efetuadas, como por exemplo as vendas ou as encomendas. A tabela de facto contém os valores do que foi medido e o seu tipo de dados é numérico. Um Facto pode ser o valor total de vendas de um produto, ou o preço de compra do mesmo, ou seja, os factos correspondem a tabelas com medidas de valor simbólico na análise dos dados. Estas tabelas relacionam-se com as Dimensões a partir de chaves estrangeiras e as suas métricas podem ser aditivas, não aditivas ou calculadas. As medidas aditivas são aquelas que se obtém com a agregação de dados, como por exemplo, a soma da quantidade vendida de um produto num determinado período de tempo. Já as

métricas não aditivas são aquelas que não refletem significância quando são agregadas, se se pretender saber o preço unitário de um artigo não faz sentido agregar a medida. E por fim, as medidas calculadas, onde o valor significativo é resultado de um cálculo sobre uma ou várias métricas, como por exemplo, a margem de lucro das vendas, que é o resultado do valor total das vendas menos o valor total do custo das vendas (Magalhães, 2017).

Tanto as tabelas de facto como as tabelas de dimensão têm de ser modeladas para tornarem o sistema eficaz e veloz na consulta de dados, construindo desta forma, as relações entre as tabelas. Como falado anteriormente, as tabelas de facto representam os decursos de atividade (ex: vendas, encomendas) e as dimensões as suas particularidades literais (ex: cliente, produto). Normalmente, na modelação dimensional existem duas abordagens muito seguidas, o esquema em estrela e o esquema em floco de neve. A estrutura do esquema em estrela tem como centro a tabela de facto e mostra as chaves estrangeiras da mesma, que formam referências com um grupo de dimensões. Os dados são armazenados para o processamento ser o mais eficiente e rápido possível pois a informação fica toda guardada na tabela de facto e nas dimensões de forma a não ser necessário usar mais que um nível de ligação, o que acaba por trazer redundância de dados e violar as regras da normalização devido a simplicidade das ligações, mas acaba por ser o mais eficiente na análise de dados (Magalhães, 2017).

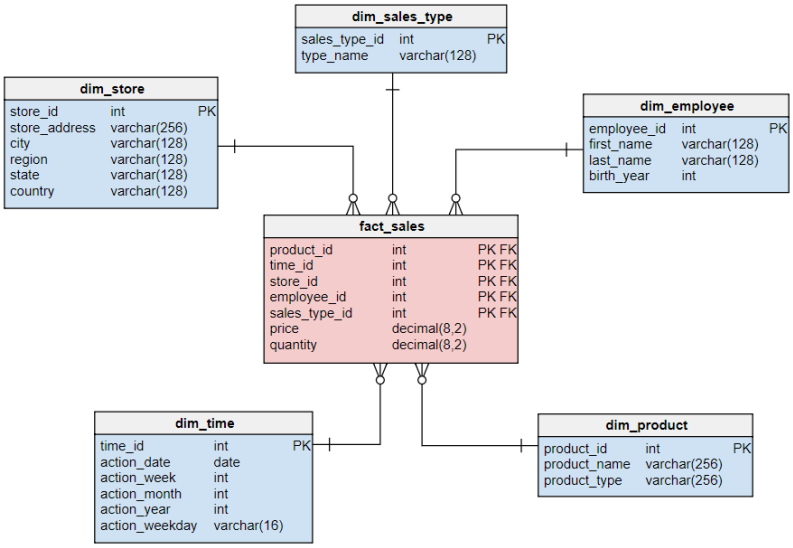


Figura 5 – Exemplo de um esquema em Estrela (Retirado de datawarehouseinfo.com)

Relativamente ao esquema em floco de neve, este apresenta uma estrutura mais complexa tornando as tabelas de dimensão mais polivalentes. Sendo uma extensão do esquema estrela, o floco de neve esfralda as suas hierarquias dimensionais de forma a suprimir a redundância de dados e a ocupar

menos armazenamento, mas utiliza mais tabelas de dimensão e com isso mais ligações piorando a performance de consulta. Outra vantagem do esquema floco de neve é a possibilidade de tabelas de facto diferentes poderem relacionar-se com diferentes níveis de detalhe nas dimensões. Por exemplo, uma tabela de facto pode relacionar-se com a dimensão Calendario, em que o nível de detalhe é o dia, e outra tabela de facto pode relacionar-se com a dimensão Mes, em que o nível de detalhe é o mês (Magalhães, 2017).

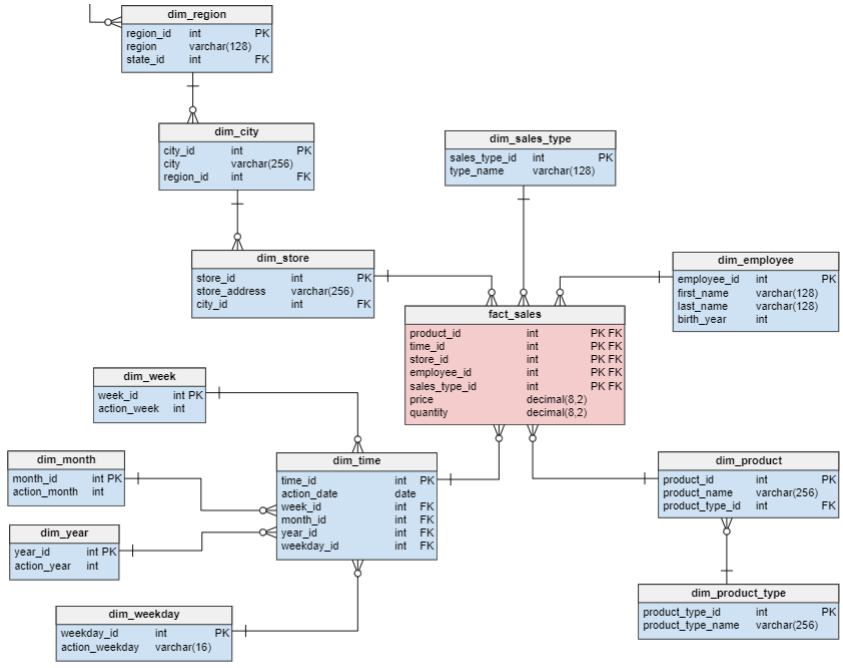


Figura 6 – Exemplo de um esquema em Floco de Neve (Retirado de datawarehouseinfo.com)

Quando mais do que uma tabela de facto partilha as mesmas dimensões, dá-se o nome de esquema em constelação, por exemplo, quando a dimensão cliente está ligada à tabela de facto das vendas e à tabela de facto das encomendas.

Em suma, para se escolher o esquema ideal a utilizar é necessário um bom conhecimento das necessidades e recursos disponíveis de negócio pois é preciso fazer um balanceamento entre o armazenamento e o processamento de dados para que o utilizador obtenha o melhor rendimento possível.

2.2.5. Hierarquias

Uma hierarquia é um grupo de relações com vários níveis numa dimensão (Malinowski e Zimányi, 2004). São representadas por ligações de nodos numa forma de árvore, essas ligações são chamadas de caminhos. Cada nível desses caminhos é chamado por nível dimensional e patenteia um determinado grau de detalhe na corrente de agregação da hierarquia. O tamanho da hierarquia é definido pelos níveis de agregação existentes. Os nodos da hierarquia podem ser pais caso tenham um caminho que levará a uma nova agregação num outro nodo, ou filho, caso sejam nodos descendentes de outros nodos da cadeia. É claro que um nodo pai pode descender de outro nodo e ser também um nodo filho simultaneamente.

As hierarquias podem ser categorizadas com vários tipos. As hierarquias simples simétricas, são as hierarquias representadas com o percurso de uma árvore em que os nodos pais têm de ter pelo menos um nodo filho e os nodos filho têm de ter obrigatoriamente apenas um nodo pai. As hierarquias assimétricas descartam a obrigatoriedade de um nodo pai possuir obrigatoriamente um nodo filho.

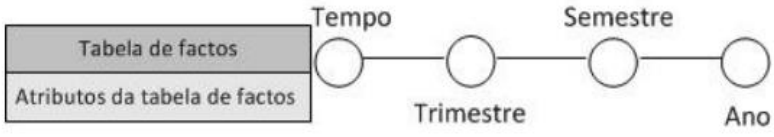


Figura 7 – Hierarquia Simétrica (Bruno Oliveira e Orlando Belo, 2012)

São chamadas de hierarquias múltiplas, as hierarquias simples em que os nodos filho podem estar relacionados com um ou mais nodos pai, não forçosamente referentes ao mesmo nível de hierarquia.

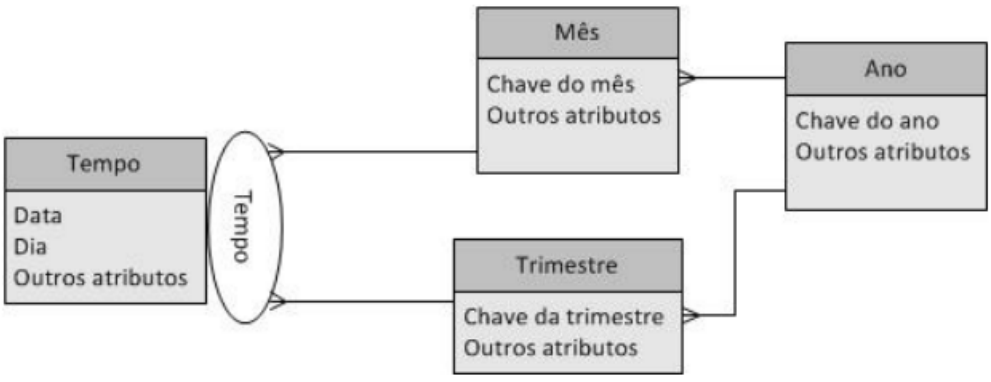


Figura 8 – Hierarquia Múltipla (Bruno Oliveira e Orlando Belo, 2012)

Por fim, as hierarquias são consideradas paralelas se uma dimensão tiver várias hierarquias, e são paralelas independentes se não houver partilha de qualquer nível ou dimensão entre as hierarquias estabelecidas, e são paralelas dependentes se existir partilha de dimensão ou nodos nas distintas hierarquias.

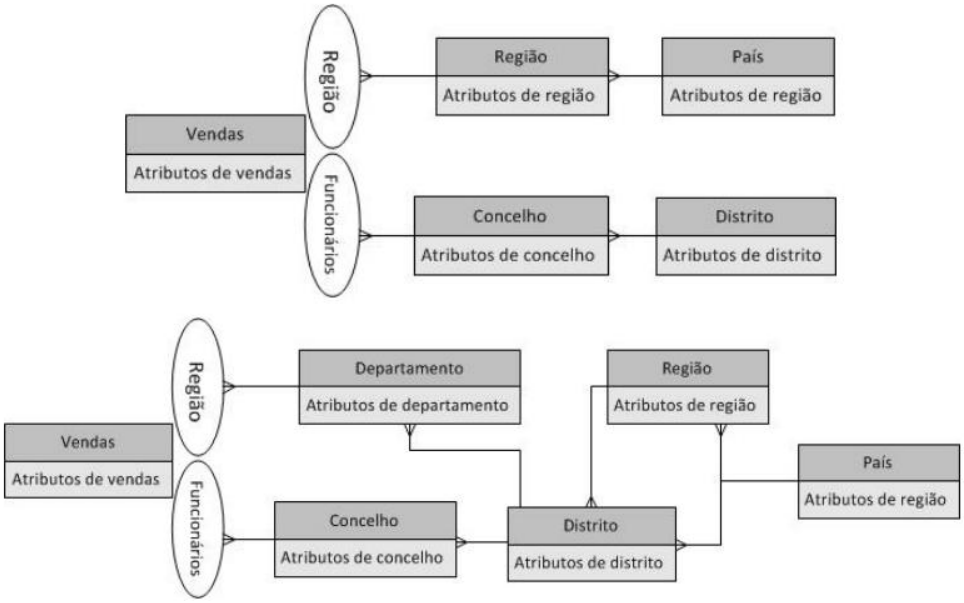


Figura 9 – Hierarquia paralela dependente (em cima), e hierarquia paralela independente (em baixo) (Bruno Oliveira e Orlando Belo, 2012)

2.2.6. Processo ETL

O processo Extração, Transformação e Carregamento de Dados (ETL) é o nome dado ao processo de extração dos dados das fontes de dados e carregamento no DW depois de algumas transformações.

A extração tem como definição a aquisição de dados de um sistema para poderem ser utilizados num sistema destino. As extrações de dados iniciam-se com uma primeira extração, estando o DW vazio, e depois são agendadas extrações incrementais para o DW que podem ser executadas uma vez por dia, duas vezes por dia ou até mesmo sempre que o sistema OLTP é atualizado (Lane, 2002).

Os dados das fontes devem ser extraídos de forma a não desregular a performance dos sistemas OLTP que, como são sistemas transacionais, são desenvolvidos para extrações de dados mais reduzidas. Estas extrações são realizadas regularmente através de conexões do sistema ETL às fontes de dados,

dado que o processo pode estar agendado no sistema OLTP ou no sistema OLAP. Outra forma de extração é o leitor log para detetar as alterações realizadas no OLTP, ou a utilização da linguagem SQL (*Structured Query Language*) para a criação de *Triggers* que extraem os dados logo que há uma atualização de dados nos sistemas transacionais. Uma das tarefas mais delongadas do processo ETL é o desenho do modelo de extração que vai ser aplicado pois as fontes de dados podem ter estruturas complexas e pode haver falta de documentação (Santos & Ramos, 2009).

No processo ETL, depois dos dados extraídos, estes ficam numa área de estágio (*Data Staging Area* - DSA), aqui os dados sofrem transformações e correções de forma a eliminar inconsistências como a duplicação de dados, erros e dados em falta (Rahm & Do, 2000). A transformação de dados, para além de conter o processo de mapeamento dos dados no sistema destino, agrupa também uma série de tarefas que podem ser utilizadas para manter a respetiva integridade dos dados permitindo o sistema cumprir os objetivos para o qual foi desenvolvido. As tarefas presentes no processo de transformação representam filtragem de dados para não serem armazenados dados sem relevância no contexto, a eliminação de dados duplicados, a substituição das chaves primárias pelas *surrogate keys* e a correção de dados entre outras (Kimball, Ralph & Caserta, 2004).

O grande objetivo deste processo de transformação é a melhorar a qualidade dos dados e manter a integridade do sistema destino. Analisar e compreender os dados é importante para detetar o tipo de inconsistências existentes que devem ser resolvidas. Depois de realizadas e testadas todas as transformações do processo ETL, chega a parte em que os dados são carregados no sistema destino. As reflexões mais relevantes dos carregamentos de dados é a firmeza com que são concretizados para não produzirem um embate negativo no sistema destino.

A alocação do sistema ETL pode ser no servidor onde se encontram os sistemas transacionais, pode ser no servidor dos sistemas OLAP ou num próprio servidor de sistemas ETL, sendo este terceiro mais dispendioso porque é o único que poderá exigir da existência de mais licenças de software, apesar de também ser o que gasta menos recursos de processamento (Rainardi, 2008).

Uma das componentes importantes no desenvolvimento de sistemas de *Data Warehousing* e do seu processo ETL são os Metadados. Os Metadados correspondem à definição dos dados, e são relevantes para compreender a estrutura e o significado dos dados tanto do *Data Warehousing* como das fontes de dados e no processo ETL. Estes podem ser técnicos ou de negócio. Os Metadados de negócio têm importância para os gestores e os Metadados técnicos ajudam os desenvolvedores de sistemas de armazém de dados a compreender a estrutura dos dados e as transformações e mapeamento dos dados

durante o processo ETL, ajudam também na parte técnica de descrição e compreensão dos dados (Inmon, 2005).

2.2.7. Processamento Analítico

Depois da organização dos dados num DW, é necessário construir infraestruturas de processamento analítico para explorar os dados. Os sistemas OLAP incorporam factos, dimensões, cubos e hierarquias que possibilitam a exploração e visualização dos dados.

Na literatura são aceites três arquiteturas destes sistemas, a arquitetura *Multidimensional OLAP* (MOLAP) armazena os dados em cubos multidimensionais e tem como principal vantagem o excelente desempenho e o rápido processamento. Na arquitetura *Relational OLAP* (ROLAP) é utilizado um Sistema de Gestão de Base de Dados relacional para gerir os dados que são analisados. A vantagem do procedimento ROLAP é a capacidade de armazenamento, mas como incorpora consultas SQL à base de dados relacional torna mais lento o processamento dos dados. A terceira arquitetura é a *Hybrid OLAP* (HOLAP), e combina as duas em cima tirando o maior proveito das duas, ou seja, a estabilidade da arquitetura ROLAP e a velocidade de processamento da arquitetura MOLAP.

Os servidores OLAP proporcionam um envolvente favorável para uma análise interativa sobre os cubos (Rainardi, 2008).

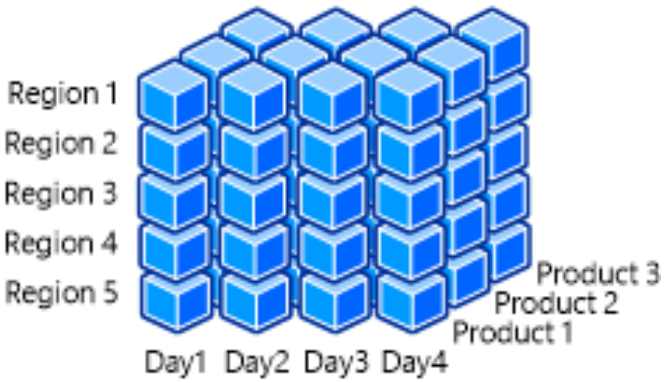


Figura 10 – Exemplo de Cubo OLAP (Retirada da página WEB: docs.microsoft.com)

Quando se fala em manipulação de Cubos, temos certas operações que podem ser executadas para navegar nos dados. A operação *Drill-Down* ou *Roll-Down* é usada para detalhar os dados pormenorizando as análises. Outra operação chamada *Roll-Up* ou *Drill-Up* faz o inverso da *Drill Down*, ou seja, pega nos dados detalhados e agrega-os.

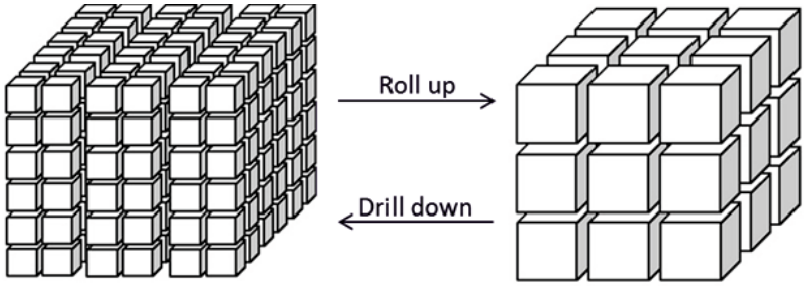


Figura 11 – Exemplo das operações Roll-Up e Drill-Down. Retirada de (Alfred Bolt, 2015)

A operação *Slice and Dice* permite restringir a visualização num sub cubo através da escolha de critérios.

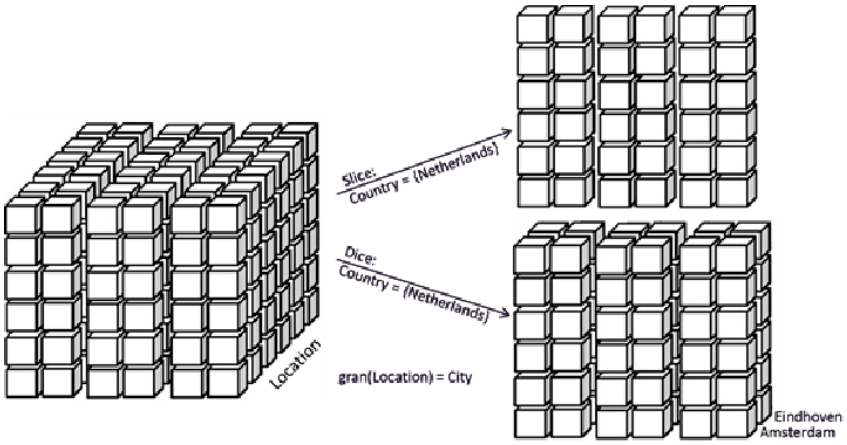


Figura 12 – Exemplo da operação Slice and Dice (Retirada de (Alfred Bolt, 2015))

Por fim, a operação *Pivot* ou *Rotate* permite a rotação dos eixos de visualização.

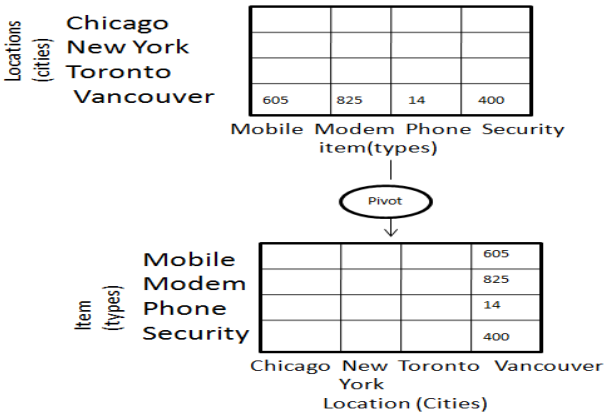


Figura 13 – Exemplo da operação Pivot (Retirada da tutorialpoint)

2.2.8. Visualização

Um sistema de *Business Intelligence* precisa de uma interface para os gestores de forma a que estes possam aceder e visualizar os dados para conseguirem compreender e explorar a informação, extraindo conhecimento e dando a garantia de que o sistema forneceu suporte nas tomadas de decisão. Assim as aplicações de *front-end* servem de interface com o utilizador e, segundo Turban (2008), esta componente é a última de uma arquitetura de *Business Intelligence*. A interface com o utilizador tem muitas formas de representação, uma delas é o formato de *dashboard* para uma análise de medidas chave de desempenho (KPI's), e a informação dos indicadores de performance é essencial para as tomadas de decisão empresariais.

Os *dashboards* têm como elementos principais os gráficos. Existem muitos tipos diferentes de gráficos que podem ser selecionados para representar informações dentro de um *dashboard*. A boa escolha de gráficos é importante para promover visualizações intuitivas aos utilizadores. Para isso podem utilizar-se gráficos de barras, de linhas, de áreas, circulares, tabelas entre outros.

Os sistemas de BI usam os indicadores de performance para compreender a realidade atual do negócio, mas a representação gráfica destes indicadores deve acomodar alguns princípios. Segundo Caldeira (2012) existem regras comuns para a construção e desenvolvimento de gráficos:

- Utilização de Informação adicional para completar a análise
- Utilização da cor mais forte para o indicador principal
- Uso do menor número de cores possível
- Atribuição de títulos e legendas
- Uso, se possível, de rótulos de dados

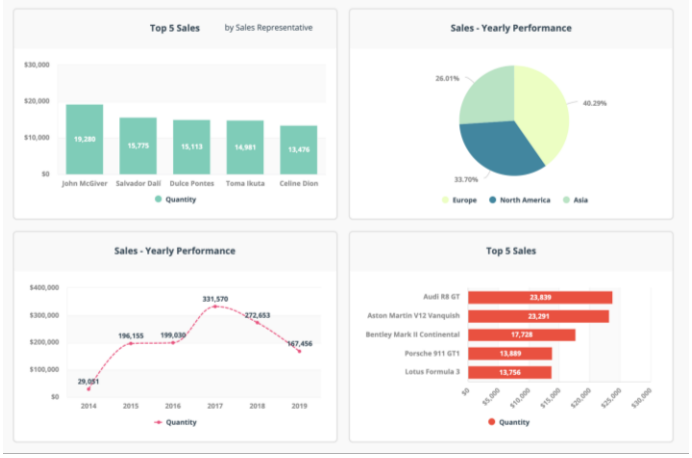


Figura 14 – Exemplo de um Dashboard (Retirado da ClicData)

Dentro da interface gráfica, é registado como boa prática o uso de filtros para o utilizador conseguir manipular a informação visível no *dashboard*.

Para concluir, o uso de aplicações de *front-end* no desenvolvimento de sistemas de BI só corresponde a cerca de 15% dos projetos, sendo que 85% do trabalho é focado na construção do sistema de DW e nos processos de Extração, Transformação e Carregamento de dados.

2.3. Sistemas de Recomendação

2.3.1. Motivação

Depois de um Sistema de *Data Warehousing* completo, o objetivo principal é focar a estrutura de dados na construção de relatórios que permitam uma análise descritiva por parte dos gestores, dando assim suporte nas suas decisões. Mas como os Sistemas de *Data Warehousing* têm uma estrutura de dados organizada, é favorável a implementação de algoritmos de aprendizagem de máquina, ou seja, técnicas de *Data Mining*, *Machine Learning* e inteligência artificial podem ser uma tecnologia integrante num sistema de *Business Inteligence*.

Todos os dias, pessoas fornecem recomendações que são confiadas por decisores. Estas podem ser críticas em livros ou até um simples "passa a palavra". A tecnologia usada em técnicas de extração de conhecimento de dados permite a criação de recomendações para utilizadores baseadas na experiência de outras decisões. Estes softwares são conhecidos como Sistemas de Recomendação, e as suas funções são compreender as necessidades e os problemas dos utilizadores para gerarem sugestões úteis (Resnick and Varian, 1997).

Existem dois tipos de Sistemas de Recomendação, os genéricos que elaboram recomendações iguais para todos os utilizadores, como por exemplo recomendar os dez artigos mais vendidos, e os tipos de sistemas personalizáveis, que são recomendações com conspeção nas avaliações dos utilizadores.

Os Sistemas de Recomendação estão completamente ligados às preferências dos utilizadores de forma a que o feedback dado por estes seja fundamental para o bom funcionamento destes sistemas. Existem várias categorias para se recolher preferências, uma delas é a forma explícita em que é expressa a admiração ou a insatisfação de um produto com base, muitas vezes, em preenchimentos de formulários utilizando escalas (1-5 por exemplo). Outra forma de recolha de feedback é a categoria implícita, em que se recolhe as preferências com base nos comportamentos dos utilizadores. Estes dois

tipos de recolhas de preferências podem ser combinados de forma a aproveitar os pontos fortes de cada um (Isinkaye, Folajimi and Ojokoh, 2015).

2.3.2. Objetivos dos Sistemas de Recomendação

O Sistemas de Recomendação são desenvolvidos para ajudaram em alguns objetivos tendo algumas tarefas específicas e importantes:

- Aumento das Vendas: o principal objetivo de um sistema de recomendação é aumentar as vendas dos artigos sugerindo mais alternativas para os utilizadores.
- Diversificação: um dos objetivos é diversificar a venda dos artigos incentivando os utilizadores a escolherem artigos não tanto populares, mas que podem ter sucesso.
- Satisfação e Fidelização: a satisfação do utilizador pode ser cada vez maior pois os artigos recomendados podem ser do seu agrado devido à capacidade de reconhecimento de histórico por parte do sistema.
- Identificar as Necessidades: Perceber as necessidades dos utilizadores é de extrema importância pois ajuda a estabelecer serviços importantes para os mesmos, e também ajuda a corrigir as roturas de stock.
- Encontrar todos ou alguns artigos adequados: o desenvolvimento de recomendações relevantes que satisfazem as necessidades dos utilizadores.
- Destacar Artigos: baseado no histórico do utilizador, alguns artigos devem ser destacados.

Outras tarefas dos Sistemas de Recomendação é a influência de terceiros, ou seja, fazer com que outros adquiram determinados itens e proporcionando satisfação ao utilizador pois este poderá sentir que está a ajudar terceiros e também poderá sentir que está a expressar o seu conhecimento, melhorando o seu perfil e criando mais confiança (Ricci 2011, Herlocker, 2004).

2.3.3. Técnicas de Recomendação

2.3.3.1. Filtragem Colaborativa

A Filtragem Colaborativa é uma técnica que gera recomendações com base nos gostos semelhantes dos utilizadores. Neste método o histórico é usado para calcular a semelhança entre

utilizadores, uma vez que contém as avaliações que os utilizadores atribuíram a cada artigo. Caso um grupo de utilizadores partilhe a mesma opinião sobre um conjunto de artigos, é provável que partilhem a mesma opinião sobre outros artigos (Herlocker, 2004).

Então, a Filtragem Colaborativa, sendo uma das técnicas mais populares na implementação de sistemas de recomendação, é baseada no cálculo das proximidades de preferências de utilizadores, minerando similaridades entre artigos e entre utilizadores. Esta técnica divide-se em duas abordagens distintas, uma filtragem colaborativa baseada em memória ou em modelos. A abordagem baseada em memória estabelece correlações entre produtos e utilizadores podendo, desta forma, ser baseada no utilizador ou no produto (Isinkaye, Folajimi and Ojokoh, 2015). A forma baseada no utilizador procura identificar utilizadores com semelhanças nos seus interesses e recomendar artigos ainda não avaliados por um utilizador, mas já bem classificados pelos seus semelhantes utilizadores. No desenvolvimento desta abordagem é necessário criar a matriz de utilizador-produto que contém as classificações dos produtos dadas pelos utilizadores. Depois de construída a matriz utilizam-se algoritmos para encontrar os utilizadores similares, e finalmente, desenvolvem-se recomendações dos produtos mais frequentes dos utilizadores similares prevendo-se um bom desempenho no utilizador que ainda não classificou o produto (Isinkaye, Folajimi and Ojokoh, 2015).

Quando o número de utilizadores é demasiado elevado, normalmente é usada a abordagem baseada em produtos, começando por desenvolver a matriz Produto-Produto através de vários algoritmos, calculando a similaridade de um artigo com outros artigos escolhidos por um utilizador e prevendo o seu desempenho para ser elaborada a recomendação.

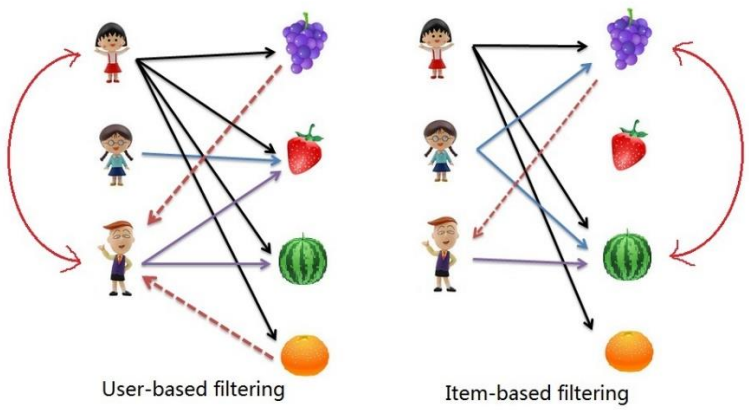


Figura 15 – Filtragem Colaborativa baseada em memória (Retirada de https://www.devmedia.com.br/apache-spark-como-criar-um-mecanismo-de-sugestao-de-produtos/33459)

A filtragem Colaborativa baseada em modelos utiliza técnicas de *Machine Learning* para criar modelos capazes de representar os interesses do utilizador, ou seja, desenvolve padrões das iterações dos utilizadores com os produtos para gerar previsões automáticas. Algumas técnicas mais utilizadas são por exemplo as Regras Associativas e o *Clustering* (Isinkaye, Folajimi and Ojokoh, 2015). São construídos padrões de utilizador/artigo que geram recomendações automáticas.

2.3.3.2. Baseado em Conteúdo

Os sistemas baseados em conteúdo formam recomendações que correspondem aos interesses dos utilizadores e são integrados por 3 elementos. O elemento de análise de conteúdo que é responsável por tratar os dados e normalizá-los. A componente de perfil de utilizador que é responsável por fazer uma análise ao histórico do utilizador e detetar a que perfil este utilizador pertence. E o último elemento que é responsável por escoar os produtos mais importantes para esse utilizador podendo gerar, desta forma, as recomendações.

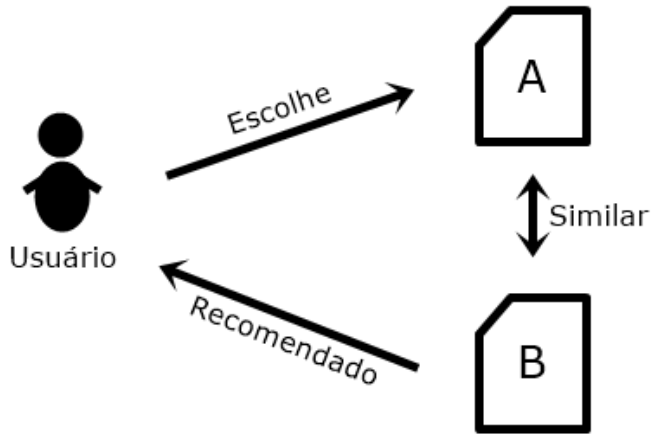


Figura 16 – Exemplo de uma recomendação baseada em Conteudo (Rolim,2017)

2.3.3.3. Híbridos

Os sistemas de recomendação híbridos juntam as melhores caraterísticas dos sistemas de recomendação para melhorar o desempenho. Como todo o tipo de sistemas tem os seus respetivos

problemas e vulnerabilidades, se forem usados hibridamente é possível reduzir essas limitações (Burke 2002).

Existem várias formas de desenvolver sistemas de recomendação híbridos, construindo, por exemplo, modelos que incorporem várias caraterísticas de sistemas distintos ou até desenvolver os modelos isoladamente e conciliar ambas as previsões.

Nos sistemas híbridos existem várias técnicas para combinar diferentes sistemas, uma delas é a forma alterada, em que o sistema altera os diferentes tipos de recomendações consoante a situação atual. Outra, é a forma misturada, em que o sistema apresenta as recomendações de diferentes técnicas ao mesmo tempo. Existe também a técnica de combinação de caraterísticas, em que o sistema usa *outputs* de técnicas de recomendação para serem *inputs* de outras técnicas (Burke 2002).

O método híbrido apresenta as vantagens de superar as limitações das técnicas de recomendação e proporciona mais flexibilidade, apesar de serem mais complexos e dispendiosos.

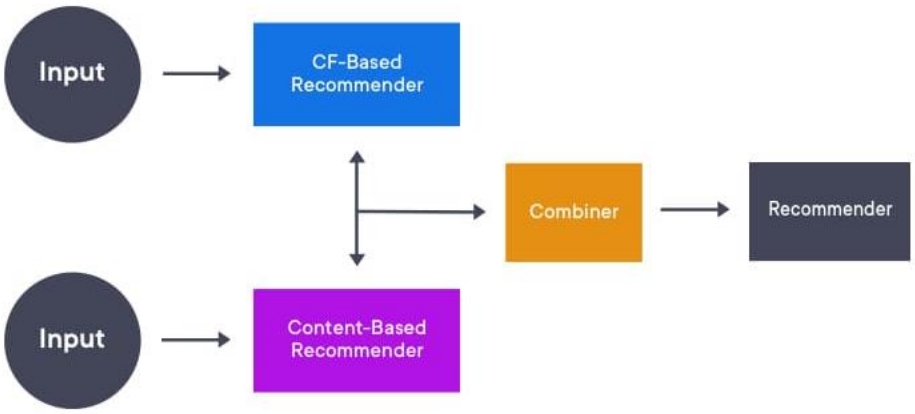


Figura 17 – Arquitetura de um Sistema de Recomendação Hibrido (Retirada de https://www.netsolutions.com/insights/building-recommendation-engine/ Autor: Lalit Singla, 2019)

2.3.4. Medidas de Similaridade

Para avaliar a similaridade entre dois objetos nos sistemas de recomendação deve-se escolher as medidas de similaridade para que o impacto na qualidade seja o melhor possível.

Algumas das medidas de similaridade mais usadas são a similaridade do Cosseno e o coeficiente de correlação de Pearson.

Similaridade do Cosseno:

$$cosine\_sim_{(u,v)} = \frac{\sum_{i \in I_{uv}} r_{ui} . r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} . \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}}$$

Coeficiente de Correlação de Pearson:

$$pearson\_sim_{(u,v)} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u) . (r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)^2} . \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \mu_v)^2}}$$

Onde:

- μ_u, média de classificações do utilizador (ou artigo) u
- μ_v, média de classificações do utilizador (ou artigo) v
- r_{ui}, a classificação do utilizador u (ou artigo) com o artigo (ou utilizador) i
- r_{vi}, a classificação do utilizador v (ou artigo) com o artigo (ou utilizador) i
- I_{uv}, o conjunto de todos os artigos avaliados pelos utilizadores u e v (ou o conjunto de todos os utilizadores que avaliaram o artigo i e j)

2.3.5. Algoritmos de *Machine Learning*

Machine Learning ou Máquinas a Aprender, é definido como uma área inserida na inteligência artificial que explora algoritmos estatísticos e matemáticos que visam o desenvolvimento de modelos que têm como input um conjunto de dados, e com base em determinados padrões são gerados os seus outputs, isto é, a área de *Machine Learning* é responsável pelo desenvolvimento de modelos analíticos, através de algoritmos que aprendem autonomamente a partir de dados e através de recursos computacionais, permitindo que as máquinas encontrem insights oclusivos sem serem explicitamente programados para procurar algo específico.

Durante o desenvolvimento dos modelos de *Machine Learning*, os conjuntos de dados são divididos em conjuntos de treino e teste. Os conjuntos de Treino são usados para treinar o modelo, ou seja, ajustar o modelo de modo interativo para produzir os melhores resultados de previsão. O conjunto de teste é usado para avaliar a performance do modelo com dados exteriores ao conjunto de treino.

Neste subcapítulo são abordados alguns algoritmos de Aprendizagem de Máquina que foram utilizados ou explorados no projeto desta dissertação.

2.3.5.1. Método Baseado na Distribuição Normal

O método baseado na distribuição normal baseia-se num algoritmo que prevê uma classificação aleatória com base na distribuição normal do conjunto de treino, usando a estimativa de probabilidade máxima.

A previsão $\widehat{r_{ui}}$ é gerada a partir da distribuição normal $N\left(\hat{\mu}, \widehat{\sigma^2}\right)$, onde $\hat{\mu}$ e $\hat{\sigma}$ são estimados com o conjunto de dados de treino onde:

$$\hat{\mu} = \frac{1}{N_t}\sum_{1}^{N_t} r_{ui}$$

$$\hat{\sigma} = \sqrt{\sum_{1}^{N_t} \frac{(r_{ui} - \hat{\mu})^2}{N_t}}$$

Sendo:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- $\hat{\mu}$, estimativa da média das classificações
- $\hat{\sigma}$, estimativa do desvio-padrão das classificações
- N_t, tamanho do conjunto de treino
- r_{ui}, a classificação do utilizador u com o artigo i

2.3.5.2. Método de *Base Line*

O método de *Base Line* é o algoritmo que prevê a estimativa da *Base Line* para determinado utilizador e artigo.

$$\widehat{r_{ui}} = \widehat{b_{ui}} = \boldsymbol{\mu} + b_u + b_i$$

Sendo:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- μ, média das classificações
- b_u, viés do utilizador u em relação a média
- b_i, viés do artigo i em relação a média

2.3.5.3. Métodos baseados em kNN

Relativamente a filtragem colaborativa, o algoritmo kNN intua que se os utilizadores têm os mesmos comportamentos e preferências em algum artigo que foi apreciado por um conjunto de utilizadores com as mesmas preferências então existe uma grande hipótese de que o utilizador vai ter uma boa avaliação nesse artigo. (Wang, 2017)

As tarefas principais deste algoritmo passam por determinar os k utilizadores vizinhos do utilizador "u", e depois arquitetar uma abordagem para prever a classificação dos artigos não avaliados por esse utilizador com base nas avaliações dos k vizinhos mais próximos. Finalmente, seleciona como recomendações os top artigos em que é prevista a maior classificação pelo utilizador "u". Este algoritmo pode ser calculado desta forma com a semelhança de utilizador-utilizador, mas também pela semelhança de artigo-artigo.

A predição é definida da seguinte forma:

$$\widehat{r_{ui}} = \frac{\sum_i^k sim(u,v).r_{vi}}{\sum_i^k sim(u,v)}$$

Sendo:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- k, número de vizinhos mais próximos
- sim(u,v), similaridade do utilizador u com o utilizador v, ou do artigo u com o artigo v
- r_{vi}, a classificação do utilizador v com o artigo i, ou do artigo v com o utilizador i

Este algoritmo pode ser aplicado considerando as avaliações médias de cada utilizador:

$$\widehat{r_{ui}} = \mu_u + \frac{\sum_i^k sim(u,v).(r_{vi} - \mu_v)}{\sum_i^k sim(u,v)}$$

Sendo:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- k, número de vizinhos mais próximos
- sim(u,v), similaridade do utilizador u com o utilizador v, ou do artigo u com o artigo v

- r_{vi}, a classificação do utilizador v com o artigo i, ou do artigo v com o utilizador i
- μ_v, a média de classificações do utilizador v, ou do artigo v

Este algoritmo pode também ser aplicado com a normalização do Z-SCORE de cada utilizador:

$$\widehat{r_{ui}} = \mu_u + \sigma_u \frac{\sum_i^k sim(u,v).(r_{vi} - \mu_v)/\sigma_v}{\sum_i^k sim(u,v)}$$

Sendo:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- k, número de vizinhos mais próximos
- sim(u,v), similaridade do utilizador u com o utilizador v, ou do artigo u com o artigo v
- r_{vi}, a classificação do utilizador v com o artigo i, ou do artigo v com o utilizador i
- μ_v, a média de classificações do utilizador v, ou do artigo v
- μ_u, a média de classificações do utilizador u, ou do artigo u
- σ_u, o desvio-padrão das classificações do utilizador u, ou do artigo u

O algoritmo pode também ser aplicado tendo em consideração uma classificação do algoritmo da *Base Line*:

$$\widehat{r_{ui}} = b_{ui} + \frac{\sum_i^k sim(u,v).(r_{vi} - b_{vi})}{\sum_i^k sim(u,v)}$$

Sendo:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- k, número de vizinhos mais próximos
- sim(u,v), similaridade do utilizador u com o utilizador v, ou do artigo u com o artigo v
- r_{vi}, a classificação do utilizador v com o artigo i, ou do artigo v com o utilizador i
- μ_v, a média de classificações do utilizador v, ou do artigo v
- b_{ui} , a previsão baseada na linha base do utilizador u com o artigo i, ou do artigo u com o utilizador i

2.3.5.4. Algoritmos baseados em Fatorização de Matrizes

Um dos algoritmos mais famosos baseado em fatorização de matrizes é o algoritmo de Decomposição em Valores Singulares, pode-se utilizar *Base Lines* onde $\widehat{r_{ui}}$:

$$\widehat{r_{ui}} = \boldsymbol{\mu} + b_u + b_i + q_i^T p_u$$

ou, usando simplesmente a fatorização da matriz probabilística:

$$\widehat{r_{ui}} = q_i^T p_u$$

o objetivo é reduzir o erro quadrático com a descida de gradiente:

$$b_i <- b_i + \gamma(e_{ui} - \lambda b_i)$$
$$b_u <- b_u + \gamma(e_{ui} - \lambda b_u)$$
$$p_u <- p_u + \gamma(e_{ui} - \lambda p_u)$$
$$q_i <- q_i + \gamma(e_{ui} - \lambda q_i)$$
$$\sum_{1}^{N_t} (r_{ui} - \widehat{r_{ui}})^2 + \lambda(b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2)$$

Onde:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- $e_{ui} = r_{ui} - \widehat{r_{ui}}$
- γ, é o termo de regularização
- λ, é a taxa de aprendizagem
- p_u, é o fator do utilizador
- q_i, é o factor do artigo
- $\boldsymbol{\mu}$, a média de classificações
- b_u, viés do utilizador u em relação a média
- b_i, viés do artigo i em relação a média
- N_t, tamanho do conjunto de treino

O algoritmo SVD apresenta uma extensão se considerarmos as calcificações implícitas, este algoritmo é conhecido como SVD ++ e a sua predição é definida como:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

Onde:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- p_u, é o fator do utilizador
- q_i, é o factor do artigo
- $\boldsymbol{\mu}$, a média de classificações
- b_u, viés do utilizador u em relação a média
- b_i, viés do artigo i em relação a média
- y_j, fatores do artigo j
- I_u, o conjunto de todos os artigos avaliados pelo utilizador u

Outro algoritmo semelhante ao SVD é o de Fatorização de Matrizes Não Negativas, ou seja, os fatores do utilizador e do artigo são sempre positivos:

$$\hat{r}_{ui} = q_i^T p_u$$

Onde o procedimento de otimização diz respeito à descida do gradiente estocástico com a garantia da não negatividade dos fatores:

$$p_{uf} \leftarrow p_{uf} \cdot \frac{\sum_{i \in I_u} q_{if} \cdot r_{ui}}{\sum_{i \in I_u} q_{if} \cdot \hat{r}_{ui} + \lambda_u |I_u| p_{uf}}$$

$$q_{if} \leftarrow q_{if} \cdot \frac{\sum_{u \in U_i} p_{uf} \cdot r_{ui}}{\sum_{u \in U_i} p_{uf} \cdot \hat{r}_{ui} + \lambda_i |U_i| q_{if}}$$

Onde:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i

- λ, é o termo de regularização
- p_u, é o fator do utilizador
- q_i, é o factor do artigo
- b_u, viés do utilizador u em relação à média
- b_i, viés do artigo i em relação à média
- N_t, tamanho do conjunto de treino
- r_{ui}, é a verdadeira avaliação do utilizador u para o artigo i

2.3.5.5. Algoritmos *Slope-One* e *Co-Clustering*

O *slope-one* é um algoritmo de filtragem colaborativa preciso:

$$\hat{r}_{ui} = \mu_u + \frac{1}{|R_i(u)|} \sum_{j \in R_i(u)} dev(i,j)$$

Onde:

$$dev(i,j) = \frac{1}{|U_{ij}|} \sum_{u \in U_{ij}} r_{ui} - r_{uj}$$

- dev(i,j), é a diferença entre a média das avaliações dos artigos i e dos artigos j
- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- u_u, é a média de classificações do utilizador u
- $R_i(u)$, é o conjunto de artigos que são relevantes, ou seja, os artigos j avaliados pelo utilizador u que tem pelo menos um utilizador comum com o artigo i
- U_{ij}, O conjunto de todos os utilizadores que avaliaram os artigos i e j
- r_{ui}, é a verdadeira avaliação do utilizador u para o artigo i

O algoritmo *Co-Clustering* é um algoritmo de filtragem colaborativa onde os artigos e os utilizadores são atribuídos a alguns clusters, onde a predição do artigo u com o utilizador i é representada por:

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i})$$

Onde:

- $\overline{C_{ui}}$, é a classificação média do co-cluster do artigo i e utilizador u
- $\overline{C_u}$, é a classificação média dos clusters de u
- $\overline{C_i}$, é a classificação média dos clusters de i
- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- u_u, é a média de classificações do utilizador u
- u_i, é a média de classificações do artigo i

2.3.6. *Performance* do Sistema

As *performances* dos sistemas de recomendação podem ser avaliadas por inúmeras métricas. A escolha dessas medidas deve ser elaborada de forma cuidada e justificada com os objetivos do sistema. Normalmente, são usadas algumas métricas estatísticas de precisão, como por exemplo, o erro médio absoluto (MAE), isto é, quanto menor for o erro médio absoluto mais preciso é o sistema. A fórmula é a seguinte:

$$MAE = \frac{1}{|\hat{R}|} \sum_{\widehat{r_{ui}} \in \hat{R}} |r_{ui} - \hat{r}_{ui}|$$

Outra métrica muito utilizada é a raiz quadrada do erro quadrático médio (RMSE) que é parecido com o erro médio absoluto mais dá mais peso e importância ao erro absoluto. A fórmula é a seguinte:

$$RMSE = \sqrt{\frac{1}{|\hat{R}|} \sum_{\widehat{r_{ui}} \in \hat{R}} (r_{ui} - \hat{r}_{ui})^2}$$

Onde:

- $\widehat{r_{ui}}$, a previsão de classificação do utilizador u com o artigo i
- r_{ui}, a classificação verdadeira do utilizador u com o artigo i
- $\hat{R}$, é o conjunto de todas as classificações

Ambas estas métricas expressam o erro médio das predições do modelo em unidades da variável de interesse. São pontuações de orientação negativa, ou seja, quanto mais baixo for valor melhor são os valores. A grande diferença entre estas duas métricas é que, a RMSE, ao utilizar a raiz torna-se mais útil quando os grandes erros são particularmente indesejáveis.

2.3.7. Considerações dos Sistemas de Recomendação

Um dos problemas mais conhecidos dos sistemas de recomendação é o *Cold-Start*. O problema surge sempre que há um novo utilizador no sistema, pois este não compõe nenhum histórico de avaliações anterior e fica difícil desenvolver novas recomendações que possam satisfazer este utilizador. O mesmo problema acontece para novos artigos, como não tem avaliações de utilizadores torna-se difícil incluir os artigos nas recomendações (Hernando, 2016). Outro problema dos sistemas de recomendação é a falta de quantidade de avaliações que é feita a determinados artigos sendo que os sucessos das recomendações colaborativas são dependentes das críticas e avaliações dos utilizadores.

Muitos utilizadores estão sempre a mudar as suas necessidades e daí as suas preferências por artigos, os sistemas de recomendação podem ter dificuldades a analisar as preferências mais atuais dos utilizadores. Quando existem utilizadores que decidem fazer avaliações que não correspondem à realidade, estes provocam uma diminuição no desempenho do sistema e na capacidade das recomendações. Estes ataques podem ser simplesmente por questões maliciosas ou de concorrência.

A técnica de recomendação baseada em conteúdo utiliza as caraterísticas dos produtos para saber se são correspondentes aos interesses dos utilizadores. Quando o produto não tem as caraterísticas bem definidas o processo de recomendação perde performance. Relativamente aos novos utilizadores, enquanto estes não integrarem um forte histórico de preferências o sistema não conseguirá gerar as melhores recomendações para eles por não conhecer as suas preferências. Os sistemas baseados em conteúdo dificilmente recomendam algo inesperado pois acompanham as preferências e as caraterísticas dos produtos (Lops, de Gemmis and Semeraro, 2011). Os sistemas de recomendação baseados em conteúdo são capazes de recomendar produtos que não possuem classificações pois o importante são as caraterísticas do produto e não as avaliações. Estes sistemas incorporam bastante transparência e o utilizador não está dependente de terceiros.

Na técnica de Filtragem Colaborativa, o problema do *Cold-Start* adequa-se quase a 100% pois se existirem produtos novos ou utilizadores novos é difícil gerar recomendações devido à falta de histórico nas avaliações. Outro problema são os utilizadores incomuns, ou seja, utilizadores com necessidades

muito diferentes dos restantes, onde os sistemas têm dificuldade em encontrar semelhanças com os outros utilizadores. É sempre necessário, neste sistema, muitas classificações dos utilizadores, neste caso o problema está na dependência do sucesso das avaliações dos utilizadores (Su and Khoshgoftaar, 2009; Isinkaye, Folajimi and Ojokoh, 2015).

Ao contrário do sistema baseado em conteúdo, este sistema pode ter recomendações inesperadas e do agrado do utilizador.

2.3.8. Síntese

Este capítulo teve como fundamento a investigação dos temas associados à matéria utilizada durante o desenvolvimento do projeto. Tendo em conta a estrutura desta dissertação, chegou-se à conclusão que foram estudadas as metodologias e arquiteturas elementares capazes de trazer todo o conhecimento necessário para a realização da parte prática do projeto. Assim, nos próximos dois capítulos, serão abordados os requisitos e o desenvolvimento prático deste, aplicando o conhecimento estudado na investigação dos temas de *Business Intelligence* e sistemas de recomendação.

3. O Projeto Foreva

3.1. Foreva, Marca de Retalho do Grupo Kyaia

A Kyaia foi fundada em 1984 e lidera um agregado empresarial com mais de 600 colaboradores, expondo um volume de negócio de 55 milhões de euros. O modelo de negócio do grupo Kyaia engloba a produção do calçado, áreas de distribuição, o retalho e investigação e desenvolvimento em novas tecnologias informáticas. Este projeto foi desenvolvido com o objetivo de trazer vantagens competitivas ao ramo do retalho do grupo, as lojas Foreva.

A Foreva nasceu em 1984 com a sua primeira loja na Rua Guerra Junqueiro em Lisboa. Hoje a marca apresenta uma elite de lojas em todos os grandes centros urbanos e está presente na globalidade dos centros comerciais de Portugal. A área Metropolitana de Lisboa é constituída por 5 lojas Foreva, outras 5 estão presentes no centro do pais, na zona de Aveiro, Viseu, Torres Vedra e Leiria. O Norte do País é constituído por 5 lojas Foreva, as regiões autónomas têm 2 e o Sul do País tem 3.

A estrutura da Foreva é constituída por um armazém de calçado onde é armazenado o seu stock para posteriormente ser distribuído pelas lojas da marca, as transferências de stock podem ser feitas do armazém para as lojas e vice-versa, ou entre as próprias lojas.

Para além das 20 lojas e o armazém, a administração da Foreva gere os stocks de uma loja da Fly London (Grupo Kyaia) no Porto, de uma loja da Overcube (Grupo Kyaia) em Lisboa, e 4 outlets espalhados pelo país. Ao todo são 27 pontos de stock físicos.

Os artigos da Foreva estão disponíveis numa plataforma de vendas online onde os clientes podem encomendar o produto pretendido dentro da gama de stock existente (stock do armazém e lojas).

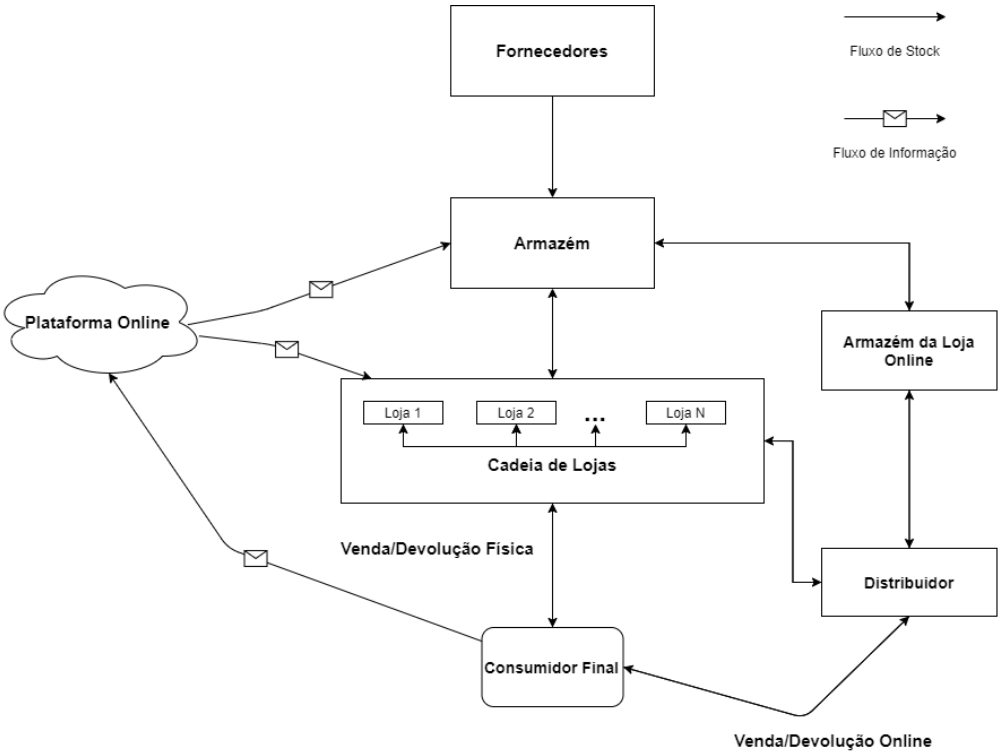


Figura 18 – Fluxo de informação e de Stock da Foreva

3.2. Contextualização do Problema e Definição dos Objetivos

Com o avanço das tecnologias, as empresas começam cada vez mais a dar atenção à importância da informação e do conhecimento, passa-se o mesmo no setor do retalho do calçado. A Foreva começou a procurar estratégias de maior sucesso, melhoria na satisfação dos clientes e nos processos de funcionamento, assim como a eficácia na escolha de produtos e no desempenho organizacional. Para isso é necessário tirar o maior proveito possível das fontes de informação disponíveis, usufruindo, desta forma, dos benefícios da tecnologia.

No fim de várias reuniões com os gestores da empresa, foram elaborados os objetivos do projeto. Projeto este em que o principal fundamento é a criação de uma plataforma de *Business Intelligence* para a marca Foreva, que seja capaz de saciar toda a escassez de informação que não chega às mãos dos gestores em tempo útil.

3.2.1. Análise de Vendas

O problema identificado, que levou ao primeiro objetivo, foi a forma de acesso aos dados das vendas por parte dos gestores. A Foreva usufrui de uma aplicação de funcionamento que permite a transferência de stocks entre as lojas e o armazém, e também a analise de vendas diárias. Contudo, o trabalho dos gestores de retirar da plataforma as vendas diárias para estudar alguns indicadores de performance considerados chave torna-se bastante exaustivo, pois têm de o fazer todos os dias para aglomerar os dados em ficheiros Excel de forma a poderem proceder ao desenvolvimento da análise. Assim foram definidos uma série de KPI's que são importantes estarem sempre atualizados numa nova plataforma em que os gestores conseguissem aceder para analisar o percurso anual da Foreva, no que diz respeito às vendas das lojas. Os KPI's definidos para este objetivo foram os seguintes:

- Variação Homóloga da Quantidade de Pares Vendidos;
- Variação Homóloga do Volume de Vendas;
- Variação Homóloga da diferença entre o Volume de Vendas e o Custo de Vendas;
- Variação Homóloga da diferença entre o Volume de Vendas, o Custo de Vendas e o Volume de Devoluções;
- Variação Homóloga do Volume de Devoluções;
- Variação Homóloga do Peso das Devoluções nas Vendas;
- Variação Homóloga da Margem de Lucro;
- Variação Homóloga do Preço Médio de Venda;
- Variação Homóloga do Valor Médio por Transação;
- Variação Homóloga da Quantidade Média por Transação;
- Variação Homóloga do *Cross Selling*;
- Variação Homóloga dos Resultados das Transferências entre Lojas;

A nova plataforma onde vão estar alocados estes indicadores tem de ter um processo dinâmico onde os gestores conseguem filtrar as vendas online ou vendas físicas e também filtrar qualquer loja que se pretenda analisar.

3.2.2. Análise dinâmica das Lojas

Outro dos objetivos definido nas reuniões com gestores foi a capacidade da nova plataforma conseguir apresentar um relatório dinâmico, em que o utilizador fosse capaz de selecionar uma loja e um intervalo de tempo, e esse relatório devolvesse uma tabela com informação de todos os artigos respetivos a essa loja, onde para cada artigo, no período de tempo selecionado, apresenta:

- O Nome do Artigo
- Stock Inicial (respetivo ao período de tempo selecionado);
- Quantidade de Entrada;
- Quantidade de Vendas;
- Quantidade de Vendas online;
- Quantidade de Devoluções;
- Quantidade de Devoluções online;
- Quantidade de Saídas;
- Quantidade de Stock Final e de Stock em Transferência;
- O *Sell-through*;
- Stock Final (respetivo ao período de tempo selecionado);
- O custo de Venda;
- O Valor em Devoluções;
- O Valor em Vendas;
- A Média de Descontos nas Vendas;
- A diferença entre o Valor das Vendas e o Custo de Vendas;
- A diferença entre o Valor das Vendas, o Custo das Vendas e o Custo das Devoluções;
- A Margem de Lucro.

Com esta informação é possível ter uma análise do valor que cada artigo representa para a loja selecionada, assim como o acesso à informação de todas as transações importantes de cada artigo naquela loja, no período de tempo que se pretende.

3.2.3. Análise dinâmica Artigo

Com o foco neste último objetivo da nova plataforma permitir selecionar uma loja e disponibilizar um relatório com a informação de todos os artigos da mesma, surgiu a vertente inversa, ou seja, de se possibilitar a seleção de um artigo e a nova plataforma facultar a informação do comportamento do artigo em todas as lojas. Este painel precisará de mais componentes gráficas onde se consiga entender facilmente, ao selecionar um artigo, os seus resultados de vendas em cada loja, o seu stock inicial e final, assim como a sua percentagem de *sell-through* para compreender rapidamente que lojas estão a carecer de stock do artigo e as que tem stock parado.

3.2.4. Análise do Cliente

Um dos pontos estabelecidos nas reuniões foi acerca dos cartões de cliente, a Foreva utiliza cartões cliente para manter os clientes associados através de descontos, mas apesar disso, os gestores querem analisar alguns indicadores de performance das lojas acerca dos processos de associação do cliente, dos quais:

- A variação Homóloga do número de clientes com registo de cartão cliente (clientes associados).
- A variação Homóloga do número de clientes associados que fizeram compras nas lojas.
- A taxa de *Churn* dos Clientes associados (% de Clientes associados que compraram no ano anterior, mas não no ano atual).
- A variação Homóloga da quantidade de vezes que se utiliza o cartão cliente.

Em relação às utilizações do cartão cliente, os gestores da Foreva querem perceber de que forma é que o número de utilizações do cartão está correlacionado com o lucro da empresa.

3.2.5. Suporte de Recomendações de Stock

O último objetivo, e muito pretendido pelos gestores, foi o desenvolvimento de uma área de recomendações na plataforma que fosse capaz de analisar o stock e as vendas de uma loja selecionada no sentido de remeter:

- Os artigos com pouca cobertura de stock (artigos com vendas na última semana e com menos 5 artigos na loja);
- Artigos que já estão na loja há mais de 2 semanas e não têm vendas;
- Recomendações de reposição de stock dos artigos com cobertura de stock (uma tabela que mostra os artigos sem vendas há 2 semanas nas outras lojas, mas com quebra de stock na loja selecionada).

O terceiro ponto tem implicações especiais porque uma loja com um artigo sem vendas só pode aparecer na recomendação de uma loja com o mesmo artigo em quebra de stock. Ou seja, se 5 lojas precisam de repor o mesmo artigo e apenas 2 lojas estão com stock e sem vendas desse mesmo artigo, então, só apenas 2 das 5 lojas é que podem ter recomendações desse artigo. Para isso ser possível ficou estabelecida uma ordenação em que a loja com mais vendas de um artigo em quebra de stock terá como recomendação o stock da loja sem vendas com mais quantidade desse mesmo artigo. Pode ver-se na figura 19, um exemplo de duas lojas que carecem de stock do artigo A e 5 lojas que estão sem vendas do artigo A há duas semanas. Como se pode constatar apenas vão ser geradas duas recomendações para restabelecer o stock das lojas que necessitam de reposição.

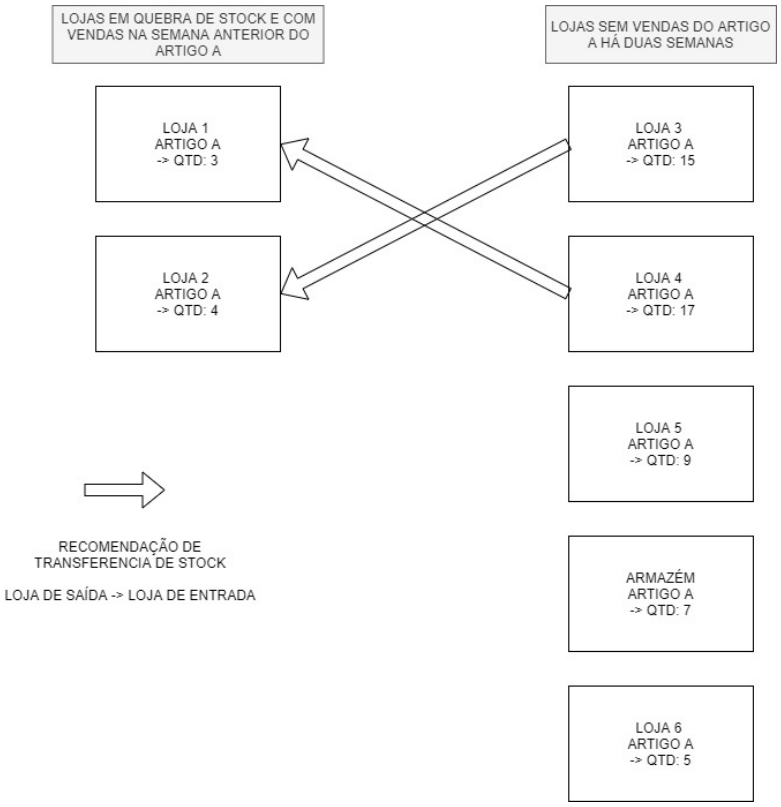


Figura 19 – Exemplo de como as recomendações de transferência de stock são geradas

O armazém da Foreva também é estabelecido para as transferências e será ordenado no algoritmo conforme uma loja normal. Neste caso, nunca tem vendas por isso qualquer artigo disponível no armazém poderá ser recomendado para restabelecer stock. Estas recomendações de transferência de stock são simples e úteis porque combinadas com o objetivo da análise do artigo descrito anteriormente, permitem aos gestores das lojas efetuar uma análise e tomar as melhores decisões no que diz respeito a reposição de stock.

A possibilidade de no futuro se automatizar as transferências de stock é bastante elevada, mas ficou claro nas reuniões que o algoritmo será desenvolvido aos poucos de forma a não gerar problemas futuros reconhecendo que os comportamentos de stock são um tema sensível.

Para além de recomendações para suportar a quebra de stock surgiram também algumas ideias para recomendar à loja que se seleciona artigos que podem ter sucesso, mas que a loja ainda não teve stock dos mesmos. Neste parecer foram abordados 3 métodos. No primeiro método são recomendados os top artigos com mais vendas que ainda não foram para a loja selecionada. O segundo método recomenda os top artigos com mais vendas que ainda não foram para a loja selecionada, mas que correspondem às categorias com melhor performance da loja selecionada. Estas categorias podem ser, por exemplo, calçado de homem, calçado de senhora, calçado de criança, entre outros. O último método seria uma abordagem com base na similaridade das vendas das lojas para prever artigos que poderiam ter sucesso na loja selecionada se fossem lá colocados. Este último objetivo descrito remete para uma abordagem de recomendações com algoritmos de filtragem colaborativa, mas com uma visão em que a loja é o utilizador. Isto é, ao invés de se desenvolverem recomendações com base nos gostos semelhantes dos utilizadores de uma plataforma web, utilizam-se os históricos das vendas dos artigos para gerar avaliações do comportamento dos artigos nas lojas. Desta forma é possível criar recomendações com base nas avaliações semelhantes que as lojas partilham dos artigos, adaptando estes algoritmos a recomendar artigos a lojas físicas. De forma a explicar melhor as afirmações anteriores, apresenta-se na Figura 20 um exemplo simples de uma forma de filtragem colaborativa em que a “LOJA 1” obteve boas avaliações nos artigos “A, B, C e D” e a “LOJA 3” obteve boas avaliações nos artigos “B,C e D” sendo que se encontra uma similaridade entre a “LOJA 1” e a “LOJA 3” podendo haver uma grande probabilidade de a “LOJA 3” obter um bom desempenho com o “ARTIGO A”, caso o artigo vá para essa loja.

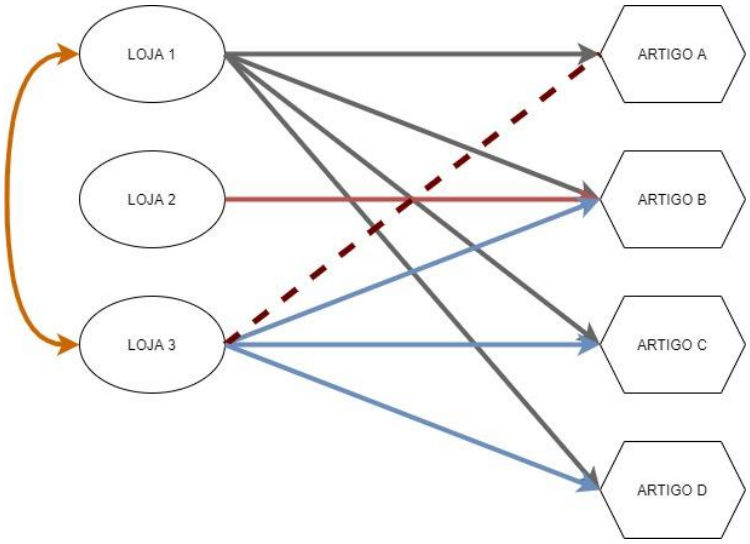


Figura 20 – Filtragem Colaborativa adaptada

Durante as reuniões com os gestores ficou decidido que o histórico de dados utilizado nos 3 métodos de recomendação falados anteriormente só poderia o da época atual, sendo que quando a época acabasse, o histórico da época terminada não poderia ser mais utilizado no algoritmo. Isto porque estas recomendações têm como principal objetivo, serem usadas durante o meio da época em questão para mostrar aos gestores que artigos é que se podem repor tentando aumentar as vendas.

3.3. Síntese

Neste capítulo ficaram definidos os principais objetivos da plataforma de *Business Intelligence* para com o cliente. Nos próximos capítulos são apresentadas as ferramentas utilizadas para o desenvolvimento da aplicação, as metodologias e arquiteturas abordadas para potencializar a plataforma assim como o seu desenvolvimento preparando o seu crescimento futuro, sabendo-se que este tipo de projetos tem um escalonamento enorme.

4. Desenvolvimento do Sistema de *Business Intelligence*

A Foreva pretende ter a informação de forma organizada, com uma visualização mais sucinta e concisa no que diz respeito às medidas de desempenho abordadas no capítulo 3. Para se satisfazer todos os objetivos foi necessário desenvolver um sistema de *Data Warehouse*, que permitiu, de forma gratificante, contemplar todos os critérios pretendidos para armazenar e suportar o processamento de informação que alimenta a plataforma de *Business Intelligence*. Desta forma é apresentado neste capítulo o "passo a passo" para o desenvolvimento desta aplicação de BI.

4.1. Fundamentação, Viabilidade e Planeamento do Projeto

A Foreva usufrui de uma ferramenta aplicacional que permite a transação de todos os dados da empresa, como por exemplo, os dados das vendas, das transferências de stock e dos clientes associados, permitindo que todo o sistema informático transacional das lojas Foreva funcione corretamente. Isto possibilita o registo das vendas, as transferências de artigos entre as lojas, o registo de clientes, a geração de listagens de stock de uma loja à escolha e a geração de listagens de vendas diárias. A base de dados deste sistema está alocada num dos servidores OLTP da empresa e, logicamente, está em constante uso pelos recursos humanos pois acarreta todo o funcionamento empresarial.

É percetível a necessidade do desenvolvimento de um sistema de *Data Warehousing* vocacionado para o apoio de decisões empresariais de forma a auxiliar os gestores com as informações necessárias e fidedignas. Desta forma, o sistema OLTP da Foreva não é sobrecarregado porque as consultas de informação serão feitas no servidor que alocará o *Data Warehouse*. Este terá uma estrutura preparada para o processamento de um grande volume de dados.

A viabilidade de implementação foi analisada assim como todos os custos e proveitos que o projeto iria trazer à empresa. Chegou-se à conclusão que seria viável o desenvolvimento de um Sistema de *Data Warehousing* para suportar a plataforma de *Front-End* que será utilizada pelos gestores para consultar informação necessária descrita no capítulo 3.

Um dos focos deste projeto é a consciência do crescimento exponencial que poderá existir no que diz respeito a objetivos e informação pretendida pelos gestores, todo o planeamento e desenvolvimento do projeto foi feito com atenção a esse critério.

A Foreva é uma empresa que tem lojas físicas espalhadas por todo Portugal e este sistema de *Data Warehouse* terá de ser capaz de garantir o acesso de forma segura e eficaz à informação considerada útil para a instituição. A informação das vendas e devoluções, de todas as transferências de stock, dos registos de stock diários, a informações dos artigos, dos clientes, das utilizações dos cartões de cliente e ainda mais informações que se entenda que sejam necessárias para garantir todos os objetivos definidos no capítulo 3. Todo o desenvolvimento das componentes algorítmicas de recomendação e os seus respetivos métodos de *Machine Learning* terão como origem de dados o *Data Warehousing*, evitando assim a sobrecarga dos sistemas OLTP.

De forma a garantir que este projeto tenha uma identidade própria, foi lhe dado o nome de "Foreva *Business Intelligence*".

Para execução deste projeto foram identificados todos os recursos e ferramentas necessárias, foi analisada toda a infraestrutura e equipamentos informáticos da empresa e foram estudadas todas as fontes de dados alocadas nos servidores OLTP. O novo armazém de dados ficou alocado num dos servidores analíticos da empresa, possibilitando o seu crescimento exponencial sem interferir noutros recursos informáticos.

As medidas de sucesso definidas para este projeto são a construção mais simples possível do DW e o baixo tempo de resposta do *front-end* da plataforma no acesso dos utilizadores.

A arquitetura do projeto foi definida de forma a ser possível realizar todos os objetivos pretendidos tornando o desenvolvimento escalável e flexível caso existam novas finalidades.

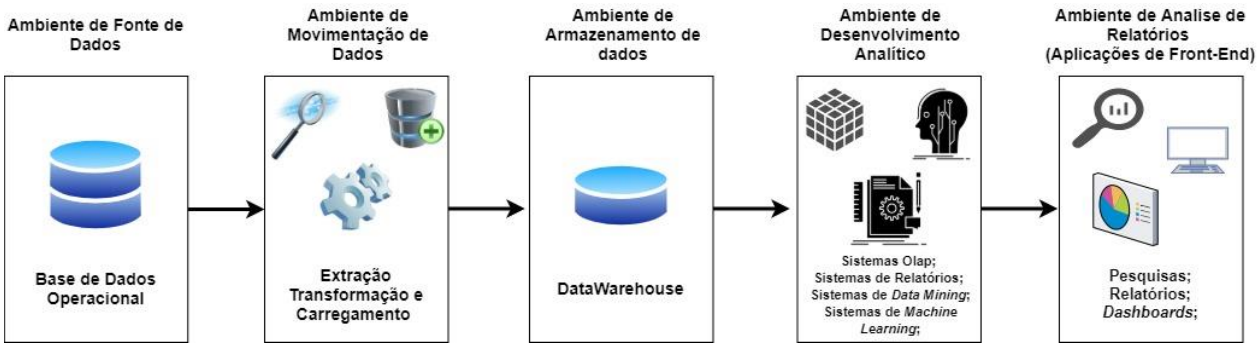


Figura 21 – Arquitetura do Sistema de BI

O plano de atividades do projeto tem como tópicos as atividades de preparação do projeto, a definição dos requisitos, a caraterização das fontes de dados, a modelação dimensional, o desenvolvimento do processo ETL, a implementação do *Data Warehouse,* a implementação do sistema de recomendação e o desenvolvimento analítico e criação da plataforma *front-end* de relatórios com a

ferramenta Power BI. No Anexo I é apresentada a descrição sucinta das ferramentas utilizadas para o desenvolvimento do projeto.

4.2. Análise de Requisitos

Durante o processo de análise de requisitos foi feita uma análise detalhada ao sistema que se pretendia desenvolver com objetivo de identificar os problemas que este terá de resolver. O sistema terá sucesso se conseguir solucionar todos os problemas pretendidos, pois um processo de levantamento de requisitos mal elaborado aumenta as probabilidades de fracasso.

Os objetivos principais do projeto são descritos no capítulo 3 e a metodologia utilizada para definir os requisitos do sistema de *Data Warehousing* foi baseada em entrevistas e reuniões com os gestores da Foreva, que resultaram na definição dos objetivos e dos critérios de sucesso assim como na definição dos requisitos de exploração, descrição e controlo de acesso.

Os requisitos de descrição que foram elaborados são:

- RD1: Todos os artigos devem estar registados com um id único que distinga somente aquele artigo do resto, mesmo sendo ele do mesmo modelo que outros. Isto porque a qualquer momento pode-se querer tentar perceber todas as transações ou movimentos de um simples artigo, isto poderá servir, no futuro, para detetar fraudes ou transferências mal executadas.
- RD2: Os artigos devem ter registada toda a informação disponível das bases de dados operacionais de forma a ser possível analisar a qualquer momento alguma caraterística dos artigos, assim como a pele, a cor, o género, o tipo e muitas mais.
- RD3: Cada Loja têm de ter registada um identificador único, a informação da sua situação, caraterística, zona, nome e se é franchisada ou não.
- RD4: Cada Cliente associado tem de ser registado com um identificador único e algumas informações pessoais aprovadas com a devida autorização do cliente segundo as novas leis de proteção de dados.
- RD5: Cada Documento de Transação tem de ser registado com identificador único e com as informações do ano, loja, tipo e o NIF do cliente que executou a compra ou devolução.
- RD6: Cada Venda efetuada tem de estar associada a apenas um artigo. Se o cliente fez mais que uma compra na mesma transação, estas terão o mesmo documento de transação. Para além disso, cada venda terá de ter registado o tamanho do artigo, a data, a hora e minuto que foi efetuada, a loja, o cliente caso este seja associado, a quantidade que será sempre igual a 1,

o preço de custo do artigo, o preço inicial do artigo, o preço de venda, o desconto, a informação se a venda foi online e o lucro da venda (preço de venda – preço de custo).

- RD7: As Devoluções tem de estar associadas a um artigo e tamanho correspondente, o cliente, o documento de transação, a data, hora e minuto, a loja, a quantidade devolvida que tem de ser exatamente igual a 1 (porque cada devolução só pode corresponder a um artigo), o preço de custo do artigo, o preço inicial, o preço da venda, se a devolução é online e o prejuízo da devolução (preço de custo – preço de venda).
- RD8: São registadas as transferências de artigos quando são executados pedidos para a transferência, depois quando a loja recetora recebe o artigo é registada a confirmação da transferência. É importante ter estas duas informações porque no futuro poderá tirar-se várias ilações a nível de qualidade de dados, realizando um estudo de transferências erradas, e também se poderá fazer um controlo do movimento do artigo.
- RD9: Em cada pedido de transferência e em cada confirmação de transferência tem de ser registado o artigo, o tamanho, a data, o minuto e hora, a loja de saída e a loja de entrada, assim como a quantidade transferida que tem de ser sempre igual a 1 (porque cada transferência corresponde a um artigo único).
- RD10: Em cada compra efetuada é registada uma transferência de pontos caso o cliente use o cartão. Para cada transferência de pontos é registado o cliente, a loja, a data, a hora e minuto e a quantidade de pontos transferidos. Caso obtenha um valor positivo são pontos acumulados no cartão, caso sejam negativos, são pontos que o cliente utilizou para obter desconto.
- RD11: O stock final diário terá de ser registado todos os dias para se ter um histórico do stock, onde cada registo terá uma data, loja, modelo de artigo, tamanho, quantidade em stock e quantidade em transação para essa loja.

Os requisitos de exploração são os requisitos que dizem respeito acerca das respostas de funcionamento do sistema. São os seguintes:

- RE1: Obter a quantidade vendida mensalmente em todas as lojas;
- RE2: Obter o volume de vendas mensais, assim como o lucro obtido;
- RE3: Obter o valor da margem de lucro e o preço médio de venda mensal;
- RE4: Obter o valor médio em cada compra, assim como a quantidade média mensal;
- RE5: Obter o *Cross-Selling* de cada loja a partir das quantidades das vendas que pertencem aos mesmo documentos;

- RE6: Obter o Registo médio de clientes;
- RE7: Obter a taxa de *churn* associada aos clientes;
- RE8: Obter o número de utilizações do cartão cliente;
- RE9: Obter o *Sell-Through* para cada artigo, em qualquer período escolhido;
- RE10: Obter os dias de Cobertura de Stock para cada artigo;

Os requisitos de controlo de acesso dizem respeito a restrição de utilizadores no sistema de *Data Warehouse*:

- RCA1: Apenas os engenheiros de informática e de sistemas monitorizam, gerem e definem as alterações e novos desenvolvimentos no sistema de *Data Warehouse*;
- RCA2: O sistema terá de ser adaptável e resiliente onde as tecnologias que o suportam não serão alteradas nem interrompidas, sendo este sistema projetado para uma evolução contínua;
- RCA3: O sistema de *Data Warehouse* terá sempre um controlo de segurança de nível elevado para os dados confidenciais;
- RCA4: As informações do *Data Warehouse* estarão seguras num sistema de *back-ups* monitorizados pelos engenheiros de informática;
- RCA5: O acesso dos utilizadores aos *Reports* da plataforma de *Business Intelligence*, que são dados provenientes do *Data Warehouse*, será disponibilizado pelos engenheiros informáticos e engenheiros de sistemas, mas definido pelos gestores da Foreva, podendo ser alterado a qualquer momento.

Por fim os Requisitos são analisados e validados através de reuniões com os gerentes e é autorizada a implementação do sistema.

4.3. Modelação Dimensional

O desenvolvimento do modelo dimensional é a componente principal de um projeto de *Data Warehouse*, é nesta fase que se interrogam os requisitos que vão trazer dependência a todas as outras fases do projeto.

4.3.1. Matriz de Decisão e Granularidade

De acordo com os objetivos principais e com os requisitos elaborados, o modelo dimensional representa uma constelação que neste caso é um conjunto de esquemas em floco de neve.

Inicialmente, foram construídos esquemas estrelas em separado e foi notório que algumas tabelas de facto partilhavam as mesmas dimensões. Foram desenvolvidas sub-dimensões nas dimensões modeladas passando os esquemas estrelas para esquemas em floco de neve, isto foi feito para retirar alguma redundância de informação e ganhar algum espaço de memória. Por fim, foram agrupados todos os modelos num só, gerando um modelo em constelação.

A decisão de granularidade de cada tabela de facto foi das partes mais importantes e críticas desta fase. O equilíbrio entre a sumarização e o detalhe tornam as granularidades modeladas com a possibilidade de obter a maior eficiência e eficácia nas consultas que serão feitas pelos utilizadores.

Nas tabelas dimensão encontram-se os registos descritivos referentes às tabelas de facto. As tabelas de facto são entidades que interligam as tabelas de dimensão associadas por chaves estrangeiras, e o facto representa uma transação de um evento associado ao tema da respetiva modelagem. A identificação das tabelas de facto está assegurada pelo seu respetivo grão. O grão é o menor grau de informação e é assente de acordo com as necessidades que são abordadas inicialmente, tanto nos objetivos como nos requisitos como no crescimento futuro do sistema. Neste caso, uma tabela de facto tem um nível de granularidade diferente das outras, fazendo com que uma das sub-dimensões de uma dimensão se torne uma dimensão para essa tabela de facto.

A formação deste *Data Warehousing* foi estabelecida a partir do desenvolvimento de 3 subconjuntos de informação classificadas em 3 diferentes assuntos, isto é, na construção de 3 *Data Marts*. Um dos *Data Marts* desenvolvidos foi o *Data Mart* Comercial, em que as tabelas de facto são correspondentes às vendas e devoluções, TF_VENDAS e TF_DEVOLUCOES respetivamente. Foi desenvolvida uma tabela de facto que guarda a informação das utilizações de cartão cliente (TF_TRANSFERENCIA_PONTOS), sendo esta tabela mais ligada ao Marketing dando assim origem ao *Data Mart* de Marketing. Outro *Data Mart* desenvolvido foi o Logístico, onde as tabelas de facto correspondem às transferências de stock e ao registo de stock diário (TF_TRANSFERENCIAS, TF_CONFIRMACAO_TRANSFERENCIAS, TF_REGISTO_STOCK). Foi mais valorizada a construção de dimensões e tabelas de facto que pudessem resolver os problemas e os objetivos da empresa do que propriamente a preocupação de quais *Data Marts* iriam ser trabalhados. Isto é, o *Data Mart* de Marketing só tem uma tabela de facto mas poderia ter mais pois existe muita informação para serem criadas mais

tabelas, mas como ainda não são necessárias mais informações relativas ao Marketing, não foram desenvolvidas. O *Data Warehousing* vai sendo desenvolvido consoante a necessidade de informações que os gestores precisam. Esse foi um dos princípios do projeto. Para além disso as tabelas de facto dos diferentes *Data Marts* partilham algumas das dimensões, sendo estas dimensões coerentes. Uma dimensão coerente é uma dimensão que é partilhada em diferentes *Data Marts*. O mesmo acontece para as tabelas de facto, uma tabela de facto coerente é também partilhada por diferentes *Data Marts*. Claramente, num contexto de marketing, se se quiser analisar os produtos mais vendidos para elaborar estratégias irá recorrer-se a informação da tabela de facto das vendas, que é usada, também, em contexto comercial, o que fará desta tabela uma tabela de facto coerente.

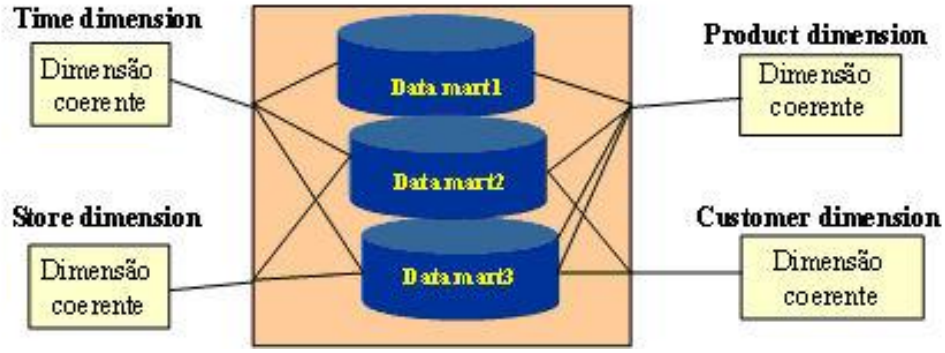


Figura 22 – Dimensões Coerentes (Adaptado de Ralph Kimball)

Foi desenvolvida uma Matriz de decisão que conjuga as dimensões com as tabelas de facto organizando a informação de forma matricial. A tabela 1 representa essa Matriz de decisão.

Tabela 1 – Matriz de Decisão

Caraterização dos Data Marts Comercial/Logístico/Marketing						
Identificação: Comercial/Logístico/Marketing						
Descrição Geral: Informação de carater logístico, comercial e de marketing com vista a apoiar a gestão do setor do retalho fornecendo dados que dizem respeito às vendas, devoluções, stock e utilização de cartões cliente.						
Estrutura Base:						
Dimensões:	TF_VENDAS	TF_DEVOLUÇÕES	TF_TRANSFERENCIAS	TF_CONFIRMAÇÃO_TRANSFERENCIA	TF_TRANSFERENCIA_PONTOS	TF_REGISTO_STOCK
DIM_ARTIGO	X	X	X	X		

DIM_TAMANHO	X	X	X	X		X
DIM_CLIENTE	X	X			X	
DIM_DOCUMENTO_NUMERO	X	X				
DIM_LOJA	X	X	X	X	X	X
S_DIM_ARTIGO						X
DIM_CALENDARIO	X	X	X	X	X	X
DIM_HORA	X	X	X	X	X	X
DIM_MINUTO	X	X	X	X	X	X
Número de Dimensões	8	8	6	6	5	6

Nas ilustrações em baixo podemos ver a descrição das tabelas de facto definidas em cima para se perceber a sua utilidade. As medidas escolhidas para as tabelas de facto vão possibilitar o cálculo dos KPI's mais importantes para satisfazer os objetivos descritos no capítulo 3.

Tabela 2 – Facto Vendas

Tabela de Facto	TF_VENDAS
Tipo	Transacional
Descrição	Transações de vendas respetivas às lojas físicas
Utilidade Estratégica	Identificar os melhores artigos em cada loja, as lojas que mais vendem e os clientes que mais compram. Elaborar vários indicadores de performance como a variação do volume de vendas, da quantidade de vendas, do preço médio de vendas, do valor médio por transação, da quantidade média por transação, do *cross selling* e da evolução dos descontos.

Tabela 3 – Facto Devoluções

Tabela de Facto	TF_DEVOLUCOES
Tipo	Transacional
Descrição	Transações de devoluções respetivas às lojas físicas
Utilidade Estratégica	Identificar os artigos mais devolvidos. Juntamente com a tabela de facto TF_VENDAS é possível avaliar o lucro total das vendas porque é possível obter as métricas das devoluções. Para além disso é possível construir os indicadores da variação das quantidades devolvidas, e a variação do valor perdido em devoluções assim como o peso das devoluções nas vendas.

Tabela 4 – Facto Pedidos de Transferências

Tabela de Facto	TF_TRASNFERENCIAS
Tipo	Transacional
Descrição	Pedidos de transferência realizados entre as respetivas lojas físicas.
Utilidade Estratégica	Para se conseguir identificar a quantidade pedida a ser transferida num determinado período de tempo. Juntamente com as informações da tabela de facto, TF_CONFIRMACAO_TRANSFERENCIA, perceber os pedidos de transferência que não foram satisfeitos ou foram feitos de forma errada.

Tabela 5 – Facto Transferências

Tabela de Facto	TF_CONFIRMACAO_TRASNFERENCIAS
Tipo	Transacional
Descrição	Transferências efetuadas entre lojas
Utilidade Estratégica	Registo das transferências de artigos entre as lojas. Esta tabela é muito importante porque permite armazenar a informação do período em que há transferências em cada loja, ou seja, quando o stock é reposto. As medidas desta tabela de facto vão ser utilizadas no cálculo das taxas de cobertura, *sell-through* e também no indicador de vendas depois de transferências para se conseguir analisar a percentagem de transferências que tiveram sucesso.

Tabela 6 – Facto Movimentos de Pontos

Tabela de Facto	TF_TRANSFERENCIA_PONTOS
Tipo	Transacional
Descrição	Movimentos de pontos de clientes
Utilidade Estratégica	Analisar as utilizações do cartão cliente, analisar as quantidades de pontos transferidos num determinado período de tempo assim como os clientes que utilizam mais vezes o cartão cliente nas suas compras. Perceber quais são as lojas onde existem mais transação de pontos e em que período de tempo é utilizado mais ou menos vezes o cartão cliente.

Tabela 7 – Facto Registo de Stock

Tabela de Facto	TF_REGISTO_STOCK
Tipo	Transacional
Descrição	Registo do stock inicial diário
Utilidade Estratégica	Tabela que vai armazenar o stock diário para se aprofundar as análises de vendas. Informação da quantidade de stock disponível de qualquer artigo, em qualquer dia, em qualquer loja. As medidas nesta tabela de facto são importantes para o cálculo de um dos indicadores mais afamado pelos gestores, o *sell-through.*

Como se pode observar, este *Data Warehouse* carateriza-se pela existência de 6 tabelas de facto, que estão associadas a 9 dimensões.

Depois de ser elaborada a matriz de decisão fica mais fácil de perceber o grão para cada tabela de facto. A caraterização dos grãos de cada tabela de facto foi feita da seguinte forma:

- TF_VENDAS: Venda de um artigo único, com um respetivo tamanho, a um respetivo cliente, numa respetiva loja, correspondente a um documento de transação, num certo dia a uma certa hora e minuto.
- TF_DEVOLUCOES: Devolução de um artigo único, com um respetivo tamanho, a um respetivo cliente, numa respetiva loja, correspondente a um documento de transação, num certo dia a uma certa hora e minuto.
- TF_TRANSFERENCIA: Pedido de transferência de um artigo único, com um respetivo tamanho, de uma respetiva loja para outra, num determinado dia, hora e minuto.
- TF_CONFIRMACAO_TRANSFERENCIA: Confirmação de transferência de um artigo único, com um respetivo tamanho, de uma respetiva loja para outra, num determinado dia, hora e minuto.
- TF_TRANSFERENCIA_PONTOS: Ganho ou Perda de pontos de um respetivo cliente, numa determinada loja, num determinado dia, hora e minuto.
- TF_REGISTO_STOCK: Registo de stock de um determinado modelo de artigo, com um respetivo tamanho, numa determinada loja, a um determinado dia, hora e minuto.

O relacionamento existente entre o detalhe e a granularidade é inverso, isto é, quanto menor o grão maior o nível de detalhe, ou seja, há mais flexibilidade de se obter respostas nos dados, mas em contrapartida pior será a velocidade das consultas e o volume de dados armazenado.

Dado o desenvolvimento da matriz de decisão e do grão de cada tabela de facto foram elaboradas as descrições e caraterizações das dimensões e factos para serem explicados os seus atributos e meta-dados.

4.3.2. Caraterização das Dimensões e das Tabelas de Facto

As dimensões consistem num conjunto de atributos que devem estar associados às tabelas de facto. Devem ser caraterizadas e a sua informação organizada de forma a serem definidos os atributos, as hierarquias e os perfis de utilização.

Na elaboração das caraterizações das dimensões deve-se fazer um preenchimento da sua descrição geral, sintetizando a área de negócio que cada dimensão representa. Em relação aos atributos, estes vão corresponder às descrições dos campos da dimensão que serão utilizados pelas tabelas de facto. As hierarquias indicam as agregações ou desagregações possíveis, algumas hierarquias foram convertidas em subdimensões para diminuir o armazenamento de dados e foram criadas as respetivas tabelas de caraterização também para as subdimensões.

Nos anexos II, III, IV, V, VI pode observar-se as tabelas de caraterização que representam as dimensões e subdimensões do *Data Warehouse*.

Começando pela dimensão DIM_ARTIGO, esta tabela contém as informações de cada artigo único, assim como o modelo de cada um. É importante a granularidade estar nivelada com os artigos únicos para se conseguir guardar informação do percurso de cada artigo único, só assim será possível entender os erros das transferências de artigos e no futuro existir a possibilidade de se criar algum sistema associado com deteção de fraudes ou desaparecimento de artigos, que é uma das abordagens que os gestores poderão querer recorrer no futuro. É claro que o motivo desta granularidade não foi apenas este, a nível de qualidade de dados é muito mais fácil entender qualquer erro ou conhecimento útil nos dados dos artigos se se armazenar as informações detalhadas, isto é, imagine-se que se quer explorar um percurso de um artigo único para se perceber se as informações estão corretas, se ele foi vendido ou não, em que lojas passou, entre outros, é muito mais simples analisar essas informações com acesso ao maior detalhe de informação que está disponível relativamente aos artigos. Essa segurança no detalhe da informação vale o armazenamento despendido na dimensão artigo.

Como existem muitos artigos únicos com o mesmo modelo optou-se por construir uma subdimensão da DIM_ARTIGO para guardar as informações de cada modelo, a tabela S_DIM_ARTIGO. Desta forma não haverá duplicação das informações de cada modelo na dimensão. Esta subdimensão é dimensão para a tabela de facto REGISTO_STOCK, porque simplesmente não faria sentido registar o stock de cada artigo único, mas sim dos seus modelos. Isto também serviria para reduzir o armazenamento que será utilizado sabendo que guardar o stock de cada artigo único todos os dias iria ser bastante dispendioso em questões de armazenamento de dados, e também porque não se iria ganhar muito mais informação tendo uma granularidade mais baixa uma vez que as outras tabelas de facto já contém uma granularidade menor, e com isso a informação de todas as transações. A tabela de caraterização da dimensão DIM_ARTIGO encontra-se no anexo II.

A subdimensão dos artigos guarda toda a informação relativa aos modelos dos artigos. No anexo II é possível visualizar-se a tabela de caraterização da S_DIM_ARTIGO.

Foram criadas as subdimensões que armazenam as informações da pele, cor, marca, época e género do artigo. As suas tabelas de caraterização encontram-se também no anexo II.

Terminada a caraterização da dimensão Artigo, passou-se à caraterização da dimensão loja. A dimensão loja contém algumas subdimensões que guardam a informação das zonas das lojas, das situações das lojas e das suas caraterísticas.

O desenvolvimento desta dimensão foi realizado para armazenar a informação das lojas Foreva, e a dimensão é do tipo 1, ou seja, na atualização, os valores antigos são substituídos pelos valores mais recentes, onde foram acrescentados os atributos Inicio e Fim caso seja necessário colocar a informação do período de tempo em que as lojas existiram. Ainda não foi necessário ter essa informação, mas, durante o desenvolvimento, foram preparados esses atributos para que no futuro, se houver necessidade, ser dispensável mudar a estrutura da dimensão. Caso seja necessário existir essa informação no *Data Warehouse* terá de ser colocada através de *updates* manuais pois não existe na origem de dados quaisquer informações sobre o período de tempo em que as lojas estiveram abertas.

No anexo III são apresentadas todas as caraterísticas da dimensão loja.

As subdimensões da zona, situação e caraterística das lojas têm o mesmo objetivo das subdimensões criadas para a dimensão artigo, isto é, para reduzir a redundância e armazenamento dos dados. As tabelas de caraterização destas subdimensões relacionadas com a dimensão da loja encontram-se, também, no anexo III.

Relativamente à DIM_TAMANHO, esta é a dimensão que guarda apenas as informações dos tamanhos disponíveis. Esta dimensão poderia ter sido considerada como uma dimensão degenerada, ou

seja, apenas aparecer como atributo nas tabelas de facto, mas foi decidido que, devido à importância da análise dos tamanhos principalmente na componente logística, que esta dimensão merecia não ser degenerada, até porque o armazenamento de dados que esta tabela representa é quase nulo. A tabela de caraterização da DIM_TAMANHO encontra-se no anexo VI.

A Dimensão Cliente foi criada com o nome DIM_CLIENTE e armazena as informações dos clientes associados (com cartão cliente). A dimensão também é do tipo 1 pois não existe nenhuma vantagem em guardar o histórico dos dados dos clientes, apenas basta manter os dados atualizados. Uma das informações mais importantes relativa aos clientes associados são os pontos que cada um tem no cartão cliente, ou seja, com esta dimensão é possível ter a informação de quantos pontos existem para ser utilizados pelos clientes, e o valor que os pontos representam. Esta informação é vista como muito útil nas estratégias de marketing, segundo os gestores da Foreva. A dimensão é caraterizada no anexo V.

A dimensão DIM_DOCUMENTO_NUMERO armazena a informação dos documentos de transação, isto é, cada transação efetuada por um cliente, tanto venda como devolução, são registadas num documento de transação. Por exemplo, se um cliente for a uma loja e comprar 2 artigos em simultâneo, essa compra terá um documento de identificação. Assim, na tabela de facto vendas, o registo da venda de cada um destes dois artigos terá associada o mesmo documento, o que permite identificar que as vendas foram feitas em simultâneo, ou seja, representam a mesma compra. O mesmo acontece na tabela de facto devoluções. Esta dimensão é muito importante porque é a partir das informações que ela faculta que é possível gerar KPI's como o *cross selling*, a média de quantidade comprada em cada transação e a média de valor gasto em cada compra. A tabela de caraterização desta dimensão encontra-se no anexo VI.

Finalmente, as dimensões temporais, a dimensão do calendário chamada de DIM_CALENDARIO, a dimensão hora chamada de DIM_HORA e a dimensão minuto chamada de DIM_MINUTO. Estas dimensões são criadas de forma separada para não existir redundância de dados, por exemplo, se existisse um atributo hora na dimensão calendário, cada data era repetida 24 vezes, o mesmo com a dimensão minuto, não existe atributo minuto na dimensão hora para evitar que cada hora seja repetida 60 vezes na base de dados, se isso acontece-se seria ocupado espaço de forma desnecessária. As dimensões temporais são caraterizadas no anexo VII.

Em relação às tabelas de facto, estas são caraterizadas no anexo VIII. No subcapítulo anterior consegue-se perceber a que dimensões vai estar ligada cada tabela de facto.

Em relação às medidas de cada tabela de facto, estas foram escolhidas com o objetivo de trazer valor simbólico para a análise de dados.

4.3.3. Esquema Dimensional

Antes do desenvolvimento do esquema dimensional final, foram desenvolvidos vários esquemas dimensionais para obter uma melhor compreensão sobre o desenvolvimento de cada tabela de facto. Depois foi desenvolvido o esquema principal, no formato de constelação.

Em baixo nas figuras 23 e 24, pode verificar-se os modelos de dados associados às tabelas de facto venda e devolução. Depois de serem caraterizadas as dimensões nos subcapítulos anteriores, assim como o grão das tabelas de facto, agora fica mais fácil de entender os modelos de dados. Em relação às vendas e devoluções observa-se que a dimensão artigo é constituída por uma subdimensão (S_DIM_ARTIGO), armazenando as informações de todos os modelos de artigo, e contando também com 5 subdimensões nesta subdimensão, provenientes de 5 hierarquias. A dimensão loja também tem 3 subdimensões.

Tanto a facto de vendas como a facto devoluções são constituídas exatamente pelas mesmas dimensões, os mesmos relacionamentos e hierarquias.

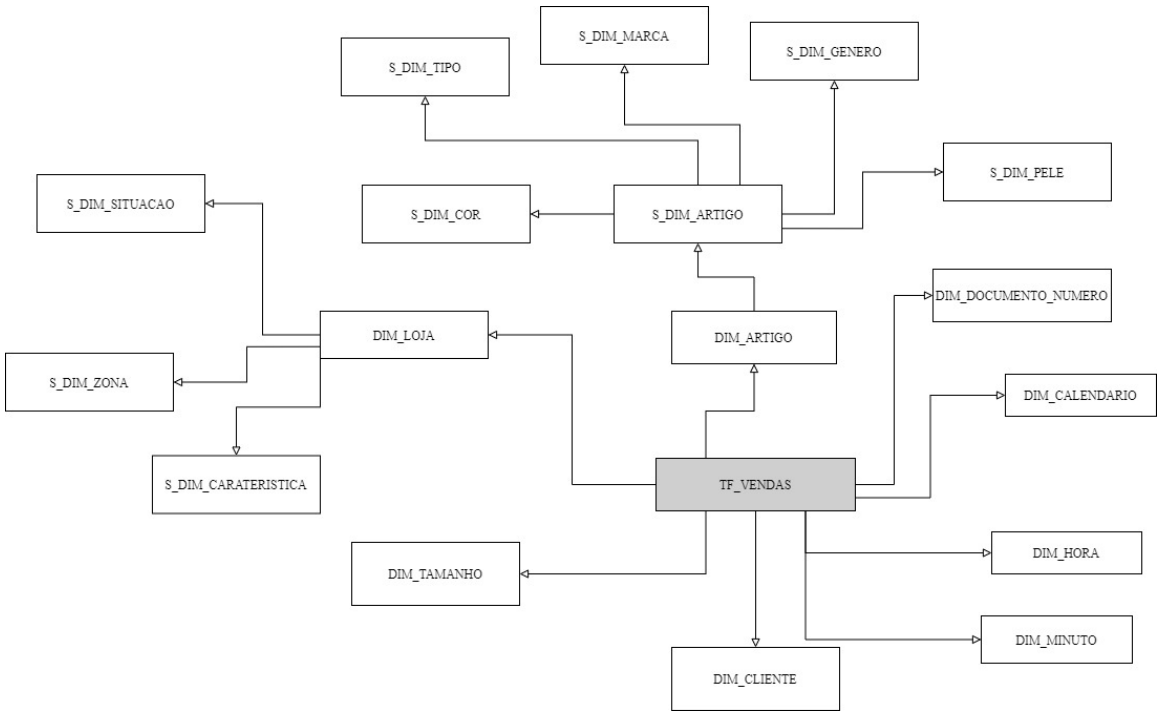


Figura 23 – Modelo de dados em Floco de Neve da TF_VENDAS

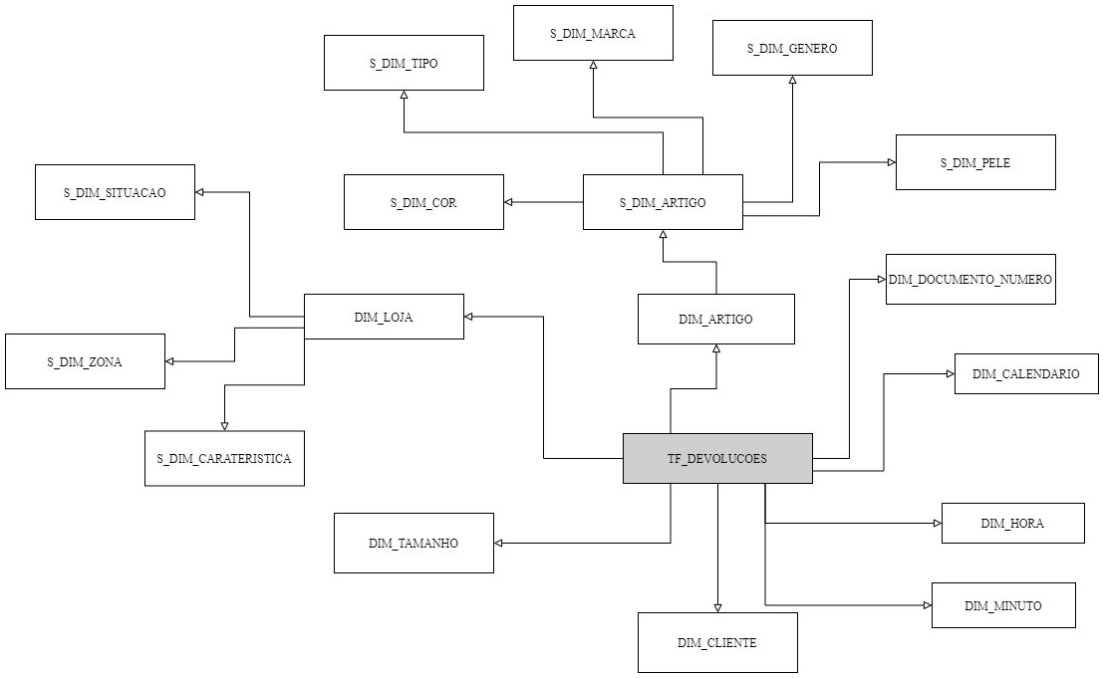


Figura 24 – Modelo de dados em Floco de Neve da TF_DEVOLUCOES

As tabelas de facto de transferências e de confirmação de transferências também constituem as mesmas dimensões. Em baixo nas figuras 25 e 26 pode verificar-se o modelo de dados de ambas as tabelas, respetivamente. Note-se que existem dois relacionamentos com a dimensão da loja, pois nas transferências os artigos têm de sair de uma loja e entrar noutra, assim como nas confirmações de transferências.

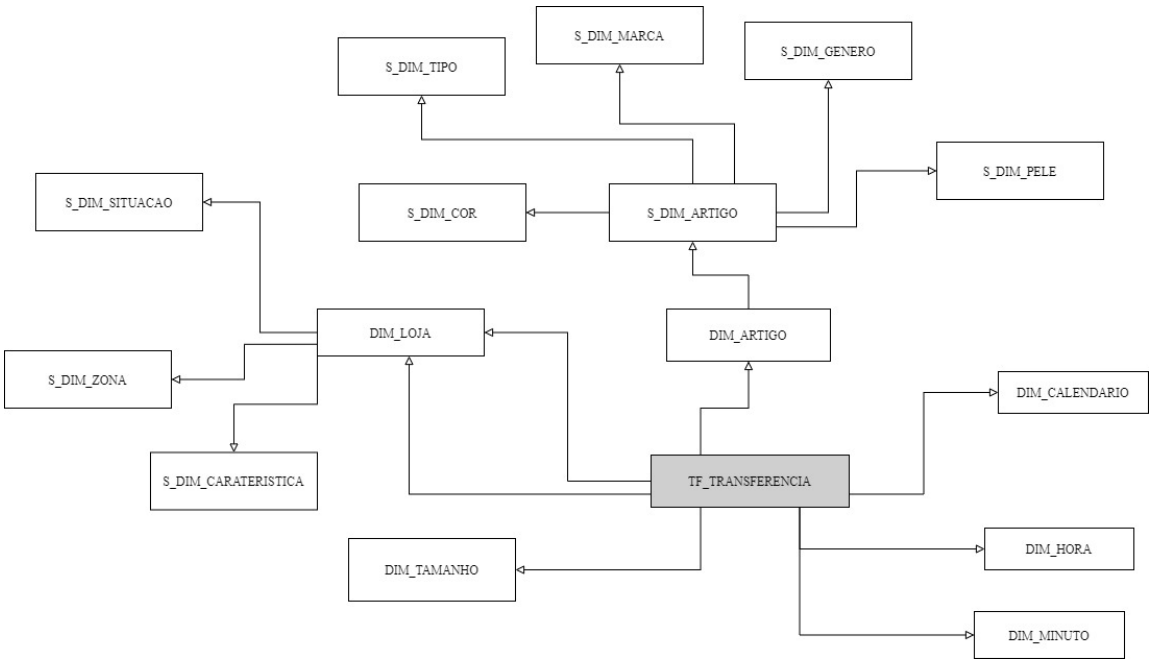


Figura 25 – Modelo de dados em Floco de Neve da TF_TRANSFERENCIAS

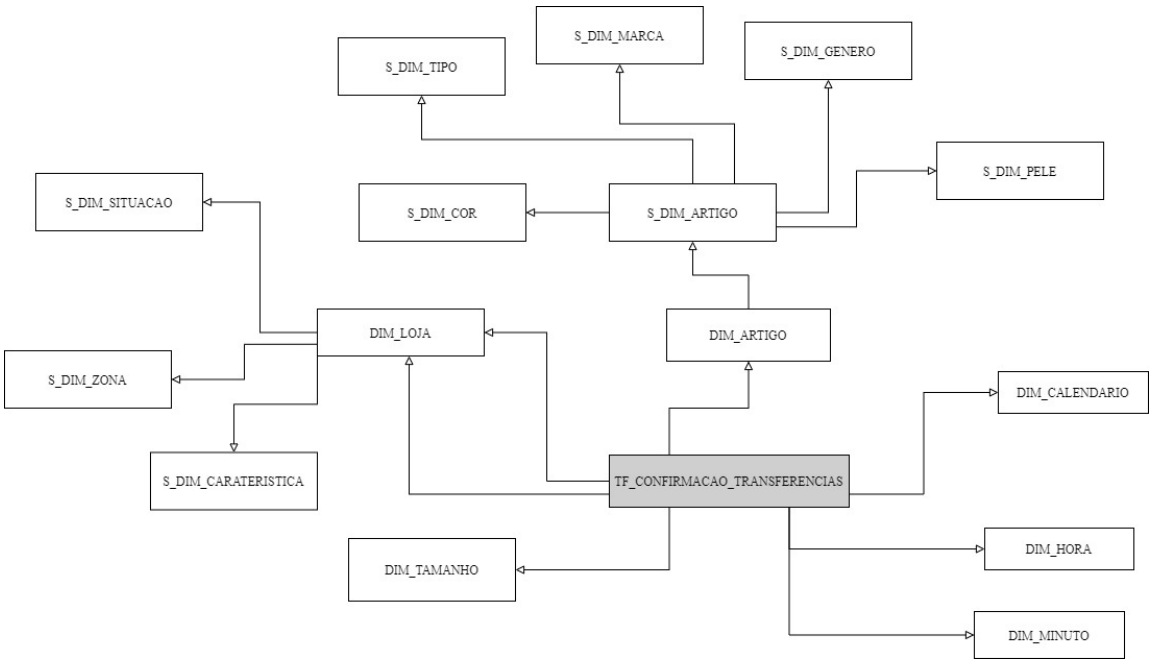


Figura 26 – Modelo de dados em Floco de Neve TF_CONFIRMACAO_TRANSFERENCIAS

A tabela de facto de transferência de pontos vai armazenar todas as transferências de pontos que os clientes executam nas lojas, tanto a perda de pontos nos descontos, como na acumulação de pontos em vendas no cartão. Esta tabela armazena as transações dos clientes registando o valor total dos pontos transferidos em cada transação. Em baixo na figura 27, observa-se o modelo de dados em floco de neve dessa tabela de facto.

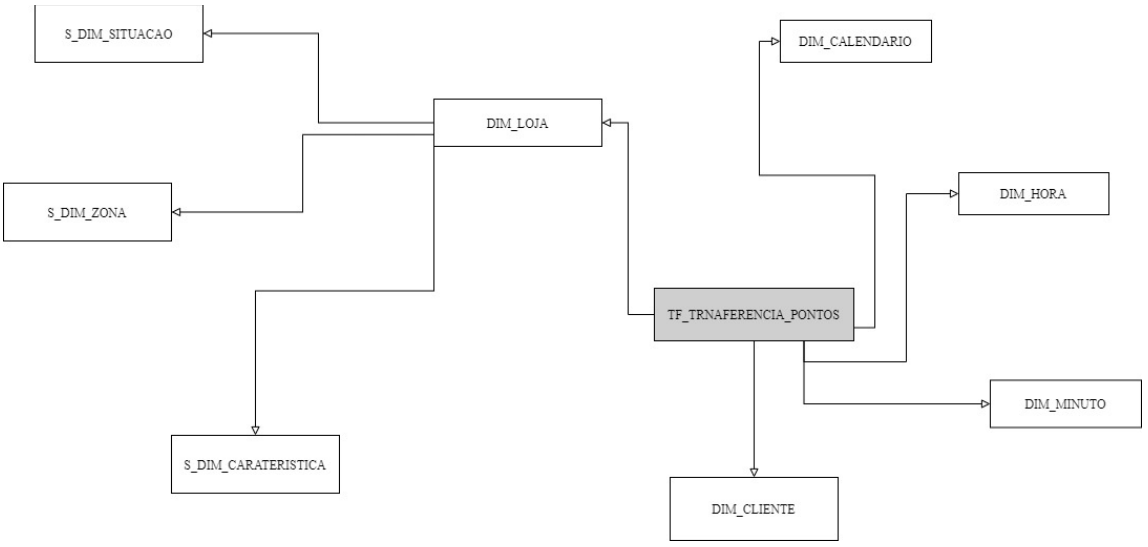


Figura 27 – Modelo de dados em Floco de Neve TF_TRANSFERENCIA_PONTOS

Finalmente, abordando a última tabela de facto, a tabela de registo de stock. Observa-se que para esta tabela de facto, a dimensão dos artigos é a subdimensão da dimensão artigo para as outras tabelas. Isto já foi explicado anteriormente, no registo de stock interessa apenas armazenar a informação da quantidade de cada modelo de artigo que está presente em cada loja. Caso o relacionamento desta tabela de facto seja igual à dimensão artigo das outras tabelas de facto, os gastos em armazenamento de informação iriam ser muito superiores e esse nível de detalhe nunca iria ser muito útil para quaisquer análises. Daí, em vez de se criar uma nova dimensão de artigo para esta tabela de facto registo de stock, optou-se por utilizar como dimensão, uma subdimensão já existente no modelo de outras tabelas de facto, sabendo que a informação necessária dos artigos para este caso, vai estar toda armazenada nessa tabela, escapando-se desta forma a criação de uma nova tabela que traria redundância de informação.

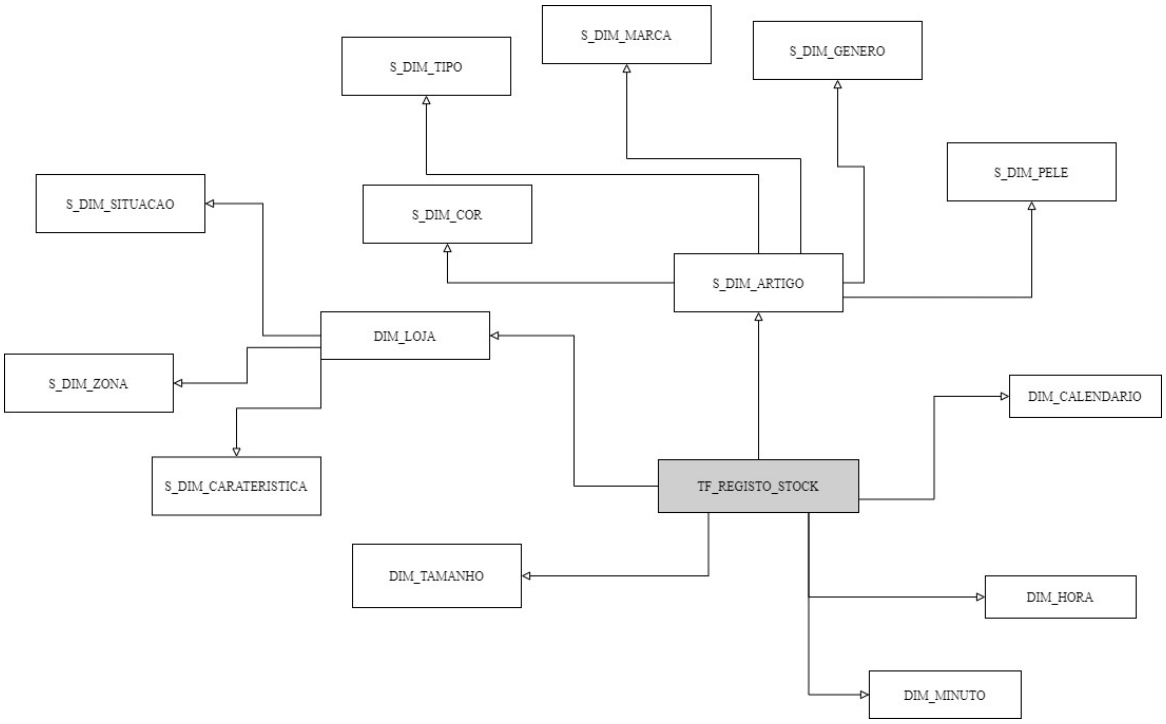


Figura 28 – Modelo de dados em Floco de Neve TF_REGISTO_STOCK

Depois de serem desenvolvidos todos os modelos de dados para cada tabela de facto e caraterizadas todas as tabelas com a devida informação dos meta dados, foi desenvolvido o modelo de dados final na estrutura de constelação, onde as tabelas de facto partilham as tabelas dimensão necessárias observadas nos modelos mostrados anteriormente. O modelo de dados final desenvolvido encontra-se no anexo IX.

A etapa subsequente do desenvolvimento do esquema dimensional consistiu na revisão do modelo com o objetivo de o analisar e verificar a sua coerência. Nesta análise, é feita uma avaliação de modo a verificar se o modelo vai conseguir corresponder ao pretendido, e se for necessário retifica-se o esquema de modo a desenvolver as alterações necessárias. Algumas das alterações que podem ser feitas nesta fase podem ser, por exemplo, a inserção de atributos caraterizando mais detalhadamente as tabelas. Depois da revisão feita e do modelo aprovado seguiu-se para o processo de caraterização das fontes de dados.

4.4. Caraterização das Fontes de Informação

Antes da elaboração do Sistema de *Data Warehousing* foi imprescindível identificar e caraterizar as fontes de informação candidatas. O sistema de base de dados da Foreva (*Microsoft SQL SERVER*) retém quase toda a informação importante e necessária para o desenvolvimento do sistema de *Data Warehousing*.

Durante o processo de análise das fontes de informação foi verificado que as informações das lojas não eram suficientes para o que era esperado no sistema. A resolução desse problema foi abordada com o desenvolvimento de uma nova fonte de informação num ficheiro Excel, que seria manipulada pelos gestores. O ficheiro contém toda a informação importante em relação às lojas e pode ser atualizada quando for necessário pelos gestores.

A componente mais crítica das fontes de informação é a tabela que contém a informação dos stocks, pois só é possível obter a informação do stock atual, isto é, o processo ETL do sistema de *Data Warehousing* terá de armazenar o histórico de stock diário na tabela de facto "TF_REGISTO_STOCK". Esse processo nunca poderá estar suscetível a falhas devido à informação na fonte de dados se modificar muito rapidamente e não ser possível recuperar o histórico.

Depois da caraterização das fontes de dados, deu-se início ao desenvolvimento do sistema de *Data Warehousing*.

4.5. Implementação do Sistema de *Data Warehousing*

4.5.1. Desenvolvimento do Sistema Físico de Dados

Nesta fase é convertida a conceção dos modelos numa etapa mais prática, sendo que se passa do esboço para a implementação física do sistema de dados. Com base no esboço do modelo dimensional apresentado no anexo IX foi desenvolvido o código SQL, através da ferramenta *Microsoft SQL Server Managment Studio*, capaz de implementar fisicamente o modelo de dados no servidor OLAP. O código é apresentado no Anexo X.

4.5.2. Desenvolvimento do Processo ETL

Este subcapítulo diz respeito ao desenvolvimento do sistema ETL respetivo à transição de dados entre as fontes de dados e o *Data Warehouse*.

A ferramenta utilizada para o desenvolvimento dos mecanismos de ETL foi o *Microsoft SQL Server Integration Services*. Inicialmente foram implementados fisicamente a *stage area* e a *quarentena*.

Foi elaborado o planeamento do processo ETL e posteriormente implementado, onde foi usado como exemplo de demonstração a dimensão artigo e a tabela de facto vendas.

A figura 29 representa o plano de execução de todos os *packages* de ETL que foram desenvolvidos.

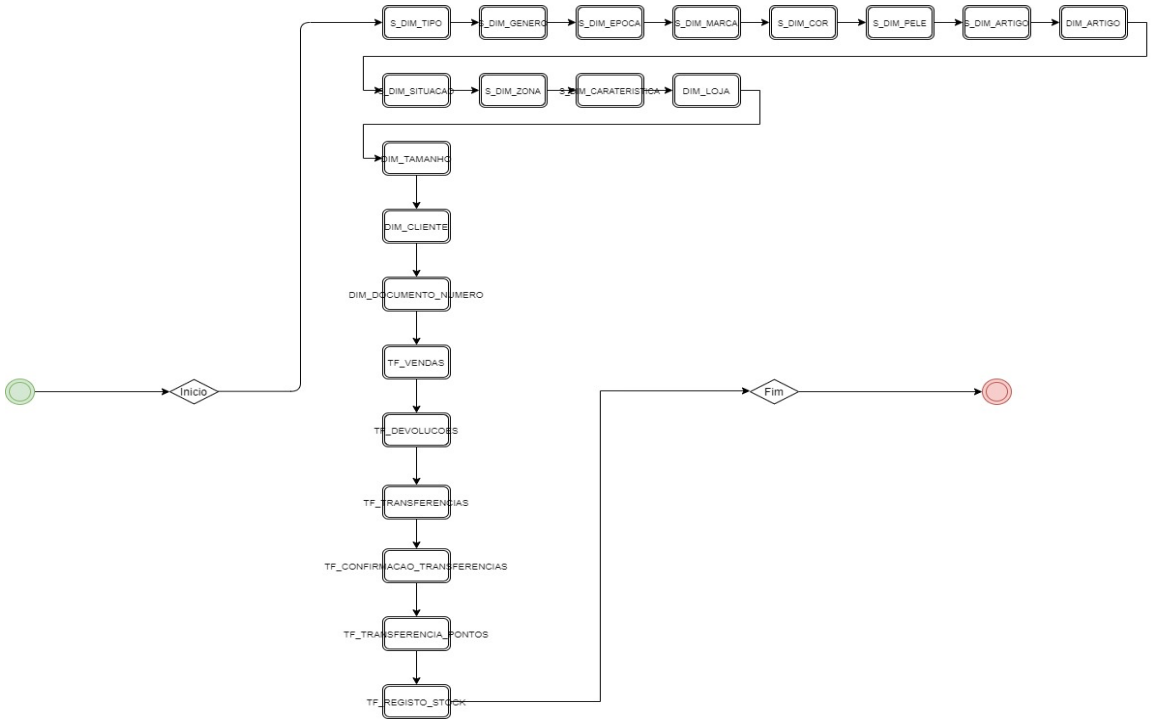


Figura 29 – Planeamento da execução dos *packages* do processo ETL

Cada *package* elaborado diz respeito ao processo ETL de uma tabela. Desta forma, o processo ETL conta com 21 *packages* que são o número total de tabelas existentes no *Data Warehouse*.

Na figura 30 é demostrado como exemplo o planeamento da execução do *package* da DIM_ARTIGO. Este *package* é o oitavo a ser processado durante o ETL.

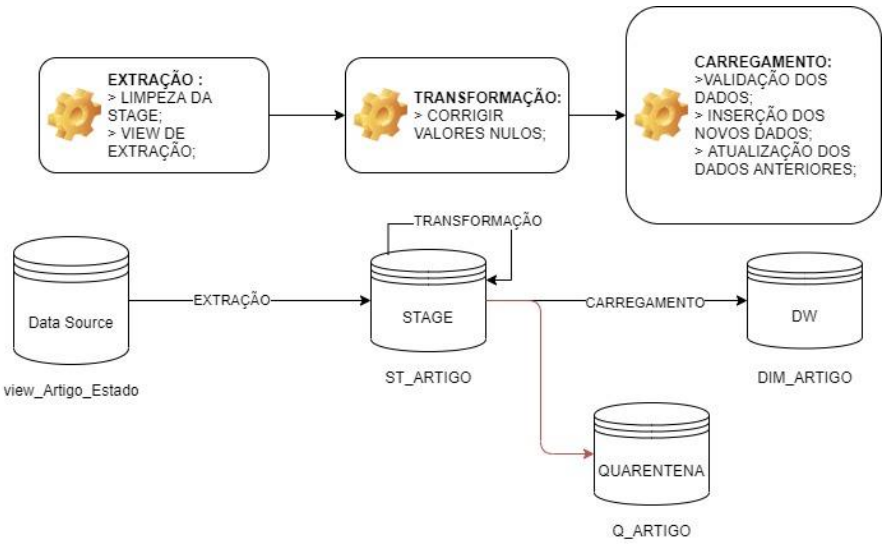


Figura 30 – Planeamento do *package* da "DIM_ARTIGO"

Como se pode apurar, este *package* foi planeado para executar tarefas individuais na fase de extração, transformação e carregamento, sendo que o desenvolvimento do seu mecanismo foi baseado neste planeamento.

A fase de extração executa uma *view* desenvolvida que organiza os dados de forma a serem carregados na *stage*. A maioria das transformações de dados é executada durante o processo de extração através da *view* de extração, deixando apenas a correção dos valores nulos para a *stage area* com a fase de transformação. As *views* ajudam também na filtragem de dados diminuindo o tempo de processamento. A *stage area* é importante para executar algumas transformações, neste caso a correção dos valores nulos, mas também é relevante porque já contém os dados na estrutura do *Data Warehouse*, e isso é fundamental para a fase de carregamento. Na fase de carregamento são validados as chaves para serem inseridos os novos dados no *Data Warehouse* ou atualizados os dados antigos. Se existir alguma anomalia, como por exemplo, serem encontrados duas chaves primárias iguais na *stage*, então esses dados anormais são carregados na tabela de quarenta aguardando correção. Na figura 31 é apresentado o *control flow*, desenvolvido no *Integration Services* para o *package* da DIM_ARTIGO.

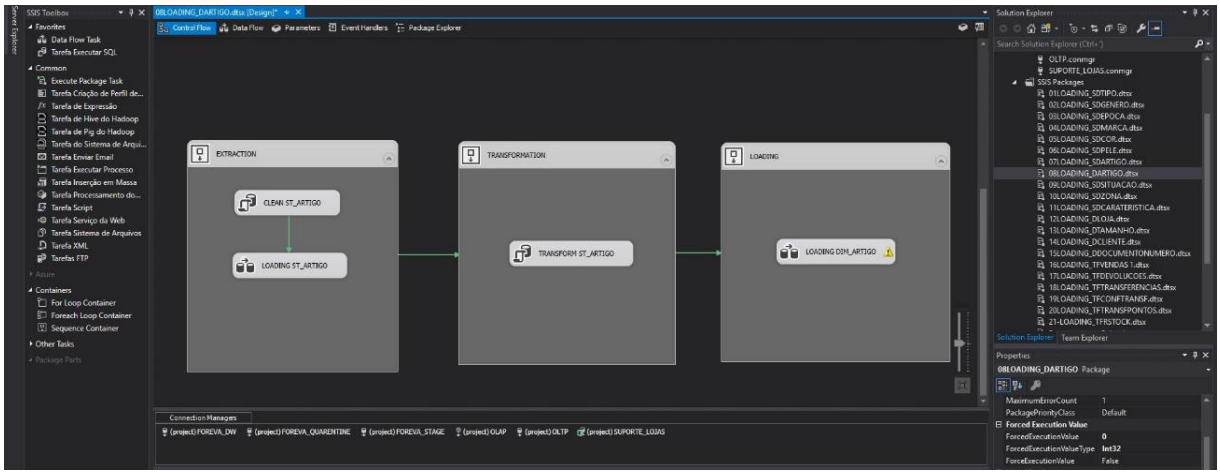


Figura 31 – *Control Flow* do processo ETL do DIM_ARTIGO

A figura 32 é usada para demonstrar como foi desenvolvido, no *Integration Services*, a fase de carregamento desta componente através da utilização do método *Slowly Change Dimension*, permitindo desta forma a verificação dos dados, a inserção dos novos, a atualização dos antigos registos e o desvio para a quarentena das anomalias detetadas nos dados.

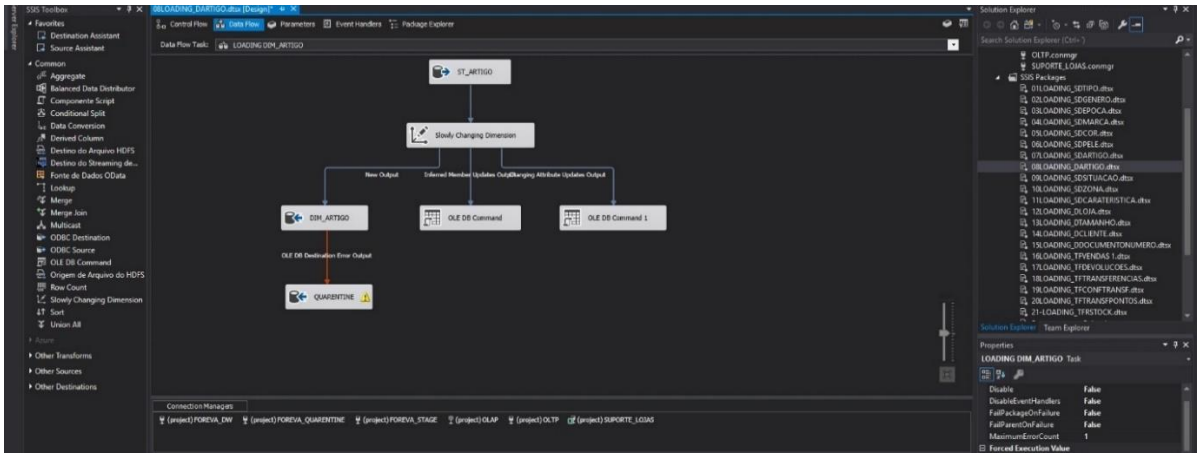


Figura 32 – *Data Flow* da fase de carregamento do processo ETL do DIM_ARTIGO

Este processo é bastante idêntico para todas as dimensões, mas no que diz respeito ao processo ETL das tabelas de facto, a fase de carregamento foi desenvolvida de uma forma um pouco diferente.

As fases de extração e transformação das tabelas de facto, são idênticas às das dimensões, já na fase de carregamento, em vez do método *slowly change dimension* são executados *scripts* que chamam os *store procedures* desenvolvidos capazes de validar os registos de uma forma temporal, só inserindo dados a partir do registo mais recente na tabela de facto. O exemplo utilizado é a *store procedure* da fase de carregamento do *package* da tabela de facto FT_VENDAS (Anexo XI).

4.5.3. Validação e Teste do Sistema ETL

Depois do Sistema ETL desenvolvido deu-se início à fase de povoamento de dados inserindo os dados desde o ano de 2018. O *Data Warehouse* ficaria assim com um histórico desde de 2018 nas tabelas de facto, excecionalmente, na tabela de facto FT_REGISTO_STOCK. Como foi dito anteriormente as fontes de dados só contém as informações do stock atual e só com o sistema de ETL em funcionamento é que é possível guardar os históricos de stock no *Data Warehouse*.

As primeiras tabelas onde foi feita a inserção dos dados foi a DIM_CALENDARIO, a DIM_HORA e a DIM_MINUTO, sendo as únicas tabelas que não têm *packages* de refrescamento pois os seus dados não precisam de ser atualizados sendo necessário fazer apenas a inserção dos dados. Para isso foram desenvolvidos 3 scripts SQL que realizam a inserção dos dados nessas tabelas. É apresentado no anexo XII o exemplo do script de inserção de dados da DIM_CALENDARIO.

Foram mudados os parâmetros do sistema ETL para se inserirem os dados desde 2018, o processo da primeira inserção correu como o esperado e foram verificadas todas as tabelas para perceber se os dados estariam corretos.

Na figura 33, apresenta-se a tabela de facto FT_VENDAS com os dados inseridos.

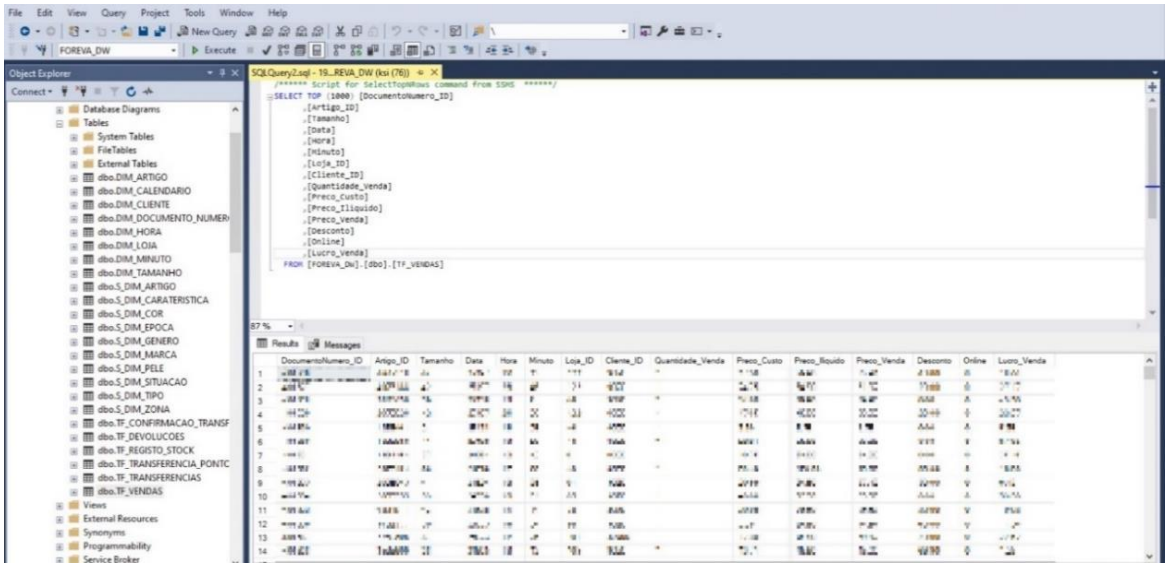


Figura 33 – Dados da tabela FT_VENDAS

Depois da primeira inserção de dados em todas as tabelas, o processo ETL foi novamente verificado e testado. Depois de validado, foi configurada uma rotina do ETL através da criação de um *job* no *SQL Server Agent*, fazendo com que os *packages* executem automaticamente todos os dias às 00h15, na forma como foi estipulado na figura 29 do subcapítulo 4.5.2. Pode ver-se na figura 34 de que forma é que foram organizadas as *tasks* dos *packages* no *job*. Depois do *job* criado foi dado como terminado o desenvolvimento do processo ETL.

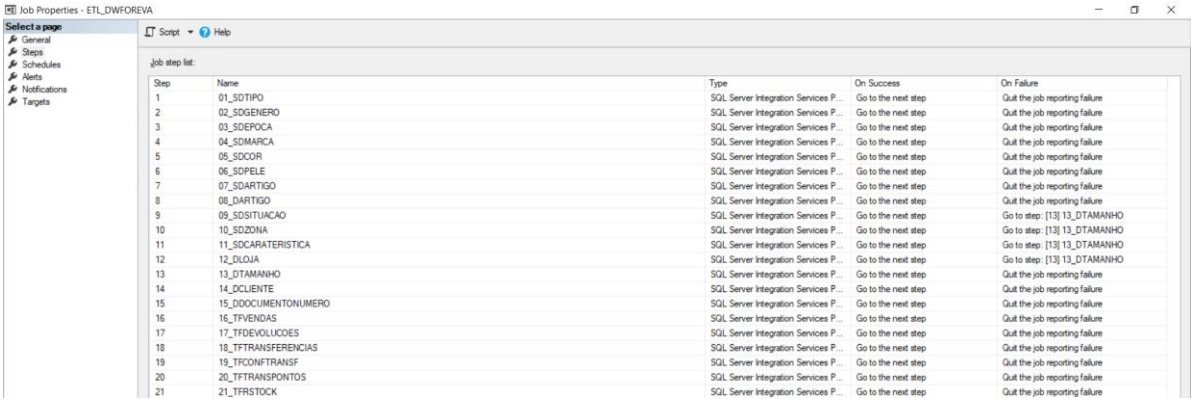


Figura 34 – *Job* do processo ETL

Desta forma é garantida a execução diária ou rotina do processo ETL sendo que foi estabelecido a criação de uma aplicação que ajuda na manutenção do processo ETL e da qualidade de dados do *Data Warehouse*.

4.5.4. Aplicação de Suporte a Manutenção do *Data Warehouse*

O desenvolvimento da aplicação de suporte de manutenção do sistema de *Data Warehousing* foi projetado sobre os requisitos de possuir um funcionamento via *mobile* ou computador para que o utilizador, que neste caso será o responsável pela manutenção do *Data Warehouse*, conseguisse aceder facilmente em qualquer local e a qualquer hora às informações do sistema.

A aplicação é capaz de transmitir ao gestor da base de dados as seguintes informações:

- Tempo diário de execução do processo ETL. Desta forma é possível analisar as variações na atualização do processo ETL;
- A data do último registo das tabelas de facto, para ser possível verificar se o processo ETL falhou durante a sua execução;
- Quantidade de registos que entram nas tabelas de quarentena;
- A diferença de quantidade de registos entre as tabelas de facto do *Data Warehouse* e as *views* de exportação dos sistemas OLAP, nos últimos 30 dias.

O último ponto é importante para detetar as diferenças de dados entre o sistema OLAP e os sistemas OLTP, sabendo que essas diferenças têm de ser sempre nulas pois as *views* de exportação preparam os dados com todos os registos que devem ser inseridos no *Data Warehouse*. Caso existam diferenças, certamente existem casos na quarentena do sistema OLAP, ou então informações anormais nos sistemas OLTP que precisam de ser analisadas com urgência. Portanto, serve este ponto como análise da qualidade da informação sendo objetivo do gestor do *Data Warehouse* manter esta diferença sempre nula.

O gestor do *Data Warehouse* terá de fazer pelo menos uma averiguação diária entre as 2 e as 9 da manhã, sendo que o processo ETL inicia à meia-noite e o sistema de manutenção atualiza logo após ao sistema ETL. O sistema OLTP da Foreva começa a ter mudanças e atualizações nos seus dados a partir das 9:30 (horário de abertura das lojas da Foreva) e é importante que o gestor do *Data Warehouse* analise se não existem erros no processo ETL.

A ferramenta utilizada para desenvolver esta aplicação foi o *Microsoft Power BI*, sendo que a sua arquitetura da aplicação está representada na figura 39.

O *Power BI Service* usa o *gateway* do servidor OLAP para se conectar ao *Data Warehouse* e ao sistema OLTP e utiliza um mecanismo ETL (desenvolvido através do *Power Query*) para criar um conjunto de dados que é armazenado no serviço *cloud* do *Power BI*, e que irá alimentar as informações da aplicação de suporte á gestão do *Data Warehouse*. Este conjunto de dados atualiza logo a seguir ao processo de refrescamento do *Data Warehouse*.

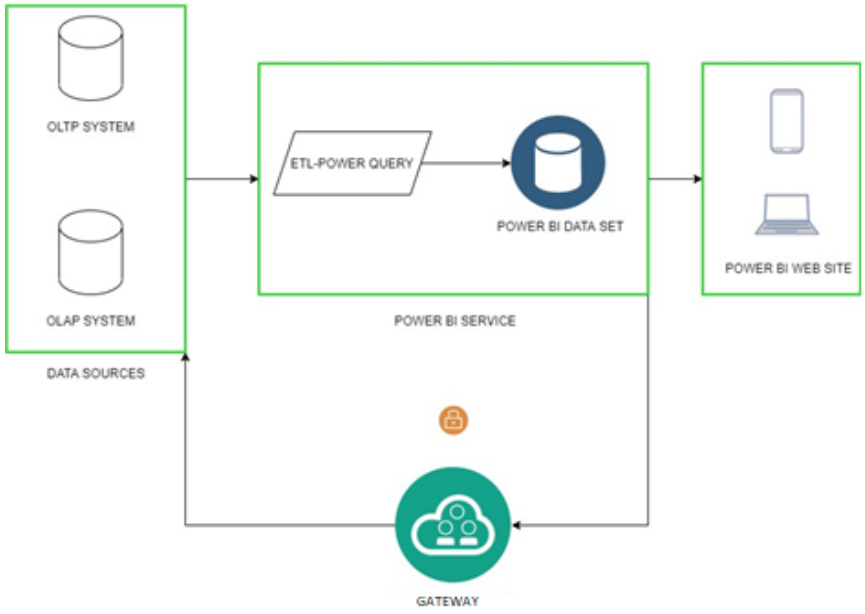


Figura 35 – Arquitetura da Aplicação de Suporte a Manutenção do *Data Warehouse*

A aplicação de manutenção foi desenvolvida e estabelecida através do *Power BI*, sendo que aumentou a performance da gestão e controlo do *Data Warehouse* e do seu processo de atualização de dados. Apesar de apenas ser uma aplicação de consulta permite ao gestor do sistema de armazenamento de dados aceder a qualquer momento e em qualquer local possibilitando a deteção de anomalias. Esta aplicação estabeleceu-se como uma componente importante neste sistema porque cumpriu todos os requisitos estabelecidos anteriormente. É apresentado nas figuras 36, 37, 38 os exemplos de funcionamento da aplicação.

A figura 36 contém a informação da quantidade de registos nas tabelas de quarentena. No exemplo pode ver-se que as tabelas estão vazias. Caso alguma tabela deixe de estar vazia, será necessário a identificação e correção urgente do problema.

Figura 36 – Casos em quarentena

A figura 37 revela os tempos de refrescamento diários do sistema de *Data Warehousing*, assim como as datas dos últimos registos. No exemplo consegue-se verificar que o *Data Warehouse* teve um refrescamento de sucesso na madrugada de dia 04/01/2021 através da informação da tabela de facto "TF_REGISTO_STOCK", que guardou os registos do stock inicial do próprio dia. As outras tabelas de facto também revelam que o último registo verificado foi no dia anterior. Caso as tabelas não tivessem como última data de registo o dia anterior, como é o caso da "TF_CONF_TRANS" que contém como última data de registo o dia 31-12-2020, significa que não existem confirmações de transferências desde esse dia.

Figura 37 – Estatísticas e últimas datas dos registos de refrescamento dos dados

A figura 38 mostra a diferença de registos entre o sistema OLTP e as tabelas de facto do sistema OLAP nos últimos 30 dias. Caso, em algum momento, as linhas dos gráficos se movam para valores diferentes de zero significa que existem diferenças entre o número de registos entre os dois sistemas. O que significa que o *Data Warehouse* não contém a informação toda que devia, ou então que contém informação a mais e é urgente identificar e corrigir o problema. Neste caso pode ver-se que o número de registos dos últimos 30 dias está totalmente igual nos dois sistemas.

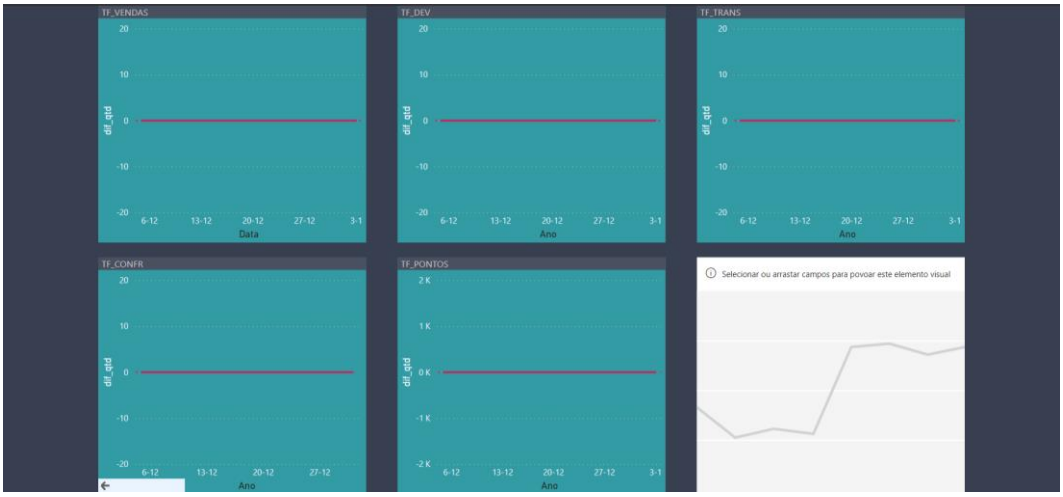


Figura 38 – Diferenças de quantidades de registos entre os sistemas OLAP e dos sistemas OLTP nos últimos 30 dias

Com o processo ETL criado e em funcionamento, assim como a aplicação de manutenção do mesmo, seguiu-se para o desenvolvimento do Sistema de Processamento Analítico.

4.5.5. Sistema de Processamento Analítico

Depois de apresentado o desenvolvimento do *Data Warehouse* e do mecanismo ETL com o respetivo povoamento, este subcapítulo apresenta o desenvolvimento dos cubos Olap que permitem a realização de consultas analíticas. A ferramenta utilizada foi o *SQL Server Analysis Services*.

Por outro lado, também foram desenvolvidas *views* diretamente do *Data Warehouse*, a ferramenta *Power BI* permite a preparação dos dados e muitas vezes pode ser mais útil a utilização de *views* do *Data Warehouse*, do que propriamente a ligação a cubos de dados. Através da ligação OLE DB com o *Data Warehouse* foi configurado o *Data Source View*. Inicialmente foi configurada a ligação ao *Data Warehouse*

através do SSAS e foram geradas as dimensões assim como as suas hierarquias. Na figura 39 é apresentada a dimensão DIM_ARTIGO com a hierarquia Tipo_Artigo configurada.

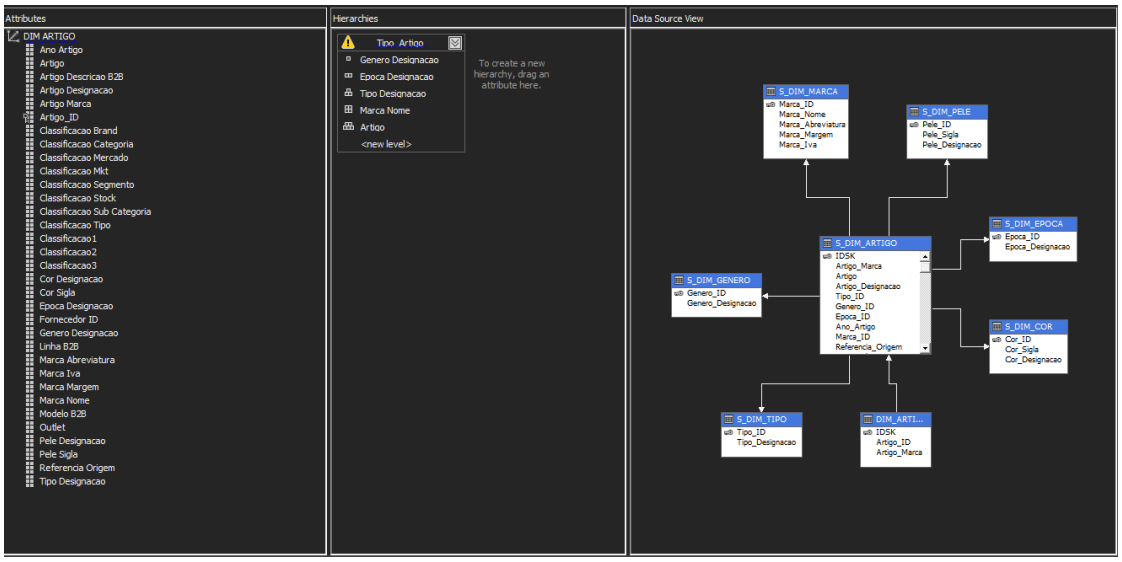


Figura 39 – Dimensão DIM_ARTIGO

Para o caso da análise de stock, a dimensão DIM_ARTIGO teria de ser substituída pela dimensão S_DIM_ARTIGO, de mais baixa granularidade, como é apresentado na figura 40.

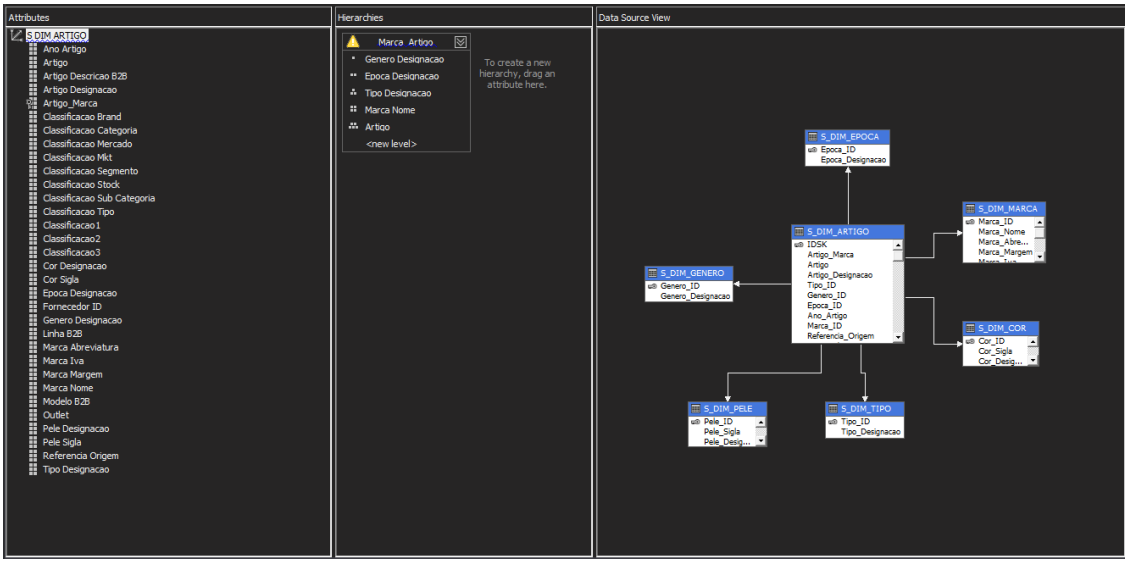


Figura 40 – Dimensão S_DIM_ARTIGO

Todas as restantes dimensões foram configuradas e foi criado um cubo para cada tabela de facto. Usamos como exemplo de demonstração o cubo da tabela de facto vendas com as respetivas medidas e dimensões (figura 41).

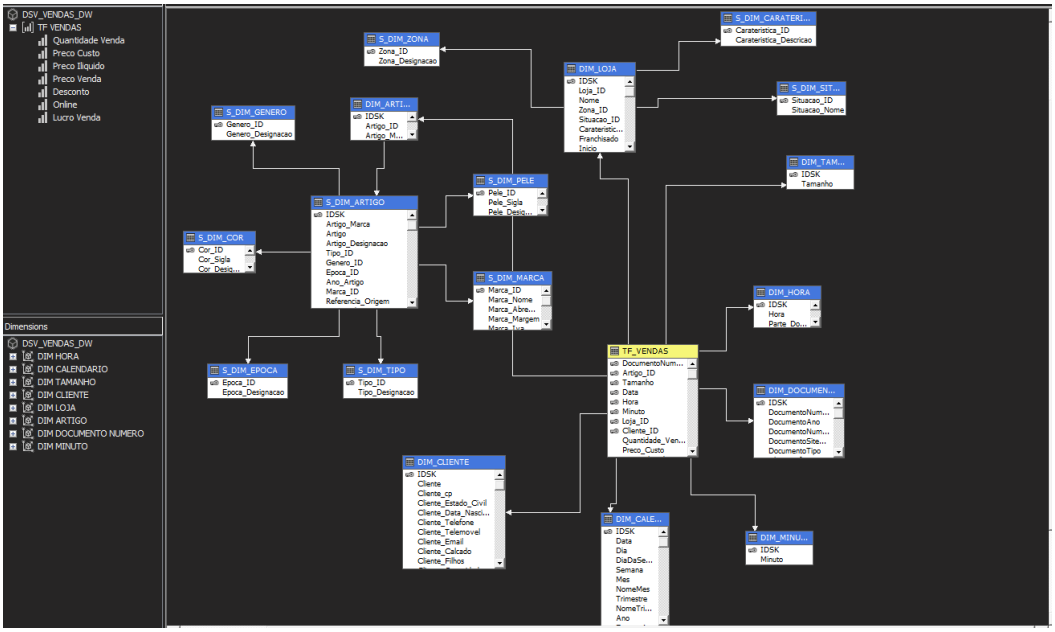


Figura 41 – Cubo OLAP DSV_VENDAS_DW

A utilização de cubos para o processamento analítico torna-se de certa forma extraordinária na feição em que se consegue entregar aos gestores mais informação do que o previsto, saciando problemas reativos e inesperados de falta informação que os tomadores de decisão necessitem no seu dia-dia. Desta forma, o sistema de processamento analítico está preparado para disponibilizar toda a quantidade de informação que seja necessária a qualquer momento, integrando no sistema de *Data Warhousing* mecanismos de pesquisa e análise.

Os cubos são processados diariamente logo após a conclusão do processo ETL.

4.6. Componente de Recomendações

4.6.1. Elaboração de Requisitos

Um dos principais objetivos deste projeto, foi criar uma componente de recomendações com várias variantes, incluindo uma componente de *Machine Learning* capaz de indicar produtos que podem ter

sucesso numa determinada loja com base no comportamento que tiveram em lojas "semelhantes", no sentido de se venderem produtos idênticos.

Os objetivos desta componente de recomendações foram explicados no subcapítulo 3.2.5, sendo que o subcapítulo atual se objetiva a demonstrar a forma como os requisitos para a solução foram elaborados. Na tabela 8, são apresentados os requisitos de funcionais e os requisitos não funcionais desta componente.

Tabela 8 – Requisitos funcionais e não funcionais da componente

Requisitos Funcionais	
RF01	Permitir ao utilizador selecionar a loja que vai analisar
RF02	Permitir ao utilizador selecionar a estação na qual vai efetuar a análise
RF03	Disponibilizar um alerta de artigos em quebra de stock na loja selecionada
RF04	Disponibilizar um alerta de artigos sem vendas a mais de 2 semanas na loja selecionada
RF05	Disponibilizar a informação das palavras-chave de cada loja em relação à categoria dos produtos
RF06	Recomendar artigos para reposição de stock de lojas com stock sem vendas há 2 semanas
RF07	Recomendar artigos com base nos resultados de vendas dos artigos de outras lojas (Modelo ML Filtragem Colaborativa)
RF08	Recomendar os top artigos
RF09	Recomendar os top artigos com as categorias mais vendidas da loja selecionada
RF10	Todos os dados utilizados terão como fonte o *Data Warehouse* desenvolvido
Requisitos Não Funcionais	
RNF01	A componente deve gerar as recomendações e disponibilizar a informação pretendida num tempo próximo do instantâneo.

4.6.2. Arquitetura do Sistema de Recomendações

A arquitetura desta componente é constituída pela fonte de dados que é o *Data Warehouse* desenvolvido ao longo do projeto. Esta arquitetura é também constituída pela componente chamada motor de recomendações, que é todo o processo que permite gerar as recomendações. Esse processo é dividido em duas fases. A primeira por um *script* em linguagem *python* que utiliza modelos de *Machine learning* desenvolvidos para prever a prestação de um produto numa loja, através de métodos de filtragem colaborativa. O *script* vai também utilizar as informações provenientes do *Data Warehouse* para definir os top artigos que não foram para determinada loja, assim como os top artigos da categoria dessa loja que não deram entrada nessa mesma loja. O *script* irá gerar recomendações e armazena-las na base de dados de recomendações desenvolvida. A segunda fase é relativa à componente *Power BI Service* que se conecta, através de *views*, ao *Data Warehouse* para realizar um processo de transformação de dados desenvolvido, e gerar as recomendações e as informações de rutura de stock. Esta componente também se conecta à base de dados de recomendações para preparar os dados, incorporando-os no serviço de *datasets* do *Power BI*, para depois alimentarem a plataforma de *Business Intelligence*.

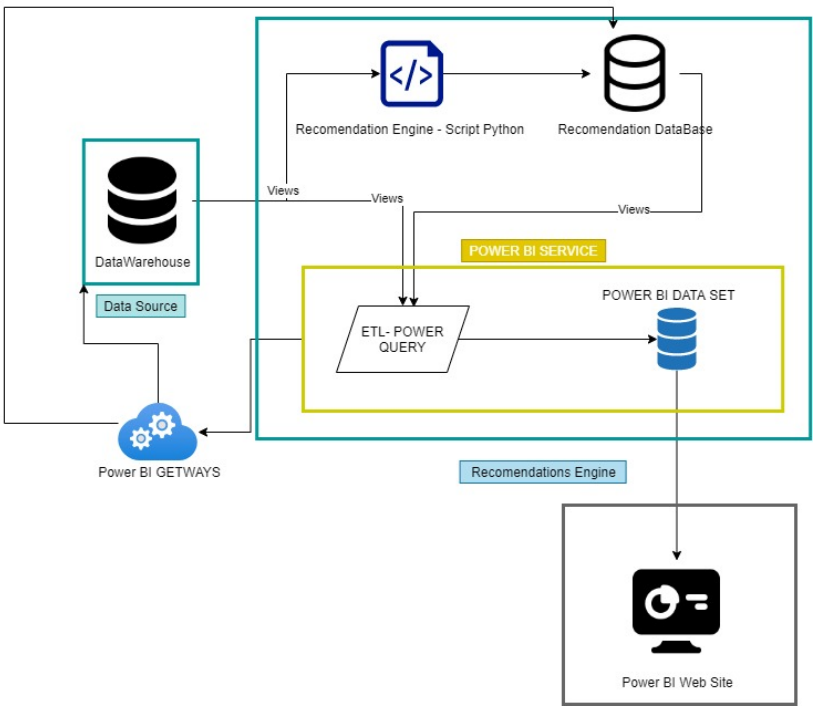


Figura 42 – Arquitetura do Motor de Recomendações

4.6.3. Desenvolvimento do Modelo de Filtragem Colaborativa

A base de dados de recomendação foi desenvolvida de forma a ser possível armazenar diferentes tipos de recomendações. Para além disso, a base de dados está preparada para guardar os diferentes produtos associados às diferentes estações do ano.

A base de dados de recomendações armazena a informação das lojas e dos artigos, assim como as categorias dos artigos que existem. Armazena as informações das estações de vendas e dos três tipos diferentes de recomendações, as de filtragem colaborativa (tabela REC_FC), top n (tabela REC_TN) e top produtos das melhores categorias da loja (tabela REC_BC). Esta base de dados foi construída para armazenar recomendações depois do processo ETL do *Data Warehouse*, ou seja, todos os dias. O utilizador só terá acesso às recomendações do dia atual, sendo que as recomendações dos dias anteriores são guardadas para contexto de análise.

A baixo é apresentada na figura 43 o desenho da base de dados de recomendações. No anexo XIII é demonstrado o seu código de implementação física.

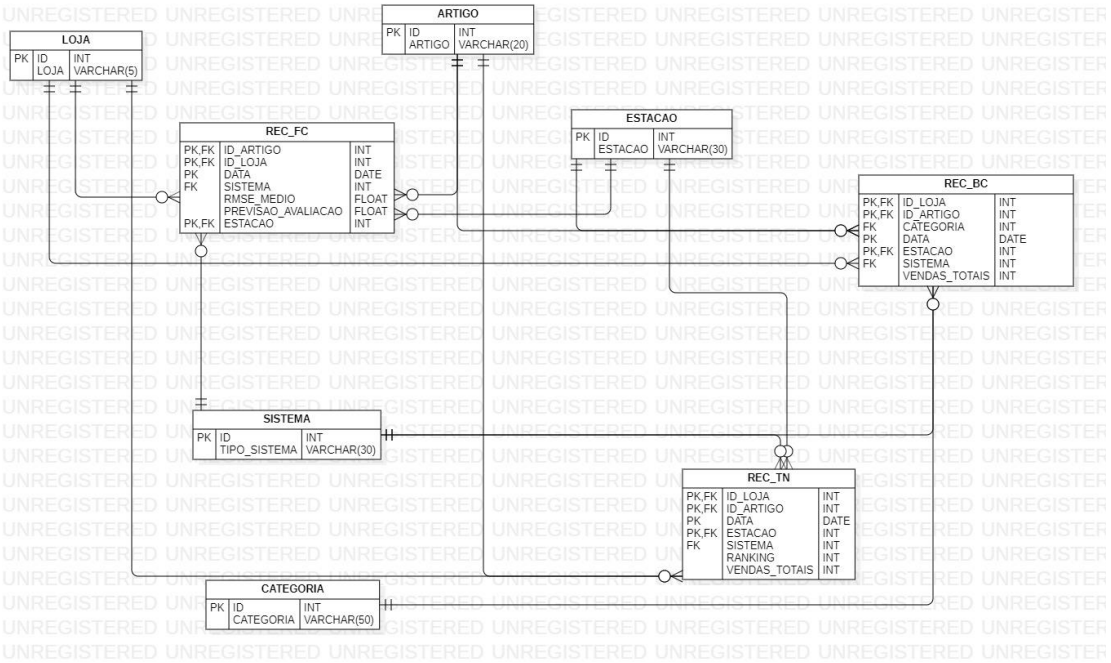


Figura 43 – Base de dados de Recomendações

4.6.3.1. Extração de Dados

O *Data Warehouse* desenvolvido durante o projeto contém a informação dos artigos que são transferidos entre as lojas e também das respetivas vendas. O passo inicial no desenvolvimento deste sistema foi a criação de *views* através do *SQL SERVER*.

Durante a fase de filtragem colaborativa, foi necessário desenvolver 3 *views*, cada uma associada a estações diferentes. Uma delas, a *view* "vw_OI_SR_FC", devolve a informação das avaliações dos artigos em cada loja na estação outono/inverno. A avaliação é feita entre 1 e 6, sendo que 1 significa que o artigo obteve a pior performance possível na loja (0% de vendas sobre as entradas) e 6 que obteve a melhor (100% de vendas sobre as entradas).

As *views* foram desenvolvidas de forma a retornar os registos a partir do início das épocas, por exemplo, ficou definido que a época de outono/inverno começa sempre no dia 1 de outubro, ou seja, sempre que chega esse dia os registos de vendas disponíveis na *view* começam do zero.

No subcapítulo 3.2.5 é explicada a razão pela qual este tipo de recomendações é definido com o histórico inicial sempre no começo de cada época, esquecendo os registos anteriores. Ao iniciar a época de outono/inverno de 2020, no dia 1 de outubro de 2020, começa a ser possível visualizar, nesta *view*, as avaliações dessa época e deixa de ser possível a visualização os dados da época outono/inverno anterior, neste caso de 2019.

Caso a meio da época um artigo contenha 11 entradas numa loja e 6 vendas nessa mesma loja, a sua avaliação estará em 3.65 nessa loja, na escala de 1 a 6. Caso esse artigo só contenha uma venda nessa loja, e 6 entradas de stock, a avaliação será 1.85. A fórmula utilizada na *view* para calcular as avaliações é: ((qtd vendas/qtd entradas) *5 + 1).

Assim, são desenvolvidas as 3 *views* que contêm as avaliações dos artigos nas lojas, tanto para outono/inverno, como para primavera/verão e artigos sem estação. As respetivas datas de início de cada uma das 3 estações são definidas pelos gestores, sendo que para a época de primavera/verão de 2020 ficou definida a data de início em 1 de janeiro de 2020. Estas datas podem ser alteradas a qualquer momento na respetiva *view* "vw_PV_FC". A *view* que contém as avaliações dos artigos sem estação não contém nenhuma data de início de estação, ou seja, contém o histórico total das vendas, sendo que para os gestores as alterações de época e a moda não contêm tanta influencia nos padrões de vendas destes artigos. A *view* é chamada de "vw_TE_SR_FC".

Na figura 44 pode observar-se os resultados da *view* correspondente às avaliações dos artigos nas lojas durante a época de outono/inverno.

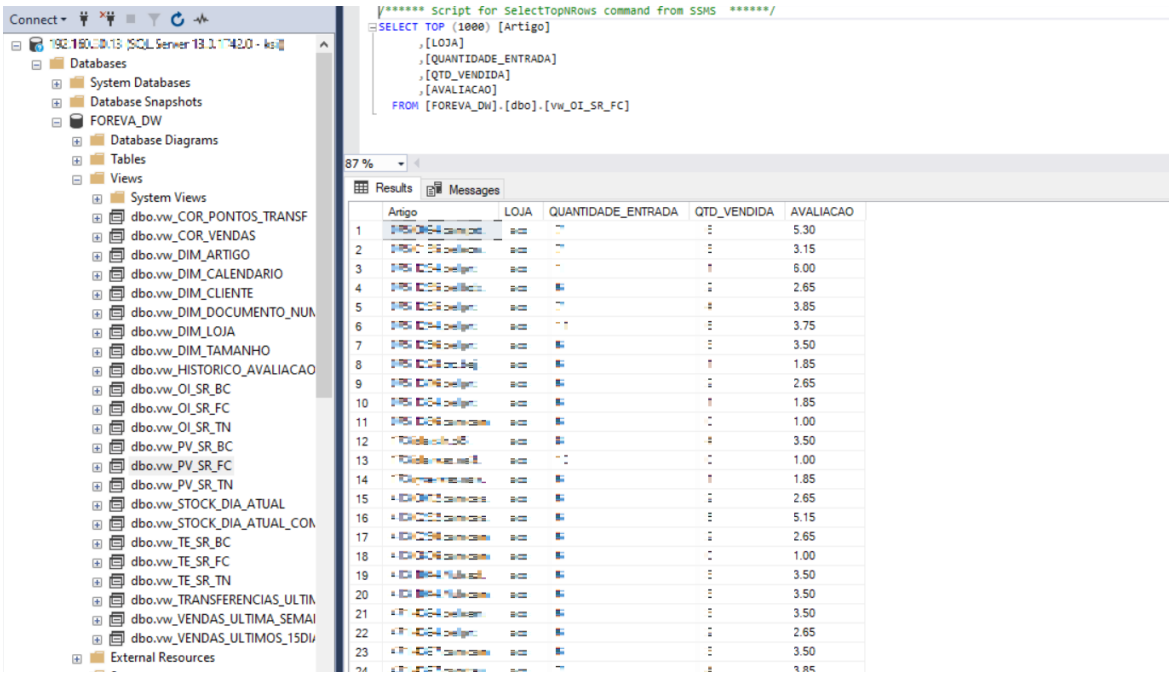


Figura 44 – Resultados da view "vw_OI_SR_FC"

4.6.3.2. Motor de Recomendações de Filtragem Colaborativa

Foi desenvolvido um *script* .py em que o objetivo será gerar recomendações de filtragem colaborativa como descrito no subcapítulo 3.2.5. O *package* que foi utilizado neste contexto foi o *Surprise*. O *Surprise* é um pacote complementar da linguagem *python* que permite construir sistemas de recomendação através de algoritmos de previsão de *Base Line*, K-Vizinhos Mais Próximos e Matrizes de Factorização. Contém também embutido várias formas do cálculo de medidas de similaridade.

Este serviço desenvolvido em linguagem *Python* está responsável por:

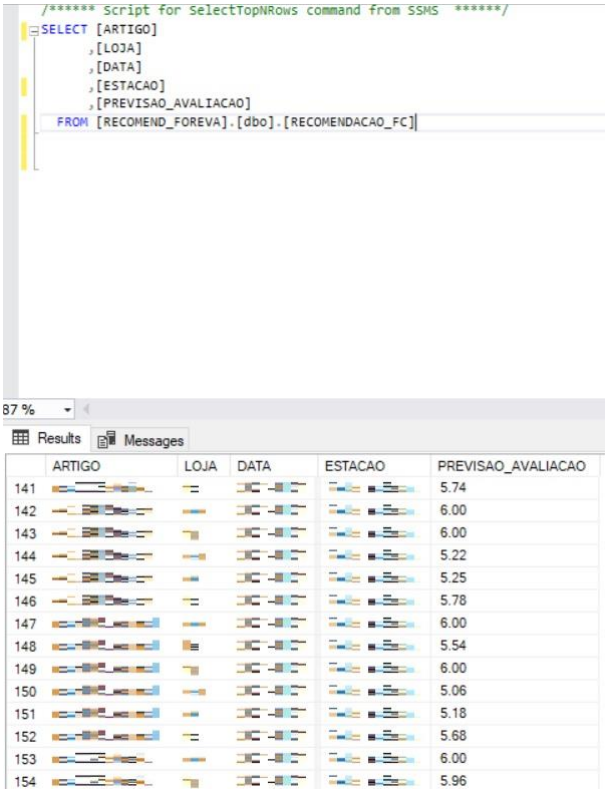
- Importar os dados do *Data Warehouse* (Através da ligação as *views* desenvolvidas) para construir o modelo;
- Calcular a previsão de avaliações através do modelo *Machine Learning*;
- Selecionar as melhores avaliações previstas de produtos nas lojas;
- Inserir as recomendações na base de dados de recomendações;

No primeiro ponto, o serviço consegue aceder aos dados necessários através das *views* correspondidas. Como é necessário dividir as épocas, o serviço é executado 3 vezes para os artigos primavera/verão, outono/inverno e artigos sem estação. Com esses dados é desenvolvido o modelo de *Machine Learning*.

No segundo ponto, é utilizado o modelo desenvolvido com o objetivo de gerar as recomendações. Nesta fase, o input são todas as possíveis combinações de artigos e lojas que não contém avaliações, ou seja, em que o artigo não teve entradas de stock nessa mesma loja. O modelo gera previsões da avaliação que cada artigo teria nessas respetivas lojas, caso seja dada entrada de stock.

Como nem todas as avaliações previstas são de boa performance, a terceira fase do serviço será selecionar as melhores previsões para cada loja. Por fim, com todas as avaliações de boa performance escolhidas o serviço irá verificar cada recomendação e inserir na base de dados de recomendações.

Na figura 45, observa-se uma *view* da base de dados de recomendações. A *view* contém algumas das recomendações geradas pelo serviço no dia 31 de janeiro de 2021.



	ARTIGO	LOJA	DATA	ESTACAO	PREVISAO_AVALIACAO
141					5.74
142					6.00
143					6.00
144					5.22
145					5.25
146					5.78
147					6.00
148					5.54
149					6.00
150					5.06
151					5.18
152					5.68
153					6.00
154					5.96

Figura 45 – *View* da Base de Dados de Recomendações (Tabela Rec_FC)

4.6.3.3. Análise e Preparação de dados

Antes do desenvolvimento do motor de recomendações, foi necessário analisar os dados para se entender o comportamento das avaliações dos artigos nas lojas. Sabe-se que a cada início de época as avaliações iniciam do zero, no entanto, foi durante a época primavera/verão do ano de 2020 que a

análise aos dados foi efetuada. É importante que a análise seja efetuada durante todas as épocas para detetar mudanças nos comportamentos dos artigos, das lojas e das avaliações.

Alguns dos pontos importantes na fase de análise de dados para o desenvolvimento do sistema de recomendação foram:

- Quantidade de Artigos
- Quantidade de Avaliações
- Número de Avaliações por Loja
- Número de Avaliações por Artigo
- Distribuição das Avaliações

Como exemplo, é apresentado o gráfico na figura 46, relativo ao último ponto, (Distribuição das Avaliações). Observa-se que a maioria das avaliações são 1 ou 6. Ou seja, neste caso, a maior parte dos artigos colocados nas lojas, ou são todos vendidos ou nenhum deles é vendido.

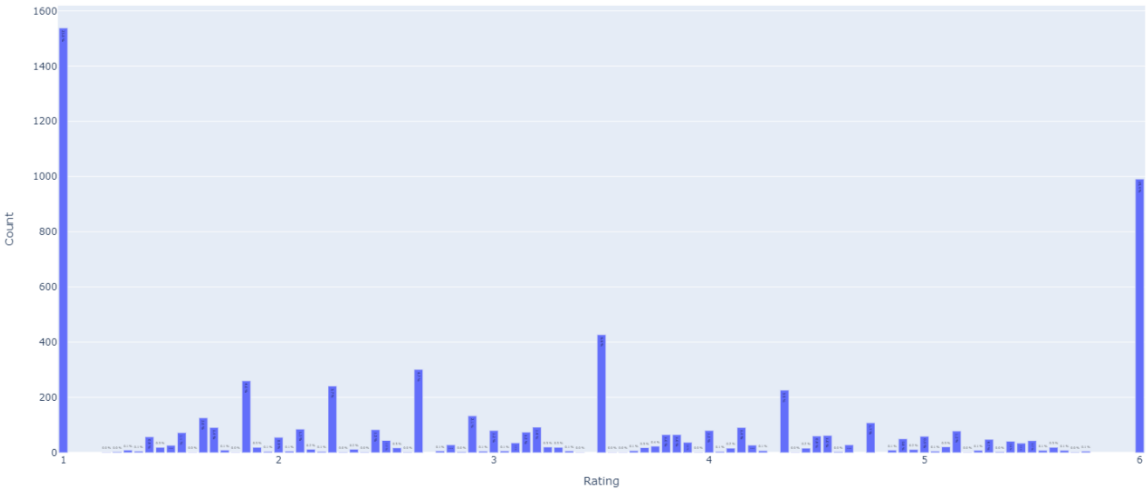


Figura 46 – Distribuição de Avaliações

O gráfico em cima permitiu detetar um problema em relação às avaliações muito altas. Ao examinar os dados dessas mesmas avaliações foi verificado que muitas delas apenas continham uma entrada e uma venda do respetivo artigo numa loja, ou seja, segundo a fórmula para calcular a avaliação fornecida no subcapítulo 4.6.3.1, o resultado dessas avaliações será 6. O problema causado neste tipo de casos é que apenas um artigo a entrar na loja é insuficiente para ser gerada uma avaliação sobre esse artigo, muito menos uma avaliação máxima. Para resolver esse problema, foi desenvolvida uma validação no *script* .py em que os dados de avaliações que são importados das *views* terão

obrigatoriamente de conter mais de 5 artigos de quantidade de entrada na loja para os dados serem utilizados na modelação.

No início de cada época, como o histórico inicia do zero foi necessário desenvolver uma validação em que o algoritmo só é executado se já contiver no mínimo 1000 registos nas *views*. Por exemplo, se iniciar a época de outono/inverno, vão haver 0 registos na *view* "vw_OI_SR_FC", e no código esta estação não passará na validação e não serão geradas recomendações para esta época. Quando a view "vw_OI_SR_FC" contiver no mínimo 1000 registos, poderá começar a haver recomendações pois o código é processado. No final de cada época a espectativa do número de avaliações das *views* referentes a primavera/verão e outono/inverno é entre 6 a 10 mil, enquanto que a *view* das avaliações dos artigos sem estação segue com cerca 2 mil registos.

Estas e mais algumas decisões foram tomadas com base numa análise rigorosa dos dados. Um dos processos que foi estruturado no *script* foi que os dados de um determinado artigo só poderiam ser processados caso esse artigo tivesse mais de 5 avaliações, ou seja, fosse no mínimo para 5 lojas. Isto porque, com base na análise de dados, existem alguns artigos de épocas anteriores em que o número de stock é baixo e estão à venda em poucas lojas, não fazendo sentido serem processados no algoritmo. Com esta validação, os artigos que são processados no modelo são os que contêm mais stock para poderem ser recomendados.

Para sintetizar, na figura 47 é possível entender o comportamento do *script* .py nas fases de extração e preparação. Apesar de parecer, na figura, que os processos das diferentes estações são executados em paralelo, na verdade são executados um de cada vez. Inicia com a extração dos dados da *view* já descrita no subcapítulo anterior. Posteriormente com a validação da existência de mais de 1000 registos de avaliações. Caso se verifique este requisito, o processo avança para a fase de preparação de dados onde são filtrados todos os registos com artigos que contêm menos de 5 avaliações, e filtrados os registos de artigos em que a quantidade de entrada do artigo na loja é menor que 5. Depois dos dados tratados é modelado o *Machine Learning* de previsão de recomendações.

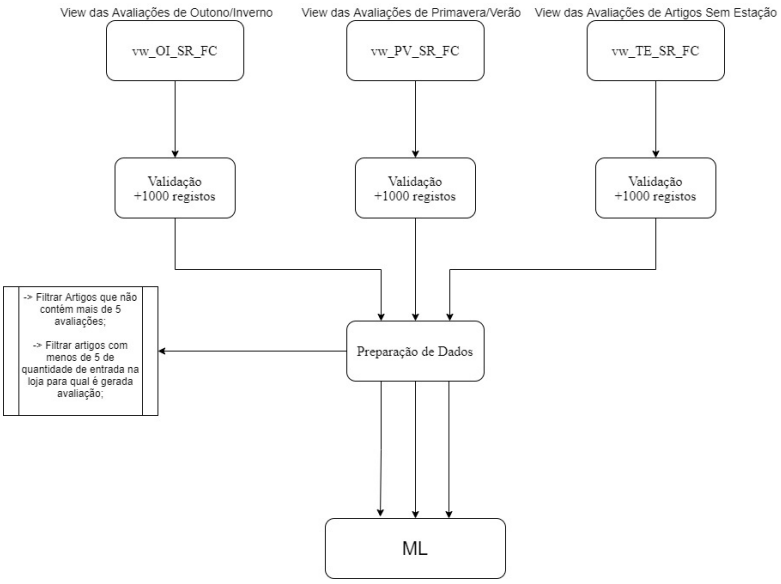


Figura 47 – Processo de Extração e Preparação de Dados

4.6.3.4. Desenvolvimento do Modelo

O desenvolvimento do Modelo de *Machine Learning* foi feito através da extensão *Surprise* da linguagem *Python*. Este *package* contém bastantes algoritmos de *Machine Learning* exclusivos para sistemas de recomendação. Todos os algoritmos disponíveis foram avaliados. Foram também utilizados métodos de validação cruzada, otimização de parâmetros e métricas de precisão incluídos no *Surprise*.

Os algoritmos comparados foram explicados no subcapítulo 2.3.6 e são:

- *Normal Predictor* (Distribuição Normal)
- *Base Line Only* (Método da Linha Base)
- SVD (Algoritmos baseados em Factorização de Matrizes)
- SVD ++ ((Algoritmos baseados em Factorização de Matrizes)
- NMF (Algoritmos baseados em Factorização de Matrizes)
- KNN *Basic* (Métodos baseados em kNN)
- KNN *with Means* (Métodos baseados em kNN)
- KNN *with Z Score* (Métodos baseados em kNN)
- *Slope One*
- *Co-Clustering*

Com base na análise de dados, explicada no subcapítulo anterior, foi possível perceber que existem muitos artigos com a avaliação igual a 1 e igual a 6. Devido a esses extremos o modelo de previsão

poderá ter uma melhor performance ao prever valores mais próximos de 1 ou de 6. Devido a isso, foi usada a medida de precisão RMSE para avaliar os modelos, dando assim, mais ênfase aos grandes erros.

Os modelos foram avaliados, e para trazer confiança aos resultados foi utilizada a validação cruzada, em que os dados utilizados para a avaliação dos modelos foram os da época primavera/verão de 2020. Como exemplo, na figura 48, são apresentados 4 resultados de interações do modelo KNN *Basic* em que o número de vizinhos k é otimizado, sendo que os melhores resultados foram mesmo para k igual a 25 (interação 3) com um RMSE de 1.15.

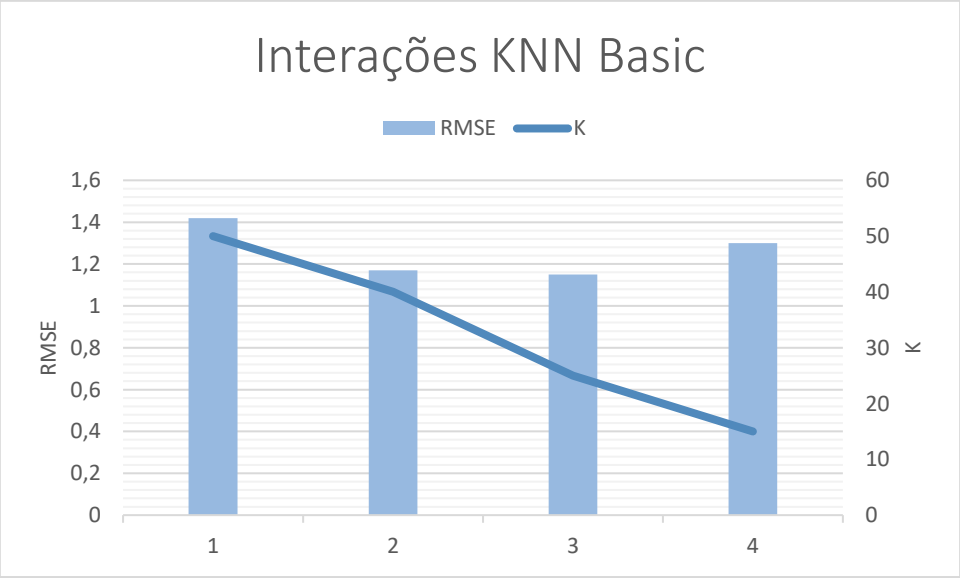


Figura 48 – Resultados das 4 melhores interações do KNN *Basic*

Para todos os algoritmos baseados no KNN foi avaliado se o algoritmo deveria ser baseado no utilizador (loja) ou no produto, e os que apresentaram melhor RMSE foram os baseados no utilizador. Nos exemplos em cima, a interação 1 é baseada no produto e as restantes no utilizador. A medida de similaridade também teve de ser avaliada, sendo que a que teve melhor aproveitamento foi a de *Pearson*.

O resultado obtido depois da otimização de parâmetros de todos os algoritmos foi bastante idêntico, sendo o melhor o do KNN *with z score*. O algoritmo *slope one* e *normal predictor* não passaram pelo processo de otimização de parâmetros pois não têm parâmetros. Na figura 49 são apresentados os melhores resultados para cada modelo. Os resultados dos 5 melhores algoritmos foram considerados satisfatórios pois numa escala de 1 a 6 o RMSE é próximo de 1.

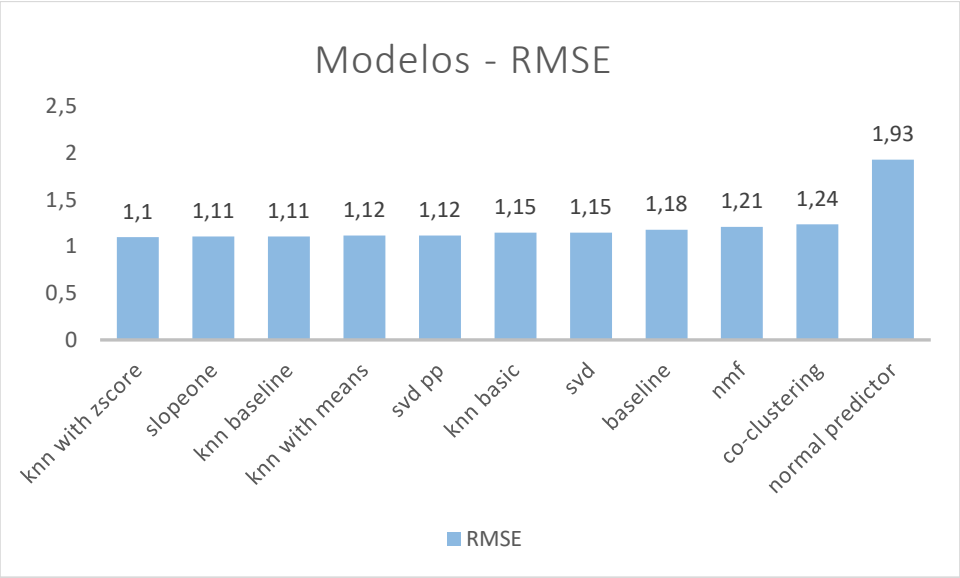


Figura 49 – Melhores resultados de cada modelo

Depois de desenvolvidas as modelagens foi necessário tomar uma decisão na escolha do processo mais adequado. Com base no facto dos resultados dos modelos serem bastante idênticos e também na informação de que existe uma grande variação dos dados desde o início até ao final de cada época, decidiu-se que seria desenvolvido um processo conjunto dos modelos, isto é, os 5 principais modelos iriam ser utilizados.

Durante a execução da fase de modelação no *script*, os modelos knn *with z score*, *slop one*, knn *baseline*, knn *with means* e svd ++ serão treinados através da validação cruzada sendo que o modelo que obtiver menor RMSE será utilizado para prever os resultados. Esta decisão tomou posse porque o número de dados não é muito elevado, e o *script* executa todas as madrugadas logo de seguida ao refrescamento do *Data Warehouse*. Para além disso as diferentes variações no número de registos podem retirar flexibilidade caso se escolhesse apenas um modelo.

Na figura 50 é apresentado o processo algorítmico de decisão do modelo escolhido, onde todos os modelos obtidos na fase anterior são avaliados, através de validação cruzada, com os dados provenientes da *view* já depois da fase de extração e preparação.

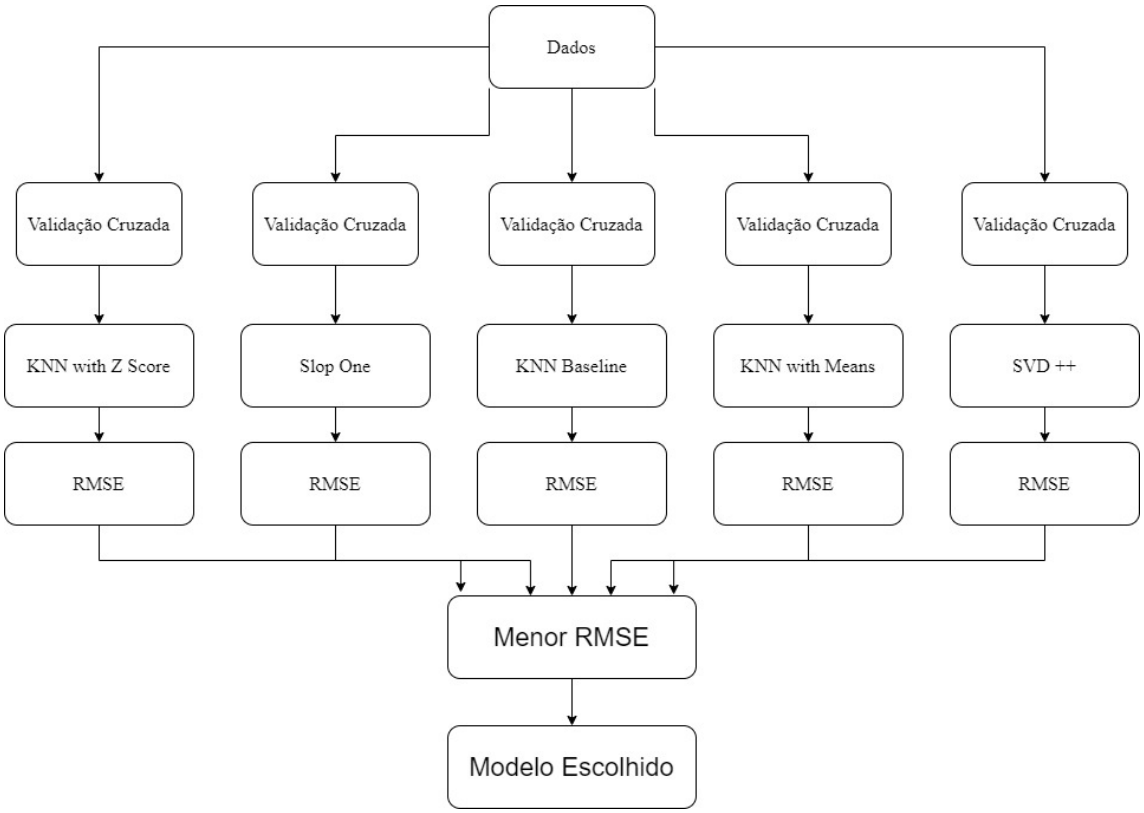


Figura 50 – Algoritmo de escolha do modelo

Os inputs são gerados através de um *cross join* entre todos os artigos e as lojas, e depois são filtradas todas as lojas-artigos que pertencem à tabela de avaliações. Desta forma, o modelo só vai executar previsões em todas as combinações de artigos e lojas que não constituem a tabela de avaliações, ou seja, em que o artigo não foi para a loja respetiva. O modelo faz a previsão da avaliação de cada uma das combinações e valida se a avaliação prevista é superior a 3+ RMSE, caso seja, é gerada uma recomendação, senão a previsão é descartada. Como o modelo é gerado sempre que o *script* executa e em cada execução a quantidade de dados é diferente, o valor do RMSE vai variar sempre. Para evitar recomendações com erros muito elevados aumenta-se o nível de exigência das recomendações somando o valor 3 (valor do meio na escala das recomendações) com o valor do RMSE do modelo gerado, assim se o RMSE for muito elevado, o valor das previsões terá de ser muito alto para as recomendações serem geradas.

Na figura 51 observa-se o processo de gerar recomendações através do modelo *Machine Learning* no *script* em *Python*.

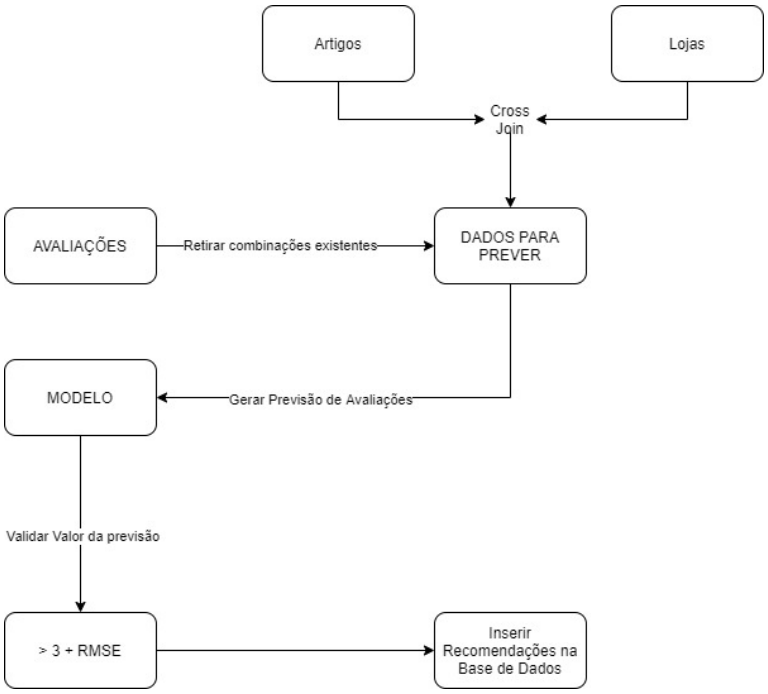


Figura 51 – Processo para gerar recomendações

O *script* foi testado e é executado todas as madrugas logo a seguir ao refrescamento do *Data Warehouse*, para isso foi alocado num servidor em um *container Docker*. Durante todas os processos do *script* existem alertas de erros, por exemplo, se falhar a conexão com o servidor da base de dados, será enviado um email de alerta ao gestor da base de dados de recomendações com o motivo da falha. Assim, o *script* para a filtragem colaborativa foi desenvolvido e caso exista alguma falha no sistema desenvolvido, o gestor do programa será imediatamente alertado com um email e poderá rapidamente tomar medidas para resolver o problema.

4.6.4. Recomendações top N e de transferências de stock

As recomendações top N foram desenvolvidas num *script Python* tal como as recomendações de filtragem colaborativa. Existem 2 tipos diferentes de recomendações de top N, as recomendações dos top produtos mais vendidos e dos top produtos mais vendidos das categorias principais das lojas. Os artigos que são recomendados a cada loja com estas abordagens top N, são artigos que não podem ter entradas de stock para a loja que se recomenda. As categorias principais de cada loja são selecionadas com base nas top vendas de cada categoria, então as categorias principais são selecionadas e é explorado o ranking de top artigos das top categorias para gerar recomendações a cada loja.

As recomendações dos top artigos são salvas na tabela Rec_TN e as recomendações dos top artigos pelas principais categorias das lojas são guardadas na tabela Rec_BC.

Na figura 52 é apresentado o processo para gerar as recomendações top N, o processo é executado para todas as lojas gerando recomendações top N para cada uma.

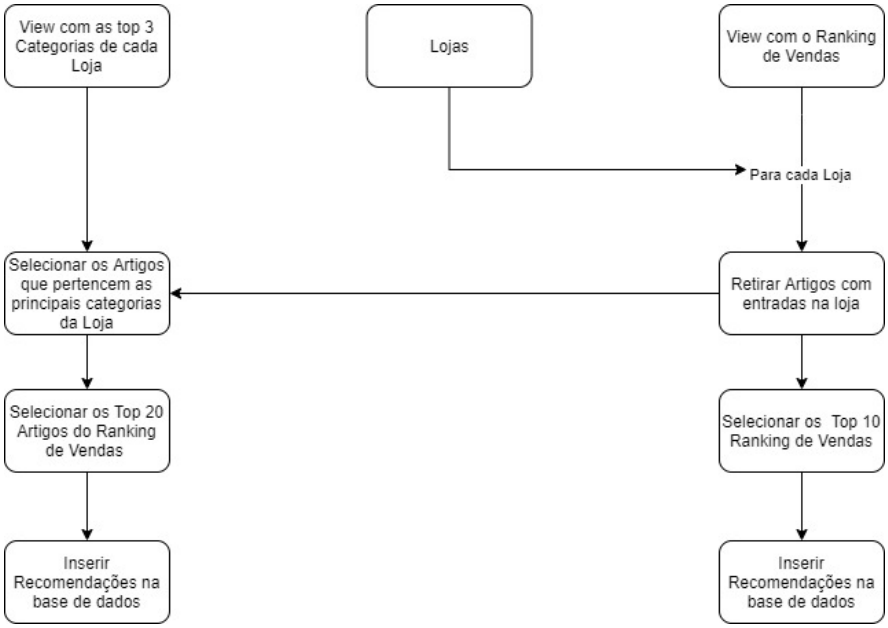


Figura 52 – Processo de recomendações top N

Tal como as recomendações de filtragem colaborativa, estas recomendações são executadas para artigos das 3 épocas e o *script* foi alocado num *container Docker*.

As recomendações de transferência de stock foram desenvolvidas na ferramenta *Power BI*, através da componente *Power Query* (linguagem M) com ligação ao *Data Warehouse*. O processo identifica os artigos com quebra de stock e com vendas na última semana numa determinada loja, e recomenda reposições de stock de outras lojas onde esse artigo não tem tido vendas. A figura 53 apresenta o processo algorítmico para as recomendações de reposição de stock. Como foi dito anteriormente, este processo foi desenvolvido no *Power BI.*

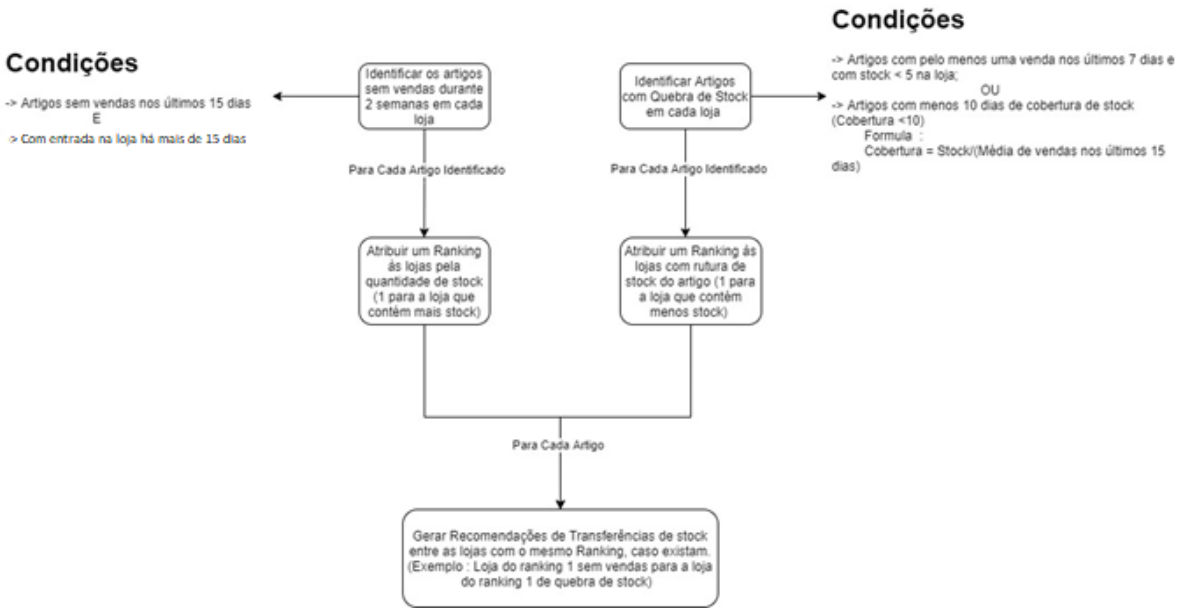


Figura 53 – Processo de Recomendações de Reposição de Stock

A linguagem M é uma linguagem presente na componente *Power Query* da ferramenta *Microsoft Power BI* e foi utilizada para desenvolver o processo descrito em cima. No *Power Query* são chamados de consultas cada pacote de tratamento de dados em linguagem M. Cada consulta guarda para si um conjunto de dados tratados e as consultas podem ser configuradas para comunicar dados entre si. Foi possível desenvolver o presente processo com a criação de algumas consultas no Power Query (figura 53). Na figura 54 é apresentado um exemplo de código da consulta que atribui o ranking aos artigos sem vendas nas lojas. Funciona da seguinte forma, a consulta tem como origem o conjunto de dados de outra consulta que deteta os artigos sem vendas há 2 semanas, depois, utiliza a função *RankFunction* para atribuir, em cada artigo diferente, um ranking das lojas com maior quantidade de stock.

```
let
//Artigos sem vendas nos ultimos 15 dias
    Origem = #"ARTIGOS SEM VENDAS ETL",
    //agrupar tudo pelo artigo
    #"Linhas Agrupadas" = Table.Group(Origem, {"ARTIGO"}, {{"Contagem", each _, type table [ARTIGO=nullable text,
        LOJA=nullable text, STOCK=nullable number]}}),
    //criar uma funcao que vai ordenar cada artigo, neste caso pelo stock
        RankFunction = (tabletorank as table) as table =>
    let
      SortRows = Table.Sort(tabletorank,{{"STOCK", Order.Descending}}),
      AddIndex = Table.AddIndexColumn(SortRows, "RANK", 1, 1)
    in
      AddIndex,
    //Aplicar a função
    AddedRank = Table.TransformColumns(#"Linhas Agrupadas", {"Contagem", each RankFunction(_)}),
    //expandir tudo
    #"Contagem Expandida" = Table.ExpandTableColumn(AddedRank, "Contagem", {"LOJA", "STOCK", "RANK"}, {"LOJA", "STOCK",
        "RANK"}),
    //alterar o rank para o tipo text
    #"Tipo Alterado" = Table.TransformColumnTypes(#"Contagem Expandida",{{"RANK", type text}}),
    //juntar a coluna artigo e rank
    #"Personalizado Adicionado" = Table.AddColumn(#"Tipo Alterado", "id", each [ARTIGO]&[RANK])
in
    #"Personalizado Adicionado"
```

✓ Não foram detetados erros de sintaxe.

Figura 54 – Exemplo do código de uma Consulta no *Power BI (Power Query)*

As grandes diferenças entre as recomendações de reposição de stock, das recomendações de top n e de filtragem colaborativa são, obviamente, os seus objetivos, em que as primeiras têm como fundamento recomendar a reposição de um artigo que já deu entrada na loja e está a obter vendas no momento atual, e as outras mostrar novas oportunidades com artigos que ainda não foram para a loja.

As recomendações top n e de filtragem colaborativa têm uma base de dados em que são armazenadas as recomendações. Posteriormente, está configurada uma conexão da base de dados com o *Power BI* que será a ferramenta de *front-end* das recomendações. No caso das recomendações de reposição de stock, existe uma conexão do *Data Warehouse* ao *Power BI*, mas estas recomendações são geradas através do *Power Query* e não são armazenadas em nenhuma base de dados, sendo eliminadas e geradas novamente sempre que o *Power BI* atualiza os dados das recomendações. A arquitetura apresentada no subcapítulo 4.6.2 representa exatamente as conexões explicadas.

4.7. Sistema de Visualização de Dados (*Front-End*)

Esta etapa é a última do projeto, e enquadra-se no desenvolvimento dos *dashboards* e dos *reports* para a análise pretendida por parte dos gestores. Nesta última fase do projeto foram utilizadas

ferramentas *de front-end* de BI, sendo que a mais utilizada foi o *Power BI* mas também foram desenvolvidos *reports* no SQL *Server Reporting Services*.

O primeiro *report* a ser apresentado foi desenvolvido no SSRS e é um relatório que mostra os dados das vendas totais, os lucros totais por ano da empresa, e também os lucros totais de cada ponto de venda em particular. O nome atribuído a este relatório é "Relatório Geral". Para desenvolvimento deste relatório foi feita uma conexão aos cubos OLAP (SSAS), através da ferramenta SSRS, a conexão é apresentada na figura 55.

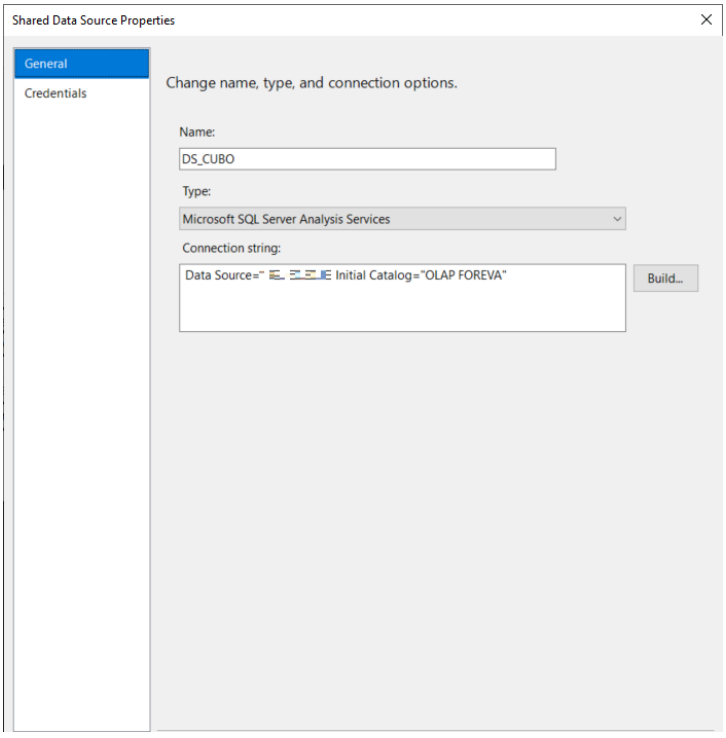


Figura 55 – Conexão aos Cubos OLAP

Depois da conexão criada foram usadas *queries* em linguagem MDX para obter os conjuntos de dados pretendidos através dos cubos para construir o relatório. É apresentada na figura 56 um exemplo de uma *query* MDX desenvolvida para obter os dados da quantidade vendida, preço das vendas, preço de custo e lucro por ano.

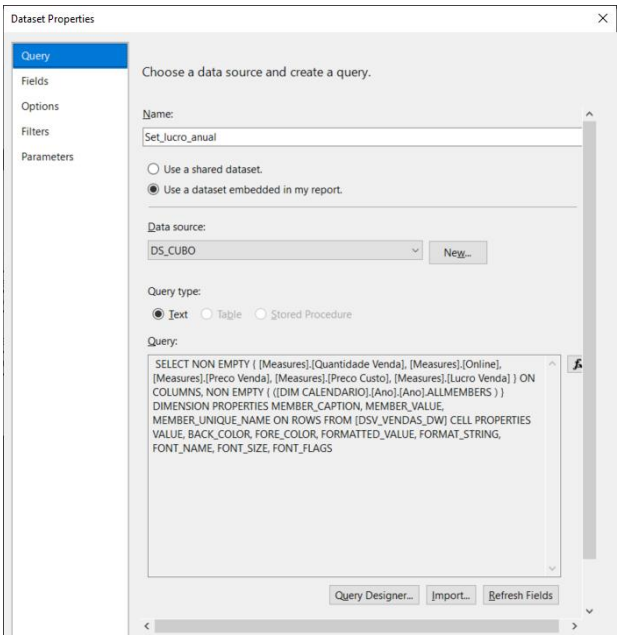


Figura 56 – Conjunto de dados criado a partir da *query* MDX

Depois da obtenção dos dados necessários através da ligação aos cubos Olap e do desenvolvimento das *queries,* foi desenvolvido o relatório que mostra a evolução geral das lojas nos últimos 4 anos. A figura 57 apresenta o relatório geral desenvolvido no SSRS.

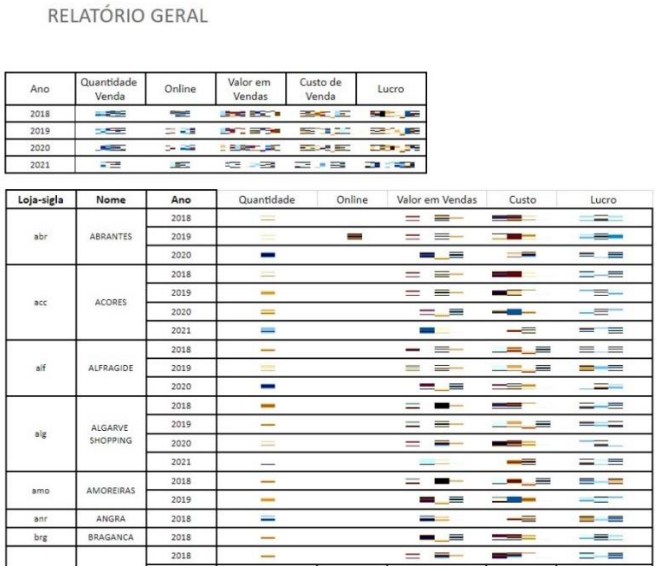


Figura 57 –Relatório Geral

O desenvolvimento dos relatórios e *dashboards* foi baseado nos pedidos dos gestores que são apresentados no capítulo 3. Um dos objetivos pretendidos foi explicado no subcapítulo 3.2.1 e diz respeito à análise de vendas de forma dinâmica onde são pedidos uma série de KPI's. A ferramenta

utilizada para desenvolver este relatório foi o *Power BI* que é uma ferramenta conhecida por permitir construir *dashboards* dinâmicos. Quando é utilizada a ferramenta *Power BI,* as conexões podem ser feitas através dos cubos OLAP ou diretamente ao *Data Warehouse*. Com ligações diretas ao *Data Warehouse* o *Power BI* permite desenvolver medidas calculadas e ter mais opções de tratamento de dados no *Power Query*, o que é uma grande vantagem. Os relatórios são desenvolvidos através do *Power BI Desktop*, mas depois são publicados para o *Power BI Web Service*, onde o utilizador poderá consultar online os *dashboards* através da sua conta de utilizador.

A conexão entre o *Power BI* e o *Data Warehouse* é feita a partir de *store procedures* ou de *views* desenvolvidas no *Data Warehouse*. Como exemplo, é apresentada na figura 58 a *store procedure* desenvolvida para consultar os dados da tabela de facto vendas de um determinado ano, e na figura 59 é apresentada a conexão do *Power BI* com a *store procedure*.

```
USE [FOREVA_DW]
GO
/****** Object:  StoredProcedure [dbo].[sp_FT_VENDAS]    Script Date: 02/17/2021 01:33:57 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[sp_FT_VENDAS] --'2020'

@ANO varchar(20)
AS
DECLARE @QUERY as nvarchar(4000),@OPENQUERY as nvarchar(4000), @AA int,@INICIO varchar(20),@FIM varchar(20),@I DATE, @F DATE
SET  @AA  = convert(int,@ANO)-2000
SET  @INICIO  = 1000000 + (@AA-1)*10000 + 1*100 + 1
SET  @FIM  = 1000000 + (@AA)*10000 + 12*100 + 31
SET @I =  CAST('20' + SUBSTRING(@INICIO, 2, 2) + '-' + SUBSTRING(@INICIO, 4, 2) + '-' + SUBSTRING(@INICIO, 6, 2) AS date)
SET @F = CAST('20' + SUBSTRING(@FIM, 2, 2) + '-' + SUBSTRING(@FIM, 4, 2) + '-' + SUBSTRING(@FIM, 6, 2) AS date)




      SELECT  D.[DocumentoNumero_ID] AS ID_DOCUMENTO
      ,AA.Artigo_Marca AS ID_ARTIGO
      ,T.[Tamanho] AS ID_TAMANHO
      ,C.[Data] AS ID_DATA
      ,H.[Hora] AS ID_HORA
      ,M.[Minuto] AS ID_MINUTO
      ,L.[Loja_ID] AS ID_LOJA
      ,CC.Cliente AS ID_CLIENTE
      ,[Quantidade_Venda] AS QTD_VENDIDA
      ,[Preco_Custo] AS CUSTO
      ,[Preco_Iliquido] AS PRECO_ILIQUIDO
      ,[Preco_Venda] AS PRECO_VENDA
      ,[Desconto] AS DESCONTO
      ,[Online] AS QTD_ONLINE
      ,[Lucro_Venda] AS LUCRO
      ,@ANO AS ANO_ESCOLHIDO
      ,CASE WHEN GETDATE() > DATEFROMPARTS(@ANO,'12','31') THEN DATEFROMPARTS(@ANO,'12','31') ELSE GETDATE() END AS DIA_HOJE
  FROM [FOREVA_DW].[dbo].[TF_VENDAS] AS V
  LEFT JOIN DIM_DOCUMENTO_NUMERO AS D ON V.DocumentoNumero_ID=D.IDSK
  LEFT JOIN DIM_ARTIGO AS A ON V.Artigo_ID=A.IDSK
  LEFT JOIN S_DIM_ARTIGO AS AA ON A.Artigo_Marca=AA.Artigo_Marca
  LEFT JOIN DIM_TAMANHO AS T ON V.Tamanho=T.IDSK
  LEFT JOIN DIM_CALENDARIO AS C ON V.Data = C.IDSK
  LEFT JOIN DIM_HORA AS H ON V.Hora=H.IDSK
  LEFT JOIN DIM_MINUTO AS M ON V.Minuto=M.IDSK
  LEFT JOIN DIM_LOJA AS L ON V.Loja_ID=L.IDSK
  LEFT JOIN DIM_CLIENTE AS CC ON V.Cliente_ID = CC.IDSK
  WHERE C.Data BETWEEN @I AND @F
  and (L.Situacao_ID = 1)
```

Figura 58 – *Store Procedure* sp_FT_VENDAS

Figura 59 – Conexão do *Power BI* ao *Data Warehouse*

Posteriormente à criação das conexões com o *Data Warehouse*, o *Power BI* permite a modelagem das consultas de forma a criar as ligações necessárias entre os dados para a criação do relatório. Na figura 60 é apresentada a modelação entre consultas utilizada neste exemplo.

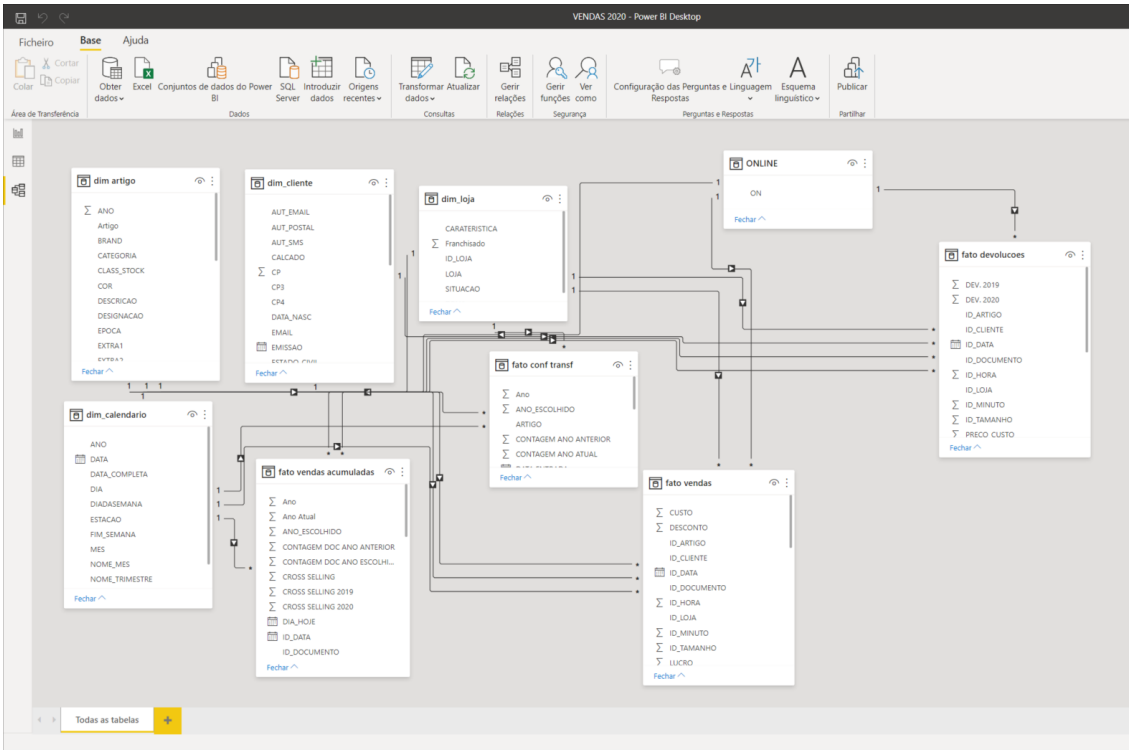


Figura 60 – Modelação no *Power BI*

Depois de criada a modelação das consultas do *Power BI* foram desenvolvidas algumas medidas calculadas que permitem a criação dos KPI's objetivados para o relatório. Na figura 61 é apresentado o desenvolvimento da medida calculada "margem de lucro" através de linguagem DAX no *Power BI*.

```
1 MARGEM LUCRO = SUM('fato vendas'[LUCRO])/SUM('fato vendas'[PRECO_VENDA])
```

Figura 61 – Exemplo da criação de uma medida calculada em DAX no *Power BI*

Ao serem reunidas todas as condições para o desenvolvimento do relatório, este foi criado e publicado e pode ser acedido pelo gestor sempre que o mesmo assim o entender.

Na figura 62 é apresentado o relatório de análise de vendas com todos os indicadores de performance solicitados pelos gestores e com uma forma dinâmica de interação. Como se pode observar no relatório, na parte de cima aparecem os KPI's pretendidos pelos gestores. O centro é composto por alguns gráficos alusivos ao KPI que está selecionado, que no caso da figura 66, é a variação da quantidade de vendas. Se o utilizador selecionar outro KPI os gráficos são alterados para gráficos alusivos ao KPI selecionado. Por exemplo, no caso da figura a baixo, um dos gráficos mostra a variação anual por mês da quantidade de vendas, mas se o utilizador selecionar o KPI da variação do *cross selling*, o gráfico será convertido para a variação anual por mês do *cross selling*.

O utilizador ainda poderá selecionar os filtros disponíveis no canto superior direito, para filtrar só vendas online ou só vendas físicas. Também é possível filtrar os dados do relatório por lojas específicas. Então, o relatório de análise de vendas é constituído por:

- Conjunto de Indicadores de Performance das Vendas;
- Gráfico com a variação homóloga anual por mês do KPI selecionado;
- Gráfico com a variação total anual do KPI selecionado;
- Tabela com a variação homóloga anual por loja do KPI selecionado;
- Possibilidade de filtrar por vendas online ou por vendas físicas;
- Possibilidade de filtrar os dados por uma ou mais lojas.

Os indicadores de performance que este relatório contém são todos os que foram pretendidos pelos gestores, e são:

- Variação Homóloga da Quantidade de Pares Vendidos (QTD VENDAS);
- Variação Homóloga do Volume de Vendas (VOL.VENDAS);
- Variação Homóloga da diferença entre o Volume de Vendas e Custo de Vendas (V-C);

- Variação Homóloga da diferença entre o Volume de Vendas, o Custo de Vendas e o Volume de Devoluções (V-C-D);
- Variação Homóloga do Volume de Devoluções (DEVOLUÇÕES);
- Variação Homóloga da Margem de Lucro (ML);
- Variação Homóloga do Preço Médio de Venda (PREÇO M VEND);
- Variação Homóloga do Valor Médio por Transação (VAL. MÉD TRAN.);
- Variação Homóloga da Quantidade Média por Transação (QTD. MÉD TRAN);
- Variação Homóloga do *Cross Selling* (*CROSS SELLING*);
- Variação Homóloga dos resultados das transferências entre lojas (RESULT. TRANSF).

Figura 62 – Relatório de Análise de Vendas

Este relatório de vendas permite aos gestores ter uma visão geral do comportamento de cada loja e do conjunto total de lojas, no que diz respeito aos indicadores performance abordados, e facilmente identificar os pontos fortes e fracos de cada ponto de venda. Este relatório permite também analisar o comportamento das vendas online e vendas físicas o que pode ajudar a perceber o caminho que empresa deve seguir no que diz respeito às estratégias de vendas.

Em resposta ao objetivo do subcapítulo 3.2.2. (Análise dinâmica das Lojas), foi desenvolvido um relatório dinâmico em que o utilizador filtra os dados por uma loja e um período de tempo e este apresenta determinadas informações numa tabela. O relatório foi desenvolvido na ferramenta *Power BI* e seguiu o mesmo procedimento que o relatório anterior.

Na figura 63 podemos observar o relatório de análise dinâmica das lojas, onde o utilizador pode selecionar o respetivo período de tempo que pretende analisar. Pode também selecionar a loja, que no caso da imagem é a loja do Braga Parque, e depois recebe uma série de informações para cada artigo. As informações apresentadas no relatório para cada artigo da loja selecionada são:

- Stock inicial na respetiva data inicial selecionada;
- O número de entradas de stock durante o período de tempo selecionado;
- O número de vendas durante o período de tempo selecionado;
- O número de vendas online durante o período de tempo selecionado;
- O número de devoluções durante o período de tempo selecionado;
- O número de devoluções online durante o período de tempo selecionado;
- O número de saídas da loja durante o período de tempo selecionado;
- O stock final da loja na respetiva data final selecionada;
- O stock em transação para a loja na respetiva data final selecionada;
- O *Sell-Through* durante o período de tempo selecionado;
- O custo de vendas no período de tempo selecionado;
- O valor das devoluções no período de tempo selecionado;
- O valor das Vendas no período de tempo selecionado;
- A média de descontos no período de tempo selecionado;
- A diferença entre o valor das vendas e o custo no período de tempo selecionado;
- A diferença entre o valor das vendas, o custo e as devoluções no período de tempo selecionado;
- A margem de lucro no período de tempo selecionado.

Figura 63 – Relatório de Análise Dinâmica das Lojas

Através deste relatório os gestores têm uma visão do que cada artigo representa para a loja em estudo num determinado período de tempo, podendo trazer um grande impacto nas tomadas de decisão a curto, médio e longo prazo. Por exemplo, este relatório permite aos gestores avaliarem rapidamente a performance de um artigo caso seja aplicado um determinado desconto, ou comparar as margens de lucro com o lucro total de determinados artigos. Poderá ajudar também os gestores a prepararem as próximas épocas de forma a analisarem os artigos com melhor e pior comportamento na época homóloga anterior.

O próximo *dashboard* a ser apresentado, surgiu para concluir os objetivos do subcapítulo 3.2.3 (Análise dinâmica Artigo). Na figura 64 pode observar-se o respetivo *dashboard*. Este *dashboard* foi desenvolvido para analisar os comportamentos de um determinado artigo, ou de um determinado conjunto de artigos. O utilizador escolhe o artigo que quer analisar colocando o nome do mesmo no *search* do canto superior direito e consegue escolher o intervalo temporal que pretende analisar no filtro de data. O *dashboard* mostra uma serie de 4 visualizações. O gráfico do canto superior esquerdo apresenta o comportamento do artigo nas lojas, com a informação do stock inicial e final do número de vendas e devoluções. Este gráfico permite aos utilizadores perceberem em que lojas o artigo tem um melhor e pior desempenho num período de tempo específico. A tabela do lado superior direito tem basicamente a informação do gráfico só que em modo tabela para facilitar a exportação dos dados ao utilizador.

Em relação às visualizações inferiores, a da esquerda é um gráfico que apresenta o stock inicial, vendas e lucro do artigo, no conjunto total das lojas, para cada semana do período selecionado. Este gráfico permite ao utilizador analisar se a quebra de vendas de um artigo, em algum período de tempo específico, foi causada pela quebra de stock e em que semanas o artigo pode ter mais vendas. A tabela do lado inferior direito mostra o stock atual do artigo (última atualização do DW), independentemente do período selecionado.

Figura 64 – Relatório de Análise Dinâmica do Artigo

Este *dashboard* permite aos utilizadores analisar o comportamento de qualquer artigo em qualquer momento, sendo indicado para tomar decisões como por exemplo, decidir colocar um artigo em desconto num determinado período de tempo, ou então, fazer uma campanha de marketing para um artigo numa determinada loja num determinado horizonte temporal.

A análise do cliente foi também um fator importante para a componente de *reporting*, foi focada no objetivo definido no capítulo 3.2.4 (Análise do Cliente). Foi desenvolvido um relatório capaz de trazer informações úteis aos utilizadores de forma a que estes possam avaliar o comportamento dos clientes associados. Foram definidos uma série de indicadores de performance para análise dos clientes e foram todos desenvolvidos no relatório. O relatório é interativo e foi desenvolvido com o *Power BI*. Os indicadores definidos foram:

- A variação Homóloga do número de clientes com registo de cartão cliente (clientes associados);
- A variação Homóloga do número de clientes associados que fizeram compras nas lojas;
- A taxa de *Churn* dos Clientes associados (% de Clientes associados que compraram no ano anterior, mas não no ano atual);
- A variação Homóloga da quantidade de vezes que se utiliza o cartão cliente.

O relatório é apresentado na figura 65. Neste caso, o indicador selecionado é a taxa de *churn* dos clientes. Os gráficos apresentados variam de acordo com o indicador selecionado. Neste caso como está selecionado o *churn rate* os gráficos têm os dados associados a esse indicador. São 4 KPI's (parte de cima do dashboard) que o utilizador pode selecionar, cada um vai responder aos pontos descritos em

cima. No gráfico superior é possível analisar a variação homóloga por mês do indicador selecionado, enquanto que no gráfico inferior aparece a variação homóloga do indicador selecionado para cada loja. As 3 visualizações do lado direito apresentam a variação homóloga do indicador selecionado e a variação total do indicador selecionado.

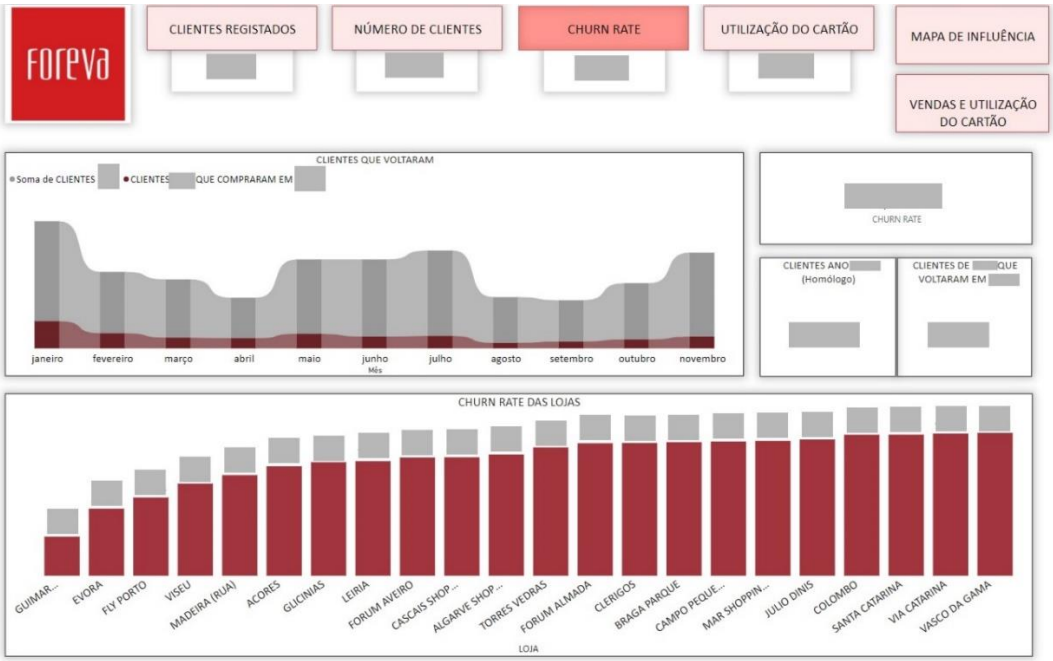


Figura 65 – Relatório de Análise de Clientes

Com este relatório, os utilizadores conseguem gerir os seus clientes associados através de *insights* importantes como por exemplo, o número de clientes que voltou a comprar, a quantidade de novos clientes associados, a quantidade de vezes em que o cliente utilizou o cartão no pagamento. Foi desenvolvido um gráfico para mostrar a relação do lucro da Foreva com o número de utilizações do cartão cliente (tanto para compra como para descontos). O gráfico é apresentado na figura 66.

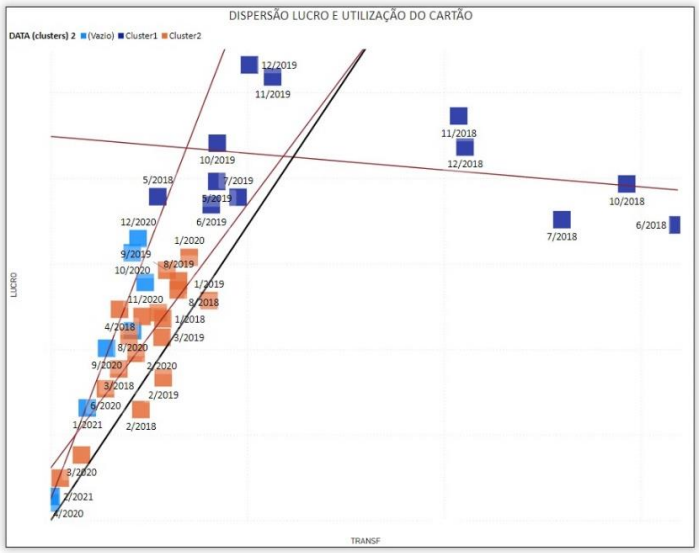


Figura 66 – Gráfico de dispersão do Lucro e Utilização do Cartão por mês

No gráfico desenvolvido, cada quadrado representa um respetivo mês, e é possível observar que nos meses em que os clientes utilizam mais vezes o cartão cliente, são os meses de maior lucro para a empresa. Este gráfico é um forte indício para se concluir que o lucro da empresa e o número de utilizações do cartão cliente estão fortemente correlacionados positivamente.

Nesta Componente de Relatórios foi também desenvolvido o *front-end* dos algoritmos de recomendação. É apresentado o relatório de recomendações na figura 67.

Figura 67 – Relatório de Recomendações

Neste relatório, o utilizador recebe recomendações e algumas informações das lojas selecionadas. Neste caso, está selecionada a loja do colombo e o filtro de outono/inverno. Na tabela superior esquerda está a informação relativa aos artigos que estão em rotura de stock na loja, ou seja, aos artigos com pelo menos uma venda nos últimos 7 dias e com stock inferior a 5 na loja, ou então, aos artigos com menos de 10 dias de cobertura de stock. Esta tabela ainda mostra a informação da quantidade vendida, do stock e do número de dias de cobertura de stock.

No caso da tabela superior direita, está uma listagem dos artigos sem vendas nos últimos 15 dias, e com entrada na loja há mais de 15 dias, com a informação do stock na loja.

Na tabela superior do meio estão as recomendações de transferência de stock, com informação dos artigos e das lojas que não contêm vendas dos mesmos, para serem transferidos para a loja selecionada.

Na tabela inferior esquerda são apresentadas as recomendações do algoritmo de filtragem colaborativa. São recomendados produtos para a loja selecionada e é apresentada a previsão da avaliação que o produto vai ter se for transferido para a loja. As previsões apresentam um intervalo devido ao RMSE calculado, por exemplo, se a previsão for 5 e o RMSE for 1, o valor da previsão que irá aparecer é "4 – 6", ou seja a previsão do comportamento do artigo é entre 4 a 6.

As duas tabelas à direita da tabela inferior esquerda apresentam as recomendações de top n. A primeira, são as recomendações dos top artigos que estão nas categorias mais vendidas da loja, isto é, se a loja vender muitos produtos da categoria criança, neste tipo de recomendações vai constar os top artigos criança que ainda não foram para a loja selecionada. A segunda tabela, são simplesmente os top artigos que não tiveram stock disponível na loja. Por fim, a última tabela, apresenta um conjunto de palavras que representam as melhores categorias da loja selecionada. As palavras contêm tamanhos diferentes, sendo que as maiores são as categorias com mais vendas. Esta tabela permite ao utilizador ter uma informação rápida das tendências da loja selecionada, na estação selecionada. Desta forma o utilizador tem acesso a um relatório com recomendações para as lojas que pretender analisar. A informação de recomendações está contida somente no relatório dinâmico desenvolvido no *Power BI* não compondo nenhum sistema transacional. Se o gestor quiser dar uso de alguma recomendação terá de utilizar a aplicação de funcionamento da Foreva para efetuar a transferência do artigo.

Depois de desenvolvidos os relatórios foram todos publicados no *Power BI Web* e foi desenvolvido um menu principal para os gestores entrarem na aplicação de BI através do link da página e poderem selecionar o relatório que querem analisar. O menu principal é apresentado na figura 68.

Figura 68 – Menu principal da plataforma de *Reporting*

Neste menu, os utilizadores podem carregar no botão do relatório que pretendem analisar. Os botões estão configurados com o link do respetivo relatório. Todos os

relatórios desenvolvidos através do SSRS têm a sua diretoria configurada no botão "REPORTING SERVICES". À medida que os gestores da Foreva pedem novos relatórios, são acrescentados botões a este menu e são configurados com os links dos novos projetos de *reporting*. O departamento de informática da empresa está responsável pela gestão dos relatórios assim como das suas atualizações e alterações.

Sendo esta informação muito sensível para a empresa cabe aos gestores da Foreva indicar por escrito os utilizadores que podem ter acesso a esta plataforma. A responsabilidade de administrar as contas de acesso ao *Power BI* e ao *Reporting Services* é do departamento de informática, assim como o dever de resolver qualquer tipo de problemas de autenticação. A partir do momento em que um utilizador tem acesso a uma conta para a ferramenta de BI é da sua responsabilidade não revelar as credencias de acesso a outros elementos privados da informação. Desta forma, é preservada a segurança de informação, que é um fator crucial a uma empresa que está presente num mercado competitivo e é elementar para a sua gestão estratégica.

4.8. Avaliação de Resultados

Durante o desenvolvimento desta dissertação, foi notória a importância de uma plataforma de *Business Intelligence* que é capaz de disponibilizar informação em tempo útil agilizando vários processos de análise empresarial e de gestão. Para além de facilitar o acesso à informação, este tipo de projetos

permite a criação de um armazém de dados capaz de integrar *datasets* para componentes de *Data Mining* e *Machine Learning* sem interferir nos sistemas operacionais da empresa promovendo, desta forma, o avanço tecnológico.

Em jeito de análise, pode constatar-se que apesar da grande dimensão do projeto, todos os objetivos propostos foram concluídos com sucesso. Todas as componentes que compõem este projeto vêm a ser úteis no que diz respeito às tomadas de decisão dos gestores da empresa. Tendo em consideração a componente do sistema de recomendação para o conjunto de lojas da Foreva, esta revelou-se bastante desafiadora devido ao facto da quase inexistência de sistemas de recomendação para este fim, sendo que a maior parte deste tipo de algoritmos são desenvolvidos tendo como utilizador o cliente final. Contudo, a criação deste sistema apresenta-se muito útil aos gestores, uma vez que, através das informações fornecidas, este algoritmo permite a mineração de novas oportunidades de vendas e consequentemente a possibilidade de gerar um maior lucro.

5. Conclusões e Trabalho Futuro

5.1. Síntese

O objetivo principal desta dissertação centrou-se no desenvolvimento e implementação de um Sistema de *Bussiness Intelligence* dando suporte à gestão estratégica de uma empresa que atua no setor do retalho da comercialização do calçado.

Para dar suporte ao armazenamento de dados e aos principais indicadores chave de performance foi necessário implementar um sistema de *Data Warehousing* assim como um sistema de ETL automatizado que fosse capaz de canalizar os dados desde as fontes até ao *Data Warehouse* desenvolvido, executando os processos de extração, transformação e carregamento necessários para um refrescamento eficaz dos dados. A elaboração de algoritmos capazes de recomendar transferências de stock entre lojas, e a recomendação de novas soluções de produtos que poderão ter sucesso nas diferentes lojas, foi também definido e implementado. Por fim, foram desenvolvidos relatórios dinâmicos numa componente de *front-end* que permite a exploração e visualização da informação. O elemento de análise de dados na ótica do utilizador não faz parte dos objetivos deste projeto.

Na tabela 9 são apresentados os objetivos definidos e a indicação do cumprimento dos mesmos.

Tabela 9 – Objetivos do projeto

Objetivo	Cumprimento
1 - Investigação do Estado da Arte de Sistemas de *Business Intelligence* e de Sistemas de Recomendação	Sim
2 - Desenvolvimento de um Sistema de *Data Warehousing*	Sim
3 - Desenvolvimento de um Sistema de Recomendações	Sim
4 - Desenvolvimento de uma componente de *Front-End* para Visualização e Exploração de Dados e KPI's	Sim

De acordo com o objetivo 2, referenciado na tabela 9, foi efetuada uma análise detalhada às fontes de dados que são provenientes dos sistemas de informação da organização. A estrutura das fontes

de dados apresenta-se bastante complexa, por isso, foram tomadas medidas para analisar e controlar a sua complexidade, como por exemplo a criação de *views* para facilitar a extração dos dados no processo ETL do sistema de *Data Warehousing*.

O modelo de dados do *Data Warehouse* desenvolvido baseia-se numa constelação constituído por 6 tabelas de facto e 9 dimensões, e suporta algumas das principais medidas de negócio da organização sendo que as tabelas de facto se encontram ligadas pelas diversas dimensões que possibilitam a análise dos dados de diferentes perspetivas.

O armazenamento de dados é realizado, diariamente, por um processo ETL implementado que permite extrair, transformar e carregar os dados das fontes de dados no *Data Warehouse* de uma forma automatizada.

Para a realização do objetivo 3 descrito na tabela 9, foi desenvolvido um sistema de recomendação, através dos dados do *Data Warehouse*, e foram aplicados alguns tipos de recomendações diferentes em que um deles levava a cabo modelos de *Machine Learning*. Como o sistema foi recentemente desenvolvido não houve espaço, ainda, para o desenvolvimento de um sistema de avaliação das recomendações.

Relativamente ao objetivo 4, referenciado na tabela 9, desenvolveram-se aplicações de *front-end* através da ferramenta *Power BI* que permitem o acesso a informação de forma intuitiva, rápida e segura. Os relatórios desenvolvidos nesta fase têm uma estrutura dinâmica permitindo aos utilizadores responderem a mais questões durante a exploração dos dados.

5.2. Trabalhos Futuros

Os projetos de *Bussiness Intelligence* carecem de uma grande manutenção e de um espaço de crescimento enorme. Há medida em que são respondidas questões através das informações disponíveis nestes sistemas, novas questões e ideias surgem dentro de uma organização. Este sistema, é focado na análise de vendas e análise do stock da empresa. Existem mais setores dentro da organização que precisam de ser abordados para facilitar a disponibilização de informação no seu contexto geral. Antes deste projeto crescer para outros departamentos dentro da empresa existem várias coisas a explorar no contexto das vendas e do stock. Assim, em plano futuro compromete-se o desenvolvimento de um relatório/ *dashboard* que permite a análise dos principais KPI's focados somente nas diferentes estações de vendas, dando resposta de forma rápida e eficaz a questões como "Que artigo foi mais vendido na estação de vendas atual?". Sendo este um relatório focado para acompanhar os artigos mais vendidos

durante a estação atual permitindo aos gestores acompanhar a tendência dos produtos de forma mais fácil.

Outra espectativa futura recai sobre o crescimento do *Data Warehouse*. Este deverá ser capaz de armazenar informações que permitem analisar a qualidade dos artigos, fornecendo informações dos defeitos e o impacto que estes têm nas vendas/devoluções.

A implementação de *data marts* para o departamento financeiro da empresa também está em plano futuro, devido à necessidade de se obter acesso às informações de uma forma mais rápida e detalhada sobre o investimento e resultados financeiros de cada loja e da empresa num contexto geral, trazendo, assim, o desenvolvimento de mais KPI's.

Devido ao facto do sistema de recomendação ter sido desenvolvido recentemente, e havendo ainda poucas recomendações utilizadas, fica previsto o desenvolvimento de um sistema que possibilitará a avaliação dessas recomendações no contexto real para ser possível compreender de que forma é que se poderá melhorar os algoritmos de recomendação.

Então, resumidamente, os pontos de desenvolvimento futuros a ter em consideração serão:

- Desenvolvimento de um relatório/*dashboard* focado no acompanhamento das estações de vendas;
- Estender o modelo do *Data Warehouse* para abranger os processos de qualidade dos artigos;
- Implementação de novos *Data Marts* para abranger os processos do departamento financeiro, definindo novos KPI's que suportem a gestão estratégica da organização;
- Desenvolvimento de um sistema de avaliação das recomendações e melhoria dos algoritmos de recomendação.

5.3. Contribuições

O projeto desenvolvido introduziu na organização novos conceitos de *Business Intelligence* para suportar a gestão estratégica tornando-se numa mais valia para a organização. Apenas com acesso a informação correta é possível tomar boas decisões para facilitar a gestão organizacional. O valor do *Business Intelligence* é predominante e proporciona novos recursos na área dos sistemas de informação capazes de suportar as tomadas de decisão.

O projeto desenvolvido revelou a facilidade de circulação e extração de informação útil através da implementação de sistemas de BI. É importante referir que este projeto é o primeiro do autor na área, sendo este da sua inteira responsabilidade. Revelou-se um trabalho importante para o desenvolvimento

das suas capacidades ao lidar com os requisitos e satisfação dos clientes, assim como no enriquecimento dos seus conhecimentos nas áreas de *Business Intelligence* e *Recomendations Systems*.

Bibliografia

Almeida, M. (2007). Gestão do Conhecimento. Universidade Federal de Santa Catarina

Anacleto R., Luz N., Almeida A., Figueiredo L., Novais P., *Shopping Center Tracking and Recommendation Systems, in Soft Computing Models in Industrial and Environmental Applications, 6th International*. Conference SOCO 2011, Corchado E.; Snasel V., Sedano J., Hassanien A.E.; Calvo J.L., Slezak D. (Eds.), Springer - Series Advances in Intelligent and Soft Computing, vol. 87, ISBN 978-3-642-19643-0, pp 299-308, 2011. http://dx.doi.org/10.1007/978-3-642-19644-7_32

Analide C., Novais P., Machado J., Neves J. *Quality of Knowledge in Virtual Entities, Encyclopedia of Communities of Practice in Information and Knowledge Management*. Elayne Coakes and Steve Clarke (Eds), Idea Group Reference, ISBN 1-59140-556-4, pp 436-442, 2006.
http://dx.doi.org/10.4018/978-1-59140-556-6.ch073

Aryachandra, T. & Watson, H. J. (2011). *Which Data Warehouse Architectures Most Sucessful? Business Intelligence Journal*. Vol 11, No. 1, 4-6

Belo, O. & Oliveira, B. (2012). Identificação de Hierarquias Incompletas em Estruturas Multidimensionais de Dados. (Id4103@alunos.uminho.pt) (obelo@di.uminho.pt) Centro de I&D ALGORITMI Universidade do Minho 4710-057 Braga PORTUGAL

Bolt, A. (2015). "Multidimensional Process Mining Using Process Cubes. Department of Mathematics and Computer Science". Eindhoven University of Technology, Eindhoven.

Burke, R. (2002). *Hybrid Recommender Systems: Survey and Experiments*. User Modeling and User-Adapted Interaction 12, no. 4: 331–370.

Caldeira, J. (2020). 100 Indicadores de Gestão: *Key Performance Indicators*. Conjuntura Atual Editora, S.A. ISBN 978-989-694-033-1.

Carneiro D., Pimenta A., Gonçalves S., Neves J., Novais P. *Monitoring and improving performance in human–computer interaction*. Concurrency and Computation: Practice and Experience, ISSN: 1532-0634, Vol. 28, n°4, pp 1291-1309, 2016. http://dx.doi.org/10.1002/cpe.3635

Costa, S. (2012). *Sistema de Business Intelligence como suporte a Gestão Estratégica*. Universidade do Minho.

Dayal, U., & Chaudhuri, S. (1997). *An overview of Data Warehousing and OLAP technology*. ACM SIGMOD Record, Volume 26, 65-74.

Donati, A. (2018). Conceção e Desenvolvimento de um Sistema de Recomendação para o Varejo Físico. Universidade Federal de Santa Catarina.

Ferreira, M. (2012). Classificação Hierárquica da Atividade Económica das Empresas a partir de Texto da Web. Universidade do Porto

Golfarelli, M. e Rizzi, S. (1998). *A Methodological Framework for Data Warehouse Design*. Proc. of the 1st ACM Int. Workshop on Data Warehousing and OLAP, pp.3-9.

Golfarelli, M. e Rizzi, S. (2009). *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill. Available at: http://books.google.pt/books?id=R7qqNwAACAAJ

Herlocker, Jonathan, L., Joseph, A., Konstan, Loren, G., Terveen, and John T. Riedl. (2004). *Evaluating collaborative filtering recommender systems*. ACM Transactions on Information Systems 22(1):5–53.

Inmon, W.H. (2002). *Building the Data Warehouse*.

Inmon, W. H. (2005). *Building the Data Warehouse*. New York: Wiley.

Isinkaye, F. O., Folajimi, Y. O. and Ojokoh, B. A. (2015). *Recommendation systems: Principles, methods and evaluation*. Egyptian Informatics Journal. doi: 10.1016/j.eij.2015.06.005.

João Carneiro, Pedro Saraiva, Diogo Martinho, Goreti Marreiros, Paulo Novais. *Representing decision-makers using styles of behavior: An approach designed for group decision support systems, Cognitive Systems Research*. Volume 47, Pages 109-132, ISSN 1389-0417, 2020. https://doi.org/10.1016/j.cogsys.2017.09.002.

Juan M. Górriz, Javier Ramírez, Andrés Ortíz, Francisco J. Martínez-Murcia, Fermin Segovia, John Suckling, Matthew Leming, Yu-Dong Zhang, Jose Ramón Álvarez-Sánchez, Guido Bologna, Paula Bonomini, Fernando E. Casado, David Charte, Francisco Charte, Ricardo Contreras, Alfredo Cuesta-Infante, Richard J. Duro, Antonio Fernández-Caballero, Eduardo Fernández-Jover, Pedro Gómez-Vilda, Manuel Graña, Francisco Herrera, Roberto Iglesias, Anna Lekova, Javier de Lope, Ezequiel López-Rubio, Rafael Martínez-Tomás, Miguel A. Molina-Cabello, Antonio S. Montemayor, Paulo Novais, Daniel Palacios-Alonso, Juan J. Pantrigo, Bryson R. Payne, Félix de la Paz López, María Angélica Pinninghoff, Mariano Rincón, José Santos, Karl Thurnhofer-Hemsi, Athanasios Tsanas, Ramiro Varela, Jose M. Ferrández. *Artificial intelligence within the interplay between natural and artificial computation: Advances in data science, trends and applications*. Neurocomputing, Volume 410, Pages 237-270, ISSN 0925-2312, 2020. https://doi.org/10.1016/j.neucom.2020.05.078.

Kimball, R., & Caserta, J. (2004). *The Data Warehouse ETL Toolkit*. Wiley Publishing, Inc.

Kimball, R., & Ross, M. (2003). *The DataWarehouse Toolkit: The Definitive Guide to Dimensional Modeling*.

Lane, P. (2002). *Oracle9i Data Warehousing Guide*. Oracle Corporation.

Lima L., Novais P., Costa R., Bulas Cruz J., Neves J. *Group Decision Making and Quality-of-Information in e-Health Systems*. Logic Journal of the IGPL, Oxford University Press, Volume 19 Issue 2, ISSN 1367-0751, pp 315-332, 2011. http://dx.doi.org/10.1093/jigpal/jzq029

Magalhães, A. (2017). *Business Intelligence no SQL Server*. FCA – Editora de Informática, Lda. ISBN 978-972-722-869-0.

Malinowski, E. e Zimányi, E., (2006). *Hierarchies in a multidimensional model: from conceptual modeling to logical representation*. Data Knowl. Eng., 59(2), pp.348-377.

Malinowski, E. e Zimányi, E., (2004). *OLAP Hierarchies: A Conceptual Perspective Advanced Information Systems Engineering*. In A. Persson & J. Stirna, eds. Springer Berlin / Heidelberg, pp. 19-35.

Neogrid (2019). *Data Quality: o que é e qual sua importância?* (https://blog.neogrid.com/data-quality-o-que-e-e-qual-sua-importancia/)

Nguyen, T. M., Tjoa, A. M., Nemec, J., & Windisch, M. (2006). *An approach towards an event-fed solution for slowly changing dimensions in data warehouses with a detailed case study*. Data & Knowledge Engineering 63 (2007) Elsevier, 26-43.

Nielsen, P., & Parui, U. (2011). *Microsoft SQL server 2008 bible (Vol. 607)*. John Wiley & Sons.

Rainardi, V. (2008). *Building a Data Warehouse: With Examples in SQL Server*. United States of America: Apress.

Resnick, P. and Varian, H. R. (1997). *Recommender systems.(Special Section: Recommender Systems)(Cover Story)*. Communications of the ACM.

Ricci, F., Lior, R. & Bracha, S. (2011). *Introduction to Recommender Systems Handbook*.

Rolim, V., Ferreira, R., Costa, E., Cavalcanti, A. & Dionísio, M. (2017). Um Estudo Sobre Sistemas de Recomendação de Recursos Educacionais. VI Congresso Brasileiro de Informática na Educação (WCBIE 2017), 1:724. Minas Gerais, Brasil. https://doi.org/10.5753/cbie.wcbie.2017.724.

Santos, A. J. (2008). *Gestão Estratégica: Conceitos, Modelos e Instrumentos*. Lisboa, Portugal: Escolar Editora.

Sá, J. V. (2009). *Metodologia de Sistemas de Data Warehouse*. Tese de Doutoramento. Guimarães: Universidade do Minho.

Santos, A. J. (2008). *Gestão Estratégica: Conceitos, Modelos e Instrumentos*. Lisboa, Portugal: Escolar Editora.

Santos, M. F., & Azevedo, C. (2005). Data Mining e Descoberta de Conhecimento em Base de Dados. Lisboa, Portugal: FCA.

Santos, M. Y., & Ramos, I. (2009). Business Intelligence – Tecnologias da Informação na Gestão de Conhecimento (2 ed.). Lisboa, Portugal: FCA.

Santos, M. Y., & Ramos, I. (2006). Como tornar o seu negócio realmente competitivo – Desafios tecnológicos e de gestão. CXO: Tecnologias de Informação para Executivos, 56-61.

Su, X. and Khoshgoftaar, T. M. (2009). *A Survey of Collaborative Filtering Techniques*. Advances in Artificial Intelligence. doi: 10.1155/2009/421425.

Takeuchi, N. (1997). Criação do Conhecimento na Empresa.

Turban, E. Sharda, R. Aronson, J. King, D. (2008). *Business Intelligence - a Managerial Approach*.

Vasconcelos, J. e Barão, A. (2017). Ciência de Dados nas Organizações: Aplicações em Python. FCA Editora de Informática, Lda. ISBN 978-972-722-885-0.

Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*. Politecnico di Milano, Italy: A John Wiley and Sons, Ltd., Publication.

Wang, Y., Deng, J., Gao, J. & Zhang, P. (2017). *A hybrid user similarity model for collaborative filtering*. Information Sciences 418–419: 102–18. https://doi.org/https://doi.org/10.1016/j.ins.2017.08.008

Anexos

Anexo I – A Descrição Sucinta das Ferramentas Utilizadas

Microsoft SQL *Server Data Tools*

O SQL *Server Data Tools* é uma ferramenta que contempla um ambiente de desenvolvimento onde se desenham pacotes para soluções de projetos. No caso de projetos de *Business Intelligenc,* o SQL *Server Data Tools* dispõe das seguintes ferramentas: *Integration Services, Analysis Services, Reporting Services.*

No caso de se desenvolver soluções de Extração, Transformação e Carregamento de dados, a ferramenta utilizada será o *Integration Services*. No caso de se desenvolver modelos multidimensionais como cubos OLAP, então utiliza-se a ferramenta *Analysis Services*, e finalmente, no caso de se desenvolverem pacotes de *reports*, utiliza-se uma ferramenta *front-end* de *Business Intelligence*, o *Reporting Services*.

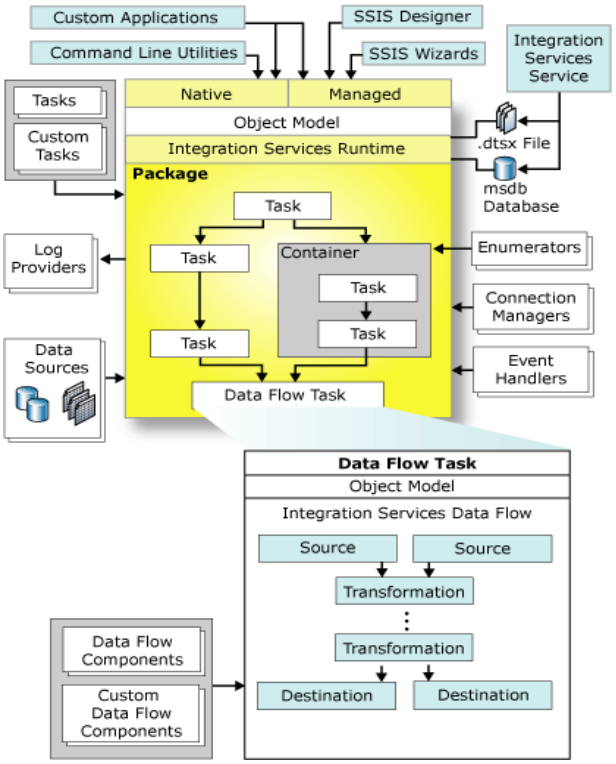
Como se pode ver o SQL *Server Data Tools* é uma ferramenta que permite realizar projetos completos de *Business Intelligence*.

Microsoft SQL *Server Integration Services*

O *Integration Services* é uma tecnologia para integração de dados quando o objetivo é construir fluxos de ETL de várias fontes para vários destinos. O SSIS é composto por 4 componentes principais:

- Serviço *Integration Services*: Monitoriza os pacotes a medida que estes são executados e gere o armazenamento;
- Modelo de Objetos: Interface de programação para aplicações personalizadas do *Integration Services*.
- *Runtime*: Administra as funcionalidades extra dos pacotes como os registos, interrupções, configurações, ligações e transações.
- Tarefas de Fluxo de Dados: Permite a movimentação de dados em memoria desde a sua origem até ao seu destino, é responsável pelas transformações dos dados e pela inserção dos dados nos destinos.

Na figura a baixo é apresentada a arquitetura do *Integration Services*.

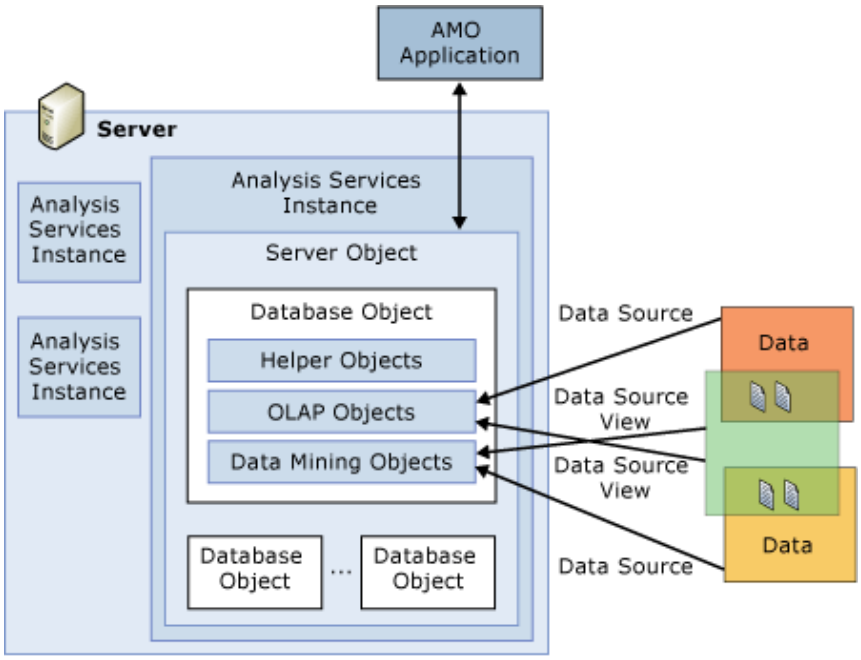


Arquitetura do *Integration Services* (adaptado de Microsoft)

Microsoft SQL *Server Analysis Services*

O *Analysis Services* é uma base de dados *OnLine Analytical Processing* (OLAP) otimizada para manipular grandes conjuntos de dados de forma simples e seletiva. O *Analysis Services* oferece 3 tipos de modelação: a modelação multidimensional, a modelação tabular que são modelos de tabelas, e modelos *PowerPivot* que são modelos especiais tabulares.

Na figura a baixo é apresentada a arquitetura do *Analysis Services*.

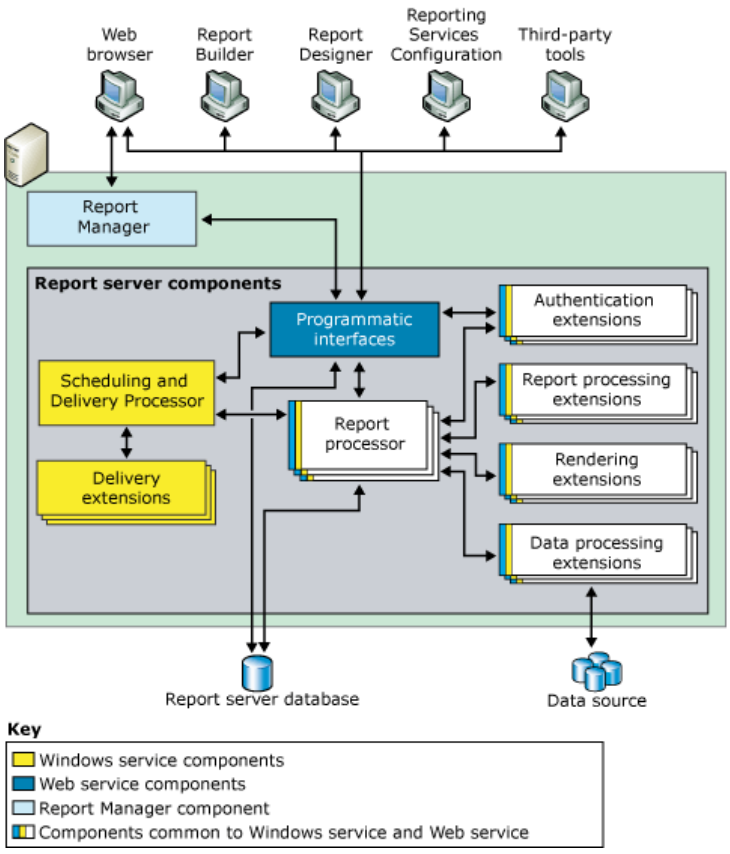


Arquitetura do *Analysis Services* (adaptada de Microsoft)

Microsoft SQL *Server Reporting Services*

O *Reporting Services* é uma tecnologia que permite desenvolver e administrar relatórios que podem ser disponibilizados na web.

Na figura a baixo é apresentada a arquitetura do *Reporting Services*.



Arquitetura do *Reporting Services* (adaptado de Microsoft)

Microsoft *Power BI*

O *Power BI* é uma tecnologia de análise de negócios da Microsoft para criação de relatórios e *dashboards* e fornece serviços baseados na *cloud*. O *Power BI Desktop* é a aplicação de desenvolvimento local e contém um ambiente integrado para transformação de dados para serem usados nos relatórios desenvolvidos. Esta plataforma não obriga a utilização de instâncias *Analysis Services* pois permite ligações locais a instâncias SQL *Server*.

A tecnologia *Power BI* consiste em 3 serviços, o *Power BI service* que é um serviço baseado em nuvem que armazena dados, *dashboards* e relatórios na plataforma *Power BI* Web, permitindo que estes sejam compartilhados por utilizadores dentro de uma organização. O *Power BI Desktop* que é apenas funcional no sistema operativo *Windows* e que permite tratamento de dados e construção de relatórios locais para depois serem publicados no *Power BI Service*. Também é possível o desenvolvimento de uma versão *mobile* dos relatórios, estes podem ser desenvolvidos através do *Power BI Desktop* ou do *Power BI Services* e acedidos pelos utilizadores pela aplicação *Power BI Mobile*. Esta ferramenta é extraordinariamente vantajosa para auxiliar as decisões de negócio.

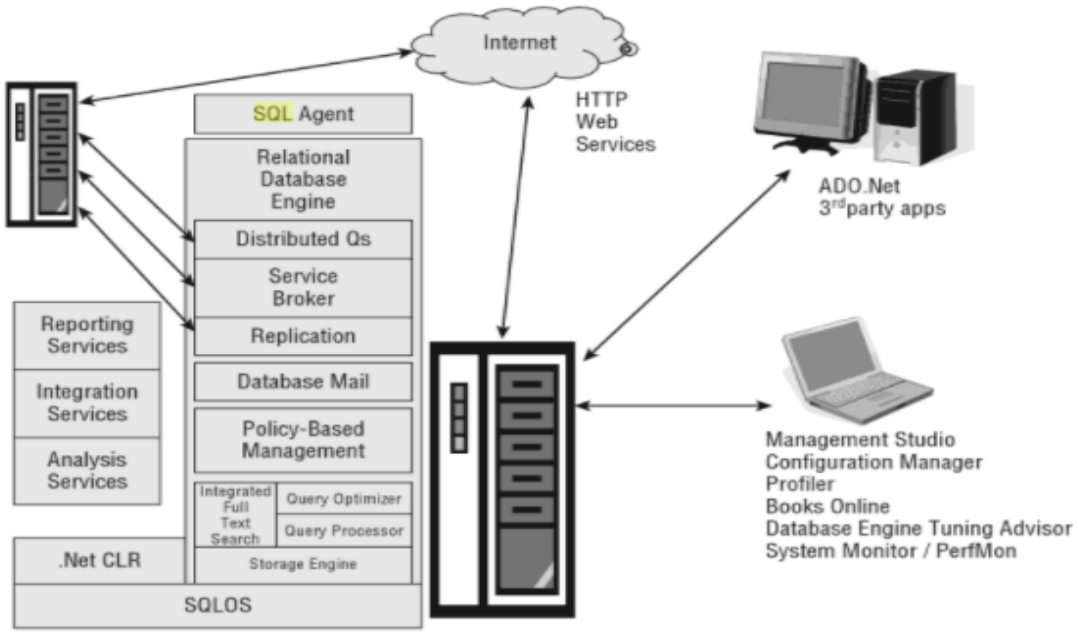
Linguagem *Python*

A linguagem de programação *Python* é de alto nível, uma linguagem de *scripting*, imperativa, funcional e orientada a objetos. É a principal linguagem utilizada na área de ciência de dados e no desenvolvimento de projetos de Engenharia de *Machine Learning*.

Microsoft SQL *Server*

O Microsoft SQL *Server* é um Sistema de Gestão de Bases de Dados relacionais cliente-servidor, onde a sua linguagem é T-SQL.

A arquitetura do SQL *Server* é apresentada na Figura a baixo.



Arquitetura do SQL Server (Adaptado de Nielsen, 2011)

Anexo II – Tabelas de caracterização da dimensão DIM_ARTIGO e subdimensões S_DIM_ARTIGO, S_DIM_PELE, S_DIM_COR, S_DIM_GENERO, S_DIM_MARCA, S_DIM_EPOCA, S_DIM_TIPO

DIM_ARTIGO

Caraterização da Dimensão							
Identificação	DIM_ARTIGO						
Descrição	Caraterização das Informações dos artigos únicos						
Tipo	Com Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para identificação do artigo no DataWarehouse	Sim	Sim	Int	Não	5
2	Artigo_ID	Identificador do artigo na origem de dados	Não	Sim	Nvarchar(20)	Não	0397741202511
3	Artigo_Marca	Id do Modelo do Artigo com a seu id de marca correspondente	Não	Não	Nvarchar(20)	Sim	442.6434.tec.prt. 0
Índices (N°)	Identificação	índice	Caraterística				
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente				
2	Artigo_Marca	Estrangeira	Referência com o atributo Artigo_Marca da tabela S_DIM_ARTIGO				
Hierarquias (N°)	Esquema						
1	Artigo_ID -> Artigo_Marca						

Perfil de Utilização	Gestores da Instituição

S_DIM_ARTIGO

Caraterização da Dimensão	
Identificação	S_DIM_ARTIGO
Descrição	Caraterização das Informações dos modelos de artigos
Tipo	Com Variação

Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para identificação do modelo de artigo no DataWarehouse	Sim	Sim	Int	Não	2653
2	Artigo_Marca	Id do Modelo do Artigo com a seu id de marca correspondente	Não	Sim	Nvarchar(50)	Não	050.4871.cam.prt. 0
3	Artigo	Id do Modelo do Artigo	Não	Não	Nvarchar(18)	Sim	050.4871.cam.prt.
4	Artigo_Designaçao	Descrição do Artigo	Não	Não	Nvarchar(30)	Sim	Botim Senhora
5	Tipo_ID	Identificador do Tipo de Artigo	Sim	Não	Int	Sim	1
6	Genero_ID	Identificador do Género de Artigo	Sim	Não	Int	Sim	2
7	Epoca_ID	Identificador da época do artigo	Sim	Não	Int	Sim	2
8	Ano_Artigo	Ano do artigo	Não	Não	Int	Sim	2007
9	Marca_ID	Identificador da Marca do Artigo	Sim	Não	Int	Sim	0

10	Referencia_Origem	Referência do Artigo	Não	Não	Nvarchar(15)	Sim	4871
11	Fornecedor_ID	Identificador do Fornecedor do Artigo	Não	Não	Int	Sim	0
12	Outlet	Atributo que indica se o artigo é de outlet ou não	Não	Não	Int	Sim	1
13	Cor_ID	Identificador da Cor do Artigo	Sim	Não	Int	Sim	0
14	Pele_ID	Identificador da Pele do Artigo	Sim	Não	Int	Sim	0
15	Classificacao_Tipo	Classificação do artigo	Não	Não	Nvarchar(20)	Sim	Clássico
16	Classificacao_Mercado	Classificação do Mercado do Artigo	Não	Não	Nvarchar(20)	Sim	Senhora
17	Classificacao_Categoria	Classificação da Categoria do Artigo	Não	Não	Nvarchar(20)	Sim	Botim
18	Classificacao_SubCategoria	Classificação da SubCategoria do Artigo	Não	Não	Nvarchar(20)	Sim	gola/dobrada
19	Classificacao_Segmento	Segmento do Artigo	Não	Não	Nvarchar(20)	Sim	Fechado
20	Classificacao_Brand	Brand do Artigo	Não	Não	Nvarchar(20)	Sim	Época
21	Classificacao_Stock	Classificação de stock do artigo	Não	Não	Nvarchar(20)	Sim	bicudo
22	Classificacao_Mkt	Classificação de Marketing do Artigo	Não	Não	Nvarchar(20)	Sim	S.Médio
23	Classificacao_1	Classificação extra do Artigo	Não	Não	Nvarchar(20)	Sim	

24	Classificacao_2	Classificação extra do Artigo	Não	Não	Nvarchar(20)	Sim	
25	Classificacao_3	Classificação extra do Artigo	Não	Não	Nvarchar(20)	Sim	
26	Artigo_Descricao_B2B	Descrição do Artigo na base de dados do B2B	Não	Não	Nvarchar(35)	Sim	NULL
27	Linha_B2B	Linha do Artigo na base de dados do B2B	Não	Não	Nvarchar(50)	Sim	NULL
28	Modelo_B2B	Modelo do artigo na base de dados B2B	Não	Não	Nvarchar(50)	Sim	NULL

Indices (N°)	Identificação	índice	Caraterística
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente
2	Tipo_ID	Estrangeira	Referência com o atributo Tipo_ID da tabela S_DIM_TIPO
3	Genero_ID	Estrangeira	Referência com o atributo Genero_ID da tabela S_DIM_GENERO
4	Epoca_ID	Estrangeira	Referência com o atributo Epoca_ID da tabela S_DIM_EPOCA
5	Cor_ID	Estrangeira	Referência com o atributo Cor_ID da tabela S_DIM_COR
6	Pele_ID	Estrangeira	Referência com o atributo Pele_ID da tabela S_DIM_PELE
7	Marca_ID	Estrangeira	Referência com o atributo Marca_ID da tabela S_DIM_MARCA

Hierarquias (N°)	Esquema
1	Artigo_ID -> Tipo_ID
2	Artigo_ID -> Genero_ID
3	Artigo_ID -> Epoca_ID
4	Artigo_ID -> Cor_ID
5	Artigo_ID -> Pele_ID

6	Artigo_ID -> Marca_ID
7	Artigo_ID -> Ano_Artigo
8	Artigo_ID -> Modelo_B2B -> Linha_B2B
Perfil de Utilização	Gestores da Instituição

S_DIM_PELE

Caraterização da Dimensão							
Identificação	S_DIM_PELE						
Descrição	Caraterização das Informações das peles dos artigos						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Pele_ID	Código interno para identificação da Pele tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	128
2	Pele_Sigla	Sigla da Pele	Não	Não	Nvarchar(20)	Não	vic
3	Pele_Designacao	Designação da Pele	Não	Não	Nvarchar(10)	Não	Victoria
Índices (N°)	Identificação	índice	Caraterística				
1	Pele_ID	Primária	Único, ordenado fisicamente de forma crescente				
Hierarquias (N°)	Esquema						

Perfil de Utilização	Gestores da Instituição

S_DIM_COR

Caraterização da Dimensão							
Identificação	S_DIM_COR						
Descrição	Caraterização das Informações das cores dos artigos						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Cor_ID	Código interno para identificação da Cor tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	1
2	Cor_Sigla	Sigla da Cor	Não	Não	Nvarchar(3)	Não	prt
3	Cor_Designacao	Designação da Cor	Não	Não	Nvarchar(10)	Não	Preto
Índices (N°)	Identificação	índice	Caraterística				
1	Cor_ID	Primária	Único, ordenado fisicamente de forma crescente				
Hierarquias (N°)	Esquema						
Perfil de Utilização	Gestores da Instituição						

S_DIM_GENERO

Caraterização da Dimensão							
Identificação	S_DIM_GENERO						
Descrição	Caraterização das Informações do género dos artigos						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Genero_ID	Código interno para identificação da Género tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	1
2	Genero_Designacao	Designação do Género	Não	Não	Nvarchar(20)	Não	Homem
Índices (N°)	Identificação	índice	Caraterística				
1	Genero_ID	Primário	Único, ordenado fisicamente de forma crescente				
Hierarquias (N°)	Esquema						
Perfil de Utilização	Gestores da Instituição						

S_DIM_MARCA

Caraterização da Dimensão	
Identificação	S_DIM_MARCA
Descrição	Caraterização das Informações do género dos artigos

Tipo	Sem Variação

Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Marca_ID	Código interno para identificação da Marca tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	2
2	Marca_Nome	Designação da Marca	Não	Não	Nvarchar(20)	Não	FLY London
3	Marca_Abreviatura	Abreviatura do Nome	Não	Não	Nvarchar(20)	Não	fly
4	Marca_Margem	Margem da Marca	Não	Não	Numeric(10,2)	Não	100.00
5	Marca_Iva	Iva correspondente	Não	Não	Numeric(10,2)	Não	20.00

Índices (Nº)	Identificação	índice	Caraterística
1	Marca_ID	Primário	Único, ordenado fisicamente de forma crescente

Hierarquias (Nº)	Esquema

Perfil de Utilização	Gestores da Instituição

S_DIM_EPOCA

Caraterização da Dimensão	
Identificação	S_DIM_EPOCA
Descrição	Caraterização das Informações da época dos artigos

Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Epoca_ID	Código interno para identificação da Época tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	1
2	Epoca_Designacao	Designação da época	Não	Não	Nvarchar(20)	Não	Primavera/ Verão
Índices (N°)	Identificação	índice	Caraterística				
1	Epoca_ID	Primário	Único, ordenado fisicamente de forma crescente				
Hierarquias (N°)	Esquema						
Perfil de Utilização	Gestores da Instituição						

S_DIM_TIPO

Caraterização da Dimensão							
Identificação	S_DIM_TIPO						
Descrição	Caraterização das Informações da época dos artigos						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Tipo_ID	Código interno para identificação do	Sim	Sim	Int	Não	1

		Tipo do artigo tanto no DataWarehouse como na Origem de dados					
2	Tipo_Designacao	Designação do tipo	Não	Não	Nvarchar(20)	Não	Primavera/ Verão
Índices (Nº)	Identificação	índice	Caraterística				
1	Tipo_ID	Primário	Único, ordenado fisicamente de forma crescente				
Hierarquias (Nº)	Esquema						
Perfil de Utilização	Gestores da Instituição						

Anexo III – Tabelas de caracterização da dimensão DIM_LOJA e das subdimensões S_DIM_ZONA, S_DIM_SITUACAO, S_DIM_CARATERISTICA

DIM_LOJA

Caraterização da Dimensão							
Identificação	DIM_LOJA						
Descrição	Caraterização das Informações das lojas						
Tipo	Com Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para identificação da Loja no DataWarehouse	Sim	Sim	Int	Não	99
2	Loja_ID	Sigla da Loja	Não	Sim	Nvarchar(3)	Não	gli
3	Nome	Nome da Loja, ou localidade da Loja	Não	Não	Nvarchar(30)	Sim	GLICINIAS
4	Zona_ID	Id da Zona da Loja	Sim	Não	Int	Sim	0
5	Situacao_ID	Id da Situação da Loja	Sim	Não	Int	Sim	1
6	Caraterisitca_ID	Id da Caraterística da Loja	Sim	Não	Int	Sim	0
7	Franchisado	Se a Loja é Franchisada ou Não	Não	Não	Int	Sim	NULL
8	Inicio	Data da Abertura da Loja	Não	Não	Date	Não	NULL
9	Fim	Data do Fecho da Loja	Não	Não	Date	Não	NULL
Índices (Nº)	Identificação	índice	Caraterística				

1	IDSK	Primário	Único, ordenado fisicamente de forma crescente
2	Zona_ID	Estrangeira	Referência com o atributo Zona_ID da tabela S_DIM_ZONA
3	Situacao_ID	Estrangeira	Referência com o atributo Situacao_ID da tabela S_DIM_SITUACAO
Hierarquias (N°)	Esquema		
1	Nome -> Zona_ID		
2	Nome -> Situacao_ID		
3	Nome -> Carateristica_ID		
4	Nome -> Franchisado		
Perfil de Utilização	Gestores da Instituição		

S_DIM_ZONA

Caraterização da Dimensão							
Identificação	S_DIM_ZONA						
Descrição	Caraterização das Informações da zona das lojas						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Zona_ID	Código interno para identificação da Zona tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	1
2	Zona_Designacao	Designação da zona	Não	Não	Nvarchar(20)	Não	NORTE
Índices (N°)	Identificação	índice	Caraterística				
1	Zona_ID	Primário	Único, ordenado fisicamente de forma crescente				

Hierarquias (N°)	Esquema
Perfil de Utilização	Gestores da Instituição

S_DIM_SITUACAO

Caraterização da Dimensão							
Identificação	S_DIM_SITUACAO						
Descrição	Caraterização das Informações da zona das lojas						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Situacao_ID	Código interno para identificação da Situação tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	1
2	Situacao_Nome	Designação da situação	Não	Não	Nvarchar(20)	Não	ABERTO
Índices (N°)	Identificação	índice	Caraterística				
1	Situacao_ID	Primário	Único, ordenado fisicamente de forma crescente				
Hierarquias (N°)	Esquema						
Perfil de Utilização	Gestores da Instituição						

S_DIM_CARATERISTICA

Caraterização da Dimensão							
Identificação	S_DIM_CARATERISTICA						
Descrição	Caraterização das Informações da zona das lojas						
Tipo	Sem Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Carateristica_ID	Código interno para identificação da Caraterística tanto no DataWarehouse como na Origem de dados	Sim	Sim	Int	Não	1
2	Carateristica_Descricao	Designação da caraterística da loja	Não	Não	Nvarchar(20)	Não	LOJA FOREVA
Índices (Nº)	Identificação	índice	Caraterística				
1	Carateristica_ID	Primário	Único, ordenado fisicamente de forma crescente				
Hierarquias (Nº)	Esquema						
Perfil de Utilização	Gestores da Instituição						

Anexo IV – Tabela de caraterização da DIM_TAMANHO

DIM_TAMANHO

Caraterização da Dimensão							
Identificação	DIM_TAMANHO						
Descrição	Caraterização das Informações dos tamanhos disponíveis						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para a identificação única do tamanho.	Sim	Sim	Int	Não	11
2	Tamanho	Tamanhos existentes	Não	Sim	Int	Não	10
Índices (N°)	Identificação	índice	Caraterística				
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente				
Hierarquias (N°)	Esquema						
Perfil de Utilização	Gestores da Instituição						

Anexo V – Tabela de caraterização da DIM_CLIENTE

DIM_CLIENTE

Caraterização da Dimensão							
Identificação	DIM_CLIENTE						
Descrição	Caraterização das Informações dos Clientes						
Tipo	Com Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para a identificação única do Cliente.	Sim	Sim	Int	Não	
2	Cliente	Código do Cliente	Não	Sim	Nvarchar(30)	Não	
3	Cliente_cp		Não	Não	Int	Sim	
4	Cliente_Estado_Civil	Estado Civil do cliente	Não	Não	Nvarchar(200)	Sim	
5	Cliente_Data_Nascimento	Data de nascimento do Cliente	Não	Não	Nvarchar(200)	Sim	
6	Cliente_Telefone	Telefone do Cliente	Não	Não	Nvarchar(200)	Sim	
7	Cliente_Telemovel	Telemovel do Cliente	Não	Não	Nvarchar(200)	Sim	
8	Cliente_Email	Email do Cliente	Não	Não	Nvarchar(200)	Sim	
9	Cliente_Calcado	Calçado do Cliente	Não	Não	Nvarchar(200)	Sim	
10	Cliente_Filhos	Número de Filhos do Cliente	Não	Não	Nvarchar(200)	Sim	
11	Cliente_Quantidade_Filhos	Número de filhos do cliente	Não	Não	Nvarchar(200)	Sim	
12	Cliente_Autorizacao_Email	Autorização para enviar emails	Não	Não	Nvarchar(200)	Sim	

13	Cliente_Autorizacao_Postal	Autorização para enviar cartas	Não	Não	Nvarchar(200)	Sim	
14	Cliente_Autorizacao_SMS	Autorização para enviar sms	Não	Não	Nvarchar(200)	Sim	
15	Cliente_Genero	Género do Cliente	Não	Não	Nvarchar(200)	Sim	
16	Cliente_Nome	Nome do Cliente	Não	Não	Nvarchar(200)	Sim	
17	Cliente_Morada	Morada do Cliente	Não	Não	Nvarchar(200)	Sim	
18	Cliente_Local	Local de residência do cliente	Não	Não	Nvarchar(200)	Sim	
19	Cliente_CP4	Código Postal do Cliente	Não	Não	Nvarchar(200)	Sim	
20	Cliente_CP3	Código Postal do Cliente	Não	Não	Nvarchar(200)	Sim	
21	Cliente_Localidade_Postal	Localidade Postal do Cliente	Não	Não	Nvarchar(200)	Sim	
22	Cliente_Pais	País do Cliente	Não	Não	Nvarchar(200)	Sim	
23	Cliente_Data_Emissao	Data de emissão do Cartão Cliente	Não	Não	Datatime	Sim	
24	Cliente_Observacoes	Observações	Não	Não	Nvarchar(200)	Sim	
25	Clientes_Pontos_Total	Pontos no Cartão Cliente	Não	Não	Int	Sim	
26	Clientes_Pontos_Valor	Pontos que o Cliente pode utilizar	Não	Não	Int	Sim	
27	Cliente_Valor	Valor em descontos	Não	Não	Numeric(10,2)	Sim	

Índices (N°)	Identificação	índice	Caraterística
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente

Hierarquias (N°)	Esquema

Perfil de Utilização	Gestores da Instituição

Anexo VI – Tabela de caraterização da DIM_DOCUMENTO_NÚMERO

DIM_DOCUMENTO_NUMERO

Caraterização da Dimensão							
Identificação	DIM_DOCUMENTO_NUMERO						
Descrição	Caraterização das Informações dos documentos de transação						
Tipo	Com Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para a identificação única do documento.	Sim	Sim	Int	Não	12
2	DocumentoNumero_ID	Código de identificação única do documento, com informações do dia, loja e o tipo do documento	Não	Sim	Nvarchar(100)	Não	20175273col7
3	Documento_Ano	Ano do documento	Não	Não	Int	Sim	2017
4	Documento_Numero	Código do Documento	Não	Não	Int	Sim	5273
5	DocumentoSiteOrigem	Loja onde se deu a transação	Não	Não	Nvarchar(20)	Sim	col
6	DocumentoTipo	Tipo do documento	Não	Não	Int	Sim	7
7	ClienteNif	Nif do Cliente, caso o cliente tenha pedido fatura com contribuinte	Não	Não	Nvarchar(50)	Sim	null
Índices (Nº)	Identificação	índice	Caraterística				
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente				
Hierarquias (Nº)	Esquema						

Perfil de Utilização	Gestores da Instituição

Anexo VII – Tabela de caraterização da DIM_CALENDARIO, DIM_HORA e DIM_MINUTO

DIM_CALENDARIO

Caraterização da Dimensão	
Identificação	DIM_CALAENDARIO
Descrição	Caraterização das Informações do calendário
Tipo	Sem Variação

Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para a identificação única de cada data.	Sim	Sim	Int	Não	25680
2	Data	Data	Não	Sim	Date	Não	2020-04-22
3	Dia	Dia	Não	Não	Char(2)	Não	22
4	DiaDaSemana	Nome do dia da semana	Não	Não	Varchar(10)	Não	Quarta
5	Semana	Semana do Ano	Não	Não	Char(2)	Não	17
6	Mes	Mês	Não	Não	Char(2)	Não	04
7	NomeMes	Nome do Mês	Não	Não	Varchar(20)	Não	Abril
8	Trimestre	Trimestre	Não	Não	TinyInt	Não	2
9	NomeTrimestre	Nome do Trimestre	Não	Não	Varchar(20)	Não	SEGUNDO
10	Ano	Ano	Não	Não	Char(4)	Não	2020
11	EstacaoAno	Estacão do Ano	Não	Não	Varchar(20)	Não	Primavera
12	FimSemana	Informação se é fim de semana ou não	Não	Não	Char(3)	Não	Não
13	DataCompleta	Data completa no formaro ‘yyyymmdd’	Não	Não	Varchar(10)	Não	20200422

Índices (Nº)	Identificação	Índice	Caraterística
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente
Hierarquias (Nº)	Esquema		
1	Data -> Semana -> Mês -> Trimestre -> Ano		
Perfil de Utilização	Gestores da Instituição		

DIM_HORA

Caraterização da Dimensão							
Identificação	DIM_HORA						
Descrição	Caraterização das Informações da hora						
Tipo	Sem Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para a identificação única de cada hora.	Sim	Sim	Int	Não	16
2	Hora	Hora	Não	Sim	Int	Não	15
3	Parte_Do_Dia	Parte do dia	Não	Não	Varchar(10)	Não	Tarde
Índices (Nº)	Identificação	Índice	Caraterística				
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente				
Hierarquias (Nº)	Esquema						
1	Hora -> Parte_Do_Dia						

Perfil de Utilização	Gestores da Instituição

DIM_MINUTO

Caraterização da Dimensão							
Identificação	DIM_MINUTO						
Descrição	Caraterização das Informações dos minutos						
Tipo	Sem Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	IDSK	Código interno para a identificação única de cada minuto.	Sim	Sim	Int	Não	2
2	Minuto	Minuto	Não	Sim	Int	Não	1
Índices (Nº)	Identificação	Índice	Caraterística				
1	IDSK	Primária	Único, ordenado fisicamente de forma crescente				
Hierarquias (Nº)	Esquema						
1							
Perfil de Utilização	Gestores da Instituição						

Anexo VIII – tabelas de caraterização das Tabelas de Facto (TF_VENDAS, TF_DEVOLUÇÃO, TF_TRANSFERENCIAS, TF_CONFIRMACAO_TRANSFERENCIAS, TF_TRANSFERENCIA_PONTOS, TF_REGISTO_STOCK).

TF_VENDAS

Caraterização da Tabela de Facto							
Identificação	TF_VENDAS						
Descrição	Caraterização das Informações das Vendas						
Tipo	Sem Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	DocumentoNumero_ID	Código interno para a identificação única do documento.	Sim	Não	Int	Não	401643
2	Artigo_ID	Código interno para identificação do artigo no DataWarehouse	Sim	Não	Int	Não	2001252
3	Tamanho	Código interno para a identificação única do tamanho.	Sim	Não	Int	Não	38
4	Data	Código interno para a identificação única de cada data.	Sim	Não	Int	Não	24880
5	Hora	Código interno para a identificação única de cada hora.	Sim	Não	Int	Não	12
6	Minuto	Código interno para a identificação única de cada minuto.	Sim	Não	Int	Não	51
7	Loja_ID	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	161

8	Cliente_ID	Código interno para a identificação única do Cliente.	Sim	Não	Int	Não	9020
9	Quantidade_Venda	Informação da quantidade vendida	Não	Não	Numeric(10,2)	Não	1
10	Preco_Custo	Informação do custo da venda	Não	Não	Numeric(10,2)	Não	7.34
11	Preco_Iliquido	Informação do preço Ilíquido	Não	Não	Numeric(10,2)	Não	29.20
12	Preco_Venda	Informação do Preço da Venda	Não	Não	Numeric(10,2)	Não	14.95
13	Desconto	Informação do desconto	Não	Não	Numeric(10,2)	Não	50.00
14	Online	Informação se a venda é online	Não	Não	Int	Não	0
15	Lucro_Venda	Informação do lucro da venda	Não	Não	Numeric(10,2)	Não	7.60

Índices (N°)	Identificação	Índice	Caraterística
1	DocumentoNumero_ID	Constituinte de chave Primária Composta	
2	Artigo_ID	Constituinte de chave Primária Composta	
3	Tamanho	Constituinte de chave Primária Composta	
4	Data	Constituinte de chave Primária Composta	
5	Hora	Constituinte de chave Primária Composta	
6	Minuto	Constituinte de chave Primária Composta	
7	Loja_ID	Constituinte de chave Primária Composta	

8	Cliente_ID	Constituinte de chave Primária Composta	
9	DocumentoNumero_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_DOCUMENTO_NUMERO
10	Artigo_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_ARTIGO
11	Tamanho	Estrangeira	Referência com o atributo IDSK da tabela DIM_TAMANHO
12	Data	Estrangeira	Referência com o atributo IDSK da tabela DIM_CALENDARIO
13	Hora	Estrangeira	Referência com o atributo IDSK da tabela DIM_HORA
14	Minuto	Estrangeira	Referência com o atributo IDSK da tabela DIM_MINUTO
15	Loja_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
16	Cliente_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_CLIENTE
Perfil de Utilização	Gestores da Instituição		

TF_DEVOLUCOES

Caraterização da Tabela de Facto							
Identificação	TF_DEVOLUCOES						
Descrição	Caraterização das Informações das Devoluções						
Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	DocumentoNumero_ID	Código interno para a identificação única do documento.	Sim	Não	Int	Não	403228
2	Artigo_ID	Código interno para identificação do artigo no DataWarehouse	Sim	Não	Int	Não	2713064

3	Tamanho	Código interno para a identificação única do tamanho.	Sim	Não	Int	Não	36
4	Data	Código interno para a identificação única de cada data.	Sim	Não	Int	Não	25129
5	Hora	Código interno para a identificação única de cada hora.	Sim	Não	Int	Não	17
6	Minuto	Código interno para a identificação única de cada minuto.	Sim	Não	Int	Não	39
7	Loja_ID	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	87
8	Cliente_ID	Código interno para a identificação única do Cliente.	Sim	Não	Int	Não	9020
9	Quantidade_Venda	Quantidade que foi vendida e esta a ser devolvida	Não	Não	Numeric(10,2)	Não	1
10	Preco_Custo	Informação do custo da venda	Não	Não	Numeric(10,2)	Não	35.20
11	Preco_Iliquido	Informação do preço Ilíquido	Não	Não	Numeric(10,2)	Não	99.90
12	Preco_Venda	Informação do Preço da Venda	Não	Não	Numeric(10,2)	Não	99.90
13	Online	Informação se a venda foi online	Não	Não	Int	Não	1
14	Reposicao_Venda	Informação do prejuízo na devolução	Não	Não	Numeric(10,2)	Não	64.70
Índices (Nº)	Identificação	Índice	Caraterística				

1	DocumentoNumero_ID	Constituinte de chave Primária Composta	
2	Artigo_ID	Constituinte de chave Primária Composta	
3	Tamanho	Constituinte de chave Primária Composta	
4	Data	Constituinte de chave Primária Composta	
5	Hora	Constituinte de chave Primária Composta	
6	Minuto	Constituinte de chave Primária Composta	
7	Loja_ID	Constituinte de chave Primária Composta	
8	Cliente_ID	Constituinte de chave Primária Composta	
9	DocumentoNumero_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_DOCUMENTO_NUMERO
10	Artigo_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_ARTIGO
11	Tamanho	Estrangeira	Referência com o atributo IDSK da tabela DIM_TAMANHO
12	Data	Estrangeira	Referência com o atributo IDSK da tabela DIM_CALENDARIO
13	Hora	Estrangeira	Referência com o atributo IDSK da tabela DIM_HORA
14	Minuto	Estrangeira	Referência com o atributo IDSK da tabela DIM_MINUTO
15	Loja_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
16	Cliente_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_CLIENTE
Perfil de Utilização	Gestores da Instituição		

TF_TRANSFERENCIA

Caraterização da Tabela de Facto							
Identificação	TF_TRANSFERENCIAS						
Descrição	Caraterização das Informações dos pedidos de transferência						
Tipo	Sem Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Artigo_ID	Código interno para identificação do artigo no DataWarehouse	Sim	Não	Int	Não	7007
2	Tamanho	Código interno para a identificação única do tamanho.	Sim	Não	Int	Não	36
3	Data	Código interno para a identificação única de cada data.	Sim	Não	Int	Não	24992
4	Hora	Código interno para a identificação única de cada hora.	Sim	Não	Int	Não	10
5	Minuto	Código interno para a identificação única de cada minuto.	Sim	Não	Int	Não	30
6	Loja_Saida	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	131
7	Loja_Entrada	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	3
8	Quantidade	Quantidade no pedido de transferência	Não	Não	Int	Não	1
Índices (Nº)	Identificação	Índice	Caraterística				

1	Artigo_ID	Constituinte de chave Primária Composta	
2	Tamanho	Constituinte de chave Primária Composta	
3	Data	Constituinte de chave Primária Composta	
4	Hora	Constituinte de chave Primária Composta	
5	Minuto	Constituinte de chave Primária Composta	
6	Loja_Entrada	Constituinte de chave Primária Composta	
7	Loja_Saida	Constituinte de chave Primária Composta	
8	Artigo_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_ARTIGO
9	Tamanho	Estrangeira	Referência com o atributo IDSK da tabela DIM_TAMANHO
10	Data	Estrangeira	Referência com o atributo IDSK da tabela DIM_CALENDARIO
11	Hora	Estrangeira	Referência com o atributo IDSK da tabela DIM_HORA
12	Minuto	Estrangeira	Referência com o atributo IDSK da tabela DIM_MINUTO
13	Loja_Entrada	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
14	Loja_Saida	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
Perfil de Utilização	Gestores da Instituição		

TF_CONFIRMACAO_TRANSFERENCIA

Caraterização da Tabela de Facto	
Identificação	TF_CONFIRMACAO_TRANSFERENCIA
Descrição	Caraterização das Informações das confirmações de transferências

Tipo	Sem Variação						
Atributos (N°)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Artigo_ID	Código interno para identificação do artigo no DataWarehouse	Sim	Não	Int	Não	7017
2	Tamanho	Código interno para a identificação única do tamanho.	Sim	Não	Int	Não	37
3	Data	Código interno para a identificação única de cada data.	Sim	Não	Int	Não	24992
4	Hora	Código interno para a identificação única de cada hora.	Sim	Não	Int	Não	10
5	Minuto	Código interno para a identificação única de cada minuto.	Sim	Não	Int	Não	32
6	Loja_Saida	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	3
7	Loja_Entrada	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	131
8	Quantidade	Quantidade na confirmação	Não	Não	Int	Não	1
Índices (N°)	Identificação	Índice	Caraterística				
1	Artigo_ID	Constituinte de chave Primária Composta					
2	Tamanho	Constituinte de chave Primária Composta					

3	Data	Constituinte de chave Primária Composta	
4	Hora	Constituinte de chave Primária Composta	
5	Minuto	Constituinte de chave Primária Composta	
6	Loja_Saida	Constituinte de chave Primária Composta	
7	Loja_Entrada	Constituinte de chave Primária Composta	
8	Artigo_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_ARTIGO
9	Tamanho	Estrangeira	Referência com o atributo IDSK da tabela DIM_TAMANHO
10	Data	Estrangeira	Referência com o atributo IDSK da tabela DIM_CALENDARIO
11	Hora	Estrangeira	Referência com o atributo IDSK da tabela DIM_HORA
12	Minuto	Estrangeira	Referência com o atributo IDSK da tabela DIM_MINUTO
13	Loja_Entrada	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
14	Loja_Saida	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
Perfil de Utilização	Gestores da Instituição		

TF_TRANSFERENCIA_PONTOS

Caraterização da Tabela de Facto							
Identificação	TF_TRANSFERENCIA_PONTOS						
Descrição	Caraterização das Informações dos pontos ganhos ou perdidos por clientes						
Tipo	Sem Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo

1	Cliente_ID	Código interno para a identificação única do Cliente.	Sim	Não	Int	Não	76
2	Loja_ID	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	49
3	Data	Código interno para a identificação única de cada data.	Sim	Não	Int	Não	25593
4	Hora	Código interno para a identificação única de cada hora.	Sim	Não	Int	Não	23
5	Minuto	Código interno para a identificação única de cada minuto.	Sim	Não	Int	Não	42
6	Pontos_Trasnferidos	Quantidade de pontos transferidos	Não	Não	Int	Não	780

Índices (Nº)	Identificação	Índice	Caraterística
1	Cliente_ID	Constituinte de chave Primária Composta	
2	Loja_ID	Constituinte de chave Primária Composta	
3	Data	Constituinte de chave Primária Composta	
4	Hora	Constituinte de chave Primária Composta	
5	Minuto	Constituinte de chave Primária Composta	
6	Cliente_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_CLIENTE
7	Loja_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
8	Data	Estrangeira	Referência com o atributo IDSK da tabela DIM_CALENDARIO
9	Hora	Estrangeira	Referência com o atributo IDSK da tabela DIM_HORA
10	Minuto	Estrangeira	Referência com o atributo IDSK da tabela DIM_MINUTO

Perfil de Utilização	Gestores da Instituição

TF_REGISTO_STOCK

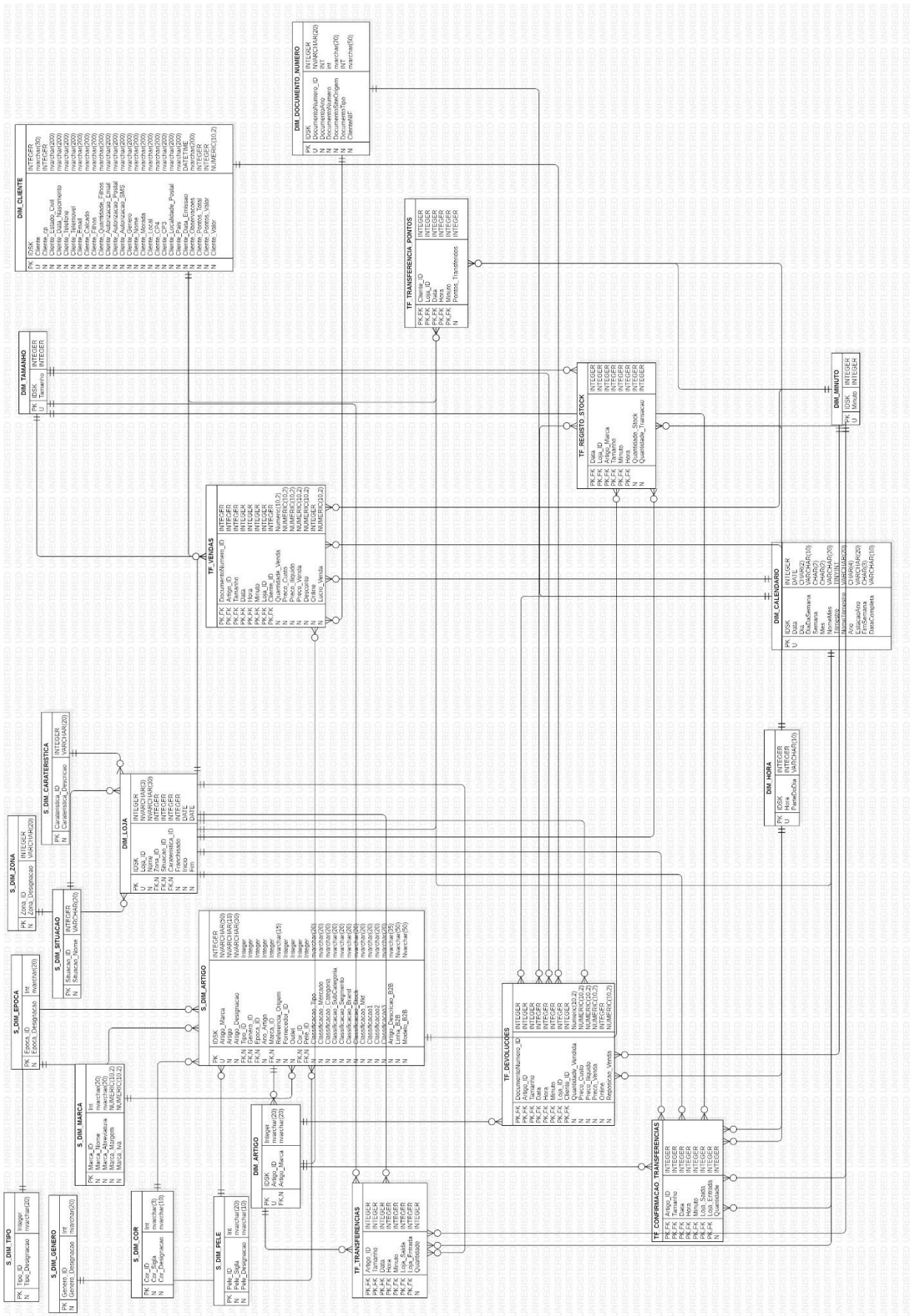
Caraterização da Tabela de Facto							
Identificação	TF_REGISTO_STOCK						
Descrição	Caraterização das Informações do stock diário						
Tipo	Sem Variação						
Atributos (Nº)	Identificação	Descrição	Chave	Único	Domínio	Variações	Exemplo
1	Data	Código interno para a identificação única de cada data.	Sim	Não	Int	Não	25789
2	Hora	Código interno para a identificação única de cada hora.	Sim	Não	Int	Não	19
3	Minuto	Código interno para a identificação única de cada minuto.	Sim	Não	Int	Não	51
4	Loja_ID	Código interno para identificação da Loja no DataWarehouse	Sim	Não	Int	Não	3
5	Artigo_Marca	Código interno para identificação do modelo de artigo no DataWarehouse	Sim	Não	Int	Não	209
6	Tamanho	Código interno para a identificação única do tamanho.	Sim	Não	Int	Não	36
7	Quantidade_Stock	Informação da quantidade de stock	Não	Não	Int	Não	10

8	Quantidade_Transacao	Informação da quantidade de stock em transporte para a loja	Não	Não	Int	Não	0

Índices (Nº)	Identificação	Índice	Caraterística
1	Data	Constituinte de chave Primária Composta	
2	Hora	Constituinte de chave Primária Composta	
3	Minuto	Constituinte de chave Primária Composta	
4	Loja_ID	Constituinte de chave Primária Composta	
5	Artigo_Marca	Constituinte de chave Primária Composta	
6	Tamnho	Constituinte de chave Primária Composta	
7	Data	Estrangeira	Referência com o atributo IDSK da tabela DIM_CALENDARIO
8	Hora	Estrangeira	Referência com o atributo IDSK da tabela DIM_HORA
9	Minuto	Estrangeira	Referência com o atributo IDSK da tabela DIM_MINUTO
10	Loja_ID	Estrangeira	Referência com o atributo IDSK da tabela DIM_LOJA
11	Artigo_Marca	Estrangeira	Referência com o atributo IDSK da tabela S_DIM_ARTIGO
12	Tamnho	Estrangeira	Referência com o atributo IDSK da tabela DIM_TAMANHO

Perfil de Utilização	Gestores da Instituição

Anexo IX – Modelo Dimensional, Conjunto de Flocos de Neve (Constelação)



Anexo X – Implementação Física do *Data Warehouse* (*Microsoft SSMS*)

```
CREATE DATABASE FOREVA_DW
GO

USE FOREVA_DW
GO

CREATE TABLE S_DIM_TIPO(
Tipo_ID INT PRIMARY KEY,
Tipo_Designacao NVARCHAR(20) DEFAULT NULL
)
GO

CREATE TABLE S_DIM_GENERO(
Genero_ID INT PRIMARY KEY,
Genero_Designacao NVARCHAR(20) DEFAULT NULL
)
GO

CREATE TABLE S_DIM_COR(
Cor_ID INT PRIMARY KEY,
Cor_Sigla NVARCHAR(3) DEFAULT NULL,
Cor_Designacao NVARCHAR(10) DEFAULT NULL
)
GO

CREATE TABLE S_DIM_PELE(
Pele_ID INT PRIMARY KEY,
Pele_Sigla NVARCHAR(20) DEFAULT NULL,
Pele_Designacao NVARCHAR(10) DEFAULT NULL
)
GO

CREATE TABLE S_DIM_EPOCA(
Epoca_ID INT PRIMARY KEY,
Epoca_Designacao NVARCHAR(20) DEFAULT NULL
)
```

```
GO

CREATE TABLE S_DIM_MARCA(
Marca_ID INT PRIMARY KEY,
Marca_Nome NVARCHAR(20) DEFAULT NULL,
Marca_Abreviatura NVARCHAR(20) DEFAULT NULL,
Marca_Margem NUMERIC(10,2) DEFAULT NULL,
Marca_Iva NUMERIC(10,2) DEFAULT NULL
)
GO

CREATE TABLE S_DIM_ARTIGO(
IDSK INT PRIMARY KEY IDENTITY,
Artigo_Marca NVARCHAR(50) unique NOT NULL,
Artigo NVARCHAR(18) DEFAULT NULL,
Artigo_Designacao NVARCHAR(30) DEFAULT NULL,
Tipo_ID INT REFERENCES S_DIM_TIPO(Tipo_ID),
Genero_ID INT REFERENCES S_DIM_GENERO(Genero_ID),
Epoca_ID INT REFERENCES S_DIM_EPOCA(Epoca_ID),
Ano_Artigo INT DEFAULT NULL,
Marca_ID INT REFERENCES S_DIM_MARCA(Marca_ID),
Referencia_Origem NVARCHAR(15) DEFAULT NULL,
Fornecedor_ID INT DEFAULT NULL,
Outlet INT DEFAULT NULL,
Cor_ID INT REFERENCES S_DIM_COR(Cor_ID),
Pele_ID INT REFERENCES S_DIM_PELE(Pele_ID),
Classificacao_Tipo NVARCHAR(20) DEFAULT NULL,
Classificacao_Mercado NVARCHAR(20) DEFAULT NULL,
Classificacao_Categoria NVARCHAR(20) DEFAULT NULL,
Classificacao_SubCategoria NVARCHAR(20) DEFAULT NULL,
Classificacao_Segmento NVARCHAR(20) DEFAULT NULL,
Classificacao_Brand NVARCHAR(20) DEFAULT NULL,
Classificacao_Stock NVARCHAR(20) DEFAULT NULL,
Classificacao_Mkt NVARCHAR(20) DEFAULT NULL,
Classificacao1 NVARCHAR(20) DEFAULT NULL,
Classificacao2 NVARCHAR(20) DEFAULT NULL,
Classificacao3 NVARCHAR(20) DEFAULT NULL,
Artigo_Descricao_B2B NVARCHAR(35) DEFAULT NULL,
Linha_B2B NVARCHAR(50) DEFAULT NULL,
```

```
Modelo_B2B NVARCHAR(50) DEFAULT NULL
)
GO

CREATE TABLE DIM_ARTIGO(
IDSK INT PRIMARY KEY IDENTITY,
Artigo_ID NVARCHAR(20) NOT NULL unique,
Artigo_Marca NVARCHAR(50) REFERENCES S_DIM_ARTIGO(Artigo_Marca)
)
GO

CREATE TABLE S_DIM_SITUACAO(
Situacao_ID INT PRIMARY KEY,
Situacao_Nome NVARCHAR(20) DEFAULT NULL
)
GO

CREATE TABLE S_DIM_ZONA(
Zona_ID INT PRIMARY KEY,
Zona_Designacao NVARCHAR(20) DEFAULT NULL
)
GO

CREATE TABLE S_DIM_CARATERISTICA(
Carateristica_ID INT PRIMARY KEY,
Carateristica_Descricao NVARCHAR(20) DEFAULT NULL
)
GO

CREATE TABLE DIM_LOJA(
IDSK INT PRIMARY KEY IDENTITY,
Loja_ID NVARCHAR(3) unique NOT NULL,
Nome NVARCHAR(30) DEFAULT NULL,
Zona_ID INT REFERENCES S_DIM_ZONA(Zona_ID),
Situacao_ID INT REFERENCES S_DIM_SITUACAO(Situacao_ID),
Carateristica_ID INT REFERENCES S_DIM_CARATERISTICA(Carateristica_ID),
Franchisado INT DEFAULT NULL,
Inicio DATE DEFAULT NULL,
Fim DATE DEFAULT NULL
```

```
)
GO

CREATE TABLE DIM_TAMANHO(
IDSK INT PRIMARY KEY IDENTITY,
Tamanho INT unique NOT NULL
)
GO

CREATE TABLE DIM_CLIENTE(
IDSK INT PRIMARY KEY IDENTITY,
Cliente NVARCHAR(30) unique NOT NULL,
Cliente_cp INT DEFAULT NULL,
Cliente_Estado_Civil NVARCHAR(200) DEFAULT NULL,
Cliente_Data_Nascimento NVARCHAR(200) DEFAULT NULL,
Cliente_Telefone NVARCHAR(200) DEFAULT NULL,
Cliente_Telemovel NVARCHAR(200) DEFAULT NULL,
Cliente_Email NVARCHAR(200) DEFAULT NULL,
Cliente_Calcado NVARCHAR(200) DEFAULT NULL,
Cliente_Filhos NVARCHAR(200) DEFAULT NULL,
Cliente_Quantidade_Filhos NVARCHAR(200) DEFAULT NULL,
Cliente_Autorizacao_Email NVARCHAR(200) DEFAULT NULL,
Cliente_Autrizacao_Postal NVARCHAR(200) DEFAULT NULL,
Cliente_Autorizacao_SMS NVARCHAR(200) DEFAULT NULL,
Cliente_Genero NVARCHAR(200) DEFAULT NULL,
Cliente_Nome NVARCHAR(200) DEFAULT NULL,
Cliente_Morada NVARCHAR(200) DEFAULT NULL,
Cliente_Local NVARCHAR(200) DEFAULT NULL,
Cliente_CP4 NVARCHAR(200) DEFAULT NULL,
Cliente_CP3 NVARCHAR(200) DEFAULT NULL,
Cliente_Localidade_Postal NVARCHAR(200) DEFAULT NULL,
Cliente_Pais NVARCHAR(200) DEFAULT NULL,
Cliente_Data_Emissao DATETIME,
Cliente_Observacoes NVARCHAR(200) DEFAULT NULL,
Cliente_Pontos_Total INT DEFAULT NULL,
Cliente_Pontos_Valor INT DEFAULT NULL,
Cliente_Valor NUMERIC(10,2) DEFAULT NULL
)
GO
```

```
CREATE TABLE DIM_DOCUMENTO_NUMERO(
IDSK INT PRIMARY KEY  IDENTITY,
DocumentoNumero_ID NVARCHAR(100) unique NOT NULL,
DocumentoAno INT DEFAULT NULL,
DocumentoNumero INT DEFAULT NULL,
DocumentoSiteOrigem NVARCHAR(20) DEFAULT NULL,
DocumentoTipo INT DEFAULT NULL,
ClienteNif NVARCHAR(50) DEFAULT NULL
)
GO

CREATE TABLE DIM_CALENDARIO(
IDSK INT PRIMARY KEY IDENTITY,
Data DATE unique NOT NULL,
Dia CHAR(2),
DiaDaSemana VARCHAR(10),
Semana Char(2),
Mes Char(2),
NomeMes Varchar(20),
Trimestre TINYINT,
NomeTrimestre Varchar(20),
Ano Char(4),
EstacaoAno VARCHAR(20),
FimSemana CHAR(3),
DataCompleta Varchar(10),
)
GO

CREATE TABLE DIM_HORA(
IDSK INT PRIMARY KEY IDENTITY,
Hora INT NOT NULL unique,
Parte_Do_Dia VARCHAR(10)
)
GO

CREATE TABLE DIM_MINUTO (
IDSK INT PRIMARY KEY IDENTITY,
Minuto INT unique NOT NULL)
```

```
GO

CREATE TABLE TF_VENDAS (DocumentoNumero_ID INT REFERENCES DIM_DOCUMENTO_NUMERO(IDSK),
Artigo_ID INT REFERENCES DIM_ARTIGO(IDSK),
Tamanho INT REFERENCES DIM_TAMANHO(IDSK),
Data INT REFERENCES DIM_CALENDARIO(IDSK),
Hora INT REFERENCES DIM_HORA(IDSK),
Minuto INT REFERENCES DIM_MINUTO(IDSK),
Loja_ID INT REFERENCES DIM_LOJA(IDSK),
Cliente_ID INT REFERENCES DIM_CLIENTE(IDSK),
Quantidade_Venda INT DEFAULT NULL,
Preco_Custo NUMERIC(10,2) DEFAULT NULL,
Preco_Iliquido NUMERIC(10,2) DEFAULT NULL,
Preco_Venda NUMERIC(10,2) DEFAULT NULL,
Desconto NUMERIC(10,2) DEFAULT NULL,
Online INT DEFAULT NULL,
Lucro_Venda NUMERIC(10,2) DEFAULT NULL,
 primary key (DocumentoNumero_ID, Artigo_ID,Tamanho,Data,Hora,Minuto,Loja_ID,Cliente_ID)
)
GO

CREATE TABLE TF_DEVOLUCOES (DocumentoNumero_ID INT REFERENCES DIM_DOCUMENTO_NUMERO(IDSK),
Artigo_ID INT REFERENCES DIM_ARTIGO(IDSK),
Tamanho INT REFERENCES DIM_TAMANHO(IDSK),
Data INT REFERENCES DIM_CALENDARIO(IDSK),
Hora INT REFERENCES DIM_HORA(IDSK),
Minuto INT REFERENCES DIM_MINUTO(IDSK),
Loja_ID INT REFERENCES DIM_LOJA(IDSK),
Cliente_ID INT REFERENCES DIM_CLIENTE(IDSK),
Quantidade_Venda INT DEFAULT NULL,
Preco_Custo NUMERIC(10,2) DEFAULT NULL,
Preco_Iliquido NUMERIC(10,2) DEFAULT NULL,
Preco_Venda NUMERIC(10,2) DEFAULT NULL,
Online INT DEFAULT NULL,
Reposicao_Venda NUMERIC(10,2) DEFAULT NULL,
 primary key (DocumentoNumero_ID, Artigo_ID,Tamanho,Data,Hora,Minuto,Loja_ID,Cliente_ID)
)
GO
```

```
CREATE TABLE TF_TRANSFERENCIAS (
Artigo_ID INT REFERENCES DIM_ARTIGO(IDSK),
Tamanho INT REFERENCES DIM_TAMANHO(IDSK),
Data INT REFERENCES DIM_CALENDARIO(IDSK),
Hora INT REFERENCES DIM_HORA(IDSK),
Minuto INT REFERENCES DIM_MINUTO(IDSK),
Loja_Saida INT REFERENCES DIM_LOJA(IDSK),
Loja_Entrada INT REFERENCES DIM_LOJA(IDSK),
Quantidade INT DEFAULT NULL
 primary key ( Artigo_ID,Tamanho,Data,Hora,Minuto,Loja_Saida,Loja_Entrada)
)
GO

CREATE TABLE TF_CONFIRMACAO_TRANSFERENCIAS (
Artigo_ID INT REFERENCES DIM_ARTIGO(IDSK),
Tamanho INT REFERENCES DIM_TAMANHO(IDSK),
Data INT REFERENCES DIM_CALENDARIO(IDSK),
Hora INT REFERENCES DIM_HORA(IDSK),
Minuto INT REFERENCES DIM_MINUTO(IDSK),
Loja_Saida INT REFERENCES DIM_LOJA(IDSK),
Loja_Entrada INT REFERENCES DIM_LOJA(IDSK),
Quantidade INT DEFAULT NULL
primary key ( Artigo_ID,Tamanho,Data,Hora,Minuto,Loja_Saida,Loja_Entrada)
)
GO

CREATE TABLE TF_TRANSFERENCIA_PONTOS (
Cliente_ID INT REFERENCES DIM_CLIENTE(IDSK),
Loja_ID INT REFERENCES DIM_LOJA(IDSK),
Data INT REFERENCES DIM_CALENDARIO(IDSK),
Hora INT REFERENCES DIM_HORA(IDSK),
Minuto INT REFERENCES DIM_MINUTO(IDSK),
Pontos_Transferidos INT DEFAULT NULL
primary key (Cliente_ID,Loja_ID,Data,Hora,Minuto)
)
GO

CREATE TABLE TF_REGISTO_STOCK (
Data INT REFERENCES DIM_CALENDARIO(IDSK),
```

```
Hora INT REFERENCES DIM_HORA(IDSK),
Minuto INT REFERENCES DIM_MINUTO(IDSK),
Loja_ID INT REFERENCES DIM_LOJA(IDSK),
Artigo_Marca INT REFERENCES S_DIM_ARTIGO(IDSK),
Tamanho INT REFERENCES DIM_TAMANHO(IDSK),
Quantidade_Stock INT DEFAULT NULL,
Quantidade_Transacao INT DEFAULT NULL
primary key (Data,Hora,Minuto,Loja_ID,Artigo_Marca,Tamanho)
)
GO
```

Anexo XI – *Store Procedure* de Carregamento da Tabela de facto Vendas

```
(Microsoft SSMS)
USE [FOREVA_DW]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROC [dbo].[CARREGA_VENDAS] AS

DECLARE @FINAL DATETIME
DECLARE @INICIAL DATETIME

SELECT @FINAL = MAX(DATA)
FROM FOREVA_DW.DBO.DIM_CALENDARIO T

SELECT @INICIAL = MAX(T.Data)
FROM FOREVA_DW.DBO.TF_VENDAS FT
JOIN FOREVA_DW.DBO.DIM_CALENDARIO T ON (FT.Data=T.IDSK)

IF @INICIAL IS NULL
BEGIN
SELECT @INICIAL = MIN(Data)
FROM FOREVA_DW.DBO.DIM_CALENDARIO T
END

INSERT INTO FOREVA_DW.DBO.TF_VENDAS(
DocumentoNumero_ID,
Artigo_ID,
Tamanho,
Data,
Hora,
Minuto,
Loja_ID,
Cliente_ID,
Quantidade_Venda,
```

```
Preco_Custo,
Preco_Iliquido,
Preco_Venda,
Desconto,
Online,
Lucro_Venda
)
Select
D.IDSK AS DocumentoNumero_ID,
A.IDSK AS Artigo_ID,
Ta.IDSK AS Tamanho,
T.IDSK AS Data,
H.IDSK AS Hora,
M.IDSK AS Minuto,
L.IDSK AS Loja_ID,
C.IDSK AS Cliente_ID,
V.Quantidade_Venda,
V.Preco_Custo,
V.Preco_Iliquido,
V.Preco_Venda,
V.Desconto,
V.Online,
V.Lucro_Venda

FROM FOREVA_STAGE.DBO.STF_VENDAS V

INNER JOIN DBO.DIM_DOCUMENTO_NUMERO AS D
ON V.DocumentoNumero_ID=D.DocumentoNumero_ID

INNER JOIN DBO.DIM_ARTIGO AS A
ON V.Artigo_ID=A.Artigo_ID

INNER JOIN DBO.DIM_TAMANHO AS Ta
ON V.Tamanho=Ta.Tamanho

INNER JOIN DBO.DIM_HORA AS H
ON V.Hora=H.Hora

INNER JOIN DBO.DIM_MINUTO AS M
```

```
ON V.Minuto=M.Minuto

INNER JOIN DBO.DIM_LOJA AS L
ON V.Loja_ID=L.Loja_ID
AND (L.Inicio<=V.Data
AND (L.Fim >= V.Data) OR (L.Fim IS NULL))

INNER JOIN DBO.DIM_CLIENTE AS C
ON V.Cliente_ID=C.Cliente

INNER JOIN DBO.DIM_CALENDARIO T
ON CONVERT(VARCHAR,T.Data,102) = CONVERT(VARCHAR,V.Data,102)
WHERE V.Data > @INICIAL AND V.Data < @FINAL
```

~

Anexo XII – Script de Povoamento da DIM_CALENDARIO

```
USE FOREVA_DW
GO

PRINT CONVERT (VARCHAR,GETDATE(),120)

DBCC CHECKIDENT (DIM_CALENDARIO,RESEED,1)

DECLARE @DATAINICIO DATETIME,
                @DATAFIM DATETIME,
                @DATA DATETIME

        PRINT GETDATE()

                        SELECT @DATAINICIO = '1/1/1950' ,
                              @DATAFIM = '1/1/2050'

                                    SELECT @DATA = @DATAINICIO

                        WHILE @DATA < @DATAFIM
                        BEGIN

                        INSERT INTO DIM_CALENDARIO
                        (
     [Data]
    ,[Dia]
    ,[DiaDaSemana]
    ,[Semana]
    ,[Mes]
    ,[NomeMes]
    ,[Trimestre]
    ,[NomeTrimestre]
    ,[Ano]
                        )

                        SELECT @DATA AS DATA,DATEPART(DAY,@DATA) AS Dia,
```

```
CASE DATEPART(DW,@DATA)
WHEN 1 THEN 'Domingo'
WHEN 2 THEN 'Segunda'
WHEN 3 THEN 'Terça'
when 4 then 'Quarta'
WHEN 5 then 'Quinta'
When 6 then 'Sexta'
WHEN 7 then 'Sábado'

END AS DiaDaSemana,

 datepart(week,@DATA) AS Semana,

        DATEPART(MONTH,@DATA) AS Mes,


CASE DATENAME(MONTH,@DATA)
WHEN 'January' then 'Janeiro'
WHEN 'February' then 'Fevereiro'
WHEN 'March' then 'Março'
WHEN 'April' then 'Abril'
WHEN 'May' then 'Maio'
WHEN 'June' then 'Junho'
WHEN 'July' then 'Junho'
WHEN 'August' then 'Agosto'
When 'September' then 'Setembro'
WHEN 'October' then 'Outobro'
WHEN 'November' then 'Novembro'
WHEN 'December' then 'Dezembro'

END AS NomeMes,


DATEPART(qq,@DATA) Trimestre,
CASE DATEPART (qq,@DATA)
WHEN 1 THEN 'PRIMEIRO'
WHEN 2 THEN 'SEGUNDO'
WHEN 3 THEN 'TERCEIRO'
WHEN 4 THEN 'QUARTO'
```

```
                    END AS NomeTrimestre

                    ,DATEPART(YEAR,@DATA) Ano

                    SELECT @DATA = DATEADD(dd,1,@DATA)
                    END

                    UPDATE DIM_CALENDARIO
                    SET Dia='0' + Dia
                    WHERE LEN(Dia) = 1

                    UPDATE DIM_CALENDARIO
                    SET Mes = '0' + Mes
                    WHERE LEN(Mes) = 1

                    UPDATE DIM_CALENDARIO
                    SET DataCompleta = Ano + Mes + Dia
                    GO

                    DECLARE C_TEMPO CURSOR FOR
SELECT IDSK, DataCompleta, DiaDaSemana, Ano FROM DIM_CALENDARIO
DECLARE  @ID INT,
                    @DATA varchar(10),
                    @DIASEMANA varchar(20),
                    @ANO char(4),
                    @FIMDESEMANA CHAR(4),
                    @ESTACAO VARCHAR(15)

                    OPEN C_TEMPO
FETCH NEXT FROM C_TEMPO
INTO @ID,@DATA,@DIASEMANA,@ANO
WHILE @@FETCH_STATUS = 0
BEGIN
IF @DIASEMANA IN ('Domingo','Sábado')
Set @FIMDESEMANA = 'Sim'
ELSE
SET @FIMDESEMANA = 'Não'
```

```
IF @DATA BETWEEN CONVERT( CHAR(4),@ano) + '0923'
AND Convert (CHAR(4),@ANO)+'1220'
SET @ESTACAO = 'Outono'

Else IF @DATA BETWEEN CONVERT( CHAR(4),@ano) + '0321'
And Convert(Char(4),@ANO)+'0620'
SET @ESTACAO = 'Primavera'

ELSE IF @DATA BETWEEN CONVERT (CHAR(4),@ano) + '0621'
AND Convert (Char(4),@ANO) + '0922'
SET @ESTACAO = 'Verão'

ELSE
SET @ESTACAO = 'Inverno'

update DIM_CALENDARIO SET FimSemana = @FIMDESEMANA
WHERE IDSK = @ID

UPDATE DIM_CALENDARIO SET EstacaoAno = @ESTACAO
WHERE IDSK = @ID

FETCH NEXT FROM C_TEMPO INTO @ID, @DATA, @DIASEMANA, @ANO

END
CLOSE C_TEMPO
DEALLOCATE C_TEMPO
GO
```

Anexo XIII – Implementação Física da Base de Dados de Recomendações

```
CREATE DATABASE RECOMEND_FOREVA
GO
USE RECOMEND_FOREVA
GO

CREATE TABLE ARTIGO (
ID INT PRIMARY KEY IDENTITY,
ARTIGO VARCHAR(20) UNIQUE NOT NULL
)

GO

CREATE TABLE LOJA (
ID INT PRIMARY KEY IDENTITY,
LOJA VARCHAR(20) UNIQUE NOT NULL
)
GO

CREATE TABLE ESTACAO (
ID INT PRIMARY KEY IDENTITY,
ESTACAO VARCHAR(30) UNIQUE NOT NULL
)
GO

CREATE TABLE SISTEMA (
ID INT PRIMARY KEY IDENTITY,
TIPO_SISTEMA VARCHAR(30) UNIQUE NOT NULL
)
GO

CREATE TABLE CATEGORIA (
ID INT PRIMARY KEY IDENTITY,
CATEGORIA VARCHAR(50) UNIQUE NOT NULL
)
GO
```

```
CREATE TABLE REC_BC (ID_ARTIGO INT REFERENCES ARTIGO(ID),
ID_LOJA INT REFERENCES LOJA(ID),
CATEGORIA INT REFERENCES CATEGORIA(ID),
DATA DATE NOT NULL,
ESTACAO INT REFERENCES ESTACAO(ID),
SISTEMA INT REFERENCES SISTEMA(ID) NOT NULL,
VENDAS_TOTAIS INT NOT NULL
 primary key (ID_ARTIGO,ID_LOJA,DATA,ESTACAO)
)
GO

CREATE TABLE REC_TN(ID_ARTIGO INT REFERENCES ARTIGO(ID),
ID_LOJA INT REFERENCES LOJA(ID),
DATA DATE NOT NULL,
ESTACAO INT REFERENCES ESTACAO(ID),
SISTEMA INT REFERENCES SISTEMA(ID) NOT NULL,
RANKING INT NOT NULL,
VENDAS_TOTAIS INT NOT NULL
 primary key (ID_ARTIGO,ID_LOJA,DATA,ESTACAO)
)
GO

CREATE TABLE REC_FC(ID_ARTIGO INT REFERENCES ARTIGO(ID),
ID_LOJA INT REFERENCES LOJA(ID),
DATA DATE NOT NULL,
ESTACAO INT REFERENCES ESTACAO(ID),
SISTEMA INT REFERENCES SISTEMA(ID) NOT NULL,
RMSE_MEDIA NUMERIC(10,2) NOT NULL,
PREVISAO_AVALIACAO NUMERIC(10,2) NOT NULL
primary key (ID_ARTIGO,ID_LOJA,DATA,ESTACAO)
)
GO
```