Mestrado em Engenharia Informática
Estágio – Sistemas de Informação
Relatório Final

Projeto UC-Num

Desenvolvimento de uma Data Warehouse para a Universidade de Coimbra

Estágio B

João Ricardo Rodrigues Correia
jrcorr@student.dei.uc.pt

Orientador:
Prof. Doutor Bruno Miguel Brás Cabral
Data: 7 de Setembro de 2015

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

Agradecimentos

Após grandes dificuldades, alegrias e tristezas este é o momento em que posso transmitir o meu profundo agradecimento a todas as pessoas que estiveram sempre ao meu lado e que não me deixaram desistir quando esse era o caminho mais fácil.

À minha namorada, família, amigos de longa data e aos que fiz durante estes 5 anos, aos meus colegas de estágio, ao meu orientador, e também professores de todos os níveis de ensino, que me formaram academicamente e pessoalmente. A todos, o meu muito obrigado.

Com muita saudade, agradeço à Fernanda, ao Nuno e ao Júnior por todo o amor e amizade.

Resumo

Para as organizações de grande dimensão, como a Universidade de Coimbra, os Sistemas de Suporte à Decisão são de uma enorme importância para a gestão estratégica e até operacional.

A análise de indicadores de gestão é uma importante tarefa em qualquer organização para as suas tomadas de decisão. Na UC essa tarefa está a ser dificultada por problemas de cálculo dos indicadores e pela demora na sua recolha. Estes problemas podem atrasar decisões importantes ou induzir em erro os *stakeholders* da UC.

O objetivo deste estágio é desenvolver uma solução de *Business Intelligence* para produção de indicadores na área dos Recursos Humanos da UC. Para alcançar este objetivo, integrei uma equipa de desenvolvimento de sistemas para produção automática, visualização e monitorização de indicadores, nas mais variadas áreas da UC.

O processo de desenvolvimento passa pela construção de uma *Data Mart*, que é um subconjunto de uma *Data Warehouse,* normalmente associado a uma temática específica. Com recurso ao modelo multidimensional é possível efetuar análises OLAP, devido à velocidade e eficácia proporcionada por esse modelo.

As atividades desenvolvidas no âmbito deste estágio incluem a identificação de novos indicadores, reuniões com utilizadores e responsáveis, identificação de requisitos, desenho dos modelos de dados, desenvolvimento de *software* e planos ETL, implementação, verificação e validação da solução, bem como escrita de documentação.

Palavras-Chave

Business Intelligence, Indicadores de Performance (KPIs), *Data Warehouse, Dashboards, Online Analytical Processing*, Recursos Humanos.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

BD	Base de Dados
CDE	*Community Dashboards Editor*
CDF	*Community Dashboards Framework*
CSS	*Cascading Style Sheets*
CSV	*Comma Separated Values*
DW	*Data Warehouse*
DEI	Departamento de Engenharia Informática
ERP	*Enterprise Resource Planning*
ETI	Equivalente a Tempo Integral
ETL	*Extract, Transform and Load*
FCTUC	Faculdade de Ciências e Tecnologia da Universidade de Coimbra
GB	*Gygabyte*
GSIIC	Gestão de Sistemas e Infraestruturas de Informação e Comunicação
HTML	*HyperText Markup Language*
HTTPS	*HyperText Transfer Protocol Secure*
JPEG	*Joint Photographic Experts Group*
KB	*Kilobyte*
KPIs	*Key Performance Indicators*
MDX	*Multidimensional Expressions*
OLAP	*Online Analytical Processing*
PDF	*Portable Document Format*
PNG	*Portable Network Graphics*
RH	Recursos Humanos
SIADAP	Sistema Integrado de Avaliação de Desempenho da Administração Pública
SQL	*Structured Query Language*
TB	*Terabyte*
XML	*Extensible Markup Language*
UC	Universidade de Coimbra

Histórico de Versões

Versão	Data	Descrição
1.0	27/01/2015	Relatório Intermédio.
1.1	05/01/2015	Reestruturação do relatório: criação do "Histórico de Versões" e do capítulo "Análise de Riscos". Foi também criado o capítulo "Estado da Arte" que passou a ser o capítulo 2, retirando essa informação do capítulo "Arquitetura", que com a introdução do capítulo "Análise de Riscos" passou a ser o capítulo 5.
1.2	09/02/2015	Atualização dos requisitos do projeto após reunião de equipa.
1.3	25/02/2015	Identificação e análise dos riscos existentes, bem como as estratégias de mitigação.
1.4	16/03/2015	Atualização dos requisitos funcionais, com a adição de novos indicadores especificados, já em fase de desenvolvimento.
1.5	30/03/2015	Alteração da ordem de apresentação dos modelos de dados, antecipando o modelo de dados da Data Mart.
1.6	25/04/2015	Criação e escrita de parte do capítulo "Desenvolvimento".
1.7	08/08/2015	Conclusão da escrita do capítulo "Desenvolvimento", alteração do capítulo "Testes" para "Testes e validação" e escrita sobre a 2ª fase de testes e do processo de validação.
1.8	10/08/2015	Revisão do relatório pelo Professor Doutor Bruno Cabral.
2.0	12/08/2015	Alterações consoante o feedback do Professor Doutor Bruno Cabral. O capítulo "Análise de Riscos" passou a ser um subcapítulo do capítulo 8 "Planeamento".
3.0	28/08/2015	Versão final.

Capítulo 1
Introdução

1.1. Enquadramento

Na atual conjuntura económica é cada vez mais importante uma boa gestão dentro das organizações, sejam elas públicas ou privadas. Essa gestão passa por tomar as melhores decisões de acordo com as necessidades e objetivos da organização, sendo extremamente importante conseguir reunir a maior quantidade de informação, maximizando o poder de decisão dos seus *stakeholders*.

É devido a essa necessidade que surge o projeto UC-Num, criação de uma *Data Warehouse* para a Universidade de Coimbra, que é parte integrante de um outro projeto intitulado SAMA (Sistema de Apoio à Modernização Administrativa), descrito na Figura 1, que assim pretende modernizar e melhorar o desempenho das infraestruturas e serviços das Tecnologias de Informação e Comunicação em diferentes áreas da UC.

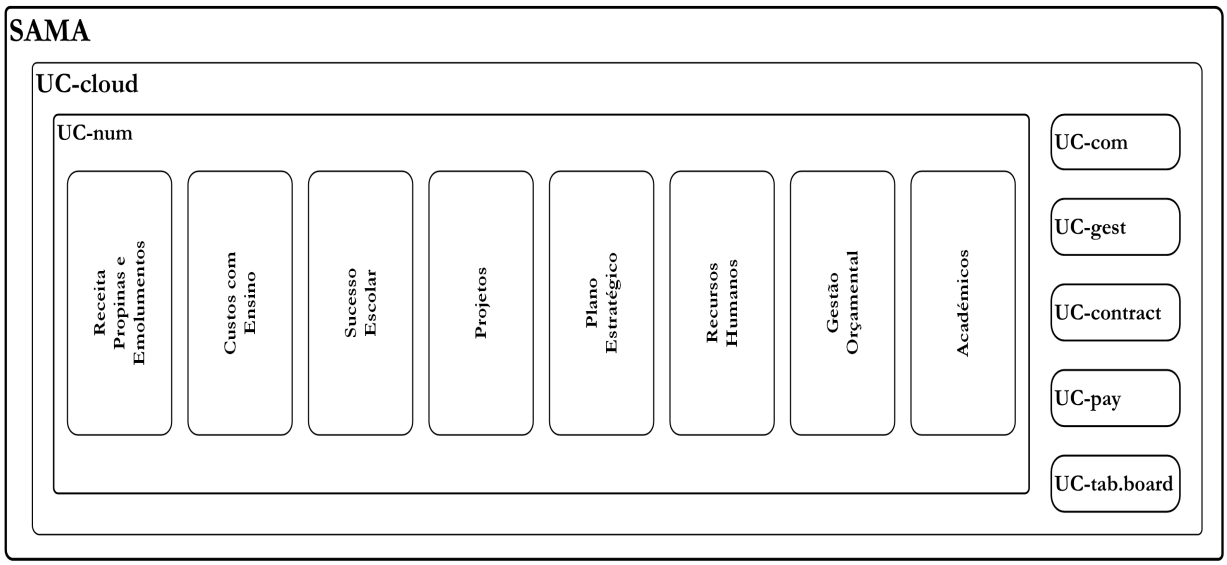

Figura 1: Integração do projeto UC-Num no projeto SAMA.

A equipa de desenvolvimento é atualmente composta por sete elementos, os Mestres, Hugo Costa, Beatriz Fragoso e Inês Domingues, e ainda quatro estagiários nos quais eu me incluo. A equipa é orientada pelo Professor Doutor Bruno Cabral e o Engenheiro Pedro Pinto, atual gestor do projeto NÓNIO.

O projeto enquadra-se na categoria de *Business Intelligence*. O BI é definido como um conjunto de técnicas e ferramentas de recolha e transformação de dados, com o objetivo de usar essa informação para o benefício do negócio. Esse benefício é fornecido através de análises de fácil interpretação, a um grande volume de dados, disponíveis em qualquer momento. Estas análises podem identificar novas oportunidades de negócio, obter vantagens competitivas e uma maior estabilidade, devido a um melhor conhecimento da organização.

Após a recolha e transformação dos dados, estes são alojados numa *Data Warehouse*, que tem como principal objetivo servir de suporte a análises OLAP, facilitando o processo moroso de recolha e análise de informação, proveniente das mais variadas fontes, pelos trabalhadores e gestores da organização. A partir deste tipo de análise é pretendido monitorizar os KPIs da UC e recolher informação de modo a responder às perguntas dos *stakeholders* acerca do negócio.

Como qualquer infraestrutura de tecnologias de informação, o BI é composto por pessoas, processos, manutenção e governação. A Figura 2 resume os componentes da *framework* de BI.

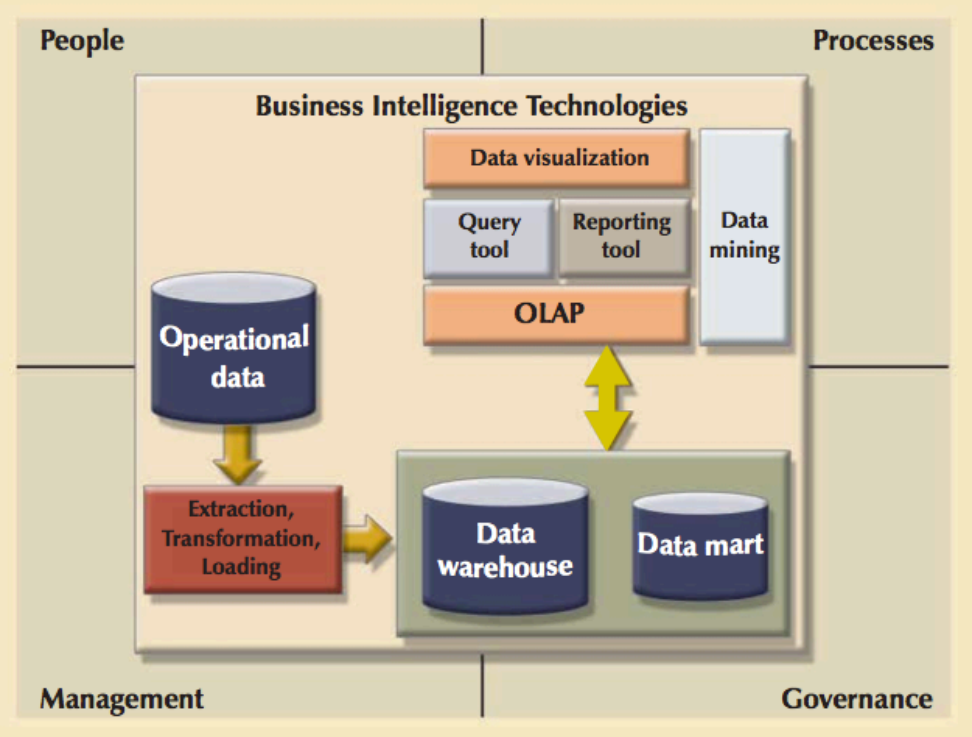

Figura 2: *Data Warehouse* na *framework* de BI[1].

O processo de cálculo dos indicadores é bastante complicado, pois é necessário que os serviços administrativos recolham a informação de diferentes fontes, como o NÓNIO, SAP ou até folhas de Excel, e que seguidamente processem essa informação e apliquem as respetivas fórmulas de cálculo. A duração deste processo pode ser de dias, semanas ou meses, consoante a complexidade do indicador e a disponibilidade dos trabalhadores.

Atualmente o projeto UC-Num está dividido em módulos para que os indicadores especificados sejam analisados consoante as necessidade identificadas. Os módulos em desenvolvimento são: Académicos, Gestão Orçamental, Recursos Humanos, Plano Estratégico, Projetos e Custos com Ensino.

1.2. Objetivos

Este estágio teve como objetivo o desenvolvimento de novos indicadores para a *Data Mart* de Recursos Humanos da Universidade de Coimbra.

Para disponibilizar análises OLAP para os Recursos Humanos é necessário recolher e tratar informação oriunda de duas diferentes fontes de dados:

- SAP RH – é um subsistema do ERP SAP da Universidade de Coimbra. O SAP RH aloja toda a informação relativa aos Recursos Humanos, como por exemplo, informações biográficas e contratuais dos trabalhadores, suspensões e faltas ao serviço.
- NÓNIO – plataforma usada para registo do serviço docente. Nesta plataforma está alojada a informação de todos os alunos e professores da UC, bem como de todas as unidades curriculares, pertencentes aos diferentes cursos e às respetivas unidades orgânicas.

Esta informação será utilizada para calcular KPIs, que permitam avaliar o desempenho da UC nas diversas vertentes previstas no atual Plano Estratégico de Ação[3].

1.3. Estrutura do Relatório

Nos seguintes capítulos deste relatório são retratadas, de forma detalhada, as várias fases do desenvolvimento do projeto de *Business Intelligence*.

No capítulo 2, é apresentado o estado da arte de possíveis tecnologias a utilizar num projeto deste género.

No capítulo 3 é descrito o processo de levantamento de requisitos funcionais e não funcionais.

De seguida, no capítulo 4 é descrita a arquitetura do sistema, desde a recolha de informação até à sua análise.

No capítulo 5 é resumido o processo de desenvolvimento.

O capítulo 6 apresenta a metodologia de testes, os testes funcionais executados à aplicação e ainda o processo de validação.

O capítulo 7 descreve o processo de planeamento do projeto e a análise de riscos inerentes a um projeto de *Business Intelligence*.

Por último, o capítulo 8 consiste numa reflexão pessoal de todo o percurso durante o estágio.

Capítulo 2
Estado da Arte

À data de início do estágio, as decisões relativas à adoção de tecnologias, já tinham sido tomadas. Com o propósito de conhecer melhor a área de BI e as suas tecnologias, foi elaborada uma pesquisa que engloba maioritamente soluções gratuitas, visto não estar definida nenhuma verba para aquisição de licenças de *software* para este projeto.

2.1. Bases de dados

Foi tomada a decisão de criar uma área temporária, com um modelo de dados relacional, que guarda a informação, proveniente das fontes de informação, para posteriormente ser tratada e carregada para a *Data Mart*. Tanto para a área temporária como para a *Data Mart* é necessário um motor de base de dados.

São várias as bases de dados de licença gratuita, sendo que algumas possuem arquiteturas diferentes, que influenciam diretamente qualidades chave como a escalabilidade e o desempenho.

Criadas na década de 1970, as bases de dados relacionais têm correspondido à maioria das necessidades. De um vasto leque de ofertas, destacam-se os principais motores *open-source:*

- *MySQL* - Com provas mais que dadas, sendo utilizado por companhias como a *Adobe*, *SAGE* e *Facebook*, intitula-se como a base de dados *open-source* mais conhecida do mundo[4] .
- *PostgreSQL* - o motor de bases de dados *open-source* mais avançado do mundo[5], é utilizado por empresas como a *CISCO* e *Microsoft* (*Skype*), bem como agências governamentais Norte-Americanas (*U.S. State Department*).

No anexo 1 encontra-se uma análise mais detalhada aos motores referidos, comparando características como os tipos de dados, índices e limites das bases de dados.

É possível optar por bases de dados distribuídas, onde com um conjunto de máquinas de baixo custo, é possível aumentar bastante a escalabilidade e o desempenho, como é o caso do *MySQL Cluster*, igualmente *open-source*. Sem um único ponto de falha, o *MySQL Cluster* deve ser sempre uma opção para um projeto de grande escala onde se pretenda um sistema relacional, fiável e capaz de corresponder a um nível de exigência elevado.

Os motores de bases de dados *NoSQL*, são cada vez mais uma opção aliciante para quem pretende alternativas aos motores relacionais. O interesse em motores *NoSQL* surge da necessidade de armazenamento de informação não estruturada e de uma escalabilidade superior à oferecida pelos motores relacionais. De acordo com a comunidade *NoSQL*[6] existem vários tipos de bases de dados, destacando-se quatro:

- Chave-valor, que armazena a informação de acordo com um par chave-valor numa ou mais tabelas.
    - Exemplos: *DynamoDB* e *MemcacheDB*.
- Orientadas a colunas, que armazenam cada coluna num ficheiro diferente, diminuindo em muito o tempo acesso a uma coluna específica.
    - Exemplos: *MonetDB* e *Hadoop*.

- Orientadas a ficheiros, permitem o armazenamento de ficheiros com uma estrutura semelhante, sendo assim possível ter informação semiestruturada.
    - Exemplos: *MongoDB* e *CouchDB*
- Baseadas em grafos, para quando a informação se encontra fortemente relacionada.
    - Exemplos: *Neo4j* e *FlockDB*.

Apesar das vantagens relativamente à escalabilidade, desempenho das consultas e armazenamento de dados, estes motores são relativamente recentes. Este fator tem uma enorme importância, pois num projeto desta envergadura são necessárias tecnologias com um elevado grau de maturidade e de suporte, o que atualmente ainda não acontece com estes motores de bases de dados.

2.2. ETL

O ETL é a fase mais crítica e trabalhosa de um projeto de BI. Consiste no desenvolvimento de processos de **recolha** de informação proveniente de uma ou várias fontes e pelo **tratamento** da informação recolhida, principalmente do ERP SAP da UC e do projeto NÓNIO. O tratamento da informação pode passar por tratar valores nulos, valores duplicados, inconsistências nos dados, causados durante a inserção no sistema fonte e pelo cruzamento das várias fontes de informação.

Seguidamente, é executado o **carregamento** de toda a informação tratada no passo anterior para a *Data Mart*. A execução deste processo pode ser demorada, consoante a quantidade de dados a tratar e as operações a realizar. Após o processo de carregamento existe uma tarefa crucial, que consiste na validação dos dados carregados. Existe a obrigatoriedade de confirmar que todas as operações ocorridas durante o processo de ETL não alteraram a consistência dos dados.

Este processo é bastante simples de compreender, mas muito difícil de implementar, sendo que a dificuldade aumenta consoante as fontes de dados. Sendo um processo crucial, o ETL facilmente atinge os 70% de recursos alocados para a implementação e manutenção de uma *Data Warehouse*.

Uma grande prioridade do ETL é garantir que o processo seja automatizado, quer isto dizer que, é pretendido que se possa extrair informação das fontes, tratá-la e carregá-la na *Data Mart*, garantindo a integridade e consistência dos dados previamente carregados, automaticamente.

Existem duas possibilidades para a execução do processo de ETL, usando uma ferramenta de ETL ou usando código próprio. Qual deveremos usar? Ralph Kimball, em *Data Warehouse ETL Toolkit*[7] opta por não tomar partido de nenhuma, limitando-se a mostrar os prós de cada uma (Tabela 1).

São várias as soluções *open-source* de ferramentas ETL existentes no mercado, destacando-se a *Pentaho Data Integration* (*Kettle*) e o *Jasper ETL*. A comparação entre estas duas ferramentas é efetuada no anexo 1.

	Vantagens
Ferramentas ETL	• Simples, económico e rápido. • Capacidade de ligação à maioria das bases de dados e reconhecimento de vários tipos de ficheiros. • Capacidade de encriptação e compressão. • Maior facilidade de adaptação a mudanças de esquemas de dados. • Bom desempenho, ideal para grandes quantidades de dados. • Possibilidade de inserção de código próprio. • Boa documentação.
Código próprio	• Técnicas de programação orientada aos objetos são uma mais valia na validação. • Existência de ferramentas e *frameworks* que facilitam a criação do código. • Tratamento mais efetivo e direto dos metadados. • Grande flexibilidade.

Tabela 1: Vantagens de ferramentas ETL e código próprio.

2.3. Aplicação *Business Intelligence*

A última etapa é a disponibilização da informação, de modo a que o utilizador consiga analisar os dados, através de análises OLAP. Para isto é necessário um servidor de BI, que, preferencialmente, tenha compatibilidades com outras tecnologias, seja seguro, ofereça bom desempenho e fácil interação.

Esta é uma área que tem de ser estudada com um pouco mais de pormenor, porque é necessário ter em conta alguns fatores enumerados por Ralph Kimball em *The Data Warehouse Toolkit*[2]:

1. Informação disponível (via metadata)
2. Forma de apresentação da informação
3. Custo

Pensando nestas características, a pesquisa elaborada focou-se maioritariamente em encontrar as principais soluções *open-source*:

- *JasperReports Server*[8] – versão *open-source* disponibilizada pela *JasperSoft*. Esta versão tem menos funcionalidades do que a *Enterprise Edition*, destacando-se mesmo assim pela geração de análises e relatórios OLAP, disponíveis para plataformas *Web* ou *mobile*.
- *Pentaho BI Server*[9] – uma ferramenta semelhante à anterior, mas as suas mais valias são a possibilidade de criação de *dashboards,* e ter incorporado um servidor OLAP bastante conhecido, *Mondrian,* conseguindo assim analisar grandes quantidades de dados devido ao seu bom desempenho. O *Mondrian*[10], permite a criação e

armazenamento de um cubo (modelo multidimensional da DW), de modo a ser possível aplicar operações de *drill-down, roll-up, slice, dice, e drill across.*

No anexo 1 encontra-se uma descrição de soluções pagas, para comparar características entre as *open-source* e principalmente os seus custos.

2.4. Sumário

Todas as soluções referidas neste estudo poderiam ser opção para este projeto. Atualmente e num futuro não muito distante as bases de dados relacionais são a opção mais acertada, porque não irão ser um ponto crítico no desempenho da arquitetura e também devido à documentação e comunidade ativa ser superior às bases de dados *NoSQL*.

As ferramentas de ETL serão uma mais valia. As ferramentas aqui referidas respondem a todas as necessidades do projeto e prováveis necessidades futuras com a possibilidade de execução de *scripts* em *Javascript*.

Das aplicações de BI estudadas, a que tem a minha preferência é a *Pentaho BI Server* devido à comunidade ativa nos fóruns da *Pentaho* ser mais ativa e pelo servidor OLAP já incorporado.

Em suma, este estudo proporcionou uma melhor integração no projeto e na área de BI.

Capítulo 3
Requisitos

Neste capítulo é apresentado o processo de identificação de requisitos, bem como as suas análises e especificações efetuadas.

3.1. Identificação de requisitos e indicadores

O processo de identificação de requisitos foi realizado junto de um grupo operacional constituído por alguns dos principais responsáveis pelos Recursos Humanos da UC. Este grupo foi definido na primeira reunião, realizada no dia 30 de outubro de 2014.

Esta reunião, presidida em conjunto, pela Vice Reitora Professora Doutora Margarida Mano e pelo Vice Reitor Professor Doutor Luís Menezes tinha também o intuito de validar o trabalho desenvolvido no ano letivo 2013/2014 no módulo RH. Estiveram presentes quatro membros da equipa da DW, dois trabalhadores do grupo de GSIIC, quatro do grupo de Serviço de Gestão de Recursos Humanos, o Dr. Paulo Simões Lopes chefe da Divisão de Auditoria e o Dr. Filipe Rocha, chefe da Divisão de Planeamento, Gestão e Desenvolvimento.

No final da reunião ficou definido que o grupo operacional de RH seria constituído pelo Vice Reitor Professor Doutor Luís Menezes, dois operacionais do Serviço de Gestão de Recursos Humanos, pelo Dr. Paulo Simões Lopes, pela Dr. Ana Cruz, pelo Dr. Filipe Rocha, um membro do GSIIC responsável pelo desenvolvimento de *webservices* e dois membros da equipa da DW, onde eu me incluía.

Na segunda reunião, presidida pelo Vice Reitor Professor Doutor Luís Menezes, foi apresentada uma lista com 61 indicadores que surgiram maioritariamente de indicadores de gestão e outros pretendidos por entidades externas.

O passo seguinte foi analisar cada um dos indicadores, de modo a identificar quais já estavam armazenados na *Data Mart* e possuíam análises associadas, bem como dos restantes, quais deviam ser analisados e especificados nas fichas de indicadores[1], para potenciar a solução e tirar todas as dúvidas associadas a cada um deles.

As seguintes reuniões serviram para executar o processo de validação, que teve como principais atributos a prototipagem rápida e as fichas de indicadores. Os protótipos foram elaborados da forma mais análoga possível à solução real, com recurso a uma ferramenta de licença gratuita, *Justinmind Prototyper*.

Durante o processo de validação surgiram novos indicadores, provenientes do módulo do Plano Estratégico, que se enquadravam nos Recursos Humanos. Foi dado início à especificação imediata desses indicadores, de modo a poder ser iniciado o desenvolvimento de todos os indicadores com a maior brevidade possível.

Todos os indicadores encontram-se analisados e validados, tendo sido a sua priorização fornecida pela Vice Reitora Professora Doutora Margarida Mano.

---

[1] Fichas desenvolvidas que especificam o indicador, a sua granularidade mínima, a fonte da informação, as operações de *drill down* e *roll up* pretendidas, os filtros a aplicar e o espaço temporal a aplicar. Disponíveis no anexo 2.

3.2. Especificação de requisitos

Um dos modelos mais utilizados na classificação de requisitos é o FURPS+, derivado do modelo criado por Robert Grady na *Hewlett Packard* (HP). O seu nome é um acrónimo que corresponde a: (F) funcionalidade, (U) usabilidade, (R) fiabilidade, (P) performance, (S) suporte e (+) outros.

Os requisitos estão divididos em duas categorias, funcionais e não funcionais, sendo que os primeiros estão relacionados com a funcionalidade da aplicação e os restantes com os seus atributos e a sua qualidade.

3.2.1. Requisitos funcionais

Foi necessário dividir os requisitos funcionais em dois tipos, os gerais e os específicos do módulo RH.

Naturalmente, os requisitos gerais já se encontram definidos desde o início do projeto, pelo que existe a obrigatoriedade de manter a conformidade com o que está desenvolvido. Estes requisitos estão apresentados na Tabela 2.

Código	**Designação**	**Prioridade**	**Dependência**	**Fonte**	**Descrição**
RF_GE_001	Autenticação	Elevada	-	Cliente	A aplicação deve permitir ao utilizador a autenticação (*login*) através das credenciais utilizadas no acesso a quaisquer serviços disponibilizados à comunidade da UC (email da UC e password). A autenticação tem de ser bem sucedida, isto é, as credenciais têm de ser válidas para que seja permitida qualquer visualização de dados ao utilizador.
RF_GE_002	Fechar sessão	Elevada	RF_GE_013	Cliente	O utilizador pode, após autenticação, efetuar o término da sua sessão (*logout*).
RF_GE_003	Término de sessão	Elevada	RF_GE_013	Cliente	Para garantir a segurança da aplicação, um utilizador, depois de autenticar-se terá associada uma sessão, esta deve ter um *timeout* para efetuar *logout* automaticamente.
RF_GE_004	Navegação entre módulos	Média	RF_GE_013	Cliente	O utilizador deve conseguir aceder a todos os restantes módulos, este acesso deve ser possível a qualquer momento.

RF_GE_005	Navegação interna	Elevada	RF_GE_011	Cliente	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	RF_GE_011	Cliente	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	RF_GE_011	Cliente	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	RF_GE_011	Equipa	Deve ser permitido ao utilizador esconder a barra onde se encontram os parâmetros gerais e de tempo.
RF_GE_009	Secção de ajuda	Elevada	RF_GE_011	Cliente	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	RF_GE_011	Cliente	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.).
RF_GE_011	Visualização: gráfico-tabela	Elevada	RF_GE_013	Cliente	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	RF_GE_011	Cliente	Exportar para formato Excel ou CSV a informação presente nas tabelas de análise. Exportar gráfico como imagem.
RF_GE_013	Autorização	Elevada	RF_GE_001	Cliente	O utilizador após autenticação, pode visualizar a informação afeta ao(s) módulo(s) em que está inserido. Por exemplo: se um utilizador pertencer ao grupo DW_SE deve conseguir aceder a todo o módulo de sucesso escolar.
RF_GE_014	Zoom de gráficos	Baixa	RF_GE_011	Equipa	Efetuar zoom *in* e *out* de um dos gráficos.
RF_GE_015	Visualização de trabalhadores (RH)	Elevada	RF_GE_013	Cliente	Para determinados indicadores, a aplicação deve disponibilizar a informação ao nível do trabalhador.

Tabela 2: Requisitos funcionais da aplicação – gerais.

Os requisitos específicos do módulo de RH são os indicadores que foram debatidos nas reuniões do grupo operacional e para os quais é necessário desenvolver análises OLAP. A Tabela 3 apresenta os indicadores a desenvolver, com o seu código identificador único, bem como a sua prioridade e uma breve descrição.

Código	**Prioridade**	**Designação**	**Descrição**
IND_RH_001	Elevada	Mobilidade (In/Out)	Indicador que representa a mobilidade dos trabalhadores da UC, por postos de trabalho ocupados e ETI.
IND_RH_002	Média	Suspensão	Indicador que representa o número de suspensões, por postos de trabalho ocupados e ETI.
IND_RH_003	Média	Modalidade horários	Indicador que representa o número de horários, por postos de trabalho ocupados e ETI.
IND_RH_004	Média	Trabalho suplementar	Indicador que representa o número horas de trabalho suplementar de trabalhadores não docentes, por postos de trabalho ocupados e ETI.
IND_RH_006	Baixa	SIADAP	Indicador que representa o número de trabalhadores não docentes da UC por avaliação (qualitativa).
IND_RH_007	Elevada	Antiguidade média	Indicador que representa a antiguidade média dos trabalhadores, por postos de trabalho ocupados e ETI.

IND_RH_008	Elevada	Índice de absentismo	Indicador que representa a percentagem de ausências ao trabalho.
IND_RH_009	Baixa	Potencial máximo anual	Indicador que representa o potencial máximo anual.
IND_RH_010	Elevada	Índice de admissão de trabalhadores	Indicador que representa a taxa de admissão de trabalhadores admitidos ao serviço.
IND_RH_011	Elevada	Índice de saída de trabalhadores	Indicador que representa a taxa de saída de trabalhadores que deixaram o serviço.
IND_RH_012	Elevada	Índice de reposição	Indicador que mensura a relação entre os trabalhadores saídos e admitidos.
IND_RH_013	Elevada	Índice de rotação	Indicador que representa a rotatividade dos trabalhadores.
IND_RH_014	Elevada	Índice de envelhecimento	Indicador que representa a relação entre o número de trabalhadores com mais de 55 anos e o número total de trabalhadores.
IND_RH_015	Elevada	Nível etário	Indicador que representa a idade média dos trabalhadores.
IND_RH_016	Elevada	Leque etário	Indicador que representa a diferença de idades entre o trabalhador mais velho e o trabalhador mais novo.
IND_RH_017	Elevada	Índice de emprego jovem	Indicador que representa a relação entre o número de trabalhadores com menos de 25 anos e o número total de trabalhadores.
IND_RH_018	Elevada	Índice de rejuvenescimento	Indicador que representa a relação entre o número de trabalhadores com menos de 25 anos e o número de trabalhadores com mais de 55 anos.
IND_RH_019	Elevada	Percentagem de postos de trabalho ocupados	Indicador que representa o ratio entre trabalhadores da UC.
IND_RH_030	Elevada	Custo hora	Indicador que representa o Custo por hora de trabalhadores docentes, investigadores e não docentes.
IND_RH_031	Média	Suplementos remuneratórios	Indicador que representa o valor de suplementos remuneratórios, o nº de suplementos e o nº de postos de trabalho ocupados, por tipo de suplemento.
IND_RH_032	Elevada	Leque salarial ilíquido	Indicador que representa a diferença entre o vencimento base ilíquido mais alto e o vencimento ilíquido mais baixo.

IND_RH_033	Elevada	Carga Salarial	Indicador que representa o encargo com salários face à despesa total.
IND_RH_034	Elevada	Despesa com pessoal "per capita"	Indicador que representa o valor da despesa "per capita" de trabalhadores.
IND_RH_035	Média	Taxa de complementos e encargos sociais	Indicador que representa a taxa de complementos e encargos sociais.
IND_RH_036	Elevada	Remuneração base média	Indicador que representa o valor de remuneração base média.
IND_RH_043	Média	Nº de suplementos remuneratórios	Indicador que representa o nº de suplementos, por tipo de suplemento.
IND_RH_044	Média	Nº de postos de trabalho ocupados por suplementos remuneratórios	Indicador que representa o nº de postos de trabalho ocupados, por tipo de suplemento.
IND_RH_045	Elevada	Nº de orientações	Nº de orientações do 2º e 3º ciclo.
IND_RH_046	Elevada	Ratio docente alunos	Indicador que representa a relação entre os professores e o número de alunos.
IND_RH_047	Elevada	Nº de horas de aulas	Indicador que representa o número de horas de aulas.
IND_RH_048	Elevada	Nº de horas previstas de aulas	Indicador que representa o número de horas, previstas, de aulas.
IND_RH_049	Elevada	Média de horas lecionadas por semana	Indicador que representa a média de horas de aulas lecionadas por semana.
IND_RH_050	Média	Nº de acidentes	Indicador que representa o número de acidentes dos trabalhadores da UC.
IND_RH_051	Elevada	Capacidade turmas	Indicador que representa a capacidade de turmas.
IND_RH_052	Elevada	Nº de unidades curriculares por docente	Indicador que representa o número de unidades curriculares por docente.
IND_RH_053	Elevada	Nº de turmas por docente	Indicador que representa o número de turmas por docente.
IND_RH_054	Elevada	Nº de aulas por docente	Indicador que representa o número de aulas por docente.

IND_RH_055	Elevada	Nº de estudantes inscritos definitivamente	Indicador que representa o número de estudantes inscritos definitivamente, em unidades curriculares, turmas e aulas.
IND_RH_056	Elevada	Nº médio de estudantes inscritos	Indicador que representa o número médio de estudantes inscritos, em unidades curriculares, turmas e aulas.
IND_RH_057	Elevada	Percentagem de ocupação das turmas	Indicador que representa a percentagem de ocupação de turmas.
IND_RH_058	Elevada	Nº de turmas com menos de x estudantes	Indicador que representa a o número de turmas com menos de x estudantes, sendo x um número variável.
IND_RH_059	Elevada	Carga horária padrão	Indicador que representa carga horária prevista.
IND_RH_060	Elevada	Custo hora lecionada	Indicador que representa o custo por hora lecionada de trabalhadores docentes.
IND_RH_061	Elevada	Carga horária contrato	Indicador que representa carga horária ponderada pela percentagem de contrato (desdobrado por tipos de turmas: T, TP, PL, etc.).

Tabela 3: Requisitos funcionais da aplicação – recursos humanos.

Todas as fichas de indicadores desenvolvidas encontram-se no anexo 2.

3.2.2. Requisitos não funcionais

Visto que este estágio visa a remodelação e manutenção de um módulo integrante de uma aplicação, é necessário que os requisitos não funcionais permaneçam inalterados e que sejam transversais a todos os módulos. A Tabela 4 apresenta uma visão geral dos requisitos não funcionais.

Código	**Designação**	**Prioridade**	**Fonte**	**Descrição**
RFN_S_001	Atualização de dados	Elevada	Cliente	Processo ETL e atualização da DW e cubo OLAP devem ser automáticos.
RFN_S_002	Compatibilidade (browser)	Elevada	Cliente	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.
RFN_S_003	Compatibilidade (SO)	Média	Equipa	Como sistema operativo para os servidores deve ser suportada a distribuição de *Linux Red Hat – CentOS* 6.x.
RFN_S_004	Licenças	Elevada	Cliente	A aplicação deve ser desenvolvida e disponibilizada através de *software* gratuito.

RFN_S_005	Monitorização de erros	Média	Equipa	Deve existir um mecanismo para gestão de *logs*.
RNF_O_001	Hardware	Elevada	Equipa	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 características 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	Cliente	Deve ser utilizado o protocolo HTTPS em toda a comunicação entre os utilizadores e a aplicação.
RNF_O_003	Autenticação	Elevada	Cliente	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.
RNF_R_001	Mecanismo de *fail over*	Elevada	Equipa	Deve existir um mecanismo que garanta o funcionamento da aplicação, mesmo quando o servidor primário está indisponível.
RNF_R_002	Mecanismo de recuperação de falhas	Elevada	Equipa	O sistema deve conter um mecanismo para iniciar automaticamente os serviços necessários ao seu correto funcionamento.
RNF_R_003	Backup dos dados	Elevada	Equipa	Armazenar periodicamente backups dos dados presentes na base de dados.
RNF_P_001	Utilizadores em simultâneo	Elevada	Cliente	A aplicação deve suportar pelo menos 60 utilizadores em simultâneo.
RNF_P_003	Desempenho do *front-end*	Elevada	Equipa	A medida de qualidade deve ser igual ou superior a 80 (considerando a pontuação disponibilizada pelo o *Yslow* - http://yslow.org).
RNF_U_001	Satisfação dos utilizadores	Elevada	Equipa	A percentagem de satisfação dos utilizadores, com a usabilidade da aplicação, não deve ser inferior a 80%.
RNF_U_002	Tempo de carregamento	Elevada	Equipa	O tempo de carregamento das páginas *web* deve ser, no máximo, 5 segundos.

Tabela 4: Requisitos não funcionais do projeto UC-Num.

Capítulo 4
Arquitetura

O presente capítulo tem como objetivo a descrição da arquitetura do sistema desenvolvido, bem como as tecnologias selecionadas para a sustentar.

4.1. Arquitetura Global

A construção de um sistema de suporte à decisão é divido em três fases, como mostra a Figura 3.

1. Criação da *Data Mart*, com recurso à extração, transformação e carregamento de informação de uma ou várias fontes de dados.
2. Desenvolvimento de análises sobre a informação disponível na *Data Mart*, a partir de uma aplicação de *Business Intelligence*.
3. Disponibilização da análise previamente desenvolvida.

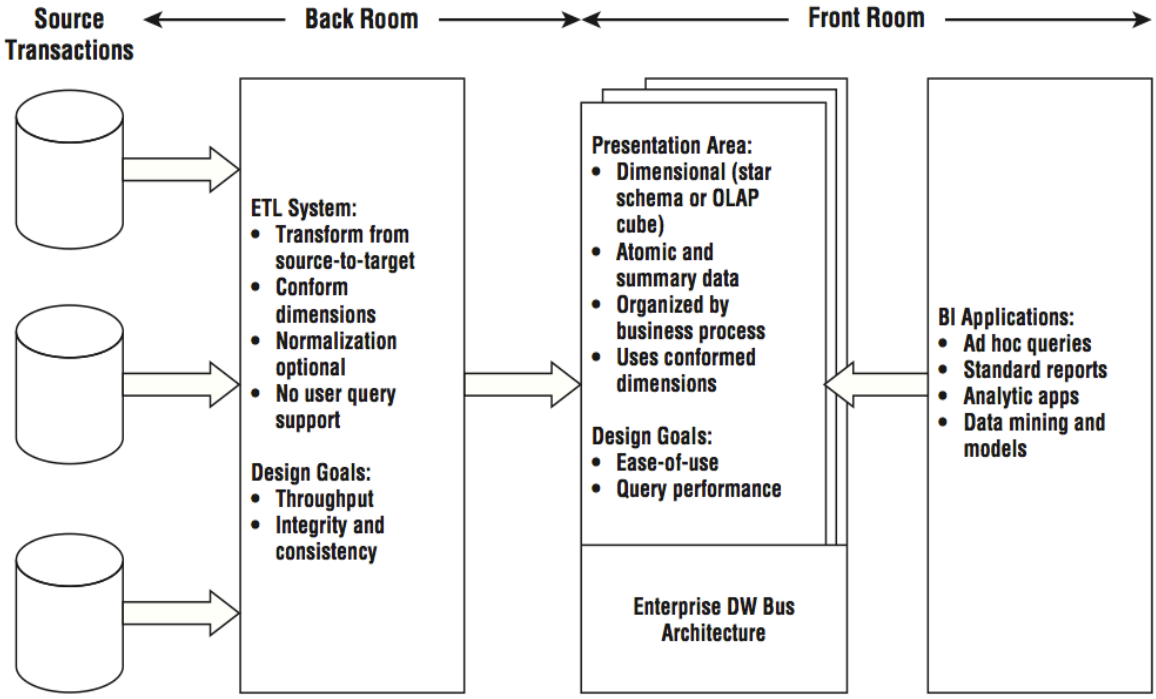

Figura 3: Arquitetura Global do Sistema, segundo Ralph Kimball[2].

4.2. Seleção de Tecnologias

Como referido no Capítulo 2, todas as tecnologias já estavam selecionadas à data de início do estágio. As tecnologias escolhidas para este projeto foram:

1. **PostgreSQL**: bom desempenho e escalabilidade, bem como pelas funcionalidades apresentadas para otimização (índices e vistas materializadas).

2. **Pentaho Data Integration (Kettle)**: facilidade de interação, rapidez e possibilidade de paralelismo.
3. **Pentaho BI** e **Mondrian:**
    a. Através do *plugin* de criação de *dashboards*, CDE, é possível criar *dashboards* com relativa facilidade, que são posteriormente disponibilizadas pelo *plugin* CDF, que faz uso de tecnologias como HTML, CSS e *Javascript*, permitindo criar e definir os componentes das *dashboards*.
    b. Para a construção do Cubo OLAP, tendo assim uma maior velocidade de acesso aos dados e capacidades de análise mais ricas.

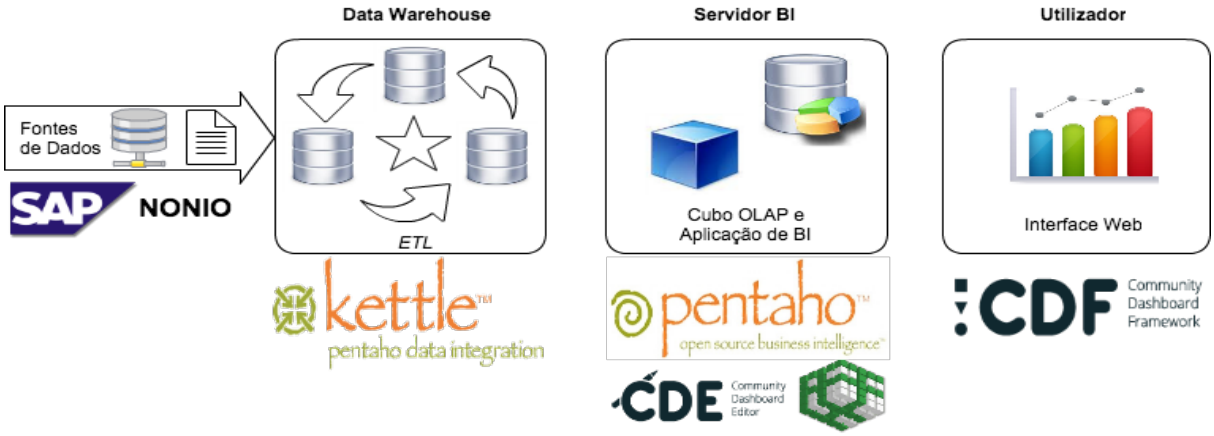

Figura 4: Visão de alto nível da arquitetura atual do sistema e tecnologias adotadas.

Após a visão de alto nível da arquitetura, na Figura 4, é necessário entender melhor como funcionam e comunicam todas estas ferramentas:

- Extração de informação: toda a informação relativa aos trabalhadores e ao seu percurso na UC é proveniente do sistema SAP, sendo acedida através de *webservices*, desenvolvidos exclusivamente para este projeto pelo GSIIC. A informação relativa ao serviço docente é proveniente do sistema NÓNIO, acedido através de uma vista materializada fornecida pelos seus administradores.
- BI/Cubo OLAP: a informação armazenada na *Data Mart* é acedida pelo cubo através de uma ligação SQL. O servidor de BI acede à informação através de duas formas distintas, ao cubo através de *queries* multidimensionais na linguagem MDX[11] e à *Data Mart* via SQL.
- Visualização da informação: para o utilizador conseguir visualizar toda a informação, necessita de fazer um *login* com as credenciais de email de trabalhador da UC. Após estar autorizado, são executados pedidos HTTPS ao servidor de BI. Os pedidos HTTPS originam um conjunto de ações específicas no *plugin* CDF, que vai permitir visualizar as *dashboards*, como é visível na Figura 5.

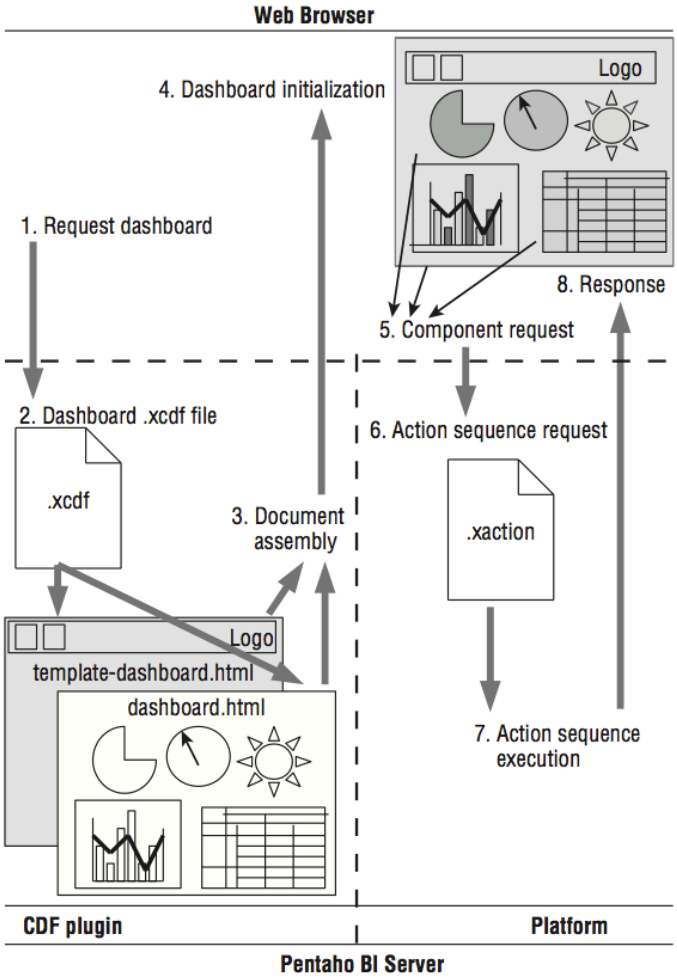

Figura 5: Visão de alto nível *plugin Community Dashboard Framework*[12].

4.3. Modelo de Dados

Com o desenrolar do projeto surge a necessidade de criação de dois modelos de dados, modelo multidimensional e modelo da área temporária.

4.3.1. Modelo multidimensional

Após a recolha e transformação dos dados, é iniciado o processo de carregamento desses dados para uma base de dados com um modelo multidimensional, como foi referido anteriormente neste capítulo. O modelo multidimensional é sobejamente conhecido como modelo em "estrela". É composto por factos e dimensões, onde factos são valores numéricos aditivos ou semiaditivos e as dimensões são atributos críticos que caracterizam os factos e que fazem o sistema de BI ser compreendido e usável.

No caso desta *Data Mart* não existirá só uma "estrela" mas sim 11, devido à necessidade de cobrir todos os indicadores especificados.

- *rh_f_admissoes_contratos*: tabela de factos com informação remuneratória e contratual dos trabalhadores.

- *rh_f_absentismo*: tabela de factos com informação relativa ao absentismo dos trabalhadores
- *rh_f_suspensoes_saidas*: tabela de factos com informações sobre as saídas e suspensões dos trabalhadores.
- *rh_f_acidentes*: tabela de factos com informação dos acidentes de trabalho.
- *rh_f_mobilidade_funcao_publica*: tabela de factos com informação das ações de mobilidade dos trabalhadores.
- *rh_f_trabalho_suplementar*: tabela de factos com informação do trabalho extraordinário realizado pelos trabalhadores.
- *rh_f_potencial_maximo*: tabela de factos com o valor do potencial máximo anual.
- *rh_f_docencia*: tabela de factos com informações sobre o serviço docente.
- *rh_f_SIADAP*: tabela de factos com os resultados das avaliações SIADAP.
- *rh_f_suplementos_remuneratorios*: tabela de factos com valores dos suplementos remuneratórios pagos aos trabalhadores.
- *rh_f_trabalhadores_ativos*: tabela de factos com informação dos trabalhadores ativos para cálculo de índices e taxas.

A Figura 6 demonstra todas as tabelas de facto presentes no modelo de dados da Data Mart.

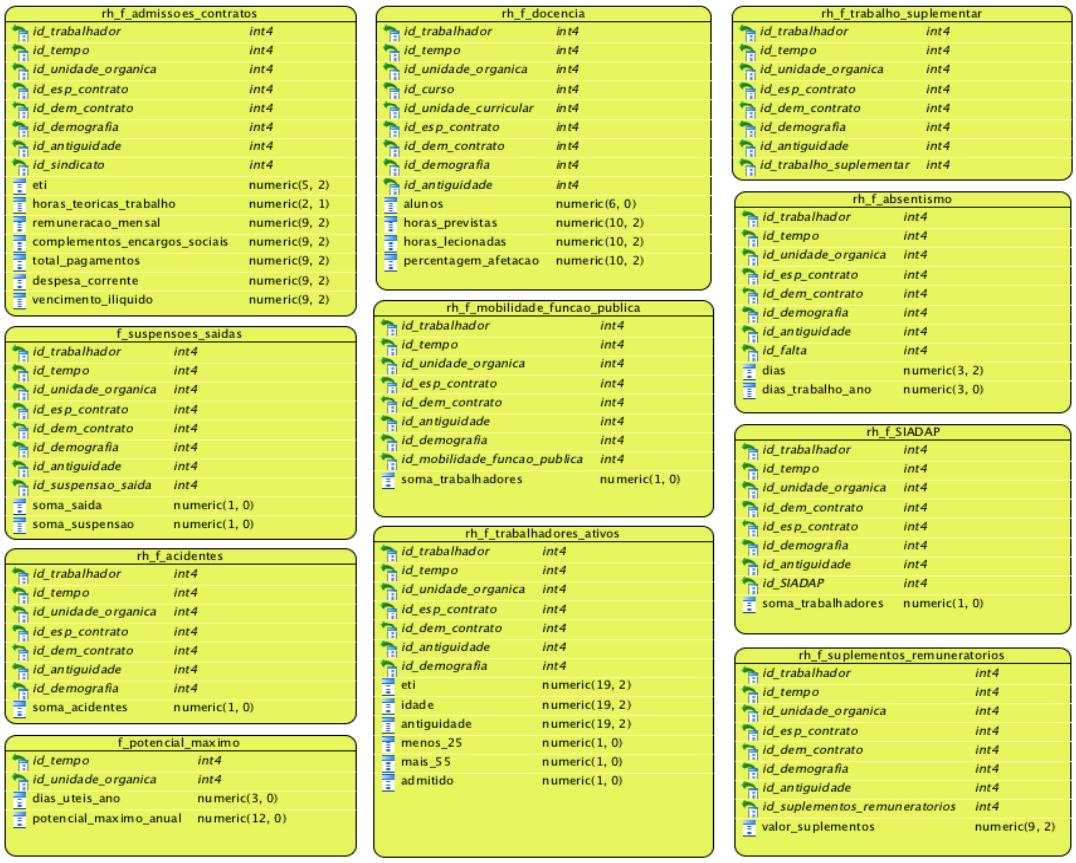

Figura 6: Tabelas de factos do modelo multidimensional.

Como caracterização aos factos existem 16 dimensões que são apresentadas na Figura 7 e descritas na Tabela 5.

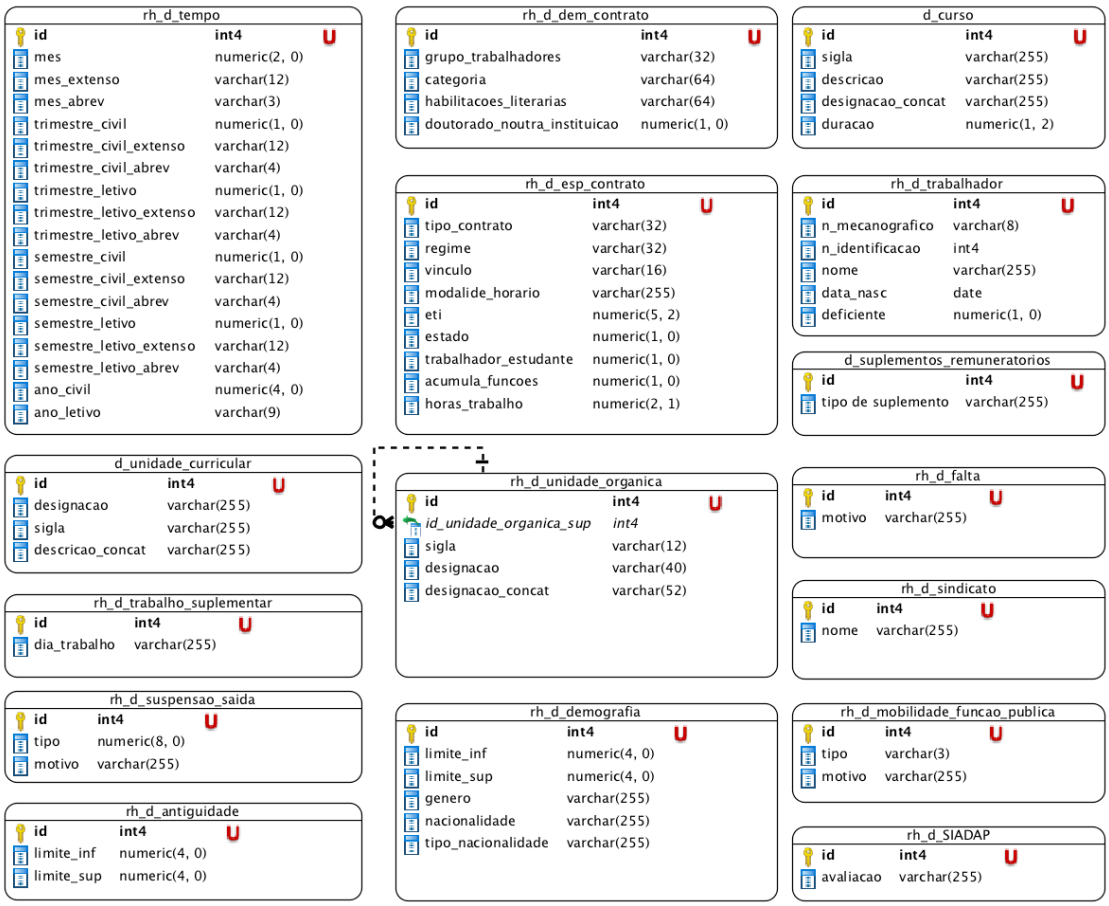

Figura 7: Dimensões do modelo multidimensional.

Dimensão	**Descrição**
rh_d_tempo	Dimensão temporal, que é essencial em qualquer *Data Mart* e serve como atributo para todas as tabelas de factos, podendo assim restringir o espaço temporal para as análises. A granularidade mínima é o trimestre relativamente ao ano letivo e o mês para o ano civil.
rh_d_dem_contrato	Dimensão que retrata os dados demográficos dos contratos dos trabalhadores da UC: o grupo de trabalhadores e a categoria profissional em que estão inseridos, as suas habilitações literárias e se é doutorado noutra instituição que não a UC. É utilizada no agrupamento de dados e também para os filtrar.
rh_d_unidade_organica	Dimensão que retrata as unidades orgânicas da UC. Serve como atributo a todas as tabelas de factos, referentes às unidades orgânicas.

rh_d_trabalhador	Dimensão que retrata os trabalhadores da UC.
rh_d_esp_contrato	Dimensão que retrata os dados mais específicos dos contratos dos trabalhadores da UC: o tipo de contrato, o regime, vínculo, a modalidade de horário, o valor de ETI, o estado do seu contrato, se é trabalhador estudante, se acumula funções noutra organização que não a UC e o número de horas de trabalho diário. É utilizada no agrupamento de trabalhadores e também como filtro.
rh_d_demogragfia	Dimensão que representa os aspetos demográficos do trabalhador, faixa etária, género e nacionalidade. É utilizada no agrupamento de trabalhadores e também como filtro.
rh_d_antiguidade	Dimensão que à semelhança da anterior também representa intervalos de 5 anos mas na antiguidade dos trabalhadores. É utilizada no agrupamento de trabalhadores e também como filtro.
rh_d_falta	Dimensão que representa as faltas dos trabalhadores, contendo o motivo da falta. É utilizada no agrupamento de trabalhadores e também como filtro.
rh_d_suspensao_saida	Dimensão que retrata as suspensões e saídas dos trabalhadores, contendo o motivo do sucedido. É utilizada no agrupamento de trabalhadores.
rh_d_mobilidade_ funcao_publica	Dimensão que retrata a mobilidade de trabalhados na função pública, contendo o motivo e o tipo da mobilidade (*in* ou *out*). É utilizada no agrupamento de trabalhadores.
rh_d_curso	Dimensão que representa um curso da UC. É um atributo dos factos da docência na UC.
rh_d_unidade_curricular	Dimensão que retrata uma unidade curricular da UC. É uma atributo dos factos da docência na UC.
rh_d_suplementos_ remuneratiorios	Dimensão que retrata o tipo de suplementos remuneratórios pagos da UC. É utilizada para filtragem.
rh_d_SIADAP	Dimensão que representa a avaliação SIADAP, para trabalhadores não docentes. É utilizada para agrupar trabalhadores por avaliação qualitativa.
rh_d_trabalho_ suplementar	Dimensão que caracteriza o trabalho suplementar pelos diferentes tipos de dias (feriados, fins de semana, etc.).
rh_d_sindicato	Dimensão que retrata os sindicatos onde os trabalhadores estão sindicalizados. É utilizada no agrupamento de trabalhadores por sindicatos e também como filtro.

Tabela 5: Descrição das dimensões do modelo de dados.

Para alojamento dos factos existem 11 tabelas, presentes na Figura 6. Abaixo estão descritos os factos de 8 tabelas, as restantes 3 tabelas de factos (*rh_f_SIADAP, rh_f_acidentes,* e *rh_f_mobilidade_funcao_publica*) servem como cruzamento de dimensões. No anexo 3 encontra-se o modelo de dados da *Data Mart*.

rh_f_absentismo		
Factos	**Nome**	**Descrição**
	dias	Facto que permite saber os dias de falta do trabalhador.
	dias_trabalho_ano	Facto que permite saber os dias de trabalho no ano com exceção das férias do trabalhador.
Granularidade		A granularidade mais fina é o trabalhador. É possível aplicar operações de *slice and dice* com todas as suas dimensões. O *drill down e roll up* é efetuado através na hierarquia da UC.

Tabela 6: Especificação da tabela de factos *rh_f_absentismo*.

rh_f_suspensoes_saidas		
Factos	**Nome**	**Descrição**
	soma_saida	Facto que permite saber se os dados naquela linha da tabela correspondem a uma saída, podendo ser feita a contagem de saídas.
	soma_suspensao	Facto que permite saber se os dados naquela linha da tabela correspondem a uma suspensão, podendo ser feita a contagem de suspensões.
Granularidade		A granularidade mais fina é o trabalhador. É possível aplicar operações de *slice and dice* com todas as suas dimensões. O *drill down e roll up* é efetuado através na hierarquia da UC.

Tabela 7: Especificação da tabela de factos *rh_f_suspensoes_saidas*.

rh_f_suplementos_remuneratorios		
Factos	**Nome**	**Descrição**
	valor_suplementos	Valor de suplementos pago a um trabalhador, no respetivo espaço temporal.
Granularidade		A granularidade mais fina é o trabalhador. É possível aplicar operações de *slice and dice* com todas as suas dimensões. O *drill down e roll up* é efetuado através na hierarquia da UC, inclusive, sendo possível descer até aos trabalhadores.

Tabela 8: Especificação da tabela de factos *rh_f_potencial_maximo*.

rh_f_admissoes_contratos		
Factos	**Nome**	**Descrição**
	eti	Percentagem de ETI do trabalhador.
	horas_teoricas	Número de horas de trabalho do trabalhador.
	remuneração_mensal	Remuneração base mensal do trabalhador.
	complementos	Valor dos complementos recebidos pelo trabalhador.
	total_pagamentos	Valor total dos pagamentos recebidos pelo trabalhador.
	despesa_corrente	Valor da despesa corrente recebido pelo trabalhador.
	vencimento_iliquido	Valor do vencimento ilíquido recebido pelo trabalhador.
Granularidade		A granularidade mais fina é o trabalhador. É possível aplicar operações de *slice and dice* com todas as suas dimensões. O *drill down e roll up* é efetuado através na hierarquia da UC.

Tabela 9: Especificação da tabela de factos *rh_f_admissoes_contratos*.

rh_trabalhadores_ativos		
Factos	**Nome**	**Descrição**
	eti	Percentagem de ETI do trabalhador.
	antiguidade	Facto com a antiguidade do trabalhador.
	idade	Facto com a idade do trabalhador.
	menos_25_anos	Facto que permite saber se o trabalhador possui menos de 25 anos, podendo ser feita a contagem de trabalhadores com menos de 25 anos.
	mais_55_anos	Facto que permite saber se o trabalhador possui mais de 55 anos, podendo ser feita a contagem de trabalhadores com mais de 55 anos.
	admitido	Facto que permite saber se o trabalhador foi admitido, podendo através deste facto saber quantos trabalhadores foram admitidos.
Granularidade		A granularidade mais fina é o trabalhador. É possível aplicar operações de *slice and dice* com todas as suas dimensões. O *drill down e roll up* é efetuado através na hierarquia da UC.

Tabela 10: Especificação da tabela de factos *rh_trabalhadores_ativos*.

rh_f_potencial_maximo		
Factos	**Nome**	**Descrição**
	dias_uteis_ano	Dias úteis do ano, excluindo os dias de férias dos trabalhadores.
	potencial_maximo_anual	Facto já calculado com o potencial máximo no respetivo espaço temporal.
	despesa_corrente	Facto da despesa corrente da UC no respetivo espaço temporal.
Granularidade		A granularidade mais fina é a Unidade orgânica. É possível aplicar operações de *slice and dice* na dimensão temporal, *drill down e roll up* na hierarquia da UC.

Tabela 11: Especificação da tabela de factos *rh_f_potencial_maximo*.

rh_f_docencia		
Factos	**Nome**	**Descrição**
	horas_previstas	Horas previstas de uma unidade curricular.
	horas_lecionadas	Horas lecionadas de uma unidade curricular.
	percentagem_afetacao	Percentagem de afetação do professor à unidade curricular.
Granularidade		A granularidade mais fina é o docente. É possível aplicar operações de *slice and dice* com todas as suas dimensões. O *drill down e roll up* é efetuado através na hierarquia da UC, cursos, unidades curriculares e por último nível o docente.

Tabela 12: Especificação da tabela de factos *rh_f_docencia*.

Tabela	**Descrição**
rh_f_SIADAP	São tabelas que servem para cruzar toda a informação das dimensões que lhes estão relacionas. A granularidade mínima em todas é o trabalhador, sendo possível operações de *slice and dice* com todas as suas dimensões.
rh_f_acidentes	
rh_f_mobilidade_funcao_publica	

Tabela 13: Especificação das restantes tabelas de factos.

4.3.2. Modelo da área temporária

A área temporária surge com a necessidade de armazenar os dados provenientes das diferentes fontes, para que posteriormente possam fazer parte do processo de ETL. Possui

um modelo relacional e vai alojar toda informação de RH com relevância para os indicadores especificados. Os dados são provenientes do sistema SAP e também do sistema NÓNIO.

No anexo 4 está representado o seu modelo de dados.

Capítulo 5
Desenvolvimento

Este capítulo consiste na descrição do processo de desenvolvimento, incluindo os desafios, os problemas ocorridos e as tomadas de decisão. O capítulo está divido em quatro subcapítulos, um por cada fase do desenvolvimento: ETL, cubos, *dashboards* e desempenho e optimizações.

5.1. ETL

O processo de ETL é composto por três fases distintas: o carregamento da área temporária, o carregamento das dimensões e por último o carregamento das tabelas de facto.

Para melhor compreensão é necessário explicar alguns conceitos e notações da ferramenta utilizada no processo de ETL. O *Pentaho Data Integration* processa a informação linha a linha, podendo cada linha sofrer um leque variado de operações. Na Tabela 14 estão presentes algumas dessas operações e alguns *steps* fundamentais para a interação de todo o processo.

Tipo	Ícone	Designação	Descrição
Flow		*Job*	Permite organizar e estruturar as chamadas a um conjunto de transformações ou inclusive outros *jobs*.
		*Transformation*	Permite executar um conjunto de *steps* que efetuam as transformações necessárias ao carregamento de dados.
		*Abort*	Permite abortar um *job* imediatamente.
		*Switch/Case*	Permite selecionar fluxos de informação consoante determinas condições introduzidas pelo utilizador.
Input		*Table input*	Permite fazer uma pesquisa SQL na base de dados.
		*CSV input*	Permite fazer a leitura de um ficheiro CSV retornando todas as suas linhas.
		*Generate rows*	Permite gerar uma ou mais linhas com vários campos preenchidos.
		*Get data from XML*	Permite obter informação de ficheiros ou repostas aos *webservices* no formato XML.
		*Microsoft Excel input*	Permite fazer a leitura de um ficheiro Excel.
Data Warehouse		*Combination lookup update*	Permite inserir informação na tabela. O seu fluxo passa por pesquisar um conjunto de atributos na tabela, caso não existam, a informação é inserida, caso contrário, é feita uma atualização sobre a existente.

Output		*Insert/update*	Permite a inserção/atualização na base de dados consoante determinadas restrições.
		*Table output*	Permite efetuar inserções na base de dados.
Transform		*Add constant*	Permite adicionar a cada linha um valor constante.
		*Calculator*	Permite fazer operações aritméticas com diferentes tipos de valor (números, datas, etc.).
		*Closure Generator*	Permite gerar uma *closure table* (calcula a distância entre pais e filhos numa hierarquia).
		*Select values*	*Step* utilizado para eliminar ou modificar os atributos presentes no fluxo de dados atual do processo.
		*Unique rows*	Remove duplicados no fluxo de dados ordenado. Pode inclusive definir quais os campos a comparar.
Utility		*If field value is null*	Substitui um valor *null* por um valor definido pelo utilizador.
		*Write to log*	Permite escrever para *log*, com uma determinada mensagem e/ou a informação de cada linha.
Scripting		*Modified Java Script Value*	Permite criar *scripts* na linguagem de *Javascript*.
		*Execute SQL script*	Permite criar *scripts* em SQL, ideal para limpar as tabelas da área temporária, criação de índices, etc.
Lookup		*HTTP Post*	Efetua a ligação a *webservices*, utilizado para recolher os dados provenientes de SAP.
		*Database Join*	Permite efetuar uma *query* SQL e fazer a junção do seu resultado com cada linha a ser processada.
		*Stream lookup*	Permite fazer uma pesquisa com o intuito de retornar um valor. Necessita de um *step "Table input"* para receber os dados onde vai procurar pelo registo pretendido.

Tabela 14: Descrição dos *steps* utilizados durante o processo de ETL.

5.1.1. Área temporária

O primeiro passo do desenvolvimento começou com o carregamento da área temporária, o que em suma consiste no carregamento de toda a informação proveniente de SAP e NÓNIO para uma base de dados com o modelo de dados presente no anexo 4.

Para o primeiro carregamento pretendia-se obter informação mais fidedigna possível, pelo que se optou por recolher informação desde janeiro de 2011. Esta opção foi tomada em conjunto com os responsáveis de RH, visto que só a partir dessa data o sistema SAP começou a ter informação estável.

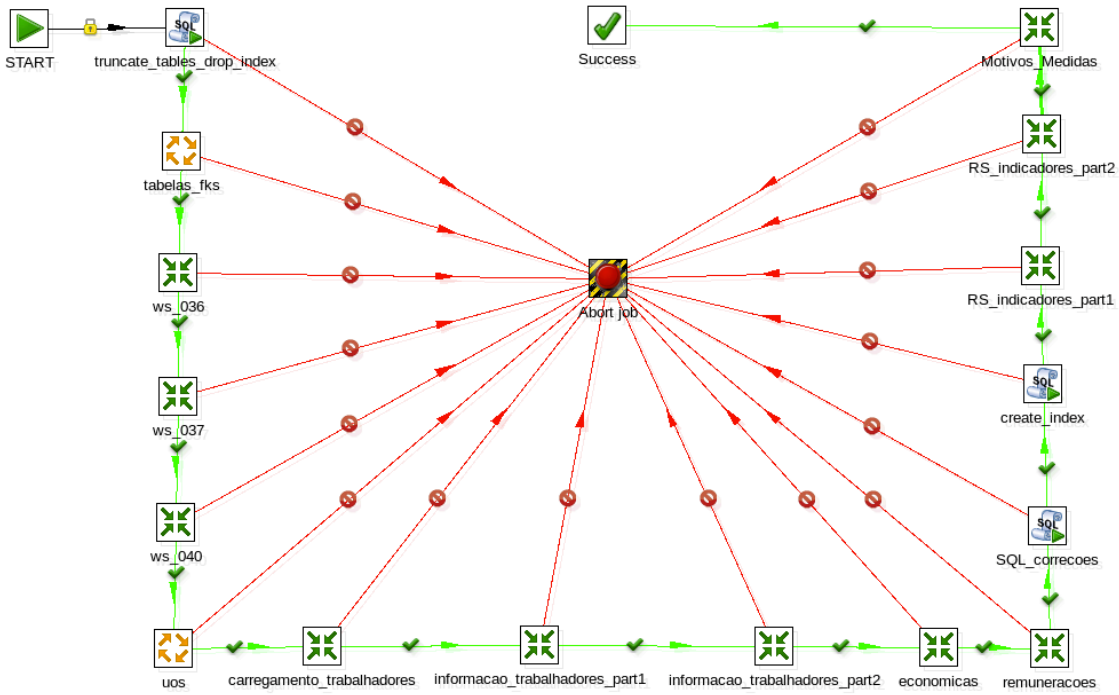

Figura 8: Job de carregamento da área temporária.

Na Figura 8 são visíveis as várias etapas do processo de carregamento da área temporária, iniciando-se com a limpeza de toda a informação e remoção de todos os índices. A etapa seguinte é a invocação dos *webservices,* onde se distinguem dois tipos: os que necessitam de parâmetros para serem invocados (informação temporal) e os que não necessitam de qualquer parâmetro. Os *webservices* que necessitam de receber parâmetros, retornam informação que não é igual todos os meses ou todos os dias, como é o caso da informação dos trabalhadores, as operações financeiras e as unidades orgânicas ativas. Exemplos de *webservices* que não necessitam de parâmetros são: todas as categorias profissionais, os grupos de trabalhadores e as rúbricas salariais existentes na UC. A resposta de cada *webservice* é um objeto XML.

Para o carregamento da informação proveniente dos *webservices* que recebem parâmetros, optei por guardar o resultado XML do *webservice* numa tabela da base de dados com o respetivo parâmetro para não sobrecarregar o sistema SAP sempre que fosse necessário executar o ETL no período de implementação e para poder otimizar o processo de leitura da informação de um determinado mês, enquanto que nos restantes a informação é extraída do XML e carregada para a base de dados.

Como era necessário efetuar um carregamento de um histórico considerável, foi necessário dividir algumas transformações de modo a não exceder a memória do processo Java associado ao *Pentaho Data Integration* o que causaria uma exceção e consequentemente o término do processo de carregamento. Contudo, esta solução verificou-se insuficiente, pelo

que foi necessário incluir nas transformações que consumiam mais memória um *step* que permitia atrasar a chegada de informação aos restantes *steps* (*delay row*), diminuindo assim a memória utilizada. A Figura 9 mostra uma transformação que tem como *input* um CSV que retorna um mês e um ano, a cada 45 segundos de modo a não possuir em memória todos os meses que tem de inserir na base de dados. Esta operação tem um custo temporal considerável no 1º carregamento, enquanto que os carregamentos subsequentes, 45 segundos terão um impacto praticamente nulo.

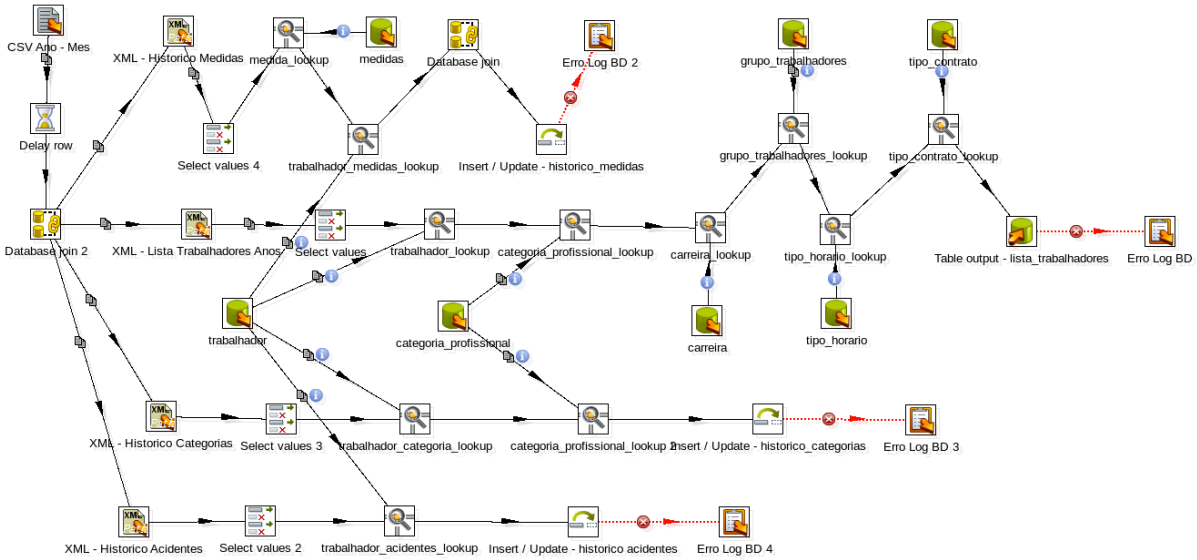

Figura 9: Transformação que possui um *step - delay row*.

Após a inserção de toda a informação proveniente dos *webservices* é necessário tratar a informação. São também criados índices de modo a otimizar as *queries* de carregamento da *Data Mart*.

A última etapa consiste na automatização do processo de ETL, pelo que foi criado um ficheiro Excel que permitirá ao grupo operacional definir quais as rúbricas salariais a utilizar para o cálculo de indicadores de despesa, caso aconteçam mudanças nesses indicadores. A informação desse ficheiro é introduzida na área temporária, de modo a que o carregamento da informação na *Data Mart* sobre os indicadores de despesa seja automático e contabilize as rúbricas salariais definidas.

O tempo total de carregamento de 54 meses (desde janeiro de 2011 até junho de 2015) foi de 2 horas e 40 minutos. O tempo do primeiro carregamento posterior, um mês, foi de 5 minutos e 23 segundos.

5.1.2. Dimensões

Após o carregamento da área temporária estar completo, o passo seguinte é o carregamento das dimensões da *Data Mart*.

As dimensões *rh_d_tempo*, *rh_d_antiguidade* não necessitam de qualquer informação da área temporária, porque só é necessário gerar informação, como é visível na Figura 10, ao contrário de todas as outras dimensões que necessitam de consultar a área temporária. As

consultas efetuadas na área temporária são feitas através de SQL, como mostra o exemplo da Figura 11.

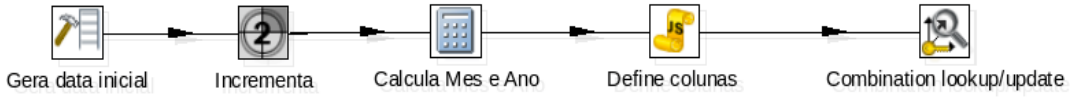

Figura 10: Transformação de carregamento da dimensão *rh_d_tempo*.

```
SELECT DISTINCT regime, tipo_docente as vinculo, grau_deficiencia, eti, estado, dias_trabalho_ano, trabalhador_estudante, 0 as acumulacao,
    tipo_contrato.descricao as "Tipo Contrato", tipo_contrato.agrupador "Agrupador Contrato", tipo_horario.descricao as "Modalidade horario"
FROM temporaria.lista_trabalhadores, temporaria.tipo_contrato, temporaria.tipo_horario
WHERE lista_trabalhadores.id_tipo_contrato = tipo_contrato.id and lista_trabalhadores.id_tipo_horario = tipo_horario.id

UNION

SELECT DISTINCT regime, tipo_docente as vinculo, grau_deficiencia, eti, estado, dias_trabalho_ano, trabalhador_estudante, 1 as acumulacao,
    tipo_contrato.descricao as "Tipo Contrato", tipo_contrato.agrupador "Agrupador Contrato", tipo_horario.descricao as "Modalidade horario"
FROM temporaria.lista_trabalhadores, temporaria.tipo_contrato, temporaria.tipo_horario
WHERE lista_trabalhadores.id_tipo_contrato = tipo_contrato.id and lista_trabalhadores.id_tipo_horario = tipo_horario.id
```

Figura 11: SQL de carregamento da dimensão *rh_d_esp_contrato*.

5.1.3. Tabelas de facto

Depois das dimensões carregadas, prossegue-se com o carregamento das tabelas de factos, que à semelhança das dimensões, utilizam SQL para trazer toda a informação da área temporária. Para cada linha retornada na *query* SQL é necessário verificar as chaves primárias de todas as dimensões que caracterizam a tabela de facto a ser carregada. Esse é o fluxo visível na Figura 12, com o carregamento da tabela de factos *rh_f_admissoes_contratos*.

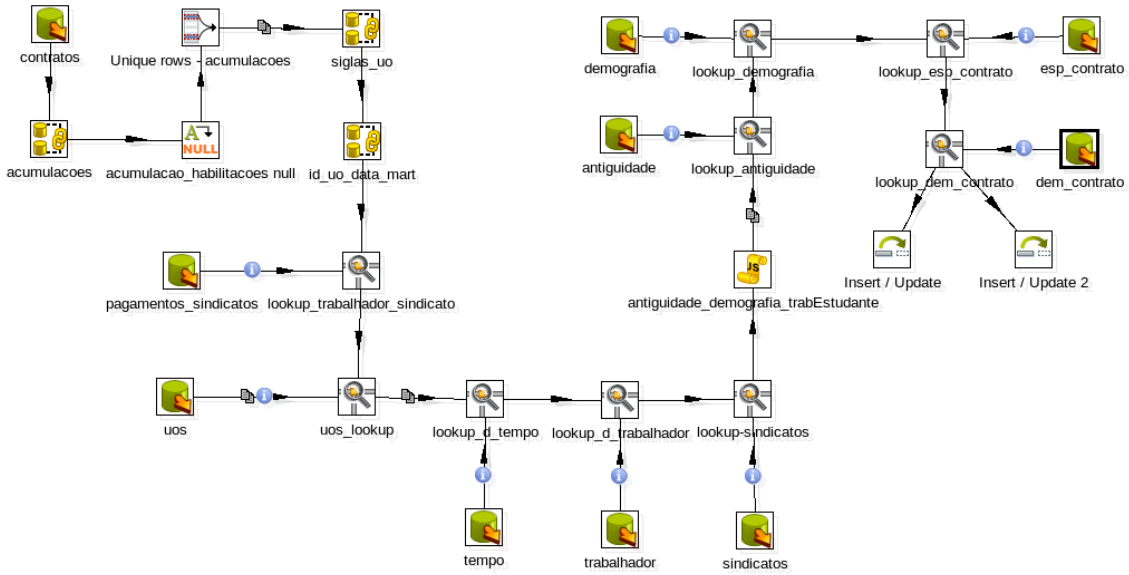

Figura 12: Transformação de carregamento da tabela de factos *rh_f_admissoes_contratos*.

As restantes tabelas de facto carregadas foram: *rh_trabalhadores_ativos, rh_f_suspensoes_saidas* e *rh_f_acidentes*. À exceção da tabela *rh_trabalhadores_ativos*, que é carregada através da tabela *rh_f_admissões_contratos*, todas as restantes utilizam um processo semelhante ao da Figura 12.

O tempo de carregamento de todas as tabelas de facto referidas e de todas as dimensões foi de 1 hora e 6 minutos para 54 meses. O tempo do primeiro carregamento posterior foi de 15 minutos e 46 segundos.

5.2. Cubos

Como foi referido no capítulo 4, a ferramenta para criação do cubo OLAP foi o *Mondrian*, com o intuito de aumentar o desempenho no acesso a grandes quantidades de dados.

Os cubos são definidos através de um esquema criado com recurso à ferramenta da *Pentaho, Schema Workbench*. Cada cubo é definido tendo em conta uma tabela de factos, juntamente com as dimensões que a caracterizam.

Para dar resposta aos indicadores de despesa, com os filtros e agregações pretendidas é necessário efetuar a sua especificação. O primeiro passo foi definir a tabela de factos a utilizar, de seguida definir os atributos necessários para filtrar e agregar trabalhadores, definir medidas através dos factos com recurso a uma função de agregação, e por último definir (se necessário) membros calculados que efetuam operações aritméticas entre duas medidas. A Figura 13 ilustra a definição de um cubo e a criação de uma medida (*min_vencimento_ilíquido*) através da função agregadora MIN (vencimento ilíquido mínimo).

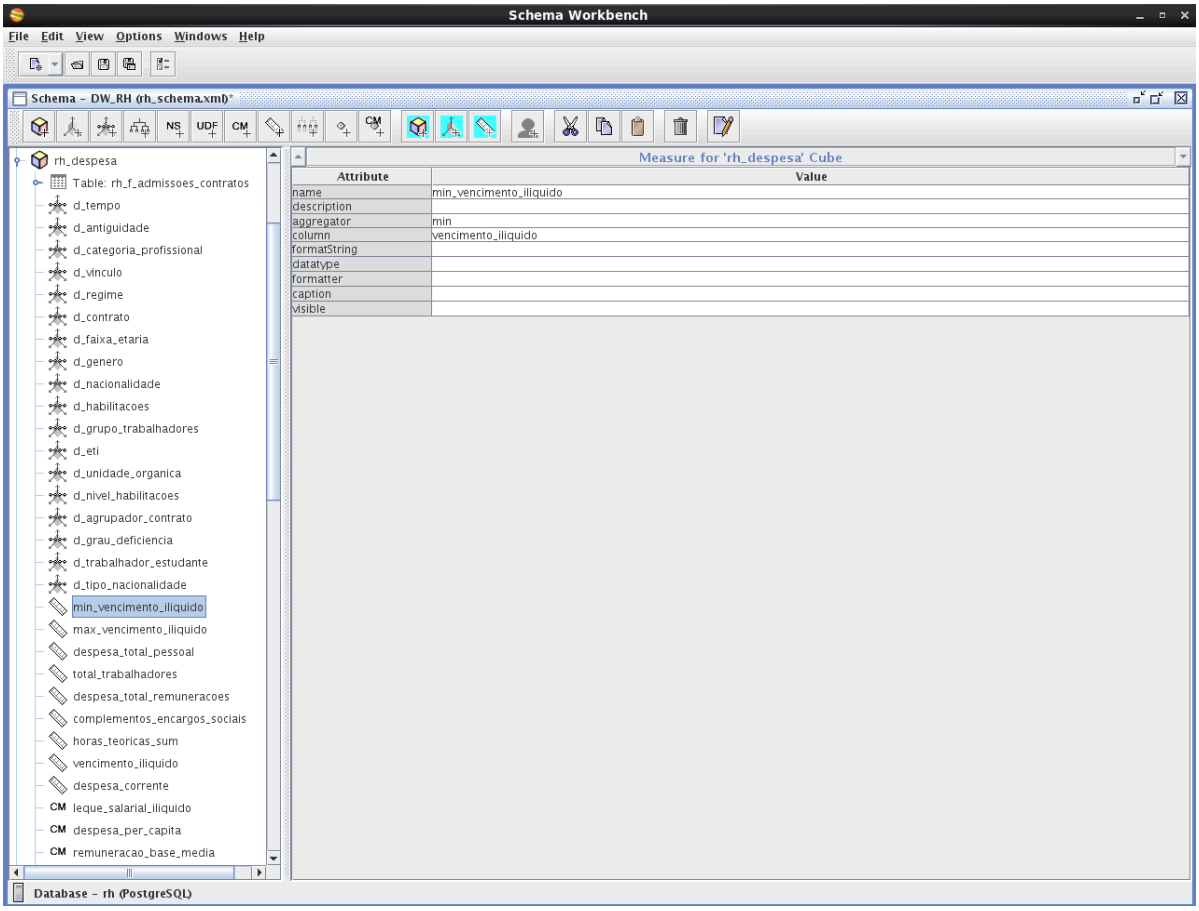

Figura 13: Definição do cubo de indicadores de despesa.

Utilizando este procedimento foram criados na totalidade quatro cubos: contratos, despesa, absentismo e demográficos.

5.3. *Dashboards*

Durante este estágio foram criados cinco *dashboards*, que pretendem fazer a análise dos indicadores especificados através da informação presente na *Data Mart*.

Os *dashboards* diferem entre si, devido à especificação dos indicadores neles existentes, mas todos possuem características comuns:

- *Drill down*: é efetuado clicando no gráfico de *snapshot* (gráfico da direita), sendo assim possível descer na hierarquia da UC.
- *Roll up*: é efetuado clicando no *breadcrums* (barra das unidades orgânicas), para a unidade orgânica em que o utilizador clicou.
- *Slice and dice*: é efetuado através da agregação dos trabalhadores e dos filtros genéricos, grupo de trabalhadores, género, faixa etária, nacionalidade, antiguidade, habilitações literárias, categoria profissional, vínculo, regime, tipo de contrato e percentagem de contrato (ETI).
- Visualização em tabelas: é possível analisar a informação em formato de tabela, clicando no botão com formato de tabela em baixo dos gráficos.
- *Download* da informação: é possível fazer *download* dos gráficos em formato de imagem e das tabelas em formato Excel.
- Informação sobre os indicadores: é possível aceder à ficha do indicador que está a ser analisado clicando no botão com o formato de símbolo de informação.

5.3.1. Caracterização

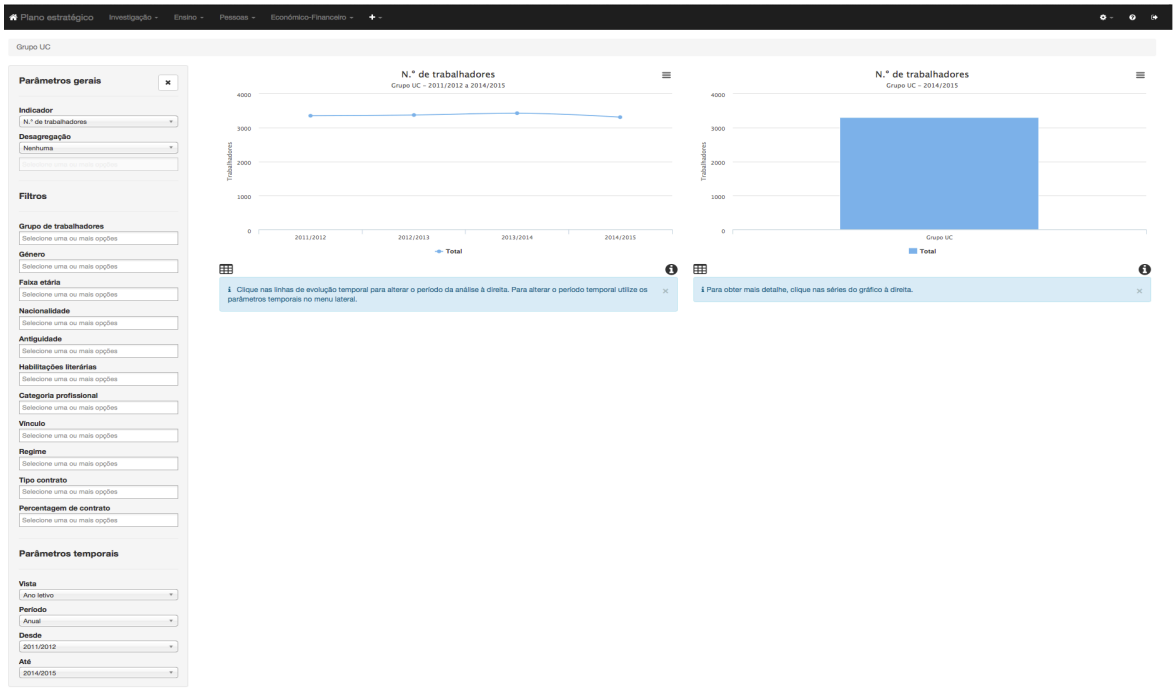

Figura 14: *Dashboard* "Caracterização".

O *dashboard* "Caracterização" aborda quatro indicadores especificados: n.º de trabalhadores, ETIs, n.º de acidentes e antiguidade média. Para a análise de todos os indicadores existe a possibilidade de não ser feita qualquer tipo de agregação, ou de visualizar os trabalhadores por categoria profissional, vínculo, regime ou tipo de contrato. A Figura 14 representa o *dashboard* da caracterização analisando o indicador n.º de trabalhadores.

5.3.2. Despesa

O *dashboard* da despesa com os trabalhadores inclui um leque de dez indicadores, sendo eles: despesa total com pessoal, despesa média mensal total com pessoal, despesa total com remunerações, despesa média mensal total com remunerações, remuneração base média, carga salarial, custo hora, despesa per capita, leque salarial ilíquido e taxa de complementos e encargos sociais. Em todos estes indicadores possuem os mesmo tipos de agregação, mantendo-se igual ao *dashboard* "Caracterização". A Figura 15 ilustra o *dashboard* da despesa com a análise do indicador despesa total com pessoal.

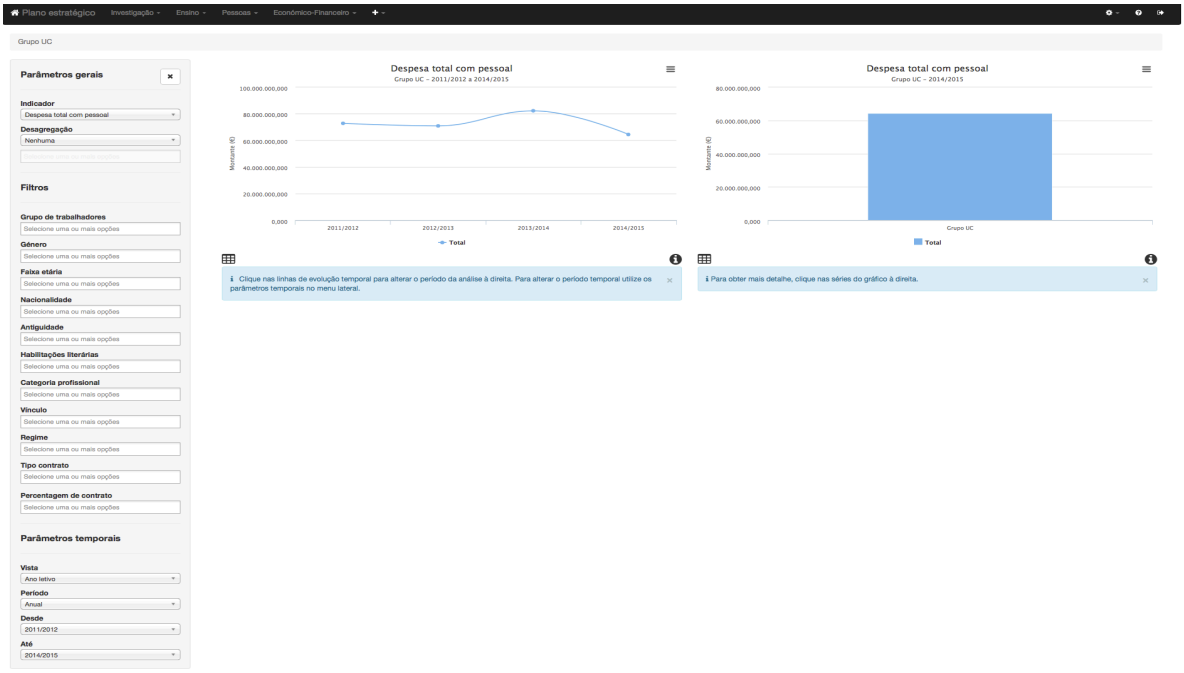

Figura 15: *Dashboard* "Despesa".

5.3.3. Demografia

Este *dashboard* aborda dois indicadores existentes na "Caracterização", n.º de trabalhadores e ETIs. A "Demografia" difere de todos os restantes nos tipos de agregações, sendo possível efetuar a agregação por informações demográficas do trabalhador (género, faixa etária, nacionalidade, tipo de nacionalidade, antiguidade, habilitações literárias e nível habilitacional), difere ainda no número de filtros, com o acréscimo dos filtros: sindicato, trabalhador estudante e deficiência. Na Figura 16, está a ser analisado o indicador n.º de trabalhadores.

Figura 16: *Dashboard* "Demografia".

5.3.4. Absentismo, admissões, suspensões e saídas

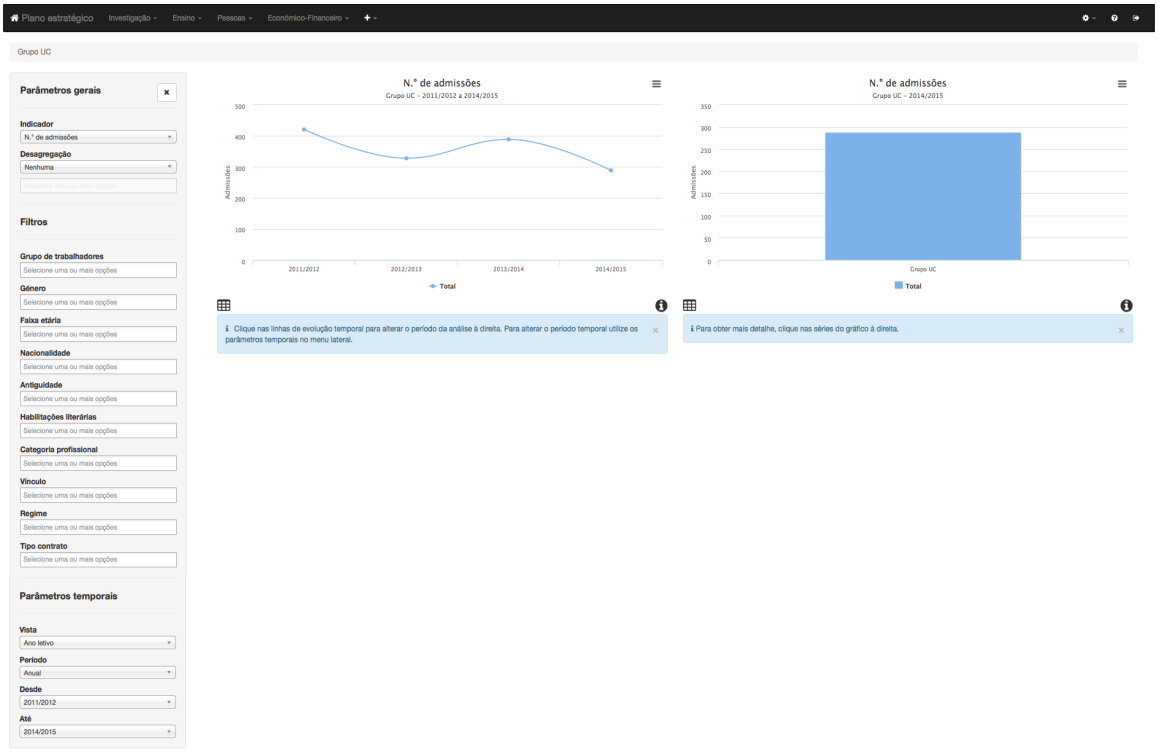

Figura 17: *Dashboard* "Absentismo, admissões, suspensões e saídas".

Este *dashboard* permite uma análise variada onde alguns indicadores possuem formas de agregação diferentes entre si. Os indicadores disponíveis para análise são: nº de admissões, índice de admissão, n.º de saídas, índice de saída, índice de absentismo, índice de reposição e nº de suspensões.

Os indicadores n.º de saídas, índice de absentismo e n.º de suspensões possuem exclusivamente o tipo de desagregação "motivo", enquanto os restantes podem ser analisados sem qualquer tipo de agregação, ou então por categoria profissional, vínculo, regime ou tipo de contrato. A Figura 17 representa o *dashboard* analisando o indicador nº de admissões.

5.3.5. Índices etários

O *dashboard* "Índices etários" analisa 5 indicadores: índice de rejuvenescimento, índice de envelhecimento, índice de emprego jovem, nível etário e leque etário. O *dashboard* não permite qualquer tipo de agregação.

A Figura 18 representa a análise aos indicadores, índice de rejuvenescimento, índice de envelhecimento e índice de emprego jovem.

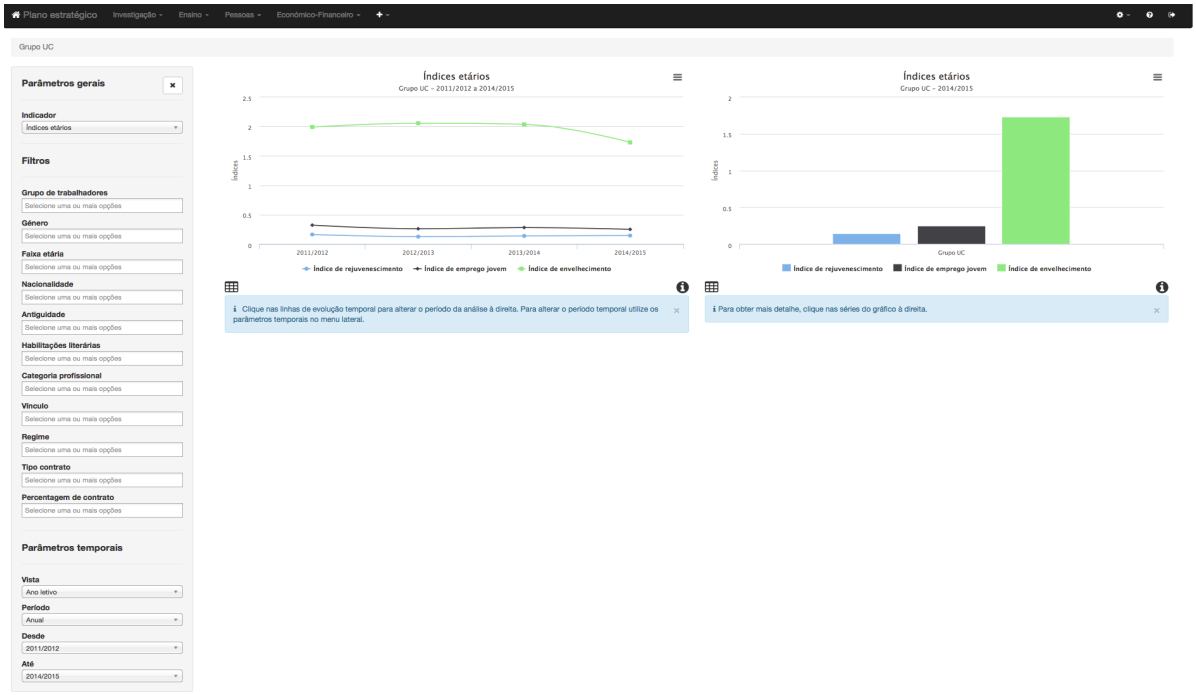

Figura 18: *Dashboard* "Índices etários".

5.4. Desempenho e otimização

Dada a quantidade de dados a analisar foi necessário otimizar as análises efetuadas pela aplicação. Com a intenção de cumprir o requisito RNF_U_002, apresentado no Capítulo 3, foram implementadas cinco soluções de otimização:

1. Índices: em primeiro lugar foram criados índices do tipo *b-tree* para cada coluna das dimensões e para cada chave forasteira das tabelas de facto. Posteriormente foram

criados índices englobando mais do que uma coluna, após analisar as *queries* SQL elaboradas pelo *Mondrian* à BD.

2. *Closure table*: é uma tabela alojada na BD que otimiza a pesquisa de unidades orgânicas feita pelo *Mondrian*. Possui a distância de uma unidade orgânica aos seus filhos e vice-versa. Como exemplo, esta tabela revela que a distância do DEI à FCTUC é de 1 nível e que do DEI à UC é de 2 níveis.
3. Agregados: são tabelas que incluem informação pré-calculada, com um número inferior de registos do que as tabelas de factos. A utilização dos agregados é mais rápida se houver poucos registos na tabela de agregados, razão pela qual foram criados vários agregados para a mesma tabela de facto com diferentes granularidades.
4. Índices sobre agregados: foram criados índices entre várias colunas da tabela de agregados, otimizando análises com vários filtros ativos.
5. *Cache*: o *Mondrian* faz uso de uma *cache* própria, contendo resultados de *queries* anteriores que são utilizados na sua totalidade ou de forma parcial para obter resultados a novas *queries* ou a *queries* que já foram feitas anteriormente.

Capítulo 6
Testes e validação

Neste estágio existiram duas fases de testes, uma no início do estágio, às funcionalidades desenvolvidas no ano letivo anterior, e outra perto do seu término, com o final do desenvolvimento.

6.1. Primeira fase de testes

Optou-se por seguir uma metodologia de testes *black-box*[13] (Figura 19), que consiste em introduzir *inputs*, receber *outputs*, analisar resultados e comparar o *output* obtido com o que seria espectável. Este processo é realizado pelo utilizador, sem ter conhecimento da aplicação.

Figura 19: Imagem alusiva aos testes *black-box*.

Primeiramente foram criados os casos de testes, com o auxílio das fichas de indicadores e junto dos responsáveis pela criação das *dashboards*.

Seguidamente foram efetuadas derivações dos casos de teste, de forma a cobrir todos os requisitos funcionais da aplicação, nomeadamente, as operações de *drill down, roll-up, slice and dice*.

Os casos de testes são identificados de acordo com a Tabela 15.

ID	Descrição	Ação	Resultado Esperado	Resultado Obtido	Validação
TF_mm_cc_yy	Visualizar mais um agregado.	Adicionar o agregado Professor Catedrático Convidado	Os gráficos irão apresentar mais uma linha, e mais uma coluna, respetivamente, à esquerda e à direita.	O resultado esperado.	Validado

Legenda:

- **TF**: Teste Funcional.
- **mm**: Módulo/Área (Recursos Humanos, Académicos e Financeiro).
- **cc**: Categoria (Gerais, Indicadores).
- **yy**: número de teste.

Tabela 15: Tabela exemplificativa da identificação de casos de teste.

Os testes desenvolvidos são relativos ao módulo de RH, e ao *dashboard* de despesa com o ensino, que estão disponíveis no anexo 5.

6.2. Segunda fase de testes

A segunda fase de testes foi em tudo semelhante à primeira. Começou-se por criar um leque de testes funcionais que cobrissem todas as funcionalidades dos *dashboards* e de seguida aplicaram-se os casos de testes registando os seus resultados. Os *dashboards* testados foram os desenvolvidos durante o estágio:

- Caracterização;
- Despesa;
- Demografia;
- Absentismo, admissões, suspensões e saídas;
- Índices etários.

Os testes referidos encontram-se no anexo 6.

6.3. Validação da implementação

Durante o processo de desenvolvimento existiram três momentos onde foram aplicadas diferentes técnicas de validação.

6.3.1. Validação do ETL

Inicialmente, o processo ETL foi executado para o mês de janeiro de 2011, verificando se todos os registos de SAP estavam a ser transferidos para a área temporária e se a integridade dos dados era mantida. Para tal foi utilizada a ferramenta *SoapUI* que permite invocar *webservices* através de uma interface gráfica. Esses resultados foram comparados com os resultados de *queries* SQL realizadas na área temporária.

De seguida, foi verificado se os registos carregados para a *Data Mart* não sofreram introdução de erros e se nas tabelas de facto não existiam chaves forasteiras a *null*. Foi também executado o processo de carregamento da *Data Mart* repetidamente, para garantir que não era introduzida informação duplicada.

6.3.2. Validação dos cubos

Para validação dos cubos especificados foi utilizada a aplicação *Saiku*, incorporada no servidor de BI. O *Saiku* permite listar os factos e as dimensões e também executar *queries* MDX em cada cubo específico.

Através da análise, da lista de factos e dimensões, era possível confirmar se a especificação estava correta. Na execução de *queries* MDX era possível analisar se as funções de agregação estavam bem definidas com base nos resultados obtidos, tendo sempre como padrão de comparação as *queries* SQL.

6.3.3. Validação dos *dashboards*

A validação dos dados apresentados na aplicação foi efetuada através da recolha de um conjunto de casos de modo a construir uma amostra que permitiu comparar o resultado esperado com o fornecido pela aplicação.

Para comparar os resultados foram utilizados os dados da área temporária, usando as mesmas funções de agregação que o *Mondrian* utiliza. Para esta validação foi ainda utilizado o *feedback* fornecido pelo grupo operacional sobre os valores apresentados.

6.4. Validação de requisitos

O processo de validação foi composto por duas fases: a validação da especificação dos indicadores e a validação dos dashboards.

A primeira fase de validação foi realizada com recurso a protótipos, como foi referido no processo de recolha de indicadores, no Capítulo 3. A Figura 20 representa um dos muitos ecrãs desenhados, onde é possível observar o nível de detalhe de interatividade e dos componentes no ecrã, de modo a abordar com objetividade e clareza o indicador no processo de validação.

Nesta fase, toda a especificação dos indicadores foi validada utilizando as fichas de indicadores como forma de aprovação.

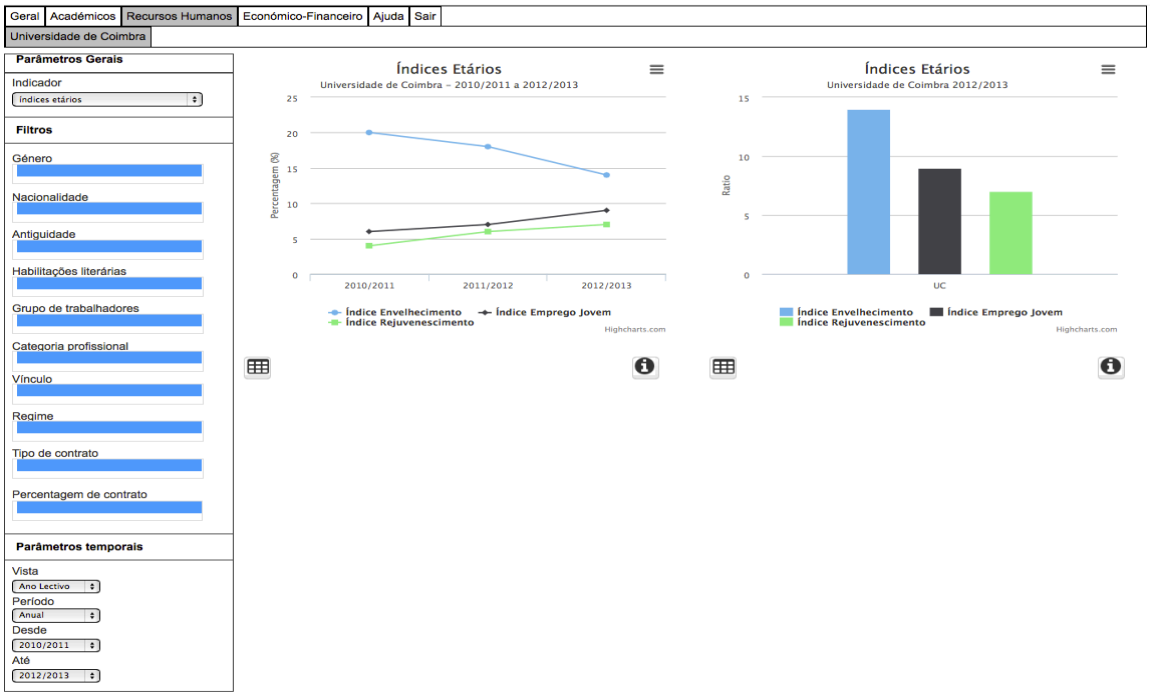

Figura 20: Ecrã exemplo dos indicadores IND_RH_14, 17 e 18, na UC.

A segunda fase foi efetuada através de fichas de validação. Estas fichas incluem um leque de ações a realizar nos *dashboards* pelo responsável do grupo operacional, nelas deve analisar os resultados das ações e responder a questões sobre os requisitos do indicador apresentado, os seus objetivos, a realidade dos valores apresentados e por último a usabilidade.

As fichas de validação realizadas para cada *dashboard* encontram-se no anexo 7.

Capítulo 7
Planeamento

Nesta secção é apresentado o planeamento do estágio, a metodologia utilizada para o seu desenvolvimento, os desvios e imprevistos ocorridos e a análise de riscos efetuada.

7.1. Metodologia *Business Intelligence*

O sucesso da implementação de um projeto de BI depende da integração apropriada das tarefas que o compõem. Na Figura 21 está representado o ciclo de vida de um processo de BI[14] e as suas tarefas, segundo o intitulado pai da *Data Warehouse*, Ralph Kimball.

Algumas destas tarefas já foram concluídas no ano letivo transato, pelos elementos mais antigos da equipa de desenvolvimento, como é o caso da arquitetura do sistema de BI e das ferramentas de implementação a utilizar.

- Planeamento: é o início do ciclo do projeto. Fornece uma maior consistência ao projeto. É necessário conhecer as necessidades de negócio, daí a ligação bidirecional.

- Gestão do projeto: foca-se na monitorização do projeto, de modo a que os objetivos da aplicação sejam alcançados, assegurando que o ciclo de Kimball seja cumprido e as tarefas estejam em conformidade.

- Definição dos requisitos de negócio: é necessário compreender os requisitos do negócio, reunindo com os principais interessados e conhecedores da área, dentro da organização. Uma correta definição dos requisitos irá construir uma base sólida para todo o processo de construção de uma aplicação de BI.

- Design da arquitetura técnica: uma aplicação de BI requer a integração de várias tecnologias, sendo necessário pensar no design da arquitetura do sistema. Para tal deve-se ter em consideração os requisitos do negócio, abstraindo-se das possíveis tecnologias a usar.

- Seleção de produtos e instalação: após o design da arquitetura é necessário escolher os produtos adequados ao desenvolvimento do projeto, passando pelas bases de dados, ferramentas de ETL, servidor de BI e ferramentas de *report* e de criação de análises. Após esta escolha, os produtos deverão ser testados para verificar a capacidade de integração entre eles.

- Modelo dimensional: o modelo de dados tem de ser corretamente construído, focando-se numa análise ao negócio, identificando os factos e a granularidade necessária, bem como os atributos e dimensões.

- Design físico: esta etapa foca-se na definição das estruturas na base de dados, incluindo a sua segurança, o desempenho e agregados.

- Design e implementação ETL: o grande risco e esforço num projeto de BI surge nesta etapa, no processo de extração, transformação e carregamento da informação. Consiste na recolha de informação das fontes de dados, tratamento dos dados, e carregamento dessa informação para a DW.

- Design da aplicação de BI: o design da aplicação é necessário para definir a sua estrutura de acordo com as necessidades do negócio e dos utilizadores bem as suas capacidades.
- Desenvolvimento da aplicação de BI: desenvolvimento e validação da aplicação de BI para análise da informação.
- *Deployment*: as principais etapas convergem aqui, onde finalmente é apresentada a aplicação de BI. Tudo deve ser previamente testado, de forma a validar o correto funcionamento.
- Manutenção: após o sistema estar em produção, são necessárias algumas tarefas de otimização, tais como, criação e manutenção de índices e sistema de *backups*.
- Crescimento: após o correto desenvolvimento do projeto, é um sinal positivo se surgirem novos indicadores de negócio. Para efetuar esse crescimento é necessário realizar uma priorização de acordo com as necessidades do negócio.

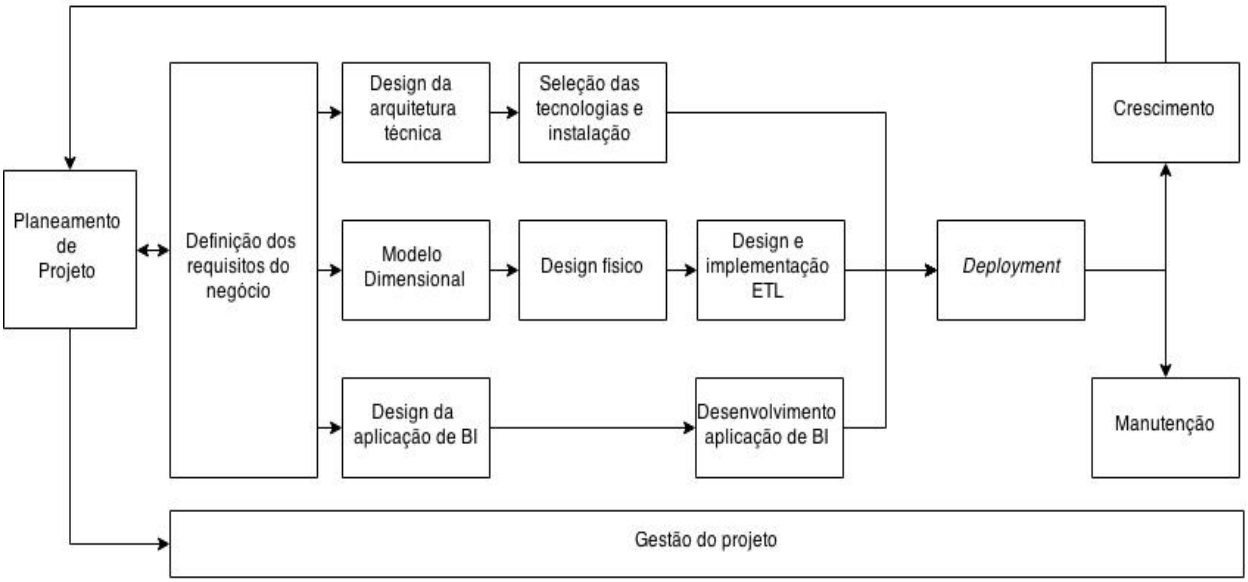

Figura 21: Ciclo de vida de um processo de BI por Ralph Kimball.

7.2. Plano de trabalho

Para a realização deste estágio foi elaborado um plano que consiste no conjunto de cinco etapas: enquadramento no projeto UC-Num, planeamento da *Data Mart* (requisitos e modelo de dados), o processo de ETL, criação do cubo OLAP e por último o desenvolvimento de *dashboards* para incorporar na aplicação.

Para o desenvolvimento da primeira etapa, foi efetuado um plano que consistiu na pesquisa de uma metodologia de testes que fosse aplicada aos módulos desenvolvidos no ano letivo anterior, bem como a criação de casos de teste e a sua realização.

A segunda etapa baseou-se na recolha e análises de indicadores, junto do grupo operacional, bem como a sua validação através de protótipos da aplicação, e de seguida criou-se o modelo de dados multidimensional e o modelo da área temporária.

A terceira etapa consistiu no desenvolvimento da área temporária e do processo de ETL.

Na quarta etapa foi criado o cubo OLAP com recurso ao modelo de dados multidimensional.

Por último, teve lugar a criação das *dashboards*, respetivos testes e a incorporação dos *dashboards* na aplicação.

As primeiras três etapas foram executadas no 1º semestre, do qual resultou o diagrama de Gantt da Figura 22, enquanto que a quarta e quinta etapas, foram desenvolvidas no 2º semestre, e estavam incorporadas num ciclo iterativo, que percorria as prioridades de implementação dos indicadores (Elevada, Média e Baixa), tentando maximizar o número de indicadores com análises disponíveis.

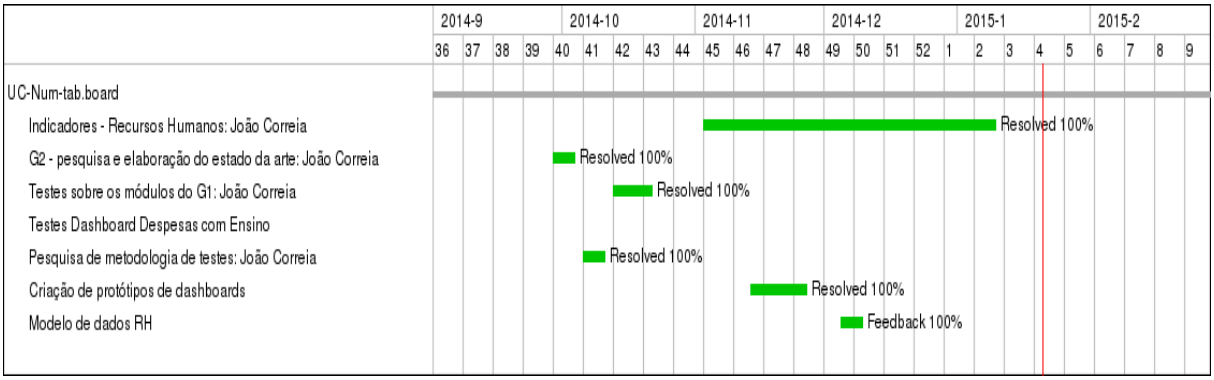

Figura 22: Diagrama de Gantt do trabalho desenvolvido no 1º semestre.

7.3. Trabalho desenvolvido 2º semestre

Era espectável que o processo de desenvolvimento dos indicadores de prioridade elevada fosse completado, mas tal não se verificou. Existiram alguns entraves durante o desenvolvimento, como erros no sistema fonte e nos *webservices*, bem como problemas com a utilização da *framework* de desenvolvimento dada a inexperiência pessoal.

Ao longo do desenvolvimento foram detetados erros no sistema fonte, como por exemplo trabalhadores com a data de nascimento errada, sem data de entrada no organismo, com o histórico de habilitações incompleto e trabalhadores alocados em unidades orgânicas que não existiam. Cada erro encontrado foi reportado ao Serviço de Recursos Humanos da UC, que prontamente os solucionou.

Os *webservices* de especificação das unidades orgânicas da UC retornavam informação incorreta devido a campos não preenchidos em SAP o que revelou a necessidade de refazer o processo de ETL e as *queries* de todos os *dashboards*, o que atrasou o processo de desenvolvimento.

Em suma, foram desenvolvidos 21 indicadores de prioridade elevada, tendo sido impossível a implementação de 8 indicadores. Foram ainda desenvolvidos 2 indicadores de prioridade média por se revelarem pouco consumidores de tempo. A Figura 23 representa o diagrama de Gantt do trabalho desenvolvido no 2º semestre.

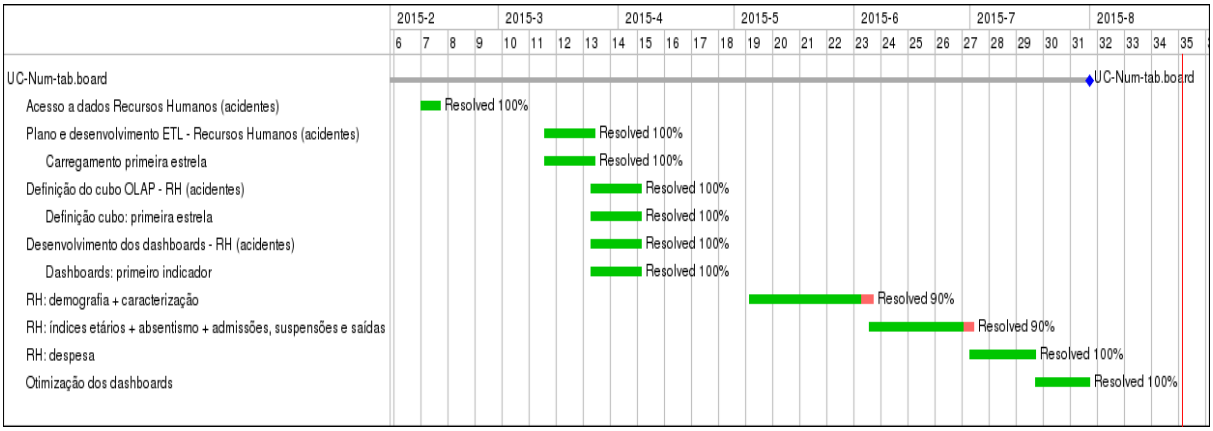

Figura 23: Diagrama de Gantt do trabalho desenvolvido no 2º semestre.

7.4. Metodologias e ferramentas de desenvolvimento em equipa

A equipa está divida segundo os diferentes módulos da DW, em desenvolvimento. Os membros com mais experiência auxiliam os estagiários no desenvolvimento, podendo ser considerados mentores.

Sendo este um estágio de continuidade e a decorrer em paralelo com a criação de outros módulos, há que seguir certas normas que são determinantes para o bom funcionamento da equipa.

Primeiramente, foram definidas quais as ferramentas em uso no projeto. A principal plataforma utilizada é o *Redmine*, uma plataforma *web* para gestão de projetos. O *Skype* juntamente com o email, são ferramentas fundamentais para comunicação tanto entre os elementos da equipa como com os *stakeholders* da UC.

Para manter acessíveis ficheiros e documentos criados durante o desenvolvimento do projeto é utilizado o *Git* para versionamento de ficheiros.

Estão assim partilhados documentos de especificação de design das *dashboards*, *templates* para a criação de vários tipos de documentos, bem como todos os documentos produzidos nos diferentes módulos da DW. Muitos destes documentos têm como intuito manter a coerência de criação de *dashboards* e de documentos, como por exemplo as fichas de indicadores e testes de casos de uso.

Na Figura 24 encontra-se uma representação da estrutura do repositório usado no projeto para o desenvolvimento dos diversos módulos. Este repositório contém cinco pastas principais:

- BD: pasta com os modelos de dados e os *scripts* SQL necessários para os diferentes módulos. Encontra-se subdividida em pastas de cada módulo.
- ETL: contêm os processos de ETL, encontrando-se subdividida em pastas de cada módulo.
- *pentaho-solutions*: encontram-se todos os *dashboards* a ser carregados pelo servidor de BI do *Pentaho*. Encontra-se, também, subdividida em pastas de cada módulo.
- Documentos: contêm toda a documentação do projeto UC-Num.
- *static*: contêm os recursos estáticos necessários para a visualização dos *dashboards*.

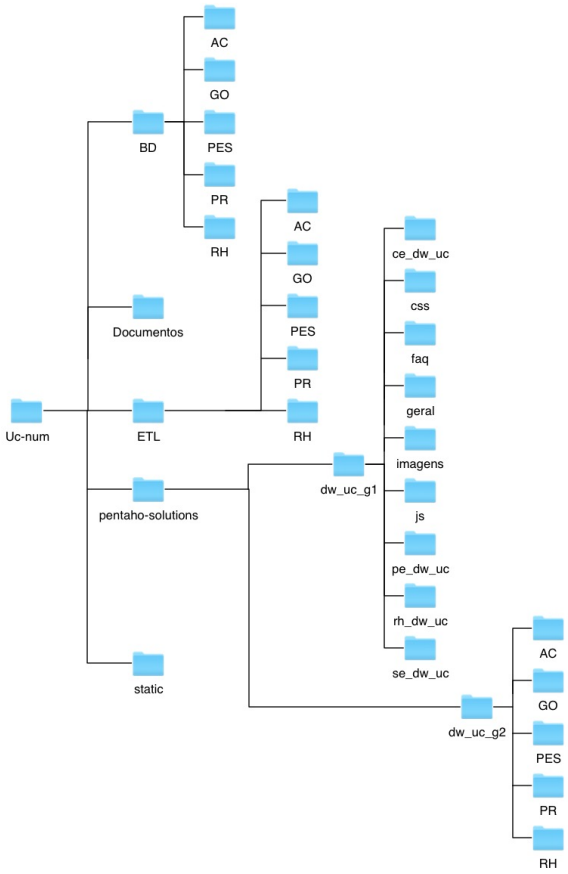

Figura 24: Estrutura do repositório *Git*.

7.5. Análise de riscos

A análise de riscos é uma estratégia fundamental em qualquer projeto de criação de *software*, principalmente num de média/grande dimensão como o projeto da *Data Warehouse* na Universidade de Coimbra. A identificação de riscos começou numa reunião de equipa onde todos os membros de desenvolvimento, inclusive os estagiários estiveram presentes e pró-ativos.

A identificação e análise de riscos é importante porque só assim se conseguem delinear estratégias de mitigação caso os riscos aconteçam. Este processo foi uma mais-valia porque possibilitava um percurso alternativo que otimizava as tomadas de decisão e consequentemente o tempo associado a cada tarefa.

Para a identificação de riscos foi criada a Tabela 16, onde cada linha identifica um risco. Cada risco possui associado um código único, uma designação, um tipo que revela se o risco é proveniente da equipa ou exterior a esta, a categoria onde o risco se enquadra (técnico ou recurso), a probabilidade do seu acontecimento, o seu impacto no desenvolvimento, uma descrição mais detalhada, o plano de contingência caso o risco aconteça, o seu estado (se já aconteceu ou ainda não) e por último a data da sua identificação. A tabela foi construída com recurso à informação de riscos disponível no website OpenUP[26].

Código	**Designação**	**Tipo**	**Categoria**	**Probabilidade**	**Impacto**	**Descrição**	**Plano de contingência**	**Estado**	**Data de Identificação**
Risco_001	Atraso na especificação dos indicadores	Indireto	Recurso	Alta	Médio	Atraso na conclusão de toda a especificação de indicadores.	Iniciar desenvolvimento incremental, de acordo com os indicadores especificados até à data. Monitorização atenta e constante dos trabalhos desenvolvidos pelas equipas operacionais.	Ocorreu e foi aplicado plano de contingência.	25/02/2015
Risco _002	Disponibilização dos dados fonte	Indireto	Recurso	Alta	Alto	Disponibilização não atempada dos dados fonte.	Análise completa de todas as necessidades dos sistemas fonte (exemplo: documentação com todos os atributos necessários). Iniciar desenvolvimento à medida que vão sendo disponibilizados dados (independentemente da prioridade do requisito). Solicitar dados fictícios enquanto os reais não estão disponíveis.	Ocorreu e foram aplicados o primeiro e segundo pontos do plano de contingência.	25/02/2015
Risco _003	Conhecimento das tecnologias utilizadas	Direto	Recurso	Média	Alto	Falta de conhecimento dos elementos da equipa nas ferramentas utilizadas para o desenvolvimento.	Auxílio constante dos elementos da equipa de desenvolvimento; Realização de *workshops*; Disponibilização e consulta de material bibliográfico.	Ocorreu e foi aplicado plano de contingência.	25/02/2015

Risco_004	Limitações das tecnologias	Indireto	Técnico	Baixa	Alto	Problemas de implementação das tecnologias utilizadas.	Atualizar ou reverter a versão das tecnologias. Corrigir os problemas no código fonte da tecnologia, só se aplica a tecnologias *open-source*.	Ainda não ocorreu.	25/02/2015
Risco_005	Validação da aplicação	Indireto	Recurso	Alta	Alto	Atrasos na validação da aplicação, pela equipa destacada para tal.	Disponibilização atempada da aplicação. Acompanhar o estado do processo de validação juntamente com a equipa destacada.	Ocorreu, sendo aplicado o segundo método de mitigação.	25/02/2015
Risco_006	Especificação de indicadores inválida	Indireto	Recurso	Média	Alto	Erros na especificação dos indicadores (fórmulas de cálculo, agregações, filtros, parâmetros temporais) cometidos pela equipa de trabalho.	Múltiplas validações da especificação de indicadores, efetuadas por diferentes *stakeholders*.	Ocorreu e foi aplicado plano de contingência.	25/02/2015
Risco_007	Saída de elementos da equipa de desenvolvimento	Direto	Recurso	Baixa	Alto	Saída de elementos do projeto, causando instabilidade, atrasos.	Reorganizar o planeamento das tarefas e afetação dos responsáveis. Renegociação dos prazos com o cliente.	Ainda não ocorreu.	25/02/2015

Tabela 16: Tabela de identificação de riscos.

Capítulo 8
Conclusão

O trabalho proposto para este estágio e a temática envolvida no projeto é bastante aliciante, sendo o BI uma área cada vez mais conhecida e desenvolvida.

Este estágio foi uma experiência extremamente interessante, devido à consolidação dos conhecimentos na área de BI e de desenvolvimento de *software*, bem como o conhecimento adquirido na área de RH.

As reuniões que decorreram com os *stakeholders* de RH foram muito enriquecedoras, porque trabalhar com pessoas de áreas diferentes ou engenheiros informáticos que trabalham com ferramentas como SAP ou o NÓNIO é sempre uma mais-valia, bem como poder partilhar a minha visão sobre os indicadores que, sendo mais pragmática, pôde ajudar a desbloquear situações de conflito na especificação de indicadores.

Sinto que o meu contributo neste projeto também passou pelo melhoramento dos dados no sistema SAP, onde detetei informação errada e que foi corrigida pelos Serviços de Recursos Humanos da UC.

Foi um trabalho desgastante, que consumiu muito tempo durante o desenvolvimento, testes e otimização mas o balanço geral foi muito positivo. Todo o esforço e dedicação foi recompensado com um convite para permanecer no projeto e dar continuidade ao módulo de RH.

O próximo passo será iniciar o desenvolvimento dos indicadores de prioridade elevada que já foram especificados e que não foram desenvolvidos em tempo útil. Todo o conhecimento adquirido vai permitir construir uma solução mais sólida e com melhor desempenho.

Anexos

[1] Análise de Tecnologias

Bases de Dados

Ambos os motores, referidos no Capítulo 2, funcionam nas principais plataformas, sendo as principais características[15][16] referidas nas seguintes tabelas.

	Plataformas Disponíveis	**Controlo de Acessos**	**Funcionalidades**	**Capacidades da Base de Dados**
MySQL 5.6	Solaris Unix IBM AIX Windows Mac OS X Linux FreeBSD	*Native Network Encryption* *Path Acess* *Run unprivileged*	*ACID* *Java Support* *Referential Integrity* *Transactions* *Unicode*	*Blobs and Clobs* *Inner Joins* *Inner Selects* *Merge Joins* *Outer Joins* *Union*
PostgreSQL 9.3	Windows Mac OS X Unix BSD IBM AIX Solaris HP/UX IRIX	*Audit* *Brute-force Protection* *Enterprise Directory Compatibility* *Password Complexity Rules* *Seurity Certification* *Native Network Encryption* *Path Acess* *Run unprivileged*	*ACID* *Java Support* *Referential Integrity* *Transactions* *Unicode* *High Availability* *Highly Scalable* *Import Data* *Multi-Core Support* *Parallel Processing* *Real Time Access to Database*	*Common Table Expressions* *Except* *Intersect* *Parallel Query* *Windowing Functions* *Blobs and Clobs* *Inner Joins* *Inner Selects* *Merge Joins* *Outer Joins* *Union*

Tabela (anexo) 1: Comparação entre *MySQL* e *PostgreSQL* - principais características.

	Vistas e Tabelas	**Índices**	**Tipos de Dados**
MySQL 5.6	Vistas	*B-Tree* *Hash* *R-Tree*	***Integer***: BIGINT (64-bits), INTEGER (32-bits), MEDIUMINT (24-bits), SMALLINT (16-bits), TINYINT (8-bits) ***Floating Point***: DOUBLE (64-bit), FLOAT (32-bits) ***Decimal***: DECIMAL, NUMERIC ***String***: CHAR, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, VARCHAR, SET, ENUM ***Boolean***: BOLEAN ***Binary***: BINARY, LONGBLOB, BLOB, MEDIUMBLOB, TTINYBLOB, VARBINARY ***Date/Time***: DATETIME, YEAR, DATE, TIME, TIMESTAMP ***Others***: SPATIAL, OPENGIS
PostgreSQL 9.3	Vistas	*B-Tree* *Hash* *GIN* *GiST* *SP-GiST*	***Integer***: BIGINT, INTEGER, SMALLINT, BIGSERIAL, SERIAL SMALLSERIAL ***Floating Point***: DOUBLE PRECISION, REAL ***Decimal***: NUMERIC, DECIMAL ***String***: CHARACTER, CHAR CHARACTER VARYING, VARCHAR TEXT ***Boolean***: BOLEAN, BIT ***Binary***: BYTEA ***Date/Time***: INTERVAL, DATE, TIME e TIMESTAMP (com ou sem *time zone*) ***Others***: PSEUDO TYPES, XML, JSON, NETWORK ADDRESS TYPES ENUM, GEOMETRIC TYPES

Tabela (anexo) 2: Comparação entre *MySQL* e *PostgreSQL* – vistas, índices e tipos de dados.

	Limites da Base de Dados				
	Tamanho máximo *Blob/Clob*	**Tamanho máximo da BD**	**Tamanho máximo por tabela**	**Tamanho máximo por linha**	**Número máximo de colunas por linha**
MySQL 5.6	4 GB	Ilimitado	• 256 TB (MyISAM) • 64 TB (InnoDB)	64 KB	4096
PostgreSQL 9.3	• 1GB (texto, bytea) • 4 TB (pg_large object)	Ilimitado	32 TB	1.6 TB	250-1600 (depende to tipo de dados)

Tabela (anexo) 3: Comparação entre *MySQL* e *PostgreSQL* – limites de tamanhos.

ETL

Após uma pesquisa[17][18][19] e uma breve utilização de cada uma das ferramentas são notórias algumas diferenças.

Ambas as ferramentas, desenvolvidas em Java, funcionam nas principais plataformas (Windows, Mac OS X e Linux).

O *Kettle* possui uma interface gráfica mais intuitiva e um fluxo de construção do processo de ETL mais simples e natural. Por outro lado, o *JasperETL* tem os processos de ligação e manutenção das bases mais otimizados, inclusive possui algoritmos que permite lidar com *Slowly Changing Dimensions.*

A forma de carregar e transformar os dados é diferente em cada uma das ferramentas. O *Kettle* tem processos mais genéricos (*Input, Output, Transform*, etc.), enquanto o *JasperETL* tem processos mais específicos, como por exemplo a ligação às Bases de Dados é diferenciada pelo motor de BD (*MySQL, PostgreSQL,* Oracle*,* ...).

Ambos permitem *scripting* em *Java, JavaScript, SQL e Shell. JasperETL* permite ainda *scripts Groovy*.

Um projeto elaborado no *Kettle* consiste num conjunto de transformações, mas é permitido a criação de *jobs*, onde é possível agregar um conjunto de transformações.

O *JasperETL* tem uma ferramenta de *debugging*, que permite acompanhar o processo em tempo real, enquanto o Kettle possui uma ferramenta mais básica. Ambas conseguem fazer ligações às bases de dados não relacionais e a *software* ERP proprietário, como por exemplo*,* SAP.

Em suma, ambas as ferramentas de ETL mencionadas são opções válidas para qualquer projeto de uma *Data Warehouse*, dando sempre preferência ao *Kettle* pela quantidade e qualidade de informação.

Aplicações *Business Intelligence*

Nesta secção do anexo foram analisadas superficialmente soluções pagas de duas das mais conceituadas empresas, de modo compreender as suas características e principalmente o seu custo.

Oracle – num só pacote é possível encontrar todas as aplicações necessárias numa solução de BI que permita uma ótima análise de negócio.

- *Business Intelligence Foundation Suite*[20]

SAP – Ao contrário da Oracle, a SAP não disponibiliza um pacote de aplicações de modo a englobar todas as que são necessárias à organização compradora. Sendo assim, comparativamente ao pacote da Oracle, seria necessário adquirir as seguintes soluções:

- *Lumira*[21]
- *BusinessObject Analysis*[22]
- *Cristal Reports*[23]
- *BusinessObjects Web Intelligence*[24]
- *BusinessObjects Design Studio*[25]

A Tabela (anexo) 4 compara características entre duas soluções *open-source* e outras duas soluções pagas.

	JasperReports Server	*Pentaho BI Server*	*Oracle Business Intelligence Foundation Suite*	*SAP* (conjunto 4 aplicações)
Criação de *Dashboards*	Não	Sim	Sim	Sim
Criação de relatórios e análises OLAP	Sim	Sim	Sim	Sim
Dispositivos móveis	Sim	Sim	Sim	Sim
Custo	*Open-source*	*Open-source*	• NUP (mínimo 25): 2901€ cada licença • Primeiro ano (única vez por licença): 638,15€ por NUP *Processor*: 165753€ **(preços para 5 anos)**	• *Lumira:* 995 USD$ • *BusinessObject Analysis*: (desconhecido) • *Cristal Reports*: 495 USD$ • *BusinessObjects Web Intelligence:* (desconhecido) • *BusinessObjects Design Studio:* (desconhecido) **(preços perpétuos, de uma licença)**

Legenda: NUP – licença atribuída a uma pessoa autorizada a usar o produto, instalado num ou mais servidores. Um dispositivo que utilize ativamente o produto também acresce do uso de uma licença deste tipo. *Processor* - uma licença para cada processador onde os produtos da Oracle são instalados.

Tabela (anexo) 4: Comparações entre quatro soluções de BI.

Os restantes anexos mencionados serão fornecidos em formato digital, juntamente com este documento.

[2] Fichas de indicadores

[3] Modelos de dados – *Data Mart*

[4] Modelos de dados – Área temporária

[5] Testes – Primeira fase

[6] Testes – Segunda fase

[7] Fichas de validação

Referências

[1] Coronel, C., Morris, S., Rob, P., *Database Systems: Design, Implementation, and Management*, 9ª edição. Cengage Learning, 2011

[2] Kimball, R., Ross, M., 2013. *The Data Warehouse Toolkit – The Definitive Guide to Dimensional Modeling*, 3ª edição. John Wiley and Sons, Inc., Indianapolis.

[3] Plano Estratégico da Universidade de Coimbra para os anos 2011-2015, http://www.uc.pt/planeamento/PEA_2011_2015_out2012.pdf acedido em 25/09/14.

[4] *MySQL*, Web Site oficial *MySQL,* http://www.mysql.com/ acedido em 27/09/14.

[5] *PostgreSQL*, Web Site oficial *PostgreSQL,* http://www.postgresql.org/ acedido em 27/09/14.

[6] *NoSQL*, Web Site oficial da comunidade *NoSQL*, http://nosql-database.org/ acedido em 29/09/14.

[7] Kimball, R., Caserta, J., 2004. *The Data Warehouse ETL Toolkit – Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data.* Wiley Publishing, Inc., Indianapolis.

[8] *JasperSoft*, Web Site oficial do *JasperReports Server* http://community.jaspersoft.com/project/jasperreports-server acedido em 02/10/14.

[9] *Pentaho BI Server*, http://community.pentaho.com acedido em 17/09/14.

[10] *Mondrian*, Documentação *Mondrian,* http://mondrian.pentaho.com/documentation/ acedido em 3/10/14.

[11] *Microsoft Corporation*, 2012. *Multidimensional Expressions (MDX) Reference.* SQL Server 2012 Books Online.

[12] Bouman, R., Dongen, J., 2009 *Pentaho Solutions - Business Intelligence and Data Warehousing with Pentaho and MySQL.* Wiley Publishing, Inc., Indianapolis, Indiana.

[13] Patton, R., 2005. *Software Testing*, 2ª edição, Sams Publishing, Estados Unidos da América.

[14] Kimball, R., 2008. *The Data Warehouse Lifecycle Toolkit*, 2ª edição. John Wiley and Sons.

[15] *PostgreSQL*, Manual *PostgreSQL* http://www.postgresql.org/files/documentation/pdf/9.3/postgresql-9.3-A4.pdf acedido em 27/09/14.

[16] *MySQL*, Manual *MySQL* http://dev.mysql.com/doc/refman/5.6/en/ acedido em 27/09/14.

[17] *Pentaho*, Documentação *Pentaho Data Integration* http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation acedido em 29/09/14.

[18] Pulvirenti, A., Roldán, M. 2011. *Pentaho Data Integration 4 Cookbook*. Packt Publisting Ltd., Birmingham.

[19] *Talend*, Componentes *Talend* http://www.talendforge.org/components/index.php acedido em 30/09/14.

[20] Oracle, *Business Intelligence Foundation Suite* http://www.oracle.com/us/solutions/business-analytics/business-intelligence/foundation-suite/overview/index.html acedido em 3/10/2014.

[21] SAP, SAP *Lumira* http://www.sap.com/portugal/pc/analytics/business-intelligence/software/data-visualization/index.html acedido em 3/10/2014.

[22] SAP, SAP *BusinessObject Analysis OLAP* http://www.sap.com/portugal/pc/analytics/business-intelligence/software/web-based-analysis-olap/index.html acedido em 3/10/2014**.**

[23] SAP (2014), SAP *Cristal Reports* http://www.sap.com/portugal/pc/analytics/business-intelligence/software/crystal-reports/index.html acedido em 3/10/2014.

[24] SAP, SAP *BusinessObjects Web Intelligence* http://www.sap.com/portugal/pc/analytics/business-intelligence/software/web-intelligence/index.html acedido em 3/10/2014.

[25] SAP, SAP *Design Studio* http://www.sap.com/portugal/pc/analytics/business-intelligence/software/design-studio/index.html acedido em 3/10/2014.

[26] *OpenUP*, Análise de riscos http://epf.eclipse.org/wikis/openup/core.mgmt.common.extend_supp/guidances/concepts/risk_AF5840DA.html acedido em 25-02-2015.