

Agradecimentos	4
Resumo	5
1. Introdução.....	6
1.1 Contextualização	6
1.2 Motivação e Objetivo.....	6
1.3 Estrutura do Relatório	8
2. Revisão de conceitos básicos de BI e DW.....	9
2.1 Business Intelligence (BI).....	9
2.1.1 OLAP.....	10
2.1.2 <i>Scorecard</i> e <i>Dashboard</i>	11
2.1.3 Vantagens no contexto do projeto.....	12
2.2 Data Warehouse (DW)	13
2.3 Modelo Dimensional	15
2.3.1 Tabelas de Dimensão	15
2.3.2 Tabelas de Factos	15
2.3.3 Métrica	17
2.4 Arquitetura para Data Warehouses	17
2.5 Dimensões "Role-Playing" e "Slowly Changing"	21
3 Tecnologias Utilizadas	24
4 Desenvolvimento do Projeto.....	27
4.1 Análise da Instituição	27
4.2 Arquitetura proposta	27
4.3 Levantamento de Requisitos	29
5 Modelação Dimensional	32
5.1 Processo de Negócio	32
5.2 Definição de Granularidade	32
5.3 Identificação das dimensões e sua caracterização	33
5.3.1 Dimensão Data.....	33
5.3.2 Dimensão Minuto.....	35
5.3.3 Dimensão Idade	36
5.3.4 Dimensão Utente.....	36
5.3.5 Dimensão Local	37
5.3.6 Dimensão Causa.....	37
5.3.7 Dimensão Prioridade	38
5.3.8 Dimensão Origem	38
5.3.9 Dimensão Destino.....	38

5.3.10 Dimensão Entidade Financeira Responsável (EFR)	39
5.3.11 Dimensão Elegibilidade	39
5.3.12 Dimensão Serviço	39
5.3.13 Utilização de Dimensões Degeneradas	39
5.4 Identificação de factos	40
5.5 Matriz de barramento dimensional	42
6 Processo de implementação de um Data Mart	45
6.1 Extração, Transformação e Carregamento	45
6.2 Processo de ETL das Dimensões	46
6.2.1 Dimensão Data.....	46
6.2.2 Dimensão Minuto.....	47
6.2.3 Dimensão Idade	48
6.2.4 Dimensão Utente.....	49
6.2.5 Dimensão Local	56
6.2.6 Dimensão Causa.....	59
6.3 Carregamento da Tabela de Factos	59
6.4 Cubo.....	62
6.5 Métricas Derivadas	65
6.6 KPI's.....	65
7 Apresentação de Resultados	67
7.1 Validações	67
7.2 Período Horário	69
7.3 Prioridade.....	70
8 Conclusão e Trabalho Futuro.....	72
Conclusão	72
Trabalho Futuro	73
Referências	74
Anexo 1 – Scripts	78
Anexo 1.1 - Carregamento da Dimensão Data.	78
Anexo 1.2 - Carregamento dos Feriados Dimensão Data.	79
Anexo 1.3 - Carregamento da Dimensão Minuto.	81
Anexo 1.4 - Carregamento da Dimensão Idade.....	82
Anexo 2 - Dicionário das Dimensões e ETL	83
Anexo 2.1 - Dimensão Utente.....	83
Anexo 2.2 - Dimensão Local.	88
Anexo 2.3 - Dimensão Causa.....	91

Anexo 2.4 - Dimensão Serviço.	93
Anexo 2.5 – Dimensão Prioridade.....	97
Anexo 2.6 – Dimensão Origem.	100
Anexo 2.7 - Dimensão Destino.....	103
Anexo 2.8 - Dimensão Entidade Financeira Responsável.	105
Anexo 2.9 – Dimensão Elegibilidade.	109
Anexo 3 – Dicionário da Tabela de Factos e ETL	112

Agradecimentos

Em primeiro lugar gostaria de agradecer à minha orientadora Prof. Ana Paula Tomás e à Dra. Andreia Pereira pelo apoio, acompanhamento e disponibilidade que foi sempre demonstrado e determinante na realização deste projeto. Estou-lhes profundamente grato.

Agradeço ao Dr. Tavares, Diretor do Departamento de Estatísticas da ARS-N, ao Dr. João Silva e em particular ao meu orientador da ARS-Norte o Dr. Castanheira pela oportunidade concedida e o apoio que manifestaram.

Gostaria também de agradecer aos meus colegas de trabalho, em particular ao Bruno pelo apoio e auxílio dado.

Por último, gostaria de agradecer aos meus amigos e à minha família, em especial um muito obrigado aos meus pais e aos meus irmãos Patrick e Daniel.

Resumo

Nos dias que correm a geração de grandes volumes de dados por parte das instituições é uma realidade. Embora dotadas de complexos sistemas de registos que possibilitam o suporte das suas atividades diárias, o grande volume de dados potencia a necessidade e a obtenção de decisões informadas. Este tipo de desafio não escapa à realidade dos hospitais, onde o crescimento de informação resultante do registo da sua operacionalidade diária é constante.

Com este relatório apresentamos o projeto realizado no âmbito de um estágio de mestrado, que decorreu na Administração Regional de Saúde do Norte (ARS-N), que consistiu no desenvolvimento de um Sistema de Business Intelligence (SBI) aplicado à atividade diária das urgências do Hospital de Braga, com a finalidade de melhorar os processos de tomada de decisão dos gestores da ARS.

1. Introdução

1.1 Contextualização

Este relatório descreve o trabalho realizado no âmbito de um estágio de mestrado que decorreu na Administração Regional de Saúde do Norte (ARSN). O projeto consistiu no desenvolvimento de um Sistema de Business Intelligence (SBI) para o tratamento dos dados das urgências que a ARS-N, como Entidade Pública Contratante (EPC) da Parceria Pública Privada (PPP) do Hospital de Braga, recebe mensalmente.

Mensalmente, o Departamento de Estudos e Planeamento da ARS-N tem de recolher e responder a um conjunto de necessidades de informação dos diversos hospitais da zona norte (incluindo o Hospital de Braga). Neste projeto, procurou-se responder de forma paralela às suas necessidades.

Este estágio teve como abordagem a implementação de um sistema de BI para um serviço concreto, o das urgências. Contudo, pretende-se que projeto tenha continuidade e que mais serviços sejam acrescentados, permitindo à ARS-N efetuar uma análise de todos os serviços de que é responsável de forma global e única. Assim, pretende-se que este projeto se realize de forma a acautelar conformidade e generalidade suficiente para que seja possível a continuidade futura de forma segura e consistente.

Este tipo de infraestruturas de BI requer uma visão integrada dos dados. A integração destes com ferramentas de consulta, exploração e análise de informação fornece suporte a aplicações que permitem apoiar a tomada de decisão.

1.2 Motivação e Objetivo

Nos dias que correm a geração de grandes volumes de dados por parte das instituições é uma realidade. Embora dotadas de complexos sistemas de registos que possibilitam o suporte das suas atividades diárias, o grande volume de dados potencia a necessidade e a obtenção de decisões informadas. Este tipo de desafio não escapa à realidade dos hospitais, onde

o crescimento de informação resultante do registo da sua operacionalidade diária é constante.

Tendo a ARS-N de analisar e validar as regras estabelecidas com a Entidade Gestora do Estabelecimento (EGEST), são-lhe disponibilizados os dados dos registos das atividades dos diferentes serviços do Hospital de Braga. É sobre esse volume de dados crescente e recebido mensalmente que surge a necessidade por parte da ARS-N, de dispor de um sistema de Business Intelligence. Tal sistema possibilitará uma resposta aos problemas e desafios inerentes a uma gestão hospitalar de forma mais célere, consistente e fundamentada.

Atendendo ao contexto descrito e ao facto da ARS-N ter como missão garantir à população da Região Norte o acesso à prestação de cuidados de saúde, foi-nos possível elencar como principais motivações deste projeto os pontos que se seguem:

- Dificuldade de validação dos registos de dados das atividades dos diferentes serviços;
- Necessidade de obtenção de meios de apoio e suporte à tomada de decisão de forma fundamentada, estruturada e célere;
- Complexidade em cruzar dados que permitam avaliar a operacionalidade dos diferentes serviços hospitalares.

Assim, os objetivos deste projeto foram:

- Perceber as principais variáveis existentes na fonte de dados disponibilizada (Excel);
- Obter uma validação dos dados credíveis, identificando e quantificando potenciais faltas de coerência;
- Identificar potenciais riscos e desafios inerentes à implementação de um modelo dimensional;
- Implementar um Data Mart em conformidade com os restantes serviços do hospital;
- Criar e implementar um processo de extração, transformação e carregamento de dados de forma automatizada;

- Criar um modelo de análise de dados integrados que permita o apoio à tomada de decisão;
- Definir um conjunto de indicadores administrativos, clínicos e financeiros de qualidade;
- Disponibilizar aos utilizadores finais um conjunto de *dashboards* que respondam a um conjunto de necessidades estabelecidas;
- Validar o sistema implementado junto dos utilizadores finais, para aferir se satisfaz os objetivos pretendidos.

1.3 Estrutura do Relatório

Para além da introdução (Capítulo 1) onde consta a contextualização, motivação e objetivo do projeto, este relatório está estruturado em mais sete capítulos:

- Capítulo 2: são apresentados alguns conceitos básicos de Business Intelligence (BI) e Data Warehousing (DW).
- Capítulo 3: são apresentadas as tecnologias utilizadas neste projeto.
- Capítulo 4: é feita uma análise da instituição, apresentada a arquitetura do DW proposta para este projeto e descritos os requisitos do projeto.
- Capítulo 5: é descrito o processo de implementação do Data Mart (DM).
- Capítulo 6: é descrito o desenvolvimento do Data Mart e o seu processo de ETL.
- Capítulo 7: é apresentada uma breve demonstração dos resultados.
- Capítulo 8: é descrita a conclusão final e o trabalho futuro.

2. Revisão de conceitos básicos de BI e DW

2.1 Business Intelligence (BI)

Business Intelligence (BI) é um termo genérico que representa uma variedade de ferramentas de software que permitem a análise estratégica dos dados de uma organização. Estes sistemas permitem a uma empresa recolher, armazenar, aceder e analisar os dados de modo a apoiar a tomada de decisões [5,6].

Embora já anteriormente utilizado, este conceito surgiu em 1989 popularizado por Howard Dresner que o definiu como um conceito "guarda-chuva" que abrange um conjunto de técnicas e métodos que possibilitam uma análise mais abrangente e fundamentada [7].

A figura 2.1 ilustra esquematicamente um sistema de BI, decompondo-o em dois processos.

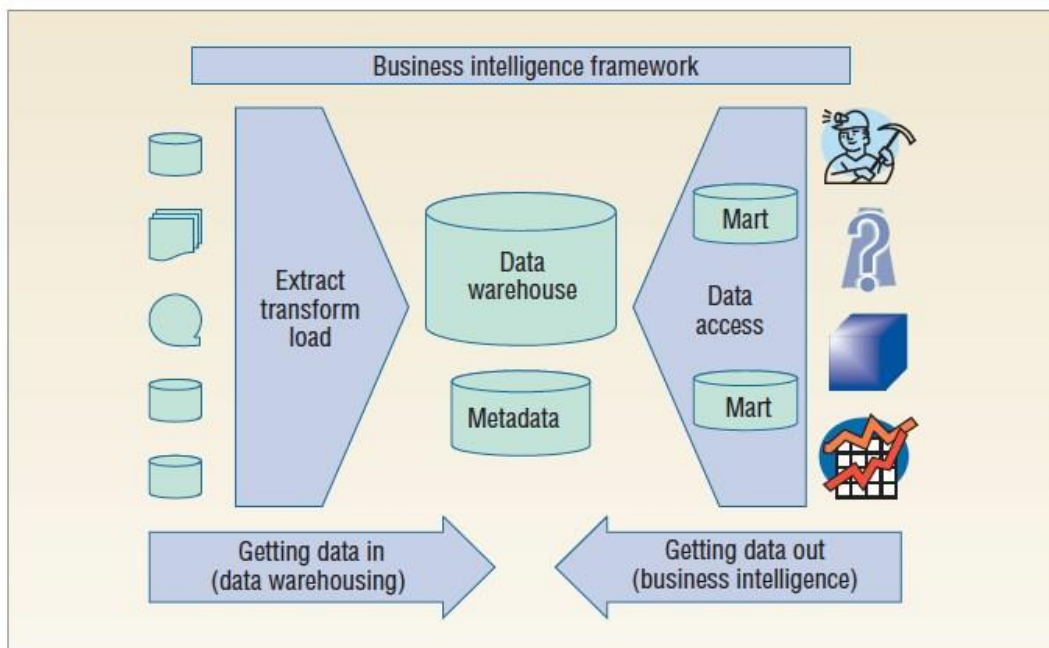


Figura 2.1: Sistema de Business Intelligence [19].

O primeiro, designado por *Getting data in*, corresponde à fase de extração, limpeza, transformação, modelação e carregamento dos dados de um sistema de origem para um armazém de dados integrado.

O segundo, *Getting data out*, corresponde a um conjunto de ferramentas que permitem ao utilizador final a obtenção, exploração e análise dos dados do Data Warehouse. Este acesso por parte de um conjunto de aplicações permite a tomada de decisões informadas, com recurso a técnicas de análise mais elaboradas, como é caso da análise OLAP.

Podemos ver o primeiro processo voltado para a equipa de técnicos especializados e o segundo direcionado para o utilizador do negócio (gestor, financeiro, analista).

2.1.1 OLAP

OLAP (*Online Analytical Processing*) é uma tecnologia de Business Intelligence para processamento de dados, com o objetivo final de transformar dados em informações capazes de dar suporte às decisões de gestão. Assim, uma das grandes vantagens do OLAP é a execução de cálculos mais complexos tais como médias, previsões e taxas de crescimento.

Num contexto de gestão, esta ferramenta possui a particularidade de possibilitar uma análise mais eficiente dos dados, permitindo a construção de relatórios de apoio à decisão, por parte dos utilizadores finais, através da consulta das bases de dados.

Estas aplicações apareceram em conjunto com os Sistemas de Apoio à Decisão para executar a consulta e análise dos dados contidos nos armazéns de dados (DWs).

Antes de especificarmos algumas das características OLAP, importa ainda fazer referência ao conceito de Cubo. O Cubo OLAP é uma estrutura de dados que agrega medidas por níveis e hierarquias de cada uma das dimensões [22]. As dimensões são perspetivas de análise, tais como por exemplo data, local e causa. Cada célula do cubo representa uma medida sobre as perspetivas escolhidas, como por exemplo número total de episódios de urgências numa determinada data, num determinado local.

As ferramentas OLAP apresentam um conjunto de características que incluem:

OPERAÇÃO	DESCRIÇÃO
<i>Drill Down</i>	Aumenta o detalhe da informação e diminui o nível de granularidade.
<i>Roll-Up</i>	Diminui o detalhe da informação e aumenta do nível de granularidade.
<i>Slice</i>	Mantém constante uma dimensão para todas as suas células de modo a ser possível a sua representação de forma bidimensional, por exemplo numa folha de cálculo.
<i>Dice</i>	Produz um subcubo que permite a escolha específica de valores de diversas perspetivas (dimensões).

2.1.2 Scorecard e Dashboard

O *scorecard* é um tipo de relatório que mede o desempenho da atividade de uma empresa, ou seja as suas metas. O *scorecard* resulta de um balanço entre objetivos de curto e longo prazo, bem como, entre indicadores financeiros e não financeiros e determina um plano estratégico da empresa, tendo esta que o seguir levando em conta os seus objetivos e o sucesso no cumprimento destes. Assim, como forma de limitar e alcançar esses objetivos são definidos *key performance indicators* (KPI) para uma empresa autoavaliar a sua atividade e verificar os desvios que prejudiquem o sucesso da atividade. É através da definição dos indicadores de forma gráfica (*scorecard*) que é possível transmitir visualmente o sucesso ou fracasso da empresa [20]. Assim o *scorecard* funciona como um instrumento de apoio à gestão e decisão estratégica.

O *dashboard* é painel para vários tipos de relatórios, incluindo *scorecards*. Este painel pode conter uma ou mais páginas, onde em cada página pode haver mais do que um módulo. Estes módulos são chamados *Web Parts*. Assim, a título de exemplo, um painel pode conter um *scorecard*, um relatório analítico, um gráfico analítico e imagens estáticas. Isto tudo, para que seja

possível ao usuário um certo nível de interatividade. O grau e o tipo de interatividade dependem do programa que é usado para criar o painel [20].

Desta forma um *dashboard* é uma forma eficaz e eficiente de apresentar e comunicar aos decisores, analistas e gestores os principais indicadores de desempenho e resultados de uma empresa.

Assim, numa empresa, poderão ser construídos e implementados diversos dashboards em vários níveis da instituição com o intuito de monitorizar a sua atividade.

O *dashboard* apresenta diversas características, tais como [21]:

- Instrumento de apoio à decisão;
- Exposição dos principais indicadores;
- *Design* eficiente para uma visualização fácil;
- Combinação de diferentes variáveis para análise global;
- Apresentação numa folha A4 ou ecrã com gráficos simples e objetivos.

2.1.3 Vantagens no contexto do projeto

O sistema OLTP (*Online Transaction Processing*/ Processamento de Transações em Tempo Real), responsável pelo registo das atividades diárias do Hospital de Braga, ao ser concebido para desempenhar o suporte às operações do dia-a-dia, acaba por ter pouca flexibilidade para desempenhar funções de análise e consulta. Os sistemas de BI vêm colmatar esta dificuldade.

Relativamente aos sistemas operacionais, os ganhos deste tipo de sistemas incluem:

- a possibilidade de ter uma visão global e agregada dos episódios de urgências, ao longo do tempo;
- flexibilidade de exploração dos dados através de análise multidimensional;
- a unificação dos dados num único repositório de dados permitindo a independência deste face ao sistema operacional;

- uma estrutura mais intuitiva para o gestor do negócio, dado que o número de ligações entre tabelas é mais simples e reduzido;
- consultas previamente calculadas e/ou automatizadas permitindo um ganho de tempo no que diz respeito à elaboração e visualização das consultas;
- acesso a mais e melhor informação;
- análise baseada em dados consistentes, o que permite a tomada de decisões informadas e mais fiáveis.

2.2 Data Warehouse (DW)

Podemos ver um Data Warehouse (armazém de dados) como um repositório de dados crescente no tempo, integrado e direcionado às necessidades de uma empresa, e que dá acesso rápido à informação necessária à tomada de decisões.

Um Data Mart (DM) é um subconjunto de dados de um Data Warehouse, sendo usualmente orientado para responder a um determinado departamento ou setor de negócio de uma organização. Esta subdivisão dos dados do Data Warehouse permite obter uma estrutura mais simples de entender e mais flexível.

Inmon e Kimball, dois dos especialistas mais influentes da área, apresentam visões diferentes sobre o que deve ser a arquitetura de um DW.

Para Inmon, um Data Warehouse é uma coleção de dados primitivos, orientado por assunto, que varia no tempo e é acessível de modo a permitir o suporte ao processo de tomada de decisão. Os dados do armazém de dados devem ser não voláteis, isto é, uma vez inseridos não ocorrem atualizações, para que seja preservado o histórico do fluxo de dados. Segundo Inmon, é a partir deste conjunto de dados integrados que são derivados os Data Marts (armazéns de dados departamentais).

Em [1], Inmon define Data Mart como um resumo e/ou especificação de dados provenientes do Data Warehouse relativos a uma determinada unidade de negócio (departamento). Esta abordagem permite tirar partido da integração dos dados, obtendo uma única visão da realidade.

Dado que a informação do Data Mart é obtida a partir do Data Warehouse estes podem estar, ou não estar, no mesmo espaço físico (servidor).

Os dados do Data Warehouse vão desde um nível mais granular de detalhe de informação, até a um nível mais resumido e/ou específico, no Data Mart.

Kimball rebate essa ideia defendendo que o Data Warehouse é o resultado da união de todos os Data Marts. Cada Data Mart contém dados detalhados e sumariados relativos a um sector de negócio.

Segundo Kimball [2], um Data Warehouse deve:

- facilitar o acesso à informação;
- possuir dados consistentes, credíveis e de fácil interpretação;
- ser adaptável e flexível a mudanças;
- ser seguro de forma a proteger a informação;
- servir de base à tomada de decisão;
- cumprir a finalidade e utilidade desejada pelos utilizadores finais, a fim de ser aceite e bem sucedido.

Kimball e Inmon defendem duas metodologias diferentes para a implementação de um Data Warehouse.

Segundo Inmon, deve-se construir primeiro o DW, moldando toda a companhia de modo a obter um único modelo central, com dados transacionais ou altamente detalhados. E, posteriormente, procede-se à implementação dos Data Marts por assuntos ou departamentos. Assim, esta abordagem é chamada de *top-down*: cria-se primeiro o DW e depois segmenta-se em áreas menores (DM) orientadas por negócio.

Kimball rebate essa metodologia, propondo justamente o contrário.

Defende que é mais fiável e seguro para as empresas desenvolver Data Mart a Data Mart (um ou vários) e depois integrá-los de modo a obter o Data Warehouse. Estes Data Marts, orientados por assunto, contêm dados detalhados e sumariados. Esta abordagem é designada por *bottom-up*: criam-se primeiro os Data Mart de forma incremental, e depois integram-se para obter o Data Warehouse.

Estas abordagens estão na base das arquiteturas de Data Warehouse propostas pelos autores, que descrevemos na secção 2.4. Antes abordaremos o modelo dimensional e alguns fundamentos do desenho e conceção de um armazém de dados (DW).

2.3 Modelo Dimensional

O modelo dimensional é um modelo de dados adequado para a conceção (representação lógica) de Dws, assente numa estrutura intuitiva, que visa permitir a consulta dos dados com um elevado desempenho. A base desta abordagem é a possibilidade de representar os dados num cubo multidimensional possibilitando uma vasta gama de consultas e análises através de sistemas OLAP (este cubo pode ser um hipercubo quando não é tridimensional e pode não corresponder a um cubo por não ter lados iguais).

Este modelo envolve dois tipos de dados – dimensões e factos – que se representam por Tabelas de Dimensão e Tabelas de Factos.

2.3.1 Tabelas de Dimensão

As Dimensões são as perspetivas pelos quais os factos (medidas) podem ser analisados. É através da Dimensão que obtemos as respostas às perguntas “quem, o quê, onde, quando, porquê e como” [14]. É a partir dos seus atributos, maioritariamente descritivos, que é possível a filtragem e o agrupamento dos factos nas consultas e análise dos dados. Sempre que conveniente, são estruturadas hierarquicamente de forma a facilitar o *drill-down* de um nível mais agregado de informação até um nível mais atómico, e vice-versa (*drill-up*). Cada Tabela de Dimensão tem uma chave primária (*Primary Key – PK*), a qual faz correspondência com uma chave estrangeira (*Foreign Key – FK*) na Tabela de Factos. É esta correspondência que permite a ligação entre as tabelas. As chaves são usualmente independentes do sistema operacional, sendo criadas no momento da implementação do Data Warehouse (começando do valor 1, e incrementando de uma unidade sempre que uma nova chave é necessária).

2.3.2 Tabelas de Factos

Uma Tabela de Factos representa a ocorrência de evento de um determinado processo de negócio. Os factos são as métricas, normalmente numéricas, que resultam desse evento, observável num espaço de tempo [14]. Além destas

medidas, a tabela de factos contém sempre as chaves estrangeiras para cada uma das suas dimensões associadas. Pode conter também dimensões degeneradas, que abordaremos adiante.

Podemos ter tabelas de factos com ou sem métrica associada. As tabelas de factos com factos podem ser classificadas como sendo de três tipos [13]:

- tabelas de factos transacionais;
- tabelas de factos de sumarização periódica;
- tabelas de factos sumarização acumulada.

Estes três tipos serão abordados em maior pormenor na Secção 5.4.

As tabelas de factos sem métricas associadas podem ser classificadas em *FactLess Table* e *FactLess Coverage Table* [10].

As *FactLess Tables* (tabelas de factos sem factos) são usadas para representar a própria ocorrência de um evento. Por exemplo, o registo da ocorrência de uma aula ou da presença de um estudante numa aula num determinado dia. Este evento poderia não ter uma métrica associada mas uma chave estrangeira para a data da aula da disciplina, o estudante, o professor o local, entre outras dimensões.

As *FactLess Coverage Table* são usualmente utilizadas quando pretendemos contabilizar um evento e a cobertura (*coverage*) desse quando não aconteceu. Normalmente, neste caso temos uma tabela com todos os eventos que aconteceram e uma segunda tabela de cobertura com todos os eventos possíveis. A tabela de cobertura permite resgatar os acontecimentos que não aconteceram, através da diferença entre a tabela de cobertura e a tabela com os registos dos eventos que aconteceram. Para além de possibilitar análises negativas, isto é, verificar eventos que não aconteceram, é especialmente útil quando queremos saber o total de eventos que poderiam ter acontecido e não aconteceram.

Uma abordagem mais óbvia, também válida segundo Kimball, seria ter uma única tabela com uma coluna, em que a ocorrência de um evento corresponderia a 1, e no caso contrário 0. Esta coluna seria preenchida unicamente com zeros e uns.

Esta abordagem poderia não ser tão prática. Por exemplo, no caso do número de vendas de um produto, com uma percentagem de vendas baixa. O facto de cada “não ocorrência” do evento, não venda de um produto, corresponder a um 0, iria logicamente expandir enormemente a tabela de factos com zeros. Para além de não fazer sentido estar a aumentar desta forma a tabela de factos, seria um desperdício de unidades de disco para armazenar todos estes zeros.

2.3.3 Métrica

Uma métrica é uma medição de um atributo de um determinado processo de negócio. Estas métricas numa tabela de factos podem ser classificadas em três categorias [14] de acordo com a possibilidade de agregação:

- Aditivas: métricas que podem ser agregadas por todas as dimensões associadas à tabela de factos;
- Semi-aditivas: métricas em que só fazem sentido serem resumidas através de algumas dimensões, mas não em todas;
- Não-aditivas: métricas que dado o contexto do negócio não faz sentido serem agregadas.

A título de exemplo, um evento seria um episódio de urgências de um hospital e o valor pago nessa urgência seria uma métrica aditiva possível. Métrica não-aditiva poderia ser a percentagem de urgências por dia.

2.4 Arquiteturas para Data Warehouse

Escolher a arquitetura a adotar na implementação de um Data Warehouse é um dos fatores cruciais no que se refere a questões de escalabilidade e desempenho. Não obstante o melhoramento dos processos de Data Warehousing ao longo dos anos, ainda não existe um consenso em relação à melhor arquitetura a adotar.

A arquitetura *hub and spoke*, defendida por Inmon, tem como base a metodologia *top-down* (Secção 2.2). Nesta arquitetura, o Data Warehouse fornece uma perspetiva global (organizacional), com um nível de detalhe mais atómico e elementar possível. Os dados são extraídos das fontes

operacionais e carregados no Data Warehouse, ao mais alto nível de detalhe. Este nível contém os dados primitivos de forma integrada, histórica e não sofre qualquer tipo de atualização. Contudo pode ainda conter alguns dados derivados [2]. Os Data Marts desenvolvidos para uma área de negócio ou departamento, são dependentes do repositório central (Data Warehouse) e criados posteriormente. Os dados no Data Mart são alimentados através do Data Warehouse e podem estar normalizados, desnormalizados ou estruturados de forma multidimensional. Por outras palavras, os Data Marts são obtidos do Data Warehouse, e a informação nos Data Marts estará a um nível mais sumariado, agregado e moldados quase que exclusivamente às exigências dos departamentos. É a partir destes dois níveis, Data Mart e Data Warehouse que é feita a análise dos dados. Os Data Marts, permitem uma visão consistente com a realidade e com um nível maior de detalhe. Por sua vez o Data Warehouse permite uma visão empresa, possibilitando uma análise global da organização. Contudo, estes dois níveis de análise, data warehouse e data mart, podem não ser tão intuitivos para o utilizador final.

Na Figura 2.2 que se segue podemos ver esquematizada a Arquitetura *Hub and Spoke* proposta por Inmon.

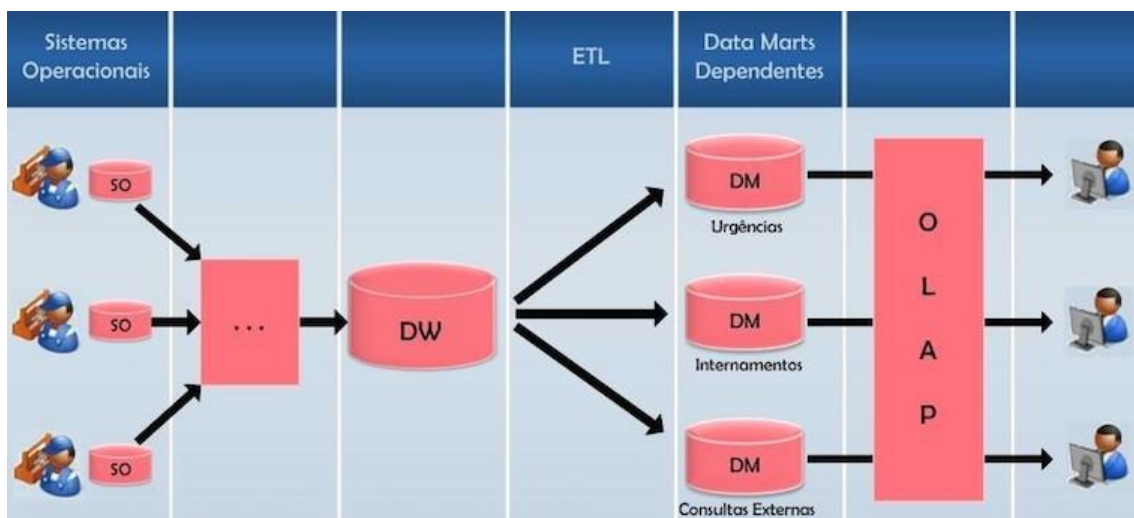


Figura 2.2: Arquitetura *Hub and Spoke* proposta por Inmon

O facto de ser necessário um conhecimento muito pormenorizado e global de todo o negócio, visto se querer projetar primeiro um único modelo corporativo (Data Warehouse), relega para segundo plano a identificação das reais necessidades do utilizador final. Tal prolonga o tempo de ausência de

resultados. Esta arquitetura acarreta custos superiores e um risco elevado na implementação, quando comparada por exemplo, com a de Kimball.

Contudo, apresenta a vantagem dar uma visão completa e flexível de toda a organização possibilitando ao utilizador obter uma única versão da realidade dada pelo Data Warehouse.

Kimball rebate esta ideia, propondo uma arquitetura baseada na metodologia *bottom-up* (Secção 2.2). Na sua proposta, designada por *Data Warehouse Bus Architecture*, a matriz de barramento dimensional constitui o ponto crucial.

Esta matriz é uma tabela que documenta a arquitetura do Data Warehouse, onde se identificam os processos de negócio (linhas da tabela) e dimensões (colunas da tabela) de toda a organização. Desta forma é possível uma análise e identificação global de todas as estrelas e dimensões que vão permitir projetar o desenvolvimento do Data Warehouse de modo mais controlado e seguro. Importa ainda realçar que as dimensões identificadas na matriz que forem comuns a mais que um Data Mart (Dimensões Conformes) serão as dimensões que permitirão a união dos diferentes Data Marts para formar o Data Warehouse.

Tendo a matriz completa, ou seja, tendo concluído a análise de todos os departamentos de modo a que todas as estrelas e dimensões estejam devidamente definidas e identificadas, o passo que se segue é a escolha e implementação de um dos processos de negócio (Data Mart) identificados. Este Data Mart e os seguintes usarão dimensões conformes para poderem ser interligados e dar origem ao Data Warehouse.

Nesta arquitetura, os dados dos Data Marts apresentam-se num modelo dimensional. O esquema mais utilizado nestes Data Marts é o esquema em estrela, que se caracteriza por uma tabela de factos central, rodeada por tabelas de dimensão. As ligações entre a tabela de factos e as tabelas de dimensão relevantes formam a estrela (Data Mart).

Este modelo possui uma estrutura com menos relações entre as tabelas e permite reduzir o número de *joins* necessários nas consultas, melhorando o desempenho. Ao possuir uma estrutura mais simples fornece um modelo mais intuitivo e mais fácil de usar.

A nível de detalhe dos dados no Data Mart, estes podem ser sumariados ou estar a um nível mais atómico e elementar, o que permite ao utilizador no

mesmo ambiente ter acesso a um nível mais rico de informação. Tal não acontece na arquitetura proposta por Inmon, onde o nível mais detalhado da informação se encontra no Data Warehouse e não no Data Mart. O facto de se concentrar numa área específica de negócio permite a visualização de resultados de forma mais rápida e com um menor investimento. O projeto é realizado a nível departamental e de forma incremental e não organizacional, como no caso da arquitetura proposta por Inmon.

Por fim, nesta arquitetura, o carregamento dos Data Marts é feito a partir de uma *Staging Area*, onde os dados provenientes dos sistemas fontes são carregados para serem transformados e posteriormente carregados nos Data Marts. Esta área possibilita uma separação dos sistemas operacionais, permitindo que os dados estejam sempre disponíveis. Assim permite começar ou recomeçar a extração dos dados para o Data Mart, sem estar dependente do sistema operacional. Para além disso, o facto de a extração inicial do sistema operacional para a *Staging Area* não ter qualquer tipo de transformação de dados torna este processo mais rápido. Importa ainda referir que esta área não é projetada para a apresentação. Os utilizadores não têm acesso. Apenas é usada para auxiliar e melhorar a performance do ETL do Data Warehouse. Na Figura 2.3, que se segue podemos ver esquematizada a *Data Warehouse Bus Architecture* proposta por Kimball.

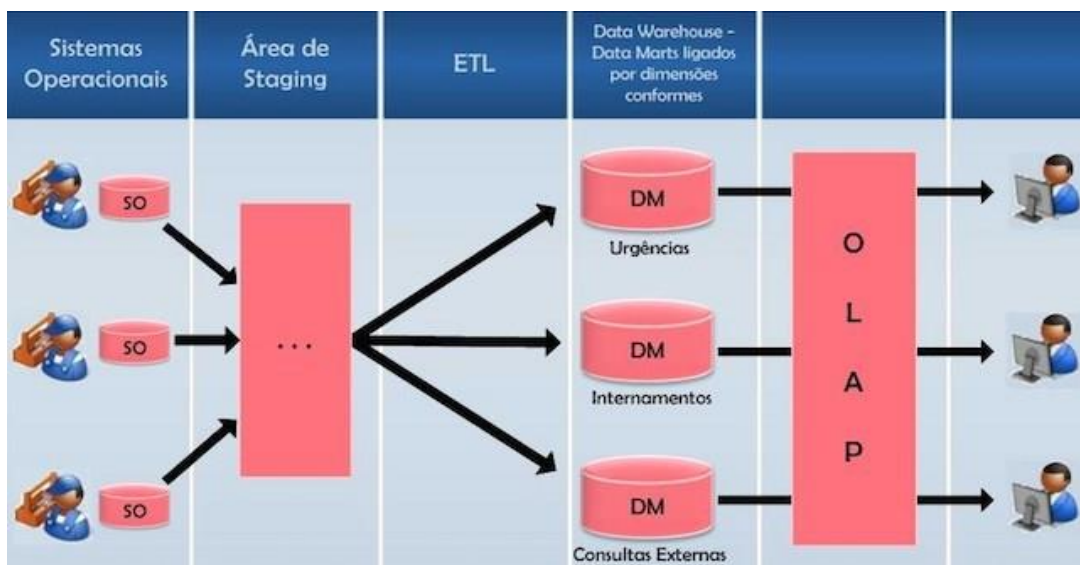


Figura 2.3 Data Warehouse Bus Architecture.

2.5 Dimensões “Role-Playing” e “Slowly Changing”

Kimball & Ross (2002) apresentam um caso de estudo relativo à implementação DW na área da saúde. Nesta área de estudo, identificam como ponto crucial o tratamento do paciente, que no contexto se traduzirá na dimensão Paciente.

Role-Playing Dimension refere-se a uma dimensão que possibilita diferentes visões dependendo do contexto, ou seja, é uma dimensão que pode ser referenciada mais que uma vez num evento de uma tabela de factos. Cada referência é ligada a uma visão lógica da *Role-Playing Dimension* distinta [17]. Para tal, terá uma chave estrangeira distinta para cada visão da Dimensão.

Um exemplo típico de utilização é a Dimensão Data. Esta abordagem permite-nos ter um acompanhamento das diferentes datas de um determinado evento com apenas uma tabela física da Dimensão Data. A tabela de factos terá tantas chaves estrangeiras para a Dimensão Data quantas as visões que o evento da tabela de factos tiver sobre a data.

No exemplo abordado no estudo, Kimball e Rose, referem alguns pontos importantes a reter no que diz respeito à implementação de *Role-Playing Dimensions* que incluem:

- a criação de chaves estrangeiras com ligação à mesma dimensão Data permite-nos diferentes visões sobre a mesma. Por outras palavras, com esta abordagem podemos ter vários pontos de vista independentes, numa única dimensão física;
- a chave estrangeira da tabela de factos não deve ser uma cópia de datas reais, pois terá de lidar com o caso “não se aplica”, “ainda não ocorreu”. Nestes casos, deve ser atribuído um valor superior, usualmente sem significado real no intervalo de tempo (por exemplo 99991231), para que essas linhas possam ser facilmente filtradas. Apesar da chave estrangeira dessa linha não dever ser *Null*, o seu conteúdo poderá ser vazio;

- a chave substituta da dimensão data é usualmente atribuída em conformidade com respetiva data. Ou seja a data "2014-10-31" corresponderá a chave "20141031", sendo este um caso particular em que a chave está relacionada com a semântica subjacente da dimensão. Assim, é possível utilizar a chave substituta para o particionamento da tabela de factos.

No contexto da área da saúde abordado neste estudo, focando-nos em particular na dimensão paciente, é fácil imaginar uma situação em que o que era verdade deixou de o ser a partir de um determinado momento. A esta mudança da precisão do fluxo de dados numa dimensão com o decorrer do tempo de forma gradual, ocasional ou imprevisível dá-se o nome de *Slowly Changing Dimensions* (SCD). Esta técnica é um procedimento importante de atualização no processo de ETL. Podendo ter diferentes abordagens perante este tipo de dimensões, Kimball & Ross enumeram oito tipos de SCD diferentes. Desses será relevante referir:

- SCD do tipo 0: é a dimensão em que não é efetuado qualquer tipo de alteração dos campos da dimensão.
- SCD do tipo 1: é a dimensão onde ocorre uma alteração do valor e é feita a substituição do valor antigo pelo novo. Esta abordagem tem a desvantagem de se perder o histórico do fluxo de dados.
- SCD do tipo 2: é a dimensão onde é adicionada uma nova linha a cada mudança do valor de um atributo. Para se poder controlar esta situação é adicionada a este tipo de SCD três novas colunas. Uma primeira coluna com a data de início, seguida da data de fim (que representa a data de alteração do atributo e data em que acabou, isto é, a data em que foi atualizado). A terceira coluna diz respeito ao estado e, usualmente será 0 se o estado está inativo e 1, caso contrário. Este último identifica como ativa a linha mais recente e atualizada. Com esta técnica conseguimos manter o histórico da dimensão ao longo do tempo, não havendo alterações na tabela de factos a não ser a atualização da chave estrangeira. Os campos de data de início e de fim permitem-nos conhecer o intervalo de tempo para qual o

valor foi ou é, verdade. Um ponto fraco desta abordagem é o crescimento rápido da dimensão, se os seus valores estiverem continuamente a sofrer alterações.

- SCD do tipo 3: é a dimensão em que é adicionada uma coluna para o novo valor que sofreu alteração. Este processo permite-nos atualizar a dimensão e manter o seu histórico, na mesma visão. Tem o inconveniente de levar ao crescimento de forma acelerada do número de novas colunas criadas, se os atributos sofrerem constantes alterações.

3 Tecnologias Utilizadas

Existem diversas plataformas disponíveis para suporte à implementação, modelação e integração de um armazém de dados, o que não torna fácil a escolha de uma solução. Contudo, no presente projeto, a escolha recaiu pela solução fornecida pela Microsoft. Esta decisão foi tomada, em grande parte, por duas razões de ordem prática.

A primeira razão foi o facto de a ARS-Norte já possuir a licença necessária. Tal possibilitou um ganho de tempo por não ser necessário uma nova aquisição. Ao já existir a tecnologia na ARS-Norte, apesar de esta ser uma solução comercial, não houve aumento dos custos.

A segunda razão é o facto de a solução Microsoft ter sido já utilizada num projeto anterior de Business Inteligente no Departamento de Estatísticas da ARS-N, facilitando o acompanhamento do projeto pelos técnicos da instituição e futuramente a sua manutenção sem perda tempo na aprendizagem de uma nova tecnologia.

A Microsoft é uma empresa com provas dadas a nível mundial no que diz respeito ao desenvolvimento de *software* e, em particular em soluções de gestão de bases de dados. Tem disponível uma ampla gama de ferramentas que permitem implementar de forma integral um armazém de dados bem como a sua análise posterior. A Figura 3.1, apresenta uma visão de uma possível arquitetura usando a solução Microsoft.

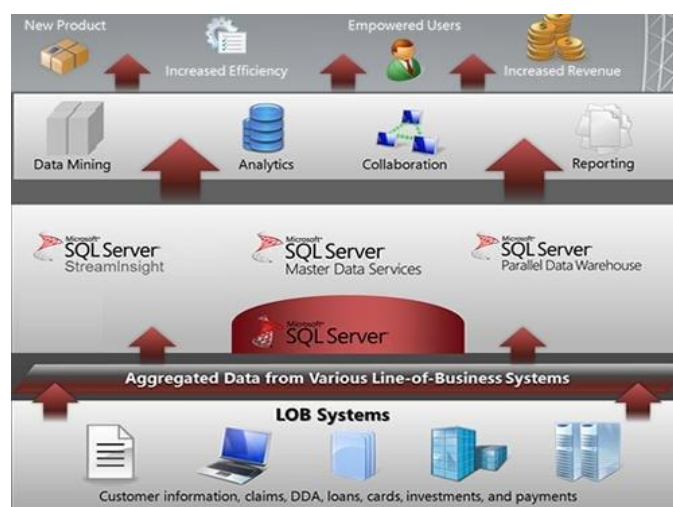


Figura 3.1: Arquitetura de um Data Warehouse da Microsoft [23].

O Microsoft SQL Server (MSS) é um sistema de gestão dados, que fornece uma plataforma robusta e escalonável para o armazenamento de dados.

A sua capacidade de integração com diferentes tecnologias de Business Intelligence e, em particular, com soluções Microsoft possibilitou não somente dar suporte à implementação do Data Mart como tirar partido dos dados.

Dentro das possibilidades oferecidas pela Microsoft para a integração de dados foi utilizado o SQL Server Integration Services (SSIS), componente que permitiu a realização de operações de extração, transformação e carregamento dos dados.

De seguida enunciamos algumas das principais funcionalidades disponíveis e utilizadas neste projeto:

- criação de pacotes de importação e exportação de grandes volumes de dados;
- capacidade de conexão e integração de dados de múltiplos sistemas fontes (Oracle, SQL, Excel, etc.);
- ampla gama de transformações pré-compiladas que permitem a limpeza, preparação e a remoção de duplicados dos dados;
- possibilidade de criação e manutenção *Slowly Changing Dimensions*;
- suporte à automatização da manutenção do armazém de dados no SQL Server;
- integração com o SQL Server Analysis Services (SSAS), permitindo a construção de cubos.

No que concerne à modelação multidimensional e sua análise por via de relatórios e/ou *dashboards*, foi utilizado o SQL Server Analysis Services. Esta ferramenta contém um conjunto de serviços que permitem a conceção de cubos multidimensionais permitindo a implementação de hierarquias, o cálculo de métricas derivadas e de indicadores chave de desempenho (KPIs). Para a implementação do Data Mart foram adotadas as seguintes ferramentas: sistema de gestão de base de dados Microsoft para o armazenamento do Data Mart; a interface de integração de dados da Microsoft SQL Server Integration Services para tarefas de extração, transformação e carregamento de dados. Como suporte a modelação multidimensional utilizou-se SQL Server Analysis Services. No Excel é feito o

acesso ao (SSAS) e realizada a apresentação dos resultados. Deste modo, a Microsoft apresenta uma oferta global de soluções que permitem administrar (SSMS), integrar (SSIS), analisar (SSAS) e apresentar (Excel) os dados que pretendemos analisar.

4 Desenvolvimento do Projeto

4.1 Análise da Instituição

A Administração Regional de Saúde do Norte (ARS-N) tem como principais atribuições a gestão e planeamento dos diferentes níveis de prestação de cuidados de saúde na Região Norte. Abrange uma população de 3.750.000 pessoas, cerca de 35% do total nacional. Para a gestão e planeamento dos diferentes níveis de cuidados de saúde, a ARS-N necessita de explorar os dados gerados por aplicações OLTP da ARS-N (Glantt no caso do Hospital de Braga), analisando e desenvolvendo perceções e entendimentos a seu respeito que permitam aos vários serviços centrais da ARS-N apoiar a tomada de decisões.

Como referimos anteriormente este projeto teve por objetivo o desenvolvimento de um Sistema de Business Intelligence (SBI) para o tratamento dos dados das urgências que a ARS-N, como representante da Entidade Pública Contratante (EPC) da Parceria Pública Privada (PPP) do Hospital de Braga, recebe mensalmente.

4.2 Arquitetura proposta

A ARS-N, além de dados sobre as urgências hospitalares recebe mensalmente dados de todas as unidades de serviço do Hospital de Braga, como consulta externa, internamento, cirurgia, entre outros.

Dada a amplitude de serviços (Data Marts) da intervenção do Hospital, pareceu-nos impensável começar por construir um Data Warehouse organizacional, de uma vez só.

Deste modo, no presente projeto adotamos a arquitetura de Data Warehouse em Bus, de Kimball, para desenvolver um Data Mart correspondente às urgências do Hospital de Braga. Esta escolha permitiu começar com um projeto mais pequeno e testar a sua usabilidade, mas não descarta a ideia de implementar mais tarde outros Data Marts de outros serviços a nível hospitalar. Isso requer a análise prévia da matriz de barramento dimensional para identificar as dimensões conformes numa visão global dos serviços. Desta forma, dada a proximidade dos serviços, foi feita uma matriz de

barramento com os Data Marts Urgências, Consulta Externa e Internamento. Esta matriz de barramento dimensional encontra-se na Secção 5.5.

No seguimento da decisão desta arquitetura, decidimos também implementar uma *Staging Area*, onde os dados recebidos mensalmente pela ARS-Norte em ficheiros de Excel serão integralmente carregados para um repositório de dados (base de dados SQL Server). As razões que justificam a utilização desta *Staging Area* foram abordadas de forma mais pormenorizada e fundamentada na Secção 2.2. Usando o SSIS é efetuada a limpeza e transformação dos dados para o Data Mart, num servidor SQL Server 2012.

Este processo de extração, transformação e carregamento dos dados de um sistema de origem para um Data Mart ou Data Warehouse é popularmente chamado de ETL (Extract, Transform and Load). Na Secção 6.1 será abordada e explicada de forma mais detalhada a implementação de cada uma destas fases.

A Figura 4.1 esquematiza a arquitetura proposta neste projeto.

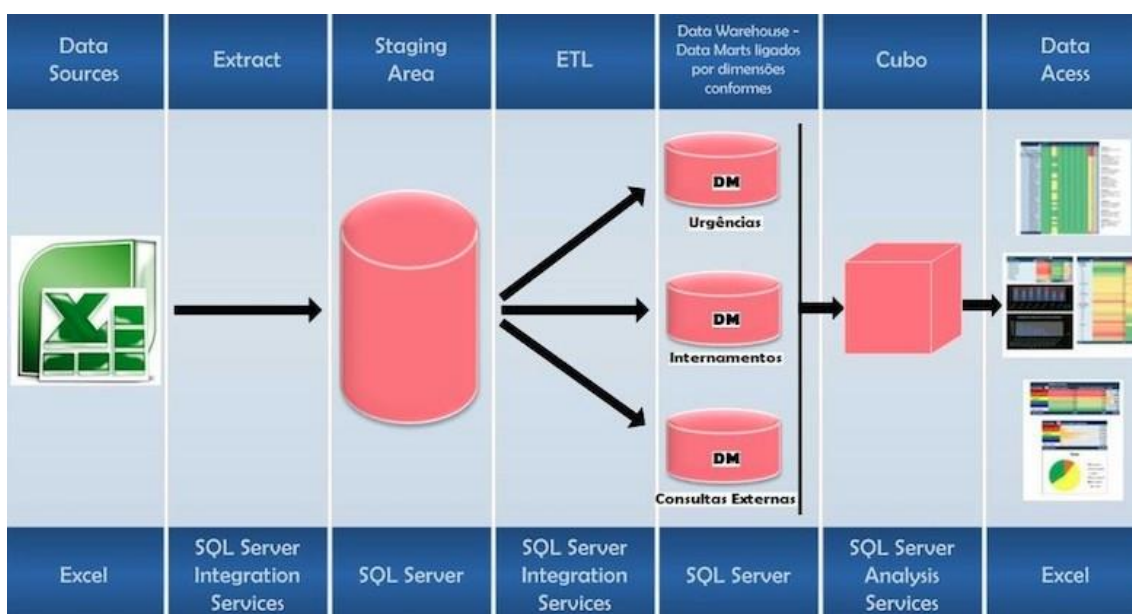


Figura 4.1: Arquitetura Proposta.

4.3 Levantamento de Requisitos

O principal foco deste projeto é utilizar os dados relativos a episódios de urgências do Hospital de Braga disponibilizados à ARS-N mensalmente, e integrar estes registos numa única base de dados para que a análise e validação das regras da elegibilidade para pagamento sejam realizadas de uma forma mais otimizada e célere.

As validações a serem realizadas deverão identificar os episódios de urgências:

- sem número de processo e que tenham sido considerados elegíveis para pagamento pela Entidade Gestora Responsável;
- sem pelo menos um dos números seguintes e que tenham sido considerados elegíveis para pagamento pela Entidade Gestora do Estabelecimento: número de beneficiário, número de cartão de utente, número de cartão europeu de saúde;
- em que se verifique abandono antes ou após a triagem e que tenham sido considerados elegíveis para pagamento pela Entidade Gestora do Estabelecimento;
- sem atribuição de cor ou em que a cor da prioridade atribuída não respeite uma das cinco cores do sistema de Triagem de Manchester (azul/ verde/ amarelo/ laranja/ vermelho) e que tenham sido considerados elegíveis para pagamento pela Entidade Gestora do Estabelecimento;
- em que o utente tenha permanecido na urgência por um período superior a 24 horas e que tenham sido considerados elegíveis para pagamento pela Entidade Gestora do Estabelecimento;
- ginecológicas/obstétricas realizadas a utentes do sexo masculino e que tenham sido considerados elegíveis para pagamento pela Entidade Gestora do Estabelecimento;
- pediátricas realizadas a adultos e que tenham sido considerados elegíveis para pagamento pela Entidade Gestora do Estabelecimento;
- considerados elegíveis para pagamento pela própria Entidade Gestora do Estabelecimento (EGEST).

Para além da identificação destas validações fornecidas pela ARS-N foi também disponibilizado pelo Departamento de Estudos e Planeamento o conjunto de indicadores seguintes:

- Evolução do número de episódios de urgência no Hospital de Braga, onde deverá ser possível a agregação e desagregação por: ano, semestre, trimestre, mês e dia.

Sendo a ARS a entidade responsável por garantir o acesso dos utentes a cuidados de saúde, a monitorização da evolução do número de urgências e possibilidade de comparação, com os períodos anteriores e homólogos é essencial para verificar afluências anómalas e fenómenos sazonais. Desta forma, consegue-se atempadamente adequar as valências necessárias, em casos que possam justificar medidas extraordinárias.

Caracterização da procura por:

- proveniência dos utentes, distinguindo os que vieram por iniciativa própria, ou a partir de outro hospital; género, idade e faixa etária dos utentes;
- residência dos utentes (distrito, concelho, freguesia);
- caracterização do destino dos utentes após a alta (internamento, exterior);
- monitorização de tempos de espera e de permanência por prioridade, de forma a perceber a percentagem de urgências que ultrapassam o tempo de espera e permanência recomendado;
- referenciação, para perceber se a maioria dos utentes nos episódios de urgências diz respeito à rede de abrangência direta do Hospital de Braga (1ª linha) ou à área de abrangência de outros hospitais (2ª linha).

Com base no exposto e no trabalho desenvolvido por parte do Departamento de Estudos e Planeamento da ARS Norte, foi possível elaborar os seguintes KPIs:

- Percentagem de episódios de urgência em que o tempo total de permanência na urgência foi superior a 4 horas. O objetivo será não ultrapassar os 10%;
- Sempre que o número de episódios de urgências face ao período homólogo ultrapasse os 20% deve ser comunicado aos serviços;
- Percentagem de episódios em que a rede de referência de 2ª linha ultrapasse os 15% num determinado concelho.

Estes indicadores foram disponibilizados para este projeto, pelo Departamento de Estudos e Planeamento da ARS-N, sendo referentes à solução de BI realizada nas Urgências Hospitalares da ARS-N.

5 Modelação Dimensional

Segundo Kimball, a metodologia para a construção de um Data Warehouse possui quatro passos, que incluem:

- seleção do processo de negócio a modelar;
- declaração da granularidade dos dados;
- escolha das dimensões a aplicar;
- identificação dos factos.

5.1 Processo de Negócio

No âmbito deste projeto pretende-se a implementação de um Data Mart, relativamente aos dados dos episódios de urgência do Hospital de Braga disponibilizados mensalmente à ARS-N. Para permitir a extensão para outros serviços futuramente, na análise foi tido em conta ainda o Data Mart Consulta Externa e Data Mart Internamento. Esta decisão foi tomada pela proximidade dos serviços potenciar a existência de dimensões conformes, como veremos na Secção 5.4.

5.2 Definição de Granularidade

Na segunda etapa, devemos escolher a granularidade da tabela de factos, que se traduz pelo significado atribuído a cada linha da tabela de factos. Ou seja, a granularidade irá expressar o nível de detalhe mais elementar que é representado na tabela de factos. Deste modo, a definição da granularidade está intimamente ligada ao volume de dados que será armazenado.

Apesar de acarretar um maior aumento do volume de dados, o que pode resultar num agravamento de desempenho, para este projeto foi dado privilégio ao nível mais baixo de granularidade. Assim, os utilizadores poderão navegar tirando partido de maior flexibilidade dos dados, podendo ver a informação sobre a perspectiva de diferentes níveis de detalhe.

A granularidade escolhida foi a seguinte:

- Tabela de factos “urgências” - cada linha corresponde a um episódio de urgência, em que a granularidade da Dimensão Data é o dia e a granularidade da Dimensão Minuto é o minuto.

- Tabela de factos "internamento" - cada linha corresponde a um episódio de internamento. A granularidade da Dimensão Data é o dia e a granularidade da Dimensão Minuto é o minuto.
- Tabela de factos "consulta externa" - cada linha corresponde a uma consulta externa em espera. A granularidade da Dimensão Data corresponde ao final de cada mês.

5.3 Identificação das dimensões e sua caracterização

Na terceira etapa, são identificadas e descritas as dimensões que possibilitarão a análise dos episódios de urgências por diferentes perspectivas. É nesta fase que definimos os atributos da dimensão, bem como se estes precisam de atualizações. No caso de requererem, é nesta fase que é acautelado se os registos necessitam de preservar o seu histórico ao longo tempo, ou não. É também nesta etapa que definimos as hierarquias da cada dimensão.

Neste passo só serão identificadas e detalhadas as dimensões correspondentes ao Data Mart Urgências.

5.3.1 Dimensão Data

A Dimensão Data é certamente a dimensão mais utilizada em praticamente todos os sistemas de Business Intelligence. Por esta razão é muitas vezes reutilizada não só num mesmo Data Warehouse através de Dimensões Conformes, mas também muitas vezes reutilizada no mesmo Data Mart através de *Role-Playing Dimensions*.

É esta abordagem, em particular *Role-Playing Dimensions*, que utilizamos para representar as diferentes datas: data entrada, data triagem, data entrada no serviço de observação, data saída do serviço de observação e data saída.

Com esta técnica teremos uma representação consistente das diferentes datas da tabela de factos, uma vez que o acompanhamento temporal no episódio de urgências é sobre a mesma dimensão.

Assim será necessário menos espaço de armazenamento e será gasto menos tempo na implementação

A Dimensão Data compreende os atributos identificados na Tabela 5.1.

Nome do Atributo	Descrição do atributo	Exemplo de valor
SK_DIM_DATA	Chave substituta da Dimensão Data	20100101; 20100102;
DIA_DA_SEMANA	Dia numérico da semana (Level Key)	6; 7;
NOME_DIA_DA_SEMANA	Dia da semana por extenso	Sexta-Feira; Sábado;
DIA_DO_MES	Indica o dia do corrente mês	1; 2; 31;
DIA_DO_ANO	Indica o dia do ano	1; 2; 365;
FDS_OU_DIA_UTIL	Identifica se é dia útil ou fim-de-semana	DIA ÚTIL; FIM DE SEMANA;
SEMANA_DO_ANO	Indica o número da semana do ano	1; 2; 52;
COD_MES	Código mês (Level key)	2010-01; 2010-02;
MES	Mês por extenso	Janeiro; Fevereiro;
MES_DO_ANO	Número do mês	1; 2;
COD_TRIMESTRE	Código Trimestre (Level key)	2010T1; 2010t2;
TRIMESTRE	Trimestre	1; 2; 3; 4;
COD_SEMESTRE	Código Semestre (Level key)	2010S1; 2010S2;
SEMESTRE	Semestre	1; 2;
ANO	Ano	2010; 2011;
UK_INSTANTE	Data do instante (Unique key)	2010-01-01; 2010-0102;

Tabela 5.1: Descrição dos atributos da Dimensão Data

A granularidade adotada para a Dimensão Data foi a diária. Embora a granularidade esteja sempre relacionada com tipo de negócio, esta escolha conduz a um modelo mais flexível e amplo. O nível atómico da granularidade permite aumentar o nível de detalhe (*drill-up*), e o contrário (*drill-down*).

A Dimensão Data apresenta uma estrutura hierárquica com cinco níveis de análise Ano-> Semestre-> Trimestre-> Mês-> Dia. Esta hierarquia poderá ser consultada em mais pormenor na secção 6.2.1.

Importa ainda referir que os códigos criados na dimensão são - *Level Key* (LK) – ou seja, chaves que definem os elementos únicos em cada nível lógico da hierarquia. Cada nível lógico, exceto o nível mais alto, é definido como um nível global, tendo uma chave de nível (LK) associada. A chave da tabela de dimensão (*Primary Key*) tem de ser associada com o menor nível da dimensão e tem de ser a chave de nível para esse nível (nível global).

A Dimensão Data será carregada com datas correspondentes a 8 anos, concretamente de 2010 a 2018, sendo necessário a sua atualização antes do final de 2018.

5.3.2 Dimensão Minuto

A Dimensão Minuto compreende os atributos: Período horário, Período, hora e minuto.

Estes períodos foram divididos com base no funcionamento do Hospital de Braga.

Na primeira coluna, Período horário, o dia é dividido em Período diurno e noturno. Cada um dos períodos corresponde a 12 horas do dia. O período diurno representa as horas decorridas entre as 8h00 as 19h59 e o Período noturno das 20h00 às 7h59. Esta divisão corresponde à mudança de turnos dos médicos.

A segunda coluna, Período, é uma subdivisão do Período noturno e diurno. O período diurno foi subdividido em três períodos 8h-14h, 14h-17h e 17h-20h. O Período noturno foi subdividido em 20h-2h e as 2h-8h. Esta divisão foi baseada na experiência de alguns profissionais do hospital, correspondendo a períodos com mais ou menos afluência às urgências.

A granularidade escolhida para a Dimensão Minuto foi o minuto. A par da Dimensão Data, a Dimensão Minuto é uma *Role-Playing Dimension*. Permite ter numa mesma tabela física a visão das cinco perspetivas horárias que ocorrem durante um episódio de urgência (hora entrada, hora triagem, hora entrada no serviço de observação, hora saída do serviço de observação, hora saída).

Será criada uma hierarquia com quatro níveis: período horário-> período-> hora-> minuto.

Como esta dimensão não está dependente de fontes externas e não está previsto a mudança de períodos, esta será integralmente carregada de uma só vez. Assim é possível sabermos de antemão que a dimensão terá 1440 linhas (60*24, minutos vezes horas, respetivamente).

Esta dimensão, uma vez criada não sofre qualquer tipo de atualização.

5.3.3 Dimensão Idade

A Dimensão Idade representará a idade dos utentes na altura do episódio de urgência.

Para além da análise de utentes por idades, é também bastante comum a análise por faixa etária. Esta organização do atributo (idade) com uma gama de valores considerável, em bandas (faixas etárias), é usualmente designada por modelação em banda (*banding*). Esta abordagem permite-nos ter os dados previamente agrupados de forma progressiva em intervalos de tamanho variado, consoante a especificidade e contexto do valor numérico alvo, neste caso particular, a idade (Subsecção 6.2.3).

Desta forma, à Dimensão Idade correspondem os atributos: Idade e Faixa Etária. A idade é compreendida entre os zero e os 130 anos. A faixa etária foi subdividida por grupos etários relacionados. Esta subdivisão pode ser vista em mais detalhe na Subsecção 6.2.3. Será criada a seguinte hierarquia com dois níveis: Idade-> Faixa Etária.

Esta dimensão, uma vez criada, não é previsto sofrer qualquer tipo de atualização.

5.3.4 Dimensão Utente

A Dimensão Utente compreende os atributos: Número de Cartão de Utente, Número de Beneficiário, Número de Cartão de Saúde, Número de Documento Identificativo, Tipo de Documento Identificativo, Telefone Contacto, Data de Nascimento, Género e Isenção de Taxa Moderadora.

Dado que a informação disponibilizada pelo utente a cada episódio de urgência nem sempre possui os campos totalmente preenchidos, decidimos atualizar todos os campos à exceção da Isenção de Taxa Moderadora, como uma SCD tipo 1. Contudo esta substituição, inerente a esta técnica, é feita

de forma cautelosa, só sendo atualizados os campos que não tiverem sido preenchidos num episódio de urgência anterior. A exceção é o campo Telefone de Contacto que é sempre atualizado com o número mais recente.

Para Isenção de Taxa Moderadora foi utilizada SCD do tipo 2. Deste modo conseguimos acompanhar o estado de isenção do utente ao longo do histórico e ao mesmo tempo é possível ter o seu valor mais atual. Não haverá perda de informação relevante e, através da Data Inicio e da Data Fim, consegue-se saber quando um utente passou de isento a não-isento, e vice-versa.

A Dimensão Utente será carregada mensalmente. No caso de o utente já existir na dimensão, será atualizado. Caso haja alterações, aplicamos uma SCD tipo 1 e/ou SCD tipo 2, mediante o atributo que estivermos a atualizar.

Na Subsecção 6.2.4, será explicado de forma mais pormenorizada todo o processo de ETL da Dimensão Utente.

5.3.5 Dimensão Local

A Dimensão Local compreende os atributos: Distrito, Concelho e Freguesia. Esta dimensão permite-nos caracterizar os episódios de urgências por local de origem do utente.

Será criada uma hierarquia com os seguintes três níveis: Distrito-> Concelho -> Freguesia.

Esta dimensão será atualizada mensalmente, com carregamento dos novos registos.

5.3.6 Dimensão Causa

A Dimensão Causa tem unicamente o atributo Causa. Esta dimensão visa possibilitar a caracterização das causas do episódio de urgência no hospital (doença, acidente doméstico, agressão).

Esta dimensão será atualizada mensalmente, com carregamento dos novos registos.

5.3.7 Dimensão Prioridade

A Dimensão Prioridade compreende os atributos: Cor e Prioridade. Esta dimensão permitir-nos-á caracterizar os episódios de urgência pela prioridade atribuída na triagem.

A cada situação é atribuído um estado emergente, muito urgente, urgente, pouco urgente e não urgente (Prioridade). A cada uma destas prioridades corresponde uma cor: vermelho, laranja, amarelo, verde e azul, respetivamente.

Esta dimensão baseia-se no Sistema de Triagem de Manchester, no qual são definidas cinco situações de prioridade. Deste modo, depois de carregada, esta a dimensão não requer futuras atualizações.

5.3.8 Dimensão Origem

A Dimensão Origem compreende os atributos Origem e Unidade de Origem. Esta dimensão permite distinguir a origem dos utentes que vêm referenciados por um centro de saúde ou por um hospital, daqueles que vêm por iniciativa própria.

No caso de provirem de um hospital ou centro de saúde este é indicado na Unidade de Origem.

Esta dimensão será atualizada mensalmente, com carregamento dos novos registos.

5.3.9 Dimensão Destino

A Dimensão Destino compreende os atributos Destino e Unidade Destino. Esta dimensão permitirá identificar o destino dos utentes após a alta (internamento, exterior, outro hospital). No caso de o destino ser um centro hospitalar, centro de saúde ou hospital é identificada a unidade correspondente no atributo Unidade Destino.

Será atualizada mensalmente, com carregamento dos novos registos.

5.3.10 Dimensão Entidade Financeira Responsável (EFR)

Esta dimensão compreende os atributos Entidade Financeira Responsável e o Grupo EFR. Esta dimensão permitirá identificar a identidade responsável pelo pagamento. O Grupo EFR indica se a EFR é do Serviço Nacional de Saúde. Será atualizada mensalmente, com carregamento de novos registos.

5.3.11 Dimensão Elegibilidade

Esta dimensão compreende os atributos Internamento, Elegibilidade e Motivo. O atributo Internamento identifica se o urgente seguiu para internamento ou não. Caso tenha seguido para internamento, não é elegível para pagamento. O atributo Elegibilidade indica se o episódio de urgência foi considerado elegível para pagamento por parte do Hospital de Braga, ou não. O atributo Motivo caracteriza a razão da não elegibilidade.

Esta dimensão será atualizada mensalmente, com carregamento de novos registos.

5.3.12 Dimensão Serviço

A dimensão compreende os atributos Serviço Alta e Tipo de Urgência. Estes atributos permitem-nos caracterizar os episódios de urgências por especialidade ou tipo de urgência (Medicina interna, Obstetrícia, Pediatria) e o serviço que foi responsável pela alta (Serviço Alta).

Esta dimensão será atualizada mensalmente, com carregamento de novos registos.

5.3.13 Utilização de Dimensões Degeneradas

Uma Dimensão Degenerada é uma dimensão que não possui uma tabela de dimensão associada, sendo formada por apenas um único valor que é colocado na tabela de fatos [18]. Para além da existência de um atributo isolado poder indiciar a presença de uma dimensão degenerada, a existência de um atributo que cresce aproximadamente do mesmo modo que a tabela de factos pode ser um sinal de que estamos perante uma dimensão degenerada.

No Data Mart urgências, foram implementadas três dimensões degeneradas. As primeiras duas dizem respeito ao número de processo e ao número do episódio de urgência. Para além de caracterizarem o episódio de urgência e podendo ser vistas como *Unique Key*, crescem exatamente ao mesmo ritmo que a tabela de factos. Portanto, decidimos colocá-las na tabela de factos, como dimensões degeneradas.

A terceira dimensão degenerada é a referência. A referência indica se um utente que veio ao Hospital de Braga é de primeira linha ou não. Se for, então o utente foi ao hospital que a rede de referência estabeleceu. Caso seja 2ª linha, o utente foi a uma urgência hospitalar que não era o hospital da sua área de referência. O hospital de Braga só possui referência de 1ª e 2ª linha. Esta dimensão degenerada permitir-nos-á analisar até que ponto a rede de referência estabelecida se adequa à realidade existente. Dado não existirem mais atributos com baixa cardinalidade, resolveu-se tratar este atributo como uma dimensão degenerada.

5.4 Identificação de factos

Na quarta e última fase, definimos o tipo da tabela de factos. É também nesta fase que são definidos os tipos de métricas (agregadas, derivadas) e se categorizam as métricas (aditiva, semi-aditivas, não-aditiva). Por fim é nesta fase que definimos o tipo da função de agregação (soma, média, count, máximo) e o tipo de dados da métrica (inteiro, moeda, entre outros).

Deste modo, sempre que tivermos perante a implementação de um armazém de dados que tenha associadas métricas, temos opções para o tipo de tabela de factos: transacionais, de sumarização periódica, de sumarização acumulada [13].

O primeiro tipo é o mais utilizado na implementação de Data Warehouse e baseia-se no registo de factos num determinado momento. Cada linha da tabela de factos representa um evento de medição, podendo a tabela de factos ser mais ou menos densa, pois a existência de linhas na Tabela de Factos depende da frequência com que o evento ocorre.

Como cada linha da tabela factos internamentos e da tabela urgências corresponde a um episódio de internamento e urgência, num determinado momento, estas tabelas foram definidas como transacionais.

O segundo tipo de tabelas de factos é sumarização acumulada. Cada linha vai representar um caso concreto e de forma acumulada, sendo atualizada desde a criação até ao momento atual. Este tipo de tabelas de factos normalmente caracteriza-se por possuir várias chaves estrangeiras para várias dimensões data. Em grande parte das vezes é usada apenas uma única tabela física com várias vistas sobre a mesma dimensão data – *Role Playing Dimension*. Deste modo, cada registo na tabela de factos é recapturado várias vezes até ao ciclo de vida do facto, usualmente curto, ter acabado.

Esta tabela de factos permite a descrição do que acontece a um certo evento durante um determinado período de tempo. No caso particular da saúde, este tipo de tabelas poderia ser útil para o seguimento ao longo do tempo de um tratamento de um paciente de curta duração sem um período de tempo de tratamento previsível e regular. Ora esta abordagem não satisfaz as necessidades pretendidas com a tabela que nos falta caracterizar, tabela de facto de consulta externa, em que a granularidade da tabela corresponde a “fotografias” de períodos mensais (previsíveis e regulares).

A granularidade escolhida para a tabela de factos de consulta externa foi a sumarização periódica. Esta tabela é usualmente utilizada para períodos temporais (dia, mês, trimestre, ano). Estes intervalos de tempos são pré-definidos e com uma regularidade associada, como acontece por exemplo na análise de extratos mensais de contas bancárias, balanço de níveis de *stock*, e em particular na tabela mensal de lista de espera da tabela de facto da consulta externa do hospital. Este tipo de tabela de factos tem a particularidade de poder ser derivada da tabela transacional, o que ao contrário já não acontece.

Estando os tipos das tabelas de factos definidos podemos agora determinar métricas. Estas podem ser classificadas em três tipos: elementares, agregadas (sum, count, avg) e derivadas (calculadas).

De seguida, caracterizamos cada uma das métricas do Cubo:

Métricas elementares:

- Valor Cobrado (decimal);
- Valor Faturado (decimal).

Métricas derivadas:

- Número de episódios de urgências (count; inteiro);
- Tempo de permanência médio em horas (média; decimal);
- Tempo de espera médio em Horas (média; decimal);
- Idade média dos utentes (média; decimal).

Métricas agregadas:

- Percentagem por Género (divisão; decimal);
- Número de episódios no período homólogo (count; inteiro);
- Desvio face ao período homólogo (subtração; inteiro);
- Número de utentes distintos que recorreram aos serviços de urgência hospitalar (count; inteiro).

5.5 Matriz de barramento dimensional

Identificadas as dimensões e as tabelas de factos e para perceber de forma mais clara como cada dimensão comunica com cada Data Mart recorreremos à criação de uma Matriz de barramento dimensional, Tabela 5.1.

Esta matriz é um dos componentes dos processos da arquitetura defendida por Kimball. Usualmente, as linhas da matriz representam os Data Marts (processo de negócio) e as colunas representam todas as dimensões existentes.

Na Tabela 5.1 para facilitar a leitura esta ordem foi invertida.

	Fact Urgências	Fact Internamento	Fact Consulta Externa
Dim_Causa	X	X	
Dim_Origem	X		
Dim_Destino	X	X	
Dim_Local	X	X	X
Dim_EFR	X	X	X
Dim_Minuto	X	X	
Dim_Data	X	X	X
Dim_Utente	X	X	X
Dim_Idade	X	X	X
Dim_Serviço	X	X	
Dim_Elegibilidade	X	X	X
Dim_Prioridade	X		
Dim_Especialidade			X
Dim_Admissão		X	
Dim_GDH		X	

Tabela 5.1: Matriz do Data Warehouse em *Bus*

Depois de sinalizadas as dimensões que cruzam com cada processo de negócio, é possível verificar a existência de dimensões comuns entre os diferentes Data Marts.

Uma vez que este projeto se desenvolveu com base na arquitetura *Bus*, de Kimball, é necessário acautelar que haja comunicação entre os diferentes Data Marts. É neste ponto que se tira partido de uma das principais vantagens da elaboração da matriz de barramento. Com a definição das Dimensões Conformes e para que seja possível relacionar os diferentes Data Marts de modo a criar um Data Warehouse que permita uma visão global do

hospital, as dimensões comuns da matriz de barramento devem ser conformes. Para tal, deve-se assegurar que as chaves, nomes das colunas e os seus valores (em significado e tipo de dados) sejam consistentes em todos os processos de negócio a que foram associadas.

Esta matriz possibilita uma análise global dos serviços dos hospitalares. Estando as dimensões conformes identificadas, é possível implementar Data Marts a Data Marts com outro cuidado. Garante-se uma evolução mais segura e consistente.

6 Processo de implementação de um Data Mart

Para que as interpretações que se pretende obter com modelos analíticos permitam uma análise fiável do negócio, é preciso assegurar a integridade e coerência dos dados do Data Mart. O processo de ETL (*Extract, Transform and Load*) tem esse objetivo. É a partir das diferentes tarefas realizadas nestes processos que são moldados os dados para o carregamento do armazém de dados, neste caso particular, para o Data Mart Urgências.

6.1 Extração, Transformação e Carregamento

- Extração

A extração é a primeira tarefa do processo ETL e consiste em extrair os dados do sistema de origem. Esta extração pode ser feita em memória, diretamente do sistema ou a partir de uma área física especialmente criada para esse efeito (Staging Area).

Os dados das urgências hospitalares recebidos mensalmente em ficheiros Excel pela ARS-N serão assim armazenados numa base de dados Microsoft SQL Sever (*Staging Area*). É a partir desta base de dados que são feitos os processos de ETL para o Data Mart Urgências.

- Transformação

Após a extração dos dados, é necessário assegurar a sua qualidade, para que façam sentido no contexto do Data Warehouse

Existem várias transformações possíveis, desde remoção de duplicados, correção de erros, conversão do formato dos dados para garantir a consistência, entre muitas outras.

- Carregamento

Após a transformação, segue-se o carregamento de dados para a respetiva tabela de destino (tabela de dimensão ou de factos).

6.2 Processo de ETL das Dimensões

Para o processo de ETL, recorreremos ao SQL Server Integration Services (SSIS) e ao SQL Server Management Services (SSMS). O SSMS foi utilizado através de *scripts* T-SQL para a transformação e carregamento das dimensões Data, Minuto e Idade.

O ETL das restantes dimensões foi realizado recorrendo ao SSIS. Foi criado um fluxo de controlo (Control Flow) para controlar o fluxo das tarefas de ETL a serem executadas – fluxo de dados. É este fluxo de dados (Data Flow) que dá suporte ao mecanismo de movimento dos dados de uma origem para uma fonte destino. Ou seja, podemos ver no Data Flow as três etapas que caracterizam um processo de ETL. O primeiro e o último operador correspondem sempre à extração e carregamento. Os operadores intermédios representam as transformações realizadas. O ciclo completo de um Data Flow representa um processo de ETL.

Nas subsecções que se seguem, apresentamos os Data Flow criados para os diferentes processos de ETL.

6.2.1 Dimensão Data

O processo de transformação e carregamento da dimensão Data foi feito recorrendo ao SQL Server Management Services (SSMS) através de *scripts* T-SQL. No script, Anexo 1.1, é criada a tabela e feito o carregamento entre 2010 e 2018 (DATA_INICIO e DATA_FIM, respetivamente).

Neste caso particular, a chave primária da dimensão foi criada em consonância com a semântica subjacente à dimensão. Ou seja, à data 2010/01/01 associou-se a chave primária 20100101. Esta prática não só facilita o mapeamento na fase do carregamento da dimensão como facilita a divisão do Data Mart, na eventualidade de ser necessário no futuro.

No Anexo 1.1.1 e 1.1.2 apresentamos o código relativo à criação da coluna Feriado. Esta coluna é preenchida com 1, no caso de ser feriado, e 0 caso contrário. O carregamento desta coluna teve em consideração as alterações ocorridas nos feriados a partir de 2013.

Novas alterações que ocorram terão de ser atualizadas manualmente.

O facto de a dimensão Data poder ser vista como múltiplas divisões de níveis, de anos para semestre, de semestres para trimestres, trimestres para meses, e por aí em diante, permite-nos tirar partido das hierarquias. Estas constituem uma das características estruturais do OLAP. Esta técnica permite agrupar os dados numa dimensão em diferentes níveis de detalhe e perspetivas. Assim, fornece um caminho de navegação de dados resumidos para um nível mais detalhado (*drill-down*). O mesmo é possível de modo inverso, isto é, de nível com mais detalhe para um nível mais resumido (*drill-up*).

Esta prática, além de enriquecer as consultas por permitir agregações ou desagregações de diferentes pontos de vista do calendário, permite um melhor desempenho das pesquisas por parte dos utilizadores. Este melhoramento justifica-se por estas agregações serem pré-calculadas.

A Figura 6.1 apresenta a hierarquia e os diferentes níveis de agregação e desagregação criados na Dimensão Data.

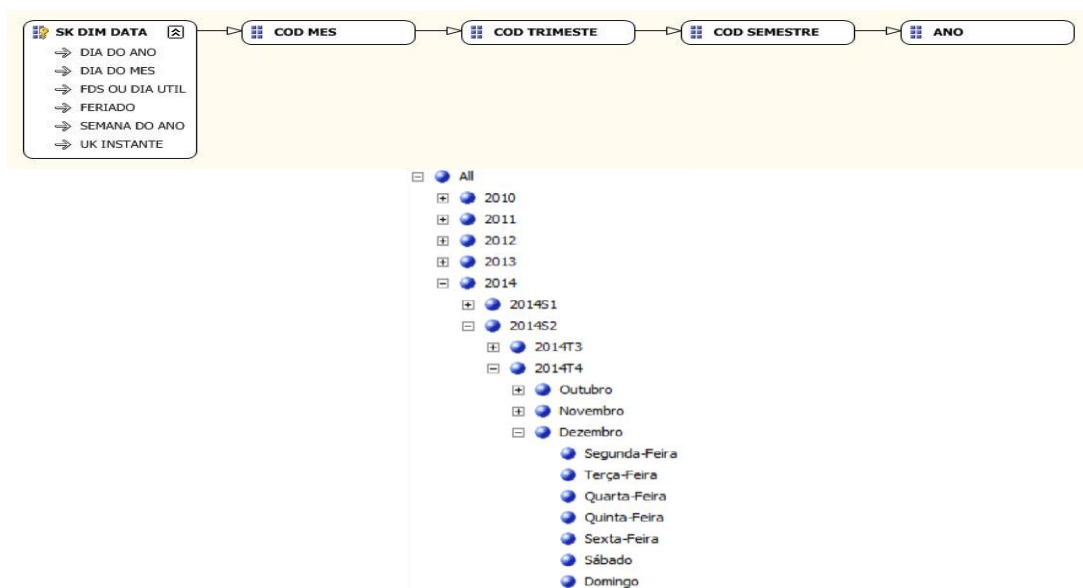


Figura 6.1: Hierarquia da Dimensão Data

6.2.2 Dimensão Minuto

O processo de transformação e carregamento da Dimensão Minuto também foi feito recorrendo ao SQL Server Management Services (SSMS) através de

scripts T-SQL. No Anexo 1.2 apresentamos o *script* relativo à criação da tabela e seu carregamento.

Para além das colunas minuto, hora e os respetivos códigos (*level key*) foi criada a chave da dimensão (SK_MINUTO) e duas colunas referentes a períodos horários.

A chave desta dimensão é um inteiro e está em consonância com a semântica subjacente à dimensão. Isto é, o instante 12h45 tem a chave 1245.

Os períodos horários foram divididos com base no funcionamento do Hospital de Braga, como se descreveu em detalhe na Subsecção 5.3.2.

Ao agruparmos os minutos em horas e estas últimas em períodos torna-se possível tirar partido da hierarquia. Assim foi criada a hierarquia minuto com quatro níveis. A Figura 6.1 mostra esquematicamente a hierarquia criada e um extrato exemplificativo.

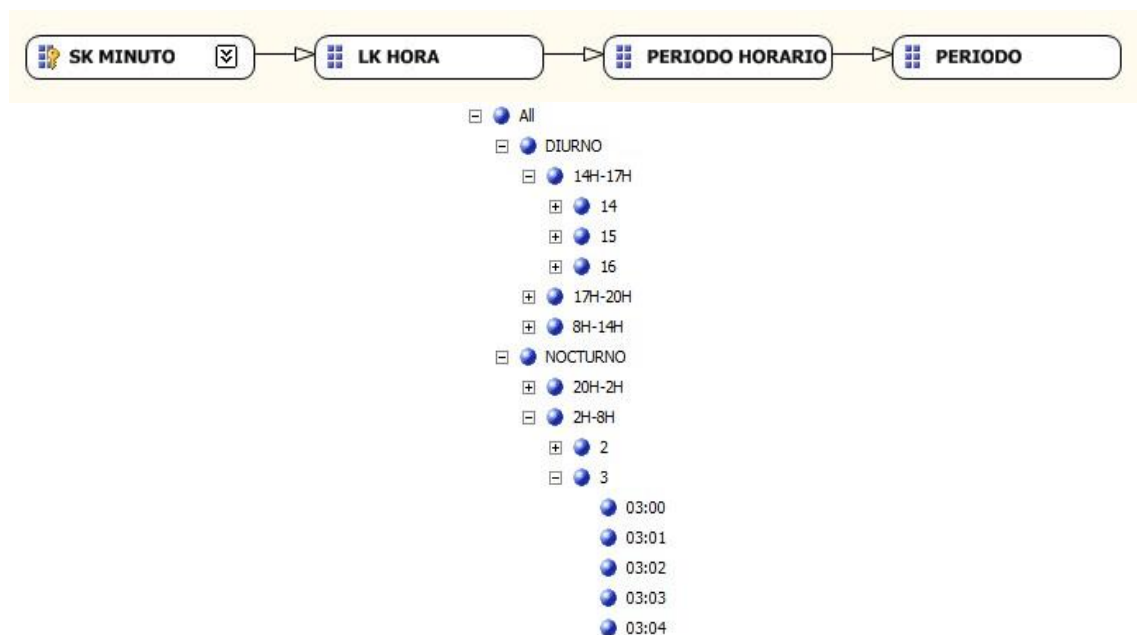


Figura 6.2 Hierarquia Minuto.

6.2.3 Dimensão Idade

O processo de transformação e carregamento da dimensão Idade foi implementado recorrendo ao SQL Server Management Services (SSMS), e consta do Anexo 1.3.

A dimensão foi carregada com 132 linhas: 131 linhas que correspondem às idades compreendidas entre os 0 e os 130 anos e, por fim, uma linha para

o caso de não estar disponível a data de nascimento. A este caso foi atribuído um valor numérico sem sentido no contexto da dimensão (999).

À semelhança das dimensões de tempo, a chave substituta foi criada em consonância com a semântica subjacente à dimensão. Ou seja, a linha correspondente a 1 ano de idade tem como a chave primária o inteiro 1, e assim por diante.

A Figura 6.3 apresenta a hierarquia que foi criada e as faixas etárias definidas.

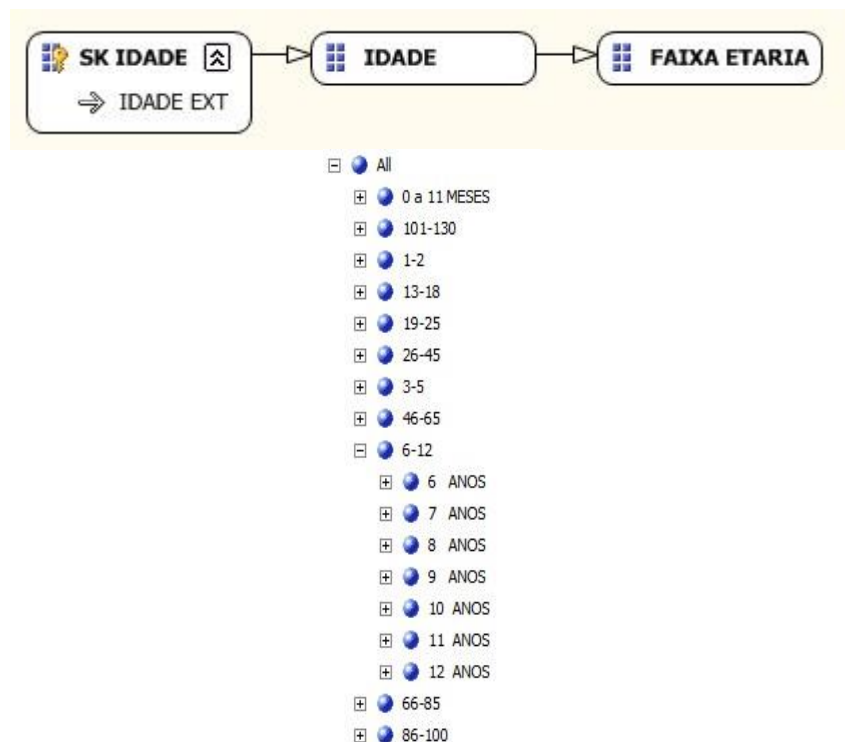


Figura 6.3: Hierarquia Idade

6.2.4 Dimensão Utente

O processo de ETL da Dimensão Utente foi realizado recorrendo apenas ao SQL Server Integration Services (SSIS). Antes de explicar todo o processo de ETL, iremos fazer uma contextualização do problema que tínhamos em mãos.

A primeira dificuldade detetada foi a existência de pacientes duplicados nos dados de origem. Na dimensão apenas pretendemos ter uma linha por paciente.

Um utente pode-se identificar através de um dos seguintes números:

Número de Cartão de Utente, Número de Beneficiário ou Número de Cartão de Saúde. Contudo, dado o campo Número de Cartão de Saúde estar pouco preenchido, utilizamos como identificador apenas os restantes dois números.

Desta forma, será a partir do Número de Cartão de Utente e do Número de Beneficiário que tentaremos fazer uma filtragem dos utentes, para que na Dimensão Utente se tenha uma linha por utente. Ou seja, há que assegurar que não haja números de Cartão de Utente ou de Beneficiário duplicados.

Para resolver esta questão resolvemos tratar um caso de cada vez, começando primeiro pelo Número de Cartão de Utente, pelo facto de ser o número mais preenchido.

A Figura 6.4 corresponde ao processo de ETL de remoção de Número Cartão de Utente em duplicado. De seguida, iremos explicar a abordagem inicial e o procedimento implementado neste processo de ETL, descrevendo em detalhe as transformações realizadas.

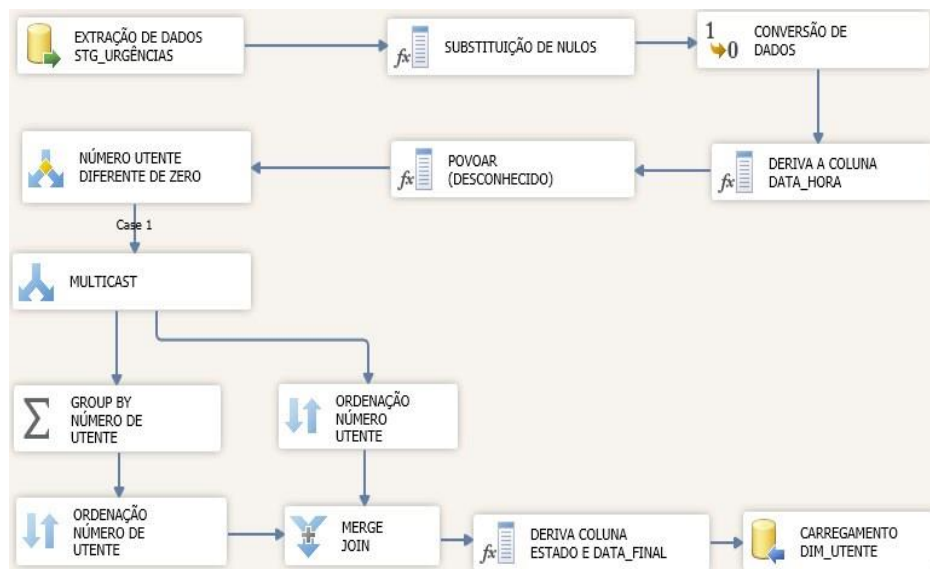


Figura 6.4: Data Flow Dimensão Utente - Número de Cartão de Utente.

Numa primeira abordagem, tentamos usar o processo *Sort*, com a opção de remoção de duplicados por Número de Cartão de Utente. Contudo este procedimento não nos garantia que, para cada Número de Cartão de Utente, se obtivesse a sua primeira ida às urgências. No caso de ser descartado um episódio mais antigo que tivesse mais dados preenchidos, tal podia levar a perdas de informação ou de atualizações relevantes.

Sendo assim, a abordagem seguida foi obter para cada Número de Cartão de Utente a sua primeira ida às urgências. Para tal utilizámos o processo *Agregate*, com agregações (*Group By*) por Número de Cartão de Utente e com a Data e Hora de entrada mais antiga (*Minimum*). Como a função *Agregate* apenas devolve os valores utilizados (neste caso Número de Cartão de Utente, Data de Entrada e Hora de Entrada) e o nosso objetivo é ter os restantes dados na dimensão, utilizamos o *Multicast* para fazer uma cópia dos dados provenientes do *Conditional Split*. De seguida, com o *Merge Join*, é feito um *Inner Join*, onde os dados originais que fizerem *match* com Número de Cartão de Utente, Data e Hora provenientes do *Agregate* são retornados. Dado a função *Merge Join* exigir um *Sort* dos dados, é efetuado previamente um *Sort Ascending* por Número de Cartão de Utente, Data e Hora.

De seguida, são derivadas duas colunas para cada linha provenientes do *Merge Join*. À primeira é atribuído o Estado 1 e à segunda atribuída uma Data Final com o valor 99991231. Estas colunas são duas de três colunas inerentes à implementação de SCD do tipo 2 (Subsecção 2.5.3). A terceira coluna é a Data de Início, a qual é derivada no quarto processo, *Derived Column*, à qual é atribuída a Data de Entrada do episódio de urgência. No terceiro Data Flow desta secção será explicado de forma mais detalhada o significado destas colunas.

Por fim, é criada a Dimensão Utente com o processo OLD DB Source, com um *autonumber* para a chave primária SK_DIM_UTENTE e as colunas da dimensão. Para além da criação da dimensão, este processo permite o seu carregamento através da realização das correspondências (*match*) entre as colunas provenientes do fluxo de dados e as colunas da Dimensão Utente.

No Data Flow anterior (Figura 6.4) resolvemos a questão relativa ao Número de Cartão de Utente. No Data Flow da Figura 6.5 resolvemos a questão do utente se poder identificar com o Número de Beneficiário. A abordagem seguida foi carregar os utentes em que o Número de Cartão de Beneficiário é diferente de zero e que ainda não estão na dimensão.

Para tal, o procedimento seguido é similar ao Data Flow anterior, exceto no operador *Lookup*. Este operador verifica, para cada episódio de urgências

proveniente do *Merge Join*, se o Número de Cartão de Utente tem correspondência com algum Número de Cartão de Utente dos utentes, na Dimensão Utente (utentes já carregados no Data Flow anterior). Se houver correspondência, dada a opção escolhida na *Lookup* ser No Match, este não é carregado, significando que o utente já lá está. Este processo é realizado de forma a não termos utentes duplicados na dimensão.

Caso contrário, ou seja, se não houver correspondência entre os Número de Cartão de Utente, o utente ainda não está na dimensão e, portanto, é carregado. Este último caso, significa que se trata de um episódio em que o utente apenas forneceu o Número de Cartão de Beneficiário e que esse número de beneficiário não pertence a nenhum utente que possa ter sido já carregado no Data Flow anterior.



Figura 6.5: Data Flow Dimensão Utente - Número de Beneficiário.

Estando agora a Dimensão Utente já carregada sem utentes repetidos, o procedimento que se seguiu foi atualizar as alterações dos utentes através da técnica de *Slowly Changing Dimensions (SCD)*.

Como se verifica na Subsecção 5.3.4, foi decidido a implementação de SCD do tipo 1 e do tipo 2 para dimensão Utente. Para tal, começamos por implementar a SCD do tipo 2 e depois SCD do tipo 1. Assim, o Data Flow que se mostra na Figura 6.6 é responsável pela atualização dos dados, implementando a SCD do tipo 2.



Figura 6.6: Data Flow Dimensão Utente – SCD tipo 2

O que se pretende com a aplicação da SCD do tipo 2 é preservar o histórico dos dados relativamente à mudança de isenção para manter o valor mais atual dessa isenção. O procedimento seguido usa o operador *Group By* para agregar por número de cartão de utente e pelo máximo da coluna derivada *Nova_Data_Hora*. Esta coluna é um inteiro derivado da junção da data de entrada do episódio de urgências com a hora de entrada. O máximo desta coluna por cartão de utente corresponde, para cada número de cartão de utente, à sua última ida às urgências. Tendo este fluxo de dados é utilizado o operador *Lookup*, com a opção - *no match output* - por isenção, ou seja, se a isenção do fluxo de dados não tiver correspondência (não for igual) à isenção que está na dimensão, significando que a isenção mudou e portanto “passa” para o operador seguinte. Caso contrário - *match output*, significa que a isenção teve correspondência, ou seja, a isenção do fluxo é igual à da dimensão e, portanto, não houve alteração, “não passando” para a operação seguinte.

Tendo agora o fluxo de dados apenas episódios de utentes em que se verificou alteração da isenção, utilizou-se o operador OLE BD Command. Este operador permite executar instruções SQL para cada linha do fluxo de dados, tais como *insert* e *update*.

Com este operador foi feito um *update* para colocar a 0 o atributo “estado” da linha do utente que está na dimensão. Tal significa que esta linha está desatualizada, sendo alterado o atributo *Data_Fim*, que inicialmente é carregado com uma data sem significado no contexto (99991231) e agora

passa a tomar o valor da data de entrada do fluxo de dados, isto é, da data de entrada do episódio no qual se verificou a alteração do estado de isenção. Esta data, apesar de não ser muito rigorosa, visto não significar exatamente que foi nessa data que a isenção do utente sofreu a alteração, permite-nos pelo menos saber que a partir desta data o utente passou a isento ou não isento, consoante o tipo de alteração que se verifique.

Estando as linhas dos utentes que “sofreram” atualizações da isenção desativadas (estado igual a 0), é derivada para cada linha do fluxo de dados uma Data_Inicio. Esta data corresponde à data de entrada do episódio de urgência. Também é derivada uma Data_Fim com o valor 99991231, sem significado no contexto e, por fim, a coluna Estado com o valor 1, significando que a linha está ativa, ou seja, é a linha mais atualizada. Derivadas estas três colunas, é carregada uma segunda linha com exatamente a mesma informação que a linha que já existe na dimensão utente, apenas com as alterações de isenção (valor atualizado), Data_Inicio, Data_fim e Estado com os valores enunciados acima. Desta forma, quando uma atualização da isenção ocorre, ficamos com mais uma linha igual onde podemos verificar quando é que um determinado utente esteve isento e passou a não isento, e vice-versa.

O Data Flow que se segue, Figura 6.7, é responsável pela atualização dos dados implementando a técnica SCD do tipo 1.

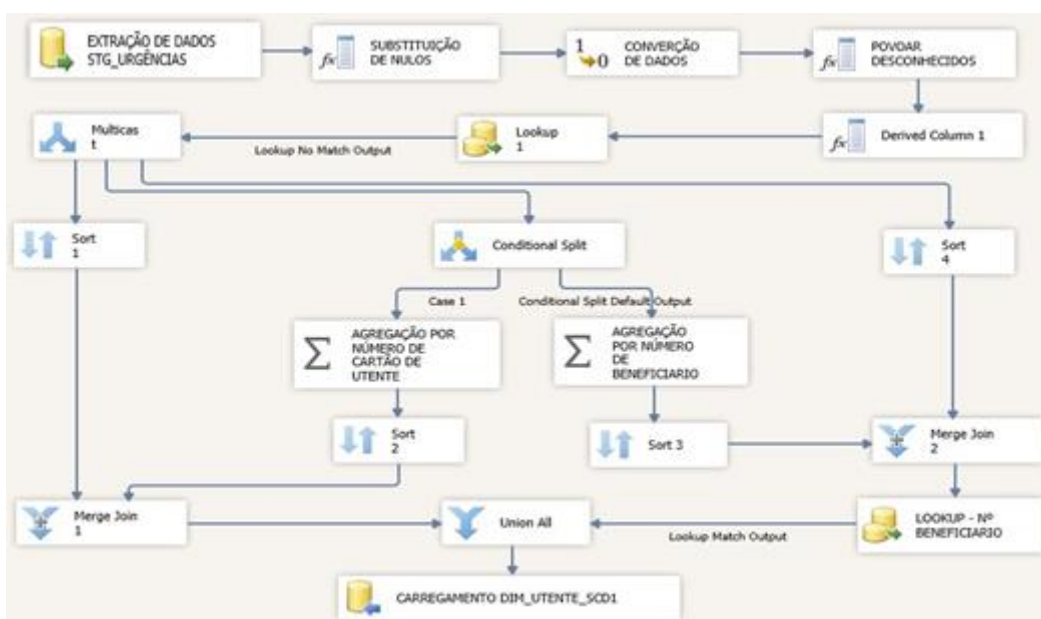


Figura 6.7: Data Flow Dimensão Utente – SCD tipo 1.

São atualizados os dados do utente sempre que estes não tenham sido preenchidos no primeiro carregamento do utente. Os atributos que são atualizados com a SCD do tipo 1 incluem: Número de Cartão de Utente, Número de Cartão de Saúde, Número Beneficiário, Tipo Documento Identificativo, Número do Documento, Telefone Contacto, Data de Nascimento e Género.

O procedimento é semelhante ao Data Flow anterior, relativo à SCD do tipo 2, à exceção do primeiro operador *Lookup* em que se verifica, para cada um dos atributos anteriormente indicados, se ocorreu alguma alteração dos dados (conforme consta da tabela no Anexo 2.1.).

Com esta *lookup* “passaram” no fluxo todos os dados que sofreram pelo menos uma alteração num dos atributos que se pretende atualizar com esta SCD do tipo 1. Contudo, esta alteração só significa que os valores provenientes do fluxo de dados são diferentes dos valores que estão na dimensão Utente. Por isso, este fluxo de dados é carregado numa tabela auxiliar. O Data Flow que se segue, Figura 6.8, através da *Lookup* garante que para cada atributo só é feita a alteração se esta fizer sentido.

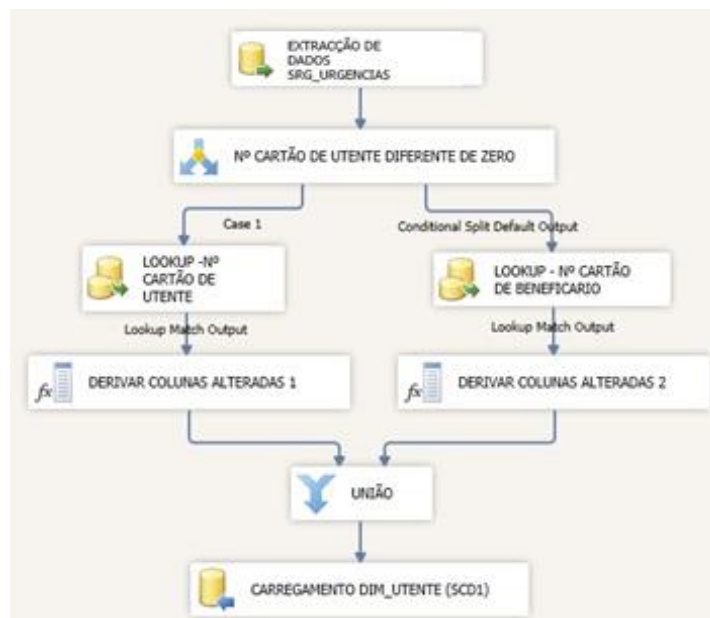


Figura 6.8: Data Flow Dimensão Utente – SCD tipo 1 (*update*).

A título de exemplo, o Número de Cartão Utente é preenchido com o valor zero quando não foi fornecido. Assim, só será atualizado o Número de Cartão de Utente se o Número de Cartão de Utente proveniente do fluxo de dados

é diferente do atual Número de Cartão de Utente, que está na dimensão e ao mesmo tempo se este for diferente de zero. Ou seja, neste caso, estamos perante um utente que na sua primeira ida às urgências não forneceu o Número de Cartão de Utente e, portanto, foi inicialmente carregado com o valor zero e, numa segunda ida às urgências, já forneceu o Número de Cartão de Utente uma vez que este é diferente de zero. Neste caso é “retornado”, o valor atualizado do Número de Cartão de Utente. Caso contrário é “devolvido” ao fluxo de dados o valor que estava, significando que não houve atualização.

Estas condições podem ser vistas com mais detalhe para todos os atributos a atualizar na tabela que consta do Anexo 2.1.

Deste procedimento obtemos os atributos atualizados, que são carregados numa segunda tabela auxiliar.

De seguida, com o operador *Script Task*, é feita uma atualização de todos os dados que estão na segunda tabela auxiliar por substituição pelos dados que estão na Dimensão Utente. A *query* de atualização pode ser consultada com mais pormenor no Anexo 2.1.

Tendo os dados atualizados, o ciclo do processo de ETL da Dimensão Utente termina fazendo um *truncate* (remoção de todas as linhas de uma tabela) com o operador *Script Task*. Assim, as duas tabelas auxiliares estarão vazias quando se for fazer o carregamento e atualização dos dados do novo mês.

6.2.5 Dimensão Local

Para o carregamento da Dimensão Local, foi realizado um *Sort* com remoção de duplicados e atribuído um inteiro *autonumber* para cada distrito carregado na STG_Distrito. A atualização desta tabela depois do *Sort* é feita com o operador *Lookup* com condição - *no match output*, de modo a só serem carregados os distritos que ainda não estejam na STG_Distrito. Proceder-se do mesmo modo para a Freguesia e Concelho.

Depois de obtermos as três tabelas com os códigos e nomes para cada freguesia, Concelho e Distrito, fazemos uma *Lookup - Match output* – para estabelecer a correspondência entre a Freguesia proveniente do fluxo de dados com a tabela STG_Freguesia, previamente carregada com os códigos.

Para as Freguesias que fizerem *match* é devolvido a Freguesia e o código respetivo, como se pode ver na Figura 6.9.

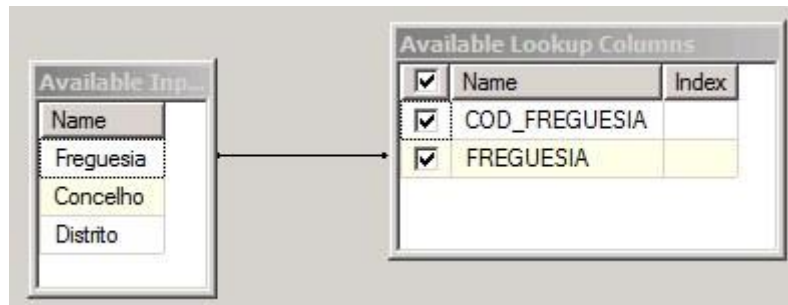


Figura 6.9: Lookup Freguesia.

O mesmo procedimento é seguido para Concelho e Distrito, nas respetivas *Lookup's*, como mostra a Figura 6.10.



Figura 6.10: Data Flow Dimensão Local

Segue-se um *Sort* com remoção de duplicados por Distrito, Concelho e Freguesia. E, por fim, antes do carregamento na Dimensão Local, são derivados novos códigos de Freguesia e Concelho através da seguinte fórmula (respetivamente):

$COD_CONCELHO * 10000 + COD_FREGUESIA$

$COD_DISTRITO * 1000 + COD_CONCELHO$

Esta reatribuição de códigos deve-se ao facto de existir o mesmo nome de freguesia em diferentes concelhos. Tal é problemático, na criação de hierarquias, pois o código de uma mesma freguesia ficaria associado a dois concelhos diferentes. Por fim, é criada a Dimensão Local através do operador OLD DB Source, com um *autonumber* como chave primária (SK_LOCAL) e

são carregados os Códigos e as Freguesias, Concelhos e Distritos do *match* entre o fluxo de dados e a Dimensão Local acabada de criar. Um exemplo desta correspondência pode ser observado na Figura 6.10.

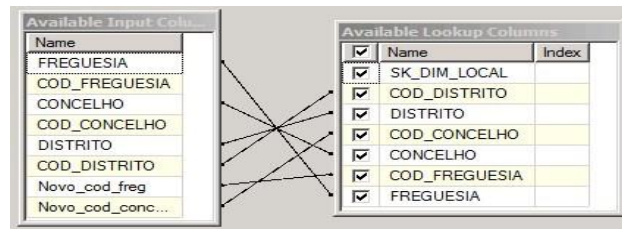


Figura 6.11: Lookup Dimensão Local.

No que diz respeito a atualizações da dimensão, o procedimento é semelhante, com o acréscimo do operador *Lookup* de que permite a deteção de novos casos, antes do carregamento para a Dimensão Local, como é possível ver na Figura 6.11. Esta *Lookup* é realizada entre o fluxo de dados e a Dimensão Local. Os códigos e nomes das Freguesias, Concelho e Distrito que não fizerem *match* com os dados carregados na Dimensão Local representam novos “locais”, e desta forma são carregados.

Na Figura 6.12 apresentamos a hierarquia e um exemplo ilustrativo da hierarquia Distrito, Concelho, Freguesia criada no SQL Server Analysis Services (SSAS).

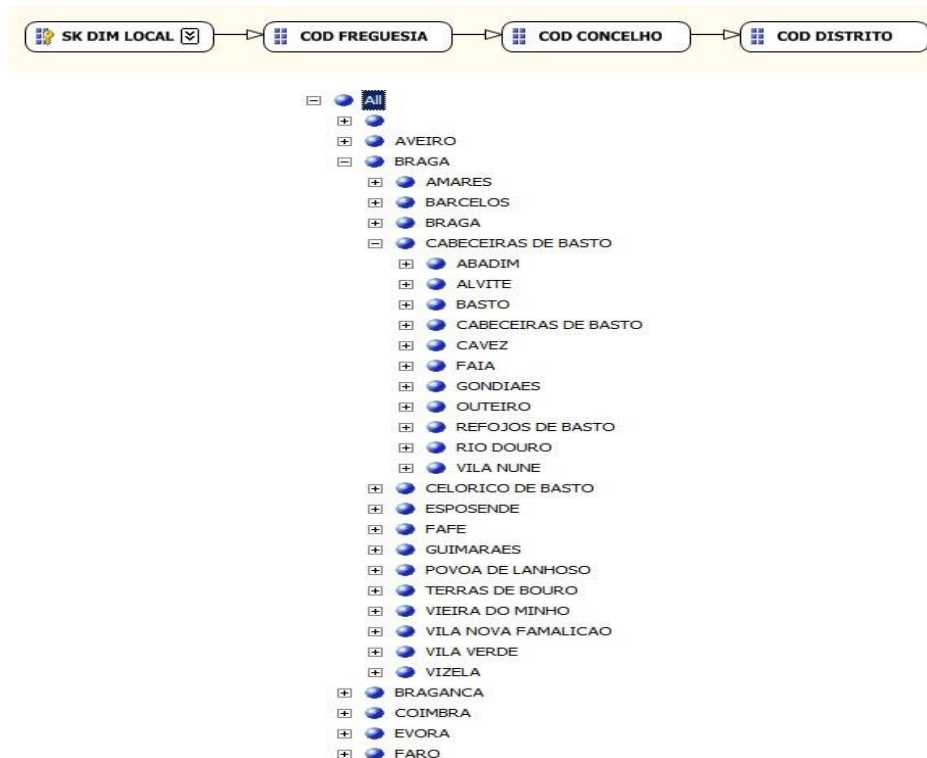


Figura 6.12: Hierarquia Dimensão Local.

6.2.6 Dimensão Causa

Para o carregamento da Dimensão Causa criou-se a tabela DIM_CAUSA, com um inteiro *autonumber* como chave primária (SK_CAUSA). A extração é feita a partir da *Staging Area* inicialmente através de um *Sort*, com remoção de duplicados. De seguida, são carregadas na dimensão as causas provenientes do *Sort*. Na atualização mensal desta tabela, é repetido o *Sort* anterior, seguido de uma *Lookup*. Esta *Lookup - no match output* - verifica, para cada nova causa, se esta existe na tabela da dimensão já carregada. As causas que não tiverem correspondência na Dimensão Causa representam novas causas, sendo então carregadas (Figura 6.13). Este processo é feito de forma similar para a Dimensão Serviço, Prioridade, Origem, Destino, Elegibilidade e EFR (entidade financeira responsável).



Figura 6.13 – Data Flow Dimensão Causa.

6.3 Carregamento da Tabela de Factos

O ETL da tabela de factos começa com a extração dos dados da *Staging Area*. No primeiro *Derived*, Figura 6.14, são calculadas as chaves estrangeiras SK_DATA's, SK_IDADE's e derivadas três colunas com os *timestamp* Data_Serviço_observação, Data_hora_entrada e Data_hora_saída. Estas derivações são necessárias para calcular o tempo de espera e de permanência (métricas derivadas). De seguida são convertidos para numérico os números de episódio e de processo. No segundo *Derived* são calculados os SK_HORAS's e os tempos de permanência e de espera. Segue-se a substituição por 0 dos *Null's* da

dimensão degenerada (Referencia) e das colunas Valor Cobrado e Valor Faturado (métricas elementares).

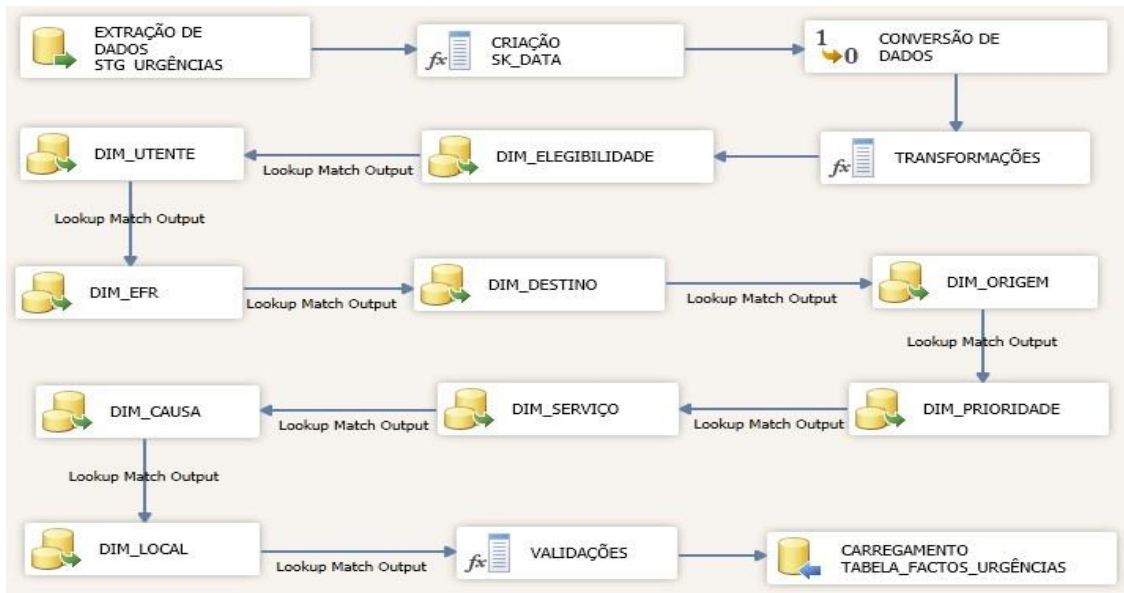


Figura 6.14: Data Flow Carregamento da Tabela de Factos.

Depois é efetuada uma *Lookup* às dimensões: Elegibilidade, Utente, EFR, Destino, Origem, Prioridade, Serviço, Causa e Local.

Estas *Lookup's* visam a obtenção das chaves primárias destas dimensões.

A título de exemplo, na Figura 6.15 é possível observar o *match* estabelecido na *Lookup* Destino. Para cada Destino e Unidade Destino proveniente do fluxo de dados que fizer *match* com o Destino e Unidade Destino da Dimensão Destino é devolvido o SK_DESTINO, para posteriormente ser carregado na tabela de factos de forma a estabelecer a ligação entre a Dimensão Destino e a tabela de factos.



Figura 6.15: *Lookup* Dimensão Destino – Tabela de Factos.

À exceção da Dimensão Utente, as chaves primária das dimensões (chaves estrangeiras da tabela de facto), são obtidas das chaves primárias das

dimensões seguindo o mesmo procedimento apresentado para a Dimensão Destino.

Para obter a chave primária da Dimensão Utente, utilizou-se a opção *advanced* da *Lookup* para construir a condição de *match*. Esta condição pode ser consultada com mais pormenor no Anexo 3.1. Esta condição deve-se ao facto da Dimensão Utente ser uma *Slowly Changing Dimension* do tipo 2 (SCD2). Quando o atributo de isenção de um determinado utente for alterado, essa atualização é feita acrescentando uma nova linha para o mesmo utente. Para fazer a correspondência, desta nova linha à tabela de factos, a *Lookup* iria encontrar para o mesmo Número de Cartão de Utente duas linhas, a antiga e a nova que foi criada com a SCD2. Esta condição permite salvaguardar que a *Lookup* faz *match* com a linha mais recente, atualizada pela SCD2.

O último *Derived* é responsável pelas validações. Estas validações são esquematizadas e descritas nos Logical Data Map que estão no Anexo 3.1.

Por fim, é criada a tabela de factos e carregados os registos na base de dados (dm). A Figura 6.16 ilustra um excerto da correspondência entre o fluxo de dados e a respetiva tabela de factos.

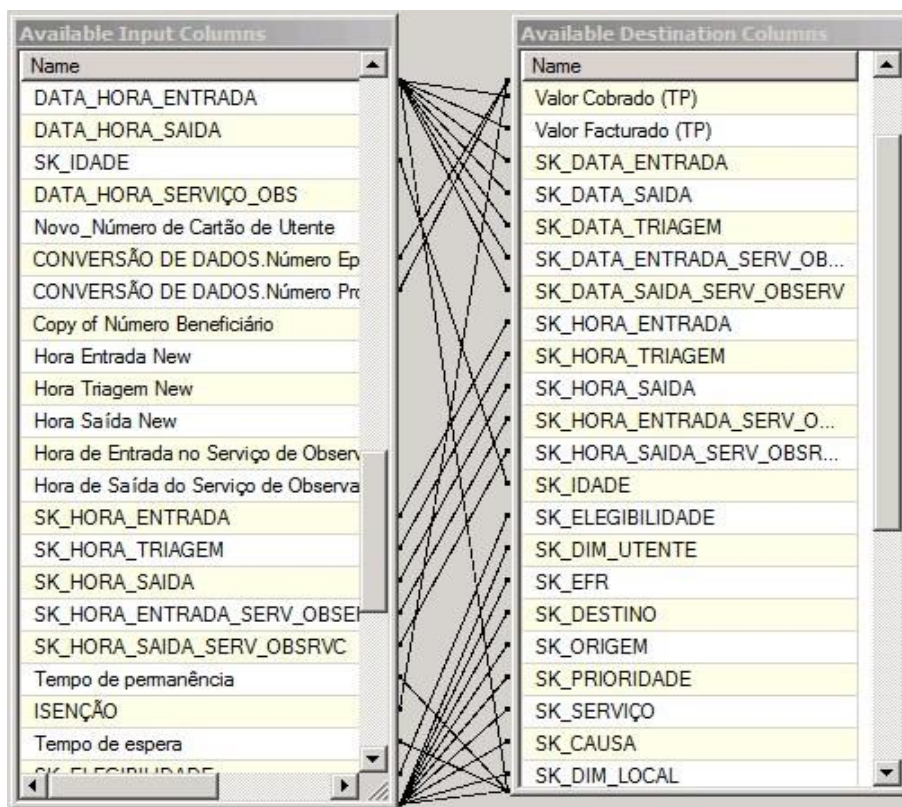


Figura 6.16: *Lookup* de carregamento da Tabela de Factos.

6.4 Cubo

Depois de todas as dimensões e a tabela de factos estarem carregadas, o passo que se seguiu foi projetar o Data Mart no SSMS, através da ferramenta de Desenho de Base de Dados. Na Figura 6.17 é possível verificar a estrela criada no SSMS.

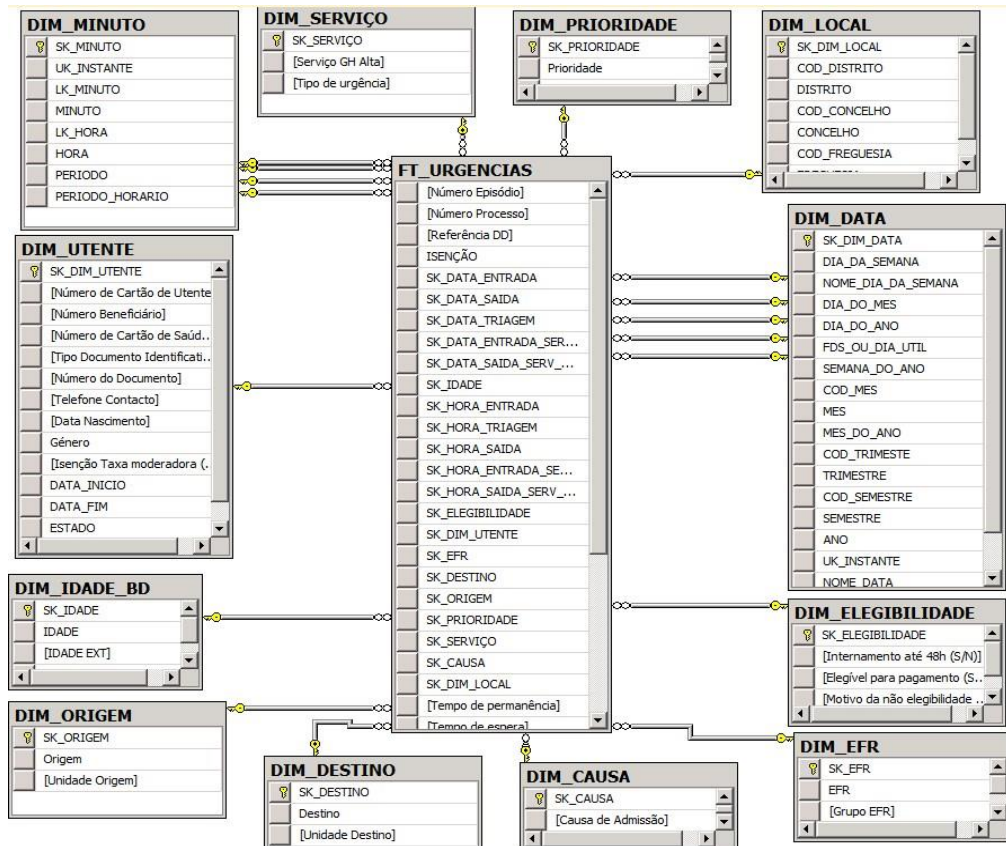


Figura 6.17: Esquema em Estrela (SSMS).

No Designer de Base de Dados, ao criar a estrela foram editadas as tabelas, identificadas as chaves primárias e as respetivas chaves estrangeiras. Uma vez que o Data Mart Urgências tem uma estrutura desnormalizada (apoiada num esquema em estrela), este tipo de estrutura de dados necessita ser guardado numa base de dados multidimensional.

Para implementação de um cubo multidimensional existem alguns passos que necessitamos analisar: o tipo de armazenamento MOLAP (ou ROLAP), as dimensões que participaram na estrela, as hierarquias a implementar, as

métricas que o cubo pretende calcular, os *KPI's* que se pretende desenvolver, entre outros.

Assim, e antes de prosseguirmos à criação do cubo, importa clarificar o tipo de armazenamento que iremos utilizar.

Multidimensional Online Analytical Processing (MOLAP) é um ambiente OLAP suportado numa base de dados multidimensional, como é o caso do SQL Microsoft Analysis Services (SSAS). *Relational Online Analytical Processing* (ROLAP) é um ambiente OLAP suportado por uma base de dados relacional.

Dado que o nosso modelo dimensional foi implementado usando um esquema em estrela (tabela de factos rodeada de um conjunto de tabelas de dimensões que formam uma estrela – *Star Schema*) será armazenado numa base de dados multidimensional. Este tipo de esquema desnormalizado permite um maior desempenho nas consultas, uma vez que o número junções a realizar entre as tabelas é menor sendo as junções menos complexas.

Neste tipo de base de dados, a informação é armazenada em cubos multidimensionais, onde todos os cálculos e combinações possíveis entre os atributos são pré-calculados quando o cubo é criado. Desta forma os dados são facilmente acessíveis, o que se traduz num alto desempenho na sua análise. A baixa escalabilidade é uma das desvantagens deste modelo. Contudo, esta pode ser acautelada no momento da implementação do cubo, através da inclusão apenas do resumo dos cálculos, ou seja, criação do cubo de forma mais detalhada [15].

Uma vez que apenas temos um Data Mart, recorrendo ao SQL Server Analysis Services (SSAS), a estrutura física das tabelas de dimensões e facto é relativamente fácil de escolher. Na Figura 6.18 podemos ver o cubo criado no SSAS.

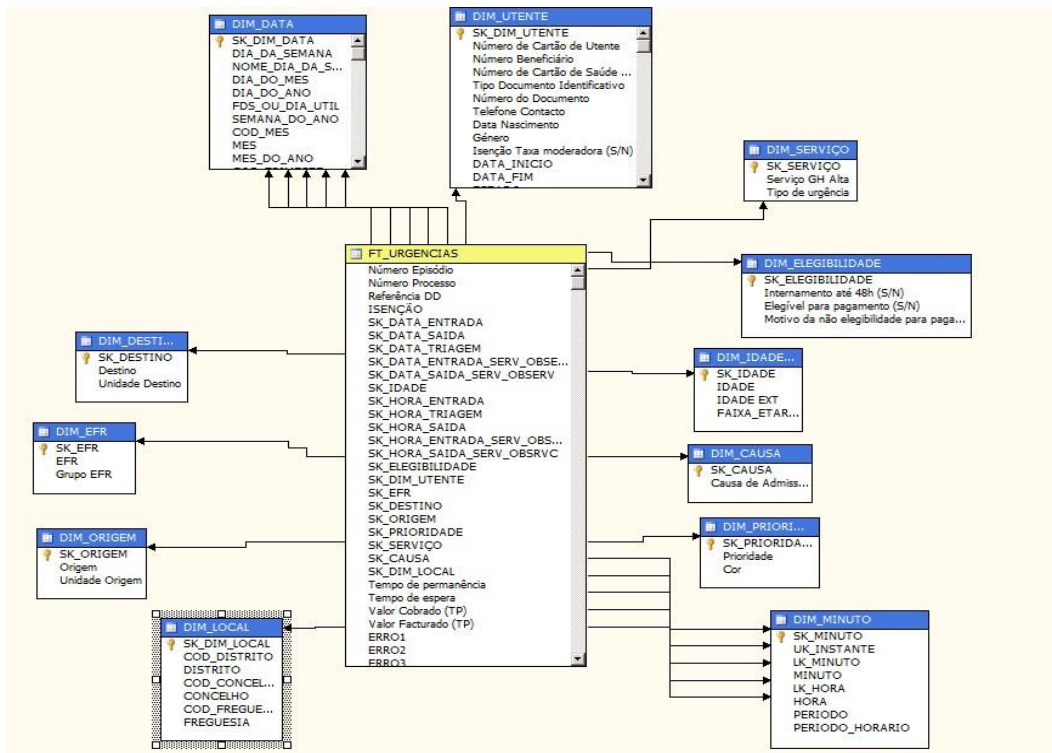


Figura 6.18: Esquema em Estrela (SSAS).

As hierarquias, que nos possibilitam tirar vantagem de operações como *drill-down* e *roll-up*, foram também criadas recorrendo ao SSAS, após a implementação do cubo. Contudo, por uma questão organizacional, estas hierarquias foram apresentadas na Secção 5.3.

6.5 Métricas Derivadas

No cubo e recorrendo ao SQL Server Analysis Services (SSAS) foram construídas as seguintes métricas derivadas:

Tempo de permanência Médio em horas	([Measures].[Tempo De Permanência]/[Measures].[FT URGENCIAS Count])/60
Tempo de espera Médio em Horas	([Measures].[Tempo De Espera]/[Measures].[FT URGENCIAS Count])/60
Período Homologo	([Measures].[FT URGENCIAS Count],PARALLELPERIOD([SK DATA SAIDA. [Hierarquia_data] .[ANO],1,[SK DATA SAIDA. [Hierarquia_data]. CURRENTMEMBER))
Período Anterior	([Measures].[FT URGENCIAS Count],[SK DATA SAIDA].[Hierarquia_data].PREVMEMBER)
Período Anterior Validação 1	([Measures].[ERRO1],[SK DATA SAIDA].[Hierarquia_data].PREVMEMBER)
Média das Idades	[DIM IDADE BD].[SK IDADE]/[Measures].[FT URGENCIAS Count]
Utentes Distintos	DISTINCTCOUNT(CROSSJOIN([DIM UTENTE].[Número De Cartão De Utente].MEMBERS,[Measures].[FT URGENCIAS Count]))
Percentagem por Género	[Measures].[DIM UTENTE].[Género]/[Measures].[FT URGENCIAS Count]

Estas métricas foram desenvolvidas em *Multidimensional Expressions*, popularmente conhecidas por MDX. Esta linguagem, introduzida pela primeira vez pela Microsoft em 1997, permite a consulta e manipulação de dados multidimensionais armazenados em base de dados OLAP.

6.6 KPI's

Dos KPI's abordados na Secção 4.3, apenas foi implementado o KPI referente à percentagem de episódios de urgência em que o tempo total de

permanências na instituição foi superior a 4 horas. A meta desse KPI é não ultrapassar os 10% em nenhum mês.

kpi	<pre>CASE WHEN (KPIVALUE("KPI Permanencia_Menor_4_Horas")/ KPIGOAL("KPI_DURACAO_MENOR_OU_IGUAL_4H"))>1 THEN 1 ELSE CASE WHEN (KPIVALUE("KPI_DURACAO_MENOR_OU_IGUAL_4H")/ KPIGOAL("KPI_DURACAO_MENOR_OU_IGUAL_4H"))>0.8 THEN 0 ELSE -1 END</pre>
-----	---

7 Apresentação de Resultados

Após a implementação do Data Mart Urgências no SSAS, bem como o desenvolvimento de algumas hierarquias, métricas e indicadores procedeu-se à apresentação de alguns resultados baseados nos dados já carregados. Até ao momento, os dados carregados refletem apenas o mês de Janeiro de 2014. No entanto, salvaguardando a privacidade dos utentes, é-nos possível mostrar alguns exemplos de matrizes, gráficos e dashboards desenvolvidos. Esta apresentação foi suportada pelo Excel 2007, a partir do qual é acedido o Cubo Urgências (SSAS).

Esta apresentação deve ser entendida como uma prova da possível usabilidade e do tipo de análise que o Sistema de Business Intelligence desenvolvido possibilita efetuar (não tendo as métricas sido sujeitas a uma avaliação criteriosa).

7.2 Período Horário

A Figura 7.2 apresenta duas matrizes e dois gráficos com diferentes comparações relativas ao período horário.

A matriz da esquerda compara o número de urgências por Período Horário (Diurno/Noturno) em cada dia da semana. Apresenta também o total global por dia. É interessante verificar que há uma redução bastante significativa do número de urgências ao fim de semana. Contudo, dado esta análise ser referente a um único mês é-nos difícil assegurar com rigor outro tipo de conclusões.

Porém este tipo de matriz poderá ser útil na reorganização estrutural das equipas de trabalho do hospital, quando o Data Mart possuir mais dados.

A matriz da direita relaciona o Período Horário com o tempo de permanência em horas do utente na urgência. É também visível o número de urgências. Nesta matriz é feito o *drill-down* do Período Horário em Horário e este nas diferentes horas. Esta abordagem permite-nos tirar partido da hierarquia criada, enriquecendo o poder da análise.

Nesta matriz, a título de exemplo, vemos que, apesar do número de urgências ser consideravelmente inferior no Período Noturno, o tempo de permanência é relativamente superior, atingindo o seu pico, neste mês, às ■ horas (com um tempo médio de permanência de ■ horas). Este ponto é atingido na hora de menor número de urgência ao longo do dia.

Por fim, é possível verificar que o tempo médio global de permanência de um utente numa unidade de urgência é de ■, ou seja, não é superior a ■ hora (■ horas é a uma diretiva defendida pelas entidades competentes, de tempo máximo aceitável de permanência numa urgência hospitalar). Contudo, o uso de médias podem não ser bom estimador.



Figura 7.2: Dashboards Período Horário.

7.3 Prioridade

No contexto das urgências hospitalares é importante perceber se a prestação de serviço do hospital é realizada de forma célere e atempada, bem como se as urgências se concentram nas situações a que verdadeiramente são vocacionadas: emergente, muito urgente e urgente.

Desta forma na Figura 7.3 são apresentadas duas matrizes relativas à prioridade nas urgências hospitalares.

A primeira relaciona o número total de urgências, o tempo de espera e tempo de permanência médio em horas com a prioridade. É possível verificar que a permanência é relativamente superior nos casos mais graves (emergente e muito urgente), como seria de esperar, decrescendo este tempo à medida que se desce na prioridade atribuída.

Com os dados até agora carregados, apenas referentes a um mês, não é possível fazer uma análise rigorosa do tempo de espera. Esta situação deve-se ao facto de a maioria dos episódios urgentes não possuir a hora de entrada no serviço de observação impossibilitando o cálculo do tempo de espera para grande parte dos casos.

É expectável que se consiga melhorar a fiabilidade desta estatística com a recolha da hora de entrada no serviço de observação e/ou que o carregamento de mais meses permita uma análise mais abrangente e fiável. A segunda matriz apresenta a percentagem por prioridade. É possível verificar que cerca de 88% das urgências hospitalares se centraram nos tipos urgente e pouco urgente.

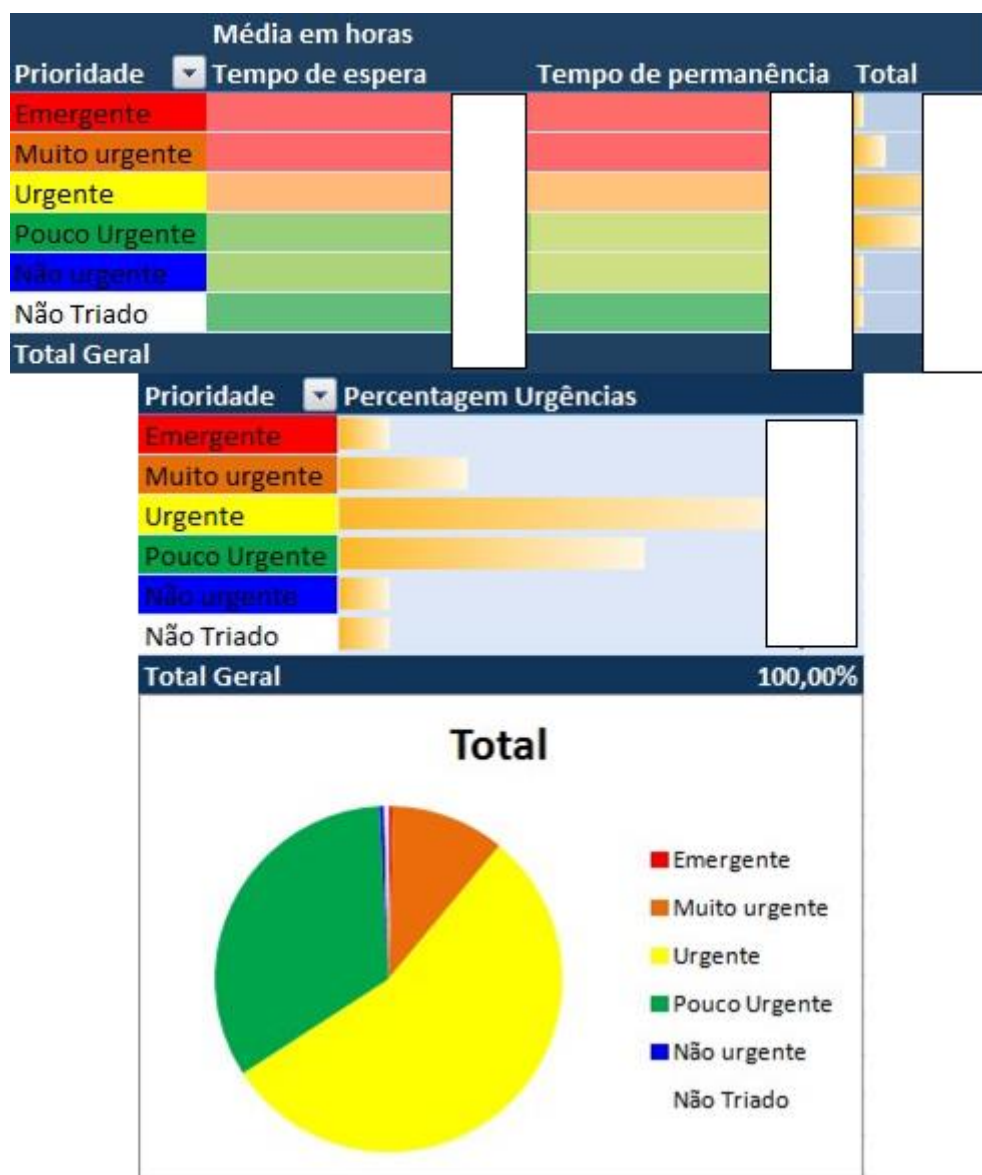


Figura 7.3: Dashboards Prioridade.

8 Conclusão e Trabalho Futuro

Conclusão

Neste relatório apresentámos o trabalho elaborado no âmbito de um estágio de mestrado realizado na Administração Regional de Saúde do Norte (ARS-N). O projeto consistia em desenvolver uma solução de Business Intelligence (BI) para os dados das urgências, que a ARS-N como Entidade Pública Contratante (EPC) da Parceria Público Privada (PPP) do Hospital de Braga, recebe mensalmente e necessita analisar. Esta plataforma tinha como objetivo conferir uma ferramenta que possibilitasse os mecanismos necessários para suportar o acesso aos dados de forma útil, eficaz e agilizada, a fim de melhorar o processo de tomada de decisão.

Foram consideradas as condições contratuais estabelecidas entre a EPC e a Entidade Gestora do Estabelecimento (EGEST) para definir as validações necessárias dos episódios de urgências. Foi também, tido em conta o levantamento de indicadores desenvolvidos no projeto de BI realizado no Departamento de Estudos e Planeamento da ARS-N, que foi disponibilizado para este projeto.

Com base nesta informação, foi realizado um Data Mart (DM) para o serviço de urgências. Ainda assim, o projeto como um todo encontra-se numa fase preliminar do seu desenvolvimento. Dado que será expectável, em termos futuros, a inclusão de mais Data Marts no que diz respeito à informação pertencente a outros serviços do Hospital de Braga, o Data Mart desenvolvido foi projetado com base na arquitetura defendida por Kimball.

Foi também desenvolvido inicialmente um processo de extração que armazene os ficheiros de Excel recebidos pela EPC numa *Staging Area*, e daí ser feito todo o processo de ETL, de forma a implementar as tabelas de dimensão e de factos necessárias para o desenvolvimento do Cubo – Data Mart Urgências.

Por fim, foi definido um conjunto de indicadores e *dashboards*, baseados em consultas ao DM desenvolvido.

Desta forma, foi descrita a solução de BI para plataforma desenvolvida e que esperamos possa ser útil e facilite a tomada de decisão.

Trabalho Futuro

O projeto encontra-se numa fase inicial do seu desenvolvimento e, portanto, em crescimento. É expectável que seja possível analisar e implementar algumas destas possíveis melhorias:

- tendo-se colocado as validações diretamente na tabela de facto (podendo ser vistas estas colunas como um índice *Bitmap*), tentou-se dar prevalência à performance em detrimento do espaço em memória gasto. Uma solução que poderá não ter tão boa performance, mas que reduziria o espaço de memória seria estruturar essas validações numa tabela de ponte. Seria interessante fazer essa comparação com mais dados para perceber até que ponto o nível de performance do Data Mart ficaria afetado.
- procurar formas de otimizar e automatizar o sistema de ETL; integrar mais serviços do Hospital de Braga tais como Consulta Externa, Internamento, Cirurgia, entre outros.
- carregar dados referentes aos dois últimos anos, para que seja possível avaliar a performance do sistema em mais detalhada e com mais precisão, bem como possibilitar uma análise dos dados do Data Mart mais abrangente. Assim, poderíamos dispor de comparações entre períodos homólogos, por exemplo.

Referências

- [1] Bill Inmon. Corporate Information Factory (CIF) - Overview. <http://www.inmoncif.com/library/cif/> [Acesso em: 2014/10/31].
- [2] Willian H. Inmon. Building the Data Warehouse. Wiley, 4th edition, 2005.
- [3] Kimball Group. Enterprise Data Warehouse Bus Architecture. <http://www.kimballgroup.com/data-warehouse-business-intelligenceresources/kimball-techniques/dimensional-modelingtechniques/enterprise-data-warehouse-bus-architecture/> [Acesso em: 2014/10/31].
- [4] Kimball Group. Conformed Dimension. <http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/conformed-dimension/> [Acesso em: 2014/10/31].
- [5] Vangie Beal. BI - business intelligence. Webopedia (online tech dictionary). http://www.webopedia.com/TERM/B/Business_Intelligence.html [Acesso em: 2014/10/31].
- [6] Ryan Mulcahy. Business Intelligence Definition and Solutions (Tutorial), CIO, 2007. <http://www.cio.com/article/2439504/business-intelligence/businessintelligence-definition-and-solutions.html> [Acesso em: 2014/10/31].
- [7] David Baum. Business Intelligence -- Interview With Gartner Group's Howard Dresner. Information Builders Magazine, vol. 11 (2): 26-28, 2001.

http://www.informationbuilders.com/new/magazine/v112/pdf/DRE_SNERI.PDF [Acesso em: 2014/10/31].

[8] Cindi Howson. *Techno Babble: Components of a Business Intelligence Architecture*, 2008.

<http://www.b-eye-network.com/view/7105> [Acesso em: 2014/10/31].

[9] Hugh J. Watson and Barbara H. Wixom. *The Current State of Business Intelligence*. IEEE Computer Society, 40:96–99, 2007.

[10] Kimball Group. *Factless Fact Tables*.

<http://www.kimballgroup.com/1996/09/factless-fact-tables/> [Acesso em: 2014/10/31].

[11] Kimball Group. *Multiple Hierarchies in Dimensions*.

<http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dimensional-modeling-techniques/multiple-hierarchies/>

[12] Benny Austin. *Dimensional Modeling: Banding, SSAS Articles*, 2013 (blog).

<http://www.ssas-info.com/analysis-services-articles/62design/4103-dimensional-modeling--banding> [Acesso em: 2014/10/31].

[13] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit*. Wiley, 2nd edition, 2002.

[14] Kimball Group. *Dimensional Modeling Techniques*.

<http://www.kimballgroup.com/data-warehouse-businessintelligence-resources/kimball-techniques/dimensionalmodeling-techniques/>

[Acesso em: 2014/10/31].

[15] Rogério Araújo. BI: Arquiteturas OLAP, 2009 (blog).
<http://rogerioaraujo.wordpress.com/2009/05/27/bi-arquiteturasolap/> [Acesso em: 2014/10/31].

[16] Ozimar Henrique. Data Warehouse x Data Mart, Microsoft, Technet, 2012 (Wiki).
<http://social.technet.microsoft.com/wiki/ptbr/contents/articles/10274.data-warehouse-x-data-mart.aspx> [Acesso em: 2014/10/31].

[17] Kimball Group. Role-Playing Dimension.
<http://www.kimballgroup.com/data-warehouse-business-intelligenceresources/kimball-techniques/dimensional-modeling-techniques/roleplaying-dimension/> [Acesso em: 2014/10/31].

[18] Kimball Group. Degenerate Dimension.
<http://www.kimballgroup.com/data-warehouse-business-intelligenceresources/kimball-techniques/dimensional-modelingtechniques/degenerate-dimension/> [Acesso em: 2014/10/31].

[19] Hugh J. Watson and Barbara H. Wixom. The Current State of Business Intelligence. IEEE Computer Society, 40:96–99, 2007.

[20] Office-Microsoft. What is the difference between a dashboard and a scorecard?
<http://office.microsoft.com/en-001/sharepoint-server-help/what-is-the-difference-between-a-dashboard-and-a-scorecardHA101772797.aspx> [Acesso em: 2014/10/31].

[21] Jorge Caldeira. 100 Indicadores da Gestão - Key Performance Indicators, Almedina, 2010.

http://monitorizaraperformance.blogspot.pt/2010_02_01_archive.html [Acesso em: 2014/10/31].

[22] Office-Microsoft. Visão geral do OLAP (processamento analítico online), Excel 2007. <http://office.microsoft.com/pt-br/excel-help/visao-geral-do-olapprocessamento-analitico-online-HP010177437.aspx> [Acesso em: 2014/10/31].

[23] Pedro Graça. Um sistema de apoio à decisão para um serviço de documentação e informação. Tese de Mestrado, FEUP, Universidade do Porto, 2013. <http://repositorio-aberto.up.pt/handle/10216/67939> [Acesso em: 2014/10/31].

[24] Data Warehouse & Business Intelligence Architecture. <http://www.olapbi.com> [Acesso em: 2014/10/31].

Anexo 1 – Scripts

Anexo 1.1 - Carregamento da Dimensão Data.

```
USE [dm]
GO
CREATE TABLE [DIM_DATA1](
    [SK_DIM_DATA] AS CONVERT(INTEGER, (CONVERT([char](4), datepart(year, [UK_INSTANTE]), (0))
    + SUBSTRING(('0' + CONVERT([char](2), datepart(month, [UK_INSTANTE]), (0))), LEN(CONVERT([char](2), datepart(month, [UK_INSTANTE]), (0))), 2)
    + SUBSTRING(('0' + CONVERT([char](2), datepart(day, [UK_INSTANTE]), (0))), LEN(CONVERT([char](2), datepart(day, [UK_INSTANTE]), (0))), 2))), 2)
    [DIA_DA_SEMANA] AS DATEPART(DW, UK_INSTANTE),
    [NOME_DIA_DA_SEMANA] AS DATENAME(DW, UK_INSTANTE),
    [DIA_DO_MES] AS DATENAME(DD, UK_INSTANTE),
    [DIA_DO_ANO] AS DATENAME(DY, UK_INSTANTE),
    [FDS_OU_DIA_UTIL] AS CASE WHEN DATEPART(DW, UK_INSTANTE) IN (1, 7) THEN 'FIM DE SEMANA'
    ELSE 'DIA UTIL' END ,
    [SEMANA_DO_ANO] AS DATEPART(WW, UK_INSTANTE),
    [COD_MES] AS DATENAME(Y, UK_INSTANTE) + '-' + RIGHT('0' + CAST(Month(UK_INSTANTE) as varchar), 2),
    [MES] AS DATENAME(MM, UK_INSTANTE) ,
    [MES_DO_ANO] AS DATEPART(MM, UK_INSTANTE),
    [COD_TRIMESTRE] AS DATENAME(Y, UK_INSTANTE) + 'T' + DATENAME(QQ, UK_INSTANTE),
    [TRIMESTRE] AS DATEPART(QQ, UK_INSTANTE),
    [COD_SEMESTRE] AS CASE WHEN DATEPART(QQ, UK_INSTANTE) = 1 OR DATEPART(QQ, UK_INSTANTE) = 2
    THEN DATENAME(Y, UK_INSTANTE) + 'S1'
    ELSE DATENAME(Y, UK_INSTANTE) + 'S2' END,
    [SEMESTRE] AS CASE WHEN DATEPART(QQ, UK_INSTANTE) = 1 OR DATEPART(QQ, UK_INSTANTE) = 2 THEN '1' ELSE '2' END,
    [ANO] AS DATEPART(Y, UK_INSTANTE),
    [UK_INSTANTE] [DATE],
    [NOME_DATA] AS CONVERT(varchar(20), [UK_INSTANTE], 106)
)
GO

DECLARE @DATA_INICIO DATETIME
DECLARE @DATA_FIM DATE
DECLARE @i INT
DECLARE @nrows INT
-- Atualizar aqui a data inicial
SET @DATA_INICIO = '2018-01-01'
SET @DATA_FIM = '2018-12-31'
SET @i = 0
SET LANGUAGE brazilian -- Usado brasileiro para colocar mes com primeira letra a maiusculo
SET @nrows = DATEDIFF(d, @DATA_INICIO, @DATA_FIM)+1
WHILE @i < @nrows
BEGIN
    INSERT DIM_DATA1 (UK_INSTANTE) values (@DATA_INICIO)
    SET @DATA_INICIO = DATEADD(d, 1, @DATA_INICIO)
    SET @i = @i + 1
END

SET LANGUAGE brazilian -- Usado brasileiro para colocar mes com primeira letra a maiusculo
SELECT *
INTO DIM_DATA
FROM DIM_DATA1
GO
```

Anexo 1.2 - Carregamento dos Feriados Dimensão Data.

```
--CRIACAO DA COLUNA FERIADO
ALTER TABLE [dbo].[DIM_DATA]
ADD FERIADO INT NULL
GO

--CORPO DE DEUS [2010-2012]
UPDATE [dbo].[DIM_DATA]
SET [FERIADO] = 1
WHERE [MES_DO_ANO] = 6 AND [DIA_DO_MES] = 3 AND ([ANO] =2010 OR [ANO] =2011 OR [ANO]= 2012)

--5 DE OUTUBRO
UPDATE [dbo].[DIM_DATA]
SET [FERIADO] = 1
WHERE [MES_DO_ANO] = 10 AND [DIA_DO_MES] = 5 AND ([ANO] =2010 OR [ANO] =2011 OR [ANO]= 2012)

--1 DE NOVEMBRO
UPDATE [dbo].[DIM_DATA]
SET [FERIADO] = 1
WHERE [MES_DO_ANO] = 11 AND [DIA_DO_MES] = 1 AND ([ANO] =2010 OR [ANO] =2011 OR [ANO]= 2012)

--1 DE DEZEMBRO
UPDATE [dbo].[DIM_DATA]
SET [FERIADO] = 1
WHERE [MES_DO_ANO] = 12 AND [DIA_DO_MES] = 1 AND ([ANO] =2010 OR [ANO] =2011 OR [ANO]= 2012)

--SEXTA FEIRA SANTA
UPDATE [dbo].[DIM_DATA]
SET [FERIADO] = 1
WHERE (([DIA_DO_MES] = 2 AND [MES_DO_ANO] = 4 AND [ANO] = 2010) OR
([DIA_DO_MES] = 22 AND [MES_DO_ANO] = 4 AND [ANO] = 2011) OR
([DIA_DO_MES] = 6 AND [MES_DO_ANO] = 4 AND [ANO] = 2012) OR
([DIA_DO_MES] = 29 AND [MES_DO_ANO] = 4 AND [ANO] = 2013) OR
([DIA_DO_MES] = 18 AND [MES_DO_ANO] = 4 AND [ANO] = 2014) OR
([DIA_DO_MES] = 3 AND [MES_DO_ANO] = 4 AND [ANO] = 2015) OR
([DIA_DO_MES] = 25 AND [MES_DO_ANO] = 3 AND [ANO] = 2016) OR
([DIA_DO_MES] = 14 AND [MES_DO_ANO] = 4 AND [ANO] = 2017) OR
([DIA_DO_MES] = 30 AND [MES_DO_ANO] = 3 AND [ANO] = 2018) )

--CARNAVAL
--SEXTA FEIRA SANTA
UPDATE [dbo].[DIM_DATA]
SET [FERIADO] = 1
WHERE (([DIA_DO_MES] = 16 AND [MES_DO_ANO] = 2 AND [ANO] = 2010) OR
([DIA_DO_MES] = 8 AND [MES_DO_ANO] = 3 AND [ANO] = 2011) OR
([DIA_DO_MES] = 21 AND [MES_DO_ANO] = 2 AND [ANO] = 2012) OR
([DIA_DO_MES] = 12 AND [MES_DO_ANO] = 2 AND [ANO] = 2013) OR
([DIA_DO_MES] = 4 AND [MES_DO_ANO] = 3 AND [ANO] = 2014) OR
([DIA_DO_MES] = 17 AND [MES_DO_ANO] = 2 AND [ANO] = 2015) OR
([DIA_DO_MES] = 9 AND [MES_DO_ANO] = 2 AND [ANO] = 2016) OR
([DIA_DO_MES] = 28 AND [MES_DO_ANO] = 2 AND [ANO] = 2017) OR
([DIA_DO_MES] = 13 AND [MES_DO_ANO] = 2 AND [ANO] = 2018) )
```

```

|--PASCOA
--SEXTA FEIRA SANTA
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE (([DIA_DO_MES] = 4 AND [MES_DO_ANO] = 4 AND [ANO] = 2010) OR
           ([DIA_DO_MES] = 24 AND [MES_DO_ANO] = 4 AND [ANO] = 2011) OR
           ([DIA_DO_MES] = 8 AND [MES_DO_ANO] = 4 AND [ANO] = 2012) OR
           ([DIA_DO_MES] = 31 AND [MES_DO_ANO] = 3 AND [ANO] = 2013) OR
           ([DIA_DO_MES] = 20 AND [MES_DO_ANO] = 4 AND [ANO] = 2014) OR
           ([DIA_DO_MES] = 5 AND [MES_DO_ANO] = 4 AND [ANO] = 2015) OR
           ([DIA_DO_MES] = 27 AND [MES_DO_ANO] = 3 AND [ANO] = 2016) OR
           ([DIA_DO_MES] = 16 AND [MES_DO_ANO] = 4 AND [ANO] = 2017) OR
           ([DIA_DO_MES] = 1 AND [MES_DO_ANO] = 4 AND [ANO] = 2018) )

--
--25 DE ABRIL
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE ([DIA_DO_MES] = 25 AND [MES_DO_ANO] = 4 AND ([ANO] BETWEEN 2010 AND 2018))
--1 DE MAIO
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE ([DIA_DO_MES] = 1 AND [MES_DO_ANO] = 5 AND ([ANO] BETWEEN 2010 AND 2018))
--10 DE JUNHO
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE ([DIA_DO_MES] = 10 AND [MES_DO_ANO] = 6 AND ([ANO] BETWEEN 2010 AND 2018))
--15 DE AGOSTO
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE ([DIA_DO_MES] = 15 AND [MES_DO_ANO] = 8 AND ([ANO] BETWEEN 2010 AND 2018))
--8 DE DEZEMBRO
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE ([DIA_DO_MES] = 8 AND [MES_DO_ANO] = 12 AND ([ANO] BETWEEN 2010 AND 2018))
--25 DE DEZEMBRO
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE ([DIA_DO_MES] = 25 AND [MES_DO_ANO] = 12 AND ([ANO] BETWEEN 2010 AND 2018))
--SÃO JOÃO - 24 DE JUNHO
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 1
    WHERE ([DIA_DO_MES] = 24 AND [MES_DO_ANO] = 6 AND ([ANO] BETWEEN 2010 AND 2018))
--COLOCAR OS NULL'S A ZERO
UPDATE [dbo].[DIM_DATA]
    SET [FERIADO] = 0
    WHERE [FERIADO] IS NULL

```

Anexo 1.3 - Carregamento da Dimensão Minuto.

```
USE [STG_URGENCIAS]
GO
CREATE TABLE [DIM_MINUTO](
[SK_MINUTO] INT PRIMARY KEY,
[UK_INSTANTE] VARCHAR(5),
[LK_MINUTO] INT,
[MINUTO] INT,
[LK_HORA] INT,
[HORA] INT,
[PERIODO] VARCHAR(10),
[PERIODO_HORARIO] VARCHAR(10)
)
GO

declare @LK_MINUTO INT, @LK_HORA INT, @PERIODO VARCHAR(10), @PERIODO_HORARIO VARCHAR(10)
SET @LK_HORA = 0
WHILE @LK_HORA < (24)
BEGIN
SET @LK_MINUTO = 0

set @PERIODO = CASE
                WHEN @LK_HORA BETWEEN 8 AND 19 THEN 'DIURNO'
                ELSE 'NOCTURNO'
                END

set @PERIODO_HORARIO = CASE
                        WHEN @LK_HORA BETWEEN 2 AND 7 THEN '2H-8H'
                        WHEN @LK_HORA BETWEEN 8 AND 13 THEN '8H-14H'
                        WHEN @LK_HORA BETWEEN 14 AND 16 THEN '14H-17H'
                        WHEN @LK_HORA BETWEEN 17 AND 19 THEN '17H-20H'
                        WHEN @LK_HORA BETWEEN 20 AND 23 THEN '20H-2H'
                        WHEN @LK_HORA BETWEEN 0 AND 2 THEN '20H-2H'
                        END

WHILE @LK_MINUTO < (60)
BEGIN
insert into [DIM_MINUTO](SK_MINUTO,UK_INSTANTE,LK_MINUTO,MINUTO,LK_HORA,HORA,PERIODO,PERIODO_HORARIO)
SELECT (100*@LK_HORA)+@LK_MINUTO AS SK_MINUTO,
       right('0'+convert(varchar(2),@LK_HORA),2)+''+right('0'+convert(varchar(2),@LK_MINUTO),2) [UK_INSTANTE],
       @LK_MINUTO AS [LK_MINUTO],
       @LK_MINUTO AS [MINUTO],
       @LK_HORA AS [LK_HORA],
       @LK_HORA AS [HORA],
       @PERIODO AS [PERIODO],
       @PERIODO_HORARIO AS [PERIODO_HORARIO]
set @LK_MINUTO= @LK_MINUTO +1
END
set @LK_HORA = @LK_HORA + 1]
END
GO
```

Anexo 1.4 - Carregamento da Dimensão Idade

```
USE [STG_URGENCIAS]
GO

CREATE TABLE [DIM_STG_IDADE](
    [SK_IDADE] INT NOT NULL,
    [IDADE] VARCHAR(9) NOT NULL,
    [FAIXA_ETARIA] VARCHAR(12),
)
GO

DECLARE @INICIO_IDADE INT, @FIM_IDADE INT
SET @INICIO_IDADE =0
SET @FIM_IDADE =131
WHILE (@@INICIO_IDADE <= @@FIM_IDADE)
BEGIN
INSERT INTO [DIM_STG_IDADE]
SELECT
    @@INICIO_IDADE [SK_IDADE],
    CASE WHEN @@INICIO_IDADE = 0 THEN 'MESES'
        WHEN @@INICIO_IDADE <= 1 THEN CONVERT(CHAR(3),@@INICIO_IDADE)+' ANO'
        WHEN @@INICIO_IDADE > 130 THEN 'SEM IDADE'
            ELSE CONVERT(CHAR(3),@@INICIO_IDADE)+' ANOS' END [IDADE],
    CASE WHEN @@INICIO_IDADE = 0 THEN '0 a 11 MESES'
        WHEN @@INICIO_IDADE BETWEEN 1 AND 2 THEN '1-2'
        WHEN @@INICIO_IDADE BETWEEN 3 AND 5 THEN '3-5'
        WHEN @@INICIO_IDADE BETWEEN 6 AND 12 THEN '6-12'
        WHEN @@INICIO_IDADE BETWEEN 13 AND 18 THEN '13-18'
        WHEN @@INICIO_IDADE BETWEEN 19 AND 25 THEN '19-25'
        WHEN @@INICIO_IDADE BETWEEN 26 AND 45 THEN '26-45'
        WHEN @@INICIO_IDADE BETWEEN 46 AND 65 THEN '46-65'
        WHEN @@INICIO_IDADE BETWEEN 66 AND 85 THEN '66-85'
        WHEN @@INICIO_IDADE BETWEEN 86 AND 100 THEN '86-100'
        WHEN @@INICIO_IDADE BETWEEN 101 AND 130 THEN '101-130' ELSE 'SEM IDADE'END [FAIXA_ETARIA]
    SET @@INICIO_IDADE = @@INICIO_IDADE +1
END
GO

--COLUNA DE ERRO
UPDATE [dbo].[DIM_STG_IDADE]
SET [SK_IDADE] = 999
WHERE [SK_IDADE] = 131
```

Anexo 2 - Dicionário das Dimensões e ETL

Anexo 2.1 - Dimensão Utente.

ETL Dimensão Utente

Nome	DIM_UTENTE
Necessidade do negócio	Identificar

Base de dados fonte	STG_URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	14200	Número de linhas a carregar no momento da atualização (estimativa)	
Descrição	Este procedimento permitirá obter uma tabela com os dados informativos de cada utente.		
Frequência da atualização	Mensal		
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA		
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS		
Erros	Fail Component.		
Estratégia de recarregamento	Verificação de novos utentes e atualização de informação de utentes já existência.		

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
Excel Urgências	STG_URGENCIAS	Nenhum

Destino

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_UTENTE	dm	SK_DIM_UTENTE	SCD 1 e SCD 2

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_UTENTE	dm	SK_DIM_UTENTE - Autonumber	

Lookups:

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_UTENTE	Localização/ Base de dados	dm
Match condition	<pre>select * from (select * from [dbo].[DIM_UTENTE]) [refTable] where ((([refTable].[Número de Cartão de Utente] = ? and [refTable].[Número de Cartão de Utente] <> 0) OR ([refTable].[Número Beneficiário] = ? and [refTable].[Número Beneficiário] <> '0') OR ([refTable].[Número de Cartão de Utente] = 0 AND [refTable].[Número Beneficiário] = '0')) AND ([refTable].[Isenção Taxa moderadora (S/N)] = ?))</pre>		

OLE DB Command:

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_UTENTE	Localização/ Base de dados	dm
Update	<pre>UPDATE DBO.DIM_UTENTE SET ESTADO=0, DATA_FIM = ? WHERE (([Número de Cartão de Utente] = ? AND [Número de Cartão de Utente] <> 0) OR ([Número Beneficiário] = ? AND [Número Beneficiário] <> '0') OR ([Número de Cartão de Utente] = 0 AND [Número Beneficiário] = '0'));</pre>		

Lookup 1:

Nome transformação da	Deteção de Novos Casos		
Tabela	DIM_UTENTE	Localização/ de dados	Base dm
Match condition	<pre>select * from (select * from [dbo].[DIM_UTENTE]) [refTable] where [refTable].[Número de Cartão de Utente] = ? AND (([refTable].[Número Beneficiário]=? OR ?='0') AND ([refTable].[Data Nascimento] = ? OR ? is null) AND ([refTable].[Género] = ? OR ? = 'Desconhecido') AND ([refTable].[Telefone Contacto] = ? OR ?='0') AND ([refTable].[Número do Documento] = ? OR ? = '0') AND ([refTable].[Tipo Documento Identificativo] = ? OR ? = 'Desconhecido'))</pre>		

Script task:

Nome transformação da	Atualização – SCD 1		
Tabela	DIM_UTENTE	Localização/ de dados	Base dm
Update	<pre>UPDATE [dbo].[DIM_UTENTE] SET [DIM_UTENTE].[Género] = [DIM_UTENTE_STG2].[Género], [DIM_UTENTE].[Data Nascimento] = [DIM_UTENTE_STG2].[Data Nascimento] [DIM_UTENTE].[Telefone Contacto] = [DIM_UTENTE_STG2].[Telefone Contacto] [DIM_UTENTE].[Número Beneficiário] = [DIM_UTENTE_STG2].[Número Beneficiário] [DIM_UTENTE].[Número de Cartão de Utente] = [DIM_UTENTE_STG2].[Número de Cartão de Utente] [DIM_UTENTE].[Tipo Documento Identificativo] = [DIM_UTENTE_STG2].[Tipo Documento Identificativo] [DIM_UTENTE].[Número do Documento]= [DIM_UTENTE_STG2].[Número do Documento] FROM [DIM_UTENTE_STG2] WHERE [DIM_UTENTE].[Número de Cartão de Utente] = [DIM_UTENTE_STG2].[Número de Cartão de Utente]</pre>		

Script task:

Nome transformação da	Atualização – SCD 2		
Tabela	DIM_UTENTE_SCD_1; Dim_utente_stg333	Localização/ de dados	Base dm
truncate	<pre>truncate table [DIM_UTENTE_SCD_1] truncate table [Dim_utente_stg333]</pre>		

DIM_UTENTE

Logical Data Map		Project Name 01 - BI_URGENCIAS				REFERENCE						
		ETL Package Name : Package.dtsx				BI01						
Developer / Designer		DATAFLOW	PURPOSE	SOURCE> TARGET	DATE	VERSION						
David Gonçalves		DIM_UTENTE	Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse	Staging Area (STG) - Data warehouse	11-09-2014	2.0						
TARGET (DW)		SOURCES (SO)										
TARGET TYPE		SQL Server 2012	SOURCE TYPE	SQL Server 2012	TRANSFORMATIONS							
DATABASE		dm	DATABASE	STG_URGENCIAS								
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE	DATABASE	TABLE NAME	COLUMN NAME	DATA TYPE	MAX	TRANSFORMATIONS
DIM_UTENTE	SK_DIM_UTENTE	int				SK	STG_URGENCIAS	URGENCIAS				Começa em 1 e incrementa, de 1 em 1
	Número de Cartão de Utente	numeric(18,0)		1	0	UK	STG_URGENCIAS	URGENCIAS	Número de Cartão de Utente	float		Conversão para numérico
	Número Beneficiário	varchar(50)		1	0	UK	STG_URGENCIAS	URGENCIAS	Número Beneficiário	nvarchar	255	Converter para nvarchar
	Número de Cartão de Saúde Europeu	nvarchar(255)		1	0	UK	STG_URGENCIAS	URGENCIAS	Número de Cartão de Saúde Europeu	nvarchar	255	Preencher os nulos com zero: REPLACENULL
	Telefone Contacto	nvarchar(20)		1	0		STG_URGENCIAS	URGENCIAS	Telefone Contacto	float		Converter para nvarchar
	Número do Documento	nvarchar(20)		1	0		STG_URGENCIAS	URGENCIAS	Número do Documento	float		
	Tipo Documento Identificativo	varchar(255)		1	Desconhecido		STG_URGENCIAS	URGENCIAS	Tipo Documento Identificativo	nvarchar	255	Preencher os nulos com zero: REPLACENULL
Género	varchar(255)		1	Desconhecido		STG_URGENCIAS	URGENCIAS	Género	varchar	255	Preencher os nulos com zero: REPLACENULL	

Anexo 2.2 - Dimensão Local.

ETL Dimensão Local	
Nome	DIM_LOCAL
Necessidade do negócio	Caracterizar os episódios de urgências por Local (Distrito, Concelho, Freguesia).

Base de dados fonte	STG_URGENCIAS	Base de dados de destino	DW_URGENCIAS
Número de linhas inicial (estimativa)	802	Número de linhas a carregar no momento da atualização (estimativa)	

Descrição	Este procedimento permitirá obter uma tabela com o código do local (Distrito, Concelho, Freguesia) e a sua descrição.
Frequência da atualização	Mensal
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS
Erros	Fail Component
Estratégia de recarregamento	Verificação de novos casos com remoção de duplicados, através de lookup. Sendo carregados os locais que não fizerem match (no match output) com nenhuma das causas já presentes na DIM_LOCAL

Fonte:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_LOCAL	dm	SK_LOCAL Autonumber	-

Lookups:

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_LOCAL	Localização/ Base de dados	dm
Match condition	[STG_URGENCIA].[dbo].[URGENCIAS].[Distrito] = [dm].[dbo].[DIM_SERVIÇO]. [Distrito]		

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_LOCAL	Localização/ Base de dados	dm
Match condition	[STG_URGENCIA].[dbo].[URGENCIAS].[Concelho] = [dm].[dbo].[DIM_SERVIÇO]. [Concelho]		

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_LOCAL	Localização/ Base de dados	dm
Match condition	[STG_URGENCIA].[dbo].[URGENCIAS].[Freguesia] = [dm].[dbo].[DIM_SERVIÇO]. [Freguesia]		

pim_LOCAL

Logical Data Map		Project Name - BI_URGENCIAS				REFERENCE						
Developer / Designer		ETL Package Name : Package.dtsx <td>BI01</td>				BI01						
David Gonçalves	DATAFLOW	PURPOSE				VERSION						
	DIM_LOCAL	Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse				2.0						
TARGET (DW)		SOURCES (SO)				DATE						
SQL Server 2012		SQL Server 2012				2014-09-20						
dim		STG_URGENCIAS										
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE	DATABASE	TABLE NAME	COLUMN NAME	DATA TYPE	MAX	TRANSFORMATIONS
	SK_LOCAL	Int				SK				Automunber		Começa em 1 e incrementa, de 1 em 1 [SK_LOCAL] INT IDENTITY (1,1)
	COD_DISTRITO	Int	Dimensão		-	LK				Automunber		Começa em 1 e incrementa, de 1 em 1 [COD_DISTRITO] INT IDENTITY (1,1)
	Distrito	nvarchar	Dimensão		-	-			Distrito	nvarchar	255	
	COD_CONSELHO	Int	Dimensão		-	LK		STG_URGENCIAS	URGENCIAS	Automunber		Começa em 1 e incrementa, de 1 em 1 [COD_CONSELHO] INT IDENTITY (1,1)
	Concelho	nvarchar	Dimensão		-	-			Concelho	nvarchar	255	
	COD_FREGUESIA	Int	Dimensão		-	LK				Automunber		Começa em 1 e incrementa, de 1 em 1 [COD_FREGUESIA] INT IDENTITY (1,1)
	Freguesia	nvarchar	Dimensão		-	-			Freguesia	nvarchar	255	

Anexo 2.3 - Dimensão Causa.

ETL Dimensão Causa	
Nome	DIM_CAUSA
Necessidade do negócio	Compreender quais são as principais causas dos episódios de urgência no hospital (doença, queda, acidente de viação, gravidez, ...)

Base de dados fonte	STG_URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	18	Número de linhas a carregar no momento da atualização (estimativa)	
Descrição	Este procedimento permitirá obter uma tabela com a descrição das causas das urgências do hospital		
Frequência da atualização	Mensal		
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA		
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS		
Erros	Fail Component		
Estratégia de recarregamento	Verificação de novos casos com remoção de duplicados, através da lookup. Sendo carregados as causas que não fizerem match (no match output) com nenhuma das causas, já presentes na DIM_CAUSA.		

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_CAUSA	dm	SK_CAUSA - Autonumber	

Lookups:

Nome da transformação	Detecção de Novos Casos		
Tabela	DIM_CAUSA	Localização/ Base de dados	dm
Match condition	[STG_URGENCIA].[dbo].[URGENCIAS].[Causa de Admissão] = [dm].[dbo].[DIM_CAUSA].[Causa de Admissão]		

Anexo 2.4 - Dimensão Serviço.

ETL Dimensão Serviço

Nome	DIM_SERVIÇO
Necessidade do negócio	Compreender quais os principais serviços requeridos no hospital (psiquiatria, pediatria, ortopedia, etc)

Base de dados fonte	STG_URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	55	Número de linhas a carregar na atualização (estimativa)	

Descrição	Este procedimento permitirá obter uma tabela com a descrição dos serviços de urgências do hospital
Frequência da atualização	Mensal
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS
Erros	Fail Component
Estratégia de recarregamento	Verificação de novos casos, com remoção de duplicados, através da lookup. Sendo carregados os serviços que não fizerem match (no match output) com nenhuma dos serviços já presentes na DIM_SERVIÇO.

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_SERVIÇO	dm	SK_SERVIÇO - Autonumber	Deteção de novos casos

Lookups:

Nome da transformação	Deteccção de Novos Casos		
Tabela	DIM_SERVIÇO	Localização/ Base de dados	STG_URGENCIAS
Match condition	[STG_URGENCIA].[dbo].[URGENCIAS].[Serviço GH Alta] = [dm].[dbo].[DIM_SERVIÇO].[Serviço GH Alta] [STG_URGENCIA].[dbo].[URGENCIAS].[Tipo de urgências] = [dm].[dbo].[DIM_SERVIÇO].[Tipo de urgências]		

DIM_SERVIÇO

Logical Data Map		Project Name - BI_URGENCIAS				REFERENCE
		ETL Package Name : Package.dtsx				VERSION
						2.0
Developer / Designer	DATAFLOW	PURPOSE	SOURCE > TARGET	DATE		
David Gonçalves	DIM_SERVIÇO	Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse	Staging Area (STG) - Data warehouse	2014-09-20		
TARGET (DW)		SOURCES (SO)				
TARGET TYPE	SQL Server 2012	SOURCE TYPE	SQL Server 2012	TRANSFORMATIONS		
DATABASE	dm	DATABASE	STG_URGENCIAS			
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE
DIM_SERVIÇO	SK_SERVIÇO	Int				SK
	Serviço GH Alta	Nvarchar(255)	Dimensão			
	Tipo de urgência	Nvarchar(255)	Dimensão			
			STG_URGENCIAS	URGENCIAS	Serviço GH Alta	255
			Automunber		Tipo de urgência	255
						Começa em 1 e incrementa, de 1 em 1
						[SK_SERVIÇO] INT IDENTITY (1,1)

Anexo 2.5 – Dimensão Prioridade.

ETL Dimensão Prioridade

Nome	DIM_PRIORIDADE
Necessidade do negócio	Compreender quais são as principais causas dos episódios de urgência no hospital (doença, queda, acidente de viação, gravidez, ...)

Base de dados fonte	STG_ URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	7	Número de linhas a carregar na atualização (estimativa)	0

Descrição	Este procedimento permitirá obter uma tabela com a descrição das prioridades nas urgências hospitalares (emergente, muito urgente, etc). Bem como as respetivas cores da Triagem Manchester.
Frequência da atualização	Mensal
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS
Erros	Fail Component
Estratégia de recarregamento	Verificação de novos casos com remoção de duplicados, através lookup. Sendo carregados as prioridades que não fizerem match (no match output) com nenhuma das prioridades presentes na DIM_PRIORIDADE.

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_ URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_PRIORIDADE	dm	SK_PRIORIDADE - Autonumber	

Lookups:

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_PRIORIDADE	Localização/ Base de dados	dm
Match condition	[STG_URGENCIA].[dbo].[URGENCIAS].[Prioridade] = [dm].[dbo].[DIM_SERVIÇO]. [Prioridade][STG_URGENCIA] .[dbo].[URGENCIAS].[Cor] = [dm].[dbo].[DIM_SERVIÇO].[Cor]		

DIM_PRIORIDADE

Logical Data Map		Project Name - BI_URGENCIAS		REFERENCE								
		ETL Package Name : Package.dtsx		B101								
Developer / Designer		DATAFLOW		VERSION								
David Gonçalves		DIM_PRIORIDADE		2.0								
		PURPOSE		SOURCE >								
		Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse		TARGET								
		SOURCES (SO)		Staging Area (STG) - Data warehouse								
TARGET (DW)				DATE								
				2014-09-20								
TARGET TYPE		SOURCE TYPE		TRANSFORMATIONS								
SQL Server 2012		SQL Server 2012										
DATABASE		DATABASE										
dm		STG_URGENCIAS										
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE	DATABASE	TABLE NAME	COLUMN NAME	DATA TYPE	MAX	
DIM_PRIORIDADE	SK_PRIORIDADE	Int				SK	Automumber					Começa em 1 e incrementa de 1 em 1
	Prioridade	Nvarchar (255)	Dimensão		-		STG_URGENCIAS	URGENCIAS	Prioridade	nvarchar	255	[SK_PRIORIDADE] INT IDENTITY (1,1)
	cor	nvarchar	Dimensão		-				Cor	nvarchar	255	

Anexo 2.6 – Dimensão Origem.

ETL Dimensão Origem

Nome	DIM_ORIGEM
Necessidade do negócio	Perceber qual a origem dos utentes que se deslocam aos serviços de urgências (centro de saúde, exterior, outro hospital...)

Base de dados fonte	STG_ URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	132	Número de linhas a carregar na atualização (estimativa)	

Descrição	Este procedimento permitirá obter uma tabela única com a proveniência dos utentes que vão às urgências do hospital
Frequência da atualização	Mensal
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS
Erros	Fail Component
Estratégia de recarregamento	Verificação de novos casos com remoção de duplicados, através lookup. Sendo carregados as origens que não fizerem match (no match output) com nenhuma das origens já carregados na DIM_ORIGEM

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_ URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_ORIGEM	dm	SK_ORIGEM Autonumber	-

Lookups:

Nome da transformação	Detecção de Novos Casos		
Tabela	DIM_ORIGEM	Localização/ Base de dados	dm
Match condition	[STG_URGENCIA].[dbo].[URGENCIAS].[Origem] = [dm].[dbo].[DIM_SERVIÇO].[Origem] [STG_URGENCIA].[dbo].[URGENCIAS].[Unidade Origem] = [dm].[dbo].[DIM_SERVIÇO].[Unidade Origem]		

DIM_ORIGEM

Logical Data Map		Project Name - BI_URGENCIAS				REFERENCE
		ETL Package Name : Package.dtsx				B01
Developer / Designer		DATAFLOW		PURPOSE		VERSION
David Gonçalves		DIM_ORIGEM		Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse		2.0
TARGET (DW)				SOURCES (SO)		DATE
						2014-09-20
TARGET TYPE		SQL Server 2012		SOURCE TYPE		SOURCE > TARGET
				SQL Server 2012		Staging Area (STG) - Data warehouse
DATABASE		dm		DATABASE		TRANSFORMATIONS
				STG_URGENCIAS		TABLE NAME
TABLE NAME		DATA TYPE		TABLE NAME		MAX
SK_ORIGEM		Int		DATABASE		DATA TYPE
M				Automumber		Começa em 1 e incrementa, de 1 em 1
Origem		nvarchar(255)		URGENCIAS		[SK_ORIGEM]
Unidade de Origem		Dimensão		Origem		INT IDENTITY (1,1)
		Dimensão		Unidade de Origem		
				de Origem		

Anexo 2.7 - Dimensão Destino.

ETL Dimensão Destino

Nome	DIM_DESTINO
Necessidade do negócio	Perceber qual é o destino dos utentes após a alta do serviço de urgências (abandono, outro hospital, serviço de internamento...).

Base de dados fonte	STG_URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	167	Número de linhas a carregar no momento da atualização (estimativa)	

Descrição	Este procedimento permitirá obter uma tabela única com a descrição do destino do utente após a alta das urgências hospitalar
Frequência da atualização	Mensal
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS
Erros	Fail Component
Estratégia de recarregamento	Verificação de novos casos com remoção de duplicados, através de lookup. Sendo carregados os destinos que não fizerem match (no match output) com nenhum dos destinos já existentes na DIM_DESTINO.

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_DESTINO	dm	SK_DESTINO - Autonumber	Slowly changing dimension

Lookups:

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_DESTINO	Localização/ Base de dados	dm
Match condition	[STG_URGÊNCIA].[dbo].[URGENCIAS].[Destino] = [dm].[dbo].[DIM_SERVIÇO].[Destino] [STG_URGÊNCIA].[dbo].[URGENCIAS].[Unidade Destino] = [dm].[dbo].[DIM_SERVIÇO].[Unidade Destino]		

DIM_DESTINO										
Logical Data Map		Project Name - BI_URGENCIAS								REFERENCE
		ETL Package Name : Package.dtsx								B01
Developer / Designer		DATAFLOW		PURPOSE		SOURCE >		DATE		
David Gonçalves		DIM_DESTINO		Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data wharehouse		Staging Area (STG) - Data warehouse		2014-09-20		
TARGET (DW)										
TARGET TYPE		SQL Server 2012		SOURCE TYPE		SQL Server 2012		TRANSFORMATIONS		
DATABASE		dm		DATABASE		STG_URGENCIAS		TABLE NAME		
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE	DATABASE	COLUMN NAME	DATA TYPE	MAX
DIM_DESTINO	SK_DESTINO	Int				SK	Autonumber			
	Destino	Nvarchar (255)	Dimensão		-			Destino	nvarchar	255
	Unidade de Destino	Nvarchar (255)	Dimensão		-		STG_URGENCIAS	Unidade de Destino	nvarchar	255
									Comença em 1 e incrementa, de 1 em 1	[SK_DESTINO] INT IDENTITY (1,1)

Anexo 2.8 - Dimensão Entidade Financeira Responsável.

ETL Dimensão EFR (Entidade Financeira Responsável)	
Nome	DIM_EFR
Necessidade do negócio	Compreender quais as Entidades Financeiras Responsáveis (EFR) dos episódios de serviço de urgência no hospital (SNS, SAD GNR, ADSE, seguros, ...)

Base de dados fonte	STG_URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	41	Número de linhas a carregar na atualização (estimativa)	

Descrição	Este procedimento permitirá obter uma tabela única com as diferentes entidades financeira do serviço de urgência.
Frequência da atualização	Mensal
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS
Erros	Fail Component
Estratégia de recarregamento	Verificação de novos casos com remoção de duplicados, através de lookup. Sendo carregados as entidades que não fizerem match (no match output) com nenhuma das entidades já presentes na DIM_EFR

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
DIM_EFR	dm	SK_EFR Autonumber	-

Lookups:

Nome da transformação	Deteção de Novos Casos		
Tabela	DIM_EFR	Localização/ Base de dados	dm
Match condition	[STG_URGÊNCIA].[dbo].[URGENCIAS].[EFR] = [dm].[dbo].[DIM_SERVIÇO].[EFR] [STG_URGÊNCIA].[dbo].[URGENCIAS].[Grupo EFR] = [dm].[dbo].[DIM_SERVIÇO].[Grupo EFR]		

DIM_EFR

Logical Data Map		Project Name - BI_URGENCIAS				REFERENCE
		ETL Package Name : Package.dtsx				B101
Developer / Designer		DATAFLOW		SOURCE > TARGET		VERSION
David Gonçalves		DIM_EFR		Staging Area (STG) - Data warehouse		2.0
PURPOSE		Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse				DATE
2014-09-20		SOURCES (SO)				
TARGET (DW)						
TARGET TYPE	SQL Server 2012	SOURCE TYPE		SQL Server 2012		
DATABASE	dm	DATABASE		STG_URGENCIAS		
TRANSFORMATIONS						
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE
	SK_EFR	Int				SK
DIM_EFR	EFR	nvarchar(255)	Dimensão		-	
	Grupo EFR	nvarchar(255)	Dimensão		-	
			STG_URGENCIAS	Urgências	EFR	Autonumber
			Grupo EFR	EFR	nvarchar	255
			EFR	Grupo EFR	nvarchar	255
						[SK_EFR] INT IDENTITY (1,1)
						Começa em 1 e incrementa, de 1 em 1

Anexo 2.9 – Dimensão Elegibilidade.

ETL Dimensão Elegibilidade	
Nome	DIM_ELEGIBILIDADE
Necessidade do negócio	Compreender quais são os episódios de urgências são elegíveis para pagamento, e os seus motivos (abandono sem observação, alta para internamento, ...)

Base de dados fonte	STG_URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	7	Número de linhas a carregar no momento da atualização (estimativa)	

Descrição	Este procedimento permitirá obter uma tabela com os episódios de urgências que são ou não elegíveis para pagamento, assim como o seu motivo.
Frequência da atualização	Mensal
Pré-processamento	Data Flow - CARREGAMENTO_STAGE_AREA
Pós-processamento	Data Flow - CARREGAMENTO_TABELA_FACTOS
Erros	Fail Component
Estratégia de recarregamento	Verificação de novos casos com remoção de duplicados, através lookup. Sendo carregados os motivos que não fizerem match (no match output) com nenhuma dos motivos já presentes na DIM_ELEGIBILIDADE.

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URGENCIAS	STG_URGENCIAS	Nenhum

Destino:

Nome da Tabela	Localização / Base de dados	Chave Primária	Modo de atualização
DIM_ELEGIBILIDADE	dm	SK_ELEGIBILIDADE - Autonumber	

DIM_ELEGIBILIDADE

Logical Data Map		Project Name - BI_URGENCIAS				REFERENCE
		ETL Package Name : Package.dtsx				B01
Developer / Designer		PURPOSE		SOURCE > TARGET		VERSION
David Gonçalves		Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse		Staging Area (STG) - Data warehouse		2.0
DIM_ELEGIBILIDADE		SOURCES (SO)		DATE		
DIM_ELEGIBILIDADE				2014-09-20		
TARGET (DW)						
TARGET TYPE	SQL Server 2012	SOURCE TYPE	SQL Server 2012	TRANSFORMATIONS		
DATABASE	dm	DATABASE	STG_URGENCIAS			
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE
	SK_ELEGIBILIDADE	Int				SK
DIM_ELEGIBILIDADE	Internamento	nvarchar	Dimensão		-	
	Elegível	nvarchar	Dimensão		-	
	Motivo	nvarchar	Dimensão		-	
			Excel Urgências	STG_URGENCIAS		
			Internamento	nvarchar	255	
			Elegível	nvarchar	255	
			Motivo	nvarchar	255	
						Começa em [SK_ELEGIBILIDADE] e incrementa de 1 em 1 (1,1)

Anexo 3 – Dicionário da Tabela de Factos e ETL

ETL TABELA DE FACTOS – FACTO_URGENCIAS			
Nome		FT_URGENCIAS	
Base de dados fonte	STG_URGENCIAS	Base de dados de destino	dm
Número de linhas inicial (estimativa)	00	Número de linhas a carregar no momento da atualização (estimativa)	16.000
Descrição	Através da tabela Fact_Urgências serão construídas grande parte das métricas.		
Frequência da atualização	Tabela atualizada mensalmente		
Pré-processamento	CARREGAMENTO do Data Flow: DIM_UTENTE; DIM_SERVIÇO; DIM_DATA; DIM_IDADE; DIM_HORAS; DIM_PRIORIDADE; DIM_ORIGEM; DIM_DESTINO; DIM_EFR; DIM_ELEGIBILIDAE; DIM_CAUSA; DIM_PRIORIDADE.		
Pós-processamento			
Erros	Fail Component		
Estratégia de recarregamento			

Fontes:

Nome da Tabela	Localização/ Base de dados	Filtro de extração
URG_EPIDODIOS	STG_URGENCIAS	Nenhum

Destino

Nome da Tabela	Localização/ Base de dados	Chave Primária	Modo de atualização
FT_URGENCIAS	dm	SK_UTENTE; SK_ELEGIBILIDADE; SK_EFR; SK_PRIORIDADE; SK_SERVIÇO; SK_CAUSA; SK_ORIGEM; SK_DESTINO; SK_LOCAL; SK_IDADE; SK_DATA_ENTRADA; SK_DATA_SAIDA; SK_DATA_TRIAGEM; SK_DATA_ENTRADA_SERVIÇO_OBSERVAÇÃO; SK_DATA_SAIDA_SERVIÇO_OBSERVAÇÃO; SK_HORE_ENTRADA; SK_HORE_SAIDA; SK_HORA_TRIAGEM; SK_HORA_ENTRADA_SERVIÇO_OBSERVAÇÃO; SK_HORA_SAIDA_SERVIÇO_OBSERVAÇÃO;	Inserção de novos casos

Lookup

Nome da transformação	DIM_UTENTE		
Tabela	DIM_UTENTE;	Localização/ Base de dados	dm
Match condition	select * from (select * from [dbo].[DIM_UTENTE]) [refTable] where [refTable].[Número de Cartão de Utente] = ? AND (DATA_INICIO <= ? AND DATA_FIM >= ?) OR (DATA_INICIO <= ? AND DATA_FIM = 99991231)		

Logical Data MAP

Logical Data Map		Project Name - BL_URGENCIAS		REFERENCE
Developer / Designer		ETL Package Name : Package.ctsx		BI01
David Gonçalves		SOURCE > TARGET		VERSION
TABELA_FACTOS		Staging Area (STG) - Data warehouse		2.0
TABELA_FACTOS		DATE		2014-09-20
TABELA_FACTOS		PURPOSE		TRANSFORMATIONS
TABELA_FACTOS		Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse		
TABELA_FACTOS		SOURCES (SO)		
TABELA_FACTOS		SOURCE TYPE		
TABELA_FACTOS		SQL Server 2012		
TABELA_FACTOS		DATABASE		
TABELA_FACTOS		dm		
TABELA_FACTOS		TABLE NAME		
TABELA_FACTOS		DIM_CAUSA		
TABELA_FACTOS		SK_CAUSA		
TABELA_FACTOS		DIM_UTENTE		
TABELA_FACTOS		SK_UTENTE		
TABELA_FACTOS		DIM_ELEGIBILIDADE		
TABELA_FACTOS		SK_ELEGIBILIDADE		
TABELA_FACTOS		DIM_EFR		
TABELA_FACTOS		SK_EFR		
TABELA_FACTOS		DIM_DESTINO		
TABELA_FACTOS		SK_DESTINO		
TABELA_FACTOS		DIM_ORIGEM		
TABELA_FACTOS		SK_ORIGEM		
TABELA_FACTOS		DIM_SERVIÇO		
TABELA_FACTOS		SK_SERVIÇO		
TABELA_FACTOS		DIM_LOCAL		
TABELA_FACTOS		SK_LOCAL		
TABELA_FACTOS		DIM_PRIORIDADE		
TABELA_FACTOS		SK_PRIORIDADE		
TABELA_FACTOS		DIM_IDADE		
TABELA_FACTOS		SK_IDADE		
TABELA_FACTOS		999		
TABELA_FACTOS		Facto		
TABELA_FACTOS		Int		
TABELA_FACTOS		FK		
TABELA_FACTOS		dm		
TABELA_FACTOS		autonumber		
TABELA_FACTOS		Começa em 1 e incrementa, de 1 em 1		
TABELA_FACTOS		[*] INT IDENTITY (1,1)		
TABELA_FACTOS		Atribuição da chave em concordância com a idade		
TABELA_FACTOS		((DATEDIFF('YEAR', [Data Nascimento]), [Data Entrada]) < 130) ? DATEDIFF('YEAR', [Data Nascimento]), [Data Entrada]) : 999		

TABELA_FACTOS

Logical Data Map		Project Name - BI_URGENCIAS				REFERENCE						
		ETL Package Name : Package.dtsx				BI01						
Developer / Designer	DATAFLOW	PURPOSE	SOURCE > TARGET	DATE								
David Gonçalves	TABELA_FACTOS	Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse	Staging Area (STG) - Data warehouse	2014-09-20								
TARGET (DW)		SOURCES (SO)										
TARGET TYPE	SQL Server 2012	SOURCE TYPE	SQL Server 2012	TRANSFORMATIONS								
DATABASE	dm	DATABAS	dm									
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE	DATABASE	TABLE NAME	COLUMN NAME	DATA TYPE	MAX	
FT_URGENCIAS	SK_DATA_ENTRADA								Data Entrada			(YEAR([Data Entrada]) * 10000) + (MONTH([Data Entrada]) * 100) + (DAY([Data Entrada]))
	SK_DATA_SAIDA								Data Saída			(YEAR([Data Saída]) * 10000) + (MONTH([Data Saída]) * 100) + (DAY([Data Saída]))
	SK_DATA_TRIAGEM								Data Triagem			(YEAR([Data Triagem]) * 10000) + (MONTH([Data Triagem]) * 100) + (DAY([Data Triagem]))
	SK_DATA_ENTRADA_SERVIÇO_OBSERVAÇÃO	Int				FK	dm	DIM_DATA	Data de Entrada no Serviço de Observação	datetime		Calcular SK_DATA_* Através das formulas: (YEAR([Data de Entrada no Serviço de Observação]) * 10000) + (MONTH([Data de Entrada no Serviço de Observação]) * 100) + (DAY([Data de Entrada no Serviço de Observação]))
SK_DATA_SAIDA_SERVIÇO_OBSERVAÇÃO								Data de Saída do Serviço de Observação			(YEAR([Data de Saída do Serviço de Observação]) * 10000) + (MONTH([Data de Saída do Serviço de Observação]) * 100) + (DAY([Data de Saída do Serviço de Observação]))	

TABELA_FACTOS

Logical Data Map		Project Name – BI_URGENCIAS		REFERE NCE								
		ETL Package Name : Package.dtsx <td>B101</td>		B101								
Developer / Designer		DATAFLOW		VERSIO N								
David Gonçalves		TABELA_FACTOS		2.0								
TARGET (DW)		PURPOSE		DATE								
		Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse		2014-09-20								
		SOURCES (SO)										
TARGET TYPE		SOURCE TYPE										
SQL Server 2012		SQL Server 2012										
DATABASE		DATABASE										
dm		dm										
TRANSFORMATIONS												
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE	DATABASE	TABLE NAME	COLUMN NAME	DATA TYPE	MAX	
FT_URGENCIAS	SK_HORA_ENTRADA								Hora Entrada			(DATEPART("Hh",[Hora Entrada]) * 100) + (DATEPART("mi",[Hora Entrada]))
	SK_HORA_TRIAGEM								Hora Triagem			(DATEPART("Hh",[Hora Triagem]) * 100) + (DATEPART("mi",[Hora Triagem]))
	SK_HORA_SAIDA								Hora Saída			(DATEPART("Hh",[Hora Saída]) * 100) + (DATEPART("mi",[Hora Saída]))
	SK_HORA_ENTRADA_SERVICO_OBSERVAÇÃO	Int				FK	dm	DIM_HORA	Hora de Entrada no Serviço de Observação	datetime		Calcular SK_HORA_* através das formulas: Entrada no Serviço de Observação))
SK_HORA_SAIDA_SERVICO_OBSERVAÇÃO								Hora de Saída do Serviço de			(DATEPART("Hh",[Hora de Saída do Serviço de Observação]) * 100) + (DATEPART("mi",[Hora de Saída do Serviço de Observação]))	

TABELA_FACTOS

Logical Data Map		Project Name - BI_URGENCIAS				REFERENCE						
		ETL Package Name : Package-dtsx				BI01						
Developer / Designer		DATAFLOW	PURPOSE	SOURCE > TARGET	DATE	VERSION						
David Gonçalves		TABELA_FACTOS	Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse	Staging Area (STG) - Data warehouse	2014-09-20	2.0						
TARGET (DW)		SOURCES (SO)										
TARGET TYPE		SQL Server 2012		SQL Server 2012								
DATABASE		Dm		STG_URGENCIAS								
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD	DEFAULT VALUE	KEY	DATA BASE	TABLE NAME	COLUMN NAME	DATA TYPE	MAX	TRANSFORMATIONS
FT_URGENCIAS	Número Episódio	numeric				UK			Número Episódio			Transformação de float para numeric
	Número Processo		Dimensão Degenerada			UK			Número Processo	float		
	Referência								Referência			Transformação para numérico e substituição de nulos por zero
	Isenção	int							Isenção			Transformação para numérico de S/N para 1,0 respectivamente
	Valor Cobrado						STG_URGENCIAS	URGENCIAS	Valor Cobrado (TP)	varchar	255	Substituição de nulos por zero
	Valor Faturado								Valor Faturado			
Tempo de permanência								Hora Entrada	DATEADD('mi',DATEPART('mi',[Hora Entrada]), DATEADD('hh',[Hora Entrada],[Data Entrada]))	datetime		DATEDIFF('mi',DATA_HORA_EN TRADA,DATA_HORA_SAIDA)
Tempo de espera								Hora Saída	DATEADD('mi',DATEPART('mi',[Hora Saída]), DATEADD('hh',[Data Saída],[Data Saída]))	datetime		DATEDIFF('mi',DATA_HORA_EN TRADA,DATA_HORA_SERVICO_OBS)

TABELA_FACTOS		Project Name – BI_URGENCIAS				REFERENCE						
Logical Data Map		ETL Package Name : Package.dtsx				BI01						
Developer / Designer		DATAFLOW	PURPOSE	SOURCE > TARGET		VERSION						
David Gonçalves		TABELA_FACTOS	Realizar o ETL (Extração, Transformação e Carregamento) da dimensão no data warehouse	Staging Area [STG] - Data warehouse		2.0						
TARGET (DW)		SOURCES (SO)				DATE						
						2014-09-20						
TARGET TYPE		SQL Server 2012	SOURCE TYPE	SQL Server 2012								
DATABASE		dqm	DATABASE	STG_URGENCIAS								
TABLE NAME	COLUMN NAME	DATA TYPE	TABLE TYPE	SCD TYPE	DEFAULT VALUE	KEY TYPE	DATABASE	TABLE NAME	COLUMN NAME	DATA TYPE	MAX	TRANSFORMATIONS
FL_URGENCIAS	Validação 1						STG_URGENCIAS	URGENCIAS	Número Processo	float	255	Identificar episódios sem Número de Processo
	Validação 2						STG_URGENCIAS	URGENCIAS	Elegível para pagamento (S/N)	nvarchar	255	Identificar episódios sem nenhum número identificativo (Número de Cartão de Utente, Número de Cartão de Saúde Europeu) <= 4 && ((Elegível para pagamento (S/N)) = "S") ? 1 : 0
	Validação 3	int	Métrica				STG_URGENCIAS	URGENCIAS	Número de Cartão de Utente	float		Identificar episódios sem nenhum número identificativo (Número de Cartão de Utente, Número de Cartão de Saúde Europeu) <= 4 && ((Elegível para pagamento (S/N)) = "S") ? 1 : 0
	Validação 4						STG_URGENCIAS	URGENCIAS	Número Beneficiário	float		Identificar episódios sem atribuição de cor ou que a cor não respeita a Triagem de Manchester (azul, verde, amarelo, laranja, vermelho)

