Universidade do Minho
Mestrado Integrado em Engenharia Informática
Mestrado em Engenharia Informática

Unidade Curricular de
Data Warehousing

Ano Lectivo de 2018/2019

Desenvolvimento de um Sistema de *Data Warehousing* – Jardim Zoológico

David Sousa – a78938
Isabel Pereira – a76550
Francisco Matos – a77688
Tiago Alves – a78218
Ricardo Neves – a78764

Janeiro, 2019

DW

Data de Recepção	
Responsável	
Avaliação	
Observações	

Desenvolvimento de um Sistema de Data Warehousing – Jardim Zoológico

David Sousa – a78938
Isabel Pereira – a76550
Francisco Matos – a77688
Tiago Alves – a78218
Ricardo Neves - a78764

Janeiro, 2019

Resumo

Este trabalho, no âmbito da Unidade Curricular de *Data Warehousing* do Mestrado em Engenharia Informática, tem como objetivo modelar e construir um sistema de apoio à decisão para o Zoo do Sr. Paulo, bem como apresentar todo o processo de implementação e migração de dados para o sistema de *Data Warehousing*.

O desenvolvimento deste sistema operacional é de alta importância para o Zoo, que pretende aumentar os seus lucros ao ter melhor atenção aos gastos que envolvem a compra de alimentos.

Neste relatório segue toda a descrição e explicação do processo de implementação, planeamentos e decisões tomadas ao longo da realização deste trabalho.

Área de Aplicação: Business Intelligence, Data Warehousing e Sistemas Operacionais.
Palavras-Chave: Bases de Dados Relacionais, Sistemas Operacionais, Business Intelligence, Entidades, Relacionamentos, Metodologia, Modelo Conceptual, Modelo Lógico.

Índice

Índice de Figuras

Índice de Tabelas

1. Introdução

1.1. Contextualização do sistema de *Data Warehouse*

O crescente sucesso empresarial de Filipe Sousa noutras áreas fê-lo afastar do negócio que tinha sido o despoletar de todo o seu sucesso, o jardim Zoológico sobre o qual tomou as rédeas depois do falecimento de seu pai. Uma vez que, o seu foco era outro, Paulo Sousa deixou o Zoo ao cargo de seu irmão mais novo, Manuel Sousa. Manuel, sempre foi bastante responsável pelo que queria demonstrar o seu valor perante o irmão. No entanto, a falta de experiência em negócios deste tipo levou a uma queda dos lucros que se foi agravando com o passar dos meses. Seguindo as pisadas do irmão, Manuel voltou a contratar a equipa de engenheiros informáticos de forma a perceber se era possível que, através da alteração da base de dados existente, se apresentassem dados que fossem mais relevantes para uma correta gestão económica do negócio.

1.2. Apresentação do caso de estudo

Este projeto surge com o intuito de minimizar os custos associados à alimentação dos animais, e consequentemente maximizar os lucros do Zoo da família Sousa. A equipa, de forma a ir ao encontro de uma solução, sugeriu a implementação de um Sistema de Suporte à Decisão capaz de analisar os dados relacionados com esses custos. Esta análise envolve todo o percurso que os alimentos tomam desde a sua compra até à sua ingestão por parte do animal. Efetivamente, a análise recairá sobre o preço associado a cada fornecedor, sobre o stock existente no Zoo, sobre a quantidade a adquirir e, finalmente, sobre a quantidade a consumir pelo animal.
Deste modo, concluímos que o sistema mais adequado será um *Data Warehouse* uma vez que estes apresentaram as seguintes características:

- Permitem extrair informação existente em várias fontes de dados;

- Preparar os dados de forma a dar respostas úteis a questões relacionadas com o negócio;
- Maior organização e consistência;
- Permite analisar o que ocorreu no passado, dando uma visão histórica de tudo o que ocorreu na empresa.

Em suma, uma vez que um *Data Warehouse* possui as vantagens enumeradas e visto que a Base de Dados já se encontra implementada, tanto relacional como não relacional, e que a equipa possui um grande conhecimento sobre a mesma, todo o processo será menos custoso.

1.3. Motivação e Objetivos

A principal motivação que levou o Sr. Manuel, a recorrer a um grupo de engenheiros informáticos foi o facto de este ter constatado uma quebra nos lucros do Jardim Zoológico. Sendo assim, o principal objetivo da equipa na elaboração deste projeto será ajudar Manuel no aumento dos seus lucros para que este possa recuperar assim o dinheiro perdido.

Efetivamente, através de um Sistema de *Data Warehouse* torna-se possível obter uma melhor gestão dos alimentos, pelo facto de se poder responder a questões relativas à compra dos mesmos como quando, a quem, quanto encomendar e o seu custo. Com isto, é possível gerar uma redução nos custos do Zoo, sendo que os dados estarão mais organizados, prontos para um acesso rápido e análise profunda, na esperança de dar um novo rumo à empresa.

1.4. Análise da Viabilidade do projeto

Tendo em conta o estado financeiro atual do Zoo, derivada da má gestão de Manuel Sousa, considera-se que o projeto deva ser desenvolvido o mais rápido possível, de forma a devolver a glória de tempos passados à empresa.

Visto que a equipa já tem conhecimento do funcionamento da empresa e das Bases de Dados implementadas, considerou-se a implementação de um Sistema de Suporte à Decisão não só viável dentro do tempo estipulado, como também completamente funcional, uma vez que, o grau de complexidade não é bastante elevado.

Em suma, espera-se que o dinheiro investido no novo sistema seja rapidamente recuperado com a gestão eficiente dos recursos.

2. Planeamento e Gestão do Projeto

2.1. Definição da identidade do projeto

Nome: Jardim Zoológico do Paulo.

Categoria: Gestão das compras.

Descrição: A implementação de um *Data Warehouse* tem como objetivo primário a gestão de despesas relativas a fornecedores do Zoo. Nesse sentido, o *Data Warehouse* deverá permitir fazer uma análise relativa a estes gastos de modo a que as decisões futuras sejam influenciadas.

Pessoas envolvidas: São várias as pessoas envolvidas no projeto. Em primeiro lugar, destaca-se a equipa de projeto. Esta equipa é constituída por cinco pessoas pelo que cada uma desempenha um papel fundamental na implementação do *Data Warehouse*. Apesar de todo o trabalho ter sido desenvolvido em conjunto, envolvendo a participação de todos os intervenientes em todas as áreas, destacam-se algumas características de cada um dos elementos. Isabel Pereira e David Sousa, a formação destes elementos no que diz respeito à recolha de requisitos fazem destes o principal ponto de contacto com o cliente. Francisco Matos e Tiago Alves, apresentam um basto conhecimento no que diz respeito à área da engenharia de conhecimento pelo que serão os principais responsáveis pela conceção e povoamento do *Data Warehouse*. No que diz respeito a Ricardo Neves, este elemento ficará mais ligado a todos os aspetos referentes à modelação. Em segundo lugar, destaca-se todos aqueles que dizem respeito ao projeto, como é o caso do dono do Zoo e seu irmão Manuel.

2.2. Identificação de recursos

Este sistema possuirá pelo menos dois tipos de utilizadores: Administradores e Analista. Aqueles que possuem o perfil de Administrador estão encarregues de, depois de criado o sistema de *Data Warehousing*, garantir que os dados continuam a ser inseridos corretamente e tratarão de possíveis falhas do processo ETL. Quanto ao Analista, este papel será desempenhado pelo dono do Jardim Zoológico, que com através aos dados estatísticos fornecidos pelo *Data Warehouse* procurará efetuar melhores decisões financeiras.

Sem os recursos materiais e financeiros nunca seria possível realizar este projeto. Assim, numa fase inicial, foi preciso analisar se estes recursos estariam disponíveis para o grupo. Deste modo, foi definido que seria necessário um computador por cada elemento da equipa, cada um contendo o software necessário para a execução do trabalho: *Kettle, Microsoft Word, Excel, Project, MySQL, Neo4J*, entre outras ferramentas essenciais. Tendo em conta que o Jardim Zoológico não se encontrava financeiramente estável, foi acordado que o trabalho realizado pela equipa seria pago na totalidade pelo cliente.

Outro recurso que não pode ser esquecido é a existência de transporte, para as deslocações dos vários elementos do grupo, de modo a juntarem-se num único local, de forma a que o trabalho seja realizado mais rápido e eficientemente.

2.3. Estabelecimento do plano de desenvolvimento

O plano de desenvolvimento é um aspeto bastante importante a ter em conta em qualquer projeto, uma vez que, proporciona uma organização detalhada, melhorando a distribuição de carga de trabalho entre os vários dias designados. Para tal, construiu-se o seguinte Diagrama de Gantt, que se trata de um gráfico utilizado para ilustrar os intervalos de tempo relativos ao início e á finalização das diferentes etapas do projeto. Efetivamente, este é bastante útil uma vez que permite efetuar um controlo relativamente ao trabalho realizado.

Nome da Tarefa	Duração	Início	Conclusão
Fundamentação do projeto	3 dias	Sex 05/10/18	Ter 09/10/18
Definição do plano de desenvolvimento do DW	3 dias	Qua 10/10/18	Sex 12/10/18
Identificação dos utilizadores do sistema	2 dias	Sáb 13/10/18	Dom 14/10/18
Identificar e analisar as fontes de informação	3 dias	Seg 15/10/18	Qua 17/10/18
Construção da matriz de decisão	3 dias	Qui 18/10/18	Dom 21/10/18
Seleção do Data Mart a desenvolver	6 dias	Seg 22/10/18	Sáb 27/10/18
Escolha do grão	4 dias	Dom 28/10/18	Qua 31/10/18
Escolha das dimensões de análise	3 dias	Qui 01/11/18	Seg 05/11/18
Desenvolvimento do diagrama de tabelas de facto	4 dias	Ter 06/11/18	Sex 09/11/18
Especificar as tabelas de facto	4 dias	Sáb 10/11/18	Qua 14/11/18
Caracterização do processo de manipulação de dados (povoamento)	3 dias	Qui 15/11/18	Dom 18/11/18
Projeção do ETL com o modelo BPMN	6 dias	Seg 19/11/18	Sáb 24/11/18
Revisão do projeto realizado até então	6 dias	Dom 25/11/18	Sex 30/11/18
Análise e caracterização das fontes de informação	6 dias	Sáb 01/12/18	Sex 07/12/18
Análise dos dados contidos nas fontes	6 dias	Sáb 08/12/18	Sex 14/12/18
Implementação do sistema de povoamento (ETL)	14 dias	Sáb 15/12/18	Qua 02/01/19
Definição do source-to-target data map	4 dias	Qui 03/01/19	Ter 08/01/19
Modelação lógica do sistema de ETL	5 dias	Qua 09/01/19	Ter 15/01/19
Implementar e testar o sistema	4 dias	Qua 16/01/19	Sáb 19/01/19
Análise crítica do resultado	4 dias	Dom 20/01/19	Qua 23/01/19

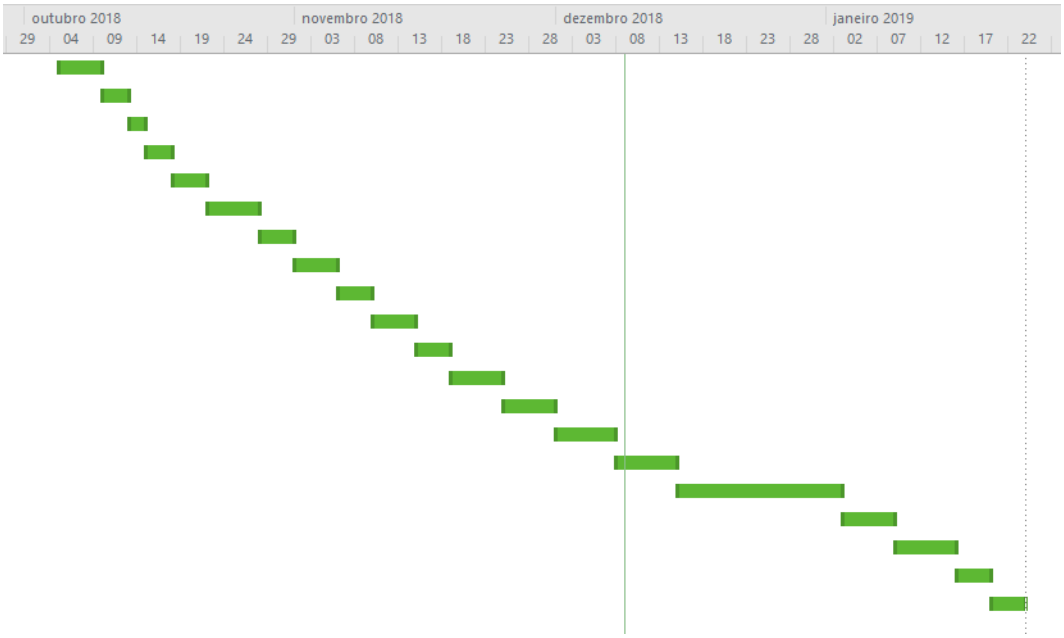


Figura 1 - Diagrama de Gantt

2.4. Definição de medidas de sucesso

Para que um *Data Warehouse* seja bem-sucedido é necessário definir um conjunto de medidas que devem ser atingidas. Nesse sentido, enumeram-se de seguida algumas dessas medidas:

- O *Data Warehouse* deve atender às necessidades do Jardim Zoológico;
- O desenvolvimento do *Data Warehouse* encontra-se dentro do valor de investimento e do tempo definido para a entrega;
- O *Data Warehouse* permite um retorno do dinheiro investido;
- Os requisitos levantados são satisfeitos.

3. Levantamento e Análise de Requisitos

3.1. Apresentação do método de aquisição de requisitos

Com o desenvolvimento de qualquer projeto é necessário estabelecer quais os requisitos do mesmo. Como este projeto é oriundo de um pedido de um cliente, a equipa escolheu realizar uma entrevista com essa entidade. A entrevista tem como intuito estabelecer os requisitos necessários para o *Data Warehouse* a implementar e, para tal, a equipa marcou uma data e local para se encontrar com o entrevistado e criou um conjunto de questões pertinentes para colocar ao mesmo.

No dia da entrevista, a equipa encontrou-se com o entrevistado no local estipulado, uma sala de reuniões localizada no Zoo.

Com o início da entrevista, todos os envolvidos cumprimentaram-se e o assunto foi logo abordado sem rodeios. Sendo assim, como os requisitos de um *Data Warehouse* baseiam-se no que é suposto analisar com os dados de uma ou mais fontes de informação, a equipa fez duas perguntas cruciais ao cliente.

A primeira pergunta passa por conhecer quais as fontes de informação que o Zoo possui para armazenar os dados. A essa questão, o cliente respondeu que possuía 3 bases de dados diferentes, das quais a equipa verificou ser uma base de dados relacional MySQL, uma base de dados não relacional orientada a grafos Neo4J (ambas já conhecidas pela equipa de desenvolvimento) e um ficheiro CSV.

A segunda pergunta teve como objetivo tomar conhecimento de quais os dados que o cliente pretende analisar. Como o intuito é de minimizar os custos da alimentação dos animais através de uma melhor gestão dos alimentos a encomendar, foi natural que a sua resposta envolvesse as palavras compra, fornecedor, alimento e refeição. Esta conclusão proveio da utilização, por parte do cliente, de um conjunto de frases que serão apresentadas de seguida.

Relativamente aos fornecedores, o cliente expressou a sua vontade ao dizer "Gostaria de saber quais são os fornecedores a quem mais compro e também seria importante para mim tomar conhecimento de quais desses é que vendem um certo alimento mais barato, assim posso verificar se estou a escolher corretamente os fornecedores.". A isto, a equipa sugeriu que se efetuasse ainda mais uma pesquisa de qual o fornecedor que vende um alimento mais barato pois o fornecedor que mais vende ao Zoo pode não ser o que vende a um preço mais acessível um certo alimento, ao que

o entrevistado concordou. Além disso, a equipa sugeriu que essas 3 análises fossem efetuadas sobre um intervalo de datas devido ao facto de se poder ver uma evolução entre as escolhas dos fornecedores ou até mesmo verificar padrões de quem efetua melhores preços de um determinado alimento em certas alturas do ano. Nesta última afirmação o cliente também se mostrou bastante recetivo e feliz com a sugestão.

Mudando o tema para os alimentos, o cliente, tendo em conta as ideias e sugestões anteriores, afirmou que "Nos alimentos gostaria de saber os mais utilizados para alimentar os animais dentro de um intervalo de datas, assim teria uma ideia de quais são os mais importantes ter sempre disponíveis. Também gostaria que de alguma forma fosse possível descobrir quanto é que tenho de comprar semanalmente de cada alimento e determinar quais os que têm stock abaixo desse valor, de modo a otimizar as minhas compras. Além disso, gostaria de ter uma funcionalidade quase idêntica a uma feita para o fornecedor, mas que neste caso me diria quando é que um alimento está mais barato, entre um intervalo de datas, para poder verificar padrões de quais os meses em que um alimento é mais acessível.".

Neste ponto o cliente confessou não ter mais ideias de análise, ao qual a equipa sugeriu analisar-se a progressão dos custos, já que o objetivo seria a minimização dos mesmos, assim poderia criar-se uma funcionalidade que me permitisse verificar os custos totais que o cliente teve durante um intervalo de datas para assim verificar se os consegui ou não reduzir.

Por último, o entrevistado pediu que a análise pudesse ser consultada por o mesmo através de uma aplicação em que demorasse no máximo 30 minutos para este aprender a trabalhar com ela, facilitando assim o seu trabalho.

Com a realização desta entrevista foi possível identificar os requisitos necessários para a construção do *Data Warehouse* e garantir que todas as vontades do cliente fossem garantidas. Desta forma, a equipa pôde estabelecer quais os atributos e as dimensões a criar (fornecedor, alimento, refeição e data), sendo que refeição será hierarquia de alimento e a compra a tabela de factos. Além disso, foi acessível definir quais os relacionamentos entre tabelas pelo enorme conhecimento que a equipa já possuía sobre as bases de dados existentes.

3.2. Requisitos de descrição

RD1: O sistema deve aceitar várias fontes de informação quer seja SQL, NoSQL ou até mesmo Excel.

RD2: O sistema deve recolher apenas informações que já não possua.

RD3: O sistema deve aceitar mais que uma compra.

RD4: O sistema deve armazenar em cada compra um identificador, a quantidade e o preço por unidade do alimento adquirido e o valor total dessa compra.

RD5: Qualquer alteração na compra é considerada uma nova compra.

RD6: O sistema deve aceitar mais que um fornecedor.

RD7: O sistema deve armazenar em cada fornecedor um identificador, uma designação, um e-mail e um contacto.

RD8: Qualquer alteração no fornecedor é considerada um novo fornecedor.

RD9: Uma compra é realizada a apenas um fornecedor, no entanto, um fornecedor pode participar em uma ou mais compras.

RD10: O sistema deve aceitar mais que uma entrada em calendário.

RD11: O sistema deve armazenar em cada entrada de calendário um identificador, uma data, um dia, uma semana, um mês, um semestre e um ano.

RD12: Qualquer alteração no calendário é considerada uma nova entrada no calendário.

RD13: Uma compra é realizada em apenas uma data do calendário, no entanto, uma data no calendário pode estar presente em uma ou mais compras.

RD14: O sistema deve aceitar mais que um histórico de stocks.

RD15: O sistema deve armazenar em cada histórico de stocks um identificador e o stock.

RD16: O sistema deve aceitar mais que um alimento.

RD17: O sistema deve armazenar em cada alimento um identificador, o seu nome e o stock.

RD18: A alteração do nome de um alimento é considerada um novo alimento.

RD19: A alteração do stock de um alimento é considerada uma nova entrada em histórico de stocks desse alimento.

RD20: Um alimento pode conter um ou mais históricos de stock, no entanto, um histórico de stock refere-se apenas a um alimento.

RD21: A compra é efetuada sobre apenas um alimento, no entanto, um alimento pode estar presente em uma ou mais compras.

RD22: O sistema deve aceitar mais que uma refeição.

RD23: O sistema deve armazenar em cada refeição um identificador e uma quantidade.

RD24: Qualquer alteração na refeição é considerada uma nova refeição.

RD25: Um alimento pode estar presente em uma ou mais refeições, no entanto, uma refeição só contém apenas um alimento.

3.3. Requisitos de exploração

RE1: O sistema deve permitir determinar quais os fornecedores (Designação, E-mail, Contacto e Quantidade vendida) que mais venderam ao Zoo, dentro de um intervalo de datas, organizando-os pelo que mais forneceu até ao menor.

RE2: O sistema deve permitir determinar quais os fornecedores (Designação, E-mail, Contacto e Quantidade vendida) que mais venderam ao Zoo, dentro de um intervalo de datas, um determinado alimento (através da pesquisa por parte do seu nome), organizando-os pelo que mais forneceu até ao menor.

RE3: O sistema deve permitir determinar qual o fornecedor (Designação, E-mail, Contacto e Preço por Unidade) que vende um determinado alimento (através da pesquisa por parte do seu nome) a um valor mais baixo, dentro de um intervalo de datas.

RE4: O sistema deve permitir determinar quais os alimentos (Nome e Stock), relativamente à sua última entrada, que apresentam um stock menor ou igual a um determinado valor.

RE5: O sistema deve permitir determinar quais os alimentos (Nome e Quantidade utilizada) mais utilizados nas refeições, dentro de um intervalo de datas, organizando-os pelo mais utilizado até ao menos.

RE6: O sistema deve permitir determinar qual a quantidade média necessária a comprar semanalmente de um determinado alimento (através da pesquisa por parte do seu nome).

RE7: O sistema deve permitir determinar qual a quantidade dos alimentos, dentro de um intervalo de datas, que foram adquiridos.

RE8: O sistema deve permitir determinar qual o montante total gasto em alimentos num determinado intervalo de datas.

RE9: O sistema deve permitir determinar qual a data (Data e Preço por Unidade), dentro de um intervalo de datas, em que um determinado alimento (através da pesquisa por parte do seu nome) apresentou um valor mais baixo.

3.4. Requisitos de controlo e acesso

RC1: O sistema deve possuir um perfil de utilização, destinado ao programador da aplicação, que apenas permita consulta.

3.5. Revisão dos requisitos com o Utilizador

Após o estabelecimento dos requisitos, a equipa reuniu-se novamente com o cliente para este verificar se todos os seus ideais estavam bem definidos.

O cliente, como entidade não entendedora da estrutura de um *Data Warehouse*, teve algumas dificuldades em compreender o significado de alguns requisitos descritivos, nomeadamente, o facto de haver dados passados e futuros fazer com que fosse necessário o estabelecimento de regras com a alteração de alguns atributos. Como esses aspetos não foram tratados na entrevista, mas sim a equipa que estabeleceu, o cliente analisou esses requisitos com maior atenção chegando à conclusão que concordava com os mesmos, menos com o facto de uma alteração no e-mail ou contacto de um fornecedor gerar um novo fornecedor. A isto a equipa respondeu que tal se devia ao facto da urgência da implementação do DW tendo em conta o problema do Zoo. Tal poderia ser tratado futuramente, já que não seria crucial para as análises requeridas pelo cliente na entrevista. Com esta justificação o cliente entendeu a situação e concordou.

Por fim, o entrevistado possuía a ideia errada que a equipa de administração de BD e DW iria implementar uma aplicação para o mesmo, tal como este teria mencionado na entrevista. A esta situação, a equipa referiu que isso se tratava de uma área de informática diferente e, para tal, o cliente teria de contratar um especialista nesse ramo. De qualquer forma, o perfil de utilização mencionado nos requisitos de controlo seria criado e seria destinado ao programador da aplicação. Por sua vez, este estaria encarregue de ceder esses privilégios ao cliente através de uma interface com as características pedidas pelo mesmo, sendo que a interface teria de permitir as várias pesquisas enumeradas nos requisitos de exploração.

Dito isto, a reunião deu-se como terminada e o cliente saiu clarificado sobre quaisquer dúvidas que possuía e houve um consenso entre as duas partes sobre os requisitos estabelecidos.

4. Modelação Dimensional do Sistema

4.1. Apresentação da metodologia de desenvolvimento

A modulação dimensional dos dados de um projeto de *Data Warehousing* é uma das atividades mais importantes. A sua relevância provém do facto de esta envolver o processo de construção dos esquemas dimensionais, estando estes de acordo com os objetivos de análise dos agentes de decisão. Desta forma, para a correta implementação de todos os seus passos, foi necessária a escolha de uma metodologia.

A metodologia escolhida para a implementação da modelação dimensional do sistema, foi o método dos “4 passos” desenvolvido por Kimball e Ross em 2002. Nesta metodologia aplica-se uma abordagem de baixo para cima (*bottom-up*), fazendo com que todos os tipos de objetos de dados, que possamos encontrar em esquemas dimensionais, sejam desenvolvidos sem esquecimentos. Sendo assim, os passos são os enumerados de seguida, sendo estes definidos nas secções posteriores.

1. Seleção da área de suporte à decisão a implementar (*data mart*);
2. Definição do grão;
3. Definição das dimensões de análise;
4. Definição das medidas da tabela de factos.

4.2. Definição e caracterização do *data mart*

A área de negócio permite que não se perca o objetivo do processo sobre o qual se pretende tomar as decisões. Sendo assim, no caso deste projeto, a área recai sobre a análise da compra de alimentos do Zoo aos fornecedores do mesmo. Desta forma, apesenta-se de seguida a matriz de decisão com a caracterização deste *data mart*.

4.2.1 Esquematização da matriz de decisão

Caracterização do Data Mart "Compra"	
Identificação: Compra de alimentos aos fornecedores	
Descrição geral: Informação para suporte à tomada de decisão na área de compras providenciando elementos de dados selecionados acerca da compra de alimentos aos fornecedores, para gestão e controlo das ações comercias realizadas	
Estrutura base:	
Tabela de factos	TF - Compra
Dimensões:	
Calendário	X
Refeição	X
Alimento	X
Fornecedor	X
Número de dimensões	4
Tipo	Transacional
Periocidade	Realizado semanalmente
Descrição	Transações de alimentos
Utilidade estratégica	Avaliação dos gastos relativos à alimentação dos animais. Identificar e caracterizar o fornecedor com ofertas mais baratas. Reconhecimento de épocas baixas. Caracterização do perfil de alimentação dos animais. Otimização de stocks
Utilizadores	Administrador e gestor do Jardim Zoológico
Observações	
Nada a assinalar	

Tabela 1 - Caracterização do Data Mart "Compra"

4.3. Definição e caracterização *do grão*

Com o grão é possível detalhar a informação presente nas estruturas de dados do *Data Warehouse* de forma mais atómica possível. A definição correta deste é crucial para a exploração consistente da tabela de factos. Desta forma, pode-se afirmar que o

grão deste projeto passa por explorar a compra de determinada quantidade, a um determinado preço (originando assim um preço por unidade), de um alimento (alterando-se o seu stock), a um fornecedor específico, efetuada numa determinada data, e que posteriormente será consumida determinada quantidade desse alimento numa refeição.

4.4. Definição e caracterização das dimensões

Os modelos dimensionais são construídos de acordo com os processos de negócio e de tomada de decisão e, sendo assim, torna-se necessário definir as dimensões que serão utilizadas no *Data Warehouse*.

Tendo em consideração os requisitos descritivos, o *data mart* e o grão (definidos nas secções anteriores) e sabendo que a modelação dimensional de dados é uma atividade que se baseia nos requisitos de tomada de decisão e não nos requisitos de processos de suporte operacional, iremos apresentar e caracterizar abaixo as dimensões do sistema.

Dimensões do Data Mart "Compra"			
Número	**Identificação**	**Descrição**	**Esquema**
1	Alimento	Tipos de alimento e as suas quantidades em stock	Alimento (com variação e com história)
2	Refeição	Informação sobre as refeições feitas pelos animais	Refeição (com variação e sem história)
3	Fornecedor	Identificação dos vários fornecedores, responsáveis por entregar os diferentes alimentos	Fornecedor (com variação e sem história)
4	Calendário	Dimensão temporal	Calendário

Tabela 2 - Dimensões do Data Mart "Compra"

4.4.1 Dimensão Alimento

Esta dimensão armazena a informação sobre os alimentos comprados pelo Zoo, que podem servir de refeição aos vários animais. Esta tem como atributos o seu identificador, quantidade em stock, nome. Temos a possibilidade de os agrupar por quantidade e stock.

Dimensão	Atributo	Descrição	Tipo de Dados	Variável	Exemplo
Alimento	idAlimento	Identificador único do Alimento	Inteiro	Não	3
	Stock	Quantidade de alimento disponível no Zoo	Inteiro	Sim (com história)	50
	Nome	Nome comum do alimento	Varchar(45)	Sim (Tipo 2)	“Carne”

Tabela 3 - Dimensão Alimento

4.4.2 Dimensão Refeição

Com esta dimensão, somos capazes de guardar a informação relativa à alimentação dos animais, tendo como atributos o seu identificador, quantidade de alimento dado ao animal, horário e ID do alimento (chave estrangeira pois esta é hierarquia de Alimento). Podemos agrupar as refeições por quantidade de alimento, horário e ID do alimento dado ao animal.

Dimensão	Atributo	Descrição	Tipo de Dados	Variável	Exemplo
Refeição (Hierarquia de Alimento)	idRefeicao	Identificador único da Refeição	Inteiro	Não	10
	Quantidade	Quantidade de alimento consumido pelo animal	Inteiro	Sim (Tipo 2)	1

	Horario	Hora da refeição do animal	DATE	Sim (Tipo 2)	2018-11-09
	Alimento_idAlimento	Identificador do alimento constituinte da refeição	Inteiro	Não	5

Tabela 4 - Dimensão Refeição

4.4.3 Dimensão Fornecedor

A dimensão Fornecedor guarda todas as informações sobre as empresas que vendem os vários alimentos ao Zoo a troco de um valor monetário decidido pelas mesmas. Esta é composta pelo atributo indentificador único, designação (nome da empresa), contacto, email.

Dimensão	Atributo	Descrição	Tipo de Dados	Variável	Exemplo
Fornecedor	idFornecedor	Identificador único do Fornecedor	Inteiro	Não	6
	Designacao	Nome do fornecedor	Varchar (45)	Sim (Tipo 2)	“Pingo Doce”
	Contacto	Número de telefone do fornecedor	Inteiro	Não	253 123 456
	Email	Email do fornecedor	Varchar (45)	Sim (Tipo 2)	fornecedor@pingodoce.pt

Tabela 5 - Dimensão Fornecedor

4.4.4 Dimensão Calendário

Com a dimensão Calendário temos a possibilidade de guardar a data exata da venda de uma certa quantidade de alimento para o Zoo. Nesta dimensão guarda-se o identificador único da data em que aconteceu a venda, a data exata, o dia, semana,

mês, semestre e ano da transação. Podemos agrupar esta dimensão por quase todos os seus atributos, excluindo apenas o seu identificador, por ser único.

Dimensão	Atributo	Descrição	Tipo de Dados	Variável	Exemplo
Calendário	idCalendario	Identificador único do Calendário	Inteiro	Não	40
	Data	Data completa	DATE	Não	2018-11-09
	Dia	Dia da data (1-31)	Inteiro	Não	09
	Semana	Semana da data (1-52)	Inteiro	Não	46
	Mês	Mês da data, por nome do mês	Varchar(45)	Não	11
	Semestre	Semestre da data (1 ou 2)	Inteiro	Não	2
	Ano	Ano da data	Inteiro	Não	2018

Tabela 6 - Dimensão Calendário

4.4.5 Atributos com Variação

Segundo conseguimos inferir das tabelas anteriores podemos distinguir dois tipos de variação. Nesse sentido *Kimball e Ross* categoriza as dimensões essencialmente em quatro tipos, sendo que para o âmbito deste projeto apenas é importante considerar dois:

- ✓ **Tipo 2:** Criação de novos registos na tabela base;
- ✓ **Tipo 4:** Criação de tabelas de histórico.

As dimensões com atributos com variação são a Dimensão Fornecedor, Refeição e Alimento.

Podemos afirmar que o Email da Dimensão Fornecedor são atributos de variação por método de substituição sem registo de histórico. Isto deve-se ao facto de, ao longo do tempo, um Fornecedor ter a capacidade de criar um novo email empresarial. Também o Nome da Dimensão Alimento e Quantidade da Dimensão Refeição podem variar, mas sempre sem qualquer gravação do histórico de alterações.

Por fim, o atributo de variação, presente no Alimento, por método de substituição com registo histórico é o stock, para assim se poder observar e analisar a variância do mesmo no armazém do Zoo.

4.5. Definição e caracterização da tabela de factos

Caracterização da Tabela de Factos					
Identificação			TF-Compra		
Descrição			Armazena todos os registos de compra de alimentos do Zoo		
Data Mart			Compra de alimentos aos fornecedores		
Tipo			Transacional		
Utilidade Estratégica			Avaliação dos gastos relativos à alimentação dos animais. Identificar e caracterizar o fornecedor com ofertas mais baratas. Reconhecimento de épocas baixas. Caracterização do perfil de alimentação dos animais. Otimização de stocks		
Povoamento			Realizado semanalmente		
Dimensão inicial			0.26MB[1]		
Crescimento			10% por semana[1]		
Período de Dados			Desde o ano de 2018. Os anos anteriores ficarão em arquivos.		
Dimensões					
Nr	**Identificação**	**Chave**	**Domínio**	**Descrição**	**Exemplo**
1	idFornecedor	S	INT	Código único do Fornecedor	1
2	idCalendario	S	INT	Código único da Data	1
3	idAlimento	S	INT	Código único do Alimento	1
Medidas					
Nr	**Identificação**	**Domínio**	**Tipo (Função)**	**Descrição**	**Exemplos**

[1] Tendo em consideração a dimensão estimada para as diversas fontes

1	Quantidade	INT	Agregável (SUM)	Quantidade de unidade compradas do alimento em questão	20
2	Valor	DOUBLE	Agregável (SUM)	Valor pago pelo Zoo ao Fornecedor em questão	15.0
3	PrecoUnidade	DOUBLE	Não Agregável	Preço unitário do alimento comprado	2.0
Índice					
Nr	**Identificação**	**Tipo**		**Descrição**	
1	idTF-Compra	Primário		Único, ordenado fisicamente de forma crescente.	
2	idFornecedor	Secundário		Ordenado de forma crescente.	
3	idCalendario	Secundário		Ordenado de forma crescente.	
4	idAlimento	Secundário		Ordenado de forma crescente.	
Perfis de Execução					
Administrador da base de dados e gestores do Zoo.					
Observações					
Todos os valores considerados encontram-se em Euros (€).					

Tabela 7 - Caracterização da tabela de factos

4.6. Esquematização do esquema dimensional

De acordo com o a modelação dimensional estipulada neste capítulo foi possível desenvolver o esquema dimensional apresentado de seguida. Este esquema foi elaborado utilizando a notação de Golfarelli, et al, de 1998.

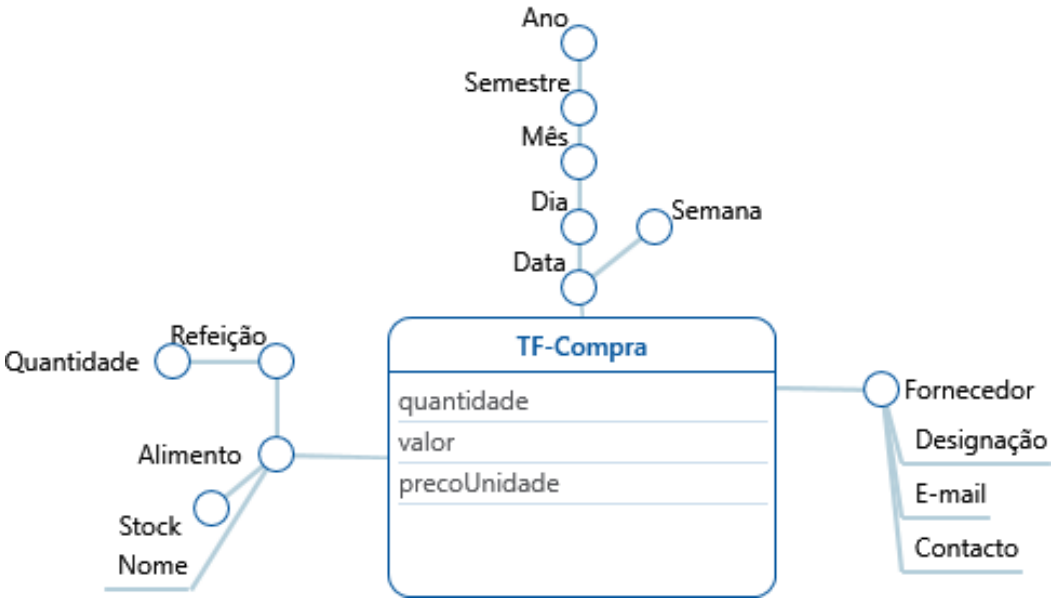


Figura 2 – Esquema do Modelo Dimensional

4.7. Revisão do esquema dimensional desenvolvido

Após a equipa de desenvolvimento ter chegado a um consenso relativo ao esquema dimensional desenvolvido, procedeu-se à revisão deste com o dono de Zoo, de forma a confirmar que este se encontrava de acordo com o desejado. Após aprovação de Manuel Sousa sem grandes discordâncias, a equipa continuou a elaboração do *Data Warehouse.*

5. Caracterização das Fontes de Informação

5.1. Identificação e descrição das fontes de informação do sistema

Visto que este projeto utiliza a base de dados já criada pela equipa para o Zoo, podemos facilmente identificar duas fontes de informação do sistema, a base de dados relacional, criada em *MySQL* e a base de dados não relacional, criada em *Neo4J*. Além destas duas, o cliente apresentou uma nova "base de dados" em formato *CSV.* Sendo assim, nesta secção encontram-se apresentadas e descritas todas estas fontes.

5.1.1 Base de dados relacional

O modelo relacional encontra-se criado no sistema MySQL e refere-se à primeira base de dados requisitada pelo dono do Zoo. Desta forma, esta possui todas as informações consideradas necessárias para a gestão do Jardim Zoológico, estando todos os dados no formato pretendido. O seu modelo lógico apresenta-se de seguida e permite assim inferir os tipos dos dados.

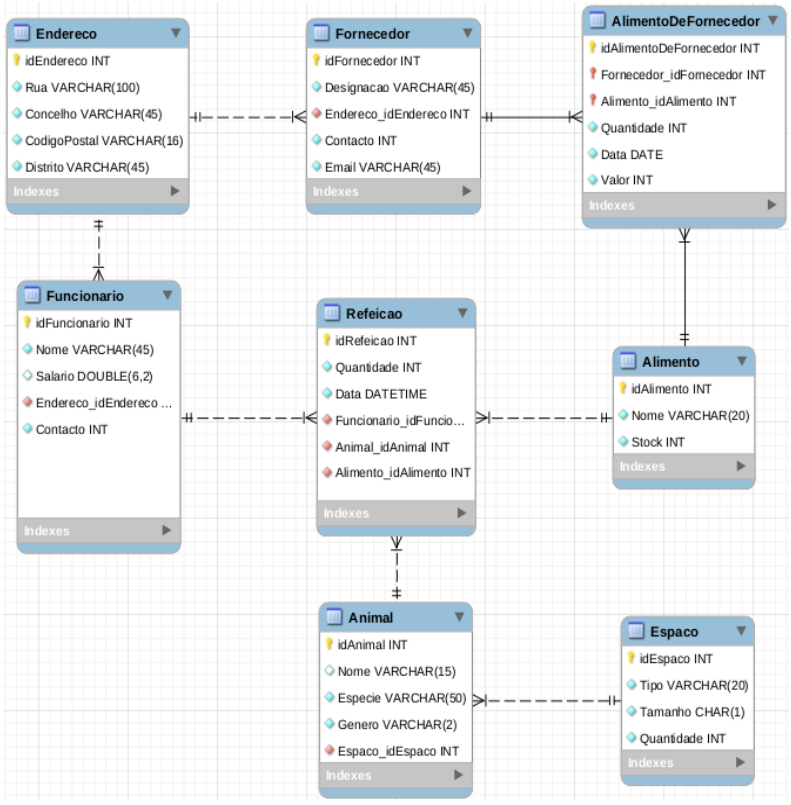


Figura 3 - Modelo Relacional Zoo: 1ªFonte de Informação

De forma a efetuar uma primeira análise efetuamos algumas *queries* que nos permitiram inferir a qualidade dos dados.

idAlimento	Nome	Stock
1	Frango	777
2	Peixinhos	230
3	Frutos	589
4	Bamboo	270
5	Atum	1000
6	Alface	2548

Figura 4 - Análise de dados da tabela Alimento

Problemas podem ocorrer relativamente à inserção do Stock, na medida em que o tipo definido abrange valores negativos, algo pouco coerente na análise dos dados. Nas tabelas de refeição podemos ter problemas relativos à quantidade, sendo que na tabela *alimentodefornecedor* pode acontecer o mesmo, no caso do atributo valor.

5.1.2 Base de dados não relacional

A base de dados não relacional refere-se ao Jardim Zoológico da Maia que resultou de uma parceria efetuada pela família Sousa e por isso contém dados diferentes da BD do modelo relacional. Para além disso, trata-se de uma base de dados relacional pelo que a quantidade de dados é significativamente superior às restantes fontes de informação.

Neste momento, a informação relativa a esta fonte é a seguinte:

Figura 5 - Modelo Não Relacional Zoo: 2ªFonte de Informação

Os problemas desta fonte estão associados aos problemas da fonte anterior, uma vez que esta se trata de uma migração da primeira. No entanto, inferimos também que o tipo de dados, no que diz respeito ao atributo Data difere em cada uma das fontes o que poderá surgir cuidados redobrados nas fases posteriores.

5.1.3 Ficheiro CSV

Existe ainda uma terceira fonte de informação, proveniente da falta de conhecimento sobre as outras duas bases de dados de Manuel Sousa, que optou por guardar informação usando o *Excel* num ficheiro *csv*. No entanto, teve em atenção os valores da base de dados, e com alguns ajustes por parte da equipa, os dados encontram-se claros. Como por exemplo, os dados referentes aos alimentos que Manuel Sousa registou no zoo:

idAlimento	Nome	Stock
7	Carne de Vaca	50
8	Algas	400

Figura 6 - CSV: 3ªFonte de Informação

Inicialmente, Manuel Sousa não considerou certos elementos, como por exemplo o stock que se encontrava escrito por extenso, e não o número inteiro em específico, algo que também foi mudado. Para além disso, não havia uso de identificadores, uma vez que este não considerou ser um aspeto relevante. Estes foram alguns dos ajustes feitos pela equipa de desenvolvimento, permitindo assim que estes sejam mais fáceis de tratar nas fases posteriores.

5.2. Desenvolvimento do esquema de mapeamento de dados - *source-to-target data map*

De forma a melhor compreender a associação entre os dados presentes das fontes de informação e os dados do *Data Warehouse,* foi desenvolvido um *source-to-target data map*.

Target				Source			
Nome da Tabela	Nome do Atributo	Meta Dados	Tipo de Tabela	Base de Dados	Nome da Tabela	Nome do Atributo	Meta Dados
Fornecedor	idFornecedor	Int	Dimensional				
Fornecedor	Designacao	Varchar(45)	Dimensional	Zoo	Fornecedor	Designacao	Varchar(45)
Fornecedor	Contacto	Int	Dimensional	Zoo	Fornecedor	Contacto	Int
Fornecedor	Email	Varchar(45)	Dimensional	Zoo	Fornecedor	Email	Varchar(45)
Fornecedor	idAnterior	Int	Dimensional	Zoo	Fornecedor	idFornecedor	Int
Fornecedor	Source	Int	Dimensional				
Alimento	idAlimento	Int	Dimensional				
Alimento	Stock	Int	Dimensional	Zoo	Alimento	Stock	Int
Alimento	Nome	Varchar(45)	Dimensional	Zoo	Alimento	Nome	Varchar(45)
Alimento	idAnterior	Int	Dimensional	Zoo	Alimento	idAlimento	Int
Alimento	Source	Int	Dimensional				
Refeicao	idRefeicao	Int	Dimensional				
Refeicao	Quantidade	Int	Dimensional	Zoo	Refeicao	Quantidade	Int
Refeicao	Horario	Date	Dimensional	Zoo	Refeicao	Data	Datetime
Refeicao	idAnterior	Int	Dimensional	Zoo	Refeicao	idRefeicao	Int
Refeicao	Source	Int	Dimensional				
Refeicao	Alimento_idAlimento	Int	Dimensional	Zoo	Refeicao	Alimento_idAlimento	Int
HistoricoAlimento	idHistoricoAlimento	Int	Histórico				
HistoricoAlimento	Stock	Int	Histórico	Zoo	Alimento	Stock	Int
HistoricoAlimento	Alimento_idAlimento	Int	Histórico	Zoo	Alimento	idAlimento	Decimal(19,4)
TF-Compra	idTF-Compra	Int	Factos				
TF-Compra	Quantidade	Int	Factos	Zoo	AlimentoDeFornecedor	Quantidade	Int
TF-Compra	Valor	Double	Factos	Zoo	AlimentoDeFornecedor	Valor	Int
TF-Compra	PrecoUnidade	Double	Factos	Zoo	AlimentoDeFornecedor	Vaor/Quantidade	Double
TF-Compra	idAntigo	Int	Factos	Zoo	AlimentoDeFornecedor	idAlimentoDeFornecedor	Int
TF-Compra	Source	Int	Factos				
TF-Compra	Alimento_idAlimento	Int	Factos	Zoo	AlimentoDeFornecedor	Alimento_idAlimento	Int
TF-Compra	Fornecedor_idFornecedor	Int	Factos	Zoo	AlimentoDeFornecedor	Fornecedor_idFornecedor	Int

5.3. Revisão do esquema de mapeamento de dados

Depois da equipa de desenvolvimento ter chegado a um consenso relativo ao mapeamento de dados definidos, procedeu-se à revisão deste com o dono de Zoo, de forma a confirmar que este se encontrava de acordo com o desejado e que os respetivos dados correspondiam de facto aos do *Data Warehouse* desenvolvido. Após aprovação de Manuel Sousa, a equipa deu continuidade ao projeto.

6. Modelação do Sistema de Povoamento

6.1. Apresentação do *Data Warehouse* a povoar

Dando seguimento ao esquema dimensional desenvolvido anteriormente, e tendo em conta as fontes de informação disponíveis, foi desenvolvido o seguinte *Data Warehouse*. Para além disso, é de referir que o esquema desenvolvido vai ao encontro da estrutura em floco de neve.

De realçar ainda que neste esquema que foi necessário acrescentar um atributo *source* que serviu para identificar a origem de cada um dos registos. O *idAntigo* servirá de auxílio para identificar quais dados é que são "novos" em cada uma das fontes. No entanto, será explicado em maior detalhe na secção 7.

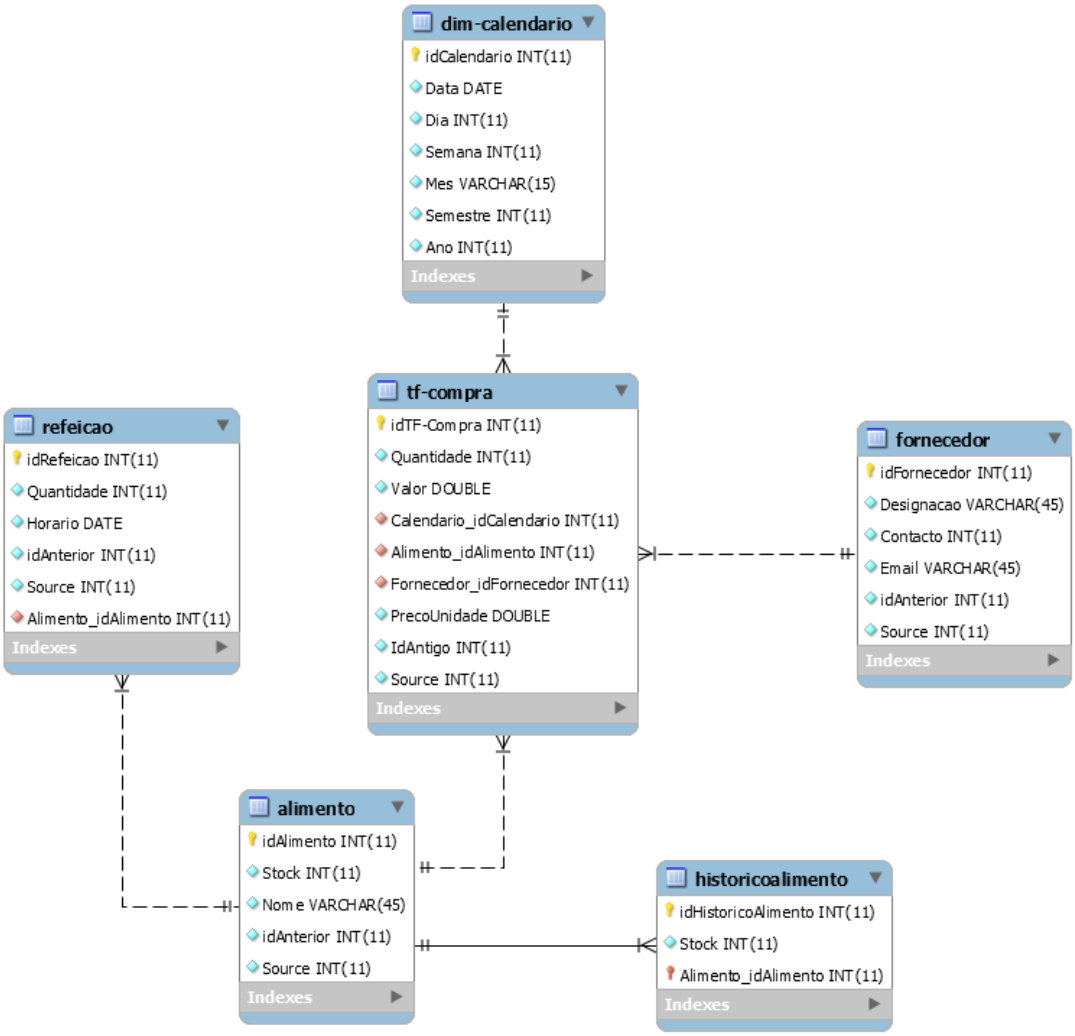


Figura 7 - Modelo lógico dimensional

6.2. Esquematização do esquema conceptual do sistema de povoamento em BPMN

6.2.1 Extração

No processo de ETL, **E**xtração é a primeira fase. Nesta, tal como o nome indica, são extraídos todos os dados necessários de cada uma das fontes de informação, e posteriormente inseridos na área de retenção. Como se pode verificar nas figuras abaixo, a extração de cada fonte ocorre em paralelo.

Figura 8 - Bpmn de extração

6.2.2 Limpeza

Esta fase consiste na análise dos dados que foram extraídos das diferentes fontes de informação e sua correspondente “limpeza”, ou **T**ransformação, isto é, verificação do tipo de dados, para que sejam posteriormente e corretamente inseridos no *Data Warehouse*.

Neste sistema, apenas um atributo deve sofrer alterações, que é a data. Este está presente tanto na Refeição como na Compra. Esta alteração é necessária devido ao diferente tipo de dados de cada uma das fontes de informação. No caso do Neo4j, a data é do tipo *String*, e no MySQL é do tipo *Date*. Para que se converta ambas num só tipo, este atributo passa por esta fase de limpeza, e é inserido nas correspondentes tabelas de auditoria de Limpeza.

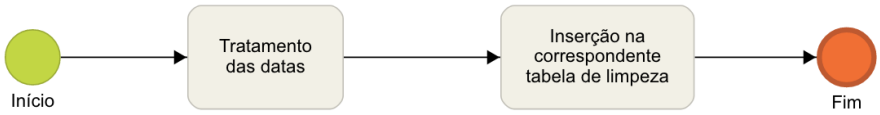


Figura 9 - Bpmn de Limpeza

6.2.3 Conciliação

Nesta fase são preparados os dados provenientes das diferentes fontes de dados para que, numa fase posterior, sejam carregados para o *Data Warehouse.* Devido à diversidade das fontes de informação, é necessário conciliar todos os tipos de dados, nomeadamente o caso em que duas fontes de informação podem utilizar a mesma chave para a mesma entidade. Dito isto, e para que seja mantida a integridade da informação, são geradas chaves de substituição para as tabelas Alimento, Fornecedor, Refeição e também para a tabela de factos.

No exemplo abaixo, temos o processo de conciliação para o caso em que haja uma inserção de um novo alimento. Primeiramente, é verificado se esse registo já existe. Caso afirmativo, é inserido nas tabelas de Quarentena. Caso contrário, é gerado um novo identificador, para que não haja perda de informação. Em seguida, é carregado para a tabela de conciliação correspondente.

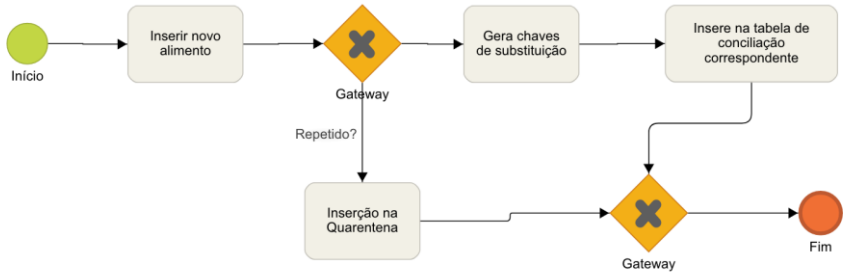


Figura 10 - Bpmn conciliação: Alimento, Inserção

As dimensões com variação que este sistema contém são o Fornecedor, Alimento, Refeição e Compra, o que significa que dados existentes nestas tabelas possam vir a ser atualizados. O procedimento, numa atualização de um determinado registo, é muito semelhante ao da inserção, tal como se pode observar na imagem abaixo.

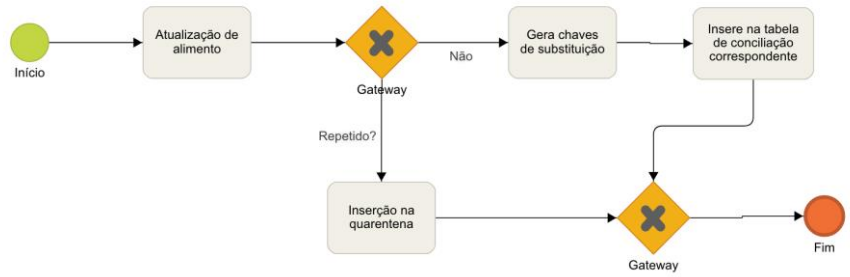


Figura 11 - Bpmn conciliação: Alimento, Atualização

Relativamente à tabela correspondente à *Compra,* antes de nesta serem inseridos os dados, é elaborado o cálculo do preço por unidade, que é relativo ao alimento. Este cálculo é feito pela divisão entre o valor final e a quantidade de alimento que foi adquirido. De seguida os dados são inseridos na tabela *AR-ConciliacaoCompra.*

Figura 12 - Bpmn conciliação: Compra

6.2.4 Carregamento

Uma vez realizadas todas as fases anteriores (Extração, Limpeza e Conciliação), segue-se o processo final do carregamento, correspondente ao último processo do ETL. Este carrega todos os dados presentes nas pré-dimensões para o *data warehouse*. O processo apresenta alguma simplicidade, pois não há necessidade de efetuar algum tipo de operação, como por exemplo limpeza, devido à igualdade do tipo de dados.

Visto que a dimensão *Alimento* apresenta variação com histórico, os tratamentos das atualizações dos dados ficam ao cargo do próprio *data warehouse*. Quando um alimento é recebido, primeiramente é verificado se se trata de uma atualização do mesmo. Caso afirmativo, insere também informação no histórico e posteriormente na dimensão alimento, caso contrário, apenas guarda os dados na dimensão, terminando assim o processo.

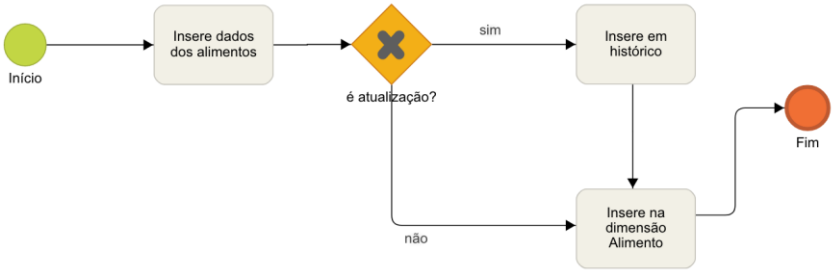


Figura 13 - Bpmn Carregamento

6.3. Descrição detalhada do sistema de povoamento

Relativamente ao sistema de povoamento tivemos em consideração dois momentos: carregamento inicial e regular.

No que diz respeito ao carregamento inicial, deverá extrair-se os dados para as áreas de extração desenvolvidas. De seguida, ocorrera a transformação. Nesta fase os dados passaram por uma série de processos. Nesse sentido entre as várias operações é de referir limpeza dos dados efetuados, e o carregamento destes nas tabelas de conciliação. Na limpeza poderão existir tabelas de quarentena de forma a identificar dados inconsistentes que não podem tratados de forma automática. De realçar que também deverá ser efetuado o tratamento das chaves, uma vez que estas poderiam ficar inconsistentes, por terem origem em diversas fontes independentes. Para tal,

deverá existir tabelas de correspondência e de *lookup*. Por fim, deverá ser efetuado o carregamento dos dados para o modelo dimensional desenvolvido, ou seja, para o *Data Warehouse*.

Já no que se refere ao carregamento regular, é necessário definir uma estratégia para captar os dados ao longo do tempo. Em síntese, existem dois métodos que poderíamos implementar. A primeira abordagem consiste em inserir todo o universo de dados sempre que for necessário atualizar o *Data Warehouse*, a segunda abordagem consiste apenas em efetuar (depois de efetuar uma inserção inicial dos dados) inserções que sejam apenas referentes a atualizações. Esta segunda abordagem é designada de incremental, sendo que pelo facto da primeira ser bastante desvantajosa em termos de desempenho, foi o método pelo qual optamos. Neste sentido, esta técnica obriga a que existam algumas alterações para que seja possível verificar quais são os dados considerados novos e quais é que não são. O processo mais adequado seria então a utilização de *procedures, trigger, timestamps* de forma a identificar registos antigos. Depois de definidos estes factos, o seguinte passo é o da transformação de conciliação tendo então em conta a atualização das diversas chaves. No que diz respeito ao povoamento do *Data Warehouse*, é ainda ter necessário ter em conta os atributos variáveis, e o correto carregamento da dimensão histórico, existente para o atributo de Stock. Nesta fase pensamos ainda em implementar um sistema de recuperação que tornasse o povoamento tolerante a falhas.

6.4. Descrição e caracterização de todos os elementos de dados (auditoria, *lookup*, quarentena, etc.) da Área de Retenção utilizados para suporte ao povoamento

Para se poder realizar as ações de Extração, Transformação e Carregamento da forma exposta nos diagramas de BPMN, é necessário estabelecer as tabelas da Área de Retenção utilizadas como auxílio ao povoamento do *Data Warehouse*.

6.4.1 Extração

Começando-se pelo processo da Extração das fontes, torna-se necessário criar uma tabela para cada uma das dimenções (Fornecedor, Alimento e Refeição) e para a tabela de factos (Compra), para assim se obter os dados de cada uma destas, sendo

colocado nestas tabelas apenas os dados relevantes para o *Data Warehouse.* Dado isto, e tendo-se então 3 fontes distintas, é necessário criar as 4 tabelas mencionadas para cada fonte, devido ao facto de os dados poderem ter tipos diferentes entre as diversas tes fontes. Sendo assim, o processo de Extração possui 12 tabelas para extrair assim os dados do MySQL, Neo4J e CSV.

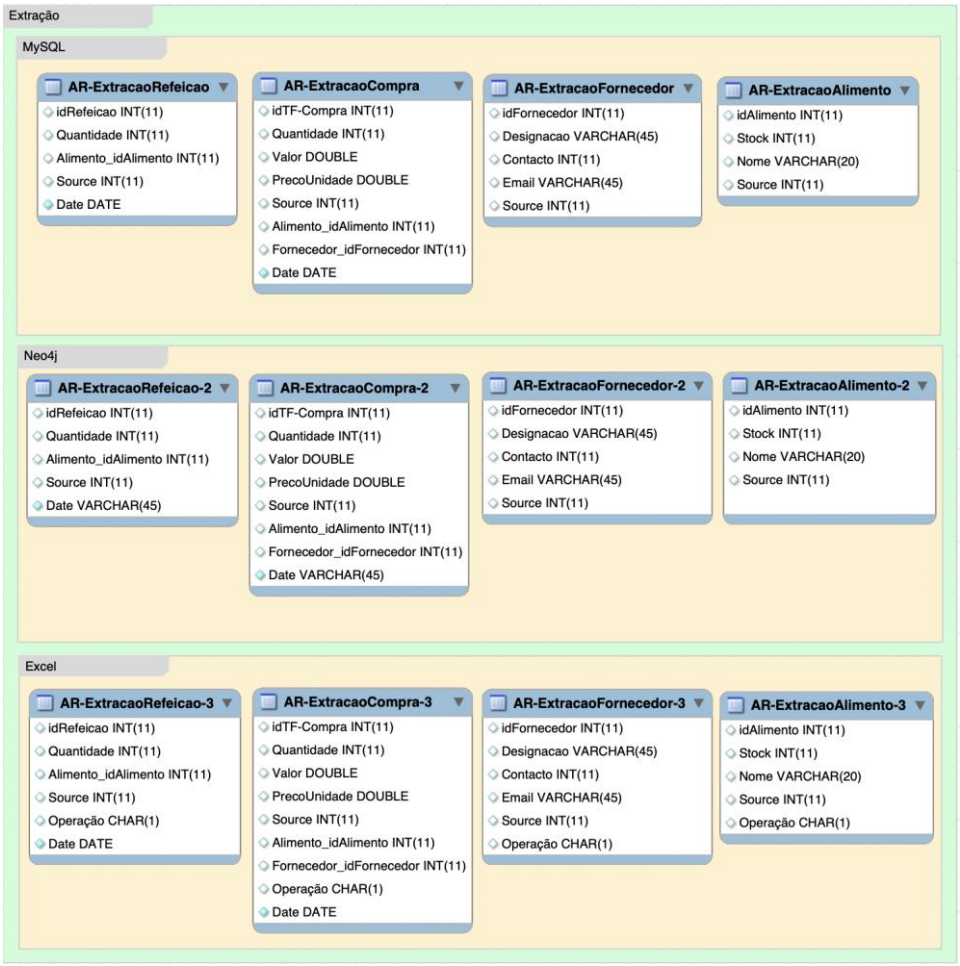


Figura 14 - Área de Retenção: Extração

6.4.2 Limpeza

Passando-se para o processo de Transformação, e tendo em conta o definido nos diagramas de BPMN associados à Limpeza, ir-se-á, nesta fase, tratar dos dados que, por exemplo, possam conter tipos diferentes entre as fontes e o *Data Warehouse.* Desta forma, torna-se necessário tratar destes casos para as 3 dimensões e para a tabela de factos, obtendo-se assim 4 tabelas para cada um destes casos. Novamente, pelo facto de haver 3 fontes, tem-se 12 tabelas para a limpeza.

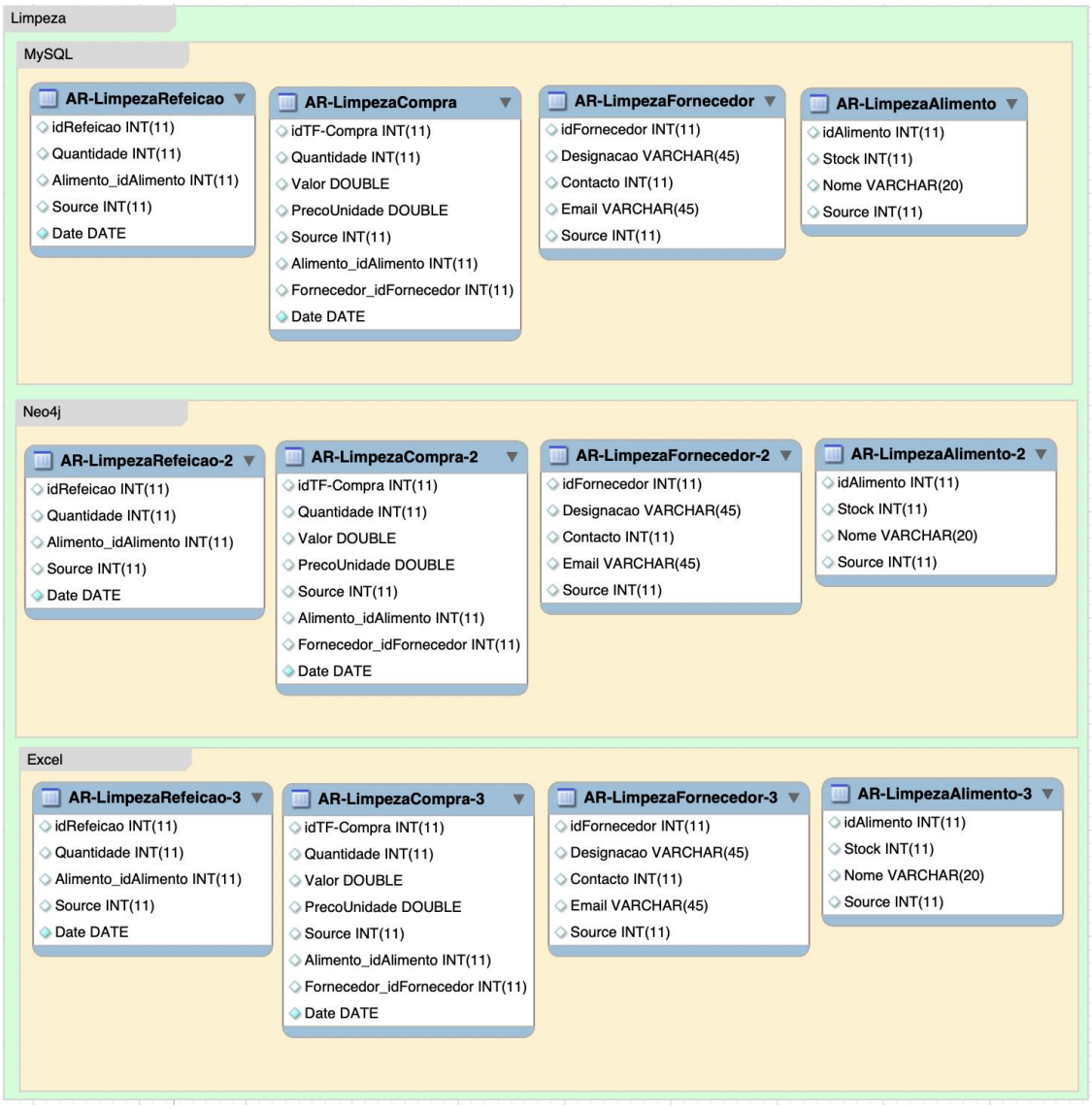


Figura 15 - Área de Retenção: Limpeza

6.4.3 Conciliação

No processo de Conciliação os dados já se encontram tratados, isto é, as questões de, por exemplo, tratamento de tipos, chaves, cálculo de atributos e updates, já se encontram resolvidas. Desta forma, é possível fazer a junção de todos os dados presentes nas 3 fontes, obtendo-se assim uma única tabela para Alimento, Fornecedor, Refeição e Compra. Após a Conciliação, é apenas necessário passar estes dados para o *Data Warehouse*.

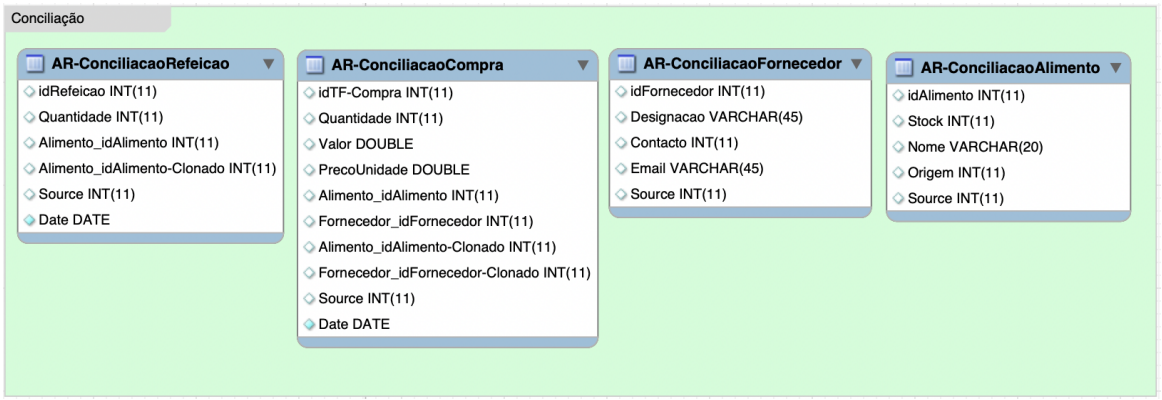


Figura 16 - Conciliação

6.4.4 Quarentena

Na Quarentena são colocados os dados considerados que devem ser o administrador a analisar e tratar manualmente, e não programa de forma automática. Como os dados já estão tecnicamente limpos, estes podem ser agrupados numa única tabela por dia.

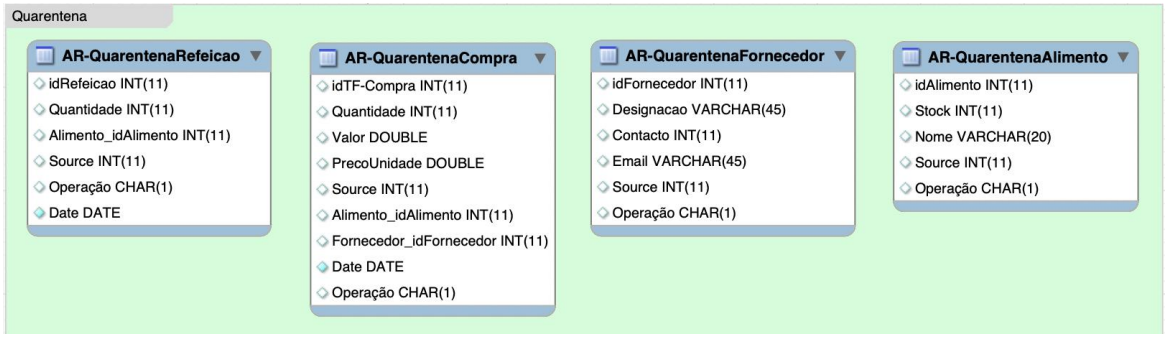


Figura 17 - Área de Retenção: Quarentena

6.4.5 Correspondência e *Lookup*

As tabelas de Correspondência e *Lookup* são puramente auxiliares para o processo de tratamento das chaves estrangeiras, desta forma, estas são usadas sobre a Compra (chaves estrangeiras de Alimento e Fornecedor) e a Refeição (chave estrangeira de Alimento), associando a chave “verdadeira” com a chave gerada para o *Data Warehouse* (Correspondência), podendo assim verificar quais chaves a modificar na tabela Compra e Refeição (*Lookup*).

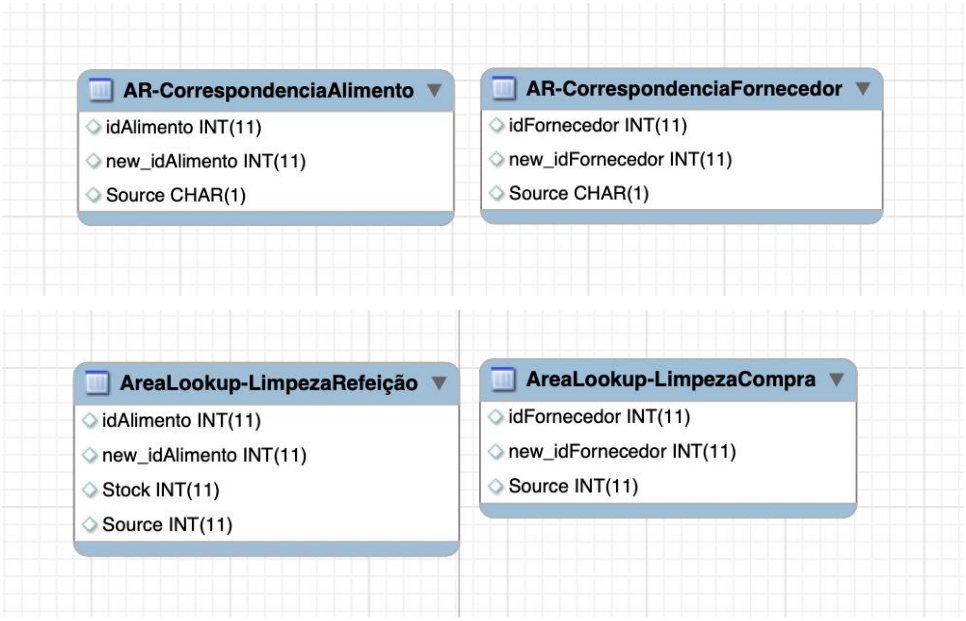


Figura 18 - Área de Retenção: Correspondência, *Lookup*

7. Implementação do Sistema de *Data Warehousing*

7.1. Escolha das plataformas computacionais

O software que utilizamos foi desenvolvido pela *Pentaho* e tem a designação de Kettle. Trata-se de um software em código aberto que permite efetuar a integração de dados, suportando assim, todo o processo de ETL. Trata-se de uma ferramenta bastante útil na medida em que facilita uma elevada quantidade de operações, como será o caso, por exemplo, da extração de informação contida em diversas fontes heterogénea.

Em suma, por todas estas razões e principalmente pelo facto de ser uma ferramenta completa e totalmente gratuita que vai ao encontro a todas as necessidades do projeto, foi a escolhida pelo grupo para a implementação de todo o processo.

7.2. Implementação dos esquemas físicos dos sistemas de dados

O MySQL possibilita, após os modelos lógicos estarem definidos, a utilização da opção Forward Engineer. Este comando transforma o modelo lógico desenvolvido em físico, facilitando por isso a implementação física duma base de dados ao gerar automaticamente o código SQL para criação de todas as tabelas necessárias.

7.3. Implementação do Sistema de Povoamento

Foram utilizados dois *jobs* principais em Kettle que permitiram tornar mais intuitivo todo este processo.

O primeiro diz respeito ao processo inicial e apresenta-se de seguida:

Figura 19 - Job, processo inicial

O segundo, por sua vez, diz respeito ao processo regular e de forma global é semelhante ao inicial, mudando apenas intrinsecamente o conteúdo de cada um dos passos. Sendo assim, pode-se dizer que cada um dos blocos, que se apresentam na imagem do processo inicial, representam um conjunto de operações realizadas com o objetivo de atingir o fim pretendido. Desta forma, insere-se de seguida um pequeno exemplo de uma dessas transformações em Kettle.

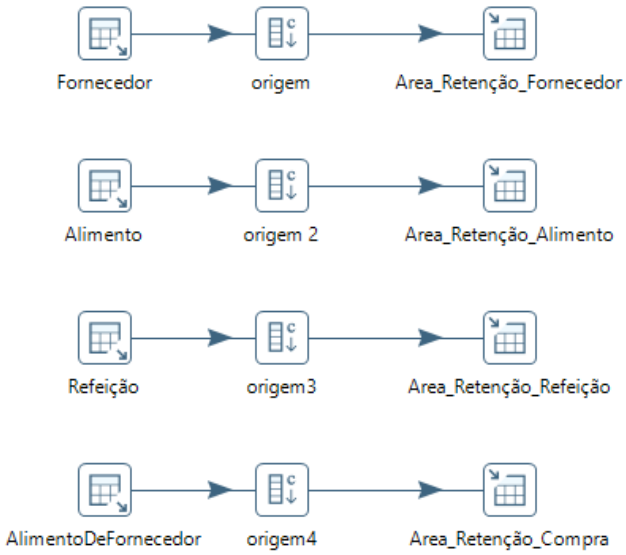


Figura 20 – Processo de Carregamento MySQL

A maior diferença entre o processo inicial e o regular surge então na diferença que ocorre na etapa de transformação de cada um destes processos. Assim a transformação inicial é dada por:

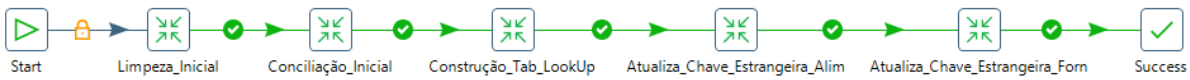


Figura 21 - Job, Processo de Transformação Inicial

Enquanto que a transformação regular, apresenta uma maior complexidade devido ao tratamento que decorre do processo de se ter de efetuar apenas as atualizações necessárias no *Data Warehouse*, sendo representada por:

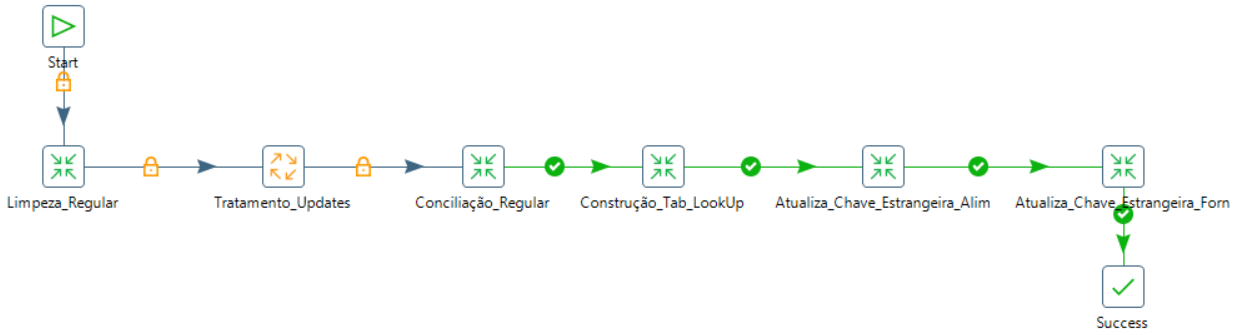


Figura 22 - Job, Processo de Transformação Regular

O grupo optou assim por guardar no modelo dimensional os id's "antigos", ou seja, os ids que se encontravam armazenados nas diversas fontes. No entanto, esta foi uma simplificação tomada a nível académico pelo que no âmbito profissional seria mais correto a utilização de técnicas como é o caso de *procedures*, *timestamps*, *triggers* entre outras.

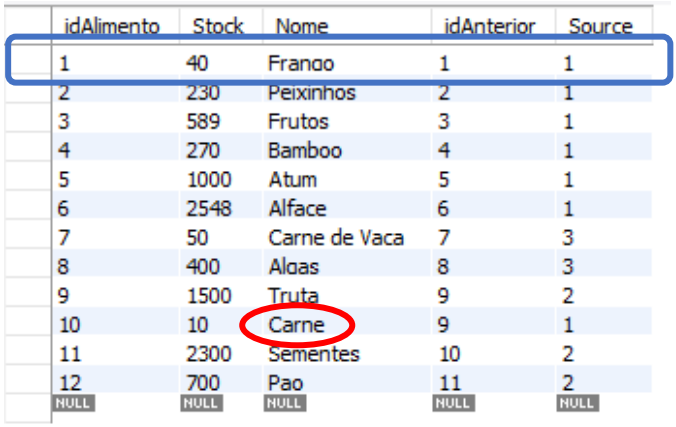
Relativamente a este povoamento é importante referir que houveram alguns desvios no que diz respeito ao que foi modelado na fase posterior em bpmn. Alguns do processo de transformação não se encontram pela ordem suposta. Outro facto de desvio foi a não implementação das tabelas de quarentena.

Para além disso, consideramos, apesar de não termos implementado, mecanismos de backup. Estes serviriam para caso ocorresse em algum momento uma falha, o sistema fosse tolerante e como tal, fosse capaz de responder a esse acontecimento. No entanto, sendo que os jobs se tratam de processos independentes, se ocorrer uma falha em cada uma das fases é possível retornar o povoamento a partir desse ponto.

7.4. Análise da execução do sistema de povoamento

De forma a procurar perceber se o sistema de povoamento se encontrava correto, efetuamos alguns testes. Desta forma tivemos em consideração o que seria esperado e comparamos com os resultados obtidos.

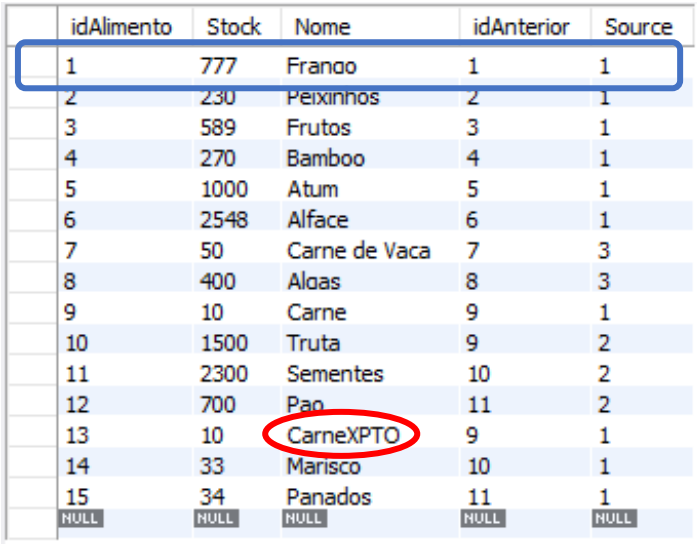
Assim, começamos por testar o carregamento inicial. Verificamos então que o carregamento no sistema de *Data Warehouse* estava a ocorrer corretamente e sem qualquer tipo de inconsistência. Apresenta-se de seguida o resultado obtido para a dimensão alimento.



idAlimento	Stock	Nome	idAnterior	Source
1	40	Frango	1	1
2	230	Peixinhos	2	1
3	589	Frutos	3	1
4	270	Bamboo	4	1
5	1000	Atum	5	1
6	2548	Alface	6	1
7	50	Carne de Vaca	7	3
8	400	Algas	8	3
9	1500	Truta	9	2
10	10	Carne	9	1
11	2300	Sementes	10	2
12	700	Pao	11	2
NULL	NULL	NULL	NULL	NULL

Figura 23 - Verificação dos resultados: Processo Inicial

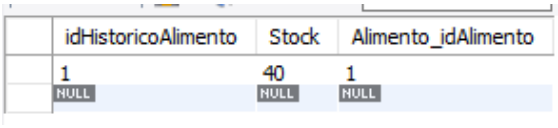
De seguida foi necessário testar se o carregamento regular se encontrava correto. Para o efeito, antes de "correr" o respetivo *job* inserimos novos elementos e atualizamos outros para verificar se as alterações efetuadas sortiam efeito no *Data Warehouse*. De facto, o processo da forma esperada.



idAlimento	Stock	Nome	idAnterior	Source
1	777	Frango	1	1
2	230	Peixinhos	2	1
3	589	Frutos	3	1
4	270	Bamboo	4	1
5	1000	Atum	5	1
6	2548	Alface	6	1
7	50	Carne de Vaca	7	3
8	400	Algas	8	3
9	10	Carne	9	1
10	1500	Truta	9	2
11	2300	Sementes	10	2
12	700	Pao	11	2
13	10	CarneXPTO	9	1
14	33	Marisco	10	1
15	34	Panados	11	1
NULL	NULL	NULL	NULL	NULL

Figura 24 - Verificação dos resultados: Processo Regular

É ainda de realçar o correto povoamento de dimensão stock, com variação e com história.



idHistoricoAlimento	Stock	Alimento_idAlimento
1	40	1
NULL	NULL	NULL

Figura 25 - Verificação dos resultados: Processo Regular

7.4.1 Indicadores de análise de dados

De forma a fornecer ao administrador do Zoo uma melhor forma de interpretar os dados contidos no Data Warehouse foram desenvolvidos *dashboards* sobre estes, usando a ferramenta *Power BI*.

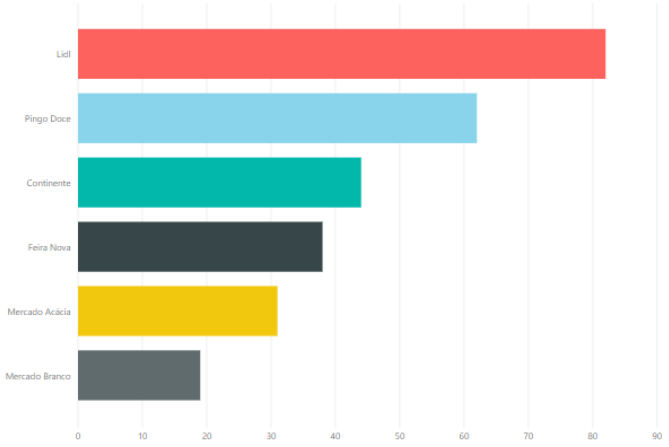


Figura 26 – Gráfico Venda Fornecedor

Este gráfico demonstra quais os fornecedores que mais venderam ao zoo, sendo visível que o Lidl tem sido o maior fornecedor, até hoje.

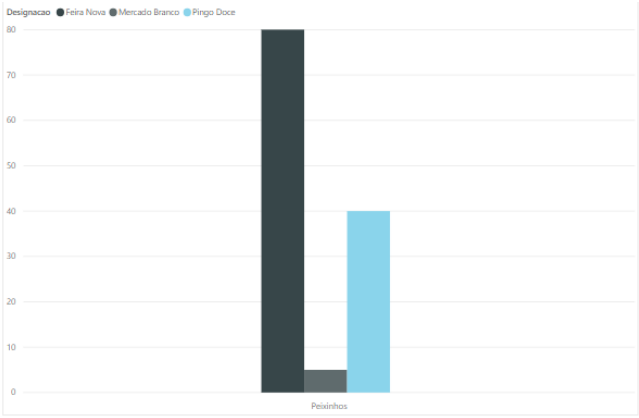


Figura 27 - Gráfico Venda Alimento

Este gráfico demonstra quais os fornecedores que venderam determinado alimento, neste caso Peixinhos, e qual a quantidade.

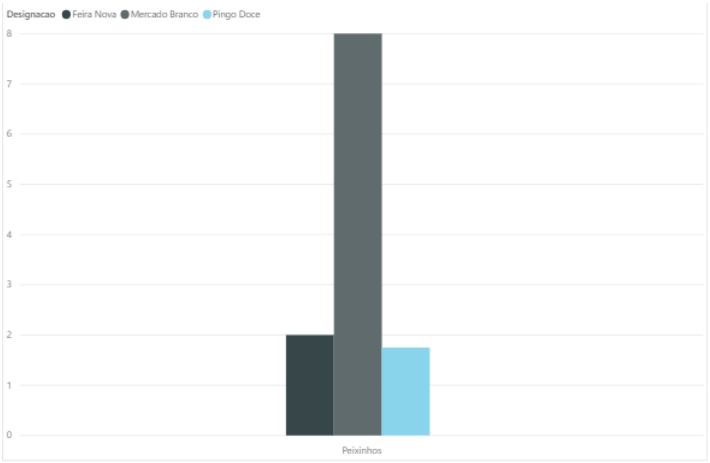


Figura 28 – Gráfico Preço Unidade Alimento

Esta imagem demonstra quais os preços por unidades de determinado alimento, neste caso Peixinhos, por fornecedor.

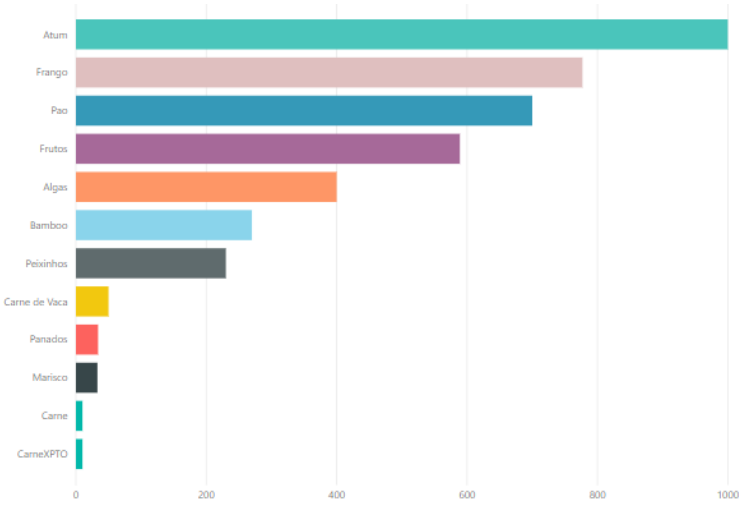


Figura 29 – Gráfico Stock de Alimentos

Este gráfico demonstra o stock de todos os alimentos, que seja menor de 1000 unidades.

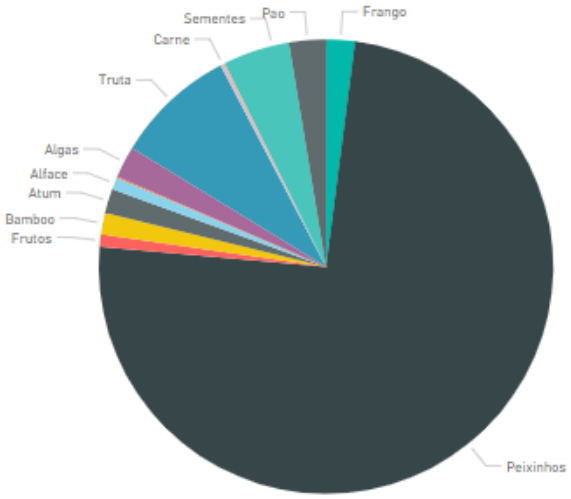


Figura 30 - Gráfico Alimentos mais consumidos

Este gráfico demonstra quais os alimentos mais consumidos nas refeições dos animais.

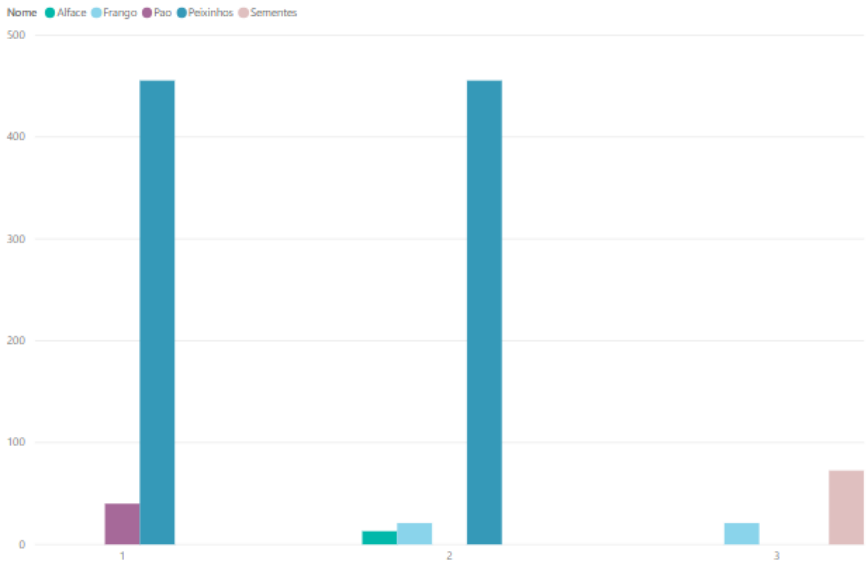


Figura 31 - Gráfico alimentos consumidos

Este gráfico demonstra qual a quantidade média dos alimentos consumida nas primeiras semanas.

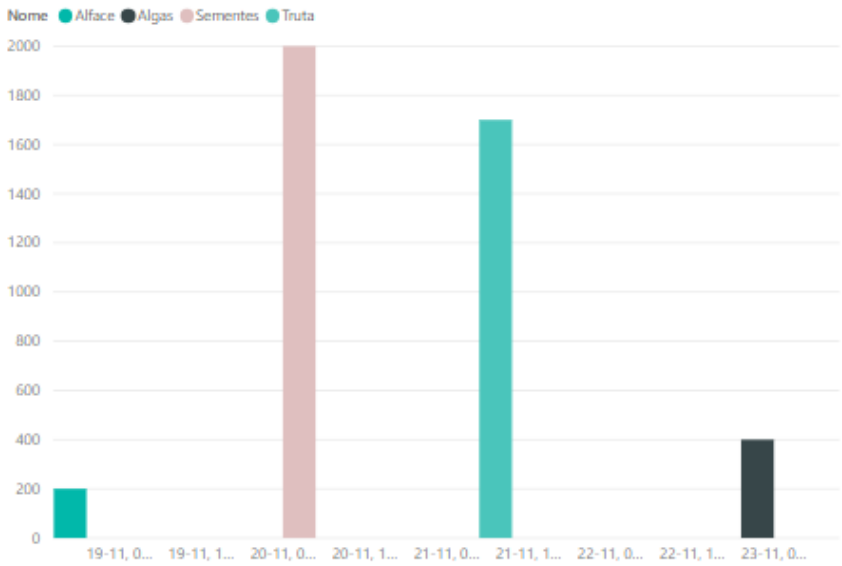


Figura 32 - Gráfico quantidade de produtos comprados

Este gráfico demonstra qual a quantidade de produtos comprados num intervalo de datas.

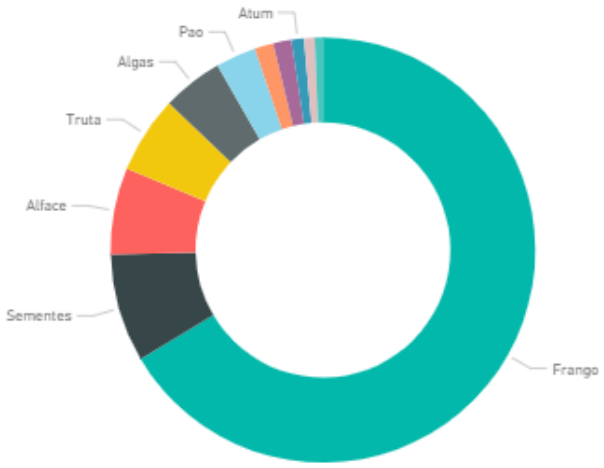


Figura 33 - Gráfico Gastos no Zoo

Este gráfico demonstra uma relação dos valores gastos pelo zoo por alimentos.

8. Conclusões e Trabalho Futuro

De acordo com o principal objetivo podemos concluir que este foi alcançado. De facto, foi possível obter um sistema Data Warehouse completamente funcional.

Por um lado, é importante referir que o planeamento do trabalho foi bem delineado (com a ajuda do Diagrama de Gantt) tendo a equipa de desenvolvimento cumprido todas as datas estipuladas. As reuniões de grupo correram conforme planeado, conseguindo sempre adiantar ao máximo o trabalho prático. Para além disso, foi efetuado uma contextualização bastante coerente, com as medidas de sucesso corretamente definidas. Ao utilizar o método dos 4 passos desenvolvido por Kimball, a estruturação do trabalho tornou-se mais simples e propícia a menos erros durante a sua conceção. O desenvolvimento deste *Data Warehouse* foi um processo bastante trabalhoso, que precisou de uma constante monitorização.

Por outro lado, o principal desafio foi então na implementação. O facto de ser um processo completamente novo levou a algumas inconsistências. É importante realçar que a utilização de uma ferramenta nova levou a alguma divergência daquilo que foi implementado e o esperado, na medida em que foram necessárias efetuar algumas simplificações. Nesse sentido, foi necessário retroceder em algumas fases para conseguir um trabalho final de acordo com o necessitado.

Em suma, todo o trabalho foi bastante útil na medida em que permitiu ter um contacto mais profundo com o conceito de Data Warehousing, e consequentemente com o processo de ETL. No entanto, uma pequena retrospeção efetuada pelo grupo permitiu ainda concluir que num posterior trabalho futuro, devido à experiência adquirida, teríamos o trabalho bastante mais facilitado.

Referências

- Golfarelli, M., Rizzi, S., Data Warehouse Design: Modern Principles and Methodologies;

- Kimball, R., Reeves, L., Ross, M., Thornthwait, W., The Data Warehouse Lifecycle Toolkit – Pratical Techiniques for Building Data Warehouse and Business Intelligence Systems;

Lista de Siglas e Acrónimos

BD - Base de Dados

DW - *Data Warehouse*

AR - Área de Retenção