Mestrado em Engenharia Informática

Estágio

Relatório Final

Projeto DW-UC

Desenvolvimento de uma *data warehouse* para a Universidade de Coimbra

Área A

Beatriz Paiva Fragoso

bfragoso@student.dei.uc.pt

Orientador:

Prof. Dr. Bruno Cabral

01 de julho de 2014

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

Agradecimentos

Ao orientador Prof. Doutor Bruno Cabral pela oportunidade de integrar um projeto tão inovador e de elevada importância. Ao co-orientador Eng. Pedro Pinto e restantes elementos da equipa do NONIO por todo o auxílio prestado. A todos os colegas estagiários pela cooperação.

À minha família, em especial à minha mãe e irmãos, que foram sem dúvida, a base de tudo o que alcancei até hoje. Ao meu namorado pelo apoio, nos bons e maus momentos. Às Mondeguinas pelo carinho e amizade. A todas as residentes do piso 3 da Pólo II-2 pelo companheirismo. Aos meus amigos, pela sinceridade e dedicação.

Resumo

A realidade com a qual a reitoria, conselho de gestão, diretores de unidades orgânicas e coordenadores de curso da Universidade de Coimbra se depara diariamente, dificulta uma gestão competitiva. A universidade define internamente, no plano estratégico estabelecido até 2015, um conjunto de indicadores de desempenho sobre a atividade de ensino. Estes indicadores permitem uma avaliação e monitorização da instituição. O problema é que para obter o valor desses indicadores é necessário esperar dias, por vezes semanas. São calculados manualmente, razão pela qual estão sujeitos a erros, agravando ainda mais a situação.

O objetivo do presente estágio é eliminar a espera e os lapsos cometidos no cálculo manual dos indicadores relacionados com a atividade de ensino, fornecendo uma análise de dados fiável, a qualquer momento que esta seja necessária.

A solução passa por criar um *data mart* (sub conjunto de uma *data warehouse* com dados de um âmbito específico) para armazenamento dos dados necessários ao cálculo dos indicadores na área do ensino, entre os quais os custos da atividade de ensino num curso e custo médio por aluno. Através do modelo multidimensional utilizado no *data mart* é possível aceder a grandes quantidades de dados de forma rápida e eficaz, o que é bastante vantajoso para, posteriormente, ser produzida uma análise OLAP (*Online Analytical Processing*) sobre a informação armazenada, através de *dashboards* interativos.

A análise é disponibilizada através de uma aplicação web. Esta plataforma permite uma gestão prática e confiável, uma monitorização de custos e uma poupança de recursos, aumentando o conhecimento e performance da gestão ao nível do ensino.

Palavras-Chave

Indicadores de gestão, Custos, Ensino, Análise de dados, *Business Intelligence, Data Warehouse, Dashboards*

Índice

Lista de Figuras

Lista de Tabelas

Glossário

BD	Base de dados
BI	*Business Intelligence*
CDE	*Community Dashboards Editor*
CDF	*Community Dashboards Framework*
CSS	*Cascading Style Sheets*
CSV	*Comma Separated Values*
DW	*Data Warehouse*
ETL	*Extraction, Transforming and Loading*
GSIIC	Gestão de Sistemas e Infra-estruturas de Informação e Comunicação
HTML	*HyperText Markup Language*
JS	*JavaScript*
KPI	*Key Performance Indicator*
LDAP	*Lightweight Directory Access Protocol*
MDX	*Multidimensional Expressions*
NONIO	Sistema de gestão académica
OLAP	*Online Analytical Processing*
SAMA	Sistema de Apoio à Modernização Administrativa
SAP	Sistema de suporte à gestão financeira
SAS	*Statistical Analysis System*
SI	Sistema de informação
SO	Sistema operativo
SOAP	*Simple Object Access Protocol*
TIC	Tecnologias de informação e comunicação
UC	Universidade de Coimbra
UO	Unidade orgânica
URL	*Uniform Resource Locator*
XML	*Extensible Markup Language*

Capítulo 1
Introdução

1.1. Enquadramento

A crescente necessidade das instituições gerirem os seus recursos, a um nível quase diário, faz com que os seus responsáveis precisem de conhecer e ter acesso a indicadores de desempenho (KPIs). A Universidade de Coimbra está incluída no role dessas instituições. Cada vez mais o reitor, vice reitores, elementos do conselho de gestão, diretores das unidades orgânicas e coordenadores de cursos – identificados como os principais *stakeholders* - mostram necessidade de ter acesso a um conjunto de indicadores que os auxilie na tomada de decisão, a qualquer momento que esta seja necessária.

O projeto DW-UC integra um outro projeto da UC – SAMA (Sistema de Apoio à Modernização Administrativa, ver Figura 1) - que pretende melhorar as suas infraestruturas e serviços TIC em áreas bem definidas da universidade: infraestruturas de suporte a serviços, integração e extensão de sistemas de gestão, integração com plataformas de contratação pública, indicadores para a gestão, sistemas de pagamentos electrónicos e monitorização da estratégia da UC e do seu desdobramento. Inclui-se nos objetivos do projeto SAMA, a criação de instrumentos de gestão e monitorização de indicadores de desempenho da UC. O projeto DW-UC enquadra-se totalmente neste objetivo e está atualmente dividido em cinco áreas: projetos de investigação, sucesso escolar, recursos humanos, custos com o ensino e receita com propinas e emolumentos; áreas definidas para desenvolvimento até julho de 2014.

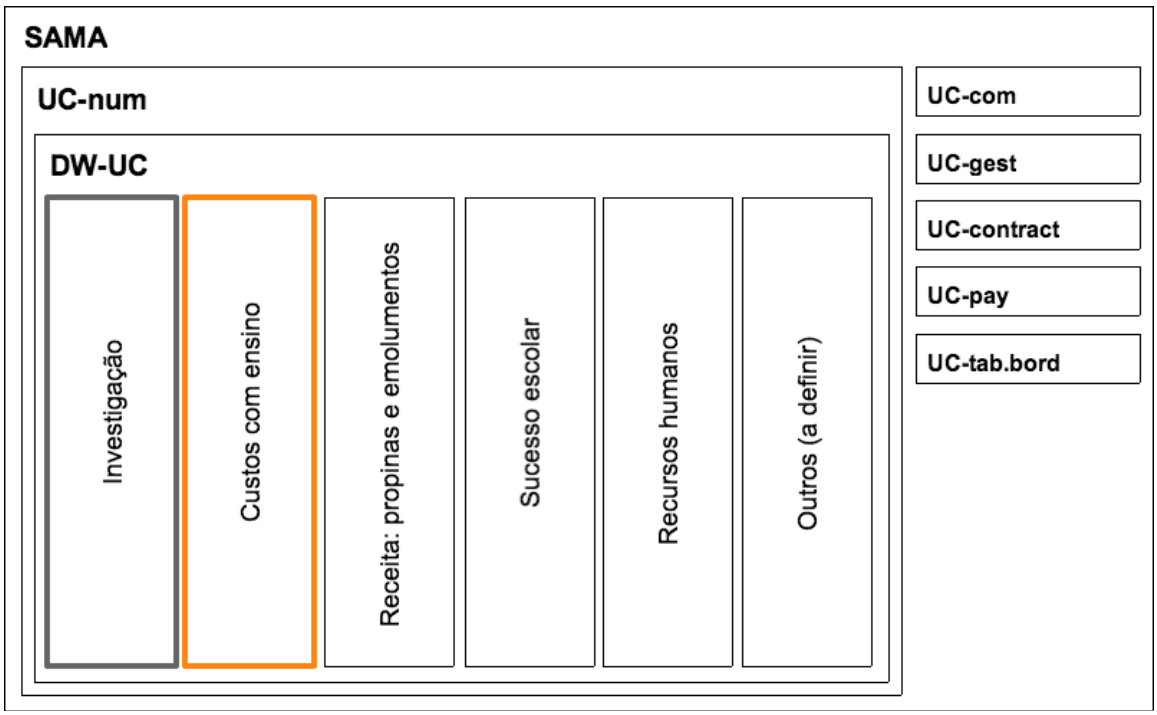

Figura 1 - Enquadramento do projeto DW UC no SAMA

A área dos projetos de investigação foi alvo de análise e execução por parte de uma equipa de três elementos durante cerca de seis meses no ano letivo anterior. Encontra-se de momento em fase de testes e validação. As restantes áreas foram analisadas e trabalhadas este ano em quatro propostas de estágio, sendo que o presente estágio integra uma dessas

propostas e diz respeito aos **custos com a atividade de ensino na UC**. Desta fazem parte indicadores de grande interesse para a equipa reitoral e têm grande destaque no Plano Estratégico e de Ação da UC até 2015, destacando-se o custo médio por aluno na UC.

A equipa atual é composta por quatro elementos e dois orientadores, o Prof. Doutor Bruno Cabral e o Eng. Pedro Pinto da equipa do sistema NONIO; a área do sucesso escolar é abrangida por este sistema, por esta razão o local de trabalho da equipa é na infraestrutura disponibilizada pelos responsáveis do NONIO.

1.2. Contexto atual

No capítulo que se segue é exposta a situação atual da UC no que respeita a análises de dados, gestão e monitorização de KPIs direcionados à atividade de ensino.

Os principais sistemas de informação, presentes nas infraestruturas de toda a UC, são o SAP e o NONIO, sendo que ambos contêm dados essenciais para a gestão financeira e académica de todos os recursos da universidade.

No que respeita a análise de dados, o NONIO apresenta um conjunto de estatísticas, bastante simples, com dados da gestão académica. É uma análise que não permite qualquer manipulação dos dados ou interatividade, estando diretamente vocacionada a indicadores de gestão e coordenação de cursos. Não integra indicadores relacionados com custos dos cursos, estes presentes no plano estratégico da UC até 2015.

O sistema SAP inclui, também, um módulo que permite gerar análises sobre os dados desse mesmo sistema, contudo, está obsoleto e não é utilizado por nenhum serviço da UC. Obter indicadores partindo de uma única fonte de dados também não é suficiente e não expressa a realidade da instituição.

Apesar das elevadas capacidades de cada um, nenhum dos atuais sistemas consegue atender à carência apresentada pelos principais *stakeholders*: obter indicadores (KPIs) relacionados com o custo a atividade de ensino na UC, de forma cómoda, rápida e fidedigna.

A solicitação de KPIs, por entidades externas à UC, é frequente. É o exemplo do custo médio por aluno por parte do secretário de estado do ensino superior. O que acontece todos os anos é que os responsáveis têm de promover um conjunto de recursos humanos para efetuar o cálculo necessário. Esta tarefa exige tempo e como consequência de ser uma tarefa humana manual pode introduzir erros (tal como resume a Figura 2).

Identificam-se neste ponto dois aspetos fundamentais: não existe nenhum sistema que concentre informação de diversas fontes e áreas da UC, consequentemente não existe forma de fornecer KPIs, a qualquer momento, aos vários *stakeholders*.

Figura 2 – Processo atual na UC para obtenção de indicadores

Dada esta inexistência e incapacidade dos atuais sistemas, fica claro que o processo atual, para que os principais *stakeholders* tenham acesso a informação relativa a custos com a atividade de ensino é praticamente manual e que, por essa razão, torna-se bastante moroso. No que respeita à tomada de decisão a questão do tempo é suprema, logo a UC pode estar em desvantagem por não conseguir ter acesso a dados pertinentes, de forma clara, no momento oportuno.

Na secção seguinte são abordados e clarificados os objetivos do estágio.

1.3. Objetivos

Para solucionar a desvantagem apresentada na secção anterior, os objetivos principais do presente estágio são:

- Construir uma *data warehouse* que agregue e armazene dados de várias fontes e áreas – no caso do custo com o ensino, da área académica e financeira (NONIO e SAP);
- Disponibilizar esses dados através de uma análise gráfica, intuitiva e interativa (OLAP) – no caso do presente projeto é pretendida uma aplicação web para facultar esta mesma análise.

Na área dos custos com o ensino, os objetivos mencionados devem responder, nomeadamente, às seguintes questões:

- Qual o custo total de uma unidade orgânica num determinado ano letivo? A evolução desse custo, ao longo do tempo, tem um comportamento regular?
- Qual a evolução do custo total de um curso nos últimos anos?
- Qual o custo médio por aluno num curso?
- Qual(ais) o(s) curso(s) que representa(m) maior custo para a unidade orgânica a que estão afetos, tendo em conta o nº de alunos inscritos em cada curso? E o(s) que representa(m) menor custo?

Estas questões sugerem um conjunto de indicadores de gestão de extrema importância para a UC: custo médio por aluno, custo de um curso, custo em docência na UC ou num curso específico. A evolução destes custos representa uma visão de grande valor para a equipa reitoral, conselho de gestão, entre outros; este aspeto é importante na área da gestão,

cenários do passado podem influenciar decisões presentes que influenciam diretamente o futuro.

Os dois objetivos fulcrais deste estágio colocam este projeto no âmbito do BI (*Business Intelligence*). É uma área recente e ainda em evolução, muitas vezes associada a sistemas de suporte à decisão. O principal objetivo é utilizar um conjunto de informação e apresentá-la de forma útil, intuitiva e de fácil acesso e manuseio. Qualquer projeto de BI divide-se em três grandes etapas:

1. Identificação dos dados fonte necessários;
2. Recolha, transformação e carregamento dos dados (também conhecido como processo ETL, inclusive considerada a fase mais complexa e morosa na construção de uma DW);
3. Por fim, a apresentação da informação aos utilizadores finais - análise OLAP (analisar informação através de um modelo multidimensional de dados, também conhecido como cubo OLAP).

O mercado nesta área tem evoluído ao longo dos últimos anos. Relativamente às instituições de ensino não é conhecido que exista um produto no mercado direcionado, especificamente, para indicadores relativos à atividade de ensino dentro da instituição. O mais comum é existirem soluções de BI em áreas gerais como: gestão financeira e contabilidade, saúde, transportes, **educação**, etc., que, quando adquiridas, são sempre ajustadas a cada caso específico (software à medida). É indiscutível a qualidade e completude da maioria das soluções, contudo, adquirir software à medida apresenta custos.

Dada a conjuntura atual e realidade económica das instituições de ensino, a aquisição de novos produtos não é uma solução praticável, constituindo mais um objetivo do estágio a utilização de ferramentas e tecnologias gratuitas.

A UC é uma instituição de ensino superior de prestígio, a sua gestão, principalmente ao nível da atividade de ensino, é de extrema importância para manter o estatuto alcançado. Os elementos da reitoria e órgãos de gestão precisam de conhecer a realidade dos números, para melhor gerirem e controlarem todo o orçamento e custos da instituição. É esta realidade que o projeto DW UC pretende fazer chegar a todos os *stakeholders*.

1.4. Estrutura do relatório

Nos capítulos seguintes é apresentada de forma mais detalhada cada uma das fases de um projeto no âmbito do BI, quais os conceitos e processos para a construção da DW.

O capítulo 2 descreve o processo de levantamento de requisitos, especifica todos os requisitos funcionais e não funcionais.

O capítulo 3 apresenta toda a especificação de arquitetura e modelos de dados.

No capítulo 4 são apresentados alguns detalhes de implementação, com grande destaque para o desenvolvimento do processo ETL.

O capítulo 5 expõe os processos de validação e testes efetuados.

No capítulo 6 é abordado o trabalho desenvolvido, o planeamento e metodologia utilizada na gestão e desenvolvimento do projeto.

No capítulo 7 encontram-se algumas conclusões relativamente ao projeto e ao trabalho realizado, bem como referência ao trabalho futuro.

Capítulo 2
Requisitos

Nesta secção é apresentado o processo de levantamento de requisitos e exposta toda a análise e especificação de requisitos efetuada.

2.1. Levantamento de requisitos

O levantamento de requisitos foi realizado junto dos principais *stakeholders*: o reitor da UC – Prof. Doutor João Gabriel Silva, a vice reitora Margarida Mano, os elementos da gestão financeira da UC – diretor Sérgio Vicente e técnico superior Carlos Aguiar, o chefe de divisão do planeamento, gestão e desenvolvimento Filipe Rocha e o consultor externo, Dr. José Morais.

Com todos eles foram efetuadas reuniões presenciais. Numa primeira fase com o reitor da UC para perceber qual o objetivo primordial do projeto, de seguida, com a gestão financeira para entender o funcionamento da UC ao nível dos custos. E na fase final, de validação dos requisitos, com a divisão de planeamento, gestão e desenvolvimento e com a vice reitora Margarida Mano.

Durante o processo de levantamento de requisitos podem ser geradas ambiguidades e interpretações erradas do que é pretendido para a aplicação final, inclusive os próprios *stakeholders* podem não ter ideia clara do que pretendem. Para eliminar o risco e o custo de efetuar alterações aquando o desenvolvimento, foi utilizada a técnica de prototipagem rápida para definição e especificação de requisitos – criação de ecrãs exemplo do que será o resultado final da aplicação, com elevada flexibilidade de alteração de acordo com as indicações dos *stakeholders*, funcionando como base durante o desenvolvimento.

O protótipo rápido foi a base de toda a validação, transmitiu aos intervenientes quais os indicadores e funcionalidades a ser contempladas na aplicação final. A sua utilização é uma prática comum e permitiu eliminar quaisquer dúvidas ou diferentes perspetivas que existiram entre as partes.

2.2. Especificação de requisitos

Um dos modelos que é utilizado para especificação de requisitos em engenharia de software é o modelo FURPS+. É um modelo bastante simples, que contempla as principais características a ser tidas em conta aquando a definição dos requisitos de qualquer sistema.

Os requisitos da aplicação estão divididos em duas categorias: funcionais (**F**URPS+) e não funcionais (F**URPS+**), uns estão diretamente relacionados com as funcionalidades da aplicação e outros caracterizam a aplicação quanto à sua qualidade, respetivamente.

Por forma a definir quais os requisitos que devem efetivamente ser cumpridos, sem comprometer o funcionamento da aplicação final, é possível em engenharia de software priorizá-los. Dado o levantamento de requisitos efetuado junto dos *stakeholders*, a equipa acordou com os mesmos três níveis de prioridades: elevado, médio e baixo. Requisitos que caso não sejam cumpridos comprometem a concretização da aplicação têm prioridade elevada. Têm prioridade média os requisitos que, quando cumpridos, acrescentam valor à aplicação final e que, quando não cumpridos, não comprometem de maneira nenhuma o

bom funcionamento da aplicação. Requisitos de prioridade baixa acrescentam valor à aplicação, mas só devem ser cumpridos caso o *budget* (tempo) do projeto permita.

Nas subsecções seguintes é exposta uma apresentação sucinta dos requisitos e protótipo. Em anexo são facultados dois documentos com descrição detalhada:

- DOC_ESPECIFICACAO_PROTOTIPOS_10-01-2014.pdf (anexo [5])*;*
- DOC_REQUISITOS_24-06-2014.pdf (anexo [8]).

Ainda relativamente à especificação dos requisitos, estes são identificados através de um código previamente definido, para melhor estruturação e posterior referência. A codificação segue os seguintes critérios:

Código	**Descrição**
RF_*xx*_00	Requisito funcional, onde *xx* corresponde à categoria e 00 ao número; GE – categoria "Gerais"; IN – categoria "Indicadores" Exemplo: RF_GE_01, requisito funcional número um da categoria "Gerais".
RFN_*y*_00	Requisito não funcional, onde *y* representa a categoria e 00 o número; Categorias: U – usabilidade; R – fiabilidade; P – performance; S – manutenção e suporte; O – outros (restrições de implementação, design e interface, *hardware*, ...).

Tabela 1 - Codificação requisitos funcionais e não funcionais

2.2.1. Requisitos funcionais

No que respeita às funcionalidades da aplicação, estas encontram-se categorizadas por **gerais** e **indicadores**.

Como a própria designação sugere, os gerais dizem respeito a funcionalidades comuns a toda a aplicação - Tabela 2 e Tabela 3.

Código	Prioridade	Designação	Descrição sucinta[1]
RF_GE_01	Elevada	Autenticação	A aplicação deve permitir ao utilizador a autenticação através das credenciais utilizadas no acesso a quaisquer serviços disponibilizados à comunidade da UC (email da UC e *password*).
RF_GE_02	Elevada	Fechar sessão	O utilizador pode terminar a sua sessão.
RF_GE_03	Elevada	Término de sessão	Para garantir segurança da aplicação, um utilizador, depois de autenticar-se terá associada uma sessão, esta deve ter um *timeout* para efetuar *logout* automaticamente.
RF_GE_04	Média	Navegação entre módulos	O utilizador deve conseguir aceder a todos os restantes módulos a qualquer momento.
RF_GE_05	Elevada	Navegação interna	A aplicação deve permitir ao utilizador efetuar *drill down* e *roll up* nos dados que pretende visualizar. Os níveis devem ser os seguintes, do mais alto para o mais baixo: 1. UC; 2. Unidades orgânicas na UC; 3. Departamentos na UO; 4. Ciclos de estudo na UO; 5. Cursos no ciclo de estudo; 6. Unidades curriculares no curso; 7. Docentes na unidade curricular.
RF_GE_06	Elevada	Parâmetros gerais	O utilizador deve ter disponível os diversos parâmetros que são permitidos aplicar sobre a análise.
RF_GE_07	Elevada	Parâmetros de tempo	Deve ser permitido ao utilizador modificar e aplicar os parâmetros temporais. Deve ser possível escolher o período de regime curricular: anual, semestral ou trimestral, associado às unidades curriculares a ter em conta no cálculo do indicador a ser apresentado.
RF_GE_08	Baixa	Esconder parâmetros	Deve ser permitido ao utilizador esconder a barra onde se encontram os parâmetros gerais e de tempo.

Tabela 2 - Visão geral requisitos funcionais: gerais (1)

[1] Consultar descrição de requisitos funcionais em mais detalhe no anexo [8].

[2] Consultar descrição de requisitos funcionais em mais detalhe no anexo [8].

Código	Prioridade	Designação	Descrição sucinta[2]
RF_GE_09	Elevada	Secção de ajuda	A aplicação deve disponibilizar uma secção de ajuda ao utilizador.
RF_GE_10	Elevada	Informação auxiliar	Cada vista de dados disponibilizado ao utilizador deve ser acompanhado de informação referente aos dados que são apresentados.
RF_GE_11	Elevada	Visualização: gráfico ⇔ tabela	A aplicação deve permitir ao utilizador visualizar a informação apresentada num gráfico em formato de tabela e vice-versa.
RF_GE_12	Baixa	Exportar informação na tabela	Exportar para formato *Excel* ou CSV a informação presente nas tabelas de análise.

Tabela 3 - Visão geral requisitos funcionais: gerais (2)

Os requisitos da categoria dos indicadores estão diretamente relacionados com os indicadores de desempenho sobre os custos com a atividade de ensino, que foram identificados como úteis para a tomada de decisão, e que devem ser disponibilizados aos utilizadores finais. Na Tabela 4 encontra-se uma breve especificação de cada um.

Código	Prioridade	Designação	Descrição sucinta[2]
RF_ IN_01	Elevada	Visão geral UC	A aplicação deve permitir ao utilizador obter uma visão geral dos custos com o ensino no orçamento total aprovado para a UC, num ano letivo. Sobre o custo com o ensino pode ser aplicado filtro por parcelas como parâmetro de agregação.
RF_ IN_02		Custo total	Para cada indicador deve ser possível efetuar *drill down* e *roll up* segundo os níveis apresentados no RF_GE_05. Cada um tem definidos parâmetros de agregação, desagregação e de tempo que podem ser aplicados sobre os dados apresentados nos ecrãs (gráficos e tabelas). Para a maioria, são apresentadas análises de evolução temporal e vista atual (*snapshot*).
RF_ IN_03		Custo médio por aluno	
RF_ IN_04		Custo médio e custo médio, por hora, em docentes	
RF_ IN_05		Dados do custo com docentes (demografia)	

Tabela 4 - Visão geral requisitos funcionais: indicadores

[2] Consultar descrição de requisitos funcionais em mais detalhe no anexo [8].

Na Figura 3 encontra-se um exemplo de um dos muitos ecrãs desenhados para o protótipo do módulo do custo com o ensino. É possível observar o grau de detalhe com o qual o protótipo foi construído, tanto ao nível do design como da interatividade dos componentes e ecrãs. O objetivo é que este seja o mais próximo possível da aplicação final, permitindo clareza e objetividade na validação de requisitos entre os intervenientes.

Figura 3 - Ecrã exemplo: Custo total por departamentos na FCTUC

2.2.2. Requisitos não funcionais

Os requisitos não funcionais estão categorizados segundo o modelo utilizado, FURPS+: usabilidade (U), fiabilidade (R), performance (P), manutenção e suporte (S) e outros (+) - Tabela 5. Para o presente projeto, só existiu necessidade de especificar requisitos não funcionais nas categorias de suporte (S) e outros (+).

Como já referido neste documento, este projeto é um módulo que integra uma aplicação conjunta final, pelo que, os requisitos não funcionais têm de ser transversais a todos os módulos. Por esta razão, estes foram definidos conjuntamente com os responsáveis dos módulos que se encontram a ser desenvolvidos atualmente, em paralelo – sucesso escolar, recursos humanos e receita de propinas e emolumentos.

Código	Prioridade	Designação	Descrição sucinta[3]
RFN_S_01	Elevada	Atualização de dados	Processo ETL, carregamento e atualização da DW e cubo OLAP devem ser automáticos.
RFN_S_02	Elevada	Compatibilidade (*browser*)	Aplicação web é compatível com os *browsers* mais modernos: *Internet Explorer 9; Firefox 20* ou superior e *Safari 6* ou superior.
RFN_S_03	Média	Compatibilidade (SO)	Existem algumas restrições também quanto ao sistema operativo, são suportados oficialmente, os SO a partir de: *Windows 7* e distribuições de *Linux*.
RFN_S_04	Elevada	Licenças	A aplicação é desenvolvida e disponibilizada através de software gratuito.
RFN_O_01	Elevada	Hardware	As máquinas a utilizar devem apresentar as seguintes características mínimas: 4*Gb* de RAM, 20*Gb* 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 no software que foi selecionado para desenvolvimento e disponibilização da aplicação.

Tabela 5 - Visão geral requisitos não funcionais: manutenção, suporte e outros

2.3. Sumário

O levantamento e análise de requisitos é uma tarefa essencial em qualquer projeto de desenvolvimento de software. Não é trivial compreender o que é pretendido pelos utilizadores, quais as funcionalidades, restrições e formatos que estes esperam que a aplicação final contemple.

A fase de levantamento de requisitos foi acompanhada pelo desenho de um protótipo rápido, que teve como objetivo despistar quaisquer possíveis mal entendidos entre quem desenvolve a aplicação e os utilizadores finais. O protótipo e respetiva documentação serviram como canal de comunicação entre ambos.

Para além de funcionalidades e interatividade, o protótipo permitiu perceber quais os indicadores relativos a custos com a atividade de ensino, que são efetivamente úteis, e a que granularidade estes podem ser calculados.

[3] Consultar descrição detalhada de requisitos não funcionais – manutenção, suporte e outros – no anexo [8].

Durante o primeiro período de estágio esta fase foi a mais trabalhada e detalhada. Foram efetuadas várias validações do protótipo junto dos *stakeholders*, inclusive foi-lhes fornecido acesso a toda a especificação, para que fosse possível conhecerem em detalhe quais os objetivos e compromissos a que se propõe quem irá implementar a aplicação.

Clarificados e definidos os requisitos foi possível avançar para a etapa seguinte: desenho e especificação da arquitetura.

Capítulo 3
Arquitetura

A arquitetura geral do sistema a desenvolver é apresentada neste capítulo, seguida da análise e seleção de tecnologias para cada um dos componentes do sistema. São também detalhados o processo ETL e os modelos de dados.

3.1. Arquitetura global

Na Figura 4 encontra-se o modelo da arquitetura geral do sistema desenvolvido. A arquitetura divide-se em três componentes principais:

1. *Data Warehouse* (extração dos dados fonte, processamento e transformação dos mesmos e construção da DW);
2. Servidor BI (cubo OLAP e construção da aplicação com análise);
3. Disponibilização da análise, aos utilizadores, através de uma aplicação web.

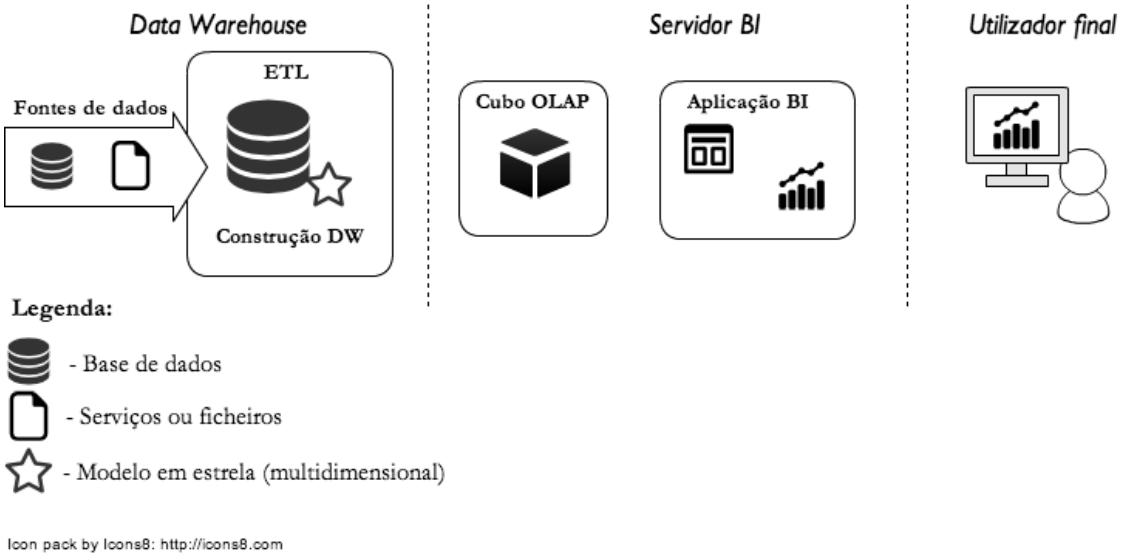

Figura 4 - Arquitetura de alto nível do sistema

O processo ETL é o componente do sistema que envolve maior complexidade. Inicialmente, é efetuada a recolha e extração das diferentes fontes de dados. De seguida, esses dados são sujeitos a um conjunto de transformações, nomeadamente no que respeita a formatação, uniformização e cálculo de agregados necessários. Na fase final do processo os dados são armazenados na DW para posterior consulta. Todo este processo encontra-se detalhado na secção 4.1. Plano do processo de extração, transformação e carregamento.

Após o carregamento dos dados para a DW, estes serão carregados para o que se define como o cubo OLAP. O conceito de cubo surge associado ao modelo multidimensional ou "em estrela" da DW, é como uma metáfora visual para o modelo: cada célula representa o facto e os limites envolventes as dimensões. No limite, cada estrela do modelo da DW será representada por um cubo. A grande vantagem deste modelo está na performance das consultas efetuadas.

Ainda no servidor de BI, encontra-se o desenvolvimento de todo o *front end* da aplicação web que é disponibilizada aos utilizadores, representada pelo último componente da arquitetura apresentada.

Na Figura 5 está apresentado o fluxo de dados, no contexto geral, o que permite mostrar como os diversos componentes comunicam entre si.

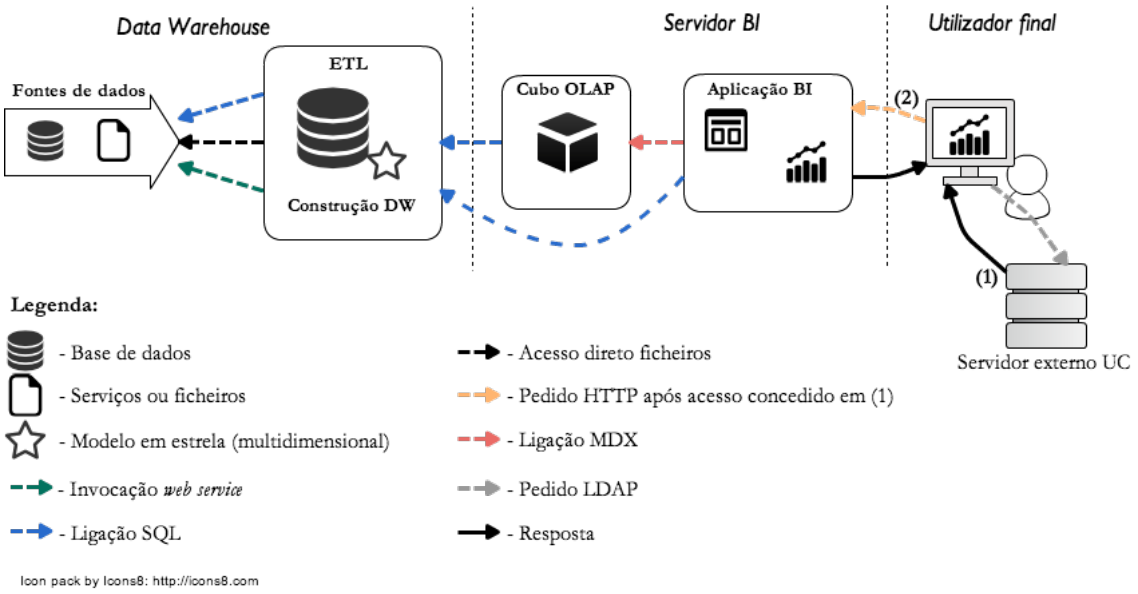

Figura 5 - Arquitetura alto nível, fluxo de dados

Para efetuar a recolha de dados das respetivas fontes são necessários três tipos de comunicação distintos: ligação a bases de dados para obter dados do serviço NONIO, invocação de *web services* desenvolvidos para aceder a informação de SAP e o acesso direto a ficheiros (atualmente apenas necessário para obter o orçamento do ano para o ensino, atividades estruturais e de suporte da UC, presente no relatório anual de contas).

Numa segunda fase, para que seja possível ao esquema do cubo conhecer o modelo multidimensional presente na DW, é efetuada também uma ligação em SQL à base de dados da DW. Após a definição e construção do esquema do cubo OLAP, o componente respeitante à análise efetua consultas em MDX (*multidimensional expressions* – linguagem direcionada aos modelos multidimensionais) sobre o mesmo, pelo que existirá a comunicação direta através deste. Mantém-se a ligação SQL por questões técnicas da tecnologia utilizada no componente servidor BI, pois este necessita de saber qual a base de dados associada ao esquema.

O utilizador terá acesso à aplicação através de pedidos HTTP, último componente da arquitetura apresentada, pois esta é disponibilizada através da web. Tal como mostra a Figura 5 antes do acesso ser concedido o utilizador deverá autenticar-se no servidor LDAP da UC.

3.2. Tecnologias

Como se pode observar pela arquitetura apresentada existem três componentes distintos para os quais foi necessário selecionar ferramentas adequadas à sua manutenção e/ou

desenvolvimento. São eles: os motores de bases de dados (BD da área temporária[4] e da DW), ferramentas para o processo ETL e tecnologia para desenvolver e criar a aplicação web que apresenta a análise OLAP.

Dois aspetos a considerar e que foram tidos em conta na seleção das diversas ferramentas:

- O *budget* inicial para o projeto não prevê quaisquer parcelas para aquisição de software, a seleção fica assim restringida a tecnologias com licença gratuita;
- Como o projeto DW UC teve um módulo a ser desenvolvido no ano letivo anterior – projetos de investigação – a generalidade das ferramentas já se encontrava selecionada à data do estágio, funcionando esta análise e seleção como uma aprendizagem pessoal para conhecer o que é exequível atualmente e o que as organizações dispõem e utilizam (mais soluções de BI existentes no mercado no anexo [2]).

3.2.1 Bases de dados

No que respeita ao armazenamento dos dados, considera-se que este contém duas etapas:

1. os dados fonte são carregados para uma área temporária, neste caso é uma BD com modelo relacional;
2. após o processo ETL é criada uma *data warehouse,* em termos práticos é uma BD com um modelo multidimensional, implementado recorrendo a tecnologias relacionais.

Dado que todo o armazenamento é efetuado em bases de dados é necessário selecionar um motor de base de dados. Os dois motores de base de dados relacionais *open source* mais utilizados são:

- *PostgreSQL – "The world's most advanced open source database"*[28];
- *MySQL (Oracle) – "The world's most popular open source database"*[29].

No anexo [1] é apresentada uma análise com as principais características destas duas bases de dados, ambas *open source*. Apesar do *MySQL* ser a "base de dados mais popular" o *PostgreSQl* tem evoluído bastante nos últimos anos, nomeadamente no que diz respeito ao aumento de performance e usabilidade, daí assumir-se como a "base de dados livre mais avançada"[28].

O mercado e as tecnologias ao nível das bases de dados também tem vindo a evoluir para o que se define como bases de dados não relacionais (*NoSQL*). Pensadas e desenvolvidas para ultrapassar algumas limitações das relacionais como: escalabilidade, replicação e dados não estruturados. No presente já existem diversos tipos de bases de dados *NoSQL*, os mais conhecidos são[31]:

- Armazenamento chave-valor, este é o tipo mais básico das bases de dados *NoSQL*, faz corresponder uma determinada chave a um valor, normalmente a estrutura utilizada são as *hashtables* – ex.: *DynamoDB, MemCacheDB.*
- Armazenamento em colunas, são vistas como um tipo mais avançado das bases de dados chave-valor, pois permite que uma chave mapeie dados de diversas colunas – ex.: *Cassandra.*
- Bases de dados em documentos, o seu funcionamento é semelhante às bases de dados em colunas, contudo permitem armazenamento dos dados em formatos como *XML, JSON,* etc., os documentos – ex.: *MongoDB.*

[4] Todos os dados armazenados entre as fontes de dados (sistemas operacionais) e a DW.

- Bases de dados em grafo, utilizadas para armazenar modelos de dados em rede, utilizadas estruturas em grafo para armazenamento – ex.: *Neo4j; AllegroGraph.*

Apesar das vantagens apresentadas por todos estes tipos de bases de dados, ao nível do espaço de armazenamento, performance das consultas, aumento da escalabilidade e facilidade na replicação, ainda são tecnologias relativamente recentes. Este fator implica desvantagens como a maturidade das tecnologias, o suporte fornecido para as mesmas, bem como a administração e os conhecimentos especializados necessários para a manutenção deste tipo de base de dados.[33]

No âmbito do BI e da análise OLAP estas bases de dados disponibilizam poucas capacidades para efetuar as consultas e análises necessárias, inclusive a maioria das ferramentas desta área não disponibiliza ligação para bases de dados *NoSQL*, o que só por este motivo elimina a sua utilização em projetos. O que também é sabido é que os modelos relacionais multidimensionais são os mais indicados para consultas complexas**,** como se pretende na maioria das vezes neste tipo de análises, pelo que muitas vezes informação armazenada em bases de dados não relacionais (ex.: *Hadoop*) é transferida para uma base de dados relacional para dar origem a análises de relatórios, etc..[32]

3.2.3. Extração, transformação e carregamento

De todas as fases de um projeto de BI, nomeadamente na construção da DW, o processo de ETL é sem dúvida a mais complexa, morosa e onde têm de ser tidos em conta todos os detalhes e exceções, pois toda a análise depende diretamente da qualidade e veracidade dos dados que se encontram na DW.

O processo consiste na recolha dos dados das diversas fontes, transformações sobre os mesmos: limpeza, formatação, normalização, agregação dos dados, entre outros, e termina com o carregamento para a DW. O foco incide na segunda tarefa, as transformações, sendo que estas nem sempre são triviais e daqui surge a complexidade de todo o processo.

Outro aspeto importante é que o processo deve ser automatizado. Existem diversos pontos de vista quanto à utilização de ferramentas concebidas especificamente para a realização e execução deste processo. Segundo *Kimball* no seu livro *Data Warehouse ETL Toolkit*[2], depende: existem vantagens na utilização de uma ferramenta e vantagens para executar as transformações com código próprio, destacam-se:

	Vantagens
Ferramenta ETL	▪ Desenvolvimento simples, rápido e económico; ▪ Qualquer profissional, sem elevados conhecimentos de programação, pode utilizar a ferramenta; ▪ Criação e armazenamento de metadados torna mais simples de visualizar o processo de transformação de dados; ▪ Mecanismo de agendamento do processo ETL; ▪ Ligação à maioria das fontes de dados necessárias: bases de dados, ficheiros, etc.; ▪ Funcionalidade de encriptação e compressão; ▪ Elevada performance com grande volume de dados; ▪ Possibilidade de introdução de código, sempre que se justificar.

Tabela 6 - Vantagens de utilizar uma ferramenta no ETL

	Vantagens
Código próprio	▪ Utilização de *frameworks* de criação de código e testes, permitindo visualização de resultados, rentabiliza o desenvolvimento e aumenta a qualidade do processo; ▪ Programação orientada a objetos pode ser vantajosa para validação e controlo de erros; ▪ Gestão mais direta dos metadados; ▪ Não existe dependência do conhecimento da linguagem associada à ferramenta; ▪ Maior flexibilidade e independência das funcionalidades impostas pela ferramenta.

Tabela 7 - Vantagens de criar código próprio no ETL

Conclui-se então que existem vantagens em utilizar uma ferramenta ou codificar de raiz todo o processo ETL, depende também dos dados e das transformações que é necessário efetuar nos mesmo. O uso de ferramentas permite desenvolvimento simples, rápido, introdução de código (na linguagem disponibilizada pela ferramenta) e agendamento, motivos pelos quais se optou pela sua utilização. A evolução destas tem sido notória e satisfaz os requisitos necessários do presente projeto.

As ferramentas a utilizar devem seguir um conjunto de pré-requisitos para a sua seleção e utilização, nomeadamente:

- Possibilidade de carregar dados de diversas fontes (diferentes bases de dados, ficheiros CSV*, Excel,* XML*,* etc.);
- Suportar diferentes tipos de dados e metadados;
- Capacidade de elaborar diversas transformações sobre os dados e metadados: limpeza de dados, transformações de valores numéricos, de texto, calcular totais, efetuar agregações, etc.;
- Possibilidade de exportar/carregar dados e metadados para diversas fontes de dados (diferentes bases de dados, ficheiros CSV*, Excel,* XML*,* etc.);
- Permitir a execução agendada do processo de forma automática.

No anexo [1] encontra-se uma análise comparativa das duas ferramentas *open source* mais conhecidas e utilizadas, especialmente tendo em conta os requisitos referidos anteriormente:

- *Pentaho Data Integration*;
- *JasperETL (Talend).*

Destaca-se no *Pentaho Data Integration,* para além da prévia utilização no módulo já desenvolvido do projeto, a possibilidade de processamento paralelo e agendamento dos processos.

3.2.4. Análise de dados

A última fase de um projeto de *business intelligence* é a disponibilização dos dados, de forma a que os utilizadores tenham acesso a estes e que seja perceptível, usualmente em gráficos, tabelas, etc.. – análise OLAP. Esta disponibilização no caso do presente projeto é feita com uma aplicação web de fácil acesso e usabilidade.

Aquando a procura e seleção destas ferramentas é necessário ter em conta um conjunto de aspetos, entre os quais:

- Funcionalidades;
- Compatibilidade;
- Tecnologias;
- Performance;
- Design;
- Segurança;
- Facilidade de utilização.

De seguida é apresentada uma breve análise comparativa entre três ferramentas para desenvolvimento de análises ou aplicações OLAP[25][26][27]:

- *BIRT* (*Eclipse*);
- *Pentaho BI Server*;
- *JasperReports Server*.

A ferramenta ***BIRT*** permite a criação de relatórios bastante básicos acedendo a fontes de dados (base de dados ou ficheiros). Permite apresentação de alguns componentes primários e estáticos (gráficos e tabelas).

O ***JasperReports Server*** (versão *open source* disponibilizada pela *Jaspersoft*), na sua versão gratuita, permite desenhar e desenvolver relatórios OLAP apresentando dados de diversas fontes através de gráficos, etc.. Usualmente são utilizados quando existe grande necessidade de efetuar a impressão da informação através desses mesmos relatórios.

O ***Pentaho BI Server*** é de entre as mencionadas a mais conhecida e com maior comunidade de desenvolvimento e destaca-se pela possibilidade de criação de ***dashboards***, permitindo grande interatividade entre os componentes (gráficos e tabelas) e os utilizadores.

Outra vantagem é que a criação de *dashboards* pode ser feita de raiz, através do *plugin* disponibilizado pela *Webdetails* – o CDE (*Community Dashboard Editor*) que tem como dependência um outro *plugin* CDF (*Community Dashboard Framework*) – incorpora tecnologias como HTML, CSS e *JavaScript*. Os passos que melhor descrevem a forma como este processa encontram-se na Figura 6, resumidamente[30]:

1. Utilizador faz o pedido HTTP para o *dashboard*;
2. Através do nome e caminho o servidor percebe que é um pedido para *dashboard* (ficheiro *.xcdf* – contém instruções HTML e *JavaScript* associadas a cada componente do *dashboard*);
3. A página web é renderizada e apresentada ao utilizador, iniciando assim o *dashboard*, cada componente é posteriormente criado no *JavaScript* e fica associado ao objecto *Dashboards*, a partir do qual é possível realizar comandos como o *update*, que não é mais do que iniciar/atualizar determinado(s) componente(s);
4. O servidor recebe os pedidos anteriormente referidos e executa-os, são o que se denominam de *action sequences* – definidas em ficheiros *XML* que contêm uma sequência de instruções a ser realizadas sobre o(s) componente(s) a ser apresentados.

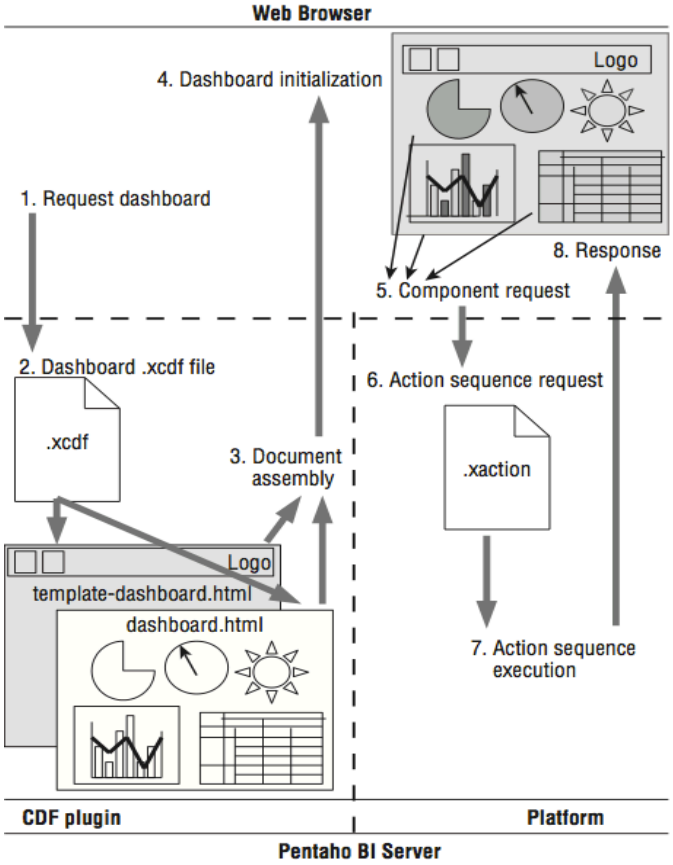

Figura 6 - Arquitetura geral do CDF[30]

Outra grande vantagem do *Pentaho BI Server* é ter integrado um servidor OLAP bastante conhecido em toda a comunidade – *Mondrian* – que permite a construção e armazenamento do cubo (modelo multidimensional definido para a *data warehouse*), aumentando a performance das consultas sobre os dados. Estas são efetuadas através de uma linguagem direcionada para as bases de dados multidimensionais – MDX (*multidimensional expressions*) – tal como já foi referido na arquitetura apresentada. A Figura 7 apresenta um exemplo de uma das muitas consultas MDX criadas.

```
SELECT NON EMPTY {[Measures].[total_alunos_inscritos],

[Measures].[custo_medio],

[Measures].[custo_total_doc]} ON COLUMNS,

NON EMPTY {Hierarchize({[d_uo_cur.hie_d_uo_cur].[All
d_uo_cur.hie_d_uo_curs]})} ON ROWS

FROM [ce_cubo_docentes_alunos]

WHERE {[d_ano_letivo.hie_d_ano_letivo].[2012/2013]}
```

Figura 7 - Exemplo de consulta sobre cubo OLAP em MDX

3.3. Seleção de tecnologias

Nesta secção é exposto um breve resumo dos critérios e motivos da escolha das ferramentas utilizadas no desenvolvimento do projeto, para cada um dos diferentes fins: base de dados

(armazenamento área temporária e DW), ETL e OLAP – consultar Figura 8 – arquitetura tecnológica.

		Observações
BD – área temporária e DW	*PostgreSQL*	▪ Várias capacidades, nomeadamente ao nível das otimizações: índices, particionamento e vistas; ▪ BD utilizada em tecnologias e produtos bastante populares, destacam-se: *IMDB.com*, *Cisco*, *Skype*, entre outros[28].
ETL	*Pentaho Data Integration*	▪ Facilidade de utilização e habituação, devido à sua interface intuitiva; ▪ Componentes fornecidos satisfatórios para as necessidades atuais, diversidade de fontes de dados; ▪ Possibilidade de efetuar transformações paralelas; ▪ Agendamento de processo.
OLAP	*Pentaho BI Server* + *Mondrian*	▪ Criação e edição de *dashboards* podem ser efetuadas com grande facilidade utilizando o *plugin* – CDE; ▪ Disponibilização de um servidor de OLAP para desenho e construção do cubo, garantindo performance elevada no acesso aos dados.

Tabela 8 - Critérios de seleção das tecnologia

Dado o desenvolvimento ter sido iniciado na sequência de outro módulo já desenvolvido, o *feedback* que foi dado pela equipa anterior contribuiu com uma percentagem para a seleção das tecnologias, mantendo-se o *Pentaho Data Integration* e *BI Server*.

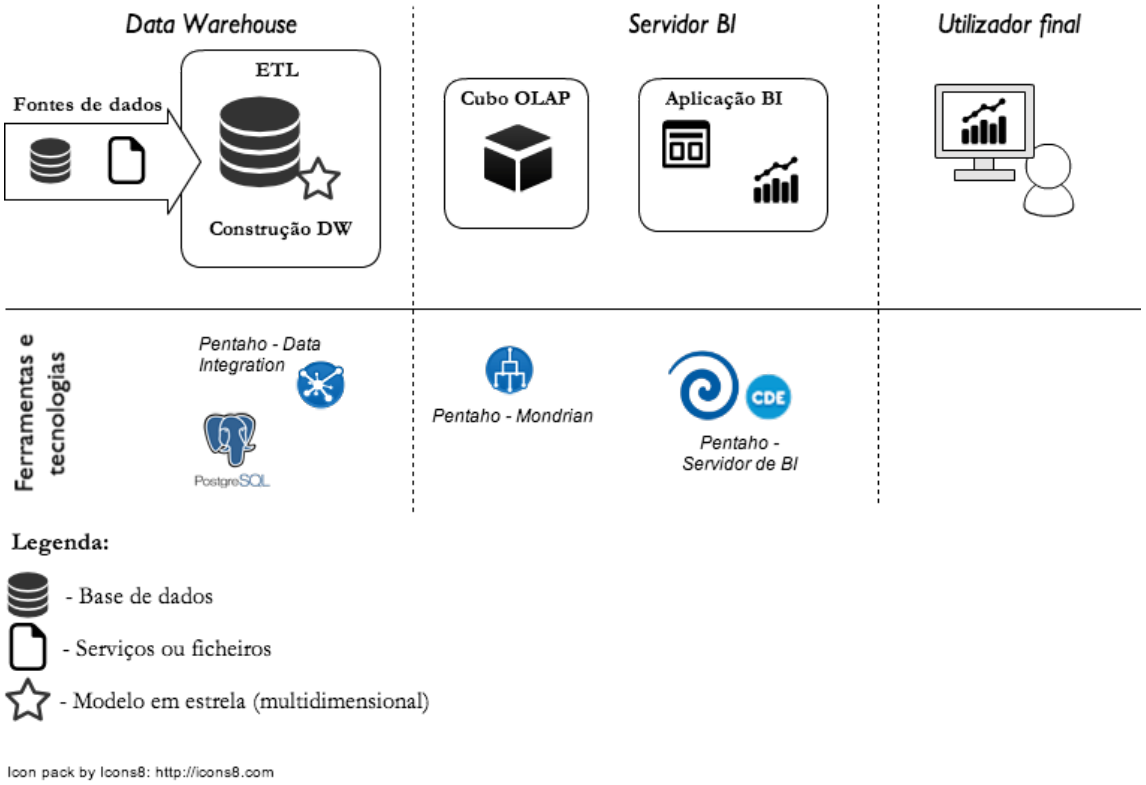

Figura 8 - Arquitetura tecnológica

Realçar que o *plugin* CDE, utilizado para desenvolver as páginas HTML com os respetivos *dashboards*, inclui a possibilidade de utilizar e definir todo o formato da página através de CSS e utilizar bibliotecas de *javascript* para diversos fins.

3.4. Considerações arquiteturais

No decorrer do desenvolvimento foram tomadas algumas decisões com importância para a arquitetura do projeto em diferentes etapas: processo ETL e modelo de dados.

No processo ETL surge a questão da recolha de dados do sistema SAP, isto é, foram propostas duas alternativas para esta extração: utilizar os serviços desenvolvidos com o propósito de alimentar apenas este projeto ou, por outro lado, aceder a uma réplica da base de dados de SAP que seria atualizada diariamente. Para a segunda alternativa foi necessário construir de raiz as consultas para obtenção de resultados, recorrendo à definição dos serviços já disponibilizados.

A alternativa mais adequada, e que faz mais sentido optar, são os serviços disponibilizados. Quando se trata de integração de dados a melhor forma de o fazer é através de serviços, visto que estes permitem a separação total da lógica presente no modelo de dados e das bases de dados de origem (que no caso concreto do sistema SAP é bastante complexa, pelo que a tarefa de construção das consultas também se revelou uma tarefa complexa).

Relativamente ao modelo de dados também foi inicialmente definido que, sendo este módulo parte de uma aplicação que engloba outros módulos, o modelo dimensional deveria ser partilhado entre módulos naquilo que se mostrasse ser comum. Contudo, levantaram-se algumas questões quanto a esta partilha, nomeadamente na manutenção e crescimento futuro de cada módulo, isto é, alterações futuras que possam surgir num determinado módulo poderiam vir a afetar o bom funcionamento dos restantes. Outro aspeto discutido está relacionado com o volume de dados partilhados. A única partilha que poderia existir verificar-se-ia ao nível das tabelas das dimensões, mas uma vez que estas ocupam o menor espaço numa DW não existe vantagem significativa em mantê-las partilhadas. Neste sentido, optou-se por manter cada módulo com o seu *data mart* – que é por definição um subconjunto de uma *data warehouse*, armazenando dados relacionados com uma área específica, neste caso dados com a atividade de ensino.

Outro aspeto importante é a atualização do *data mart*, isto é, no primeiro carregamento todos os dados recolhidos e agregados são armazenados no *data mart*. No primeiro carregamento são armazenados dados desde o ano letivo 2011/2012 até ao presente. O agendamento do processo ETL é anual, pelo que nos carregamentos consecutivos, por forma a otimizar e diminuir significativamente o tempo, são tomados em consideração os seguintes pontos:

- Na UC, a efetuar alguma alteração ao nível da carga horária, esta alteração ocorrerá no ano letivo imediatamente anterior, pelo que apenas será necessário verificar alterações nesse mesmo ano letivo;
- No que diz respeito às remunerações dos funcionários, não existe qualquer alteração das mesmas, pois os acertos são efetuados ao longo de todo o ano civil (consequentemente letivo).

Neste ponto, referenciar também que o requisito RF_GE_01 – autenticação – foi modificado já durante o desenvolvimento. Os *stakeholders* verificaram a necessidade de o acesso à aplicação ser mais restrito do que tinham pensado inicialmente, razão pela qual foi

necessário estudar alternativas e soluções possíveis tendo em conta o momento da alteração. Após análise e pesquisa verificou-se ser possível restringir o acesso a um determinado grupo de utilizadores (estes grupos passaram a estar definidos no servidor LDAP da UC), a cada módulo da aplicação. No caso do presente módulo apenas terão acesso utilizadores que pertençam aos grupos: DW_CE e DW_GLOBAL, este último inclui utilizadores que tenham acesso a todos os módulos da aplicação.

3.5. Modelo de dados

No que respeita a modelos de dados existem dois momentos em que é preciso desenhar e ter em conta um: na área temporária e no carregamento dos dados para o *data mart*. Nas subsecções que se seguem ambos são abordados e especificados.

3.5.1. Modelo área temporária

A área temporária é uma componente do processo de extração de dados no ETL, existindo mais do que uma fonte de dados é comum que surja a necessidade de armazenar estes de forma temporária, para que possam posteriormente entrar no processo de transformação.

Para melhor compreensão sobre quais os dados da área temporária é necessário conhecer quais as fontes de dados: sistema de gestão académica NONIO, que fornece informação sobre o serviço docente e alguns dados demográficos dos docentes, e o sistema de gestão SAP, que fornece informação completa sobre a demografia e remuneração mensal dos recursos humanos da UC. Os dados do NONIO são obtidos através de vistas materializadas; os de SAP através de serviços especificamente desenvolvidos para este projeto. A especificação das fontes de dados pode ser consultada respetivamente nos anexos: NONIO_VistasServiçoDocente_06-12-2013.pdf (anexo [9]) e DW_DadosWebServices_v2.pdf (anexo [10]).

Para este caso, opta-se por criar uma base de dados com um modelo relacional, que permita conjugar a informação que vem de ambas as fontes, apresentado na Figura 9.

A informação armazenada, proveniente das fontes dados, diz respeito a cada funcionário da UC (docente ou não docente):

- dados demográficos de funcionários docentes e não docentes;
- remunerações segundo as rúbricas económicas (fixadas pelo ministério das finanças para a despesas das instituições públicas);
- no caso dos docentes, informação sobre o serviço de docente;
- informação sobre as inscrições de alunos em cursos e unidades curriculares.

Existe um conjunto de informação demográfica que pode alterar-se ao longo do tempo: categoria, habilitação literária e unidade orgânica do funcionário. Este é um aspeto pertinente a ter em conta no desenho do modelo para o *data mart*, apresentado na subsecção seguinte.

Quanto aos valores do orçamento para o ano civil na UC, provenientes do ficheiro, como é um valor único por ano civil, o mapeamento é praticamente direto entre a fonte e a tabela de factos correspondente no *data mart*. Por este motivo esse valor não é armazenado no modelo, aqui apresentado, para a área temporária.

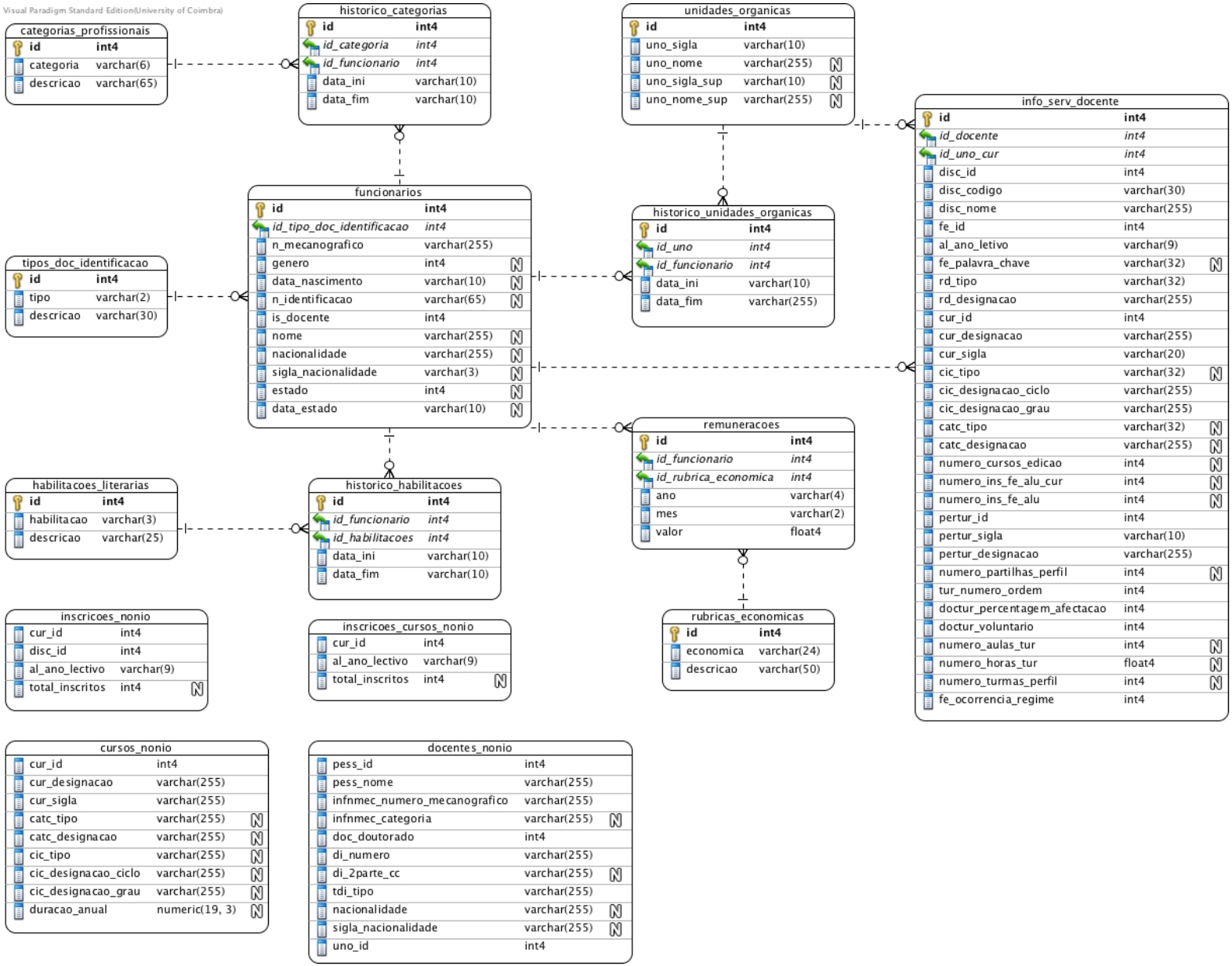

Figura 9 - Modelo de dados da área temporária

3.5.2. Modelo multidimensional

Como já referido neste capítulo, após a recolha de dados e transformação sobre os mesmos estes são carregados para uma base de dados com um modelo multidimensional, ou em "estrela" como também é conhecido. Este modelo é composto por factos e dimensões, factos são valores, métricas aditivas ou semi-aditivas; as dimensões são atributos que caracterizam os factos. Num *data mart* é permitido efetuar operações de:

- *drill down* – obter dados de um nível de granularidade inferior ou acrescentar uma dimensão (desagregar dados);
- *roll up* – operação inversa do *drill down*, obter dados de um nível de granularidade superior ou remover uma dimensão (agregar dados);
- *drill across* – quando o modelo contém várias tabelas de factos, é possível, obter dados de múltiplas tabelas de factos;
- *slice* – obter dados restringindo o valor de uma dimensão;
- *dice* – obter dados restringindo valores de várias dimensões;
- *slice and dice* – combinação entre as duas operações.

No caso do presente projeto, o esquema em "estrela" do *data mart* é uma constelação de factos. Este termo surge quando existem múltiplas tabelas de factos que partilham entre si dimensões. O esquema pretende demonstrar também a complexidade do modelo multidimensional.

As dimensões estão relacionadas com os níveis de granularidade disponíveis (*drill down* e *roll up*) e com os atributos necessários para efetuar o *slice and dice* sobre os factos. Os factos são todos os valores necessários ao cálculo dos indicadores a disponibilizar aos utilizadores.

As tabelas das dimensões devem conter caracterização até à granularidade mais fina possível, pelo que o seu crescimento é na horizontal – número considerável de atributos (colunas). Nas tabelas de facto é o oposto, para além das métricas deve conter chaves das diversas dimensões segundo a sua granularidade, cresce na vertical – elevado número de registos (linhas). Na Tabela 9, Tabela 10 e Tabela 11, são discriminadas todas as dimensões, identificadas a branco no modelo da Figura 10. É feita referência a decisões que foram tomadas no desenho do modelo multidimensional, derivadas das particularidades dos dados recolhidos. Algumas também relacionadas com a otimização e performance de utilização do *data mart.*

Dimensão	**Descrição**
Tempo (*d_tempo*)	O tempo é uma dimensão essencial num *data mart*. Para este caso específico o tempo está relacionado com o período de regime das unidades curriculares. A granularidade mais fina será o trimestre de um ano letivo.
Parcela (*ce_d_parcela*)	A parcela representa todas as parcelas que entram para os custos, à exceção dos docentes. Não docentes, instalações, material de laboratório, etc., são exemplos de parcelas que podem ser armazenadas nesta dimensão. Até ao momento apenas foi possível recolher informação sobre a parcela de não docentes.

Tabela 9 - Descrição geral das dimensões do modelo do *data mart* (1)

Dimensão	Descrição
Unidade orgânica (*d_unidade_organica*)	As faculdades, departamentos, institutos e colégios inserem-se nesta dimensão. Na UC, apesar de um departamento ser considerado uma UO, existem faculdades que são compostas por departamentos, hierarquicamente. Para representar este nível hierárquico existe uma segunda chave para esta mesma dimensão nas respetivas tabelas de factos. Existe ainda um atributo que indica se determinada UO contém departamentos (*tem_departamento*), se tiver valor 1 significa que a UO é uma faculdade com departamentos. Valor 0 caso contrário. Para além da sigla e designação da UO, é armazenada a designação que deve aparecer nas opções de parâmetros, resulta da combinação de ambas. Exemplo: FCTUC (Faculdade de Ciências e Tecnologia).
Curso (*d_curso*)	Nesta dimensão são armazenados os cursos da UC. Para além da sigla e designação, é armazenada a designação que deve aparecer nas opções de parâmetros, resulta da combinação de ambas. Exemplo: LEI (Licenciatura em Engenharia Informática). Foi também adicionada informação relativa ao ciclo de estudos do curso nesta dimensão, pois a probabilidade de um curso mudar de ciclo de estudo é nula. Os atributos do ciclo de estudos são: ciclo, grau e categoria. Para os parâmetros a selecionar pelo utilizador deve aparecer uma combinação entre o grau e a categoria. Exemplo: Mestrado (Integrado), pelo que também é guardado este campo.
Unidade curricular (*d_unidade_curricular*)	A designação das unidades curriculares de cada curso é armazenada nesta dimensão, bem como o regime de frequência afeto à mesma (anual, semestral ou trimestral). Existe também referência à ocorrência do regime no caso de ser semestral ou trimestral: primeiro, segundo ou terceiro.
Funcionário (*ce_d_funcionario*)	Informações sobre cada funcionário (docente ou não docente) estão nesta dimensão: número mecanográfico, nome, tipo de identificação, número de identificação, tipo (docente ou não docente), estado (ativo, pensionista, suspenso ou saiu da empresa) e data de alteração e data de nascimento. Estes atributos são passíveis de sofrerem alterações ao longo do tempo, um exemplo comum, é quando um funcionário se casa ou divorcia. O campo nome será atualizado. Quando uma dimensão permite que os atributos se alterem ao longo do tempo, designa-se, em *data warehousing*, como *slowly changing dimension* (SCD). Existem diversas soluções para estas situações, dado que o objetivo é manter o histórico, a solução passa por criar um novo registo na dimensão com os novos valores. No exemplo do nome, numa situação de divórcio, o mesmo funcionário fica com dois registos, um com o nome de casado e outro, com o nome após divórcio.

Tabela 10 - Descrição geral das dimensões do modelo do *data mart* (2)

Dimensão	Descrição
Demografia do funcionário (*ce_d_demografia_funcionario*)	A demografia armazena também informação sobre cada funcionário (docente ou não docente). São atributos vulneráveis a alterações, com maior frequência, e utilizados em várias consultas de *slice and dice*. Por esta razão, considera-se uma mini dimensão da dimensão funcionário. Permite aumentar a performance das consultas sempre que esta dimensão é utilizada, e sempre que é necessária uma atualização sobre algum dos atributos, não existe necessidade de duplicar os que estão sujeitos a menos alterações (dimensão Funcionário). A informação aqui armazenada diz respeito a: género, faixa etária (limites inferior e superior), nacionalidade, grau académico e categoria.

Tabela 11 - Descrição geral das dimensões do modelo do *data mart* (3)

As dimensões caracterizam os factos (métricas) e definem a sua granularidade. No caso do módulo com os custos com o ensino distinguem-se seis tabelas de factos, identificados a cinza no modelo da Figura 10, com diferentes granularidades:

- Métricas individuais de cada funcionário – *ce_f_funcionarios*;
- Métricas relativas a docentes, até ao nível da unidade curricular – *ce_f_docentes*;
- Número de alunos inscritos, cuja granularidade mais baixa é ao nível do curso – *ce_f_alunos_curso*;
- Número de alunos inscritos, cuja granularidade mais baixa é a unidade curricular – *ce_f_alunos;*
- Custos com outras parcelas (não docentes, instalações, etc.) que apresentam o curso como nível mais baixo – *ce_f_outras_parcelas*;
- Orçamento para o ano para a atividade de ensino, atividades estruturais e de suporte na UC, cuja granularidade é ao ano letivo – *ce_f_orçamento*.

Também aqui, tal como nas dimensões, são tomadas decisões específicas, quanto aos factos armazenados e respetivas tabelas. De seguida, é apresentada uma análise e descrição sobre cada uma das tabelas de factos.

ce_f_funcionarios		
Factos	**Nome**	**Descrição**
	n_horas	Número de horas lecionadas por determinado docente.
	custo_total_docente	Custo total, do funcionário, no caso dos docentes com a atividade de ensino (remuneração tendo em contas as horas do serviço docente) e para os não docentes toda a massa salarial (remuneração ano letivo).
	custo_medio_hora	Custo médio, por hora, do docente.
Granularidade	A granularidade mais fina desta tabela é o funcionário, individual. É possível aplicar *slice and dice* por todas as dimensões, excepto por parcelas. O *drill down* e, respetivo *roll up*, podem ser efetuados da unidade orgânica, cursos, unidades curriculares até aos docentes.	

Tabela 12 - Especificação da tabela de factos: funcionários

ce_f_docentes		
Factos	**Nome**	**Descrição**
	n_docentes	Número total de docentes.
	n_horas	Número de horas lecionadas pelos docentes.
	custo_total_docentes	Custo total, dos docentes, com a atividade de ensino (remuneração tendo em contas as horas do serviço docente).
	custo_medio_docentes	Custo médio dos docentes, tendo em conta a atividade de ensino.
	custo_medio_hora	Custo médio, por hora, dos docentes.
Granularidade	A granularidade mais fina desta tabela é a unidade curricular. É possível aplicar *slice and dice* nas dimensões de tempo, unidade orgânica, curso e unidade curricular. O *drill down* e, respetivo *roll up*, podem ser efetuados da unidade orgânica, cursos até às unidades curriculares. Os factos presentes nesta tabela, são calculados através dos agregados da tabela *ce_f_funcionarios*, pois esta apresenta granularidade mais fina. Opta-se por criar esta tabela de factos para evitar efetuar novos cálculos, sempre que é necessário o custo de um conjunto de docentes por unidade orgânica, curso ou unidade curricular. Esta decisão contribui para o aumento de performance do *data mart* e consequentemente do sistema.	

Tabela 13 - Especificação da tabela de factos: docentes

ce_f_alunos_curso		
Factos	**Nome**	**Descrição**
	n_alunos	Número de alunos inscritos.
Granularidade	A granularidade mais fina desta tabela é o curso. É possível aplicar *slice and dice* nas dimensões de tempo (ano letivo), unidade orgânica e curso. O *drill down* e, respetivo *roll up*, podem ser efetuados da unidade orgânica aos cursos.	

Tabela 14 - Especificação da tabela de factos: alunos no curso

ce_f_alunos		
Factos	**Nome**	**Descrição**
	n_alunos	Número de alunos inscritos.
Granularidade	A granularidade mais fina desta tabela é a unidade curricular. É possível aplicar *slice and dice* nas dimensões de tempo (ano letivo), unidade orgânica, curso e unidade curricular. O *drill down* e, respetivo *roll up*, podem ser efetuados da unidade orgânica, cursos e unidade curricular.	

Tabela 15 - Especificação da tabela de factos: alunos na unidade curricular

ce_f_outras_parcelas		
Factos	**Nome**	**Descrição**
	custo_total	Custo total, das outras parcelas (não docentes, instalações, etc.) na atividade de ensino.
Granularidade	A granularidade mais fina desta tabela é o curso. É possível aplicar *slice and dice* nas dimensões de tempo, unidade orgânica, curso e parcela. O *drill down* e, respetivo *roll up*, podem ser efetuados da unidade orgânica até ao curso. Os factos presentes nesta tabela, são calculados através dos agregados da tabela *ce_f_funcionarios*, pois esta apresenta granularidade mais fina, e como a referência a funcionários não docentes é sempre como um todo e não individualmente como acontece com os docentes. Na prática, o custo com docentes, é uma parcela do custo com o ensino, contudo ele não se encontra representado nesta tabela. Opta-se por considerar que o custo proveniente da remuneração de docentes, não é, segundo o modelo de dados uma parcela. A razão está no facto de ser necessário obter uma lista com os docentes de uma dada unidade curricular, logo a granularidade é diferente desta tabela, ficando a parcela de custo com docentes nas tabelas de factos *ce_f_funcionarios* e *ce_f_docentes*.	

Tabela 16 - Especificação da tabela de factos: outras parcelas

ce_f_orcamento		
Factos	**Nome**	**Descrição**
	total_orcamento_aprovado	Total do orçamento para o ano, para a atividade de ensino, atividades estruturais e de suporte na UC, no ano civil.
Granularidade	A granularidade mais fina desta tabela é a UC. Em relação à dimensão temporal, apenas é possível obter uma granularidade ao ano letivo.	

Tabela 17 - Especificação da tabela de factos: orçamento total aprovado

No momento do desenho do modelo multidimensional deve ser efetuada uma estimativa do tamanho que este irá ocupar na base de dados, tanto em dimensões como em factos, excluindo desta estimativa quaisquer índices ou vistas materializadas. Nas tabelas do anexo [3], é detalhada a estimativa para 10 anos. O tamanho estimado é de 51 MB, que tem apenas como referência os registos na base de dados, contudo o espaço ocupado por todo o *data mart* é maior devido à utilização do cubo. Isto é, o servidor OLAP *Mondrian* armazena em *cache* os valores dos agregados para os diferentes níveis de granularidade (processamento *in-memory*), possibilita elevada performance no acesso aos dados, contudo fica dependente da memória *cache* disponível – efetua internamente o controlo da *cache* tendo em conta as consultas que vão sendo efetuadas, a partir do momento que um valor é armazenado permanece em memória e é utilizado em consultas subsequentes[5].

Apesar do tamanho estimado ser aparentemente reduzido, a utilização de um modelo multidimensional para análise de dados é bastante vantajosa ao nível da performance das consultas sobre o mesmo. Vamos supor que a análise apresentada era efetuada utilizando a base de dados operacional. Existem em média, 15000 registos de serviço docente por ano e 137000 registos de remunerações a ser processados. O volume de informação é demasiado elevado para processar em tempo real e apresentar a informação necessária. Por esta razão é bastante favorável a utilização do modelo em "estrela", cujos os agregados já se encontram calculados, na granularidade mais alta, para toda a UC. Outro aspeto positivo é que mantendo os dados para análise numa base de dados desacoplada do sistema operacional retira qualquer sobrecarga neste último.

No modelo multidimensional as tabelas de factos ocupam mais de 90% do espaço, enquanto que o restante é ocupado pelas dimensões[1].

[5] Documentação (arquitetura e controlo de cache):
http://mondrian.pentaho.com/documentation/architecture.php;
http://mondrian.pentaho.com/documentation/cache_control.php

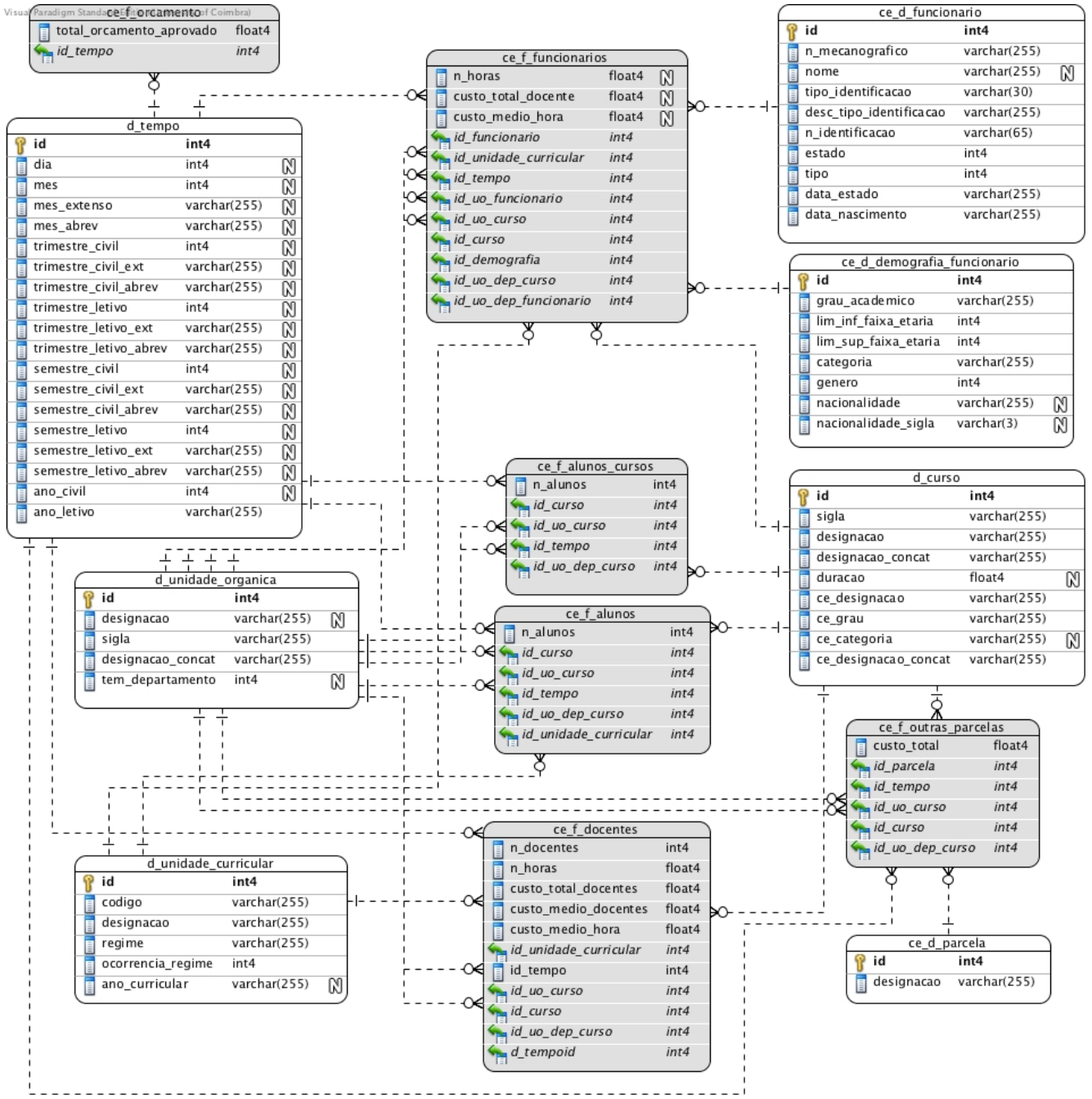

Figura 10 - Modelo multidimensional da DW

3.6. Sumário

A arquitetura de alto nível do sistema foi definida, bem como as tecnologias utilizadas em todos os seus componentes. As tecnologias foram alvo de um estudo e análise que permitiram conhecer o mercado do BI atualmente, e que alternativas este apresenta ao nível do software livre.

Os modelos de dados para área temporária e *data mart* foram desenhados e especificados, e o modelo multidimensional foi alvo de revisão e validação, segundo os requisitos definidos através dos protótipos.

Capítulo 4
Implementação

Esta secção expõe detalhes ao nível da implementação, nomeadamente no que respeita ao desenvolvimento do processo de ETL, é também feita referência ao funcionamento do cubo OLAP. No final é apresentado o resultado final da aplicação desenvolvida.

4.1. Plano do processo de extração, transformação e carregamento

Como já mencionado no documento, o processo ETL apresenta elevada complexidade e é o principal componente na construção de um *data mart*.

É composto por três fases, na primeira – **extração** – é efetuada a recolha dos dados fontes e estes reúnem-se na área temporária (consultar modelo da Figura 9) para posterior utilização – **transformações**.

Na fase de transformação os dados são limpos e formatados (remoção de duplicados, eliminação de espaços, substituição de valores, etc.), de seguida são normalizados, por forma a estarem de acordo com o modelo do *data mart* e por fim, é possível efetuar o cálculo de agregados.

É na segunda fase do processo – transformação - que reside a principal dificuldade. Os dados provenientes das fontes de dados podem conter erros, inconsistências e formatações incoerentes. Ter conhecimento aprofundado sobre as fontes de dados permite definir todas as transformações, contudo este processo é automático, logo prever o que irá ser recolhido é uma tarefa exigente. O processo de limpeza, formatação e normalização deve contemplar o máximo número de possibilidades, para que nenhum dado inconsistente ou inválido seja guardado no *data mart* e, posteriormente, utilizado na análise disponibilizada aos utilizadores.

A última fase do processo diz respeito ao **carregamento** dos dados para o *data mart* (consultar modelo da Figura 10), o mais comum é efetuar o carregamento/atualização das dimensões, seguido dos factos.

Segundo *Kimball R.*, mais de 50% do esforço embutido na construção do *data mart* encontra-se nesta fase, e mais de 70% de projetos mal sucedidos, na área do *data warehousing*, são provenientes do ETL. Estes dados mostram a importância, complexidade e minúcia da definição e desenvolvimento de todo este processo.

Na secções que se seguem o processo ETL é abordado com maior detalhe, apresentando quais as transformações utilizadas e exemplos de maior impacto no desenvolvimento do mesmo.

4.1.1. Transformações – conceitos e notações

Nesta subsecção são abordados alguns conceitos e notações próprios da ferramenta utilizada para o desenvolvimento de todo o processo ETL – *Pentaho Data Integration*. É dado ênfase aos que foram utilizados durante o desenvolvimento deste processo em específico e na Tabela 18 e Tabela 19 encontra-se uma breve descrição de cada um deles.

Esta exposição é importante para uma melhor e mais rápida compreensão das subseções que se seguem.

Tipo	Ícone	Designação	Descrição
Flow		*Job*	Permite organizar e estruturar as chamadas a um conjunto de transformações ou inclusive outros *jobs*.
		*Transformation*	Conjunto de passos (diversos componentes que permitem efetuar todas as transformações pretendidas com os dados recolhidos e com os que irão ser armazenados posteriormente).
Data Warehouse		Combinação *lookup/update*	Este passo é utilizado no carregamento das dimensões do *data mart*. O que ele faz é pesquisar segundo um conjunto de atributos a linha na tabela, caso ela não exista é inserido caso contrário, é feita uma atualização sobre a existente.
Lookup		*Join* BD	Executa uma consulta SQL, permitindo que sejam utilizados atributos ou variáveis que estejam presentes naquele instante, no processo.
		*Lookup* BD	Dado um conjunto de atributos, pesquisa e retorna os respetivos registos presentes na BD.
		Pedido HTTP	Efetua um pedido a um *web service* dado o URL e os diversos parâmetros definidos dinamicamente. Utilizado para recolher os dados provenientes dos serviços de SAP.
Joins		*Merge Join*	Este passo une dois fluxos de dados segundo um determinado atributo. Um exemplo da sua utilização é no carregamento dos cursos para a respetiva dimensão, em que esta é preenchida com dados vindos do serviço docente e dados da vista que contém todos os cursos presentes no sistema NONIO.
Scripting		Executa *script* SQL	Como o nome indica, executa comandos de SQL como *insert, update, delete,* etc. Utilizado no presente projeto atual para atualizar as dados dos funcionários que existam no sistema NONIO.
		*Script*	Permite definir e executar código em *JavaScript*. Muito útil em todo o desenvolvimento do processo, para efetuar cálculos, trabalhar com datas, etc.

Tabela 18 - Descrição dos principais componentes do processo ETL (1)

Tipo	Ícone	Designação	Descrição
Transformar		Adiciona constante	Passo que permite adicionar, ao fluxo de dados atual, um atributo com valor constante. Útil para a junção de diferentes fluxos de dados, em que para tal acontecer é necessário que tenham o mesmo conjunto de campos.
		Registos únicos	Remove duplicados no fluxo de dados ordenado. Pode inclusive definir-se quais os campos a comparar.
		Separa campo por linhas	Dado um atributo do tipo *string*, este passo separa-o por diversas linhas dado o delimitador (por exemplo, uma vírgula).
		Ordenar linhas	Passo que ordena as linhas do fluxo de dados, segundo determinados atributos.
		Calculadora	Permite efetuar um conjunto de cálculos predefinidos, utilizando os atributos do fluxo de dados dando origem a novos.
		Operações com *strings*	Aplicar determinadas operações sobre campos do tipo *string* como *trim,* maiúscula e minúsculas, etc.
		Seleciona valores	Passo utilizado com alguma frequência para eliminar ou modificar os atributos presentes no fluxo de dados atual do processo.

Tabela 19 - Descrição dos principais componentes do processo ETL (2)

4.1.2. Fluxo geral do processo ETL

O processo ETL apresentado na Figura 11 é o *job* mais geral. É a partir deste que são efetuadas chamadas a todos os outros *jobs*, respetivamente: carregamento da área temporária, carregamento de dimensões e por fim, carregamento das tabelas de factos.

Este é o *job* que deve ser executado tanto para o primeiro carregamento, como para os carregamentos subsequentes.

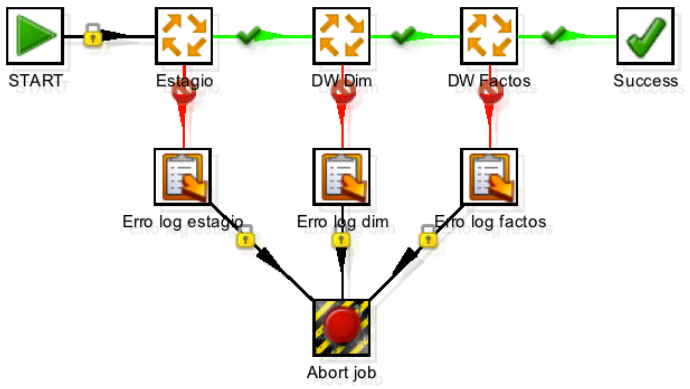

Figura 11 - Processo ETL geral (*ce_geral.kjb*)

Na definição de cada um dos *jobs* encontram-se chamadas a diversas transformações específicas de cada fase por ele representada. Todo o processo está faseado para uma melhor compreensão, *log* e correção de erros, minimizando o custo de alterações futuras. No total foram implementadas 35 (*jobs* e *transformations*) – estas podem ser consultadas no anexo [13].

4.1.3. Carregamento da área temporária

Como já referido anteriormente, a recolha e o carregamento dos dados fontes para a área temporária é a primeira fase de todo o processo. Na Figura 12 é apresentado o *job* onde estão definidos e são chamadas todas as transformações necessárias para esta fase. Resumidamente os passos neste processo são os seguintes:

- Recolha e armazenamento das unidades orgânicas provenientes do sistema NONIO;
- Recolha e armazenamento das habilitações literárias, categorias profissionais e tipo de documentos de identificação existentes no sistema SAP;
- Extração e armazenamento dos funcionários presentes no sistema SAP;
- Pesquisa dos funcionários que são comuns em ambos os sistemas para atualizar informações presentes no NONIO dos funcionários docentes;
- Recolha e armazenamento do histórico das unidades orgânicas, categorias profissionais e habilitações literárias dos funcionários do sistema SAP;
- Para auxiliar a inserção do serviço docente, através do identificador único dos docentes, são inseridos todos os docentes presentes no sistema NONIO;
- Extração e armazenamento dos cursos e inscrições em cursos e unidades curriculares do NONIO;
- Recolha e armazenamento das rúbricas e respetivas remunerações dos funcionários presentes no sistema SAP;
- Por fim, a recolha e armazenamento do serviço docente dos docentes presentes no sistema NONIO.

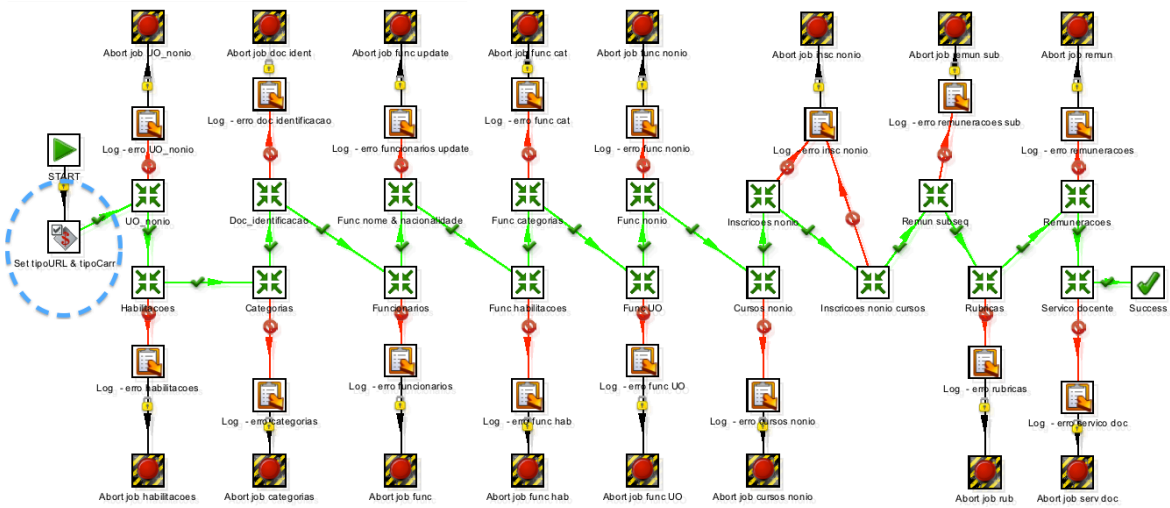

Figura 12 - Processo de carregamento da área temporária (*ce_job_extracao_estagio.kjb*)

Destaca-se no processo apresentado o segundo passo *"Set tipo & tipoCarr"*. Aqui são definidas duas variáveis de grande importância: o tipo de URL que deve ser utilizado nos pedidos aos *web services* de SAP, isto é, se é pretendido utilizar os de desenvolvimento (dev) ou os de produção (prod). Para implementação e testes são utilizados somente os de desenvolvimento para não sobrecarregar o sistema fonte. É também especificado o tipo de carregamento, se é o primeiro carregamento (valor 0) ou subsequentes (valor 1).

Na presente seção não se encontram detalhadas todas as transformações expostas na Figura 12, será efetuada uma apresentação exemplificativa de algumas mais significativas.

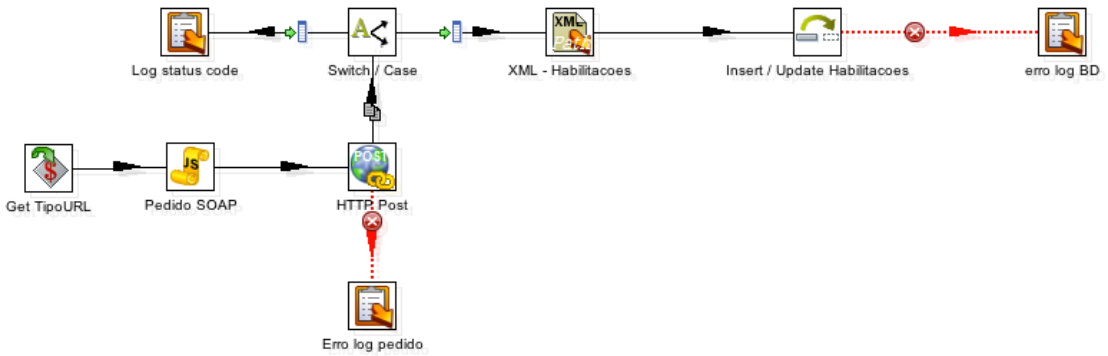

Figura 13 - Processo de recolha e armazenamento das habilitações literárias (*ce_input_hab.ktr*)

Na Figura 13 encontra-se todo o fluxo de dados para a recolha e armazenamento das habilitações literárias do sistema SAP. Inicialmente é obtido o valor do tipo de URL a ser utilizado para aceder aos *web services* (desenvolvimento ou produção) e de seguida constrói-se dinamicamente o pedido SOAP (*Simple Object Access Protocol* - Figura 14) a ser feito aos serviços disponibilizados através do pedido HTTP.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sap-com:document:sap:soap:functions:mc-style">
   <soapenv:Header/>
   <soapenv:Body>
      <urn:ZhrWs028>
         <IUser>{user}</IUser>
      </urn:ZhrWs028>
   </soapenv:Body>
</soapenv:Envelope>
```

Figura 14 - Exemplo pedido SOAP para *web service*

No caso do pedido ser bem sucedido o passo seguinte é obter o resultado em XML, para na fase final introduzir ou atualizar os registos já existentes na tabela correspondente às habilitações na área temporária. De notar que todo o fluxo de dados é acompanhado de um controlo de erros para *log*.

Outras das transformações de destaque neste processo da área temporária é a relativa à inserção do serviço docente - Figura 15 – onde inicialmente se obtém qual o tipo de carregamento que está a ser efetuado (primeiro ou subsequente) e consoante esse tipo é recolhido todo o serviço docente ou apenas o do ano letivo anterior e o atual.

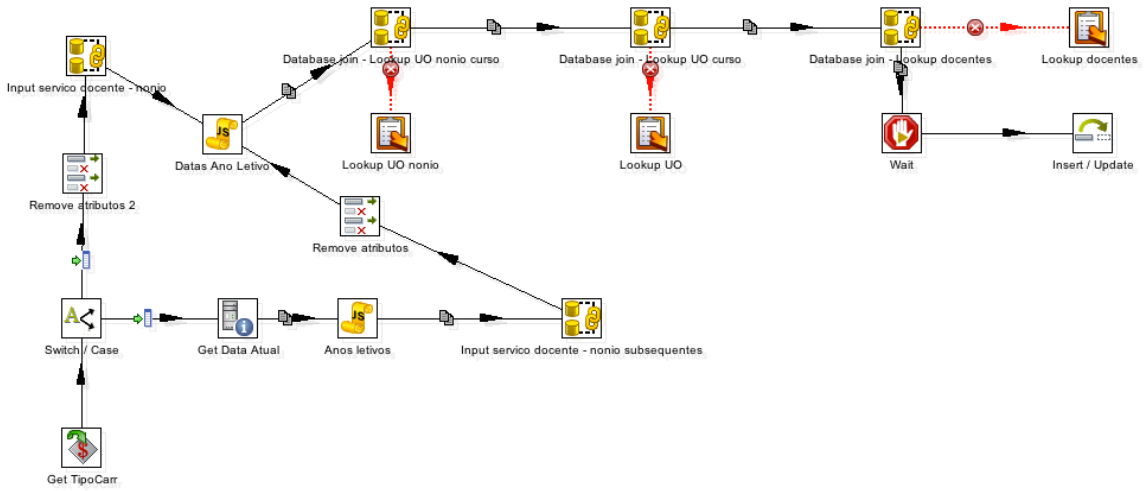

Figura 15 - Processo de fluxo de dados para extração e armazenamento do serviço docente (*ce_input_serv_doc.ktr*)

O passo seguinte é obter para cada registo do serviço docente qual o identificador da unidade orgânica associada ao curso: numa primeira etapa obter o nome e a sigla associadas no sistema fonte NONIO e posteriormente com esses campos procurar o respetivo identificador na tabela de unidades orgânicas na área temporária.

De seguida é feita a pesquisa do identificador do docente para que**,** por fim**,** possam ser introduzidos os registos de serviço docente na tabela correspondente na área temporária.

As transformações relativas à recolha e armazenamento das remunerações de docentes são igualmente de destacar. Nesta fase também é necessário ter em conta o tipo de carregamento que está a ser efetuado, isto é, no primeiro carregamento são carregados todos os anos de histórico – opta-se por colocar estes num ficheiro CSV – logo nos carregamentos subsequentes este ficheiro tem de ser atualizado para apenas serem carregadas as remunerações do ano letivo atual. Para realizar tal operação é efetuado o processo apresentado na Figura 16.

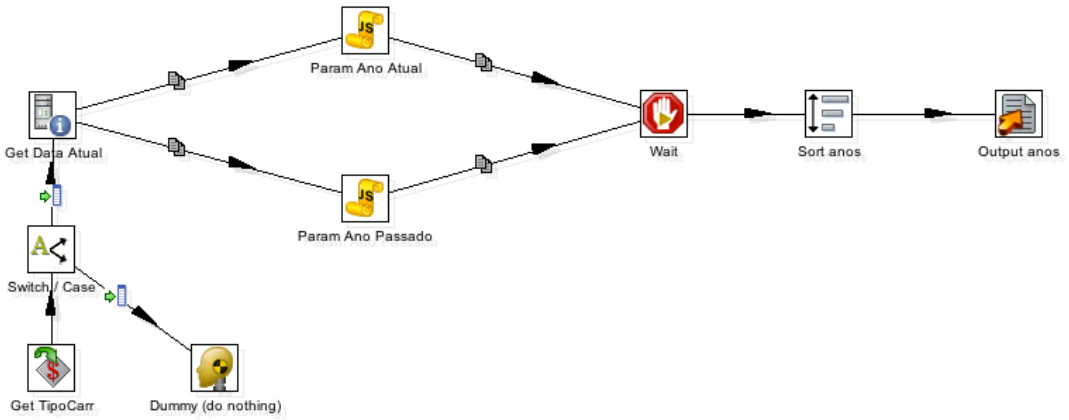

Figura 16 - Processo para atualização do ficheiro de anos letivos a considerar na recolha das remunerações (*ce_output_anos_rub&remun.ktr*)

Conclui-se pela complexidade da Figura 12 que o carregamento para a área temporária é constituído por várias etapas, essencialmente devido à existência de duas fontes de dados distintas onde a informação deve ser unida desde logo no modelo que corresponde a esta área.

4.1.4. Carregamento das dimensões

O carregamento das dimensões é a segunda fase do processo geral de ETL. Este contém também um *job* específico para executar as diferentes transformações correspondentes a cada uma das tabelas de dimensão do modelo do *data mart* – Figura 17.

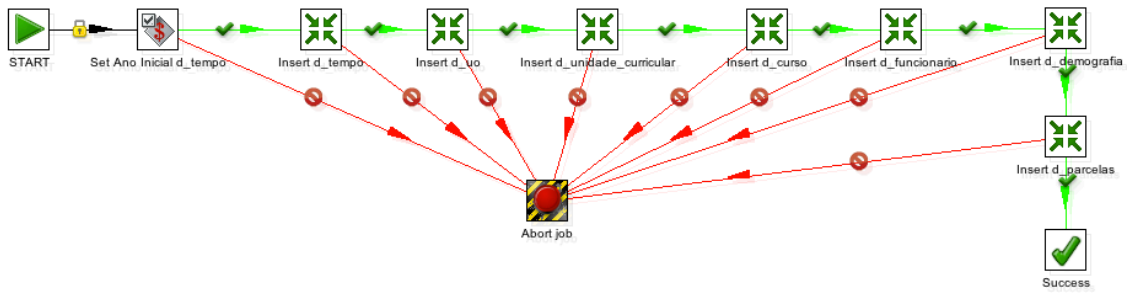

Figura 17 – Processo de carregamento das dimensões (*ce_job_carregamento_dim.kjb*)

No modelo desenhado para o *data mart* existem sete dimensões: tempo, unidade orgânica, unidade curricular, curso, funcionário, demografia de funcionário e parcela.

Destacar novamente, é necessária a criação de uma variável que auxilia o carregamento da dimensão de tempo, isto é, no caso de se tratar do primeiro carregamento esta variável deve ter o ano letivo a partir do qual se pretende criar registos temporais (por exemplo, 2011/2012, ano a partir do qual existe registo consistente de serviço docente no NONIO), para carregamentos subsequentes deve ter valor 0. Esta variável é então utilizada no processo apresentado na Figura 18 no cálculo do intervalo de anos a gerar para a dimensão de tempo.

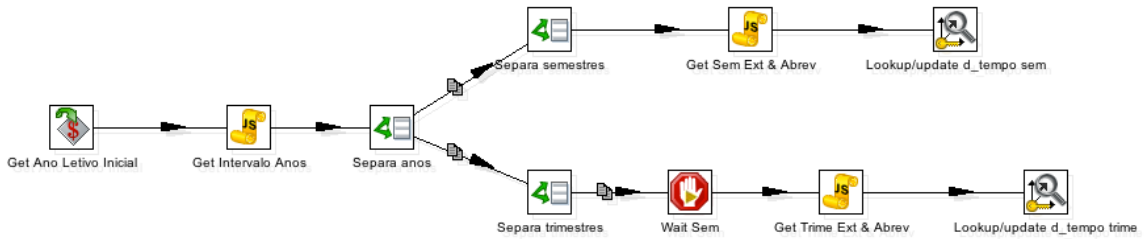

Figura 18 – Carregamento dos registos temporais na dimensão de tempo (*ce_insert_d_tempo.ktr*)

Neste processo de carregamento existem transformações mais simples que outras, isto é, em que os dados são uma correspondência com aqueles que já se encontram nas tabelas da área temporária, exemplo das unidades curriculares – Figura 19.

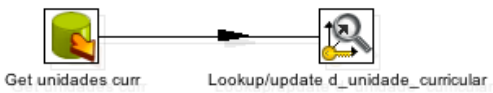

Figura 19 - Carregamento da dimensão de unidades curriculares (*ce_insert_d_unidade_curricular.ktr*)

Contudo existem outras transformações um pouco mais complexas, exemplo dos cursos, onde existe necessidade de unir duas tabelas da área temporária para assim manter uma lista mais completa dos mesmo – Figura 20.

É também neste processo que é criado o campo que armazena a designação do ciclo de estudo a ser utilizado na análise, exemplo "Mestrado (Integrado)", união da categoria e grau dos cursos.

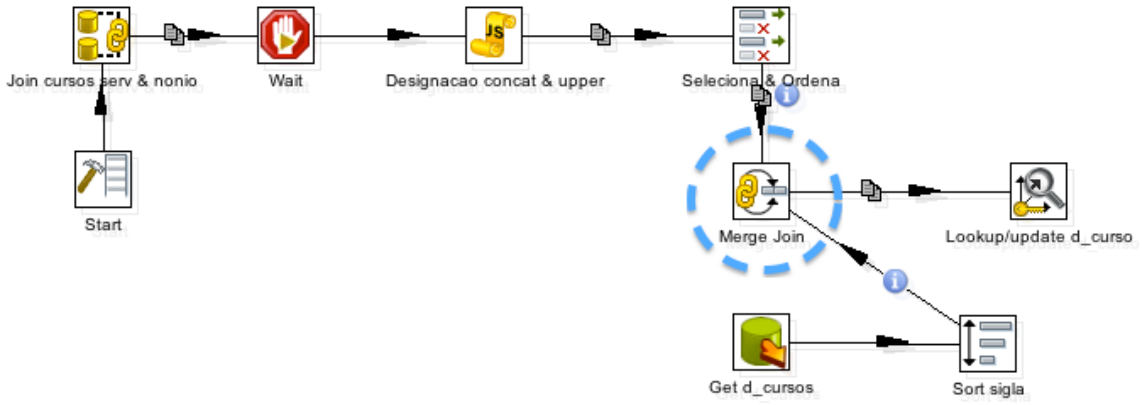

Figura 20 - Carregamento da dimensão de cursos (*ce_insert_d_curso.ktr*)

Comparativamente ao carregamento da área temporária, o processo de carregamento das dimensões apresenta menor complexidade, aspeto que reflete o tempo de carregamento (apresentado na secção 4.1.6.)

4.1.5. Carregamento dos factos

A última fase de todo o processo ETL é o carregamento das tabelas de factos, o *job* correspondente é apresentado na Figura 21. Ao contrário do que acontece no processo de carregamento das dimensões, não existe uma relação direta entre o número de tabelas de factos e transformações existentes. Esta situação deve-se ao facto da inserção de funcionários docentes e não docentes ser efetuada em transformações distintas. Isto acontece porque inicialmente não estava previsto o armazenamento dos funcionários não docentes à mesmo granularidade dos docentes (tabela *ce_f_funcionarios*), sendo que estes seriam apenas uma parcela já agregada na tabela de factos *ce_f_outras_parcelas*. Para análises futuras, considera-se vantajoso manter, desde já, esta informação no *data mart*.

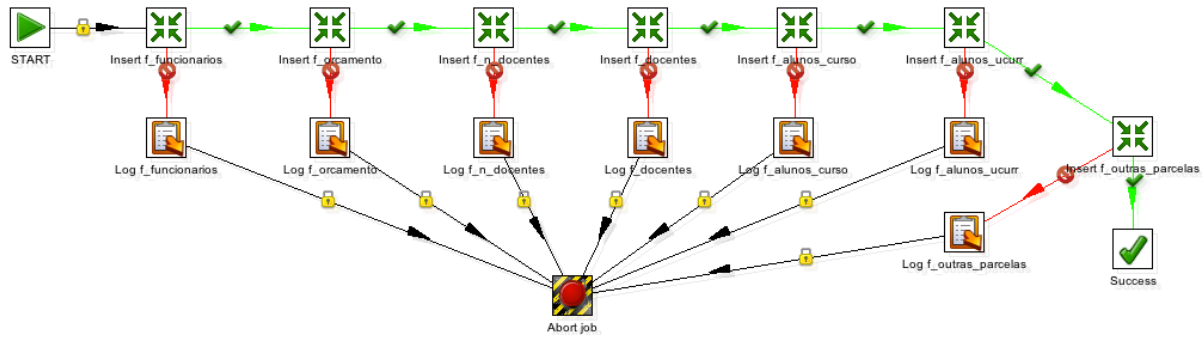

Figura 21 - Processo de carregamento dos factos (*ce_job_carregamento_facto.kjb*)

A complexidade no carregamento das tabelas de factos é bem maior quando comparada às dimensões, exemplo disso é a transformação apresentada na Figura 22 e Figura 23 para o carregamento da tabela *ce_f_funcionarios* para os funcionários docentes.

Aquando os carregamentos para as tabelas de factos é necessário obter todos os identificadores únicos das respetivas dimensões. Para esta tarefa é necessário cruzar informações nas diversas tabelas da área temporária e, por isso, esta pesquisa é o que ocupa grande parte do processo. O caso da tabela aqui exposta é necessário obter nove chaves.

A transformação começa por obter o tipo de carregamento que deve ser efetuado (primeiro ou subsequente). De seguida, obtém todos os registos da tabela da área temporária relativa ao serviço docente (agrupado por docente, unidade orgânica, unidade curricular, edição da unidade curricular, ano letivo, regime e ocorrência da unidade curricular, curso, grau e categoria) e para cada registo pesquisa pela ordem que se segue, os identificadores dos registos nas dimensões:

1. Unidade orgânica do funcionário (*id_uo_funcionario, id_uo_dep_funcionario*);
2. Demografia do funcionário (*id_demografia*);
3. Unidade curricular (*id_unidade_curricular*);
4. Funcionário (*id_funcionario*);
5. Unidade orgânica do curso (*id_uo_curso, id_uo_dep_curso*);
6. Curso (*id_curso*);
7. Tempo (*id_tempo*).

Após obtenção dos identificadores é efetuado o cálculo das métricas do docente para cada registo do serviço docente: número de horas, custo total do docente e custo médio por hora.

Por fim, o registo é inserido ou atualizado na tabela de factos, consoante a existência de alterações ou não no ano letivo imediatamente anterior ao que está a ser carregado.

O processo aqui exemplificado foi o que apresentou mais obstáculos ao nível do desenvolvimento do mesmo, não só pelo elevado número de dimensões que caraterizam os factos aqui presentes, mas também pelo cálculo do custo que deve estar atribuído a cada docente tendo em conta a lecionação do mesmo nesse ano letivo.

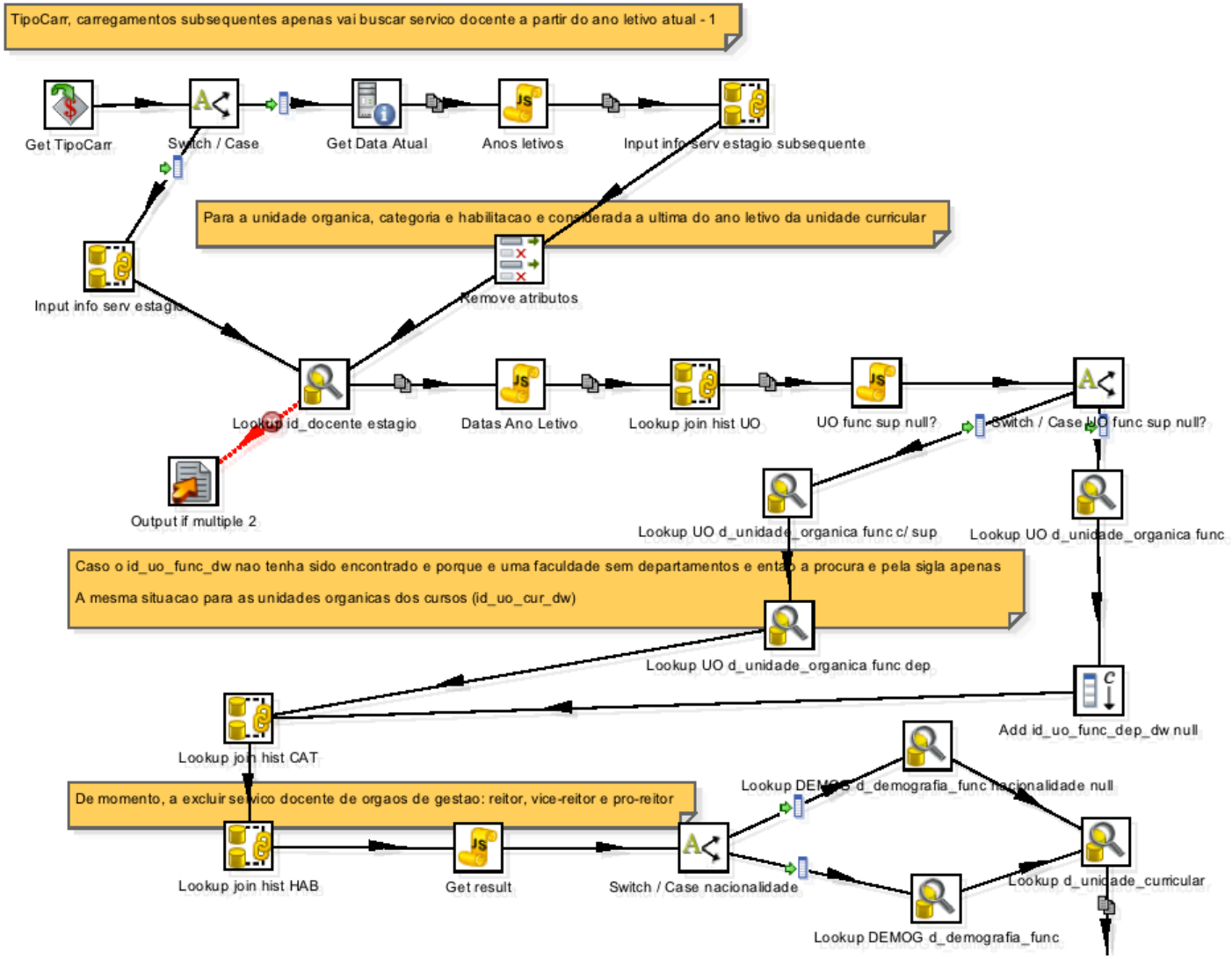

Figura 22 - Processo para carregamento da tabela de factos dos funcionários (1) - *ce_f_funcionarios* (*ce_insert_f_funcionarios.ktr*)

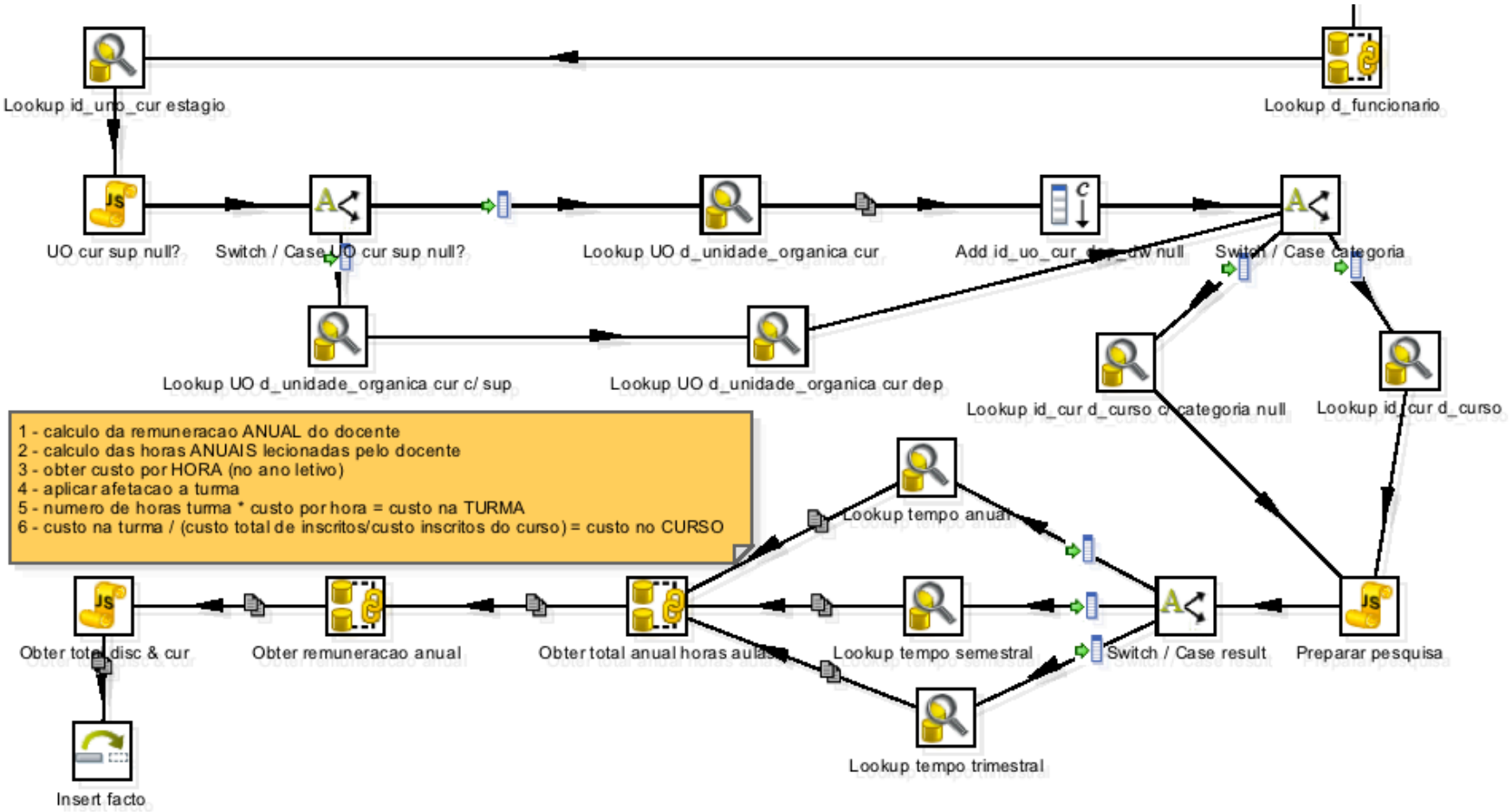

Figura 23 - Processo para carregamento da tabela de factos dos funcionários (2) - *ce_f_funcionarios* (*ce_insert_f_funcionarios.ktr*)

4.1.6. Métricas e volume de dados

Nesta secção são exibidas algumas métricas temporais relacionadas com o processo ETL e volumes de dados nos modelos apresentados anteriormente.

Importante referir as características da máquina responsável pela execução deste processo, pois influencia bastante todo o desempenho e performance do mesmo:

- Processador: *Intel(R) Xeon(R) CPU X5650 @ 2.67GHz*
- Memória (*RAM*): 4 GB

	Primeiro carregamento		**Carregamento subsequente**	
	Tempo	**Volume dados**	**Tempo**	**Volume dados**
Área temporária	4.65 horas	41 MB	3.60 horas	52 MB
Dimensões e factos	44.84 minutos	4896 KB	35.13 minutos	5520 KB
Área temporária, dimensões e factos	5.39 horas	45.78 MB	4.22 horas	57.39 MB

Tabela 20 – Tempos de carregamento e volume de dados no processo ETL

Observando os dados da tabela anterior, destaca-se o elevado tempo de carregamento da área temporária em relação às dimensões e factos no modelo multidimensional, tanto no primeiro carregamento como em carregamentos subsequentes. Estes valores corroboram a complexidade referida nas secções anteriores relativamente às transformações efetuadas na área temporária.

4.2. Cubo OLAP

Como foi referido no capítulo de especificação da arquitetura, a utilização de um cubo OLAP para a análise é uma vantagem. O *Pentaho BI Server* já incorpora a tecnologia *Mondrian* que permite analisar grandes quantidades de dados de forma rápida e eficaz[6].

O cubo é definido através de um esquema num ficheiro XML, que pode ser construído com o auxílio de uma aplicação gráfica desenvolvida especificamente para esta tarefa, o *Schema Workbench*[7] também da *Pentaho* – Figura 24.

O esquema é baseado no modelo multidimensional desenhado para o *data mart* e deve conter a definição de todas as dimensões necessárias para análise. Posteriormente deve ter definidos todos os cubos, que corresponderão às tabelas de factos no modelo multidimensional. Para cada cubo define-se também as métricas e como estas devem ser agregadas - existe possibilidade de criar novas métricas a partir das que existem no modelo multidimensional. É ainda possível definir cubos virtuais, sendo estes por definição o cruzamento de duas tabelas de factos com a mesma granularidade (*drill across*).

[6] http://community.pentaho.com/projects/mondrian/

[7] http://mondrian.pentaho.com/documentation/workbench.php

A utilização de cubos virtuais foi bastante vantajosa para, por exemplo, na análise do custo médio por aluno, em que é necessário cruzar o custo total dos docentes e o número de alunos inscritos que se encontram em tabelas de factos distintas, *ce_f_docentes* e *ce_f_alunos*, respetivamente.

O esquema do cubo é por fim, publicado no servidor de *BI* do *Pentaho* e a partir deste momento podem ser efetuadas sobre o cubo quaisquer consultas em MDX – como a do exemplo da Figura 7.

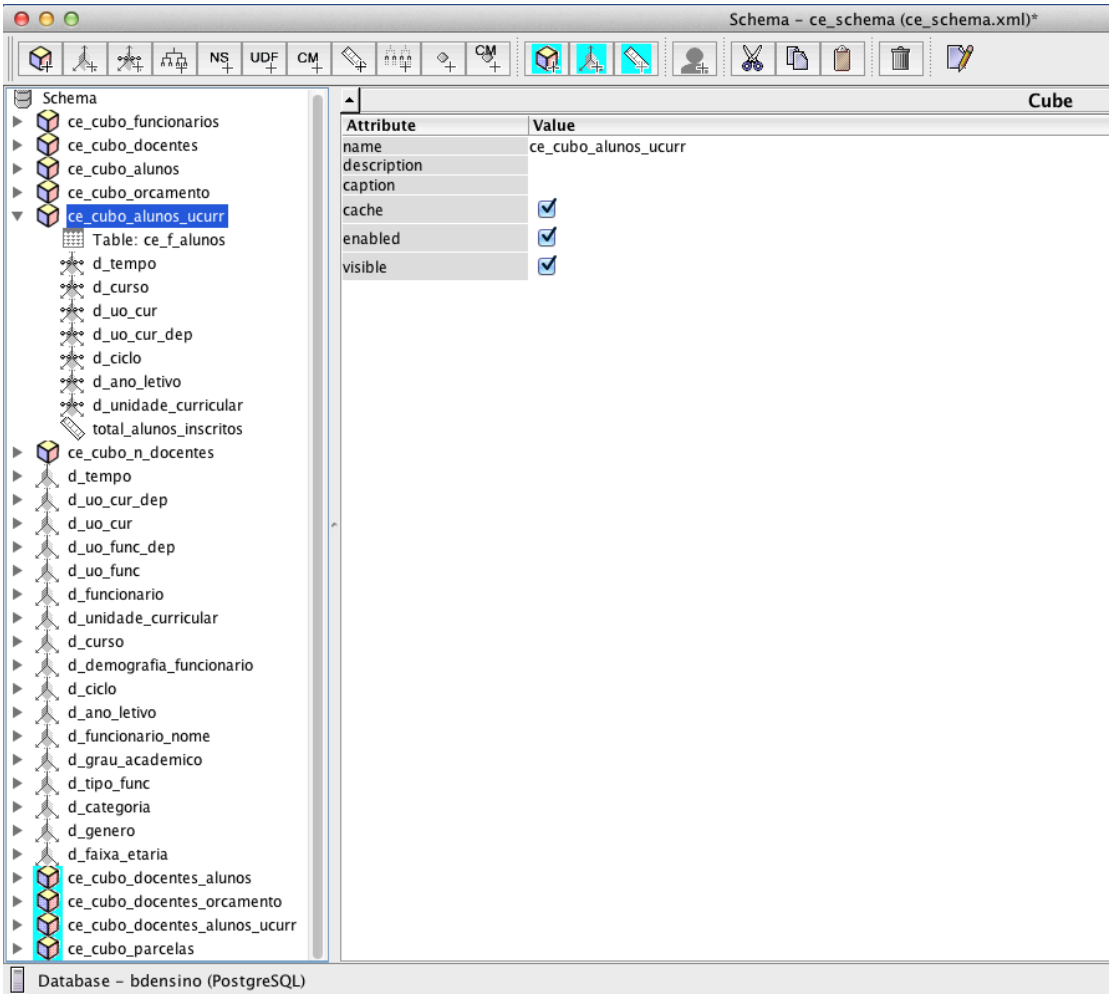

Figura 24 - Definição do esquema para cubo OLAP no *Schema Workbench – Mondrian*

A grande mais valia na utilização deste cubo OLAP do *Mondrian* reside na gestão de memória *cache* que este efetua, armazenando dados de pré-agregados e resultados de consultas efetuadas, permitindo aumento na velocidade e capacidade de resposta do modelo de dados. Com estes pré-cálculos efetuados pelo cubo, esclarece-se o grande volume de dados que é ocupado por todo o *data mart*.

4.3. Dashboards

Nesta secção são abordados detalhes ao nível da implementação dos *dashboards*. Como foi referido no capítulo da arquitetura, foi utilizado o *plugin* disponibilizado pelo servidor de BI da *Pentaho, o CDE (Community Dashboard Editor)*, incluído no ambiente de desenvolvimento do servidor de BI – *Pentaho Console* (Figura 25).

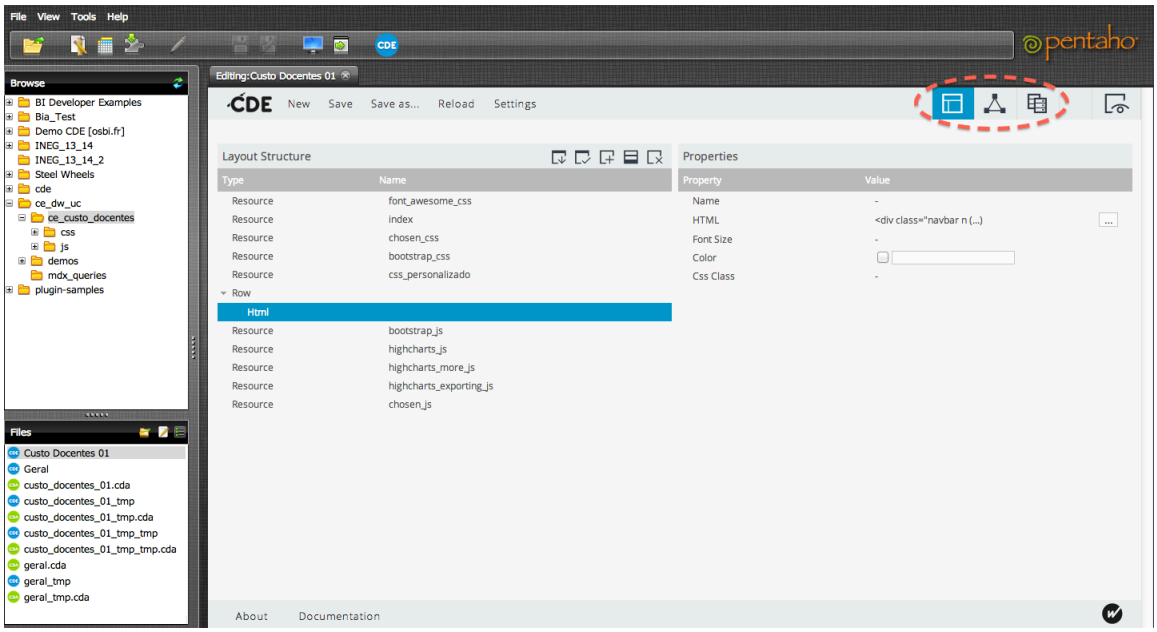

Figura 25 – *Pentaho Console*, ambiente de desenvolvimento do CDE (*Community Dashboards Editor*)

Destaca-se na Figura 25 os menus que permitem uma melhor compreensão sobre o desenvolvimento fazendo uso do CDE: *layout*, componentes e *datasources* (da esquerda para a direita, respetivamente). Na figura encontra-se selecionado o primeiro ícone que diz respeito a todo o *layout* do *dashboard*, aqui é definido o código HTML da página, quais as classes CSS devem ser utilizadas como estilo e quais os *scripts* a ter em conta. Optou-se por fazer uso do estilo *bootstrap*[8] bastante em voga, recorrendo ainda a *plugins* como o *chosen*[9] para as caixas de seleção, o *fontawesome*[10] para os ícones e da biblioteca em *javascript* do *highcharts*[11] para os gráficos mais apelativos e com diversas funcionalidades já incorporadas.

Na Figura 26 o menu selecionado diz respeito aos componentes presentes nos *dashboards*. Existem diversos tipos de componentes, os utilizados nesta implementação foram:

- Caixas de seleção;
- Múltipla seleção;
- Gráficos;
- Tabelas;
- Parâmetros;
- Funções (*javascript*);
- Consultas.

Os componentes de seleção, gráficos e tabelas têm de estar ligados, individualmente, a um elemento definido no menu anterior no HTML do *dashboard*. Esta ligação é definida através da propriedade *HtmlObject* de cada componente - Figura 26.

Os parâmetros têm grande importância, pois com estes é possível criar toda a interatividade entre os componentes. Por exemplo, a alteração do valor da caixa de seleção do ano letivo final tem de afetar a informação que é apresentada nos gráficos, logo tem de existir um parâmetro para guardar o valor associado a esta caixa de seleção. Por sua vez, o parâmetro

[8] *Bootstrap* - http://getbootstrap.com/

[9] *Chosen* - http://harvesthq.github.io/chosen/

[10] *Font Awesome* - http://fortawesome.github.io/Font-Awesome/

[11] *Highcharts* - http://www.highcharts.com/products/highcharts

também tem de estar associado aos componentes dos gráficos. E mais, para que os gráficos detetem a alteração do valor do parâmetro este tem de estar definido *listener* destes componentes.

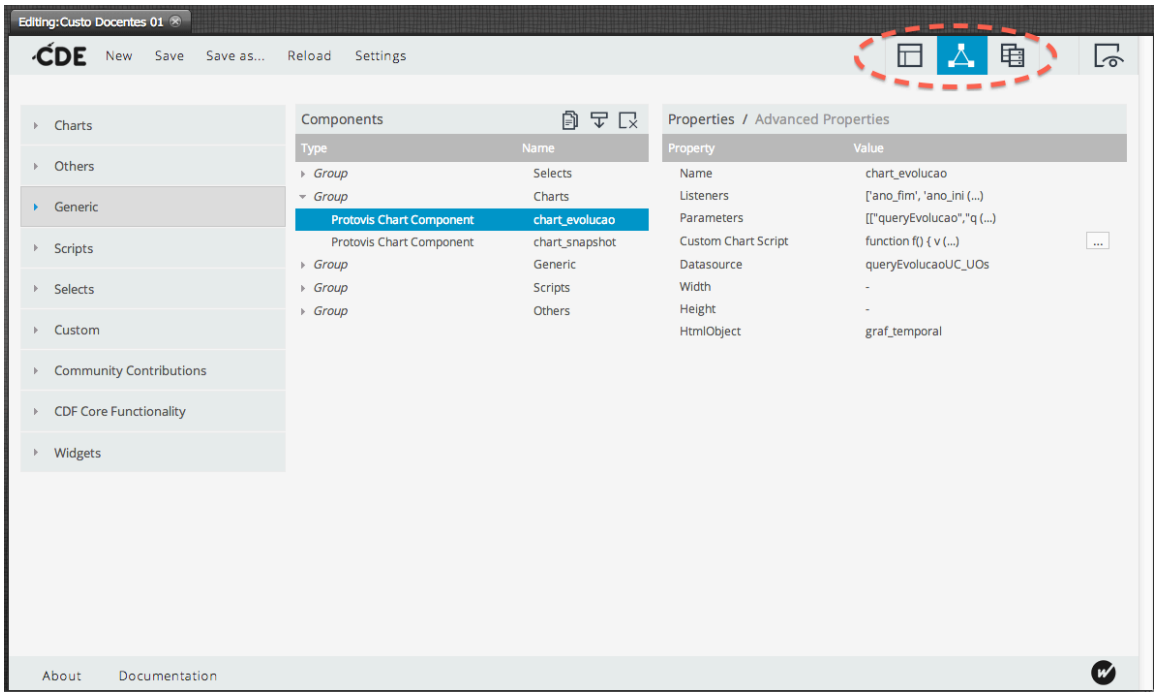

Figura 26 – Ambiente desonvolvimento CDE, menu de componentes

No caso dos componente gráficos, como é utilizada a biblioteca externa do *highcharts* para definir os mesmos, o código *javascript* é colocado na propriedade *Custom Chart Script.*

Por exemplo, cada componente gráfico necessita de dados para utilizar na apresentação do mesmo. Estes dados podem ser provenientes de um *datasource*, indicado na propriedade correspondente e criado no terceiro e último menu do CDE

Como o próprio nome sugere, é no menu *datasources,* que se pode definir as acessos a informação da base de dados e esquemas de cubos OLAP, através de consultas MDX, SQL, entre outros (XML, resultado de transformações do *Pentaho Data Integration,* etc.).

No presente desenvolvimento foram utilizadas as consultas MDX *e* SQL, como se comprova pela Figura 27.

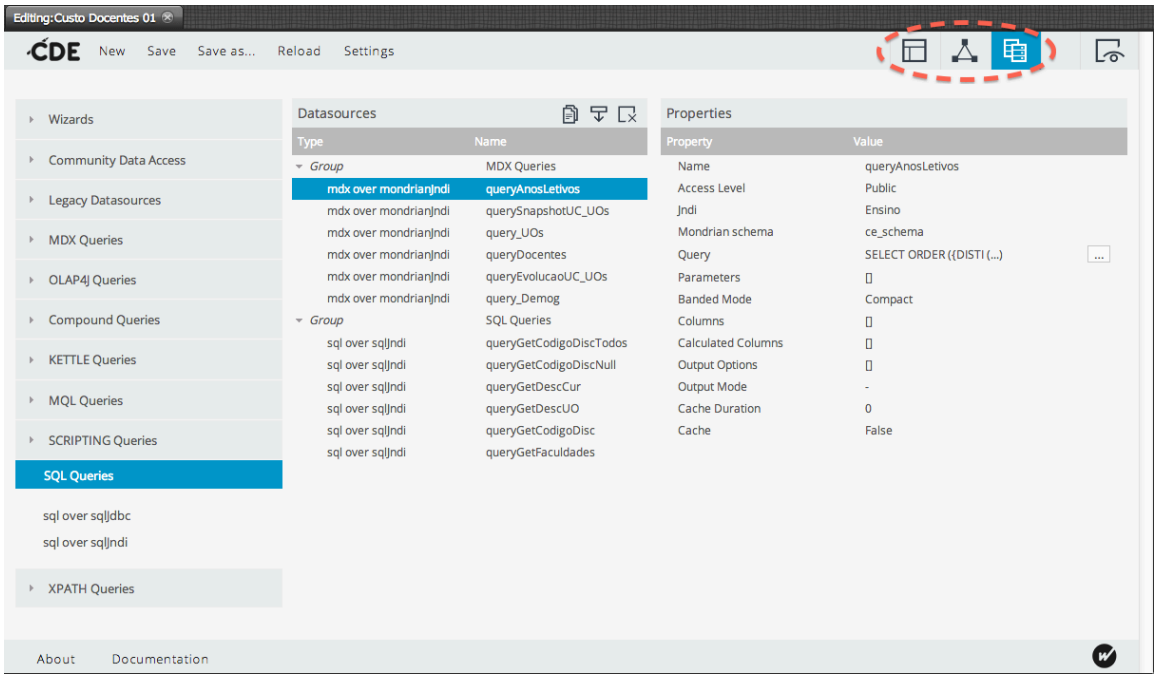

Figura 27 - Ambiente desonvolvimento CDE, menu de componentes

As consultas MDX são efetuadas sobre o esquema do cubo OLAP definido no servidor do *Mondrian*, para cada consulta é necessário definir a ligação (JNDI) e o esquema *(Mondrian schema)*.

No caso das consultas em SQL basta definir qual a ligação a utilizar (JNDI). Foram utilizadas para efetuar consultas bastante simples na BD, como por exemplo obter a descrição de uma unidade curricular através do código.

É na propriedade *Query* que deve ser colocada a consulta a efetuar. Um aspeto a mencionar, é que as consultas também podem receber parâmetros, pelo que na maioria dos casos, a consulta é criada dinamicamente no menu de componentes através de um *Custom Parameter* (função *javascript* que retorna a consulta numa *string*). Para criar a consulta são tidas em conta todas as opções do menu lateral (indicador, filtros e período temporal) e o nível de granularidade. Com este desenvolvimento é possível otimizar e reutilizar código, evitando a criação de uma consulta para cada caso possível abrangido pelas análises disponibilizadas no *dashboard*. Posteriormente é utilizado diretamente como parâmetro na consulta no menu de *datasources* – exemplo através da Figura 28 e Figura 29.

Um exemplo de uma *string* retornada pelo *Custom Parameter* pode ser consultada na Figura 7.

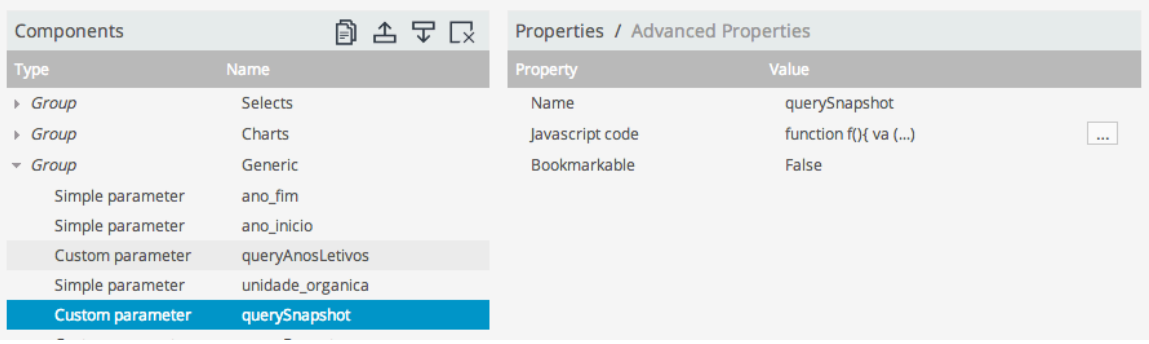

Figura 28 - Definição de um *custom parameter*, retorna consulta *string*

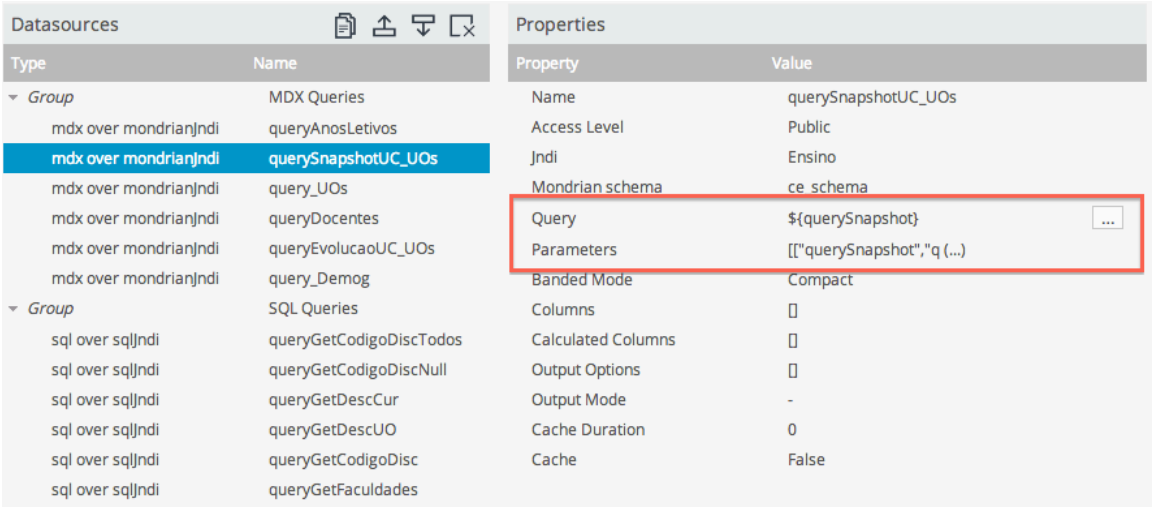

Figura 29 - Utilização do *custom parameter* na consulta MDX

Resumidamente, os menus disponibilizados estão interligados e permitem modular a criação do *dashboard* em três etapas distintas. No final o CDE efetua a compilação da informação presente em todos e cria o ficheiro com a extensão *.xcdf* que contém a definição de todo o *dashboard* – abordado na secção 3.2.4. Análise de dados quando é apresentada a arquitetura do *plugin*.

Na secção seguinte são apresentados alguns exemplos dos *dashboards* desenvolvidos.

4.5. Resultados

Na presente seção são expostos alguns ecrãs, resultado da aplicação final desenvolvida à qual o utilizador final tem acesso atualmente.

O indicador de grande destaque neste módulo é o custo médio por aluno. É o mais solicitado por entidades externas à UC e é muito útil a todos os órgãos de gestão da universidade - Figura 30 –motivos pelos quais é tomado como exemplo nesta secção.

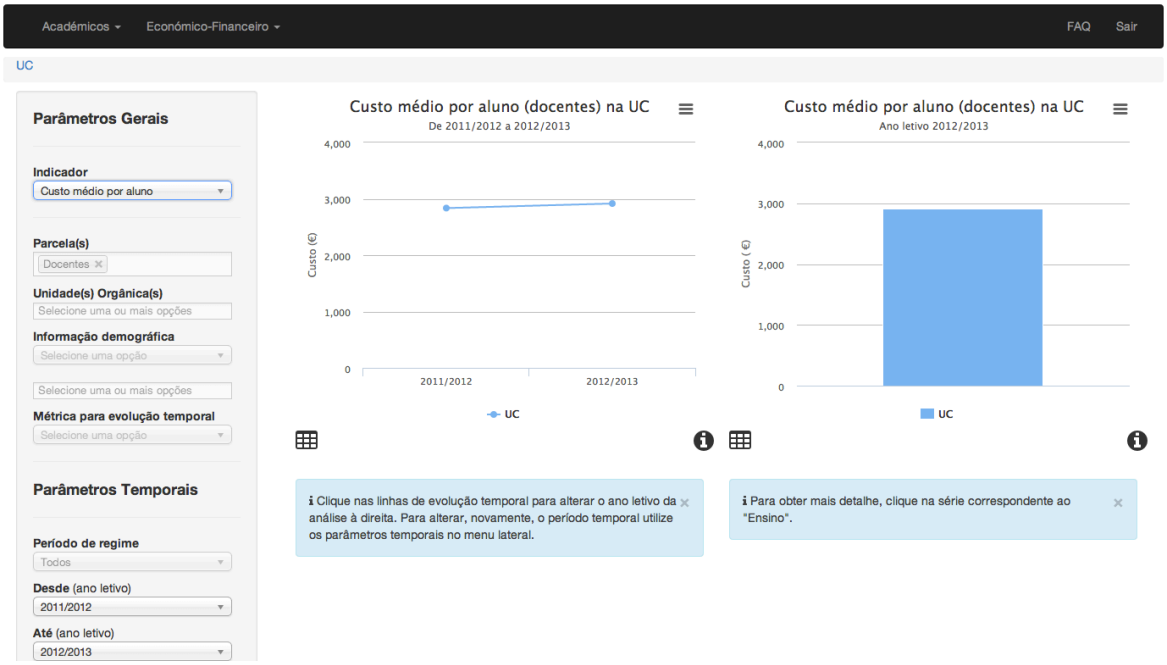

Figura 30 - Ecrã do custo médio por aluno na UC

O nível seguinte de análise é o das unidades orgânicas, onde é possível comparar o custo médio por aluno nas diferentes unidades orgânicas da universidade – Figura 31.

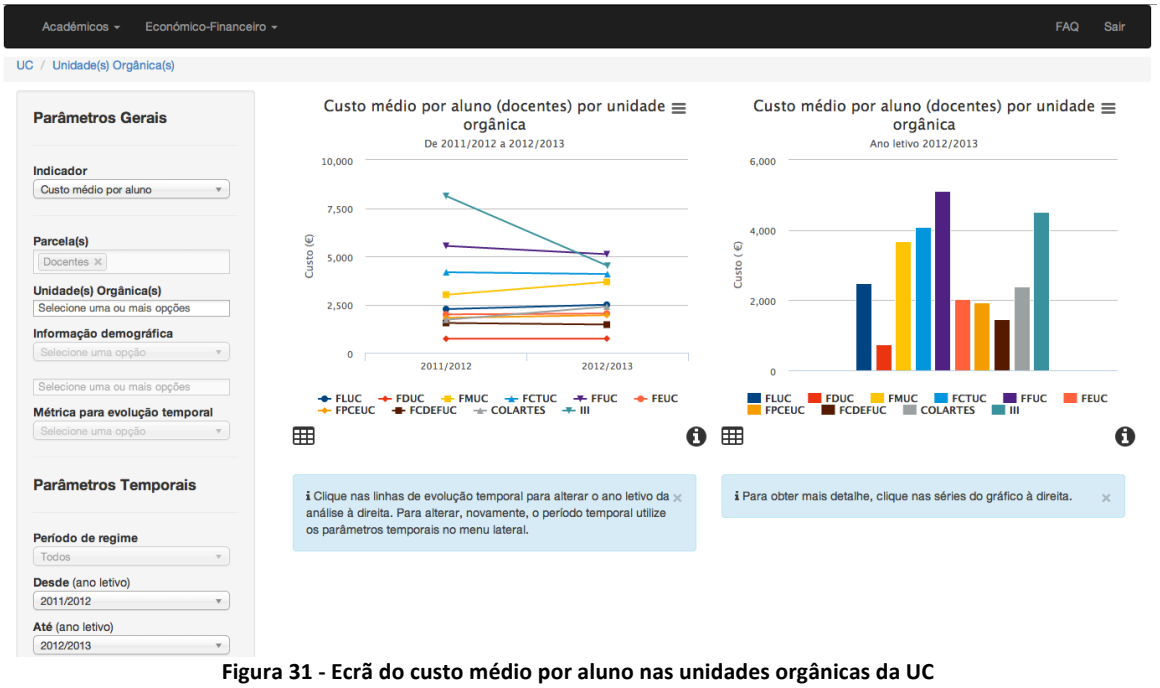

Figura 31 - Ecrã do custo médio por aluno nas unidades orgânicas da UC

Um exemplo da granularidade mínima possível para o indicador aqui apresentado são as unidades curriculares num determinado curso de um ciclo de estudos, num departamento de uma unidade orgânica da UC - Figura 32.

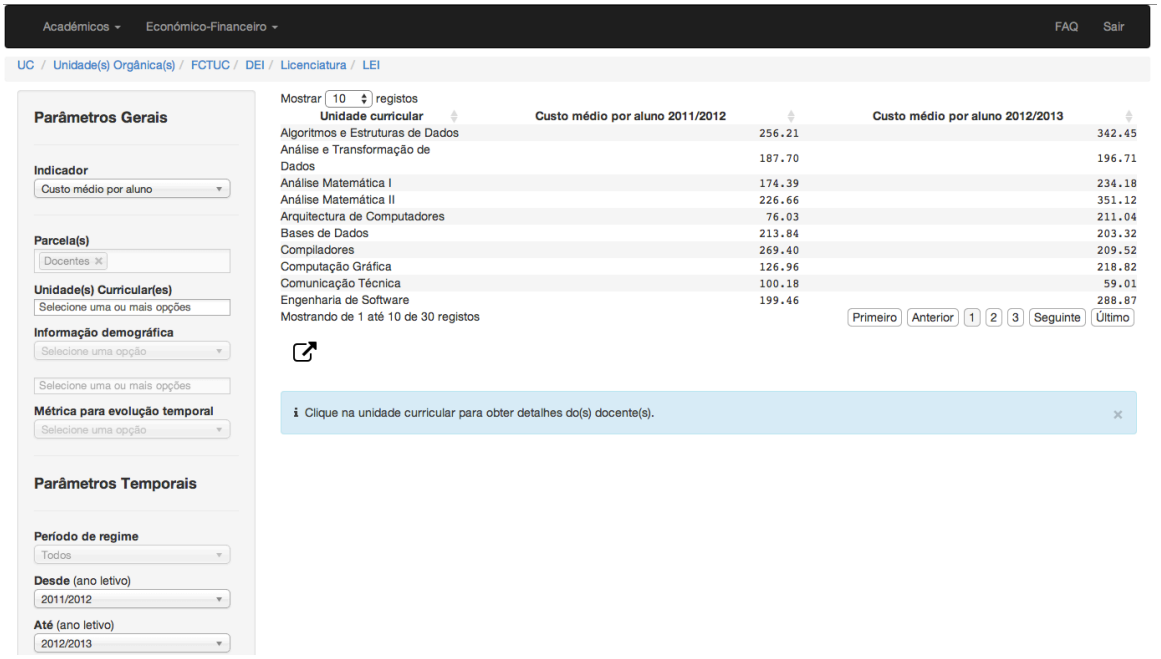

Figura 32 - Ecrã do custo médio por aluno, granularidade mínima

A aplicação final encontra-se desenvolvida segundo os requisitos definidos na primeira fase do estágio. Foi validada e testada, tal como se demonstra no capítulo seguinte, pelos principais *stakeholders* pelo que se encontra disponível para utilização, manutenção e crescimento.

Capítulo 5
Testes e validação

Nesta secção são apresentados a validação dos dados armazenados para cálculo dos indicadores e os testes efetuados para validação das funcionalidades da aplicação final.

A equipa da UC, responsável pela validação do presente módulo**,** é constituída por 3 elementos:

- Chefe de Divisão de Planeamento, Gestão e Desenvolvimento, Filipe Rocha;
- Técnico Superior da Divisão de Orçamento e Conta, Carlos Aguiar;
- Consultor externo, Dr. José Morais.

Dois dos elementos estiveram presentes e participaram da validação dos requisitos funcionais da aplicação e indicadores. Para documentação e auxílio neste processo de validação junto dos referidos *stakeholders* foram elaborados os documentos que podem ser consultados nos anexos [11] e [12], este último ainda é uma primeira versão e permite à universidade arquivar a especificação de cada indicador apresentado no módulo dos custos com o ensino.

5.1. Validação dos dados

Para validar os dados calculados procedeu-se essencialmente à recolha de um conjunto de casos para constituir uma amostra, por forma a verificar se o valor esperado corresponde efetivamente ao que é fornecido pela aplicação final. Estes casos foram selecionados com o auxílio de um elemento da equipa do NONIO, Eng. Ricardo Bica, que apresenta elevado conhecimento dos dados que dizem respeito ao serviço docente.

A validação relativamente à carga horária dos docentes foi também possível através das análises e relatórios já disponibilizados pelo serviço NONIO no *InforGestão* – plataforma de gestão da Universidade de Coimbra.

5.1.1. Casos por amostragem

O caso selecionado, para primeiro exemplo de validação, apresenta a granularidade mais fina, para ser possível efetuar os cálculos necessários. É também o caso que se identificou onde é possível representar todas as situações que tiveram de ser tidas em consideração para o cálculo das horas lecionadas e**,** consequentemente, do custo associado. Foram considerados os seguintes pontos quanto à partilha de horas:

- Unidades curriculares;
- Edições da unidade curricular;
- Curso.

A informação relativa ao caso de exemplo encontra-se na Tabela 21, onde se verificam os seguintes princípios:

- 4 docentes da FMUC;
- 2 cursos (com unidades orgânicas distintas, FMUC e FCTUC – DCV);
- 2 unidades curriculares /edições.

Dados fonte (NONIO e WS)											Análise OLAP			
Ano letivo	Unidade curricular	Edição	Curso	Unidade orgânica curso	Docente(s)	Unidade orgânica docente	Horas lecionadas	Remuneração anual	Custo por hora (anual)	Custo total	Horas lecionadas	Custo total	Custo médio por hora	Resultado da validação
2012/2013	Patologia Forense	157144	Mestrado em Medicina Legal e Ciências Forenses - 6517	FMUC	00016807	FMUC	16.8	8913.72	234.572	3940.810	16.80	3940.80	234.57	✓
					00014429	FMUC	51.2	24972	99.872	5113.446	51.20	5113.45	99.87	✓
	Tanatologia Forense	107379	Doutoramento em Antropologia - 3	FCTUC - DCV	00016634	FMUC	7.2	9193.86	304.130	2189.736	7.20	2189.74	304.13	✓
					00016422	FMUC	4.8	9137.86	102.904	493.940	4.80	493.94	102.90	✓

Tabela 21 - Informação de validação do caso de exemplo: Patologia Forense e Tanatologia Forense

Com o caso de exemplo que se segue, é pretendido mostrar que ao descer na granularidade da aplicação, o cálculo dos agregados é efetuado de forma correta.

- Descer na granularidade da UC até à lista de docentes da unidade curricular Análise Matemática I (AM I), no ano letivo 2012/2013, na Licenciatura em Engenharia Informática – indicador "Custo total (docentes)":

Nível	**Valores apresentados**		**Σ**	**Acumulado nível anterior ⇓**
Docentes	16,422.41		54,095.71	-
	23,742.46			
	13,930.84			
LEI	30 unidades curriculares		1,164,973.84	54,095.71
	AM I	**54,095.71**		
Licenciatura	LDM	579,023.56	1,743,997.37	1,164,973.84
	LEI	1,164,973.81		
DEI	Doutoramento	85,981.89	2,860,153.69	1,743,997.37
	Licenciatura	**1,743,997.50**		
	Mestrado (Continuidade)	966,091.53		
	Mestrado (Formação ao Longo da Vida)	64,082.77		
FCTUC	DARQ	1,246,950.44	30,880,404.39	2,860,153.69
	DCT	1,533,861.38		
	DCV	5,686,308.50		
	DEC	4,820,713.00		
	DEEC	2,385,382.13		
	DEI	**2,860,153.75**		
	DEM	3,104,439.13		
	DEQ	1,548,252.75		
	DF	3,707,764.51		
	DM	1,594,956.86		
	DQ	2,391,621.94		

Tabela 22 – Informação de validação do caso de exemplo: descida na granularidade (1)

Nível	Valores apresentados		Σ	Acumulado nível anterior ⇓
Unidades orgânicas	COLARTES	71,711.62	67,592,210.48	30,880,404.39
	FCDEFUC	1,406,979.13		
	FCTUC	**30,880,408.00**		
	FDUC	2,331,785.25		
	FEUC	5,031,846.40		
	FFUC	7,005,108.87		
	FLUC	8,345,047.66		
	FMUC	8,818,900.38		
	FPCEUC	3,474,766.50		
	III	225,656.67		
UC	**67,592,280.0**		67,592,280.0	67,592,210.48

Tabela 23 - Informação de validação do caso de exemplo: descida na granularidade (2)

5.1.2. Casos discrepantes

Durante o desenvolvimento do processo ETL e validação da aplicação final existiram valores que se destacaram dos restantes, alguns por serem relativamente baixos outros elevados. De seguida é exposto um exemplo, visto como *outlier*, com a respetiva justificação de valores.

O exemplo reflete uma situação em que o docente tem serviço docente registado no NONIO, contudo não existe qualquer registo de pagamentos em SAP - Figura 33. A análise, apesar de parecer inconsistente, pois pode incitar que o docente tem custo zero, essa não é a análise que deve ser retirada. Efetivamente o docente lecionou as horas de aulas apresentadas, não existe é qualquer informação de remuneração paga no sistema fonte.

Figura 33 - Exemplo de docente com horas positivas, sem custo total associado

Estas situações acontecem por diversos motivos, alheios à análise aqui fornecida, pelo que estes casos são considerados como válidos e consistentes de acordo com a informação que se pretende apresentar nesta aplicação específica. Consequência destes casos é que as unidades curriculares a eles associadas apresentam um valor bastante inferior comparativamente às restantes dentro do mesmo curso.

Tomando como referência e exemplo o ano letivo de 2012/2013, existem 28 casos como o apresentado anteriormente, que representam 4932.4 horas registadas sem custo associado. Com base no custo médio por hora na UC para 2012/2013 - 178€, são perdidos com estes casos aproximadamente 877,967.2€, valor consideravelmente significativo para a análise.

Casos como estes refletem a complexidade de reunir informação de diversas fontes de dados, onde nem sempre a informação se mantém atualizada, consistente e em unidade, para que uma leitura de dados corresponda à realidade. Mais uma vez, destacar a importância da aplicação desenvolvida, onde é possível identificar estes casos de forma rápida e eficaz.

5.2. Testes funcionais

Na Tabela 24 e Tabela 25 são expostos um conjunto de testes funcionais, por forma a validar os requisitos solicitados e definidos para a aplicação final. Todos os requisitos de prioridade elevada e média foram implementados e a sua utilização é válida e funcional.

Código	Designação	Teste (descrição)	Validação
RF_GE_01	Autenticação	Efetuar *login*.	✓
RF_GE_02	Fechar sessão	Efetuar *logout*.	✓
RF_GE_03	Término de sessão	Efetuar *login*, permanecer com um ecrã de análise no *browser*, sem efetuar quaisquer ações e após, 15 minutos tentar modificar a análise.	✓
RF_GE_04	Navegação entre módulos	Ir para o módulo "Académico".	✓
RF_GE_05	Navegação interna	Ir para o nível dos departamentos na FCTUC e posteriormente regressar ao nível geral da UC.	✓
RF_GE_06	Parâmetros gerais	No menu lateral, alterar o indicador para custo médio por aluno e a parcela para "Docentes".	✓
RF_GE_07	Parâmetros de tempo	Alterar o período temporal: de 2011/2012 a 2012/2014.	✓
RF_GE_08	Esconder parâmetros	Esconder menu lateral.	✗[12]
RF_GE_09	Secção de ajuda	Abrir a secção de ajuda.	✓

Tabela 24 - Conjunto de testes funcionais (1)

[12] Requisito de prioridade baixa, não houve *budget*, tempo, suficiente para implementar.

Código	Designação	Teste (descrição)	Validação
RF_GE_10	Informação auxiliar	Obter informação auxiliar num dos gráficos de evolução temporal.	✓
RF_GE_11	Visualização: gráfico ⇔ tabela	Apresentar a informação de um gráfico de evolução temporal em forma de tabela.	✓
		Regressar da vista em formato de tabela para o gráfico.	✓
RF_GE_12	Exportar informação na tabela	Exportar informação de uma tabela.	ⓘ[13]
RF_ IN_01	Visão geral UC	Obter informação custo total com a parcela de docentes na UC para o ano letivo 2012/2013.	✓
RF_ IN_02	Custo total	Obter o custo total em docentes na licenciatura em jornalismo no ano letivo 2012/2013.	✓
RF_ IN_03	Custo médio por aluno	Visualizar a evolução do custo médio por aluno no departamento de mecânica.	✓
RF_ IN_04	Custo médio e custo médio, por hora, em docentes	Obter o custo médio em docentes nos cursos de mestrado integrado na FMUC, no ano letivo 2011/2012.	✓
		Obter o custo médio, por hora, em docentes na FMUC no ano letivo 2011/2012.	✓
RF_ IN_05	Dados do custo com docentes (demografia)	Obter custo total dos docentes e número de docentes, por faixa etária, no ano letivo 2011/2012.	✓

Tabela 25 - Conjunto de testes funcionais (2)

[13] Requisito de prioridade baixa, implementado exportar informação, apenas, para *csv*.

5.3. Validação requisitos não funcionais

Nas tabelas que se seguem são apresentados os testes efetuados para os requisitos não funcionais.

Código	Designação	Teste (descrição)	Resultado
RFN_S_01	Atualização de dados	Foi agendada a execução do *job ce_geral* para todos o dia 29 de junho às 00h10 - Figura 34.	Figura 35 ✓

Tabela 26 - Testes para requisitos não funcionais (2)

ce_etl_script

```
#!/bin/bash
clear
cd /opt/data-integration
./kitchen.sh -file="/home/bfragoso/Documents/ce_repo/etl/prod/ce_geral.kjb" --level=Minimal >> /home/bfragoso/Desktop/etl_log.log

[bfragoso@dw-dev ~]$ crontab -l
10 00 29 06 * /home/bfragoso/Desktop/ce_etl_script
[bfragoso@dw-dev ~]$
```

Figura 34 - Agendamento automático do processo ETL

```
2014/06/29 00:10:02 - Kitchen - Start of run.
2014/06/29 00:10:04 - ce_geral - Start of job execution
2014/06/29 00:10:04 - ce_geral - Starting entry [Estagio]
2014/06/29 00:10:04 - ce_job_extracao_estagio - Starting entry [Set tipoURL & tipoCarr]
2014/06/29 00:10:04 - ce_job_extracao_estagio - Starting entry [UO_nonio]
2014/06/29 00:10:04 - UO_nonio - Loading transformation from XML file [file:///home/bfragoso/Documents/ce_repo/etl/prod/ws_sap/ce_input_uo_nonio.ktr]
2014/06/29 00:10:04 - ce_input_uo_nonio - Dispatching started for transformation [ce_input_uo_nonio]
2014/06/29 00:10:05 - Input UO.0 - Finished reading query, closing connection.
2014/06/29 00:10:05 - Trim campos.0 - Finished processing (I=0, O=0, R=115, W=115, U=0, E=0)
2014/06/29 00:10:05 - Input UO.0 - Finished processing (I=115, O=0, R=0, W=115, U=0, E=0)
2014/06/29 00:10:06 - Insert / Update UO.0 - Finished processing (I=115, O=0, R=115, W=115, U=6, E=0)
2014/06/29 00:10:06 - ce_job_extracao_estagio - Starting entry [Habilitacoes]
2014/06/29 00:10:06 - Habilitacoes - Loading transformation from XML file [file:///home/bfragoso/Documents/ce_repo/etl/prod/ws_sap/ce_input_hab.ktr]
2014/06/29 00:10:06 - ce_input_hab - Dispatching started for transformation [ce_input_hab]
2014/06/29 00:10:06 - Get TipoURL.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
2014/06/29 00:10:06 - Pedido SOAP.0 - Optimization level set to 9.
2014/06/29 00:10:07 - Pedido SOAP.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)
```

Figura 35 - Início do ficheiro de *log* da execução do *job ce_geral*, exemplo de teste

Código	Designação	Teste (descrição)	Resultado
RFN_S_02	Compatibilidade (*browser*)	A aplicação web foi testada nos diversos *browsers*: *Internet Explorer 11.0; Firefox 30.0*; *Safari 6.1.3* e *Chrome 35.0*.	ⓘ[14]
RFN_S_03	Compatibilidade (SO)	Testado em sistema *Unix*.	ⓘ[15]
RFN_S_04	Licenças	Realizada validação das licenças de todas as tecnologias e ferramentas utilizadas no desenvolvimento do projeto.	✓
RFN_O_01	Hardware	Verificar características das máquinas, onde se encontra o servidor que disponibiliza a aplicação.	✓

Tabela 27 - Testes para requisitos não funcionais (2)

Os requisitos não funcionais foram cumpridos de acordo com a especificação, a aplicação cumpre assim com os atributos de qualidade atribuídos.

[14] Exportar conteúdo da análise para *csv* não é compatível com o *Internet Explorer 11*.

[15] O sistema operativo selecionado nos servidores para desenvolver e disponibilizar toda a aplicação é baseado em *Unix*.

Capítulo 6
Planeamento

Nesta secção é apresentada a metodologia utilizada para o desenvolvimento do projeto, expostas as atividades desenvolvidas, incluindo alguns detalhes de implementação e resultado final da aplicação.

6.1. Metodologia

No livro *The Data Warehouse Toolkit – The Complete Guide to Dimensional Modeling* de *Kimball*[1] é apresentado um modelo para o ciclo de vida de um projeto de BI, sendo este autor pioneiro na área do BI e DW é esta a metodologia adotada para o presente estágio (ver Figura 36).

No primeiro período do estágio foi realizado trabalho nas seguintes etapas: planeamento do projeto, requisitos, arquitetura e seleção de produtos, especificação da aplicação OLAP e modelação multidimensional. Durante o segundo período foi efetuado o desenvolvimento da área temporária, desenvolvimento da aplicação de análise e foi realizado o *deployment* da aplicação junto dos utilizadores. A partir de agora, será efetuada a manutenção do mesmo.

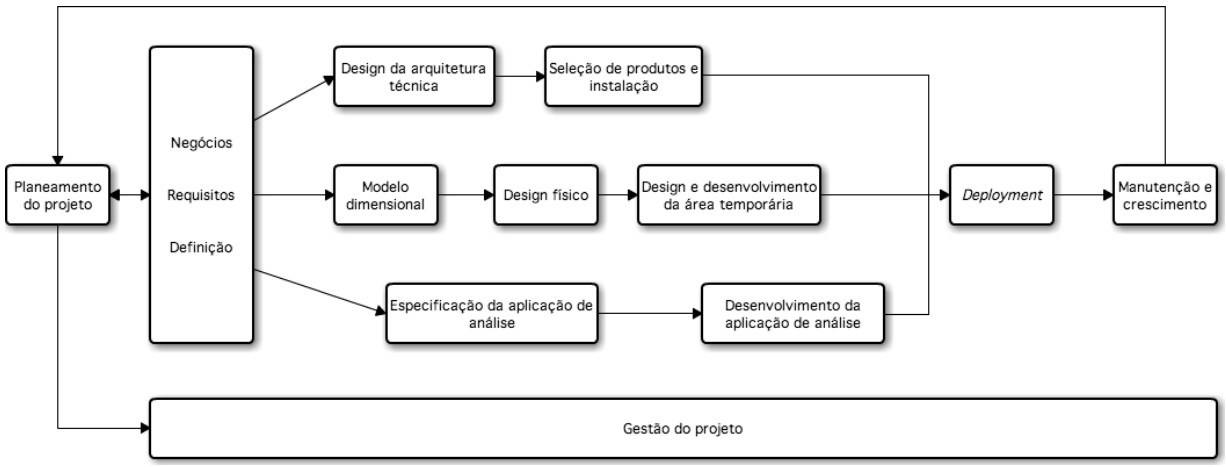

Figura 36 - Ciclo de vida de um projeto de BI

A figura apresenta as atividades num plano de alto nível, estas são especificadas através dos diagramas de *gantt* facultados no anexo [4].

6.2. Plano de trabalho

Relativamente à gestão de projeto foi efetuado um planeamento para o primeiro período de estágio, que consistiu no desenho da aplicação a desenvolver, levantamento e análise de requisitos e modelação. No segundo período de estágio o planeamento incidiu no desenvolvimento dos requisitos propostos e respetiva validação.

Em ambas as fases o planeamento sofreu alguns ajustes. Estes devem-se a atividades que surgiram de forma inesperada. No primeiro período de estágio relacionadas com a validação dos protótipos e requisitos junto dos *stakeholders* e no segundo período relacionados com:

- Atrasos no desenvolvimento pela complexidade na construção do processo ETL;
- Obstáculos relacionados com a memória *cache* da tecnologia *Mondrian*, onde foi necessário investir tempo considerável para encontrar uma solução eficaz e fiável;

- Questões levantadas pelos *stackeholders* relativamente à autenticação, onde também foi investido uma grande quantidade de tempo para solucionar o novo requisito solicitado: restrição por grupos de utilizadores a cada módulo, consequentemente também ao módulo dos custos com o ensino.

Todos os aspetos apontados conduziram também a uma reorganização das tarefas e metas definidas (diagramas de *gantt* no anexo [4] - Figura (anexo) 5 e Figura (anexo) 6).

Para melhor acompanhamento do trabalho a ser desenvolvido foram realizadas reuniões semanais, com o orientador e colegas responsáveis pelos restantes módulos, incluindo o orientador por parte da equipa de trabalho do NONIO – Eng. Pedro Pinto – orientador do módulo do sucesso escolar. A gestão, acompanhamento e monitorização do projeto por parte dos orientadores foi também possível a partir do final do mês de novembro, através do *Redmine* – ferramenta baseada na web que permite efetuar gestão de projetos.

Sendo este um projeto a desenvolver paralelamente com outros módulos existiram decisões e atividades que foram realizadas em conjunto. Para além da definição dos requisitos não funcionais já mencionada neste documento destacam-se ainda:

- Especificação do *design* geral da aplicação OLAP final, documento disponível em anexo - DOC_ESPECIFICACAO_DESIGN_10-01-2014.pdf (anexo [6]);
- Especificação de protótipos geral (documento que serviu de base para a validação dos protótipos e requisitos), também disponível em anexo - DOC_ESPECIFICACAO_PROTOTIPOS_ALL_22-12-2013.pdf (anexo [7]).

Para melhor partilha de informação e colaboração, a equipa de trabalho criou um repositório com o serviço da *Dropbox*. Como um dos elementos se encontrava no estrangeiro foi criado um grupo fechado através de uma rede social, para sempre que fosse necessário fornecer informações rápidas e de conteúdo reduzido. Pelo mesmo motivo, foram efetuadas algumas reuniões de equipa através de vídeo conferência (*Skype*).

No primeiro período de estágio foi necessário reunir com elementos do GSIIC, técnico Nelson Costa e Eng. Jorge China, que foram determinantes para obter uma forma de acesso aos dados necessários do sistema SAP. Relativamente ao dados do sistema NONIO foram efetuadas reuniões com dois elementos da equipa do NONIO, Eng. Pedro Pinto e Eng. Ricardo Bica, que desenvolveram e facultaram as vistas necessárias para obter informação do serviço de docente, que é registado nesse mesmo sistema. Como local de trabalho para desenvolvimento do presente projeto, foi junto com a equipa do NONIO, existindo um grande acompanhamento por parte destes dois elementos para uma compreensão rápida dos dados presentes no sistema por eles desenvolvido.

O desenvolvimento do projeto no segundo período de estágio incidiu em três focos: implementação, documentação e testes. A abordagem foi cíclica, isto é, a implementação, integração, testes e documentação foi feita por indicador, definido nos requisitos. Abordagem fiável, pois existindo necessidade de integrar quatro módulos, é também necessário criar e manter a integração desde o início do processo de desenvolvimento, ultrapassando possíveis falhas de forma incremental, aumentando a qualidade da aplicação final. O plano da Figura 37, mostra o planeamento geral das *milestones* no desenvolvimento do projeto.

De realçar a dificuldade em validar as diversas versões ao longo do tempo, junto dos *stakeholders*. Revelou-se ser uma tarefa bastante complexa de efetuar no período de tempo estipulado.

Figura 37 – Metas de desenvolvimento do projeto no período de estágio

Capítulo 7
Conclusões

Com o fim do período de estágio é possível efetuar um balanço geral sobre o mesmo e identificar os passos quanto ao crescimento da solução desenvolvida.

7.1. Balanço do estágio

Durante o desenvolvimento do projeto, foram atingidos todos os objetivos macro planeados:

- Construção de um *data mart* para reunir informação das diferentes fontes de dados – NONIO e SAP – por forma a obter indicadores de gestão na área da atividade de ensino;
- Desenvolvimento de uma aplicação para disponibilizar uma análise OLAP sobre a informação presente no *data mart*.

A aplicação encontra-se a funcionar e está já disponível para um grupo restrito de utilizadores da UC.

O estágio proporcionou elevada experiência a nível pessoal, académico e profissional, através do contato direto com uma equipa de desenvolvimento como a do NONIO. Alguns pontos a destacar:

- O contato com diversos elementos da reitoria e da administração para perceber as suas necessidades e obstáculos, transformam-se num desafio, a obtenção dos dados fonte e validação final, foram sem dúvida os de maior destaque;
- Desde o começo que este estágio representa grande desafio a nível pessoal dada a sua dimensão e valor acrescentado que trará a uma instituição como a UC;
- Aquisição de competências em tecnologias diretamente relacionadas com cubos OLAP e linguagem MDX, que motivaram a autoaprendizagem;
- Motivação adquirida relativamente ao trabalho em equipa e gestão e planeamento de um projeto de sofware em ambiente real.

7.2. Perspetivas futuras

Com o desenvolvimento da presente solução está criada a base para que o projeto cresça em diferentes sentidos: levantamento e apresentação de novos indicadores aos quais o *data mart* consiga dar resposta, restrições de acesso a informação/módulo por unidades orgânicas ou grupos específicos e disponibilização de relatórios periódicos com informação de destaque.

Anexos

[1] Análise de tecnologias

Bases de dados relacionais

Análise comparativa entre as diferentes características do *PostgreSQL* e *MySQL (Oracle)*, para dar a conhecer os pontos positivos e negativos de cada uma das bases de dados em estudo[20][21].

As otimizações são um aspeto bastante importante no desenvolvimento de uma *data warehouse*, os dados têm de ser disponibilizados com um tempo de resposta aceitável ao utilizador e a quantidade de informação cresce muito rapidamente neste tipo de sistemas, é mantido um histórico. O *PostgreSQL* disponibiliza uma maior diversidade de **índices** e de técnicas para **particionamento** (Tabela (anexo) 1) bem como a possibilidade de efetuar **vistas materializadas** sobre os dados (processo de optimização muito utilizado em aplicações OLAP) - Tabela (anexo) 2.

	Sistemas operativos	**Índices**	**Particionamento**
PostgreSQL	▪ *Linux;* ▪ *UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64);* ▪ *Windows*	▪ *Bitmap;* ▪ *Expression;* ▪ *Full-text;* ▪ *GIN;* ▪ *GiSt;* ▪ *Hash:* ▪ *Partial;* ▪ *R-/R+ Tree;* ▪ *Reverse.*	▪ *Composite (Range + Hash);* ▪ *Hash;* ▪ *List;* ▪ *Native replication API;* ▪ *Range;* ▪ *Shadow,*
MySQL (Oracle)	▪ *Multiplataforma*	▪ *Full-text;* ▪ *Hash;* ▪ *R-/R+ Tree.*	▪ *Composite (Range + Hash);* ▪ *Hash;* ▪ *List;* ▪ *Range.*

Tabela (anexo) 1 - PostgreSQL vs. MySQL – Sistemas Operativos, índices e particionamento

Relativamente a outras características e capacidades da base de dados destaca-se no *PostgreSQL* a possibilidade de *backups* o que aumenta a **segurança** dos dados bem como a integridade dos dados. Apresenta também, **elevada disponibilidade e escalabilidade**, que são aspetos importantes, tendo em conta o aumento da informação, na *data warehouse*, ao longo do tempo. O **processamento paralelo** e o acesso em **tempo real** são outras características que tornam esta base de dados uma das mais avançadas - Tabela (anexo) 3. A quantidade de diferentes tipos de dados disponíveis, incluindo *arrays* e tipos definidos pelo próprio utilizador é outro aspeto a distinguir no *PostgreSQL* - Tabela (anexo) 4.

O *MySQL* é mais leve, consequência do reduzido número de funcionalidades. Quanto à **velocidade** de leitura e execução de consultas é superior comparativamente ao *PostgreSQL*. Faz uso de mecanismos de armazenamento como o *MyISAM* e o *InnoDB,* apesar deste

último acrescentar melhorias na **corrupção dos dados** na base de dados, acrescenta *overhead* que afeta a performance.

Outro aspeto diz respeito ao acesso concorrente, tanto o *PostgreSQL* como o *MySQL (InnoDB)* utilizam um mecanismo de *locking* designado de *MVCC*, cada um apresenta algumas variantes mas a ideia base é a mesma: cada acesso à base de dados detém uma cópia temporária da mesma, quaisquer alterações efetuadas não são refletidas, imediatamente, nos restantes acessos.

Quanto a espaço e tamanho de dados, o *PostgreSQL* pode considerar-se inferior como consequência do uso de mecanismo de armazenamento como o *MyISAM* e o *InnoDB,* o *MySQL* consegue fornecer mais espaço para cada tabela - Tabela (anexo) 5 e Tabela (anexo) 6.

Em suma, partindo dos aspetos aqui expostos, o *PostgreSQL* destaca-se pelo elevado número de funcionalidades: tipos de índices, possibilidade de particionamento, vistas materializadas e realização de *backups*. Todas estas características são vantajosas quando se trata de construir e manter uma DW, nomeadamente a longo prazo.

	Outros objetos	**Tabelas e vistas**
PostgreSQL	▪ Cursores; ▪ Domínio de dados; ▪ Rotinas externas; ▪ Funções; ▪ Procedimentos; ▪ *Trigger*.	▪ Vistas materializadas; ▪ Tabelas temporárias.
MySQL (Oracle)	▪ Cursores; ▪ Rotinas externas; ▪ Funções; ▪ Procedimentos; ▪ *Trigger*.	▪ Tabelas temporárias.

Tabela (anexo) 2 - PostgreSQL vs. MySQL – Outros objetos, tabelas e vista

	Capacidades sobre a base de dados	Características
PostgreSQL	▪ *Blob (Binary Large Object) and clob (Character Large Object);* ▪ *Common table expressions;* ▪ *Except;* ▪ *Inner joins;* ▪ *Inne selects;* ▪ *Intersect;* ▪ *Merge joins;* ▪ *Outer joins;* ▪ *Parallel query;* ▪ *Union;* ▪ *Windowing functions.*	▪ ACID; ▪ *Backup;* ▪ Funções customizadas; ▪ Importar base de dados; ▪ Exportar e importar dados; ▪ Extensível; ▪ Elevada disponibilidade e escalabilidade; ▪ Suporte para *Java* e *multi core;* ▪ Processamento paralelo; ▪ Acesso em tempo real; ▪ Integridade referencial; ▪ *Templates;* ▪ Transações; ▪ *Unicode;* ▪ Suporte para o formato XML.
MySQL (Oracle)	▪ *Blob (Binary Large Object) and clob (Character Large Object);* ▪ *Inner joins;* ▪ *Inne selects;* ▪ *Merge joins;* ▪ *Outer joins;* ▪ *Union.*	▪ ACID; ▪ Suporte para *Java;* ▪ Integridade referencial; ▪ Transações; ▪ *Unicode.*

Tabela (anexo) 3 - PostgreSQL vs. MySQL – Capacidades sobre as bases de dados e características gerais

	Tipos de dados
PostgreSQL	▪ BIGINT, INTEGER, SMALLINT ▪ DOUBLE PRECISION, REAL ▪ *DECIMAL, NUMERIC* ▪ *CHAR, CHARACTER, CHARACTER VARYING, TEXT, VARCHAR* ▪ *BIT, BOOLEAN* ▪ *BYTEA* ▪ *DATE, INTERVAL, TIME, TIMESTAMP* ▪ *ARRAYS, BIT, CIDR, CIRCLE, ENUM, GIS, INET, MASCCADDR, MONETARY, PATH, POLYGON, SEQUENCE, USER DEFINED DATA TYPES, USER DEFINED TYPES, UUID, XML*
MySQL (Oracle)	▪ *BIGINT 64 bit, INTEGER 32 bit, MEDIUMINT 24 bit, SMALLINT, TINYINT 8 bit* ▪ *DOUBLE 64 bit, FLOAT* ▪ *DECIMAL* ▪ CHAR, TEXT, VARCHAR ▪ BIT ▪ BINARY, LONGBLOB, MEDIUMBLOB, TINYBLOB, VARBINARY ▪ DATE, DATETIME, TIMESTAMP, YEAR ▪ ENUM, GIS, SET

Tabela (anexo) 4 - PostgreSQL vs. MySQL – Tipos de dados

	Limites tipos de dados			
	Max valor DATE	**Max CHAR**	**Max NUMBER**	**Min valor DATE**
PostgreSQL	5874897	1GB	Ilimitado	-4713
MySQL (Oracle)	9999	64KB (*text)*	64 bits	1000

Tabela (anexo) 5 - PostgreSQL vs. MySQL – Limites tipos de dados

	Limites base de dados				
	Max *Blob/Clob*	**Max BD**	**Max tabela**	**Max linha**	**Max colunas por linha**
PostgreSQL	4GB	Ilimitado	32TB	1.6TB	256-1600 depende do tipo
MySQL (Oracle)	▪ 1GB (*text, bytea*); ▪ 4TB (*pg_largeobject*)	Ilimitado	▪ 256TB (*MyISAM*); ▪ 64TB (*InnoDB*)	64KB	4096

Tabela (anexo) 6 - PostgreSQL vs. MySQL – Limites base de dados

Extração, transformação e carregamento

A análise comparativa das duas ferramentas *open source* mais conhecidas e utilizadas:

- *Pentaho Data Integration*;
- *JasperETL (Talend)*.

Analisando os dados apresentados nas tabelas que se seguem, pode afirmar-se que ambas as ferramentas têm grande potencial e satisfazem os requisitos necessários para desenvolver o processo *ETL* de forma simples e eficaz[22].

Ao nível da **performance**, o *Pentaho Data Integration* é mais rápido na maioria dos casos, pois não depende de quaisquer códigos *Java* nem de um extenso repositório. Contudo no que diz respeito a cálculos e agregados conseguem-se melhores resultados com a ferramenta da *Talend*[22].

	Interface gráfica	**Multiplataforma**
Pentaho Data Integration	▪ Muito intuitiva; ▪ Não unificada entre componentes.	✓
JasperETL (Talend)	▪ Pouco intuitiva, complexa; ▪ Unificada para todos os componentes (baseada no *Eclipse*).	✓

Tabela (anexo) 7 - Pentaho Data Integration vs. JasperETL – Interface gráfica e multiplataforma

	Metadados	*Scripting*
Pentaho Data Integration	▪ Dependem diretamente das ligações às bases de dados/fontes de dados (ficheiros); ▪ Podem ser partilhadas entre transformações e *jobs*; ▪ São armazenados por etapas, logo não é possível a sua reutilização a partir de outros componentes.	▪ *Javascript;* ▪ *Java;* ▪ *SQL;* ▪ *Shell;* ▪ Fórmulas do *OpenOffice.*
JasperETL (Talend)	▪ *Links* para a base de dados e objetos associados; ▪ Armazenados centralmente; ▪ Podem ser partilhados e reutilizados por outros componentes.	▪ *Java;* ▪ *Groovy;* ▪ *SQL;* ▪ *Shell.*

Tabela (anexo) 8 - Pentaho Data Integration vs. JasperETL – Metadados e *scripting*

	Componentes[23][24]
Pentaho Data Integration	Categorizados por: ▪ *Input* ▪ *Output* ▪ *Transform* (qualidade dos dados) ▪ *Flow* ▪ *Utility* ▪ *Lookup* ▪ *Data Warehouse* ▪ *Validation* ▪ *Bulk loading* ▪ *Scripting* ▪ *Statistics* ▪ *Joins* ▪ *Delete* ▪ *Job* (cada projeto no *Pentaho Data Integration* é considerado um conjunto de transformações que podem agregar-se num único *Job*). Permite que a integração se torne mais abrangente às fontes e tipos de dados, sendo que os componentes comportam-se consoante a ligação selecionada (base de dados).
JasperETL (Talend)	▪ Específicos para cada ação/transformação e para cada *input/output* base de dados (*Oracle*, *MySQL*, *Access*, ...) e ficheiros (*JSON, XML, ...)*; ▪ Componentes específicos para a qualidade dos dados: transformações com *strings*, remoção de duplicados, agregados, etc.

Tabela (anexo) 9 - Pentaho Data Integration vs. JasperETL – Componentes

	Repositórios	***Log* e *debug***
Pentaho Data Integration	*Jobs* e transformações são armazenados em ficheiros *XML* o que permite utilizar estes no sistema de ficheiros local ou através duma base de dados para partilha dos mesmos.	▪ Vários níveis de *log*; ▪ Possibilidade de guardar estes *log* com algumas limitações; ▪ Ferramenta de *debug* básica e simples.
JasperETL (Talend)	Tudo é armazenado no sistema de ficheiros local.	▪ Sistema de *log* bastante desenvolvido, fornece métricas, estatísticas; ▪ Podem ser guardados numa base de dados, ficheiro ou concola; ▪ O *debug* é feito através da vista gráfica utilizada pelo *Eclipse*.

Tabela (anexo) 10 - Pentaho Data Integration vs. JasperETL – Repositórios, *log* e *debug*

	Paralelismo	**Reutilização de código**
Pentaho Data Integration	Grande facilidade com a opção de "distribuição de dados" entre os componentes.	▪ A reutilização de código entre os componentes não é possível, pois este está definido para cada um de forma individual.
JasperETL (Talend)	Limitado.	▪ É possível definir código e utilizar o mesmo entre diversos componentes.

Tabela (anexo) 11 - Pentaho Data Integration vs. JasperETL – Paralelismo e reutilização de código

[2] Soluções e mercado do BI

Existem no mercado do BI diversas tecnologias direcionadas para análise de dados OLAP (relatórios, *dashboards*), aqui são apresentadas as mais faladas.

O SAP disponibiliza um conjunto de soluções na área do BI ao nível de relatórios, *dashboards*, , das quais se destacam:

- *Lumira*[5][6][7][8];
- *Crystal Solutions*[4][9][10];
- *BusinessObjects*[3].

Lumira é uma solução de BI disponibilizada pelo SAP (inclusive através da *cloud*) para manipulação e visualização de dados, estes podem ser carregados de diversas fontes de dados, podem ser manipulados ou transformados, podem dar origem a novos dados (métricas) e por fim, pode ser colocada em forma de gráficos, tabelas, etc.. É possível utilizar o conceito de *stories* que permite uma narrativa gráfica para descrever os dados para que seja possível agrupar gráficos num único ecrã (*dashboard*).

A informação gráfica pode ser guardada localmente, impressa e partilhada através de email.

A origem dos dados está restringida a:

- *MS Excel;*
- *CSV file;*
- *SAP HANA;*
- *SAP BW* dados como vistas no *SAP HANA;*
- *SAP BusinessObjects Universe;*
- Consulta em *SQL*.

No que diz respeito à manipulação e limpeza de dados existe uma vasta lista de operações que podem ser feitas sobre os mesmos, alguns exemplos: editar, filtrar, remover, utilizar funções pré-definas, criar agregados, entre outros.

A lista com os tipos de gráficos que podem ser utilizados é também extensa por tipos de gráfico: comparação, percentagem, correlação, tendência e geográficos.

O SAP disponibiliza uma versão gratuita do *Lumira* que contém as seguintes funcionalidades básicas: dados fonte de ficheiros *MS Excel* e *CSV,* manipulação e limpeza de dados e visualização dos gráficos interativos. Mas a restrição da fonte de dados torna-se insuficiente na maioria dos casos e a versão completa representa custos elevados com aquisição e licenças (943€), no valor não estão incluídos quaisquer *upgrades* ou suporte técnico.

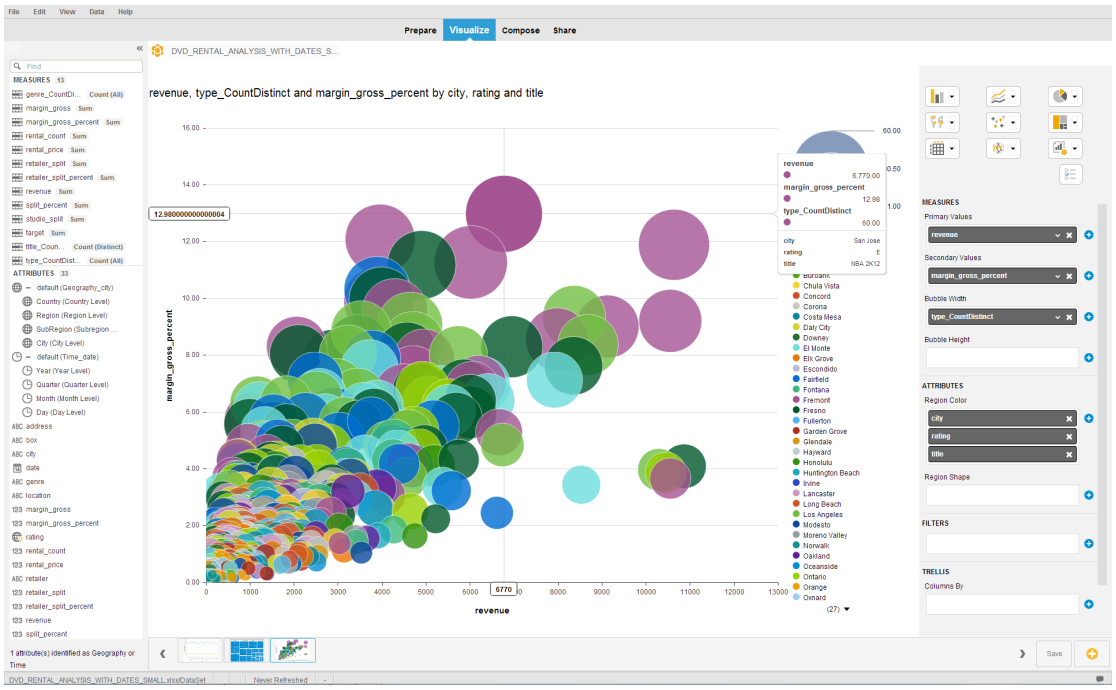

Figura (anexo) 1 - Ecrã exemplo de utilização do SAP *Lumira*

Crystal Server é um servidor de BI que permite obter relatórios, *dashboard* e dados provenientes de todas as fontes dentro da instituição. Contém uma versão mais avançada e ajustada às necessidades do ***Crystal Reports*** q ue permite o desenho de relatórios (existe uma versão deste que pode ser adquirida separadamente) e permite integração com ferramentas da *Microsoft* como o *Office* e *SharePoint*. Permite acesso e partilha da informação a partir de dispositivos móveis que integrem ferramentas como *Extend SAP Crystal Reports* ou *SAP BusinessObjects Data Explorer*. É ainda possível criar e editar *dashboards* com toda a informação utilizando uma outra ferramenta – ***Crystal Dashboard Design*** – esta está disponível em duas edições: *personal* ou *departamental* sendo que a diferença reside na possibilidade de fontes de dados em tempo real que é possível com a edição *personal*.

Existe ainda uma edição do *Crystal Server* direcionada explicitamente para a volumes de negócio mais reduzidos – *Crystal Server, Analytics Edition*.

Os custo de aquisição do *Crystal Server* depende do tipo de licença pretendido: NUL (*Named User License* – cada utilizador tem a sua licença) ou CAL (*Concurrent Access License* – vários utilizadores associados). Importante referir que a aquisição do *Crystal Reports, Crystal Dashboard Design* e *add-ons* para dipositivos móveis não está incluída, têm de ser adquiridos à parte.

Na tabela que se segue são expostos alguns dos custos com as soluções até aqui descritas disponibilizadas pelo SAP:

Produto	**Licença**	**Custo (€)**
Crystal Server	5 NUL	2294.25
Crystal Server, Analytics	5 NUL	2754.25
Crystal Server	5 CAL	6635.50
Crystal Server, Analytics	5 CAL	7963.75
Add-ons para soluções móveis	-	944.15
Crystal Reports	-	550.85
Crystal Dashboard Design	*Personal*	2366.70
Crystal Dashboard Design	*Departmental*	550.85

Tabela (anexo) 12 - Custos com produtos e soluções SAP

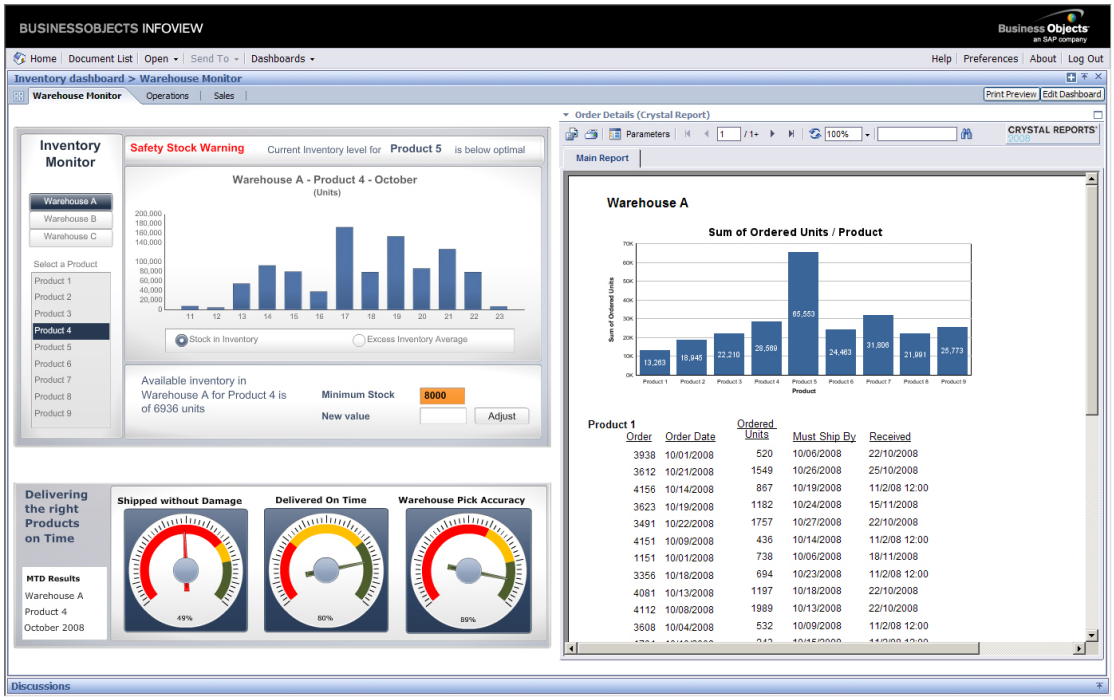

Figura (anexo) 2 – Ecrã exemplo de utilização do SAP *Crystal Server*

BusinessObjects é a BI do SAP - onde estão incluídos alguns dos produtos já descritos - contém as diversas funcionalidades necessárias à criação e manutenção de um projeto de BI: relatórios (utiliza o *Crystal Reports*), consultas a diferente fontes de dados (nativos em sistemas SAP, outras bases de dados relacionais, cubos OLAP, etc.), possibilita os utilizadores de efetuarem pesquisa sobre os dados (*BusinessObjects Explorer*), criação e edição de *dashboard* e previsão (*data mining*). Já se encontra disponível na versão 4.0 e desde a sua existência apresenta uma evolução notória ao nível das funcionalidades.

Como é uma solução implementada à medida de cada instituição, os custos podem variar consoante as necessidades de cada uma.

SAS ("*statistical analysis system*") empresa com soluções de sistemas e serviços direcionados para a análise e gestão do negócio. Apresenta na maioria soluções para disponibilização de dados estatísticos, *data mining* e *forecanting*. Na área do BI e visualização de informação gráfica fornece algumas soluções:

- *Analytics Pro*[11];
- *Enterprise BI Server*[12].

O ***Analytics Pro***, é uma das soluções mais populares da SAS, combina três dos produtos mais utilizados, da SAS, em dados estatísticos: *Base* SAS, SAS/STAT e SAS/GRAPH. Fornece uma linguagem de programação 4GL de fácil aprendizagem para que qualquer utilizador possa utilizá-la para recolher, manipular, limpar e armazenar os dados provenientes das mais diversas fontes de dados, suporta SQL. Disponibiliza diversos tipo de métodos estatísticos para análise de dados e possibilita o *drill down* direto nos dados. Ao nível da apresentação gráfica fornece diversos tipo de gráficos, inclusive com interatividade e *drill down*, podem ser incorporados em páginas web ou em ferramentas da *Microsoft Office*. A informação apresentada em gráficos está disponível também em formato de tabelas.

Ao contrário do SAP *Lumira*, não existe qualquer versão gratuita (pode solicitar-se período experimental). Quanto a custos de aquisição rondam os 6000€ por licença, apesar de existirem preços especiais para quando o número de licenças é superior a 3 ou quando se trata de instituições governamentais ou educação. O suporte e manutenção estão incluídos.

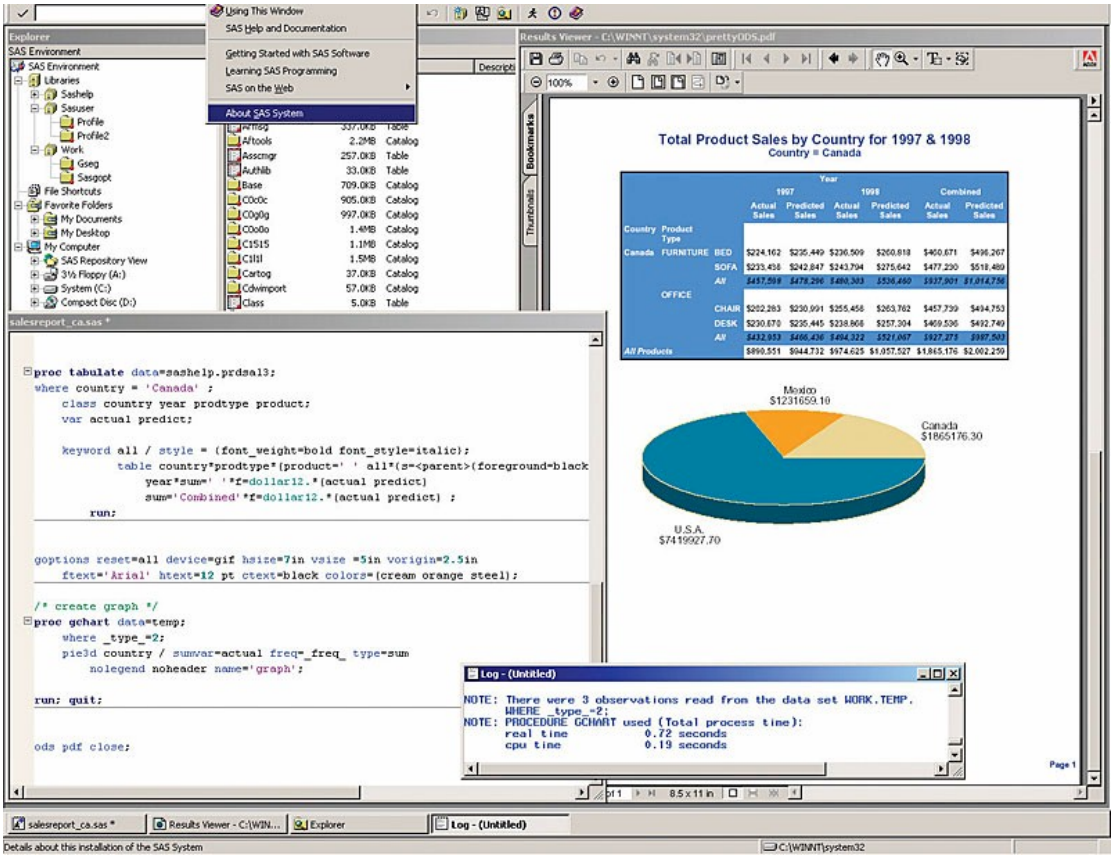

Figura (anexo) 3 - Ecrã exemplo de utilização do SAS *Analytics Pro*

O ***Enterprise BI Server*** do SAS é uma solução muito mais completa no que diz respeito à apresentação da informação, através de um portal web podem ser criados *dashboards* interativos, relatórios, permite que sejam efetuadas consultas em ambiente gráfico com informação provenientes de diversas fontes. Os relatórios podem ser partilhados para diversos destinos, podem também ser integrados com ferramentas da *Microsoft Office* e existe também uma aplicação para dispositivos móveis.

Para aumentar a performance é possível gerir a informação em metadados, permitindo carregar e combinar dados de diferentes fontes.

Através deste servidor é possível desenhar aplicações web de origem, cujo conteúdo são a informação de negócio disponível em *dashboards* e relatórios interativos.

Esta é uma solução completa, concorrente da SAP *BusinessObjects* de forma direta. De igual modo é uma solução que se adapta às necessidades do negócio pelo que o custo também varia de caso para caso.

A IBM fornece um conjunto de produtos da gama *Cognos*, soluções diretamente relacionadas com a área do BI para auxiliar na gestão, monitorização e performance das instituições. Estes produtos encontram-se também categorizados pelo volume de negócio: *Cognos Enterprise*[18] e *Cognos Express*[19] (pequenas e médias empresas).

Relativamente ao ***Cognos Enterprise*** é uma solução de BI completa que trabalha com *big data*, em todos os locais (web, dispositivos móveis e *desktop*) em tempo real. Das diversas capacidades apresentadas as que mais se destacam são:

- Acesso a dados como métricas e possibilidade de interagir com esses mesmos dados através de consultas (filtros, pesquisas, etc.), através de relatórios e *dashboards*;
- Performance elevada no acesso aos dados em modelos OLAP utilizando mecanismos de *in-memory* para acelerar o tempo de resposta das consultas;
- Flexível a alterações nas diversas formas de visualização sem existir necessidade de auxilio neste processo (acrescentar, modificar ou eliminar parâmetros, gráficos, tipos de gráficos, etc.).

Figura (anexo) 4 - Ecrãs exemplo *Cognos Enterprise* no *Desktop* e *Tablet*

Quanto aos custos desta solução também são adaptáveis às necessidades de cada empresa ou instituição, logo variam de caso para caso.

O ***Cognos Express*** é uma solução com objetivos iguais à descrita anteriormente, adaptada a volumes de negócio mais reduzidos. É composta por três produtos: *Reporter, Xcelerator* e *Planner*, através deste é possível:

- Fornecer respostas rápidas sobre a informação através do mecanismo de *in-memory*;
- Informação proveniente de diversas fontes de dados (modelos OLAP, bases de dados relacionais, outros);
- Portal na web, seguro, onde é possível visualizar toda a informação disponibilizada;
- Gestão de planeamento, *budget* e *forecast* de processos efetuados pelas equipas dentro da instituição.

A IBM disponibiliza na sua página oficial um *trial* de 30 dias de utilização desta edição *express*.

ORACLE®

A Oracle, empresa de renome na área das bases de dados, disponibiliza também um cojunto de soluções no âmbito da BI:

- *Oracle Business Intelligence Enterprise Edition 11g*[13];
- *Oracle Business Intelligence Standard Edition One*[16];
- *Oracle Business Intelligence Publisher*[14];
- *Oracle Business Intelligence Foundation Suite*[15].

O ***Business Intelligence Enterprise Edition 11g*** é uma plataforma *web based* de BI e análise de dados, que disponibiliza um vasto conjunto de funcionalidades, das quais se destacam:

- Criação e edição de *dashboards* e relatórios interativos;
- Disponibiliza diversos tipos de gráficos com opções de interatividade;
- Análise OLAP e respetivas opções de pesquisa sobre o modelo (*drill down*);
- Permite criação de *templates* para visualização dos dados adaptados às necessidades da instituição;
- Fornece um sistema de detecção de eventos sobre os dados que gera notificações para o utilizador;
- Possibilita que sejam efetuadas ações nos processos de negócio através dos dados apresentados e do conhecimentos sobre os mesmo;
- Integração dos relatórios e *dashboard* em ferramentas da *Microsoft Office*;
- Disponibiliza aplicações para dispositivos móveis;
- Permite que os dados sejam visualizados através de mapas.

A solução ***Business Intelligence Standard Edition One*** é baseada maioritariamente na solução descrita anteriormente, contudo é direcionada para instituições com volumes de negócio mais reduzidos. Permite ao nível da visualização da informação a criação de relatórios, *dashboard* e análise *ad-hoc*, possibilita o desenho do modelo de dados, do processo ETL e armazenamento na base de dados (faz uso do *Oracle Datawarehouse Builder*).

O ***Business Intelligence Publisher*** é uma solução que permite a criação e manutenção de relatórios e *dashboards* de forma rápida e bastante intuitiva. Encontra-se embutido em ambas as soluções anteriormente descritas (*Business Intelligence Enterprise Edition 11g, Business Intelligence Standard Edition One*).

A solução ***Business Intelligence Foundation Suite*** é a mais completa e que se adapta às necessidades de todas as instituições. De todas as funcionalidades que este incorpora as de maior destaque são:

- *Business Intelligence Enterprise Edition 11g* é o principal componente desta solução, permite a criação e visualização de relatórios, *dashboards* (faz uso do *Business Intelligence Publisher*), análises *ad-hoc*, gestão de metadados e acesso a fontes de dados;
- *Oracle Exalytics In-Memory Engineered System*, sistema para modelagem e análise de dados em tempo real, garantindo elevada performance em todas as consultas efetuadas pelos utilizadores;
- *Oracle Essbase*, é o servidor de OLAP disponibilizado pela Oracle para um armazenamento que permita melhor performance no acesso à informação;
- *Oracle Scorecard and Strategy Management* auxilia a definição de estratégia e objetivos das instituições e estabelece uma ligação entre estes e indicadores de performance e desempenho;
- Integração com ferramentas da *Microsoft Office* e solução para dispositivos móveis.

No que diz respeito a custos, tem de ser adquirido o software e ser paga uma licença para suporte e atualizações durante o primeiro ano. Existem três tipos de licença *Named User Plus* (NUP – licença individual instalada num servidor ou em múltiplos, um dispositivo que utilize ativamente o produto tem uma licença deste tipo), *Processor* (CPU - uma licença para cada processador onde o software é instalado) ou *Employee* (licença por cada empregado ativo associado à instituição), pode ainda ser definido o termo de licença: contínuo ou 1, 2, 3, 4 ou 5 anos (os custos serão muito superiores consoante o tempo). Também é imposto um número mínimo de licenças que deve ser adquirido.

Na tabela que se segue são apresentados alguns exemplos de custos com produtos Oracle, sobre as soluções apresentadas:

Produto	**Licença/Termo (mínimo)**	**Custo(€)**	**Custos adicionais – 1º ano (€)**
Business Intelligence Suite Enterprise Edition Plus – solução completa	20 NUP / Contínuo	1579.00 / NUP	347.29 / NUP
Business Intelligence Suite Enterprise Edition Plus – solução completa	1 CPU / Contínuo	174633.00 / CPU	38419.18 / CPU
Business Intelligence Server Enterprise Edition	20 NUP / Contínuo	276.00 / NUP	60.78 / NUP
Business Intelligence Server Enterprise Edition	1 CPU / Contínuo	40886.00 / CPU	8994.86 / CPU
Business Intelligence Standard Edition One	5 NUP / Contínuo	947.00 / NUP	208.38 / NUP
Business Intelligence Standard Edition One	5 NUP / Contínuo	947.00 / NUP	208.38 / NUP
Business Intelligence Standard Edition One	CPU não disponível	-	-

Tabela (anexo) 13 - Custos dos produtos Oracle (1)

Produto	Licença/Termo (mínimo)	Custo(€)	Custos adicionais – 1º ano (€)
Business Intelligence Foundation Suite	25 NUP / Contínuo	2901.00 / NUP	638.15 / NUP
Business Intelligence Foundation Suite	1 CPU / Contínuo	236790.00 / CPU	52093.80 / CPU
Business Intelligence Publisher	50 NUP / Contínuo	363.00 / NUP	79.88 / NUP
Business Intelligence Publisher	1 CPU / Contínuo	36308.00 / CPU	7987.72 / CPU
Business Intelligence Publisher	1000 *Employee* / Contínuo	36.00 / *Employee*	7.00 / *Employee*
Business Intelligence Mobile	20 NUP / Contínuo	284.00 / NUP	62.51 / NUP
Business Intelligence Mobile	CPU não disponível	-	-

Tabela (anexo) 14 - Custos dos produtos Oracle (2)

A Tableau[17] é mais uma empresa cujo o objetivo é "auxiliar as pessoas a ver e entender dados". Disponibiliza os seguintes produtos:

- *Tableau Desktop;*
- *Tableau Server;*
- *Tableau Online.*

O ***Desktop*** é direcionada a qualquer pessoa, permitindo a criação e visualização de *dashboards* que podem posteriormente ser partilhados com as restantes soluções da *Tableau*. O custo é de 1471.48€ por utilizador.

O ***Server*** funciona então como uma plataforma de BI cuja a estrutura e funcionamento se afasta das tradicionais:

- permite criar e partilhar *dashboards* interativos na web e em dispositivos móveis que podem consultados em tempo real;
- elevada performance no acesso aos dados, mesmo em grande volumes de dados, estes podem ser carregados para o mecanismo disponibilizado que utiliza a memória como reconhecimento da arquitetura. Ou como na maioria das ferramentas pode simplesmente efetuar uma ligação à fonte dos dados e utilizar um híbrido dos dois mecanismos (dependendo dos casos);

- combina diferentes fontes de dados.

Esta é uma solução completa cujo o custo se adapta às necessidades e características de cada empresa ou instituição.

O *Tableau Online* é uma solução idêntica ao *Server* contudo é disponibilizada através da *cloud* permite partilha de dados e *dashboards* de forma muito mais rápida, bem como acesso a estes a qualquer momento. Garante atualizações de dados e segurança. Tem um custo de aproximadamente 368€ por utilizador.

Sumário

Após estudo e análise das soluções existentes no mercado, é apresentada uma análise geral comparativa de todas elas, dado critérios essenciais para a aplicação a desenvolver com o presente projeto.

Recorde-se que o principal objetivo do projeto DW UC, na área do ensino, é desenvolver uma aplicação web, capaz de produzir *dashboards* interativos que contenham informação de indicadores relacionados com o custo com a atividade de ensino na UC.

Os principais critérios para comparação das soluções, foram definidos através da necessidade imposta pelo projeto:

- *Web based* e multiplataforma, a aplicação a desenvolver para o utilizador é disponibilizada na web e deve ser acessível na maioria dos *browsers*;
- *Dashboards* e relatórios interativos, um dos requisitos dos ecrãs é que para além dos diferentes tipos de gráficos a utilizar estes devem ser interativos (utilizador pode selecionar parâmetros e visualizar os dados correspondentes nos gráficos);
- Os dados utilizados para a análise pode ser proveniente de diversas fontes, quer seja uma base de dados ou um ficheiro;
- A partilha e exportação das análises é também um critério a ter em conta, bastante relevante quando existe necessidade de armazenar determinada informação;
- A solução deve apresentar um nível de performance considerável, as análises pretendidas podem ser demasiado complexas e é pouco vantajoso que o utilizador espere demasiado;
- Com o crescimento da utilização de *smartphones* e *tablets*, o suporte para aplicações móveis é outro aspeto a ter em conta;
- Ao nível do desenvolvimento o software deve se de fácil aprendizagem e utilização, simples e intuitivo;
- A solução deve ser de licença gratuita, este é um requisito imposto para o desenvolvimento do projeto.

Na tabela que se segue é apresentado um resumo comparativo das soluções apresentadas anteriormente:

	SAP	SAS	IBM	ORACLE	tableau
Dashboards e relatórios interativos	✓	✓	✓	✓	✓
Diversidade de gráficos	✓	✓	✓	✓	✓
Múltiplas fontes de dados	✓	✓	✓	✓	✓
Desenvolvimento e usabilidade	ⓘ	ⓘ	✓	✓	✓
Web based	✓	✓	✓	✓	✓
Performance	✓	✓	✓	✓	✓
Dispositivos móveis	✓	✓	✓	✓	✓
Partilha e exportação	✓	✓	✓	✓	✓
Multiplataforma	ⓘ	ⓘ	✓	ⓘ	ⓘ
Open source	✗	✗	✗	✗	✗

Legenda:

✓ - cumpre o critério;

ⓘ - apresenta algumas falhas/entraves/incompatibilidade;

✗ - não cumpre o critério.

Tabela (anexo) 15 - Análise comparativa geral

A maioria das soluções apresentadas cumpre efetivamente com a generalidade dos critérios. Todas elas têm grande potencial de mercado e têm evoluído de tempos a tempos numa competitividade estonteante para que as empresas e instituições consigam cada vez mais estar um passo à frente umas das outras.

Contudo, uma das condições para o desenvolvimento e execução do presente projeto é que não pode ser gasto qualquer tipo de *budget* ao nível da aquisição de software e licenças, logo o desenvolvimento terá de ser feito através de ferramentas *open source* que possibilitem a visualização dos indicadores pretendidos e interação sobre os mesmos.

[3] Estimativa do tamanho das tabelas do modelo multidimensional

		Número de registos (estimado)	Tamanho por registo (*bytes*)	Tamanho (≈)
Dimensões	Funcionário	4500	936	4113 Kb
	Demografia de funcionário	8300	815	6606 Kb
	Tempo	48	1352	63 Kb
	Parcela	10	268	3 Kb
	Unidade orgânica	123	800	961 Kb
	Curso	345	1862	627 Kb
	Unidade curricular	5000	1064	5195 Kb
		Total		≈ 17 Mb

Tabela (anexo) 16 - Tamanho estimado das tabelas de dimensões

	Campo	**Tipo**	**Tamanho**
Factos alunos (*ce_f_alunos*)	*n_alunos*	Inteiro	4 *bytes*
	id_uo_curso	Inteiro	
	id_curso	Inteiro	
	id_uo_dep_curso	Inteiro	
	id_tempo	Inteiro	
	id_unidade_curricular	Inteiro	
		Sub total	24 *bytes*
		Overhead	5 *bytes*
		Total (registo)	29 *bytes*
Após 10 anos estima-se 13800 registos, para cada ano letivo a granularidade mais baixa é o curso, logo no limite, 345 cursos × 40 registos de tempo ⇒ 13800 × 29 *bytes* ≈ 400200 *bytes*			

Tabela (anexo) 17 - Tamanho estimado para a tabela de factos de alunos na unidade curricular

	Campo	Tipo	Tamanho
Factos alunos (*ce_f_alunos_cursos)*	*n_alunos*	Inteiro	4 *bytes*
	id_uo_curso	Inteiro	
	id_curso	Inteiro	
	id_uo_dep_curso	Inteiro	
	id_tempo	Inteiro	
		Sub total	20 *bytes*
		Overhead	5 *bytes*
		Total (registo)	25 *bytes*
Após 10 anos estima-se 200000 registos, para cada ano letivo a granularidade mais baixa é o curso, logo no limite, 5000 cursos × 40 registos de tempo ⇒ 200000 × 25 *bytes* ≈ 5000000 *bytes*			

Tabela (anexo) 18 - Tamanho estimado para a tabela de factos de alunos no curso

	Campo	Tipo	Tamanho
Factos docentes (*ce_f_docentes)*	*n_docentes*	Inteiro	4 *bytes*
	n_horas	Real	
	custo_total_docentes	Real	
	custo_medio_docentes	Real	
	custo_medio_hora	Real	
	id_uo_curso	Inteiro	
	id_curso	Inteiro	
	id_uo_dep_curso	Inteiro	
	id_tempo	Inteiro	
	id_unidade_curricular	Inteiro	
		Sub total	40 *byes*
		Overhead	5 *bytes*
		Total (registo)	45 *bytes*
Após 10 anos, 200000 registos, para cada ano letivo tem-se em média 5000 unidades curriculares: 5000 unidades curriculares × 40 registos de tempo ⇒ 200000 × 45 *bytes* ≈ 9000000 *bytes*			

Tabela (anexo) 19 - Tamanho estimado para a tabela de factos de docentes

	Campo	Tipo	Tamanho
Factos funcionários (*ce_f_funcionario*)	*n_horas*	Real	4 *bytes*
	custo_total_docente	Real	
	custo_medio_hora	Real	
	id_funcionario	Inteiro	
	id_demografia	Inteiro	
	id_uo_funcionario	Inteiro	
	id_uo_curso	Inteiro	
	id_uo_dep_funcionario	Inteiro	
	id_uo_dep_curso	Inteiro	
	id_curso	Inteiro	
	id_tempo	Inteiro	
	id_unidade_curricular	Inteiro	
		Sub total	48 *bytes*
		Overhead	5 *bytes*
		Total (registo)	53 *bytes*
Após 10 anos, estima-se, que em média o docente leciona até 2 unidades curriculares em simultâneo num ano letivo: (1900 docentes × 2 unidades curriculares × 40 registos de tempo) + (2500 não docentes × 40 registos de tempo) = 252000 registos × 53 *bytes* ≈ 13356000 *bytes*			

Tabela (anexo) 20 - Tamanho estimado para a tabela de factos de funcionários

	Campo	Tipo	Tamanho
Factos outras parcelas (***ce_f_outras_parcelas***)	*custo_total*	Real	4 *bytes*
	id_curso	Inteiro	
	id_uo_curso	Inteiro	
	id_uo_dep_curso	Inteiro	
	id_tempo	Inteiro	
	id_parcela	Inteiro	
		Sub total	24 *bytes*
		Overhead	5 *bytes*
		Total (registo)	29 *bytes*
Após 10 anos, estima-se 276000 registos, em média 20 parcelas para cada curso, 20 parcelas × 345 cursos × 40 registos de tempo ⇒ 276000 × 29 *bytes* ≈ 8004000 *bytes*			

Tabela (anexo) 21 - Tamanho estimado para a tabela de factos de outras parcelas

	Campo	Tipo	Tamanho
Facto orçamento (***ce_f_orcamento)***	*total_orcamento*	Real	4 *bytes*
	id_tempo	Inteiro	
		Sub total	8 *bytes*
		Overhead	5 *bytes*
		Total (registo)	13 *bytes*
Após 10 anos, estima-se 400 registos, para cada ano letivo um valor para o total do orçamento aprovado ⇒ 400 × 13 *bytes* ≈ 5200 *bytes*			

Tabela (anexo) 22 - Tamanho estimado para a tabela de facto para o orçamento do ano

[4] Planeamento – diagramas

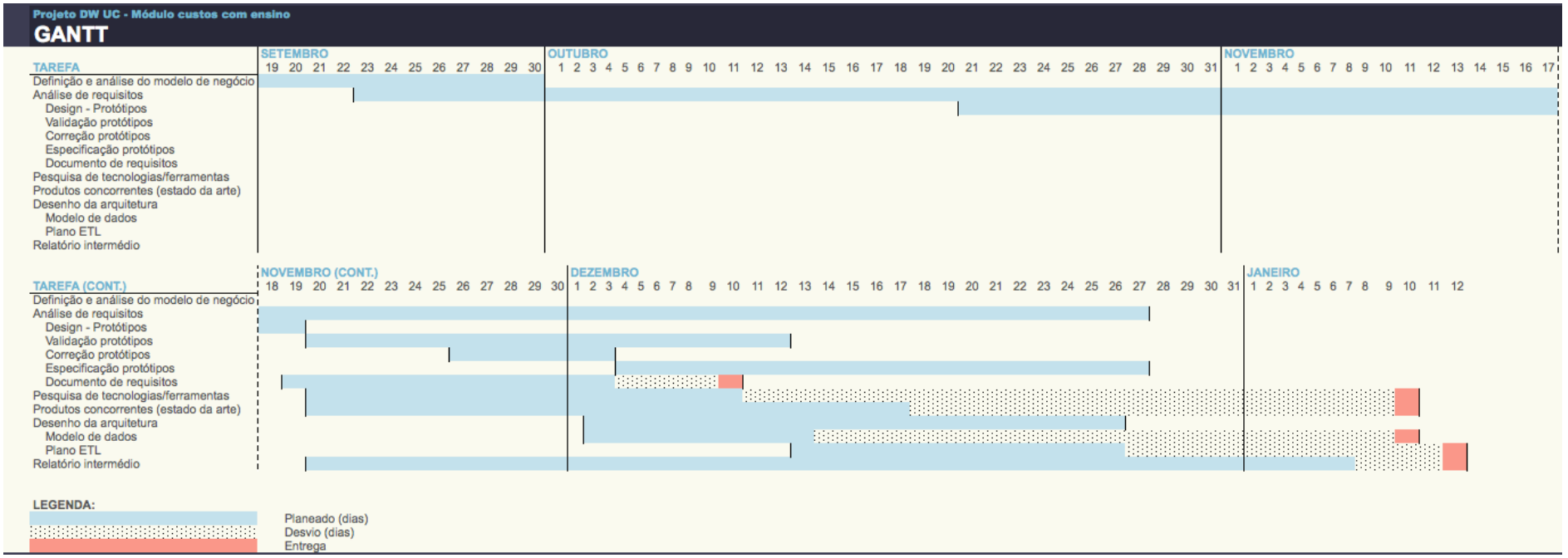

Figura (anexo) 5 - Planeamento e execução do primeiro período de estágio

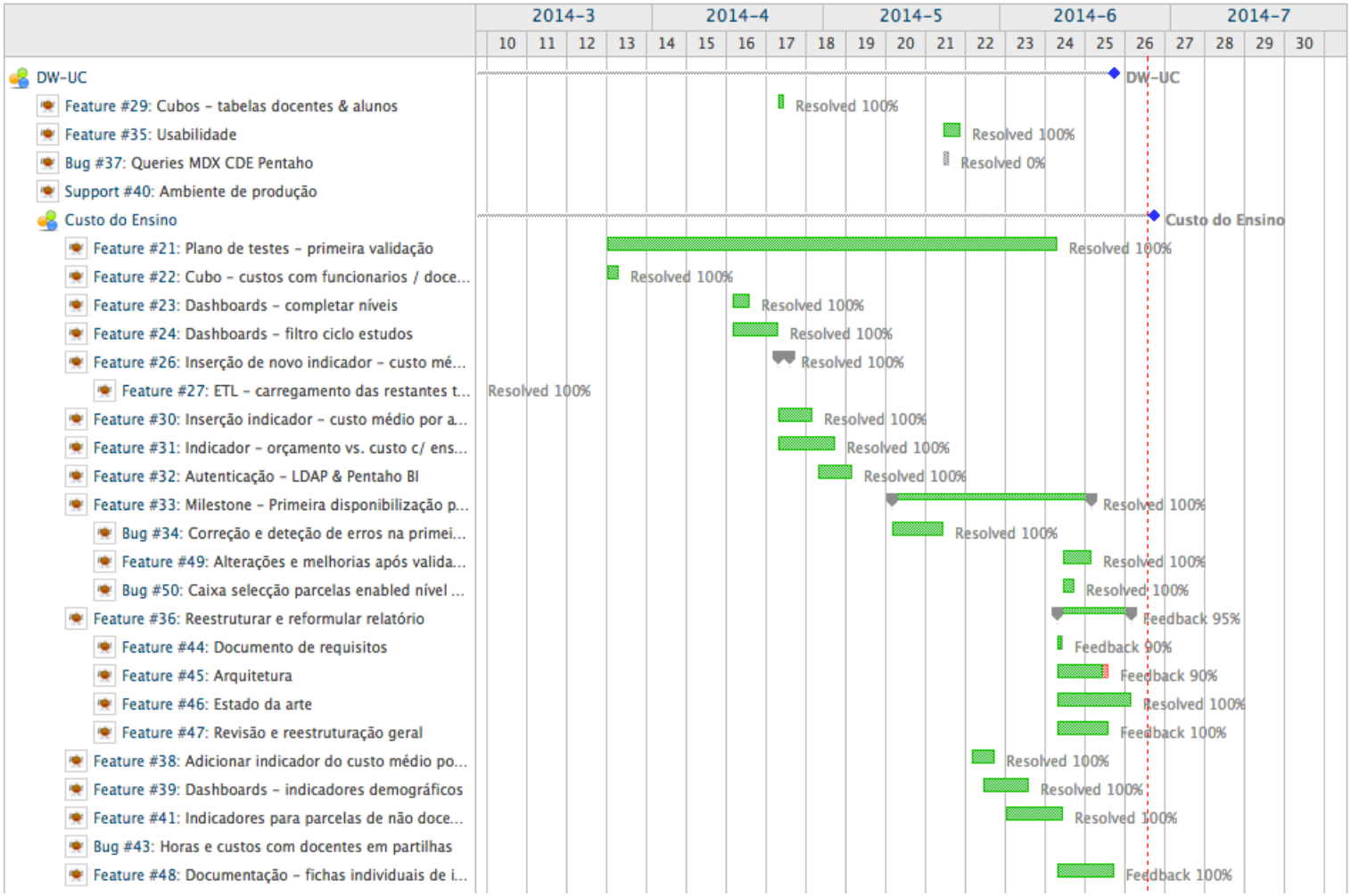

Figura (anexo) 6 - Registo de atividade no *Redmine*

Os anexos aqui listados são fornecidos em formato digital junto com o presente documento.

[5] Documento de especificação de protótipos:
DOC_ESPECIFICACAO_PROTOTIPOS_10-01-2014.pdf

[6] Documento de especificação de design e interação:
DOC_ESPECIFICACAO_DESIGN_10-01-2014.pdf

[7] Documento de especificação de protótipos (conjunto):
DOC_ESPECIFICACAO_PROTOTIPOS_ALL_22-12-2013.pdf

[8] Documento de requisitos: DOC_REQUISITOS_24-06-2014.pdf

[9] Especificação das vistas disponibilizadas pelo NONIO:
NONIO_VistasServiçoDocente_06-12-2013.pdf

[10] Especificação dos serviços disponibilizados pelo GSIIC para aceder aos dados de SAP:
DW_DadosWebServices_v2.pdf

[11] Documento de validação de funcionalidades e dados, distribuído pela equipa da UC:
DOC_VALIDACAO_19-06-2014.pdf

[12] Fichas de indicadores: CE_FICHAS_INDICADORES.pdf

[13] Processo ETL, *jobs* e transformações: CE_TRANSFORMACOES.pdf

Referências

[1] KIMBALL, R., ROSS, M. 2002. The Data Warehouse Toolkit – A Complete Guide to Dimensional Modeling (Second Edition). John Wiley and Sons, Inc., New York, Chichester, Weinheim, Brisbane, Singapore, Toronto.

[2] KIMBALL, R., CASERTA, J. 2004. The Data Warehouse ETL Toolkit – Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley Publishing, Inc., Indianapolis.

[3] SAP (2014). BusinessObjects Business Intelligence 4.0 - Empowering the Real-Time, Mobile, Social, and Global Enterprise.

[4] SAP (2011). SAP Crystal Solutions. Retirado de http://gc.digitalriver.com/store/bobjamer/en_US/DisplayCategoryListPage&categoryID=57065700.

[5] SAP (2013). SAP Lumira User Guide. Retirado de http://help.sap.com/businessobject/product_guides/vi01/en/lum_114_user_en.pdf.

[6] SAP (2014). SAP Lumira Tap into your data – big and small – and discover answers. Retirado de http://www.saphana.com/community/learn/solutions/sap-lumira/sap-lumira-desktop.

[7] Analytics Solutions from SAP. (2013, novembro 19). Market Like Never Before with SAP Lumira. Retirado de http://www.youtube.com/watch?v=89dZNPPYOdk.

[8] Leroux P. (2013, setembro 23). Agile Visualization: What's Coming in Service Pack 12 (SP12) for SAP Lumira?. Retirado a 8 de janeiro de 2014 de http://blogs.sap.com/analytics/2013/09/23/agile-visualization-whats-coming-in-service-pack-12-sp12-for-sap-lumira/.

[9] SAP (2014). SAP Crystal Server, Features & Functions. Retirado de http://global.sap.com/hk/solutions/sap-crystal-solutions/information-infrastructure/sapcrystalserver/featuresfunctions/index.epx.

[10] SAP (2014). SAP Crystal Server, Licensing terms. Retirado de http://global.sap.com/hk/solutions/sap-crystal-solutions/information-infrastructure/sapcrystalserver/licensing/index.epx.

[11] SAS (2012). SAS Analytics Pro, A powerful and comprehensive analytical toolset for analysts, researchers, statisticians, engineers and scientists. Retirado de http://www.sas.com/content/dam/SAS/en_us/doc/factsheet/sas-analytics-pro-103851.pdf.

[12] SAS (2013). SAS Enterprise BI Server, Provides a fast, flexible business intelligence solution. Retirado de http://www.sas.com/content/dam/SAS/en_us/doc/factsheet/enterprise-bi-server-102115.pdf.

[13] Oracle (2013). ORACLE BUSINESS INTELLIGENCE ENTERPRISE EDITION 11g. Retirado de http://www.oracle.com/us/bi-enterprise-edition-plus-ds-078848.pdf.

[14] Oracle (2014). Oracle Business Intelligence Publisher. Retirado de http://www.oracle.com/technetwork/middleware/bi-publisher/overview/index.html.

[15] Oracle (2014). Oracle Business Intelligence Foundation Suite. Retirado de http://www.oracle.com/us/solutions/business-analytics/business-intelligence/foundation-suite/overview/index.html.

[16] Oracle (2014). Oracle Business Intelligence Technology. Retirado de https://shop.oracle.com/pls/ostore/f?p=dstore:2:0::NO:RIR,RP,2:PROD_HIER_ID:4509889075291805720003.

[17] Tableau (2013). Produtos – Tableau Software. Retirado de http://www.tableausoftware.com/pt-br/products.

[18] IBM Corporation (2013). IBM Cognos Enterprise Powerful and scalable business intelligence and performance management. Retirado de http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&appname=SWGE_YT_YU_CAEN&htmlfid=YTD03178CAEN&attachment=YTD03178CAEN.PDF.

[19] IBM Corporation (2014). IBM Cognos Express, Integrated BI and planning for midsize organizations. Retirado de http://www-01.ibm.com/software/analytics/cognos/express/index.html.

[20] FindTheBest (2014). MySQL vs. PostgreSQL in Database Management Systems. Retirado de http://database-management-systems.findthebest.com/compare/30-43/MySQL-vs-PostgreSQL?rnd=1223850141.

[21] Why PostgreSQL Instead of MySQL: Comparing Reliability and Speed in 2007 (2009). Retirado a 7 de janeiro de 2014 da Wiki do PostgreSQL: http://wiki.postgresql.org/wiki/Why_PostgreSQL_Instead_of_MySQL:_Comparing_Reliability_and_Speed_in_2007.

[22] Espinosa R. (2010, 1 de junho). ETL's: Talend Open Studio vs Pentaho Data Integration (Kettle). Comparative. Retirado a 7 de janeiro de 2014 de http://churriwifi.wordpress.com/2010/06/01/comparing-talend-open-studio-and-pentaho-data-integration-kettle/.

[23] Talend (2013). Talend – Components. Retirado de http://www.talendforge.org/components/index.php.

[24] Pentaho Data Integration Steps (2013). Retirado a 7 de janeiro de 2014 da Wiki da Pentaho Community: http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+Steps.

[25] Dietz B., Singh L. (2009, setembro). Open Source BI Reporting Tool Review. Retirado a 7 de janeiro de 2014 de http://timreview.ca/article/288.

[26] Bluter M. (2013, 11 de junho). 5 Free Open Source BI Suites. Retirado a 7 de janeiro de 2014 de http://butleranalytics.com/5-free-open-source-bi/.

[27] Cogswell J. (2012, 18 de dezembro). Pentaho and Jaspersoft: Good Alternates to Bigger-Name Software?. Retirado a 7 de janeiro de 2014 de http://slashdot.org/topic/bi/pentaho-and-jaspersoft-good-alternates-to-bigger-name-software/.

[28] PostgreSQL (2013). Site oficial do PostgreSQL. Retirado de http://www.postgresql.org/.

[29] MySQL (2013). Site oficial do MySQL. Retirado de http://www.mysql.com/products/community/.

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

[31] NoSQL. Site oficial da comunidade. Retirado de http://nosql-database.org/.

[32] Dash J. (2013, 18 de setembro). RDBMS vs. NoSQL: How do you pick?. Retirado a 23 de junho de 2014 de http://www.zdnet.com/rdbms-vs-nosql-how-do-you-pick-7000020803/.

[33] Harrison G. (2010, 26 de agosto). 10 things you should know about NoSQL databases. Retirado a 23 de junho de 2014 de http://www.techrepublic.com/blog/10-things/10-things-you-should-know-about-nosql-databases/.