Mestrado em Engenharia Informática
Dissertação

Projeto UC-Num: desenvolvimento de uma Data Warehouse para a Universidade de Coimbra

Relatório Final

Aluno:
Ivo Emanuel Ferreira Gouveia
igouveia@student.dei.uc.pt

Orientador:
Professor Doutor Bruno Cabral
DEI

1 de Setembro de 2016

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Resumo

Para se conseguir uma boa gestão em organizações de grande dimensão, é cada vez mais importante efetuar decisões informadas, e a Universidade de Coimbra não é uma exceção. Atualmente, a recolha de indicadores de performance(KPI) é difícil, sendo também um trabalho demorado que, em muitos casos, pode atrasar as decisões de gestão ou até mesmo, no pior dos casos, a informação recolhida encontrar-se já desatualizada.

Para resolver este problema, a Universidade de Coimbra lançou o projeto UC-Num, cujo principal objetivo é a construção de uma solução de *business intelligence* para calcular, em tempo útil e de maneira automática, os vários indicadores necessários à boa gestão da universidade. Como objetivo deste estágio, pretende-se criar um novo módulo, com indicadores da área dos Serviços de Apoio Social para adicionar aos já existentes no sistema atual.

Assim, prentende-se criar uma *data mart* específica para a UC, realizando todos os passos típicos no desenvolvimento de um sistema de *business intelligence*, tais como: a recolha de indicadores, a criação de um plano de ETL e o desenvolvimento de *dashboards* interativos para o acesso à análise dos indicadores.

Palavras-chave

Business Intelligence, Data Warehouse, Dashboards, Online Analytical Processing, ETL, Indicadores de Performance

Conteúdo

Lista de Acrónimos

BI *Business Intelligence*. 10–13, 35, 36, 41

CDE *Community Dashboard Editor*. 23–25, 27

ETL *Extração Transformação e Carregamento*. 2, 3, 10–12, 17, 20, 30, 35, 36, 38, 40

KPI *Key Performance Indicators*. 1, 2

OLAP *Online Analytical Processing*. 1, 2, 10, 13, 17, 22, 23, 38, 41

PASEP Programa de Apoio Social a Estudantes através de Atividades de Tempo Parcial. 2, 18, 19, 21–23

SAMA Sistema de Apoio à Modernização Administrativa. 1

SASUC Serviços de Ação Social da Universidade de Coimbra. 2–5, 13, 17, 30, 34–36, 38

UC Universidade de Coimbra. 1, 2, 20

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Neste capítulo, é feito um enquadramento do projeto e seus objetivos, sendo também apresentada a estrutura usada na redação deste relatório de estágio.

1.1. Enquadramento

Instituições de grande dimensão, como a Universidade de Coimbra (UC), têm a necessidade de tomar decisões de forma rápida e informada. Porém, atendendo à dinâmica e complexidade inerentes a estas instituições, é cada vez mais difícil a boa gestão sem o auxílio de ferramentas especializadas para o apoio à decisão.

Atualmente na UC, a recolha dos *Key Performance Indicators* (KPI), necessários para auxiliar as decisões, é feita de modo manual recorrendo a diferentes sistemas operacionais, muitas das vezes únicos aos respetivos grupos operacionais, tornando-se um processo demorado e complexo.

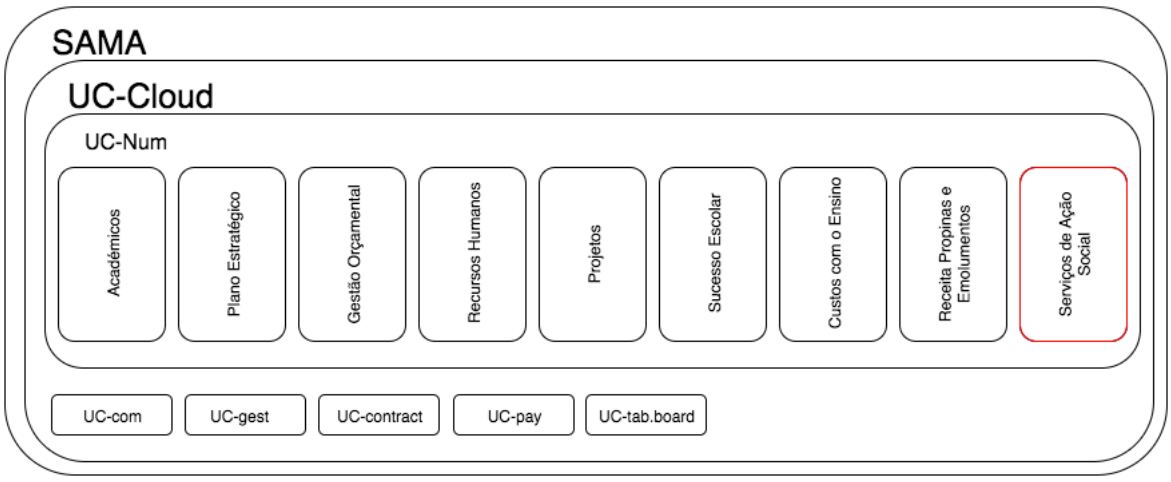

Figura 1.1: Enquadramento do Projeto UC-Num

Assim, foi criado o projeto UC-Num, integrado no projeto Sistema de Apoio à Modernização Administrativa (SAMA), que tem como objetivo a criação de uma *Data Warehouse* com um sistema OLAP, que permita um acesso fácil e rápido à

informação necessária para o apoio à decisão. A equipa de desenvolvimento responsável por este projeto é orientada pelo Professor Doutor Bruno Cabral e constituída por seis Mestres (Ana Ilharco, Carlos Lênduca, Beatriz Fragoso, Hugo Costa, Inês Domingues e João Correia) e um estagiário.
Este projeto encontra-se dividido em vários módulos, cada um ao encargo de um membro da equipa de desenvolvimento. De acordo com a figura 1.1 existem nove módulos: Académicos, Custos com Ensino, Gestão Orçamental, Plano Estratégico, Projetos, Receita com Propinas e Emolumentos, Recursos Humanos, Serviços de Ação Social e Sucesso Escolar. É importante salientar que este projeto já se encontra em desenvolvimento desde Setembro de 2013, portanto, alguns dos módulos referidos já se encontram desenvolvidos e validados, estando já disponíveis à gestão de topo da UC. De notar também que, devido ao estado atual do projeto, muitos dos requisitos já se encontram definidos, embora seja necessário garantir que o módulo a desenvolver seja capaz de os cumprir.
Já participo neste projeto desde Setembro de 2014, no entanto, apenas desempenhei funções de suporte à equipa de desenvolvimento. Até à data de início deste estágio foi desenvolvida por mim uma aplicação de gestão de *logs* para notificar os membros da equipa quando ocorrem erros no etl ou no sistema de BI, um sistema de upload de ficheiros que permite correr as correspondentes transformações automaticamente e também instalação e configuração de ferramentas de apoio ao desenvolvimento como, por exemplo, Gitlab e Taiga.
Este estágio é referente ao desenvolvimento do módulo de Serviços de Ação Social da Universidade de Coimbra (SASUC), que inclui os âmbitos de Alojamento, Apoios Diretos, Apoio à Infância e Apoios PASEP (PASEP). Pretende-se, no decorrer deste estágio, desenvolver uma *Data Mart* para efetuar o cálculo de vários KPI, por exemplo, Taxa de ocupação das residências universitárias, Montantes atribuídos em bolsas de estudo ou número de crianças inscritas nos serviços de apoio à infância, com o objetivo de apoiar decisões de gestão do SASUC

1.2. Objetivos

Como já foi referido, o objetivo deste estágio é o desenvolvimento de uma *Data Mart* para os SASUC, incluída na *Data Wharehouse* do projeto UC-Num. Para tal, é necessário fazer um levantamento de requisitos com o propósito de identificar os KPI com maior interesse para a sua gestão, a forma como estes são calculados e quais os dados necessários para efetuar o respetivo cálculo. Com esta informação é feito o planeamento do modelo de dados da *Data Mart* que seja capaz de responder aos indicadores encontrados. De seguida, é necessário desenvolver processo *Extração Transformação e Carregamento* (ETL), com o objetivo de recolher e transformar toda a informação necessária, proveniente das fontes de dados dos SASUC. Por fim, os KPI devem ser disponibilizados para consulta através de ferramentas de OLAP. Estas ferramentas consistem em *dashboards* interativos, disponíveis via *web*, que permitem aos utilizadores fazer uma análise dos dados utilizando *queries ad hoc*. Uma vez desenvolvidos, é preciso realizar um conjunto de teste funcionais para garantir que todos os requisitos estão a ser cumprido. Por fim, procede-se a uma fase de validação com o objetivo de confirmar os resultados através dos *dashboards* estão de acordo com a realidade.

1.3. Estrutura do Relatório

No Capítulo 2, apresenta-se a especificação de requisitos funcionais e não funcionais transversais ao projeto UC-Num e específicos ao módulo de SASUC.

No Capítulo 3, é descrita a arquitetura do sistema, incluindo uma análise comparativa das tecnologias disponíveis.

No Capítulo 4, é apresentado a fase de implementação do módulo de SASUC, desde o processo de ETL até ao desenvolvimento dos *dashboards*.

No Capítulo 5, são descritos os testes funcionais realizados sobre os vários *dashboards* assim com o atual estado da validação deste módulo

No Capítulo 6, encontra-se descrito o processo de desenvolvimento do projeto e o plano de trabalhos para ambos os semestres do estágio.

Finalmente, no Capítulo 7 é feita uma breve reflexão sobre este estágio.

Capítulo 2

Definição de Requisitos

2.1. Levantamento de Requisitos

Como em qualquer projeto de engenharia de software, a fase de levantamento de requisitos é de grande importância, e permite que todos os *stakeholders* envolvidos tenham uma ideia mais concreta do que devem ser as funcionalidades do sistema. Neste caso, como se pretende a construção de um módulo para o sistema UC-Num e os requisitos gerais já se encontram definidos, o foco deste levantamento de requisitos recaiu na especificação dos indicadores deste módulo. Para tal, foram realizadas reuniões com vários membros dos SASUC, a fim de recolher a informação necessária para construir e validar as fichas de indicadores. No entanto, é de salientar que, no início do meu estágio muitos indicadores já se encontravam em fase de especificação pois grande parte são comuns ao módulo do plano estratégico da Universidade de Coimbra, que já se encontrava em desenvolvimento durante o ano anterior ao início deste estágio.

Foram também realizados protótipos rápidos, não só para auxiliar na especificação destes indicadores, mas também para evitar grandes alterações aquando a fase de desenvolvimento. Este tipo de prototipagem permite mostrar de antemão o funcionamento do módulo em desenvolvimento aos *stakeholders* intervenientes no levantamento de requisitos. Na figura 2.1 podemos ver um dos protótipos desenvolvidos.

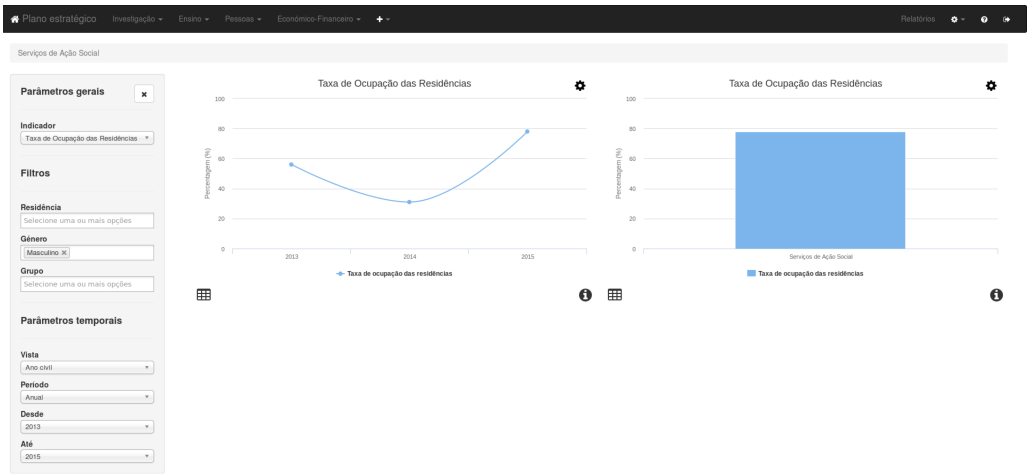

Figura 2.1: Exemplo de um protótipo rápido para um indicador de SAS

2.2. Especificação de Requisitos

Para a especificação de requisitos, a equipa escolheu um dos modelos mais usados, o modelo FURPS+ (*Functional Usability Reliability Performance and Supportability*, o + representa categorias adicionais e.g. requisitos físicos ou de implementação) que deriva do modelo FURPS, proposto por Robert Grady da *Hewlett-Packard*[7]. Este modelo contém duas categorias: requisitos funcionais e requisitos não funcionais. Os requisitos funcionais descrevem as funcionalidades da aplicação enquanto que os requisitos não funcionais descrevem os atributos de qualidade do sistema. Estes requisitos são no fim priorizados de acordo com a sua importância (Elevada, Média, Baixa).

2.2.1. Requisitos Funcionais

Como o objetivo deste estágio é o desenvolvimento de um módulo adicional para um sistema que já se encontra em desenvolvimento há dois anos, muitos dos requisitos já se encontram especificados e são transversais a todos os módulos já implementados ou em desenvolvimento. A tabela 2.1 lista os requisitos gerais do sistema. Por sua vez, a tabela 2.2 lista os indicadores referentes ao módulo de SASUC a desenvolver na segunda fase deste estágio. As fichas de indicadores usadas para o levantamento dos requisitos da tabela 2.2 podem ser consultadas as fichas de indicadores levantados no ficheiro Fichas_Indicadores.pdf, assim como o documento com todos os requisitos da aplicação UC-Num, disponíveis como anexos digitais.

Código	Designação	Prioridade	Descrição
RF_GE_004	Navegação entre módulos	Média	O utilizador deve conseguir aceder a todos os restantes módulos, este acesso deve ser possível a qualquer momento.

Código	Designação	Prioridade	Descrição
RF_GE_005	Navegação interna	Elevada	A aplicação deve permitir ao utilizador efetuar *drill down* nos dados que pretende visualizar, esta "descida"no detalhe da informação deve ser efetuada diretamente nos dados que vão sendo apresentados na vista de snapshot. Para que seja possível ao utilizador efetuar *roll up* dos dados, isto é, subir no nível de detalhe, a aplicação deve disponibilizar uma forma de navegação estrutural (*breadcrumbs*), que vá acrescentando o nível onde o utilizador se encontra. A hierarquia de níveis pode ser diferente consoante a área de cada módulo.
RF_GE_006	Parâmetros gerais	Elevada	O utilizador deve ter disponível os diversos parâmetros (seleção de indicadores, agregadores e/ou filtros) que são permitidos modificar nos dados que este está a visualizar.
RF_GE_007	Parâmetros de tempo	Elevada	Deve ser permitido ao utilizador modificar os parâmetros temporais (dimensão tempo) a aplicar nos dados que este está a visualizar.
RF_GE_008	Esconder parâmetros	Baixa	Deve ser permitido ao utilizador esconder a barra onde se encontram os parâmetros gerais e de tempo.
RF_GE_009	Secção de ajuda	Elevada	A aplicação deve disponibilizar uma secção de ajuda ao utilizador, transversal a todos os módulos, para esclarecer quaisquer dúvidas que sejam suscitadas nos utilizadores - FAQ.
RF_GE_010	Informação auxiliar	Elevada	Cada vista de dados disponibilizada ao utilizador (gráfico ou tabela) deve ser acompanhada de dois mecanismos que permita consultar informação: uma referente aos dados que são apresentados, corresponde a um botão de informação (remetendo para a secção de ajuda) e outra de navegação (sugestões, erros, etc).

Código	Designação	Prioridade	Descrição
RF_GE_011	Visualização: gráfico - tabela	Elevada	A aplicação deve permitir ao utilizador visualizar a informação apresentada num gráfico em formato de tabela e vice-versa. Os dados apresentados na tabela deverão, pelo menos, corresponder à informação que se encontra no gráfico.
RF_GE_012	Exportar informação da tabela/gráfico	Baixa	Exportar para formato excel ou csv a informação presente nas tabelas de análise. Exportar gráfico como imagem.
RF_GE_014	Zoom de gráficos	Baixa	Efetuar *zoom in* e *out* de um dos gráficos.

Tabela 2.1: Requisitos Funcionais Gerais

Código	Designação	Prioridade	Descrição
IND_PES_SAS_ALOJ_001	Taxa de Ocupação das Residências	Elevada	Percentagem de lugares ocupados face aos lugares disponíveis para alojamento.
IND_PES_SAS_ALOJ_002	Número de Residências	Elevada	Número de residências a cargo dos Serviços de Ação Social.
IND_PES_SAS_ALOJ_003	Número de Quartos	Elevada	Número de quartos existentes em cada residência.
IND_PES_SAS_ALOJ_004	Número de Camas	Elevada	Número de camas previstas na capacidade de residências.
IND_PES_SAS_ALOJ_005	Número de Alojados	Elevada	Número de pessoas alojadas nas residências.
IND_PES_SAS_APINF_011	Taxa de Utilização dos Serviços de Apoio à Infância	Elevada	Número de crianças inscritas em relação à capacidade das valências de apoio à infância.
IND_PES_SAS_APINF_012	Número de Crianças Inscritas	Elevada	Número de crianças inscritas, a frequentar ou que frequentaram os serviços de apoio à infância no período considerado.
IND_PES_SAS_BOLS_015	Beneficiários de Apoios Diretos	Elevada	Número de beneficiários de apoios diretos.

Código	Designação	Prioridade	Descrição
IND_PES_SAS_BOLS_017	Candidatos a Apoios Diretos	Elevada	Número de estudantes candidatos a apoios diretos da ação social na UC.
IND_PES_SAS_BOLS_018	Candidaturas indeferidas	Elevada	Número de processos de candidatura a apoios diretos indeferidos.
IND_PES_SAS_BOLS_019	Montantes Anuais de Apoios Diretos	Elevada	Montantes anuais dos apoios diretos atribuídos aos estudantes.
IND_PES_SAS_PASEP_055	Benefícios Atribuídos	Elevada	Montante de benefícios atribuídos em ofertas PASEP.
IND_PES_SAS_PASEP_020	Número de Ofertas PASEP	Elevada	Numero de ofertas PASEP disponíveis para os alunos.
IND_PES_SAS_PASEP_021	Número de Candidatos	Elevada	Número de candidatos a ofertas.
IND_PES_SAS_PASEP_022	Número de Beneficiários	Elevada	Número de estudantes colocados em ofertas.

Tabela 2.2: Lista de Indicadores do Módulo de SAS

2.2.2. Requisitos Não Funcionais

De maneira semelhante aos requisitos funcionais, que são transversais a todos os módulos, os requisitos não funcionais já se encontram especificados e devem ser respeitados durante o desenvolvimento do módulo; a tabela 2.3 lista os requisitos não funcionais desta aplicação.

Código	Designação	Prioridade	Descrição
RNF_S_001	Atualização de dados	Elevada	Processo ETL e atualização da DW e cubo OLAP devem ser automáticos.
RNF_S_002	Compatibilidade (*browser*)	Elevada	Aplicação web deve ser compatível com os *browsers* mais modernos, a partir das versões mencionadas: Internet Explorer 9; Firefox 20, Safari 6 e Chrome 31.
RNF_S_003	Compatibilidade (Sistema Operativo)	Média	Como sistema operativo para os servidores deve ser suportada a distribuição de *Linux Red Hat - CentOS 6.x.*

Código	Designação	Prioridade	Descrição
RNF_S_004	Licenças	Elevada	A aplicação deve ser desenvolvida e disponibilizada através de software gratuito.
RNF_S_005	Monitorização de erros	Média	Deve existir um mecanismo para gestão de logs.
RNF_O_001	Hardware	Elevada	O software deve executar numa máquina com as seguintes características mínimas: 4Gb de RAM, 20Gb de espaço em disco e um processador dual core, não tem necessariamente de ser um ambiente de 64 bits. Estas caraterísticas estão diretamente relacionadas com as mínimas exigidas pelo software que foi selecionado para desenvolvimento e disponibilização da aplicação.
RNF_O_002	Confidencialidade na comunicação	Elevada	Deve ser utilizado o protocolo HTTPS em toda a comunicação entre os utilizadores e a aplicação.
RNF_O_003	Autenticação	Elevada	Para aumentar a segurança da aplicação a autenticação e validação do acesso à aplicação deve ser efetuada com as credenciais dos utilizadores do sistema LDAP da UC.

Tabela 2.3: Requesitos Não Funcionais

Capítulo 3

Arquitetura

Este capítulo apresenta uma arquitetura geral do sistema de *Business Intelligence* (BI) a desenvolver, uma reflexão sobre as tecnologias existentes que permitam implementar este sistema. Por fim, é apresentado um modelo de dados para a respetiva *data mart*. De notar que o projeto UC-Num já está em desenvolvimento há cerca de dois anos, portanto, a arquitetura geral já se encontra definida assim como as tecnologias a usar já se encontram escolhidas.

3.1. Arquitetura Global

A construção de um sistema de BI divide-se em três fases principais:

1. Criaçao de um processo de ETL, não só para obter as informações necessárias das fontes de dado, mas também realizar operações de limpeza e transformação sobre os dados recolhidos e, finalmente, carrega-los para a *Data Mart*.

2. Criação de cubos OLAP, usando um servidor de BI

3. Criação de um *frontend* (*dashboards*) para visualização da informação obtida através dos vários cubos OLAP

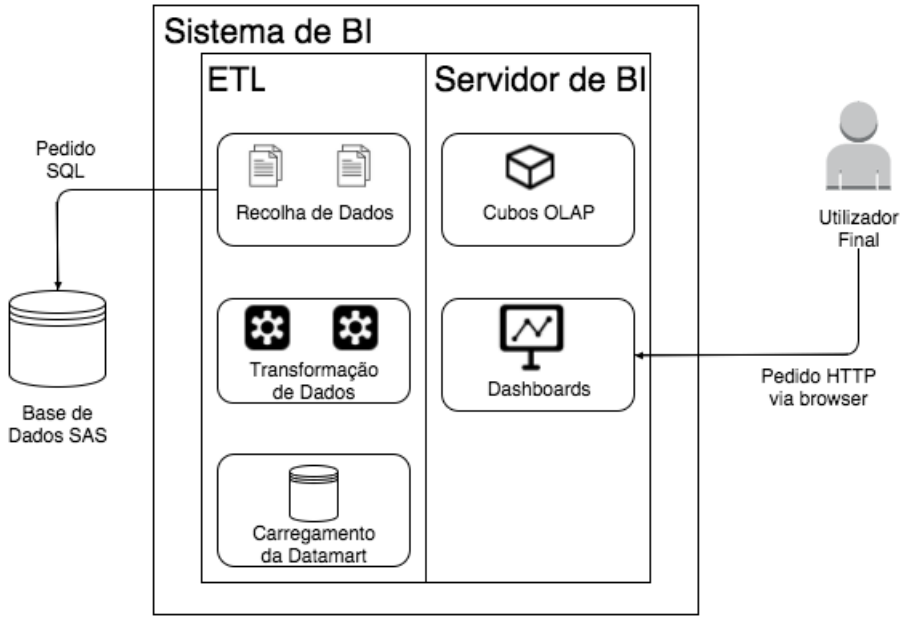

Figura 3.1: Arquitetura de Alto Nível do Sistema

3.2. Tecnologias

Como já foi referido, as tecnologias a usar para este projeto já se encontram escolhidas. No entanto, não só para formação do estagiário nas tecnologias da área, mas também para referência da equipa deste projeto, foi realizada uma reflexão sobre as várias tecnologias existentes e quais poderiam ser uma alternativa às escolhidas.

De acordo com a arquitetura do projeto, são necessárias ferramentas para cada uma das fases de desenvolvimento de um sistema de BI. É necessário um motor de base de dados, uma ferramenta de ETL e um servidor de BI. De notar que não existem fundos para aquisição de licenças de software, portanto apenas se fez uma análise a ferramentas gratuitas.

3.2.1. Bases de dados

Os motores de bases de dados são indispensáveis para a construção de um servidor de BI. É num motor de base de dados que toda a informação da *data mart* fica armazenada, sendo também nele que se armazenam as informações da área de estágio durante todo o processo de ETL.

Os motores de base de dados relacionais são os mais usados num projeto desta dimensão e, tendo em conta, a limitação de licença gratuita, destacam-se dois grandes nomes: o *MySQL*[8] e o *PostgreSQL*[9]. Ambos os motores são viáveis para este projeto e é de salientar que o *PostgreSQL* tem mostrado melhor performance, principalmente devido ao uso de vistas materializadas e a um leque mais abrangente de índices. Esta análise pode ser consultada no anexo A

Em alternativa às bases de dados relacionais, uma outra possibilidade seria um motor de base de dados *NoSQL*[10]. Este tipo de motor é cada vez mais popular, pois possui

uma escalabilidade muito superior aos motores de bases de dados relacionais. Existem 4 principais categorias de bases de dados *NoSQL*:

- **Chave-valor**: este tipo de base de dados armazena a informação emparelhada com uma chave, semelhante a uma *hashtable*. E.g., *DynamoDB, Redis e Voldemort.*
- **Armazenamento por colunas**: armazena os dados por colunas, em vez de linhas, o que permite uma maior rapidez de acesso aos dados, pois encontram-se de forma contígua no disco, diminuindo o tempo de *IO*. E.g., *Cassandra e MonetDB.*
- **Armazenamento por documentos**: semelhante ao modelo chave-valor, cada chave corresponde a um documento com dados semi-estruturados (e.g., XML ou JSON). E.g., *CouchDB e MongoDB.*
- **Grafos**: este modelo baseia-se em grafos para representar relações entre os dados. E.g., *AllegroGraph e Neo4j.*

Apesar da sua popularidade, estes motores possuem limitações no que toca a um sistema de BI. Em primeiro lugar, não possuem a capacidade de lidar tão eficazmente com *queries* complexas como os motores relacionais, tornando-os numa escolha inferior no geral, sendo apenas viáveis em algumas situações (em que a complexidade dos dados não seja elevada e/ou que o tamanho dos dados seja muito elevado). Em segundo lugar, estas ferramentas ainda não amadureceram o suficiente, o que as torna instáveis quando comparadas com os motores relacionais que vêm a ser desenvolvidos e aperfeiçoados há mais de 40 anos.

3.2.2. Processo de ETL

Esta fase representa grande parte do esforço do processo de construção de uma *data warehouse* e, como tal, a escolha de ferramentas adequadas, que facilitem o seu desenvolvimento, é muito importante. Existem vários critérios para a escolha de uma ferramenta de ETL, mas os principais são: motores de bases de dados suportados, agendamento automático dos processos desenvolvidos, tipo de fonte de dados suportados e inserção de código personalizado. Tendo em conta estes critérios e o facto de apenas se pretender usar ferramentas que não exijam o pagamento de licenças, foram escolhidas duas para análise: *Pentaho Data Integration*[11] e *Jaspersoft ETL*[12]. Apesar das semelhanças entre ambas, *Pentaho data integration* destaca-se pela sua usabilidade, processamento em paralelo e pela facilidade de agendamento dos processos. No anexo B encontra-se uma análise mais detalhada destas ferramentas.

3.2.3. Servidor de BI

Por fim, nesta última fase do processo de construção de uma *data wharehouse*, é necessário escolher a ferramenta onde a informação é estruturada e apresentada ao utilizador final através de *dashboards*. Mais uma vez, como as ferramentas devem respeitar a restrição de não exigir pagamento de licenças, foram escolhidas para análise ferramentas com este critério: *Pentaho BI Server*[13] e *Jasper Reports Server*[14].

Ambas são muito conhecidas, principalmente pela sua facilidade de instalação e usabilidade, no entanto, *Jasper Reports Server* está orientada para a geração de relatórios enquanto que *Pentaho BI Server* contém funcionalidades mais abrangentes de um sistema de BI, destacando-se a capacidade de criação de *dashboards*, disponibilizados como páginas web através do plugin CDE, e também a integração com a ferramenta *Mondrian* para a geração de cubos OLAP, usados para a análise dos dados, aplicando operações típicas de BI como *drill-down, roll-up, drill-across, slice and dice*. No Anexo C pode-se encontrar uma análise com maior pormenor destas tecnologias.

3.3. Seleção de Tecnologias

As tecnologias selecionadas estão limitadas às que foram decididas e se encontram em utilização nos últimos dois anos. No entanto, a análise efetuada, apesar de ser principalmente académica, dado o custo de mudança de tecnologias ser demasiado elevado para o projeto nesta fase, prova que atualmente ainda está adequada às necessidades do projeto. As ferramentas escolhidas são:

- **Base de dados:** *PostgresSQL* devido, principalmente, à existência de um maior leque de índices e vistas materializadas, que aumentam consideravelmente a sua performance.
- **ETL:** *Pentaho Data Integration* destaca-se, principalmente, pela facilidade de agendamento dos processos de ETL e da sua usabilidade.
- **Servidor de BI:** O facto de permitir a criação de *dashboards* na forma de páginas *web*, associado ao facto de já integrar uma ferramenta que permite a criação de cubos OLAP, tornam *Pentaho BI Server* a melhor escolha para o desenvolvimento deste sistema de BI.

3.4. Modelo de Dados

Na construção de uma *data mart*, um passo muito importante é o desenho de um modelo de dados que seja capaz de responder a todos os indicadores especificados. Para tal, optou-se por um diagrama em estrela, proposto por Ralph Kimball[15], que é constituído por tabelas de factos e por dimensões. O modelo resultante para o módulo de SASUC é constituído por sete tabelas de factos e por catorze dimensões. Na figura 3.2 é apresentado o modelo proposto para a *data mart* dos SASUC; na tabela 3.1 encontram-se uma descrição das dimensões existentes no modelo de dados; e na tabela 3.2 está detalhada a granularidade mínima das tabelas de factos deste modelo.

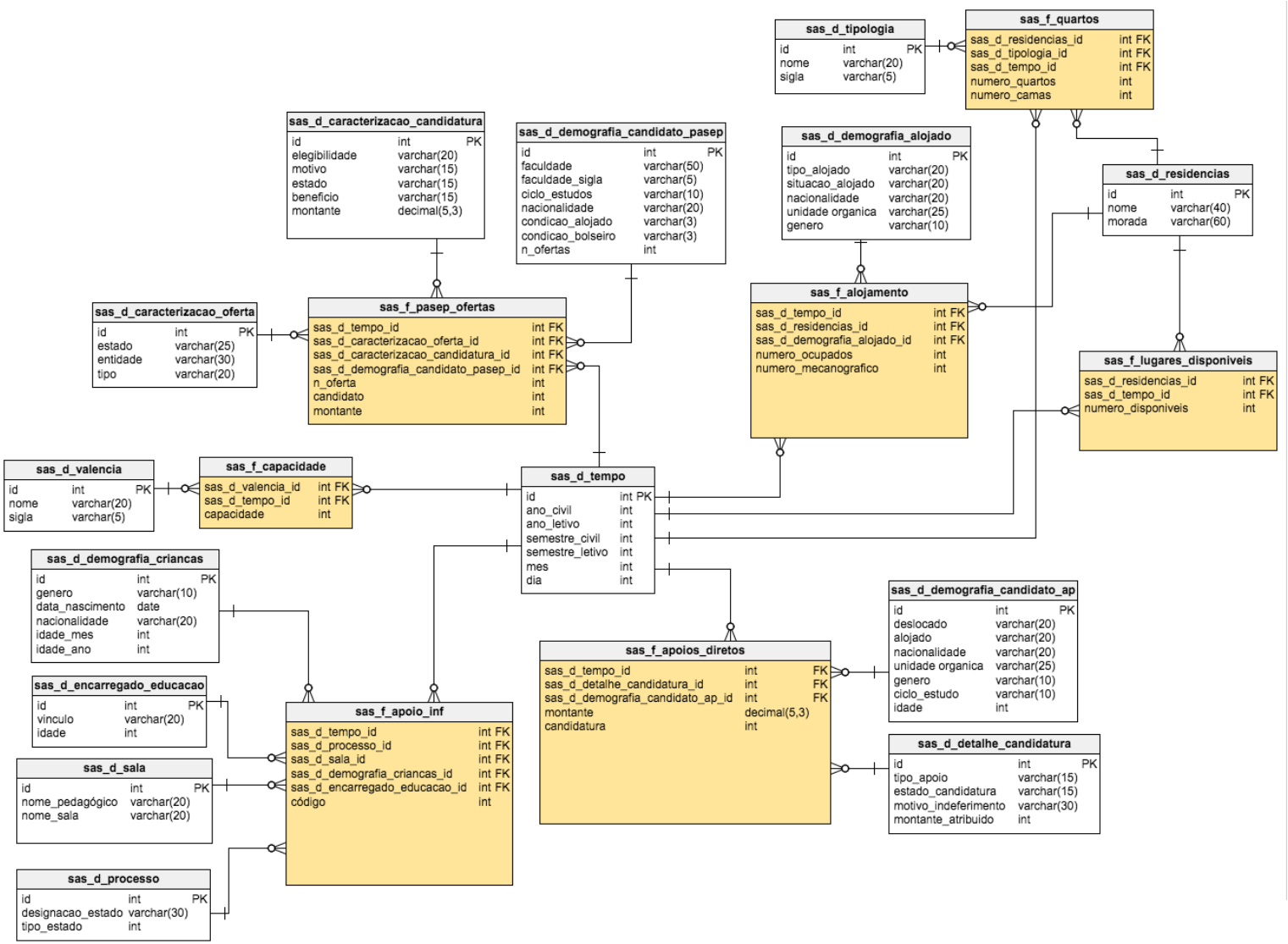

Figura 3.2: Modelo de dados do módulo de SAS

Dimensão	Descrição
sas_d_tipologia	Dimensão que retrata os vários tipos diferentes de tipologias de quartos existentes nas residências da UC.
sas_d_residencias	Dimensão que identifica as diferentes residências existentes na UC.
sas_d_demografia_alojado	Dimensão que descreve um alojado de uma residência como o seu género, a que unidade orgânica pertence, a sua nacionalidade, a sua situação como bolseiro e o tipo de regime.
sas_d_demografia_candidato_pasep	Dimensão que descreve o candidato a uma oferta PASEP, como a sua unidade orgânica, o ciclo de estudos, nacionalidade, a sua condição como alojado ou bolseiro e o número de ofertas em que participou.
sas_d_caracterizacao_candidatura	Dimensão que descreve uma candidatura PASEP que contém informações como o seu estado (em curso ou concluído), qual o tipo de benefício (Alimentação, Propinas ou Alojamento), a sua elegibilidade (aceite ou não aceite) e, caso exista, o motivo de não elegibilidade.
sas_d_caracterizacao_oferta	Dimensão que retrata as ofertas PASEP por estado, entidade proponente e a sua tipologia.
sas_d_valencia	Dimensão que identifica as valências existentes nos serviços de apoio à infância.
sas_d_demografia_criancas	Dimensão que descreve as crianças de acordo com o seu género, data de nascimento, nacionalidade e idade.
sas_d_encarregado_educacao	Dimensão que retrata os os vínculos do encarregado de educação com a UC.
sas_d_sala	Dimensão que identifica as várias salas existentes nos serviços de apoio à infância.
sas_d_processo	Dimensão que descreve o estado de um processo em frequência nos serviços de apoio à infância.
sas_d_demografia_candidato_ap	Dimensão que descreve um candidato aos apoios diretos e contém a sua nacionalidade, género, idade, a que ciclo de estudos e unidade orgânica pertence, a sua condição de alojado e de deslocado.
sas_d_detalhe_candidatura	Dimensão que descreve uma candidatura aos apoios diretos como o tipo de apoio, qual o estado da candidatura, montante e, caso aplicável, o motivo de indeferimento.

Dimensão	Descrição
sas_d_tempo	Dimensão que retrata a linha temporal usada em todas as tabelas de factos e que permite restringir a análise de dados a um período de tempo.

Tabela 3.1: Dimensões existentes na *data mart* dos SAS.

Tabela de factos	Granularidade mínima
sas_f_quartos	Número de quartos/camas existentes em uma residência específica e de uma determinada tipologia, para um mês específico.
sas_f_lugares_disponíveis	Número de lugares disponíveis numa residência, para um dia específico.
sas_f_alojamento	Um aluno alojado numa determinada residência para um dia específico.
sas_f_apoios_diretos	Uma candidatura a um apoio direto por um determinado aluno, para um ano específico.
sas_f_apoio_inf	Um processo de uma criança com um encarregado de educação e que frequenta uma determinada sala, para um dia específico
sas_f_capacidade	Número de lugares para uma determinada valência (creche ou jardim de infância) para um mês específico
sas_f_passep_ofertas	Uma candidatura feita por um aluno para uma determinada oferta, para um ano específico.

Tabela 3.2: Granularidade das tabelas de factos

Capítulo 4

Implementação

Neste capítulo são apresentados os detalhes de implementação do módulo de SASUC desenvolvido para a ferramenta UC-Num, é descrito todo o processo de ETL, a construção do cubo OLAP e, por fim, serão apresentados os respetivos *dashboards* finais.

4.1. ETL

O processo de ETL é a principal fase de construção de uma *Data Mart*, a sua grande complexidade faz deste o processo com mais importância, sendo este base para a construção de uma *Data Mart* que seja capaz de responder, de forma correta e com rapidez, aos indicadores levantados. A grande maioria dos problemas de projetos nesta área são provenientes do ETL.

Este processo pode ser dividido três fases principais:

- Extração - nesta fase é efetuada a recolha dos dados para serem carregados na área temporária, provenientes das respetivas fontes. É sobre estes dados que se realizam as duas fases seguintes.
- Transformação - durante esta fase é efetuada uma limpeza dos dados como, por exemplo, a remoção de duplicados, e uma transformação e normalização dos mesmos, para que se adequem ao modelo de dados desenhado.
- Carregamento - por fim, quando todos os dados se encontram de acordo com o modelo de dados planeado, é efetuado o carregamento destes para *Data Mart*, permitindo a realização do cubo OLAP e as análises subsequentes.

4.1.1. Transformações Pentaho Data Integration

Nesta subsecção são descritos alguns detalhes sobre a ferramenta usada para a realização do ETL e também a descrição de transformações criadas durante o seu desenvolvimento, de modo a compreender a complexidade deste processo.

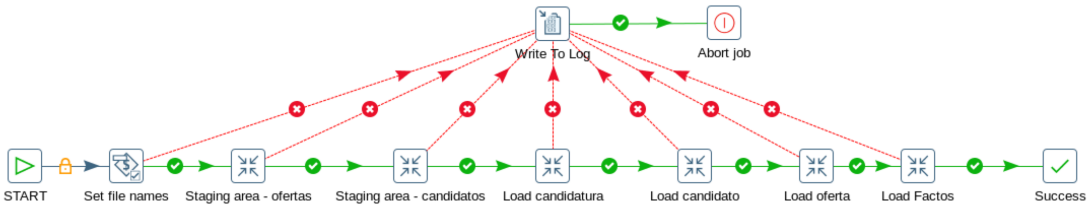

Figura 4.1: *Job* usado no âmbito do PASEP

O *flow* de uma transformação nesta ferramenta pode-se dividir em dois componentes principais: a transformação propriamente dita, e *jobs* que servem para correr outros *jobs* e/ou as transformações necessárias a este processo. É possível também definir variáveis de ambiente necessárias para correr com sucesso as suas transformações e ainda criar exceções e escrita de logs, para o caso de ocorrerem erros durante esta fase. Por exemplo, o *job* do PASEP cria uma variável com o nome do ficheiro *excel* onde se encontram os dados usados na transformação da área de estágio. Caso a transformação não consiga ler o ficheiro, o *job* aborta a transformação e escreve no *log* o erro encontrado. Isto permite uma maior facilidade no *debug* das transformações quer no seu desenvolvimento quer no futuro durante carregamentos subsequentes.

4.1.2. Transformações da área de estágio

Para o âmbito do PASEP foi criado um *job*, representado na figura 4.1.2, que cria duas variáveis com o nome dos ficheiros a ler(recebido como parâmetro na linha de comandos) e corre as transformações necessárias para o carregamento dos dados necessários para a criação da respetiva *Data Mart*. Este carregamento é feito anualmente.

Para o carregamento de dados para a área de estágio deste mesmo âmbito foram criadas duas transformações, uma para cada ficheiro, para carregar a informação necessária sobre as ofertas PASEP existentes e outra para as informações respetivas aos candidatos a estas ofertas.

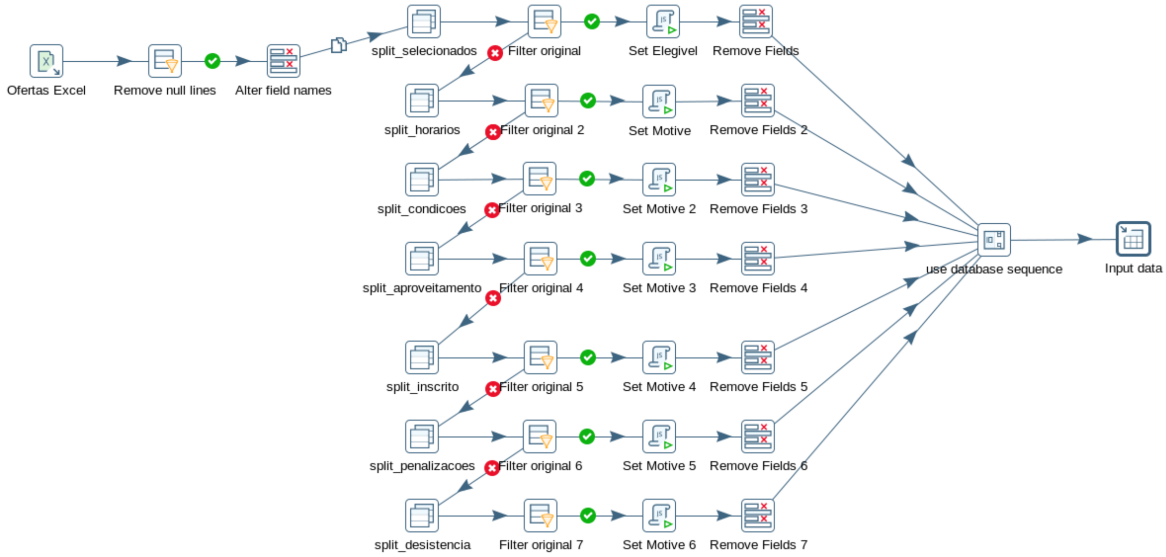

Figura 4.2: Transformação dos dados de ofertas no âmbito do PASEP

Como se pode ver na figura 4.1.2, esta transformação começa por eliminar linhas vazias que existem no ficheiro, desnecessárias para o carregamento de dados na área temporária. Uma vez eliminadas, divide-se os candidatos elegíveis dos não elegíveis usando um filtro e, para cada um, é criada uma variável que representa a elegibilidade e, caso aplicável, outra para o motivo de indeferimento. Por fim, estes dados são introduzidos na área de estágio da *Data Mart* para as ofertas.

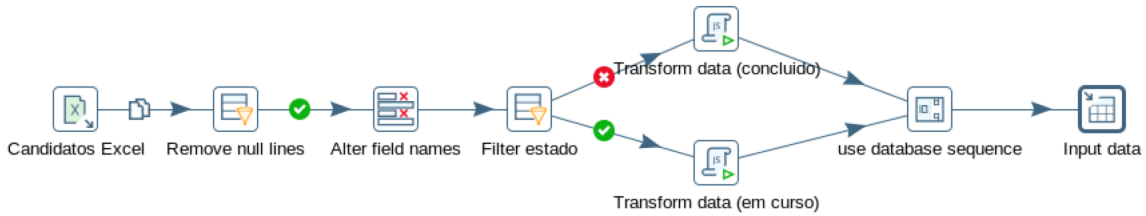

Figura 4.3: Transformação dos dados de candidatos no âmbito do PASEP

Esta segunda transformação carrega as informações dos candidatos correspondentes às ofertas da transformação anterior. Mais uma vez, esta transformação começa por eliminar linhas em branco e, de seguida, os dados são filtrados de acordo com o estado da sua candidatura (caso seja em curso é necessário atribuir um montante igual a zero pois no ficheiro original estes campos podem não se encontrar preenchidos) é feita também uma normalização dos dados da nacionalidade e da faculdade, mais precisamente retirar as letras maiúsculas destes campos, de forma a evitar ambiguidade dos dados existentes (portuguesa e Portuguesa). Feitas as transformações necessárias, estes dados são carregados para a respetiva área de estágio, terminando assim esta fase do carregamento.

Para o carregamento da área de estágio do âmbito de Apoios Diretos, foram criadas várias transformações (uma para cada vista disponível na fonte de dados), onde é feita uma seleção prévia dos dados carregados. Atualmente, existem dados com o nome da residência seguido de "(Inactiva)"que não devem ser introduzidos na *Data Mart*. Com este exemplo já se pode encontrar um dos grandes desafios do ETL. De seguida mostra-se um pequeno exemplo de dados encontrados como nome de residências:

- João Jacinto (Inactiva)
- João Jacinto (Desactivada)
- POLOII-1(Inactivo)

Como solução foi desenvolvido um pequeno código *javascript*, onde foram introduzidas todas estas exceções, e que altera o nome da residência para "REMOVER", de seguida, a *stream* de dados é filtrada por esse nome. Neste caso, dado o pequeno valor de dados (existiam 37 nomes distintos em vez dos supostos 24) foi possível resolver este problema, no entanto, em casos com volumes de dados bastantes superiores, torna-se uma tarefa bastante difícil de resolver. Neste caso o problema já foi reportado e encontra-se atualmente a ser corrigido.

Figura 4.4: Transformação dos dados de residências no âmbito do Alojamento

Por fim destaca-se também o âmbito dos Apoios Diretos, onde a fonte de dados é um *webservice* externo à UC. Para tal, é necessário fazer um pedido pedido SOAP. Este pedido retorna um XML com os dados necessários que, no fim de transformados, são inseridos na respetiva área de estágio.

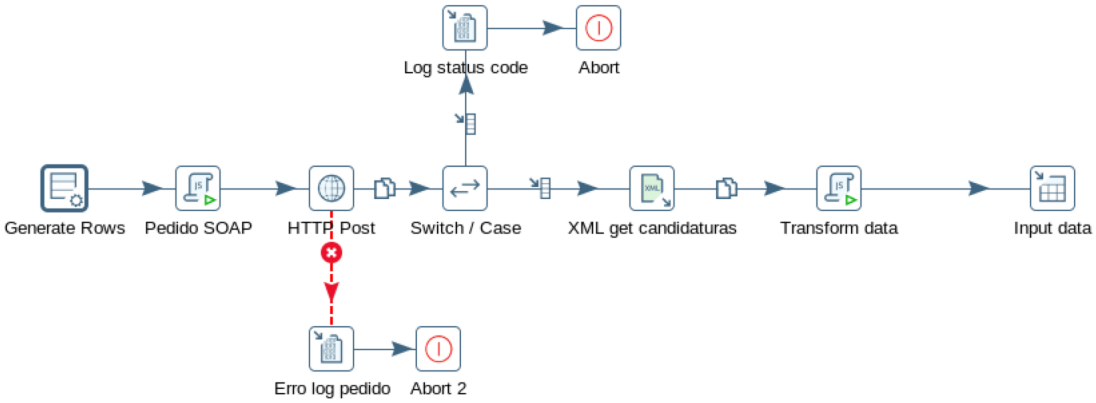

Figura 4.5: Transformação dos dados de candidaturas no âmbito de Apoios Diretos

4.1.3. Carregamento das dimensões

Como se pode ver no modelo de dados, a *Data Mart* do âmbito PASEP tem três dimensões para preencher: dimensão do candidato, dimensão da candidatura e dimensão das ofertas. Foi desenvolvida, para cada uma, a sua transformação de carregamento.

Na transformação de carregamento da dimensão dos candidatos e das ofertas é feito o carregamento diretamente da área de estágio, pois os dados já se encontram de acordo com o necessário para serem inseridos na *Data Mart*. É feita apenas uma operação para verificar a consistência dos dados e, por fim, são carregados nas respetivas tabelas de dimensões.

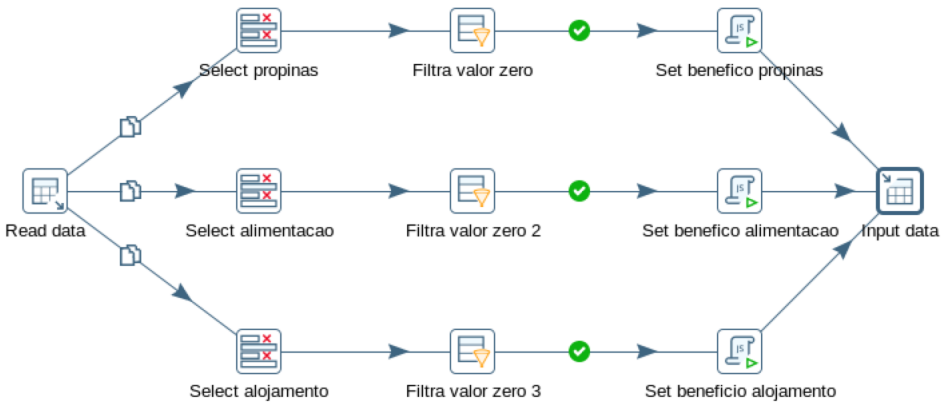

Figura 4.6: Transformação para carregamento da dimensão candidatura no âmbito PASEP

No caso da dimensão das candidaturas é ainda necessário descriminar as entradas por tipo de benefício atribuído, sendo gerado um valor identificador para cada tipo de benefício. Por fim, estes dados são inseridos também na respetiva tabela de dimensão de candidaturas.

4.1.4. Carregamento dos factos

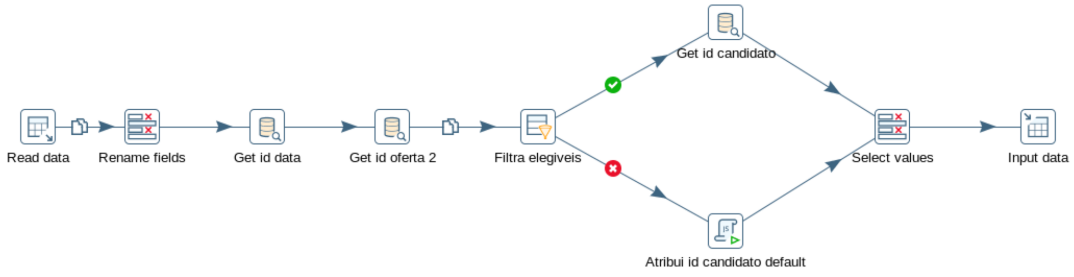

Figura 4.7: Transformação para carregamento de factos no âmbito PASEP

Para o carregamento dos factos do âmbito PASEP foi criada uma transformação que lê os ids da dimensão tempo e da dimensão oferta correspondentes. Os dados da tabela de factos são divididos, caso tenham o aluno descriminado ou não (no caso de candidaturas indeferidas não existe dados dos candidatos), Se tiverem um aluno associado, é feito um *lookup* na tabela de dimensões do candidato correspondente, caso contrário é atribuído um *id default* que representa "sem aluno atribuído". Por fim, estas duas *streams* de dados são agregadas e é feito o *lookup* para obter o id da oferta. Terminada a pesquisa dos índices das dimensões, os dados são inseridos na respetiva tabela de factos, prontos para a criação do cubo OLAP.

4.2. Cubos OLAP

Para a construção do cubo OLAP é usada uma ferramenta, com o nome de Schema Workbench, para definir o esquema a usar no servidor de BI, representado por um ficheiro XML, usando um ambiente gráfico para facilitar a sua construção.

O esquema contém as várias definições de acordo com o modelo da *Data Mart*. Nele devem estar representadas as dimensões e métricas, assim como estas métricas devem ser calculadas, necessárias para as respetivas análises que serão efetuadas nos *dashboards*. Estas análises devem serem construidas através de *queries* MDX.

Este ferramenta permite também a criação de cubos virtuais, que permitem análises sobre duas tabelas de factos distintas com a mesma granularidade. No âmbito deste estágio foram usados cubos virtuais para realizar o cálculo da "Taxa de alojamento"na módulo de Alojamento para cruzar os dados existentes na tabela de facto de alojados e lugares disponíveis das residências, que se encontram em duas tabelas de factos diferentes. A este tipo de análise da-se o nome de *drill-across*.

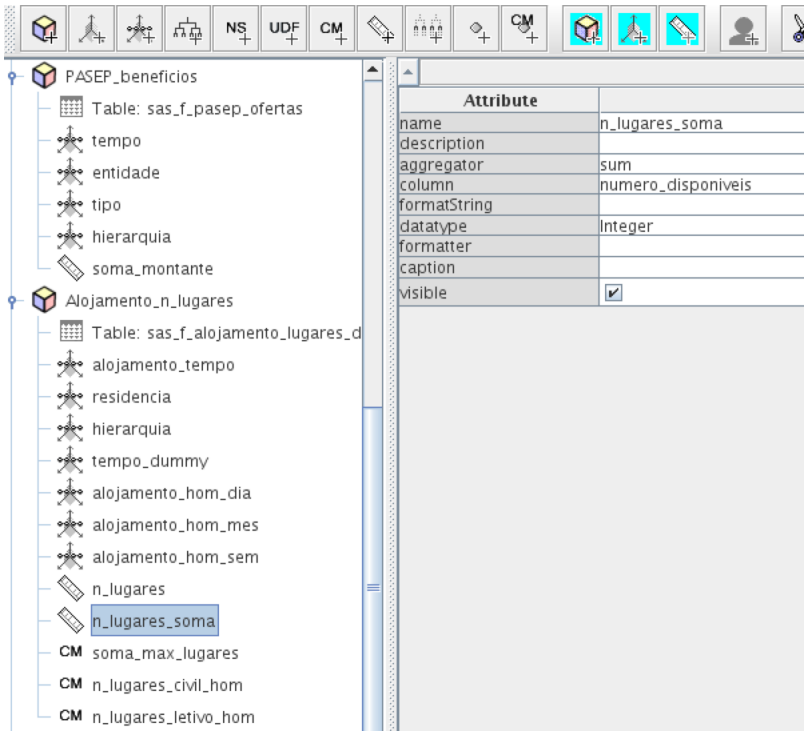

Figura 4.8: Cubos OLAP no âmbito de Alojamento e PASEP

É de salientar também que o Pentaho BI Server usa a tecnologia Mondrian para realizar as análises efetuadas pelos *dashboards*. As *queries* MDX são traduzidas pelo Mondrian para SQL e realizados sobre os dados da *Data Mart*. A principal vantagem desta ferramenta está no bom controlo de memória cache, que permite acelerar consideravelmente a resposta às análises realizadas mais frequentemente.

4.3. Dashboards

A implementação dos dashboards é feita usando o Pentaho BI Server, mais concretamente um plugin para o mesmo com o nome de CDE. Esta ferramenta encontra-se dividida em três painéis principais:

- Layout
- Componentes
- Datasources

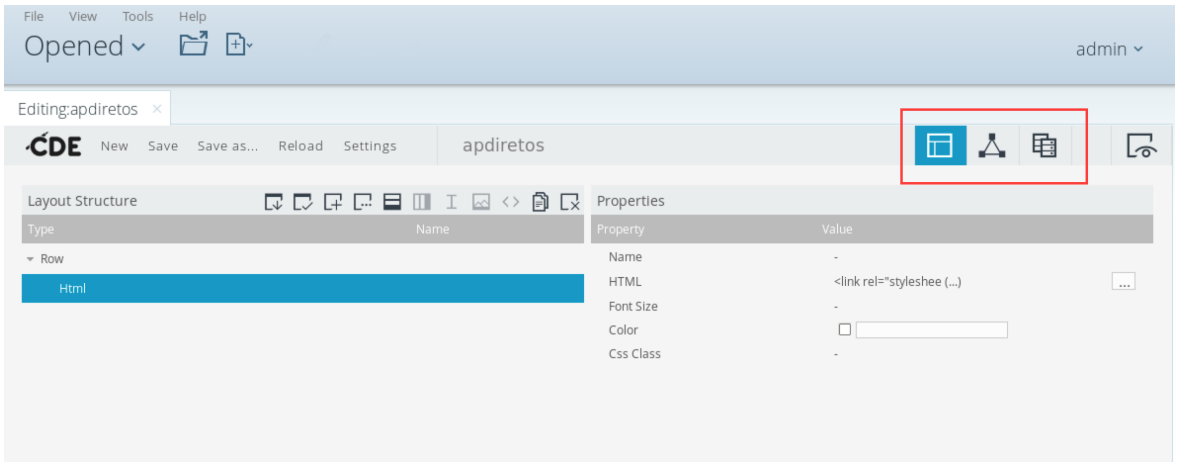

Figura 4.9: Painel de layout do CDE

O painel de *layout*, como o o nome indica, diz respeito ao *layout* do *dashboard*: aqui é definido o HTML , as bibliotecas de *javascript* a serem importadas e também os estilos CSS a ser utilizados no *dashboard*. No entanto, por escolha da equipa, a introdução destas bibliotecas não é feita da maneira tradicional. Estas bibliotecas são importadas em um ficheiro que por sua vez é importado quando se define o html. Isto permite uniformizar todos os *dashboards* caso haja a necessidade de alterações futuras, apenas é preciso alterar esse ficheiro evitando assim fazer as alterações em cada *dashboard*. As principais bibliotecas usadas são:

- Chosen - plugin de jQuery que torna as caixas de seleção um pouco mais intuitivas. Adiciona também caixas de pesquisa para caixas com um numero elevado de valores.
- bootstrap - *Framework* de HTML, CSS e *javascript* para desenvolvimento de páginas web adaptáveis a várias plataformas.
- fontawesome - Biblioteca de ícones
- highcharts - *Framework* de *javascript* para a construção de gráficos em ambientes web

O painel de componentes é usado para criar toda a lógica usada no *dashboard*. Existem vários tipo de componentes que foram mais usados no desenvolvimento, são eles:

- Parâmetros
- Caixas de seleção e múltipla seleção.
- Funções de javascript
- Gráficos
- Tabelas

Os parâmetros são a base principal para a lógica interna do dashboard. Regra geral as caixas de seleção tem um parâmetro associado onde é guardado o valor selecionado nessa caixa (ou valores no caso de caixas de múltipla seleção). Este valor pode depois ser acedido posteriormente para preencher de forma correta outros componentes que dependem desse valor ou mostrar/esconder componentes. Por exemplo, cada indicador contem filtros diferentes e/ou um período temporal diferente, então, dependendo do valor do parâmetro que identifica o indicador selecionado as caixas de seleção de filtros e/ou as caixa de seleção temporal são renderizadas de forma correta.

Para alem do referido em cima, os parâmetros têm outra função muito importante, estes podem ser usados como listeners. Isto permite que quando o valor de um parâmetro é alterado, todos os componentes que dependem dele sejam novamente renderizados. Isto permite ao dashbaord ter uma grande interatividade com o utilizador sem ser necessário fazer refresh á pagina do dashboard. Por exemplo quando ao visualizar um indicador, se selecionar filtros usando as respetivas caixas de seleção os gráficos devem apresentar as análises corretas, assim, com o listener definido para esses parâmetros, o gráfico é novamente renderizado apresentado os valores corretos.

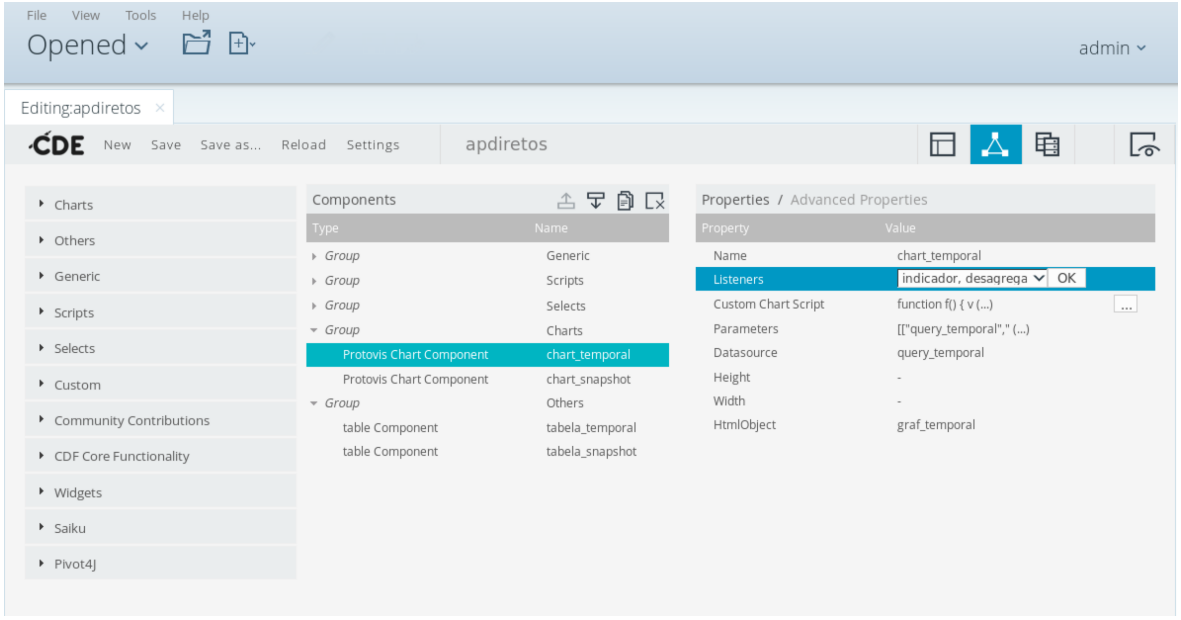

Figura 4.10: Painel de componentes do CDE

As Caixas de seleção permitem definir o valor destes parâmetros através das escolhas definidas pelo utilizador. É apenas necessário definir quais os parâmetros que estão associados com cada caixa permitindo assim representar essas escolhas como estados e renderizar os componentes necessários para o dashboard apresentar os dados corretamente.

As funçoes de javascript, embora possam ser usadas para adicionar funcionalidades extra ao dashboard, permitindo assim uma grande capacidade de personalização do mesmo, são usadas principalmente para criar as queries MDX de forma dinâmica,

dependendo das escolha inseridas pelo utilizador, em forma de string para serem depois usadas pelos gráficos ou outros componentes que dependam do valores existentes na datamart. O resultado final destas funções é guardado também em um parâmetro que é vai ser posteriormente usado pelos componentes que dele dependam.

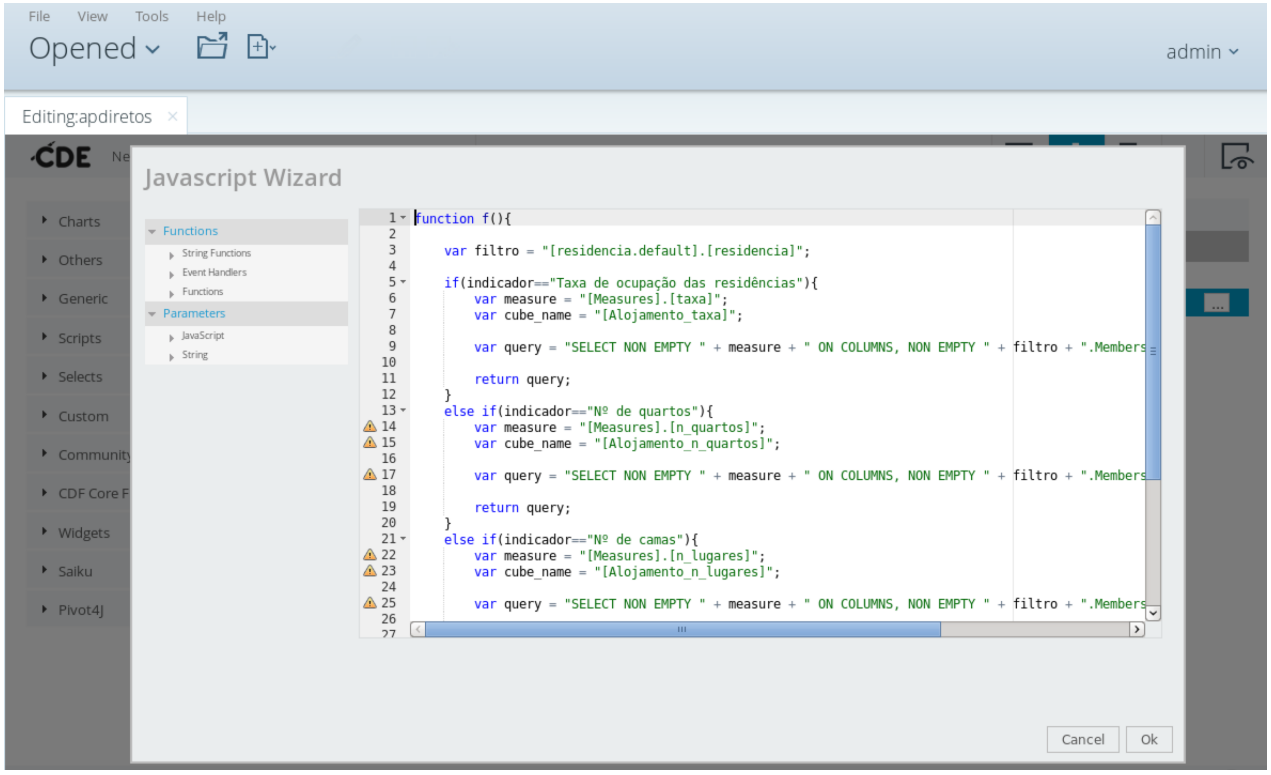

Figura 4.11: Javascript para criar queries MDX dinamicamente

Os gráficos e as tabelas são funcionam de maneira semelhante. Em ambas é definido um datasource que devolve o resultado da querie MDX previamente construida. e escreve os resultados em um gráfico/tabela para serem apresentados ao utilizador.

Todos estes componentes, com a exceção dos das funções de javascript e dos parâmetros, estão associados a um elemento do HTML definido no painel de layout. Isto indica ao CDE em que zona do HTML estes componentes devem ser renderizados.

Por fim, o painel de datasources permite definir como o servidor de BI deve aceder à informação existente na datamart. Para tal é necessário configurar a ligação JNDI para a base de dados da datamart e escolher o esquema Mondrian a ser utilizado. É necessário também definir qual a querie MDX a ser realizada pela datasource, no entanto esta pode ser definida por parâmetro proveniente do componente do gráfico, como referido em cima.

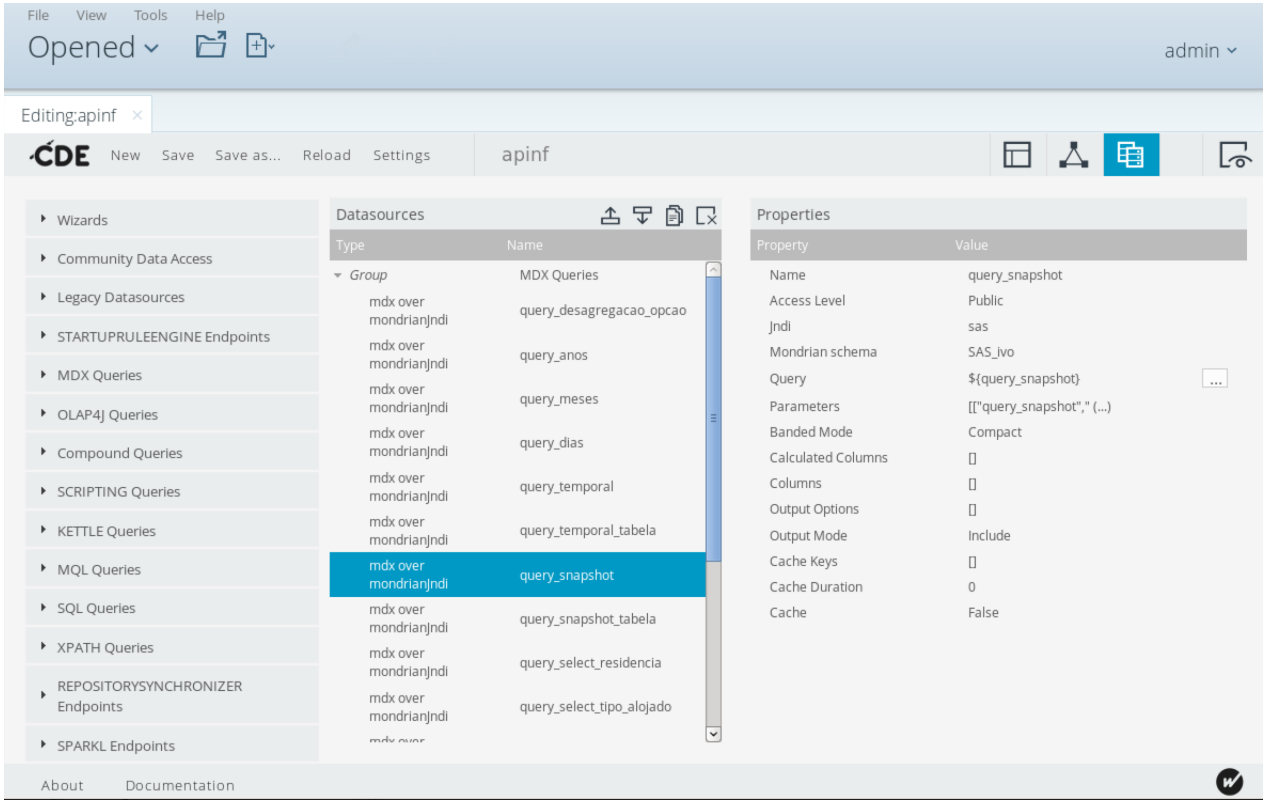

Figura 4.12: Painel de datasources do CDE

4.4. Aplicação final

São aqui apresentados alguns *screenshots* do módulo desenvolvido que se encontra atualmente em validação.

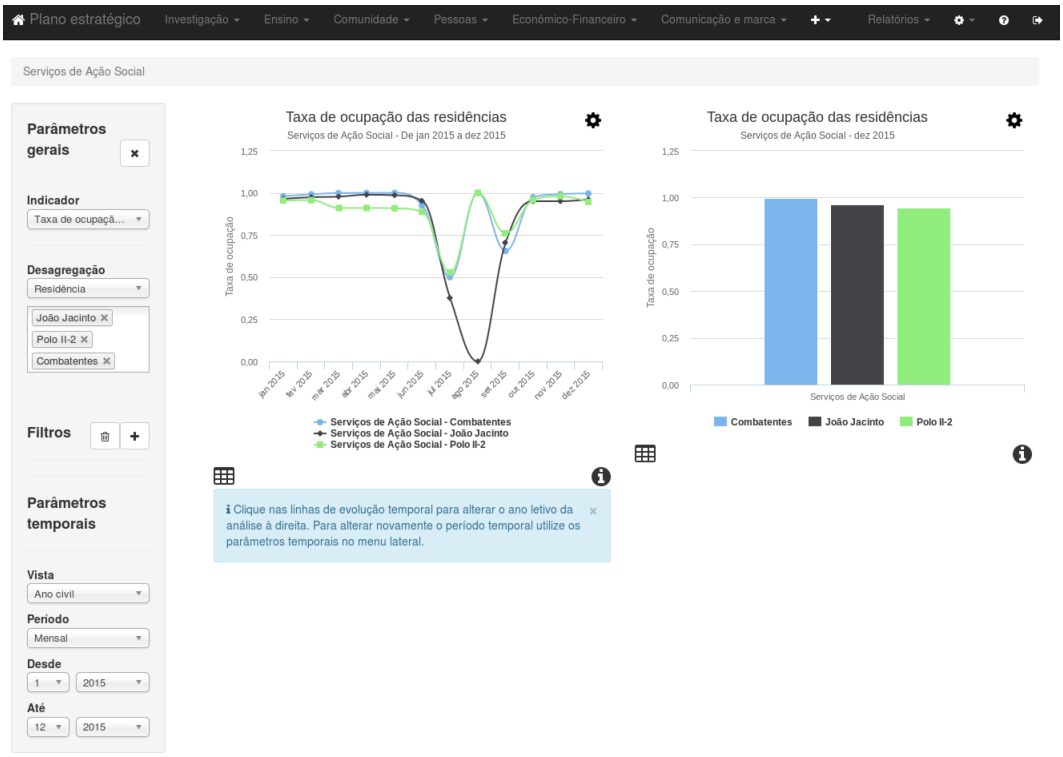

Figura 4.13: Dashboard de Alojamento com análise temporal

Na figura 4.4, pode-se observar uma das análise possíveis no indicador taxa de ocupação das residências no âmbito de Alojamento. Aqui está representado, a linha temporal do ano civil de 2015. De notar que o painel lateral permite fazer as funções de desagregação, filtragem e definir a linha temporal apresentada nos gráficos.

Alterando a vista para homólogo semestral, como apresentado na figura 4.4, pode-se ver a mesma análise anterior mas representada com a nova vista temporal (1º e 2º semestre para cada residência).

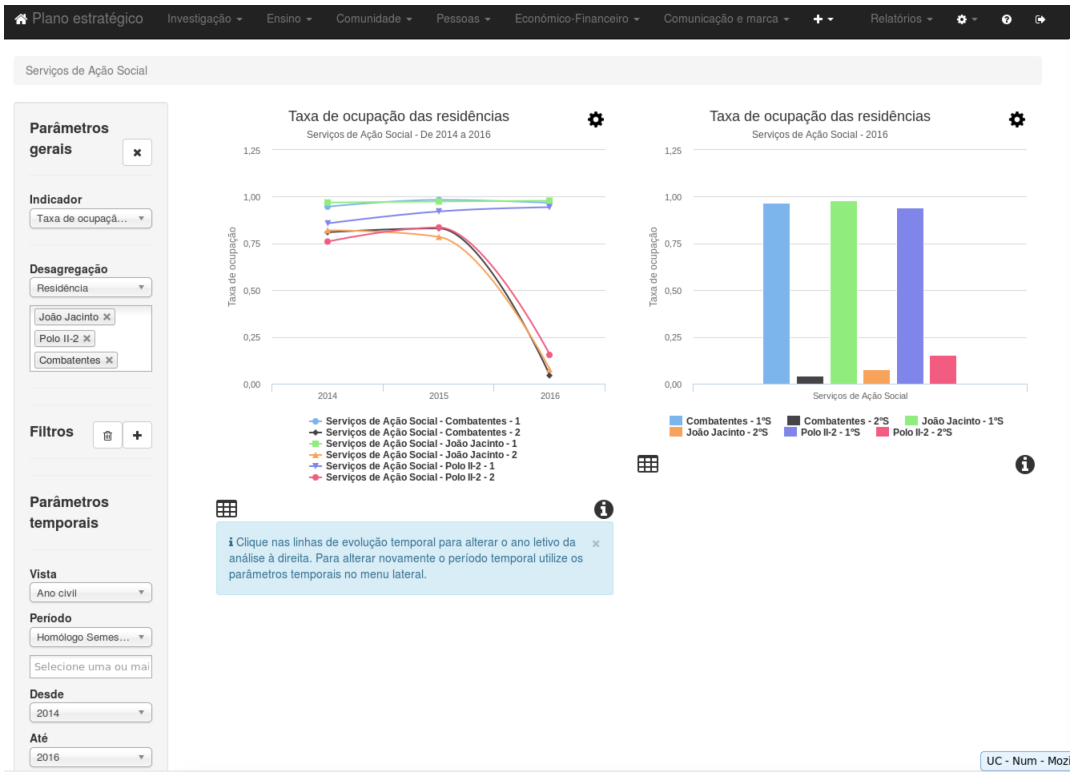

Figura 4.14: *Dashboard de Alojamento com análise homóloga semestral*

Capítulo 5

Testes e validação

Neste capítulo é abordado a metodologia e realização de teste funcionais sobre o módulo de SASUC e também a respetiva validação com o grupo operacional.

5.1. Metodologia de Testes

Neste projeto, a equipa de desenvolvimento deparou-se com a dificuldade de criar um processo de testes automatizados, devido a uma grande diferença entre os vários indicadores existentes e até mesmo pequenas diferenças entre os *dashboards* de cada indicador. Assim, escolheu-se uma abordagem de *blackbox testing*, que consiste na criação de vários conjuntos de *inputs* válidos e na observação do resultado, comparando com o resultado esperado. Este tipo de testes têm uma grande vantagem, na medida em que podem ser realizados por utilizadores sem conhecimento interno do sistema. Estes testes irão ser desenvolvidos durante o segunde semestre, assim que todo o processo de ETL e respetivo *dashboard* estejam desenvolvidos.

Os testes a desenvolver no segundo semestre vão ser especificados de acordo com a tabela 4.1.

Código	Descrição	Ação	Resultado Esperado	Resultado Obtido
Código único para identificação do teste	Descrição do teste (e.g. Exportar gráfico)	Ação necessária para realizar o teste (e.g. carregar no ícone no canto superior direito e escolher imagem PNG)	Descrição do resultado esperado (e.g. download da imagem efetuado)	Descrição do resultado obtido quando realizado o teste (e.g. download não efetuado)

Tabela 5.1: Exemplo de formulação de testes

5.2. Testes funcionais

De acordo com a secção anterior foram realizados testes sobre os vários *dashboards* desenvolvidos, repetindo caso tenham sido efetuadas alterações nos mesmos. Na tabela seguinte são apresentados alguns exemplos dos teste realizados.

Código TST	Descrição caso de teste	Ação	Resultado esperado	Resultado obtido
TF_SAS_023	É possível visualizar/analisar os indicadores PASEP?	Selecionar um dos indicadores que se encontram no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado
TF_SAS_036	É possível visualizar/analisar os indicadores de Alojamento?	Selecionar um dos indicadores que se encontram no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado
TF_SAS_043	É possível visualizar/analisar os indicadores de Apoios Diretos?	Selecionar um dos indicadores que se encontram no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado
TF_SAS_058	É possível visualizar/analisar os indicadores de Apoio á Infância?	Selecionar um dos indicadores que se encontram no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado

Código TST	Descrição caso de teste	Ação	Resultado esperado	Resultado obtido
TF_SAS_024	É possível visualizar/analisar o indicador nº de ofertas?	Selecionar o indicador nº de ofertas no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado
TF_SAS_038	É possível visualizar/analisar o indicador nº de residências?	Selecionar o indicador nº de residências no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado
TF_SAS_061	É possível visualizar/analisar o indicador montantes anuais de apoios diretos?	Selecionar o indicador montantes anuais de apoios diretos no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado
TF_SAS_038	É possível visualizar/analisar o indicador nº de crianças inscritas?	Selecionar o indicador nº de crianças inscritas no menu lateral do dashboard.	Visualizar a informação respetiva ao indicador selecionado, nos gráficos de análise temporal e snapshot.	Igual ao esperado

Código TST	Descrição caso de teste	Ação	Resultado esperado	Resultado obtido	
TF_SAS_035	A opção do menu dos gráficos "Esconder/Mostrar filtros"adiciona ao gráfico os filtros selecionados para cada indicador?	Filtrar a informação e clicar na opção "Esconder/Mostrar filtros"para cada indicador.	Filtros selecionados são adicionados à área de desenho do gráfico.	Igual ao esperado	Sim
TF_SAS_033	Se o intervalo de tempo escolhido for inválido nos vários indicadores, o utilizador é notificado?	Selecionar um intervalo de tempo inválido para cada indicador.	Visualizar uma mensagem de erro em vez do gráfico temporal	Igual ao esperado	
TF_SAS_034	Os botões de informação mostram os dados de ajuda respetivo ao indicador selecionado?	Clicar no botão de informação para cada indicador.	Visualizar uma mensagem com informações sobre o indicador selecionado.	Igual ao esperado	

Tabela 5.2: Exemplos de testes funcionais efetuados nos *dashboards*

O documento de testes pode ser consultado nos anexos digitais.

5.3. Validação

O processo de validação é um passo de grande importância no desenvolvimento deste módulo. É através da validação que se pode comprovar que todos os requisitos levantados foram cumpridos e que corretamente e que no final as análises efetuadas através deste módulo apresentam os dados reais que seriam obtidos caso estes indicadores tivessem sido calculados manualmente. Uma vez terminado o processo de validação o módulo desenvolvido durante este estágio está em condições de ser integrado com o produto final.
Assim que terminados os testes funcionais foi agendadas reuniões no dia 22 de Julho e

9 de Agosto, com o grupo operacional dos SASUC, destacando a Dra. Regina Bento e a Dra. Maria João Rodrigues, responsáveis pela validação deste módulo. Apesar de o processo de validação ainda estar a decorrer à data de escrita deste documento, estas reuniões contribuíram bastante para a correção dos *dashboards* desenvolvidos. De momento, todas as alterações pedidas pelo grupo já foram realizadas e os problemas encontrados durante estas reuniões estão já resolvidos.

Capítulo 6

Metodologia e Planeamento

6.1. Metodologia

A metodologia adotada para o desenvolvimento deste sistema de BI tem como base a metodologia proposta por Ralph Kimball[16], que tem sido aplicada a um vasto número de projetos, com grande sucesso. As principais etapas desta metodologia, para o módulo de SASUC são: planeamento do projeto, definição de requisitos, desenho do modelo de dados, desenvolvimento do plano de ETL e desenvolvimento de *dashboards*. As três primeiras etapas foram realizadas durante este semestre, enquanto que as restantes duas estão planeadas para o segundo semestre.

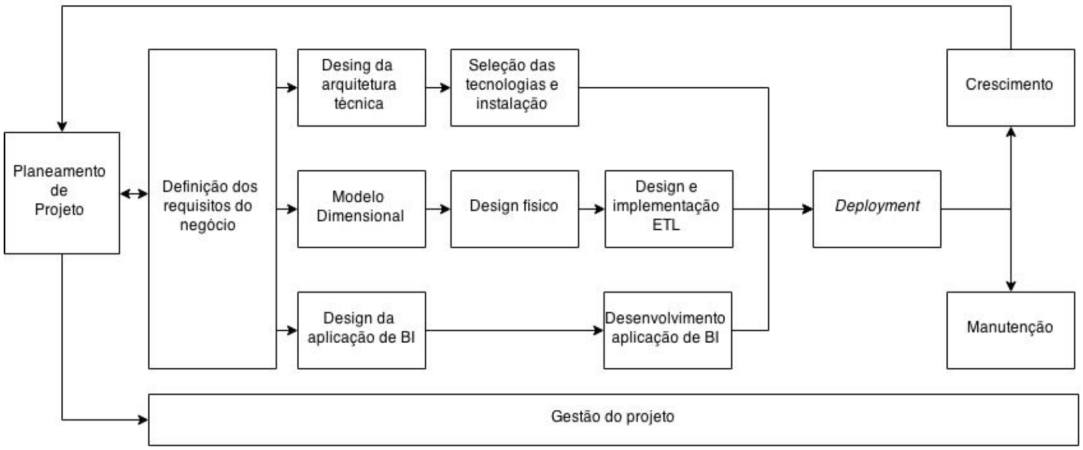

Figura 6.1: Metodologia de desenvolvimento de um sistema de BI, segundo Ralph Kimball

A etapa de planeamento de projeto foi realizada no início do semestre com toda a equipa de desenvolvimento e com o orientador, onde se definiram as metas a atingir durante o percurso do estágio. De seguida, houve um processo de levantamento de requisitos, já referido no capítulo 2, para o módulo de SASUC pois os requisitos gerais já se encontram definidos e, por fim, foi realizado o desenho do modelo de dados para a *data mart*, que se encontra especificado na secção 3.4 do capítulo 3.

Por fim, como este é um projeto em equipa e com vários módulos diferentes em desenvolvimento, é necessário adotar medidas para garantir o bom funcionamento da equipa. Cada membro da equipa tem acesso a uma máquina virtual com todas as ferramentas necessárias ao projeto. Como o desenvolvimento é feito em paralelo nestas máquinas, surgiu a necessidade de usar ferramentas de versionamento de código, para agilizar o processo de partilha de código, comum a todos os módulos, e garantir que, em caso de acidentes, todo o trabalho feito é recuperável. Para este propósito escolheu-se o *git*[17], que é uma ferramenta de versionamento de código orientada para colaboração entre grande número de programadores. Esta ferramenta também é usada para versionar os documentos existentes e partilhar toda a documentação entre as várias máquinas. Como ferramenta de comunicação entre a equipa é usado o *skype* e/ou email. Para garantir que, tanto o orientador como a gestora de equipa estão a par do progresso dos vários elementos, são realizadas reuniões semanais com toda a equipa de desenvolvimento. Por fim, para auxiliar a gestão do projeto é usada a ferramenta web *taiga*[18] onde são inseridas as tarefas em desenvolvimento que se baseia no *kaban*[19]

6.2. Planeamento

Para este estágio, tendo em conta a metodologia proposta, planearam-se as seguintes tarefas para o primeiro semestre: pesquisa e elaboração de estado da arte sobre as tecnologias possíveis para um sistema de BI, pesquisa sobre a metodologia usada no desenvolvimento de testes, especificação de indicadores para o módulo de SASUC, criação de protótipos para os respetivos indicadores e elaboração de um modelo de dados para a *data mart*. Deste planeamento realizado para o primeiro semestre resultou o diagrama de *gantt*, representado na figura 4.2

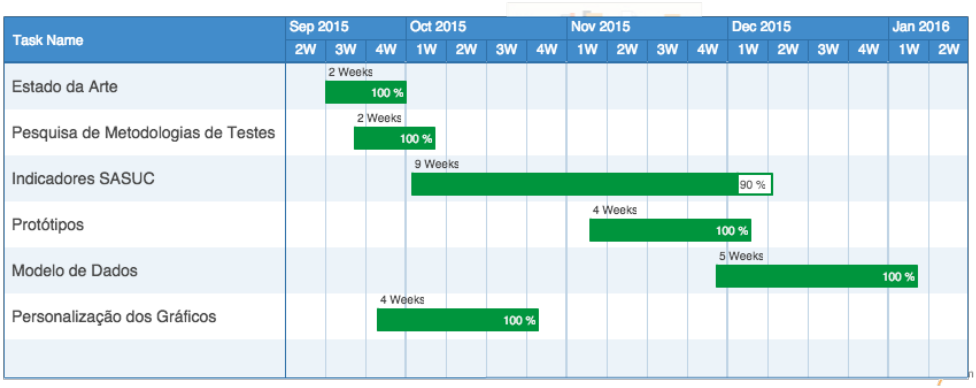

Figura 6.2: Diagrama de Gantt para o primeiro semestre

Relativamente ao segundo semestre, o planeamento assenta em três tarefas principais: desenvolvimento do processo de ETL e cubos *olap*, desenvolvimento dos *dashboards* e, por fim, os respetivos testes e validações, com a sua disponibilização no sistema (em caso afirmativo). Na figura 4.3 está representado o diagrama de *gantt*, com o plano de trabalhos para segundo semestre. É importante salientar que este é um processo interativo, ou seja, o plano de ETL e os respetivos *dashboards* são feitos sucessivamente para cada âmbito.

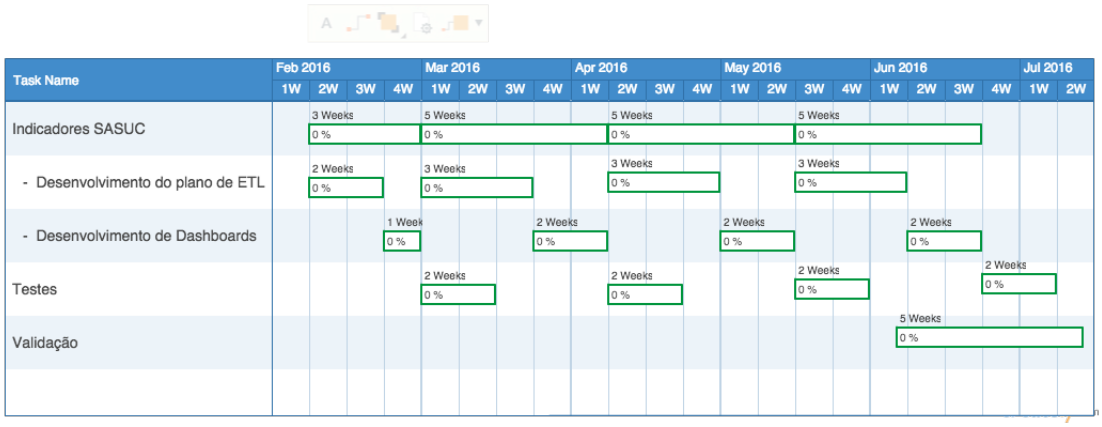

Figura 6.3: Diagrama de Gantt para o segundo semestre

Durante o segundo semestre ocorreram atrasos no desenvolvimento dos vários *dashboards*, que por sua vez atrasaram o processo de validação. Este processo começou no final de Julho e foi difícil agendar reuniões durante o mês de Agosto, encontrando-se ainda a decorrer. Está previsto recomeçar agora em Setembro. Na figura abaixo pode-se ver o diagrama real 2º semestre.

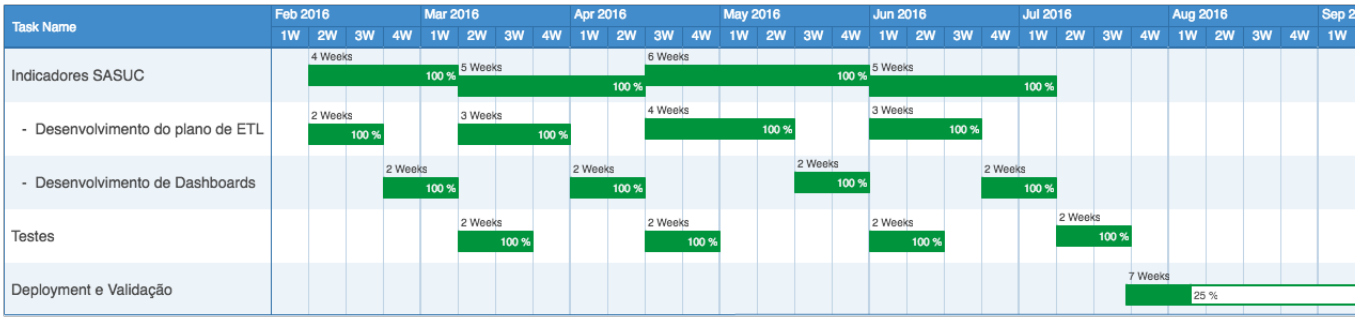

Figura 6.4: Diagrama de Gantt real para o segundo semestre

Capítulo 7

Conclusão

Neste documento foram apresentadas as várias etapas do desenvolvimento do módulo de SASUC para o projeto UC-Num.

Começando pelo levantamento de requisitos, foram apresentados os requisitos que já se encontravam definidos no início deste estágio, para a aplicação UC-Num, assim como os requisitos levantados para os indicadores a serem calculados para este módulo.

Foi feita uma análise da arquitetura já definida para aplicação, assim como das tecnologias alternativas existentes atualmente, que possam ser uma mais valia para a aplicação UC-Num.

Foram detalhados os vários passos de implementação deste módulo, a começar pela elaboração do modelo de dados, o processo de ETL, a criação do cubo OLAP, terminado no desenvolvimento dos dashboards.

Foi também descrito o processo usado para a realização de testes funcionais, verificando assim que, quer os requisitos já existentes, quer os requisitos levantados para este módulo são cumpridos.

Foi explicado também o processo de validação que, apesar de ainda não estar concluído à data de escrita deste documento, não deixa de ser uma etapa importante deste estágio.

Para terminar, este estágio foi uma mais valia para mim pois não só permitiu aumentar os meus conhecimentos sobre uma área do meu agrado - e que é cada vez mais solicitada - como também permitiu a aquisição das competências sociais necessárias ao mercado de trabalho dos dias de hoje, através da integração numa equipa de desenvolvimento e da comunicação com os diferentes stakeholders envolvidos na aplicação UC-Num.

Anexos

A - Análise comparativa de bases de dados

Na tabela seguinte é apresentada uma análise comparativa entre os motores de bases de dados relacionais e *opensource*, com as características mais relevantes para uma *data wharehouse*.

		PostgresSQL	MySQL
Licença		Open Source	Open Source
ACID		Sim	Sim
Alta escalabilidade		Sim	Não
Alta disponibilidade		Sim	Não
Processamento paralelo		Sim	Não
Transações		Sim	Sim
Triggers		Sim	Sim
Suporte para Java		Sim	Sim
Vistas materializadas		Sim	Não
Tamanho total		Sem limite	Sem limite
Plataformas	Windows	Sim	Sim
	Mac OS	Sim	Sim
	Linux	Sim	Sim
	Unix	Sim	Sim
	Android	Sim	Sim
Índices	B-ree	Sim	Sim
	R-Tree	Sim	Sim
	Hash	Sim	Sim
	Partial	Sim	Não
	Bitmap	Sim	Não
	GIN	Sim	Não

Tabela 7.1: Comparação entre MySQL e PostgreSQL[1, 2]

B - Análise comparativa de ferramentas de ETL

Nesta secção, é apresentada uma tabela com as principais características das ferramentas de ETL *opensource* escolhidas para análise.
De notar que *Pentaho Data Integration* e o *Jaspersoft ETL* são muito semelhantes, embora separadas por uma caraterística muito importante: o processamento paralelo está desativado na versão gratuita do *Jaspersoft ETL*, o que por si só é suficiente para escolher o *Pentaho Data Integration* quando não existe dinheiro para aquisição de licenças. No entanto, é preciso salientar que a performance do *Jaspersoft ETL* é superior e que seria uma boa escolha no caso de existir a possibilidade de pagar por uma licença.

		Pentaho Data Integration	Jaspersoft ETL
Licença		Open Source	Open Source
Processamento paralelo		Sim (dividindo o flow de dados)	Não (na versão gratuita)
Usabilidade		Simples, com interface gráfica	Simples, com interface gráfica
Suporte		Apenas Comunitário (com suporte profissional pago)	Apenas Comunitário (com suporte profissional pago)
Scripting	Java	Sim	Sim
	Javascript	Sim	Sim
	SQL	Sim	Sim
Plataformas	Windows	Sim	Sim
	Mac OS	Sim	Sim
	Linux	Sim	Sim

Tabela 7.2: Comparação entre Pentaho Data Integration e Jaspersoft ETL[3, 4]

C - Análise comparativa de servidores de BI

Nesta secção é feita uma análise comparativa entre duas ferramentas *opensource*, para um servidor de BI.

O *Pentaho BI Server* apresenta uma solução mais abrangente, principalmente devido à possibilidade de instalação de *plugins* quando funcionalidades são inexistentes, já o *JasperReports Server* foca-se principalmente, como o nome indica, na elaboração de relatórios. O *Pentaho BI Server* tem outra grande vantagem, que é a integração de uma ferramenta, conhecida por *Mondrian* ou *pentaho schema workbench*, para a especificação de cubos OLAP, que podem ser facilmente integrados no servidor de bi.

	Pentaho BI Server	JasperReports Server
Licença	Open Source	Open Source
Criação de Dashboards	Sim (com plugins)	Não
Criação de Relatórios	Sim	Sim
Análise de cubos OLAP	Sim (com plugins)	Sim
Suporte	Apenas Comunitário (com suporte profissional pago)	Apenas Comunitário (com suporte profissional pago)

Tabela 7.3: Comparação entre Pentaho BI Server e JasperReports Server[5, 6]

D - Anexos Digitais

Os seguintes anexos são fornecidos em formato digital em conjunto com este documento.

1. Fichas_Indicadores.pdf
2. Documento de requesitos.xls

Bibliografia

[1] The PostgreSQL Global Development Group, "PostgreSQL Documentation," http://www.postgresql.org/docs/9.5/interactive/index.html, acedido em 2015-11-14.

[2] Oracle Corporation, "MySQL Documentation," http://dev.mysql.com/doc/refman/5.7/en/, acedido em 2015-12-12.

[3] "Pentaho Corporation," http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation, acedido em 2016-12-5.

[4] "TIBCO Software, Inc." http://community.jaspersoft.com/wiki/community-wiki, acedido em 2016-12-8.

[5] "Pentaho Corporation," http://wiki.pentaho.com/display/ServerDoc2x/BI+Server+2.x-3.x+Community+Documentation, acedido em 2016-12-17.

[6] "TIBCO Software, Inc." http://community.jaspersoft.com/documentation?version=29351, acedido em 2016-01-13.

[7] R. B. Grady, *Practical Software Metrics for Project Management and Process Improvement*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[8] Oracle Corporation, "MySQL," https://www.mysql.com/, acedido em 2015-11-9.

[9] The PostgreSQL Global Development Group, "PostgreSQL," http://git-scm.com/, acedido em 2015-11-9.

[10] "NoSQL databases," http://nosql-database.org/, acedido em 2016-1-5.

[11] "Pentaho Corporation," http://community.pentaho.com/projects/data-integration/, acedido em 2016-12-5.

[12] "TIBCO Software, Inc." http://community.jaspersoft.com/project/jaspersoft-etl, acedido em 2016-12-8.

[13] "Pentaho Corporation," http://community.pentaho.com/, acedido em 2016-12-17.

[14] "TIBCO Software, Inc." http://community.jaspersoft.com/project/jasperreports-server, acedido em 2016-01-13.

[15] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, 2nd ed. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[16] R. Kimball, L. Reeves, W. Thornthwaite, M. Ross, and W. Thornwaite, *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses with CD Rom*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1998.

[17] Software Freedom Conservancy, "Git," http://git-scm.com/, acedido em 2015-12-13.

[18] Taiga Agile, "Taiga," https://taiga.io/, acedido em 2015-12-23.

[19] Atlassian, "A Brief Introduction to Kanban," https://www.atlassian.com/agile/kanban/, acedido em 2015-12-23.