Mestrado em Engenharia Informática

Dissertação/Estágio

Relatório Final

Projeto UC-Num: Desenvolvimento de uma Data Warehouse para a Universidade de Coimbra – Estágio C

Carlos Mário Ferreira Lênduca

lenduca@student.dei.uc.pt

Orientador:

Professor Doutor Bruno Cabral

Data: 02 de Setembro de 2015

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

Elementos do júri:

Juri Arguente: Henrique Madeira (henrique@dei.uc.pt)

Juri Vogal: Paulo Pereira Carvalho (carvalho@dei.uc.pt)

Agradecimentos

Quero agradecer ao meu orientador, Prof. Dr. Bruno Cabral pela oportunidade de poder desenvolver este projeto. Agradeço a toda equipa da UC-Num pelo apoio e força que me deram. Agradeço a minha namorada, os meus amigos pelo apoio e também agradeço incondicionalmente a minha família por tudo.

Resumo

A Universidade de Coimbra como muitas outras organizações, necessita de sistemas de suporte à decisão. Apesar de existir um sistema de suporte à decisão, os principais *stakeholders* continuam a ter dificuldades ao darem respostas às questões da área académica em tempo útil. Para isso, é necessário identificar o maior número de indicadores de desempenho no âmbito da área académica. Porque quanto melhor for a definição e especificidade dos mesmos, melhor é o sistema de suporte à decisão.

É de salientar que este tipo de sistemas permite à Universidade acompanhar a evolução das atividades académicas, por exemplo, saber se a Universidade continua a atrair um grande leque de estudantes estrangeiros, qual a percentagem de estudantes captados de entre os 25% de melhores candidatos no concurso nacional de acesso, etc. Este tipo de informação permite aos responsáveis obter conhecimento do que acontece na Universidade e se estão a conseguir alcançar os objetivos definidos no plano estratégico e de ação.

O objetivo deste estágio é desenvolver um sistema de suporte à decisão. Assim, passa por construído uma *Data Mart* e uma análise *OLAP* para a área académica da UC. Neste sentido, é desenvolvido o processo *ETL*, o modelo de dados e as análises *OLAP*. Nas análises *OLAP* são desenvolvidos *dashboards* à medida para apresentar a informação em gráficos e tabelas, que dão suporte às decisões dos dirigentes da UC no seu dia-a-dia.

Palavras-Chave

Data Warehouse, Data mart, Dashboard, Cubo, Indicador, Académicos

Índice

Lista de Acrónimos

KPI	Indicador de desempenho (em inglês *Key Performance Indicator*)
UC	Universidade de Coimbra
UC-Num	Universidade de Coimbra em Números
SAMA	Sistema de Apoio à Modernização Administrativa
ETL	*Extract, Transform and Load*
OLAP	*Online Analytical Processing*
SGBD	Sistema de gestão de base de dados
BI	*Business Intelligence*
CPAL	*Common Public Attribution License*
ESB	*Enterprise Service Bus*
MDX	*Multidimensional Expressions*
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
DGES	Direção Geral de Ensino Superior
SGA	Serviço de Gestão Académica
RT	*Request Tracker*
FUc	Ficha da unidade curricular
TSV	*Tab Separated Value*
CSV	*Comma Separated Value*
XML	*eXtensible Markup Language*
JSON	*JavaScript Object Notation*

Lista de Tabelas

Lista de Figuras

Capítulo 1
Introdução

Este relatório de estágio é elaborado no âmbito da Unidade Curricular de Dissertação/Estágio inserido no plano de estudos do Mestrado em Engenharia Informática na Faculdade de Ciências e Tecnologia da Universidade de Coimbra, no ano letivo 2014/2015.

Atualmente, com a crescente utilização dos sistemas de informação e o aumento do volume de dados, a informação tornou-se extremamente valiosa para as empresas de pequeno ou grande porte. Nesta ótica surgem as *Data Warehouses*, que desde 1996[1] têm vindo a ganhar terreno na área de inteligência no negócio, tendo como objetivo o armazenamento massivo de informação e a sua apresentação. A necessidade de utilização deste tipo de sistemas surge frequentemente quando o responsável pela tomada de decisão de uma empresa necessita de obter indicadores de desempenho em tempo útil, no universo das informações disponíveis. É de realçar que estas informações estão, normalmente, distribuídas em vários sistemas de informação ou bases de dados operacionais. Os próximos subcapítulos descrevem o enquadramento do projeto UC-Num, os objetivos e a estrutura do documento.

1.1. Enquadramento

O presente projeto faz parte de um dos cinco módulos em desenvolvimento do projeto UC-Num, que por sua vez integra-se noutro projeto, o UC-Cloud e este num projeto maior, o SAMA (Sistema de Apoio à Modernização Administrativa). Cujo objetivo é monitorizar e melhorar as infraestruturas de informação que auxiliam as atividades e gestão nos diversos sectores existentes na Universidade de Coimbra, figura 1:

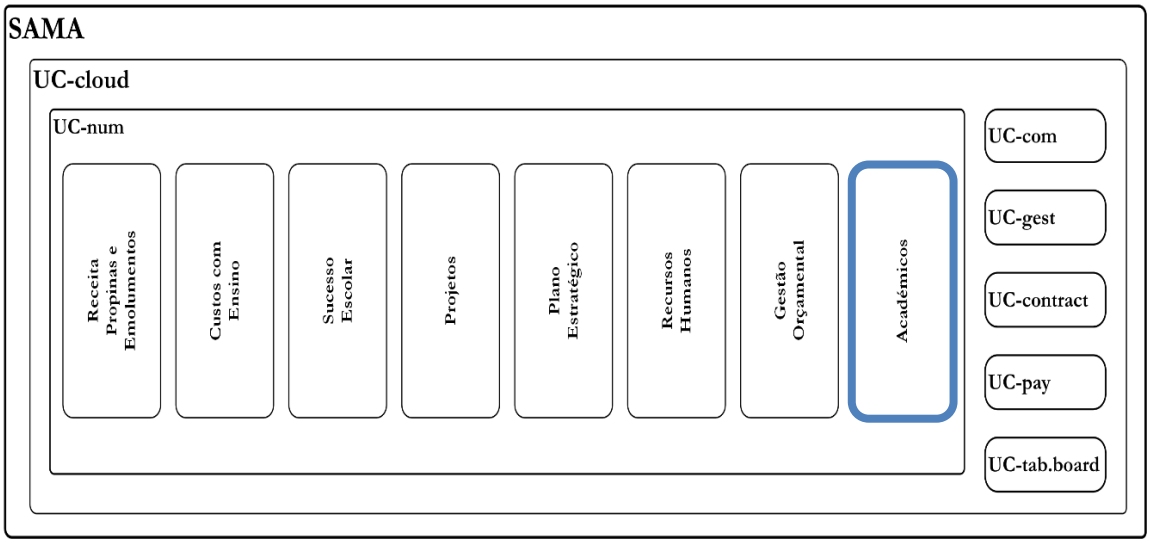


Figura 1 - Enquadramento do projeto UC-Num no SAMA

[1] Altura em o Ralph Kimball publicou a primeira edição da *Data Warehouse Toolkit*

Atualmente a UC dispõe, entre outras, das seguintes soluções de *software* para a gestão operacional e a monitorização da informação produzida: SAP, NÓNIO e uma *Data Warehouse*. A *Data Warehouse* é constituída, de momento, por quatro *data marts* inseridos nas grandes áreas (Académicos (sucesso escolar), Recursos Humanos, Financeiro ("receita propinas e emolumentos" e "despesa com ensino")) existentes no plano estratégico e de ação definido para 2011-2015 da universidade. Estes *data marts* foram desenvolvidos nos anos 2013-2014, e vieram melhorar a forma como os *stakeholders* da Universidade consultam e analisam a sua informação estratégica.

Apesar de existirem estes *data-marts*, ainda não estão disponíveis todos os indicadores desejados, pelo que, os principais *stakeholders* continuam a ter dificuldades em obter indicadores de desempenho (*KPIs*) para responder às questões frequentes relacionadas às atividades académicas. Neste sentido, o presente projeto visa desenvolver uma solução para recolher, tratar, armazenar e analisar informação originária de diversas fontes de dados existentes dentro da UC, com o intuito de calcular e disponibilizar de forma automática e continuamente atualizada os indicadores de desempenho que permitam monitorizar e avaliar o desempenho da área académica nas diversas vertentes previstas no seu Plano Estratégico e de Ação da UC.

1.2. Objetivo

Este estágio consiste no desenvolvimento de um sistema de suporte à decisão. Cujos objetivos que me foram incumbidos são:

- Extrair e identificar indicadores de desempenho no Plano Estratégico e de Ação, e em conjunto com os órgãos de decisão e administração da UC na área académica;
- Definir e desenvolver o plano *ETL* para extração, tratamento e carregamento dos dados na *data mart*;
- Desenvolver o modelo de dados da *data mart*;
- Desenvolver uma interface web para análises *OLAP*.

1.3. Estrutura do documento

O presente documento encontra-se estruturado da seguinte forma:

- **Introdução**: faz a contextualização do projeto e apresenta os seus objetivos;
- **Metodologia**: descreve a metodologia de desenvolvimento utilizada neste projeto, assim como o planeamento do 1 e 2 semestre;
- **Estudo das tecnologias existentes**: descreve o estudo feito das tecnologias que se enquadram neste projeto;
- **Análise de requisitos**: descreve os requisitos funcionais e não funcionais;
- **Arquitetura**: descreve a arquitetura, as tecnologias presentes na arquitetura, o plano *ETL* e o modelo de dados;
- **Implementação**: descreve a fase de implementação do *ETL*, dos *dashboards* e o produto final*;*
- **Validação**: demonstra quais foram os processos utilizados na validação dos dados;
- **Metodologia de teste**: descreve a metodologia utilizada nos testes efetuados neste projeto;
- **Conclusão**: faz o balanço do desenvolvimento do projeto efetuado.

Capítulo 2
Metodologia

Este capítulo descreve a metodologia do desenvolvimento utilizada neste projeto, assim como o plano de trabalho do primeiro e segundo semestres. Qualquer projeto de *software* precisa de seguir uma metodologia de desenvolvimento bem estruturada. Através dela consegue-se prever em qualquer fase se o projeto conseguirá, ou não cumprir com a data da entrega final do produto, ou seja, é um guia pela qual as equipas inseridas num projeto se orientam.

2.1. Metodologia de desenvolvimento

Este projeto utiliza uma metodologia de desenvolvimento designada como cíclica. É normalmente recomendada por Ralph Kimball[2] para o desenvolvimento de projetos na área de *BI*. Esta abordagem concentra-se em primeiro lugar, nas necessidades do negócio e em segundo lugar, no tratamento dos dados e por último, na consulta dos mesmos pelos utilizadores. Esta metodologia funciona por iterações que é formada por um conjunto de tarefas sequenciais que são dependentes umas das outras.

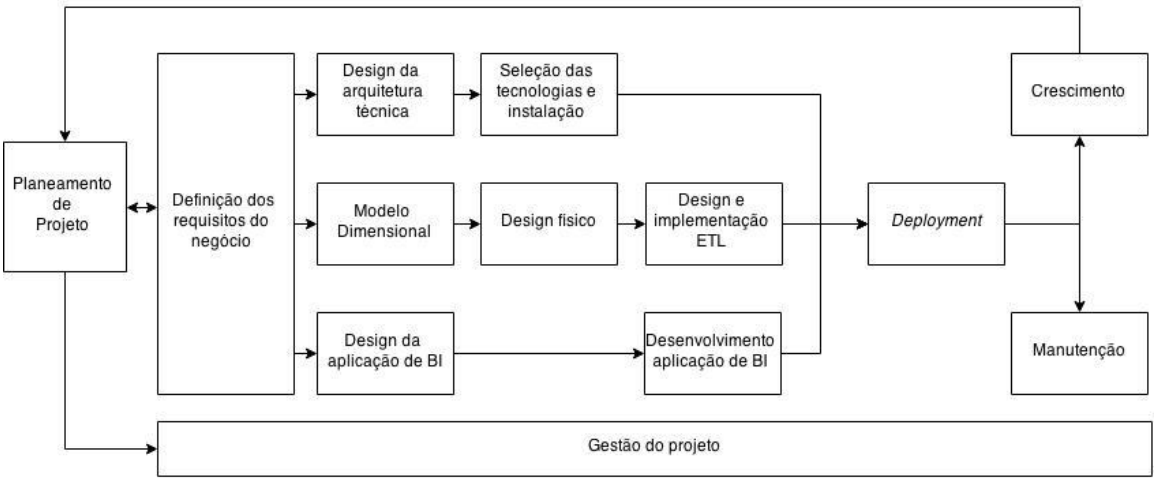


Figura 2 – Metodologia de desenvolvimento do projeto BI – Ciclo de vida do projeto

O ciclo de vida do desenvolvimento do projeto (ilustrado na figura 2) começa com o planeamento do mesmo. Sendo uma fase importante, necessita de grande esforço e tempo para este refletir a realidade. Este planeamento foi feito com a ajuda da equipa UC-NUM, uma vez que existiam outros módulos a serem desenvolvidos em simultâneo. As estimativas e os *milestones* das tarefas foram definidas tendo em conta dois aspetos:

1- A curva de aprendizagem das ferramentas (*Kettle e BI-server*);
2- A complexidade do desenvolvimento que cada indicador podia apresentar.

Como existiam reuniões semanais ao longo do ano, estas permitiram resolver problemas, decidir a melhor abordagem a seguir e até reajustar o planeamento quando foi necessário.

Inicialmente o trabalho foi organizado através da plataforma *Redmine* da UC e armazenado na plataforma *dropbox*. Com o objetivo de melhorar e controlar as versões da documentação e do

[2] A metodologia foi desenvolvida através de décadas de experiências de projetos de *data warehouse*

desenvolvimento, migrou-se para a ferramenta *GITLab*. Para gerir e organizar melhor o projeto optou-se por utilizar a ferramenta *Taiga*.

Durante o período de desenvolvimento do 2º semestre, para cada âmbito de indicador foi criado uma iteração com as seguintes fases de desenvolvimento: 1- *ETL*; 2- Definição do(s) cubo(s) *OLAP*; 3- Desenvolvimento das análises *OLAP*(*dashboard);* 4- Testes; 5- Disponibilização do indicador.

2.2. Planeamento

Este subcapítulo apresenta o plano definido e o trabalho realizado no 1º e 2º semestres. O trabalho realizado no primeiro semestre está representado na figura 3. Este seguiu a metodologia descrita no subcapítulo 2.1, onde foram definidas um conjunto de tarefas que dependem umas das outras.

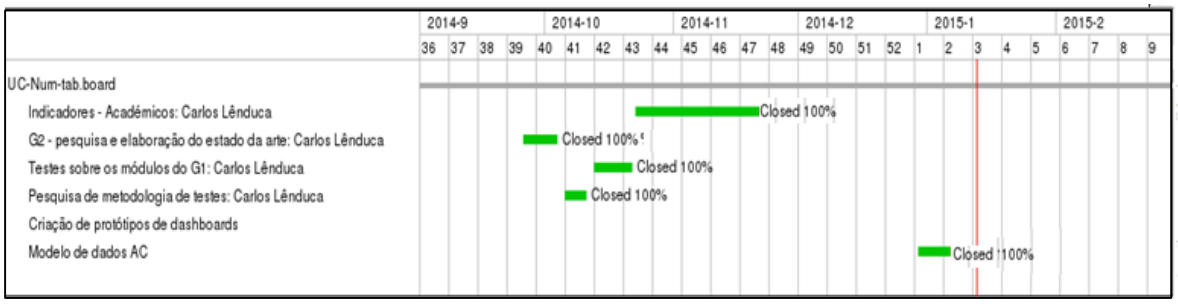


Figura 3 - Planeamento do 1º semestre

Inicialmente, comecei por fazer um estudo das tecnologias ou ferramentas que se enquadram na área deste projeto.

Depois efetuei outro estudo para definir uma metodologia de teste que será utilizada posteriormente na fase de testes.

Quando a metodologia de teste ficou definida, efetuei testes ao módulo de "Sucesso escolar" atualmente em funcionamento na plataforma da *data warehouse* da UC consoante os requisitos funcionais do módulo.

De seguida, comecei a definição dos indicadores, como pode ser visto na figura 3, demorou mais tempo que as restantes tarefas, pois existiam muitos indicadores que necessitavam ser definidos e analisados.

Em paralelo com a definição dos indicadores desenvolvi protótipos que ajudaram na definição e especificidade dos mesmos.

Depois da definição e validação dos indicadores, desenvolvi modelos de dados para a área de estágio e *data mart*.

Por fim, a escrita do relatório intermédio foi feita em paralelo com as tarefas citadas anteriormente.

Já no 2º semestre a metodologia seguiu estritamente a mesma metodologia cíclica que são ilustrados nas figuras 4, 5 e 6.

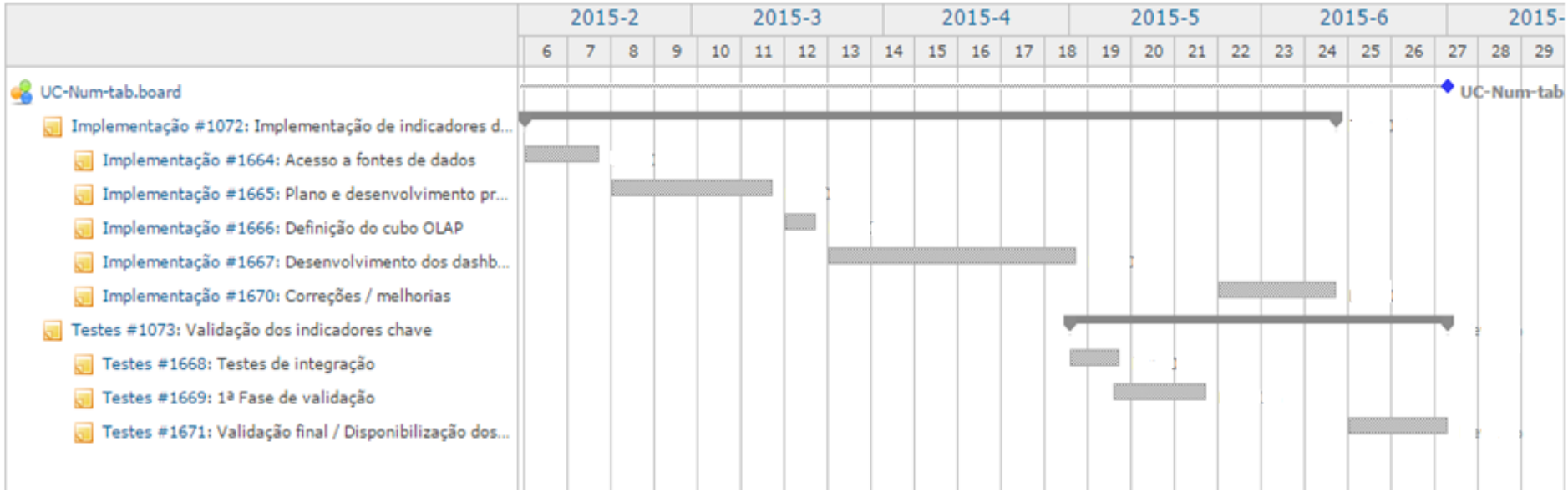


Figura 4 - Planeamento do 2º semestre

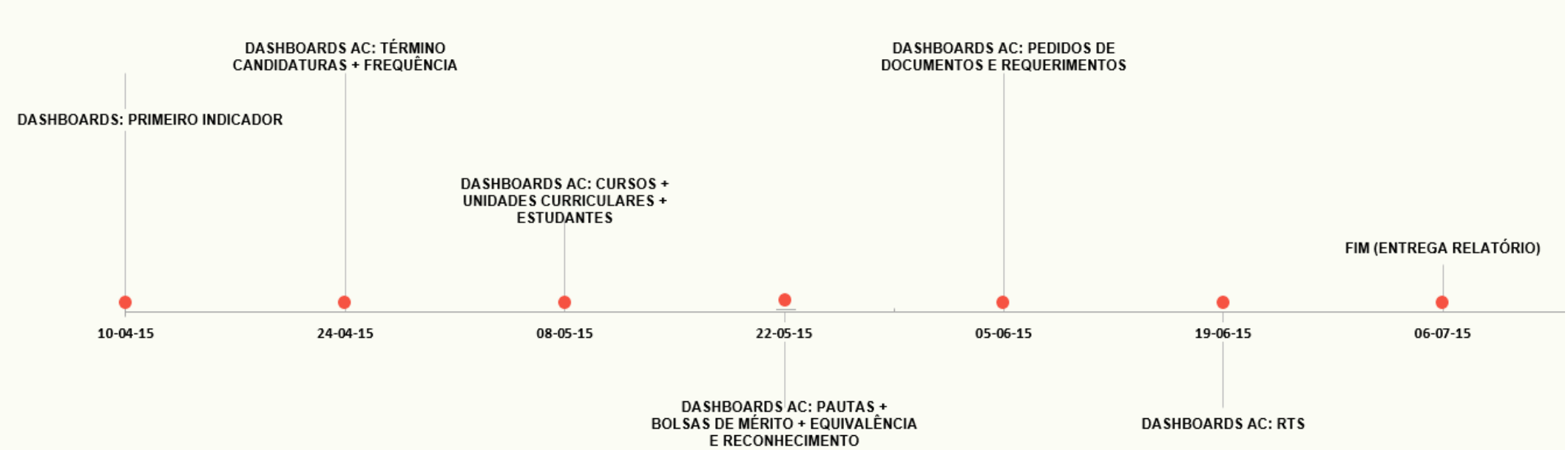


Figura 5 – *Milestone 1*: planeamento do segundo semestre

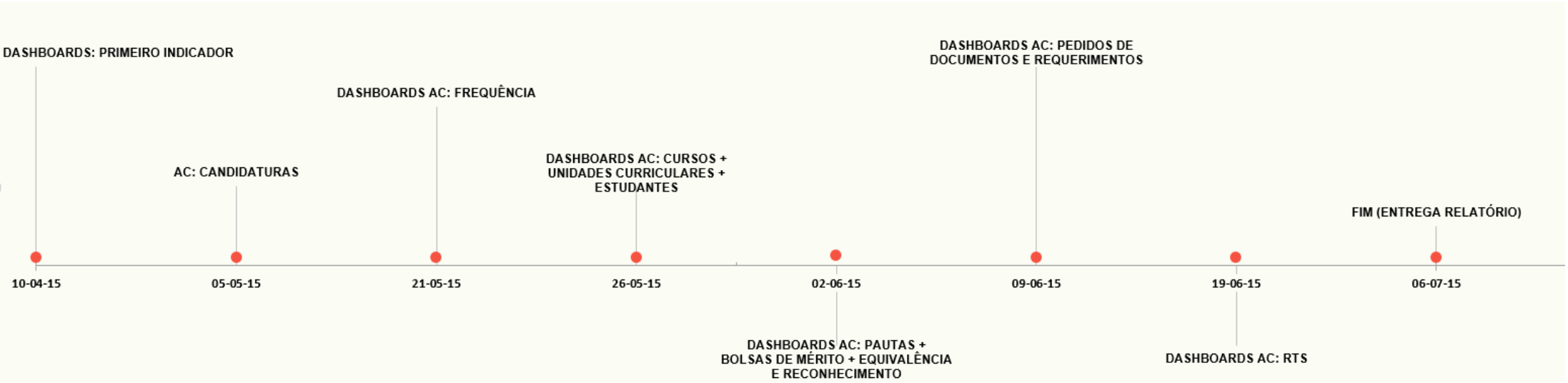


Figura 6 - *Milestone* 2: planeamento do segundo semestre

A figura 4 apresenta o plano do 2 semestre, já a figura 5 apresenta os principais *milestones*. A figura 6 ilustra os reajustamentos que foram efetuados no planeamento. Os atrasos que a figura 6 demostra foram causados relativamente à não disponibilização atempada dos dados fonte, à pouca experiência com as ferramentas utilizadas e por último à complexidade de desenvolvimento que cada indicador apresentou.

Portanto, houve várias iterações de desenvolvimentos que permitiram desenvolver os indicadores de prioridades "Elevada" descritos no subcapítulo 4.3. Estes foram dos seguintes âmbitos: candidaturas, frequência, cursos, unidades curriculares e estudantes (grau de satisfação e taxa de empregabilidade). Ficando por desenvolver os indicadores de prioridade "Média" e "Baixa".

2.3. Análise de riscos

Qualquer projeto de *software* está sujeito a imprevistos durante todo o processo do seu desenvolvimento. Estes imprevistos podem afetar o cumprimento do planeamento do projeto. Neste sentido, foram identificados e analisados os riscos que poderiam ocorrer durante o desenvolvimento do projeto, assim como o seu tipo que pode ser direto ou indireto, a sua categoria (técnico ou recurso), a probabilidade deste acontecer, o impacto, o plano de contingência e o estado (se este risco ocorreu ou não).

Risco **01**	Tipo **Indireto**	Categoria **Recurso**	Probabilidade **Alta**	Impacto **Médio**
Descrição				
Atraso na conclusão de toda a especificação de indicadores.				
Plano de contingência				
Iniciar desenvolvimento incremental, de acordo com os indicadores especificados até à data. Monitorização atenta e constante dos trabalhos desenvolvidos pelas equipas operacionais.				
Estado				
A ocorrência deste risco verificou-se e foi aplicado o plano de contingência.				

Risco **02**	Tipo **Indireto**	Categoria **Recurso**	Probabilidade **Alta**	Impacto **Alto**
Descrição				
Disponibilização não atempada dos dados fontes.				
Plano de contingência				
Análise completa de todas as necessidades dos sistemas fonte (exemplo: documentação com todos os atributos necessários). Iniciar desenvolvimento à medida que vão sendo disponibilizados dados (independentemente da prioridade do requisito). Solicitar dados fictícios enquanto os reais não estão disponíveis.				
Estado				
A ocorrência deste risco verificou-se e foram aplicados o primeiro e segundo pontos do plano de contingência.				

Risco **03**	Tipo **Direto**	Categoria **Recurso**	Probabilidade **Média**	Impacto **Alto**
Descrição				
Falta de conhecimento nas ferramentas utilizadas para o desenvolvimento.				
Plano de contingência				

Auxílio constante dos elementos da equipa com mais experiência; Realização de *workshops*; Disponibilização e consulta de material bibliográfico.
Estado A ocorrência deste risco verificou-se e foi aplicado o plano de contingência.

Risco **04**	Tipo **Indireto**	Categoria **Técnico**	Probabilidade **Baixa**	Impacto **Alto**
Descrição Limitações das tecnologias				
Plano de contingência Atualizar ou reverter a versão das tecnologias. Corrigir os problemas no código fonte da tecnologia, só se aplica a tecnologias *open source*.				
Estado Ainda não ocorreu.				

Risco **05**	Tipo **Indireto**	Categoria **Recurso**	Probabilidade **Alta**	Impacto **Alto**
Descrição Atraso na validação da aplicação pela equipa destacada para tal.				
Plano de contingência Disponibilização atempada da aplicação. Acompanhar o estado do processo de validação juntamente com a equipa destacada.				
Estado A ocorrência deste risco verificou-se e foi aplicado o segundo método de mitigação.				

Capítulo 3
Estudo das tecnologias existentes

O presente capítulo aborda o estudo realizado das tecnologias *open source* de *BI*, tendo em conta o orçamento financiero da Universidade de Coimbra para o desenvolvimento deste sistema. Este estudo, permitiu-me adquirir um conhecimento abrangente das ferramentas ou soluções de *BI* que se enquadram na área do negócio deste projeto. Neste sendido, os próximos subcapítulos descrevem as várias soluções de *BI* agrupadas em soluções de *ETL* e análise *OLAP*. Também é efetuada uma discusão dos sistemas de gestão de base de dados para o armazenamento de dados na *Data Warehouse*.

3.1. Ferramentas de *ETL*

Este subcapítulo descreve algumas ferramentas de extração, transformação e carregamento de dados existentes no mercado na atualidade. As ferramentas selecionadas foram o *Kettle (Pentaho Data Integration),* o *Mule ESB* e o *Google Refine*.

O *Kettle* ou *Data Integration* é uma ferramenta *open source* e faz parte das soluções *BI* do *Pentaho*[2]. Esta ferramenta permite fazer todo o processo de extração, transformação e carregamento dos dados. Possuí uma interface gráfica *drag-and-dr*op intuitiva, fácil de manter e flexível na realização de transformações. Por outro lado, o *Mule ESB*[3][4] é uma ferramenta de integração de sistemas[3] e *open source* (na licença *CPAL*[4]), permite conectar várias aplicações de forma fácil e rápida. Também tem uma interface gráfica de fácil utilização, tipo *drag-and-drop*, mas o utilizador necessita de noções de programação em JAVA para poder ser usado no desenvolvimento do *ETL*, uma vez que contém todos requisitos necessários ou obrigatórios.

O *Google Refine* [5] é uma ferramenta *open source*, utilizada normalmente para tratamentos de dados muito irregulares ou sujos. Faz limpeza e transformação dos dados de um formato para outro, comporta-se como uma base de dados relacional, ou seja, as operações são efetuadas em linhas de dados que têm células e colunas. Permite ao utilizador escolher as linhas que pretende proceder à limpeza utilizando facetas[6]. O utilizador também consegue ver em tempo real as transformações ou limpezas que está a efetuar. Só trabalha com ficheiros e permite a importação dos formatos *TSV*, *CSV*, Texto, *XML*, *JSON*, etc[5].

Para uma melhor comparação entre as ferramentas citadas anteriormente, defini alguns critérios de *ETL* ilustrados na tabela 1:

Critério	*Google Refine*	*Mule ESB*	*Kettle*
Agendador de eventos (*Scheduler*)	✗	✓	✓
Expressões regulares	✓	✓	✓
Open Source	✓	✓	✓

[3] Os sistemas incluem *JMS, Web Services, JDBC, HTTP*, etc.
[4] Common Public Attribution License
[5] Primeira versão disponibilizada a 10 de novembro de 2010
[6] As facetas são expressões regulares aplicadas.

Formatos de leitura/escrita (*.LIST, CSV,JSON,XML*)	✓	✓	✓
Transferência de ficheiros	✗	✓	✓
Transformação de *Playload*	✗	✓	✓
Orquestração	✗	✓	✓
Gestão de fluxos	✗	✓	✓
Modelação visual	✓	✓	✓
Integração com base de dados (*Oracle, PostgreSQL*)	✓	✓	✓
Carregamento massivo de dados	✗	✓	✓
Processamento de componente *Java*	✗	✓	✓
Descompressão de ficheiros	✗	✓	✓

Tabela 1 - Critérios de comparação de *ETL* entre as ferramentas *Google Refine, Mule ESB e Kettle*

A tabela 1 mostra que tanto a ferramenta *Kettle* como o *Mule ESB,* apresentam semelhanças em relação aos critérios e requisitos pretendidos. No entanto, o *Mule ESB* seria a escolha ideal para este projeto uma vez que apresenta uma curva de aprendizagem bastante reduzida quando comparado ao *Kettle* (*Pentaho Data Integration*) e também possui uma ampla e crescente comunidade que fornece material de suporte. Por outro lado, o *Google Refine* segundo alguns testes efetuados, aparenta ser ideal para limpeza de dados mas peca na parte do processo de automação.

A ferramenta (*Kettle*) utilizada neste projeto já se encontrava selecionada pela equipa da UC-Num, e encontra-se em utilização pela atual *Data Warehouse* da Universidade de Coimbra. A ferramenta também possui uma grande comunidade que dá suporte, assim como muita documentação.

3.2. Ferramentas de análise *OLAP*

Quanto às soluções de *BI* para análise *OLAP*, foram selecionadas o *Pentaho Analysis Services, Instant OLAP e Oracle Database OLAP Option*. A maior preocupação foi a solução permitir maior agilidade possível nas suas análises. Neste sentido, a solução deve ser *open source* visto que é um dos objetivos deste projeto, deve possuir interface gráfica para que seja possível desenvolver *dashboard* à medida, deve permitir exportar a informação nos principais formatos, deve suportar linguagem *MDX* sendo esta a usada para aceder a informação em modelos multidimensionais, deve poder ser executado em multiplataforma, deve ter integração com as principais base de dados e deve permitir pré-cálculo. Portanto, a tabela 2, mostra os critérios de comparação tidos em consideração.

Critério	*Instant OLAP*	*Pentaho Analysis Services*	*Oracle Database OLAP Option*
Open Source	✓	✓	✗
Interface web	✓	✓	-
Formatos de leitura/escrita (*PDF*, *CSV*, *XML*, etc)	✓	✓	-
MDX	✓	✓	✓
Multiplataforma	✓	✓	✓
Integração com base de dados (*Oracle*, *MySQL* ou *PostgreSQL*)	✓	✓	Oracle
Pré-Cálculo	-	✓	-

Tabela 2 - Critérios de comparação entre ferramentas *Instant OLAP, Pentaho Analysis Services e Oracle Database OLAP Option*[7]

A tabela 2 mostra que as três soluções apresentam muitas semelhanças. A solução da *Oracle* é proprietária e é conhecida como uma das melhores desta área, sendo completa e com boa performance. Por outro lado, o *Instant OLAP* parece ter pouca documentação e uma comunidade reduzida. Já o *Pentaho Analysis Services* é robusto, responde aos critérios existentes para este tipo de ferramenta, tendo alguma documentação disponível e uma grande comunidade de suporte, etc. É a escolha ideal e também já se encontrava selecionada e em uso pela equipa.

3.3. Sistema de gestão de base de dados

O sistema de gestão de base de dados (SGBD) é de extrema importância quando o assunto é *Data Warehouse*. O SGBD deve garantir alguns requisitos essenciais que são citados a seguir:

- **SQL** – O sistema de gestão de base de dados deve possuir linguagem SQL, sendo uma linguagem relacional e muito utilizada neste tipo de sistema. Esta garante propriedades de atomicidade, consistência, isolamento e durabilidade, ao contrário das base de dados colunares não conseguirem garantir algumas dessas propriedades;
- **Tempo de carregamento** – É necessário que o SGBD garanta boas velocidades de carregamento, visto que a *Data Warehouse* vai conter milhares de dados;
- **Vistas materializadas, índices e particionamento** – Também devem estar disponíveis estas opções, visto que aceleram o desempenho da *Data Warehouse*.

A tabela 3, ilustra todos critérios que foi tido em consideração:

[7] Comparação entre as ferramentas *OLAP*: http://en.wikipedia.org/wiki/Comparison_of_OLAP_Servers

Critério	*MySQL*	*Oracle*	*PostgreSQL*
Open Source	✓	✗	✓
SQL	✓	✓	✓
Tempo de carregamento	Lento	Alto	Lento
Vistas materializadas	✓	✓	✓
Índices	✓	✓	✓
Particionamento	-	-	-

Tabela 3 - Critérios de comparação entre os SGBD (*MySQL*, *Oracle* e *PostgreSQL*)

Tal como a tabela 3 demonstra, o *Oracle* é o SGBD ideal para este projeto, pois este apresenta grandes velocidades de carregamento e consulta dos dados quando comparado com o *PostgreSQL* ou o *MySQL*. Por outro lado, o *Oracle* é proprietário (necessita de uma licença para ser utilizado), e é uma das melhores base de dados existente na atualidade. Se no futuro a Universidade de Coimbra pretender investir num SGBD melhor para a *Data Warehouse*, o *Oracle* seria a escolha certa.

Alguns estudos[8] efetuados demostram que o *PostgreSQL* possui mais recursos (features) em relação ao *MySQL*. Por outro lado, considera-se que o *MySQL* apresenta melhores performances que o *PostgreSQL*. Dado isso, efetuei alguns testes de *benchmark*[9] entre os dois SGBD, e os resultados mostraram que de facto o *PostgreSQL* é mais rápido. Este sistema de gestão de dados também já se encontrava definido (*PostgreSQL)*.

[8] https://www.techunblocked.org/2015/01/postgresql-vs-mysql.html

[9] Utilizei o tutorial do projeto *TPC-H* para efetuar os respetivos testes: http://www.pilhokim.com/index.php?title=Project/EFIM/TPC-H

Capítulo 4
Análise de requisitos

A análise de requisitos é uma das fases cruciais no desenvolvimento dos projetos de *software*. Sendo necessário perceber muito bem o problema do negócio que se pretende resolver. Para alcançar este objetivo é de extrema importância compreender as reais necessidades das partes interessadas (*stakeholders* da Universidade de Coimbra), e identificar quais são as questões que lhes são colocadas com frequência por parte dos responsáveis de topo (equipa reitoral, diretores de unidades orgânica, etc.) e é nelas que tem de ser concentrado o desenvolvimento das análises finais.

Os requisitos são divididos em: funcionais e não funcionais. A fase de definição dos requisitos funcionais é de extrema importância, uma vez que é nesta em que são definidas as funcionalidades do projeto e a fonte dos futuros problemas. Estes são causados porque o cliente pode ter uma ideia errada do projeto às vezes por falta de experiência na área de negócio.

Para evitar esse tipo de problema mencionado anteriormente, foram produzidos protótipos rápidos em paralelo com a definição dos requisitos funcionais. Entretanto, este tema é aprofundado num dos próximos subcapítulos.

Os requisitos funcionais são identificados utilizando duas designações: IND_AC_XX e RF_GE_XX. O RF_GE_XX representa os requisitos gerais comuns a todas as áreas da plataforma implementados até ao momento. A designação IND_AC_XX representa os requisitos funcionais priorizados pelo grupo operacional (ver subcapítulo 4.1), onde o XX representa um identificador único de cada requisito. Já os não funcionais, contêm a designação RNF_XX, onde o XX representa o identificador do requisito.

O modelo utilizado para levantamento e especificação de requisitos: *FURPS+* - usabilidade (U), confiabilidade (R, do inglês *reliability*), performance (P) e suporte (S), o símbolo + permite acrescentar requisitos relacionados com implementação, design/interface ou hardware.

A tabela 4 apresenta as prioridades utilizadas na priorização dos requisitos:

Prioridade	Definição
Elevada	Requisito de extrema importância para o funcionamento do projeto. Por serem imprescindíveis devem ser os primeiros a serem implementados.
Média	Requisito não muito importante, se não for implementado o projeto funciona sem problemas.
Baixa	É o requisito que não compromete as funcionalidades básicas do projeto. São requisitos que podem ser implementados nas próximas versões do projeto.

Tabela 4 - Prioridades utilizada na classificação dos requisitos

Gerais

Requisitos ou funcionalidades que a plataforma contém e que o módulo dos Académicos deve suportar, sendo funcionalidades comuns a todos módulos em funcionamento na plataforma do projeto UC-Num.

				PRIORIDADE		
Identificação	**Dependência**	**Requisito**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
RF_GE_01	-	Autenticação	A aplicação deve permitir ao utilizador a autenticação (*login*) através das credenciais utilizadas no acesso a quaisquer serviços disponibilizados à comunidade da UC (email da UC e password). A autenticação tem de ser bem sucedida, isto é, as credenciais têm de ser válidas para que seja permitida qualquer visualização de dados ao utilizador.	☐	☐	☒
RF_GE_02	RF_GE_013	Fechar sessão	O utilizador pode, após autenticação, efetuar o término da sua sessão (*logout*).	☐	☐	☒
RF_GE_03	RF_GE_013	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	RF_GE_013	Navegação entre os módulos	O utilizador deve conseguir aceder a todos os restantes módulos, este acesso deve ser possível a qualquer momento.	☐	☒	☐
RF_GE_05	RF_GE_011	Navegação interna	A aplicação deve permitir ao utilizador efetuar *drill down* nos dados que pretende visualizar, esta "descida" no detalhe da informação deve ser efetuada diretamente nos dados que vão sendo apresentados na vista de *snapshot*. Para que seja possível ao utilizador efetuar *roll up* dos dados, isto é, subir no nível de detalhe, a aplicação deve disponibilizar uma forma de navegação estrutural (*breadcrumbs*) que vá acrescentando o nível onde o utilizador se encontra. A hierarquia de níveis pode ser diferente consoante a área de cada módulo.	☐	☐	☒
RF_GE_06	RF_GE_011	Parâmetros gerais	O utilizador deve ter disponível os diversos parâmetros (seleção de indicadores, agregadores e/ou filtros) que são permitidos modificar nos dados que este está a visualizar.	☐	☐	☒
RF_GE_07	RF_GE_011	Parâmetros de tempo	Deve ser permitido ao utilizador modificar os parâmetros temporais (dimensão tempo) a aplicar nos dados que este está a visualizar.	☐	☐	☒
RF_GE_08	RF_GE_011	Esconder parâmetros	Deve ser permitido ao utilizador esconder a barra onde se encontram os parâmetros gerais e de tempo.	☒	☐	☐
RF_GE_09	RF_GE_011	Secção de ajuda	A aplicação deve disponibilizar uma secção de ajuda ao utilizador, transversal a todos os módulos, para esclarecer	☐	☐	☒

			quaisquer dúvidas que sejam suscitadas nos utilizadores - FAQ.			
RF_GE_10	RF_GE_011	Informação auxiliar	Cada vista de dados disponibilizada ao utilizador (gráfico ou tabela) deve ser acompanhada de dois mecanismos que permita consultar informação: uma referente aos dados que são apresentados, corresponde a um botão de informação (remetendo para a secção de ajuda) e outra de navegação (sugestões, erros, etc).	☐	☐	☒
RF_GE_11	RF_GE_013	Visualização: Gráfico ↔ Tabela	A aplicação deve permitir ao utilizador visualizar a informação apresentada num gráfico em formato de tabela e vice-versa. Os dados apresentados na tabela deverão, pelo menos, corresponder à informação que se encontra no gráfico.	☐	☐	☒
RF_GE_12	RF_GE_011	Exportar informação da tabela/gráfico	Exportar para formato *Excel* ou *CSV* a informação presente nas tabelas de análise. Exportar gráfico como imagem.	☒	☐	☐
RF_GE_13	RF_GE_001	Autenticação	O utilizador após autenticação, pode visualizar a informação afeta ao(s) módulo(s) em que está inserido. Por exemplo: se um utilizador pertencer ao grupo DW_SE deve conseguir aceder a todo o módulo de sucesso escolar.	☐	☐	☒
RF_GE_13	RF_GE_011	Zoom de gráficos	Efetuar zoom in e out de um dos gráficos.	☒	☐	☐

Tabela 5 - Requisitos gerais comuns a todas as áreas da plataforma

4.1. Grupo operacional

O grupo operacional responsável por analisar, validar e aprovar os indicadores de desempenho e futuras funcionalidades da plataforma de suporte à decisão é constituído pelos seguintes elementos:

- Vice-reitora Prof.ª Dr.ª Madalena Alarcão: Responsável;
- Filipe Rocha;
- Dr.ª Ângela Ferreira;
- Eng.ª Sílvia Figueiredo;
- Eng. Paulo Pereira;
- Eng. Pedro Pinto.
- Eng. Carlos Lênduca

4.2. Protótipos rápidos

Os protótipos rápidos ou de alto nível serviram para facilitar a compreensão na correção dos requisitos identificados pela equipa perante o cliente. Estes protótipos permitiram melhorar a definição dos requisitos funcionais. Neste sentido, foram desenvolvidos protótipos para os âmbitos da Candidatura, Frequência e *RTs*. O âmbito da Candidatura contém a informação do concurso nacional de acesso e do concurso via escola, o da Frequência contém a informação dos alunos inscritos na UC e o *Request Tracker(rt)* contém a informação dos pedidos de *tickets* nas filas tratadas pelo Serviço de Gestão Académica.

No âmbito das candidaturas do concurso nacional de acesso, é demonstrada como a informação deve ser apresentada nos gráficos ao nível da Universidade de Coimbra, Unidades orgânicas, Departamentos e Cursos, assim como o funcionamento de um filtro. Também foram criados alguns exemplos de como a informação deve ser apresentada através de uma desagregação. Para o concurso via escola (da Candidatura), os protótipos demonstram o funcionamento das desagregações (candidaturas concluídas e não concluídas) e filtros.

Na Frequência, os protótipos apresentam os filtros contemplados e como é apresentada a informação da desagregação "programa de mobilidade". O âmbito RTs contém uma hierarquia diferentes das restantes, os protótipos demonstram como a informação deve ser apresentada ao nível dos Serviços de Gestão Académica, filas e subfilas. Também difere em termos temporais, uma vez que a informação pode ser consultada por ano civil, semestre, trimestre, mês, semana e dia.

A título de exemplo, as figuras que se seguem apresentam alguns dos protótipos produzidos[10]. São relativos a taxa de preenchimento de vagas no concurso nacional de acesso desagregado por género mostrado na figura 7. Já a figura 8 apresenta a mesma taxa no nível de granularidade dos departamentos da FCTUC.

[10] Os protótipos produzidos encontram-se no anexo C.

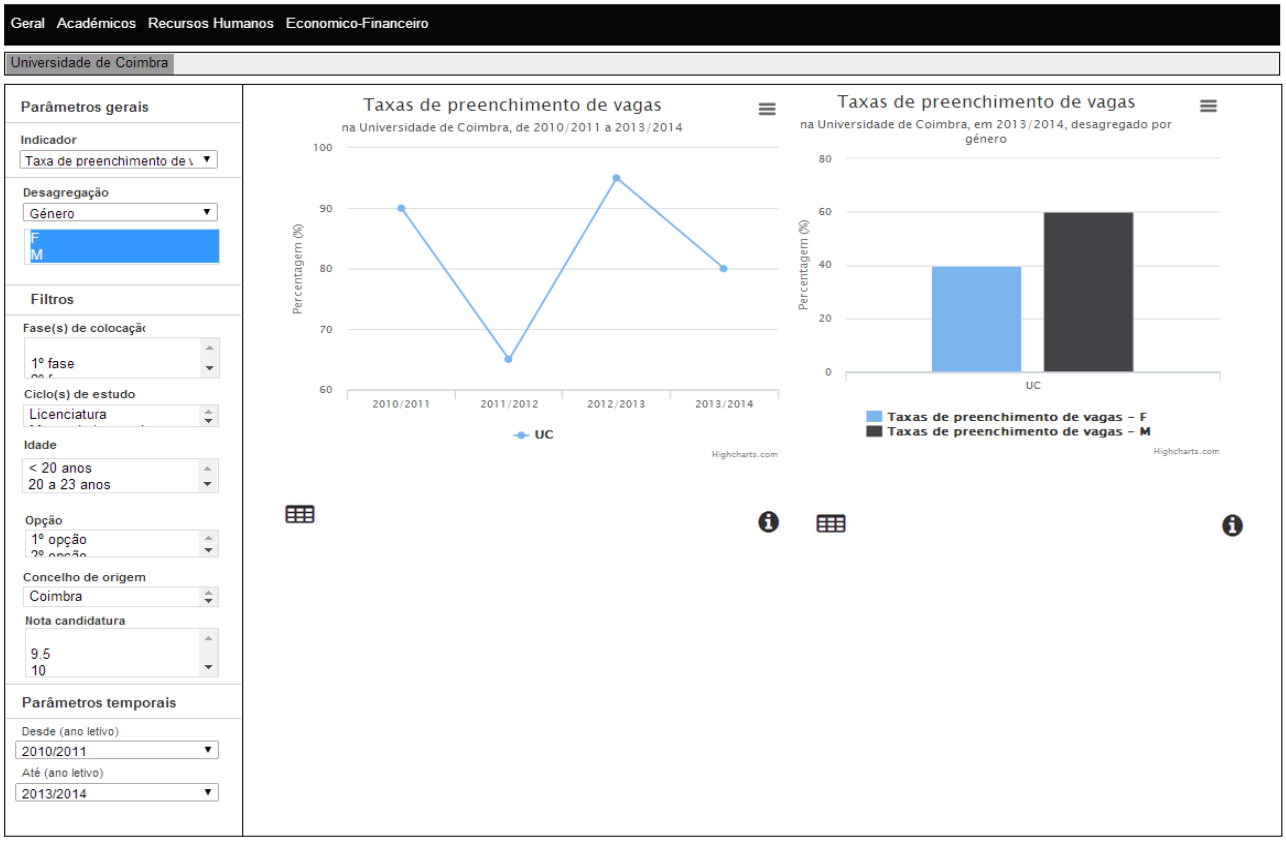


Figura 7 - Protótipo em relação a taxa de preenchimento de vagas na UC

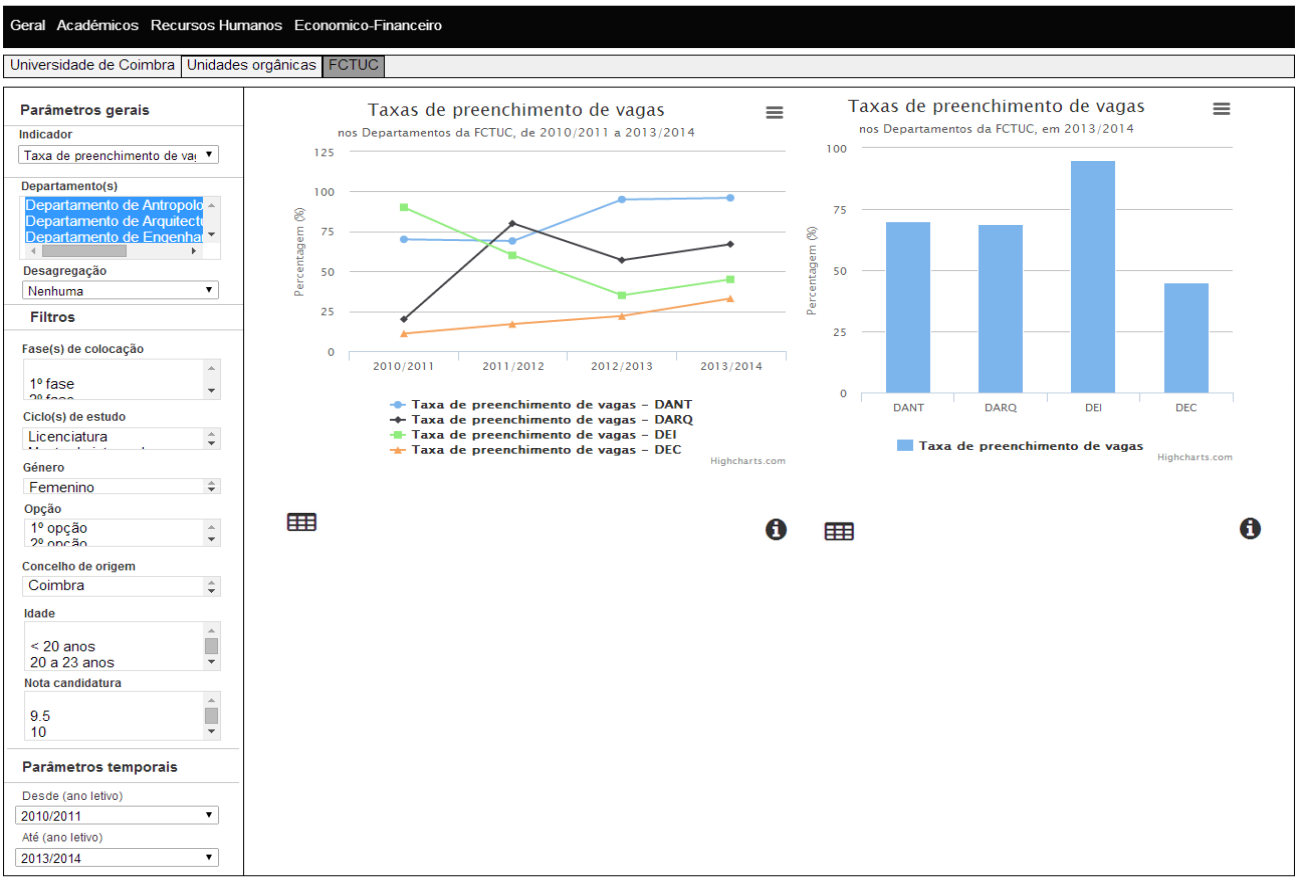


Figura 8 – Protótipo em relação a taxa de preenchimento de vagas em alguns departamentos da FCTUC

Para terminar, o tempo médio de resposta a *tickets* recebidos no nível de granularidade das filas de RTs é demostrado na figura 9. A figura 10 mostra o tempo médio de resposta a *tickets* recebidos na fila "GA" e filtrado por tipo de remetente igual a "Estudante". Os protótipos foram produzidos com ajuda da ferramenta "*Justinmind Prototyper*".

Figura 9 - Protótipo em relação ao tempo médio de resposta a *tickets* recebidos nas filas RT

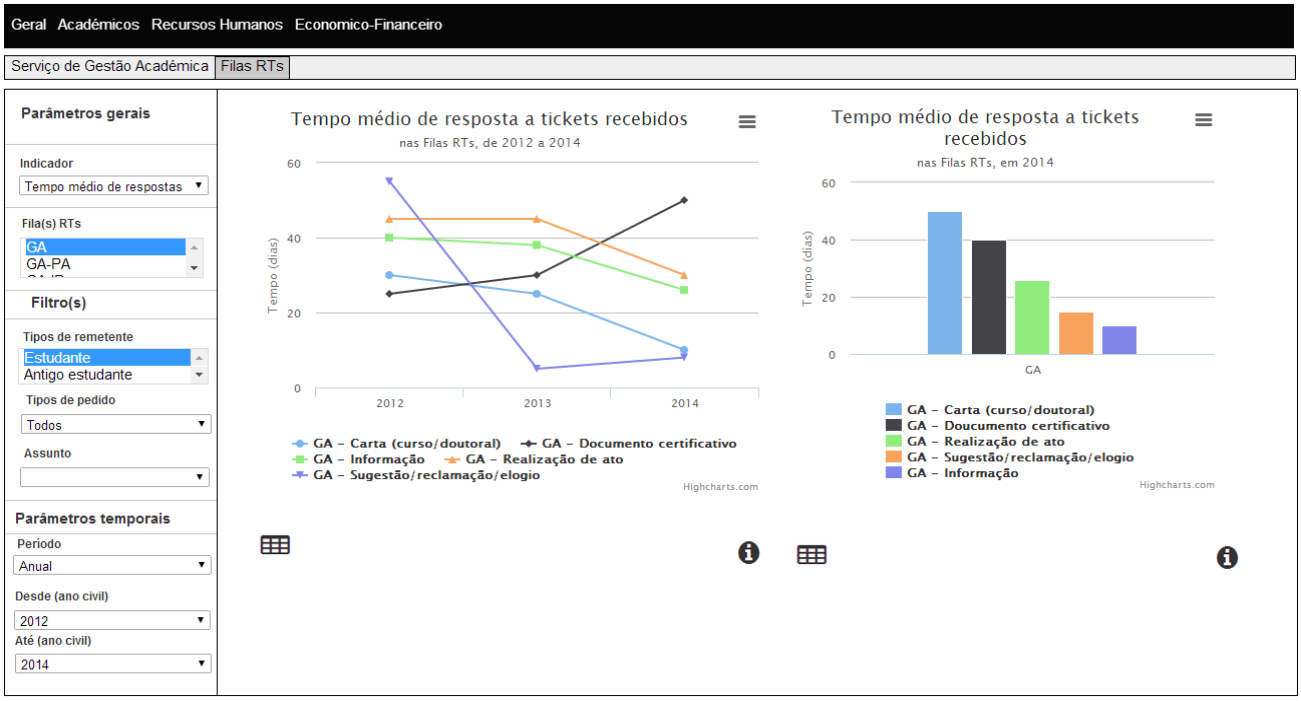


Figura 10 - Protótipo em relação ao tempo médio de resposta a *tickets* recebidos na fila "GA" e "Estudante" como remetente

4.3. Requisitos funcionais

Como foi referido anteriormente, os requisitos funcionais servem para especificar todas as funcionalidades presentes na plataforma. Os requisitos foram definidos com base nos indicadores de desempenho presentes no plano estratégico de 2011 a 2015 (Anexo A) da Universidade de Coimbra e outros documentos fornecidos pelo grupo operacional da área académica. Para facilitar o processo da definição e especificidade dos indicadores foram definidas algumas normas, que contém a designação de ficha de indicador. No entanto, cada ficha representa toda a informação detalhada do indicador (Anexo B). O ciclo utilizado para a definição dos indicadores é ilustrado na figura abaixo.

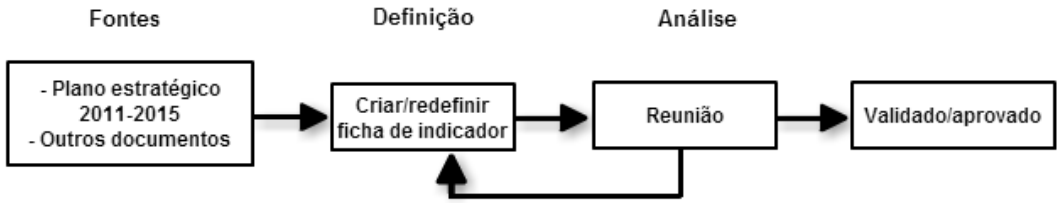


Figura 11 - Ciclo de vida utilizado na definição dos indicadores

Neste sentido, as fichas de indicadores foram criadas de acordo com as informações presentes nos documentos e plano estratégico da universidade de 2011 a 2015. E posteriormente analisadas com o grupo operacional da área académica.

Como o grupo operacional é constituído com elementos de níveis operacionais, os indicadores foram definidos com garantia de que é possível obter as respetivas informações através dos sistemas fonte. Por existir um número considerável de indicadores, foram necessárias várias reuniões para analisar todos eles e posteriormente serem validados e aprovados pela responsável da área académica.

A seguir são descritos os requisitos funcionais designados como indicadores. Estes representam todos que foram extraídos a partir do plano estratégico e de outros documentos fornecidos pelo grupo operacional. É de realçar que a priorização dos indicadores foi feita pelo grupo operacional, consoante a importância que estes representam à Universidade de Coimbra.

Candidaturas:

Este âmbito engloba candidaturas do concurso nacional de acesso, via escola[11] e as candidaturas a disciplinas isoladas, esta informação é apresentada através de tabelas e gráficos (temporal e de barras). Para cada indicador presente na tabela 6 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra, e ir descendo até ao curso, para o caso do concurso nacional de acesso e via escola, para o caso das disciplinas isoladas continua a descer até ao nível unidade curricular.

[11] Concurso via escola são considerados regimes como reingresso, mudança e transferência, concursos Especiais, etc. diferente do concurso nacional de acesso.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
Concurso nacional de acesso					
IND_AC_001	Taxa de preenchimento de vagas no concurso nacional de acesso	Indica a taxa de preenchimento de vagas do concurso nacional de acesso na UC. Durante um período de tempo específico. É calculada através do número de colocados a dividir pelo número de vagas.	☐	☐	☒
IND_AC_002	Número de vagas	Indica o número total de vagas na UC para o concurso nacional de acesso. Durante um período de tempo específico.	☐	☐	☒
IND_AC_003	Número de candidatos	Indica o número total de candidatos na UC pelo concurso nacional de acesso, durante um período de tempo específico.	☐	☐	☒
IND_AC_004	Número de candidatos colocados	Indica o número total de candidatos colocados na UC, pelo concurso nacional de acesso. Durante um período de tempo específico.	☐	☐	☒
IND_AC_005	Número de matriculados e inscritos	Indica o número total de matriculados e inscritos na UC, colocados através do concurso nacional de acesso. Durante um período de tempo específico.	☐	☐	☒
IND_AC_006	Nota mínima e máxima de entrada dos colocados	Indica a nota mínima e máxima dos colocados na UC pelo concurso nacional de acesso. Durante um período de tempo específico.	☐	☐	☒
IND_AC_007	Percentagem de estudantes captados de entre os 25% de melhores candidatos no concurso nacional de acesso	Indica a percentagem de estudantes captados de entre os 25% melhores candidatos no concurso nacional de acesso na UC. Durante um período de tempo específico. A sua fórmula de calcula: n.º de candidatos de entre os 25% melhores candidatos do concurso nacional de acesso a dividir pelo n.º total de candidatos entre os 25% melhores e multiplicado por 100	☐	☐	☒

Concurso via escola					
IND_AC_008	Taxa de preenchimento de vagas (concurso via escola)	Indica a taxa de preenchimento de vagas criadas para o concurso via escola na UC – inclui-se todas as candidaturas admitidas/colocadas. Durante um período de tempo específico. O cálculo é feito usando a mesma fórmula do indicador IND_AC_001.	☐	☐	☒
IND_AC_009	Taxa de procura (concurso via escola)	Indica a taxa de procura que é representada pela relação entre o número de vagas criadas e candidaturas recebidas no concurso via escola na UC. Durante um período de tempo específico. A sua fórmula de cálculo é número total de candidaturas recebidas a dividir pelo número de vagas.	☐	☐	☒
IND_AC_010	Número de candidaturas concurso via escola	Indica o número total de candidaturas (concluídas e não concluídas) no concurso via escola na UC. Durante um período de tempo específico.	☐	☐	☒
IND_AC_011	Número de vagas (concurso via escola)	Indica o número total de vagas no concurso via escola na UC. Durante um período de tempo específico.	☐	☐	☒
Disciplinas isoladas					
IND_AC_012	Candidatos a unidades curriculares isoladas	Indica o nº de candidatos a unidades curriculares isoladas	☐	☐	☒

Tabela 6 - Indicadores do grupo candidaturas

Frequência:

O âmbito da frequência apresenta a informação sobre os estudantes inscritos e a taxa de crescimento do número de estudantes a frequentar a UC, a informação é apresentada através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 7 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até á lista de estudantes de uma unidade curricular.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_021	Número de estudantes inscritos na UC	Indica número total de estudantes inscritos na UC, num período de tempo específico.	☐	☐	☒
IND_AC_022	Taxa de crescimento do número de estudantes a frequentar na UC	Indica a taxa de crescimento do número de estudantes a frequentar o 1.º, 2.º e 3.ºciclo na UC, por um período de tempo específico. Possui a seguinte fórmula: A: estudantes a frequentar no ano letivo n-1 B: estudantes a frequentar no ano letivo n. (B-A)/A*100	☐	☐	☒

Tabela 7 - Indicadores do grupo da frequência

Unidade curricular:

O âmbito da unidade curricular apresenta informação sobre o número das unidades curriculares que são apresentadas através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 8 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até a unidade curricular.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_039	Número de unidades curriculares	Indica o número total de unidades curriculares, por curso e ciclo de estudos. Estes valores são relativos a um período de tempo específico.	☐	☐	☒

Tabela 8 - Indicador do grupo unidade curricular

Curso:

O âmbito do curso apresenta a informação sobre o número de cursos que são representados através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 9 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até ao departamento caso exista.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_038	Número total de cursos	Indica o número total de cursos abertos a matrículas e inscrições na UC. Estes valores são relativos a um período de tempo específico.	☐	☐	☒

Tabela 9 - Indicador do grupo curso

Estudante:

O âmbito do Estudante apresenta a informação sobre a taxa de assiduidade média dos estudantes, grau de satisfação, taxa de empregabilidade e número de estágios de versão na UC. Estas informações são apresentadas através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 10 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até ao nível do curso no caso do número de estágio de verão e da taxa de empregabilidade. No caso da taxa de assiduidade média dos estudantes e grau de satisfação continua a descer até ao nível da unidade curricular.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_018	Taxa de assiduidade média dos estudantes	Indica a taxa de assiduidade média dos estudantes nas unidades curriculares da UC. Durante um período de tempo específico.	☒	☐	☐
IND_AC_019	Grau de satisfação dos estudantes das entidades acolhedoras que participam nos estágios de verão	Indica o grau de satisfação dos estudantes das entidades acolhedoras que participam nos estágios de verão na UC. Durante um período de tempo específico.	☒	☐	☐
IND_AC_020	Taxa de empregabilidade	Indica a taxa de empregabilidade dos diplomados na UC, um ano apos a conclusão. Durante um período de tempo específico.	☐	☐	☒

IND_AC_036	Grau de satisfação dos estudantes participante que participam nos estágios de verão	Indica o grau de satisfação dos estudantes participante que participam nos estágios de verão na UC. Durante um período de tempo específico.	☒	☐	☐
IND_AC_037	Nº de estágios de verão	Indica o número total de estágios de verão na UC. Durante um período de tempo específico.	☒	☐	☐
IND_AC_041	Grau de satisfação dos estudantes	Indica o grau de satisfação dos estudantes na UC. Durante um período de tempo específico.	☐	☐	☒

Tabela 10 - Indicadores do grupo estudante

***Request Tracker* (*RT*):**

No âmbito do *RTs* é apresentada a informação sobre os *tickets* nas filas do Serviço de Gestão Académica. A informação é apresentada através de tabelas e gráficos (temporal e de barra). Para cada indicador presente na tabela 11 deve ser possível navegar desde o nível mais alto de granularidade que é o Serviço de Gestão Académica até ao mais baixo que são as Filas *RT* ou as Subfilas *RT* caso exista.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_031	Número de *tickets* nas filas relativas ao interface do SGA.	Indica o número total de *tickets* nas filas tratadas pelo SGA. Durante um período de tempo específico.	☒	☐	☐
IND_AC_032	Média do número de *tickets* resolvidos nas filas relativas ao interface do SGA	Indica a média do número de *tickets* resolvidos nas filas do SGA. Durante um período de tempo específico. A sua fórmula de cálculo é: Nº de *tickets* com estado resolvido / Dias	☒	☐	☐
IND_AC_033	Tempo médio de respostas a *tickets* recebidos nas filas relativas	Indica a tempo médio de respostas a *tickets* recebidos nas filas relativas ao interface do SGA. Durante um período de	☒	☐	☐

	ao interface do SGA	tempo específico. A sua fórmula de cálculo é: Somatório *tickets* resolvido/nº total de pedidos			

Tabela 11 - Indicadores do grupo RTs

Pedido de documento e requerimento:

As informações deste âmbito são apresentadas através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 12 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até ao curso.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_026	Número de documentos emitidos	Indica o número total de documentos de natureza académica emitidos na UC. Durante um período de tempo específico.	☒	☐	☐
IND_AC_027	Tempo médio de resposta (a pedido de documento)	Indica o tempo médio de resposta a pedidos de documentos de natureza académica na UC. Durante um período de tempo específico. A sua fórmula de cálculo é: B-A/N: onde A: Data de criação/pagamento, B: Data de emissão e N: nº de pedidos existente naquele período.	☒	☐	☐
IND_AC_028	Número de requerimentos	Indica o número total de requerimentos na UC. Durante um período de tempo específico.	☒	☐	☐
IND_AC_029	Tempo médio de resposta (a pedidos de requerimento)	Indica o tempo médio de resposta a pedidos de requerimentos na UC. Durante um período de tempo específico. Utiliza a mesma fórmula do indicador IND_AC_027.	☒	☐	☐
IND_AC_030	Tempo de tomada de decisão	Indica o tempo de tomada de decisão a pedidos de requerimentos na UC. Durante um período de tempo	☒	☐	☐

		específico. Também utiliza a mesma fórmula do indicador IND_AC_027.			

Tabela 12 - Indicadores do grupo pedido de documento e requerimento

Títulos académicos:

No âmbito de títulos académicos a informação é apresentada através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 13 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até ao nível do curso no caso de título de agregado. No caso do doutoramento "honoris causa" desce até ao nível da unidade orgânica.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_034	Nº total de títulos de agregado na UC	Nº total de títulos de agregado na UC, num período de tempo específico.	☐	☒	☐
IND_AC_035	Número total de doutoramentos "Honoris Causa"	Número total de doutoramentos "Honoris Causa" na UC, em um período de tempo específico.	☐	☒	☐

Tabela 13 - Indicadores do grupo títulos académicos

Equivalência e reconhecimento estrangeiro:

No âmbito de equivalência e reconhecimento estrangeiro, a informação é apresentada através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 14 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até ao curso no caso do nº de equivalências de grau e unidade orgânica no caso do nº de reconhecimento de habilitação.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_015	Número de equivalências de	Indica o número total de equivalências de grau	☐	☒	☐

	grau estrangeiro pedidas	estrangeiro pedidas na UC. Durante um período de tempo específico.			
IND_AC_016	Número de reconhecimento de habilitação pedidos	Indica o número total de reconhecimento de habilitação pedido na UC. Durante um período de tempo específico.	☐	☒	☐
IND_AC_017	Tempo médio de resposta a pedidos de equivalências de grau estrangeiro	Indica tempo médio de resposta a pedidos de equivalências de grau estrangeiro na UC. Durante um período de tempo específico. A sua fórmula de cálculo é: Somatório de pedidos resolvido/ nº total de pedidos.	☐	☒	☐
IND_AC_040	Tempo médio de resposta a pedidos de reconhecimento de grau estrangeiro	Indica tempo médio de resposta a pedidos de reconhecimento de grau estrangeiro na UC. Durante um período de tempo específico. Possui a mesmo forma de cálculo do indicador IND_AC_017.	☐	☒	☐

Tabela 14 - Indicadores do grupo equivalência e reconhecimento estrangeiro

Pauta:

No âmbito de pauta a informação é apresentada através de tabelas e de gráficos (temporal e de barra). Para cada indicador presente na tabela 15 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até à unidade curricular ou pauta.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_024	Número de pautas emitidas	Indica o número total de pautas emitidas na UC, durante um período de tempo específico.	☐	☒	☐
IND_AC_025	Tempo de disponibilização de classificações	Indica o tempo de disponibilização até um estudante obter a sua	☐	☒	☐

	definitivas aos estudantes	avaliação definitiva, por ano letivo. O indicador é calculado da seguinte forma: data de aceitação – data de avaliação.			

Tabela 15 - Indicadores do grupo pauta

Bolsas e prémios:

No âmbito das bolsas e prémios, as informações são apresentadas através de gráficos (temporal e de barra) e tabelas. Para cada indicador presente na tabela 16 deve ser possível navegar desde o nível mais alto de granularidade que é a Universidade de Coimbra e ir descendo até ao curso.

			PRIORIDADE		
ID INDICADOR	**NOME**	**DESCRIÇÃO**	**Baixa**	**Média**	**Elevada**
IND_AC_013	Nº de bolsas de mérito e prémios escolares	Indica o número total de bolsas de mérito e prémios escolares na UC, num período de tempo específico.	☒	☐	☐
IND_AC_014	Nº total de bolsas de mérito atribuídas pelo Senado aos 3% dos melhores alunos	Indica o número total de bolsas de mérito atribuídas pelo Senado aos 3% dos melhores alunos na UC, num período de tempo específico.	☒	☐	☐

Tabela 16 - Indicadores do grupo bolsa e prémio

4.4. Requisitos não funcionais

Estes requisitos são diferentes dos funcionais, por esse motivo é que não foram definidos pelo grupo operacional. Estes requisitos foram definidos em conjunto com a equipa UC-NUM. A seguir são apresentados os requisitos não funcionais com a sua prioridade:

				PRIORIDADE		
Identificação	**NOME**	**DESCRIÇÃO**	**Fonte**	**Baixa**	**Média**	**Elevada**
RNF_S_001	Atualização de dados	Processo *ETL* e atualização da *DW* e cubo *OLAP* devem ser automáticos.	Cliente	☐	☐	☒
RNF_S_002	Compatibilidade (browser)	Aplicação web deve ser compatível com os *browsers* mais modernos, a partir das versões mencionadas: Internet Explorer 9; *Firefox* 20, Safari 6 e *Chrome* 31.	Cliente	☐	☐	☒
RNF_S_003	Compatibilidade (SO)	Como sistema operativo para os servidores deve ser suportada a distribuição *de Linux Red Hat - CentOS 6.x*.	Equipa	☐	☒	☐
RNF_S_004	Licenças	A aplicação deve ser desenvolvida e disponibilizada através de *software* gratuito.	Cliente	☐	☐	☒
RNF_S_005	Monitorização de erros	Deve existir um mecanismo para gestão de logs.	Equipa	☐	☒	☐
RNF_O_006	*Hardware*	O software deve executar numa máquina com as seguintes características mínimas: 4Gb de RAM, 20Gb de espaço em disco e um processador dual core, não tem necessariamente de ser um ambiente de 64 bits. Estas características estão diretamente relacionadas com as mínimas exigidas pelo *software* que foi selecionado para desenvolvimento e disponibilização da aplicação.	Equipa	☐	☐	☒
RNF_O_007	Confidencialidade na comunicação	Deve ser utilizado o protocolo *HTTPS* em toda a comunicação entre os utilizadores e a aplicação.	Cliente	☐	☐	☒
RNF_O_008	Autenticação	Para aumentar a segurança da aplicação a autenticação e validação do acesso à aplicação deve ser efetuada com as credenciais dos utilizadores do sistema LDAP da UC.	Cliente	☐	☐	☒

RNF_R_009	Mecanismo de fail over	Deve existir um mecanismo que garanta o funcionamento da aplicação, mesmo quando o servidor primário está indisponível.	Equipa	☐	☐	☒
RNF_R_010	Mecanismo de recuperação de falhas	O sistema deve conter um mecanismo para iniciar automaticamente os serviços necessários ao seu correto funcionamento.	Equipa	☐	☐	☒
RNF_R_011	*Backup* dos dados	Armazenar periodicamente *backups* dos dados presentes na base de dados.	Equipa	☐	☐	☒
RNF_P_012	Utilizadores em simultâneo	A aplicação deve suportar pelo menos 60 utilizadores em simultâneo.	Cliente	☐	☐	☒
RNF_P_013	Desempenho do front-end	A medida de qualidade deve ser igual ou superior a 80 (considerando a pontuação disponibilizada pelo o *Yslow* - http://yslow.org).	Equipa	☐	☐	☒
RNF_U_014	Satisfação dos utilizadores	A percentagem de satisfação dos utilizadores, com a usabilidade da aplicação, não deve ser inferior a 80%.	Equipa	☐	☐	☒
RNF_U_015	Tempo de carregamento	O tempo de carregamento das páginas web deve ser, no máximo, 5 segundos.	Equipa	☐	☐	☒

Tabela 17 - Requisitos não funcionais

Capítulo 5
Arquitetura

O presente capítulo apresenta a arquitetura do sistema utilizado neste projeto. É feita uma pequena descrição de cada componente/tecnologia presente na arquitetura, assim como do processo *ETL*, do modelo de dados e do desenvolvimento da análise *OLAP*.

5.1. Arquitetura do sistema

Esta secção apresenta a arquitetura do sistema (figura 12). O sistema é composto por três áreas – 1º *ETL* (processo de extração, transformação e carregamento de dados) – 2º Servidor *OLAP* (Criação de cubos e *dashboards*)- 3º Utilizador final (vai consumir os dados do servidor *OLAP*):

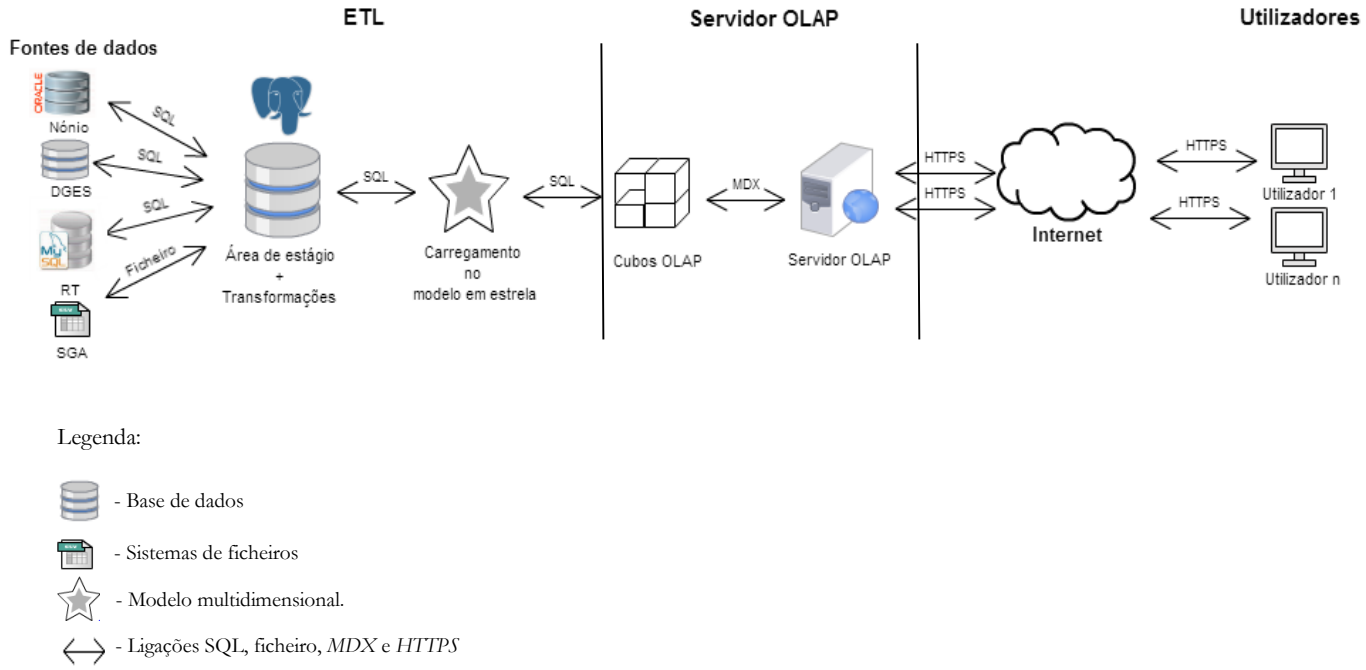


Figura 12 - Arquitetura do sistema

Como é ilustrado na figura 12, existem três camadas distintas. A primeira camada é referente ao *ETL*, que descreve o processo de extração, transformação e carregamento dos dados. Os dados a serem extraídos encontram-se em sistemas operacionais que são bases de dados relacionais e/ou ficheiros, estes dados são extraídos das respetivas fontes e transferidos para à área de estágio. Por sua vez a área de estágio armazena os dados extraídos, em tabelas, estes mesmos dados precisam passar por um processo de tratamento para terem qualidade. Depois de passarem pelo processo anterior os dados são carregados para a *Data MART* (modelo multidimensional).

A segunda camada é responsável pelas ferramentas de acesso aos dados. Inicialmente, são criados os vários cubos *OLAP*. Os cubos *OLAP* representam os dados em vários níveis, permitindo fácil acesso aos mesmos no momento das pesquisas, ou seja, são estruturas multidimensionais específicas para cada âmbito da área académica. De seguida, são

desenvolvidos os *dashboards* que vão apresentar os dados em gráficos e tabelas. É através dos *dashboards* que os utilizadores (terceira camada) podem aceder aos dados através de pedidos *HTTPS* nos seus computadores.

5.2. Tecnologias presentes na arquitetura

Neste subcapítulo são descritas as tecnologias presentes na arquitetura apresentada na figura 12. Neste sentido, começo por citar os sistemas presentes nas fontes de dados. O NÓNIO é a fonte maioritária dos dados deste projeto, visto ser o sistema que suporta todo processo de gestão académica da Universidade de Coimbra, utiliza o *OracleSQL* como sistema de gestão de base de dados. Por outro lado, o *RT* (*Resquest Tracker*), é um sistema de gestão de *tickets* gratuito que a Universidade utiliza para gerir, organizar e centralizar os pedidos dos utilizadores, e armazena essa informação no sistema de gestão de base de dados *MySQL*. Já o sistema da DGES (Direção Geral de Ensino Superior) armazena a informação em base de dados Access e o último sistema (SGA – Serviço de gestão académica) utiliza base de dados *Excel*.

Na parte do processo *ETL*, foi utilizado o *Pentaho Data Integration* mais conhecido como *Kettle,* sendo uma ferramenta bastante completa e intuitiva para este tipo de processo. Para o armazenamento dos dados tanto na área de estágio como na *Data Warehouse* foi utilizado o *PostgreSQL* (como sistema de gestão de base de dados). É uma base de dados *open source* e robusta com muitos recursos disponíveis que vão desde procedimentos de armazenamento, vistas, triggers, etc.

Na fase de acesso aos dados ou análise *OLAP*, foi utilizado o *Mondrian (Schema Workbench)* na criação de cubos *OLAP* para otimizar as consultas futuras. No entanto, os mesmos dados armazenados nos cubos são apresentados em formato de gráfico e tabela pelo servidor *OLAP*. Este utiliza a ferramenta *Pentaho BI Server* com o auxílio do *plugin CDE* (*Community Dashboard Editor*) para desenvolver os *dashboards* (através do *HTML, JavaScrit e CSS*). O sistema utiliza consultas *MDX (ad-doc)* nos *dashboards*, sendo muito comum nesse tipo de sistema. As mesmas foram desenvolvidas com o auxílio dos *plugins Analysis view e Saiku.*

Para terminar, os clientes (utilizadores) vão aceder aos *dashboards* desenvolvidos através de *browsers* (*Chrome, Firefox*, etc) devidamente autenticados. Cada utilizador deverá ter uma conta no servidor de autenticação (*LDAP*) da Universidade de Coimbra para poder aceder ao sistema.

5.3. Processo de *ETL*

O processo *ETL (Extract-Transform-Load*) é sempre um desafio para qualquer projeto, sendo o alicerce de uma *Data Warehouse*. Por este motivo, é investida a maior parte do tempo no desenvolvimento da mesma – O Ralph Kimball[7] menciona no seu livro, que o tempo gasto é normalmente 70%, sendo uma fase muito delicada que necessita de ser muito bem pensada para garantir a qualidade e consistência dos dados. O tratamento destes dados não é uma tarefa fácil porque estas informações vêm com muito ruído das fontes ou bases de dados relacionais. O ruído significa campos vazios, duplicados, dados não consistentes, etc. Este subcapítulo descreve o plano *ETL* mostrado na figura 13:

Figura 13 - Arquitetura do plano *ETL* do projeto

Como já foi referido o plano *ETL* é o processo de extração, transformação e carregamento dos dados. A figura 13 mostra que o primeiro passo a ser efetuado é a extração dos dados nas várias fontes. Estas fontes identificadas para este projeto foram o NÓNIO, *Request Tracker*, DGES e ficheiros da SGA. O NÓNIO e a DGES disponibilizaram vistas materializadas específicas (as vistas do NÓNIO encontram-se descritas no anexo G), o sistema *RT* disponibilizou uma réplica das suas tabelas da base de dados (descrita no anexo H) e os ficheiros foram disponibilizados pelo SGA.

O objetivo deste tipo de sistema é apresentar dados históricos mas não muito antigos, por exemplo não é muito relevante para os responsáveis da Universidade de Coimbra analisarem dados de 10 ou 20 anos. Neste sentido, a primeira extração foi desde o ano letivo 2008/2009 até 2014/2015. Só a partir do ano letivo 2008-2009 é que o NÓNIO entrou em funcionamento para todas as unidades orgânicas e começou a ter dados fidedignos de serem consultados. É de realçar que existem alguns indicadores, como "Candidatos a unidades curriculares isoladas" em que só existe informação a partir do ano letivo 2013/2014, também do "Grau de satisfação" dos anos letivos 2010-2011 a 2014-2015 e da "Taxa de empregabilidade" de 2008-2009 a 2012-2013. Para estes indicadores o NÓNIO só começou a registar as respetivas informações há relativamente pouco tempo, o mesmo acontece com os outros sistemas como o do *RT* e SGA. Nos próximos carregamentos será considerada uma janela temporal de três (ano letivo atual e dois anteriores) com o objetivo da *data mart* estar sempre consistente com os sistemas fonte. É frequente, os responsáveis dos serviços académicos, efetuarem algumas alterações nos dados com menos de 3 anos. A periocidade de extração dos dados varia consoante o indicador que pode ser diária, semanal, mensal e anual (ver anexo B).

Os dados extraídos dos sistemas fonte são armazenados na área de estágio da arquitetura. Umas das vantagens do armazenamento ser feito em base de dados relacionais é garantir a durabilidade e o fácil acesso aos dados em comparação com outras estruturas de carregamentos como é de ficheiros (*XML*, texto, etc.). Os dados precisam de passar por um longo processo de limpeza. Começa-se por analisar toda a informação existente, com o intuito de identificar campos com erros ortográficos, que não façam sentido e/ou campos não normalizados[12]. Neste sentido, são aplicadas várias transformações aos dados, como por exemplo, no caso do âmbito *RTs* é preciso formatar para maiúscula todas as siglas relativas às filas e subfilas, fazer correções ortográficas que possam vir a existir nos dados, criar o campo

[12] Estes são os mais complicados de serem tratados porque leva muito tempo até toda informação ser normalizada. E se o campo não for normalizado a partir do sistema fonte, o tratamento será contínuo sendo sempre necessário uma intervenção humana a cada extração.

da faixa etária através do ano de nascimentos dos estudantes, concatenar a designação e o código das unidades curriculares, no caso do período de tempo concatenar os trimestres, semestres (1º s ou 2º s), etc.

De seguida é necessário identificar e tratar corretamente os registos repetidos que podem ser muitos ou nenhuns, consoante o âmbito do indicador. Por último, fazer agregados para serem apresentados nos *dashboards*.

Depois dos dados estarem tratados são efetuados os agregados e carregados no modelo em estrela (multidimensional), ou seja, *data mart*. Inicialmente são carregadas as tabelas de dimensões e posteriormente as tabelas de factos. O carregamento é feito nesta ordem porque as tabelas de factos são constituídas pelas várias chaves estrangeiras das dimensões. Para tal, é necessário existir procedimentos que vão recolher as chaves em cada dimensão ligada à tabela de factos e inseri-las na mesma. Para terminar, todo o processo *ETL* descrito anteriormente funciona de forma automatizado através de um plano de agendamento que *Kettle* permite realizar.

5.4. Modelo de dados

O modelo de dados é uma das partes mais importantes e é a chave do sucesso para os projetos de *data warehouse*. Como já foi mencionado na arquitetura (subcapítulo 5.1), o processo *ETL* é composto pela área de estágio e *data mart* para armazenamento dos dados. Segundo o Ralph Kimball[7], pode manter-se os dados na área de estágio de forma persistente (física) ou processá-los em memória. Por um lado, o processamento em memória é muito mais rápido que o armazenamento persistente mas por outro lado, pode tornar-se um problema caso exista uma falha no momento do processamento. Como se precisa também garantir a durabilidade optou-se pelo armazenamento persistente, para ser mais específico, armazenamento em tabelas de acordo com as vistas dos dados. Neste sentido, o presente subcapítulo faz a descrição da área de estágio (com 18 tabelas) e do modelo em estrela (*data mart*) composta por 45 tabelas.

5.4.1. Área de estágio

O modelo para a área de estágio foi construído com base nas vistas materializadas e réplicas das tabelas relacionais, que estão definidas nas fontes de dados. A área de estágio é composta por 18 tabelas onde, cada atributo id das tabelas é preenchido através dos ids de cada registo extraído das fontes de dados. A tabela "histórico_carregamento" é responsável por armazenar a informação da designação do âmbito do(s) indicador(es), ano(s) letivo(s) e data do carregamento na *data mart*. Em caso de existir alguma auditoria sobre os dados existentes na *data mart*, a tabela poderá ser útil nesse aspeto. Por outro lado, também tem a função de gestão das extrações dos dados nos sistemas fonte. A gestão passa pela definição de uma janela temporal em que os dados são extraídos. Após o primeiro carregamento de cada âmbito de indicador, a tabela permitirá atualizar os registos na *data mart,* assim como adicionar novos. Assim a figura 14 ilustra o modelo completo da área de estágio.

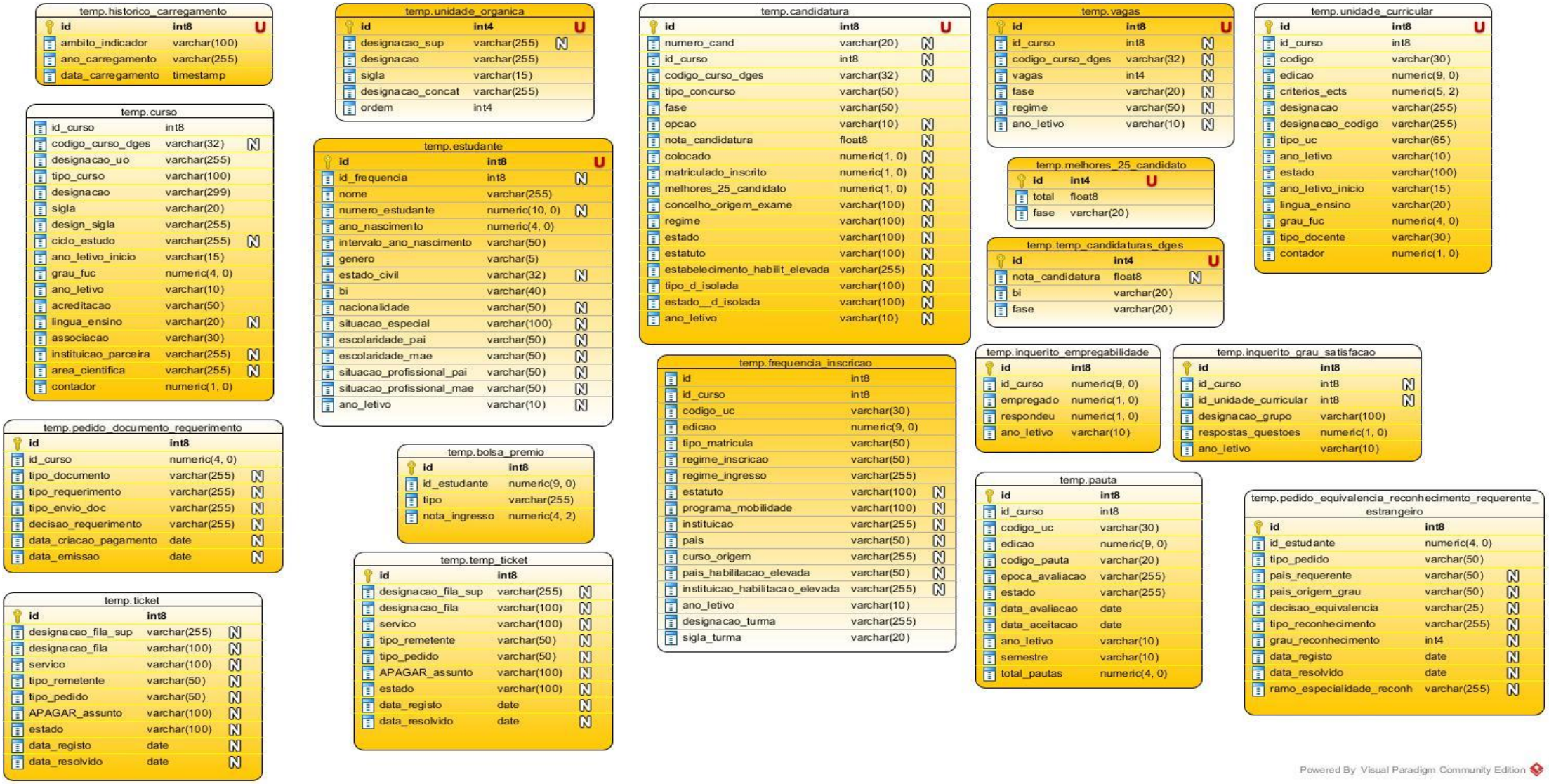


Figura 14 - Modelo de dados da área de estágio $^{Anexo\ (D)}$

O modelo apresentado na figura 14 não possui restrições de chave forasteira por questões de desempenho na importação da informação mas as tabelas estão relacionadas logicamente pelos respetivos identificadores. Como já foi dito anteriormente, o modelo de dados foi construído com base na lógica presente das vistas materializadas, neste sentido, foram extraídos os identificadores de cada vista para relacionar as tabelas na área de estágio. Esta solução permite a reutilização de dados extraídos em outros contextos, por exemplo, o curso necessita da informação do grau de preenchimento das FUC[13] com o objetivo de identificar os cursos com grau de preenchimento das FUC a 100%. Esta informação existe na tabela "unidade_curricular" e como esta tem os identificadores dos cursos, no momento da extração dos dados dos cursos no sistema fonte é feita uma pesquisa através do identificador do curso na tabela da 'unidade_curricular' para obter a FUC. Isto permite reduzir o peso da extração no sistema fonte assim como o volume de dados na área de estágio.

A tabela 'historico_carregamento' é responsável pela gestão da extração dos dados nos sistemas fonte como já referido. Nesta ótica, foi desenvolvida uma função em *PLSQL* com a designação "janela_temporal()", que recebe como parâmetro a designação do âmbito do indicador e verifica se houve algum carregamento, no caso dos primeiros carregamentos a função retorna desde 2008-2009 até ao ano letivo atual. Nos próximos carregamentos só retornará o ano letivo atual e mais dois anteriores, para permitir atualizar os dados existentes na *data mart*, caso seja necessário. Foi aplicada esta abordagem porque é a utilizada no módulo do Sucesso Escolar desenvolvido no ano passado e também foi tido em consideração o funcionamento dos serviços académicos. É frequente atualizarem informações dos estudantes dos anos anteriores, e com este funcionamento a *data mart* estará consistente com os dados dos sistemas fonte. Para terminar, o modelo descrito anteriormente encontra-se em anexo (I).

5.4.2. *Data Warehouse*

A *data warehouse* é utilizada para armazenar grandes quantidades de dados de uma ou mais áreas de negócio, cujo objetivo é a análise de dados e produção de relatórios. Entretanto, ela é composta por uma ou mais *data mart*, onde cada *data mart* representa uma área de negócio na organização.

A *data mart* modela as hierarquias de uma organização através de um modelo multidimensional (conceito criado por Ralph Kimball). Estes modelos são compostos por tabelas de dimensões e de factos. A tabela de factos é construída em torno de registos mensuráveis que podem ser aditivos e/ou semi-aditivos, normalmente esta tabela tende a crescer exponencialmente em relação às tabelas de dimensões. Os dados são modelados entre as dimensões e uma tabela de factos. Para cada registo presente na tabela de facto é um evento ocorrido que é descrito pelas respetivas dimensões.

O modelo de dados proposto para o projeto é composto por 25 tabelas de dimensões e 20 tabelas de factos, que correspondem aos âmbitos da área académica: Candidaturas, frequência, Request Tracker (RTs), bolsas de mérito, curso, equivalência e reconhecimento de habilitações, pauta, pedido de documento e requerimento, unidade curricular, título académico e estudantes (grau de satisfação, taxa de empregabilidade, etc.). A seguir são apresentadas as estrelas que constituem o âmbito da candidatura nas figuras 15, 16 e 17:

[13] Ficha da unidade curricular

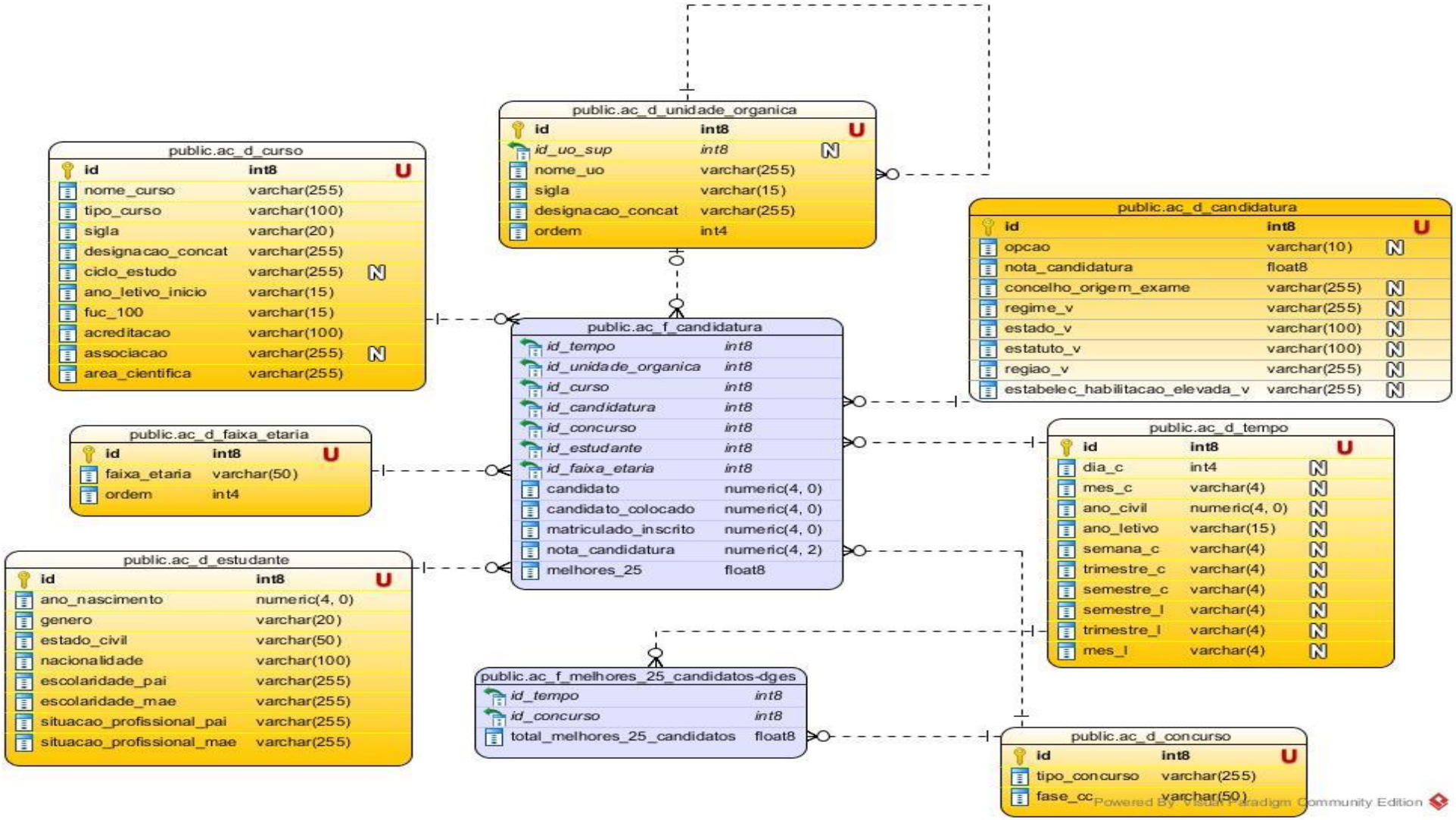


Figura 15 - Parte 1 - Modelo de dados: Candidaturas

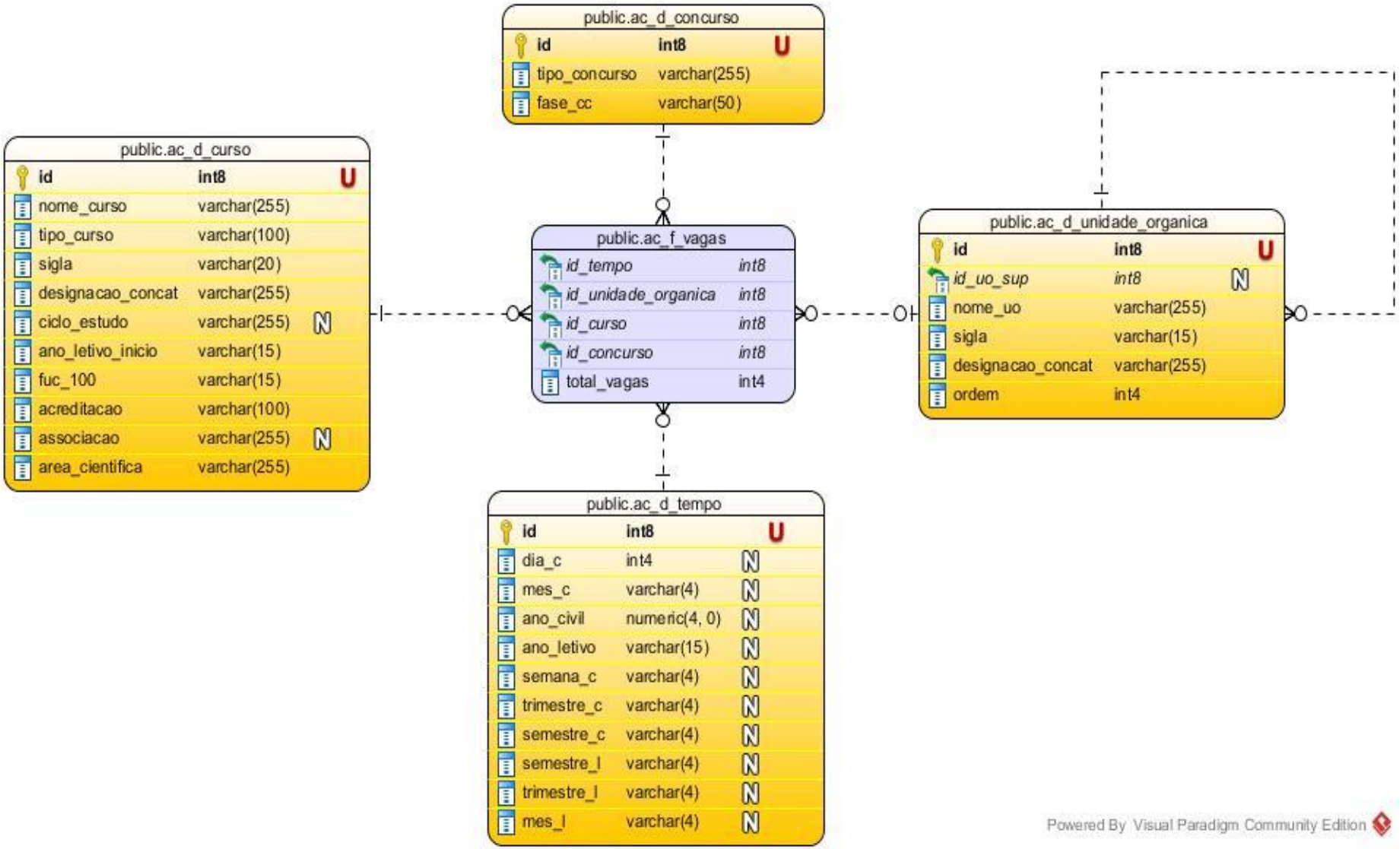


Figura 16 - Parte 2 - Modelo de dados: Vagas das candidaturas

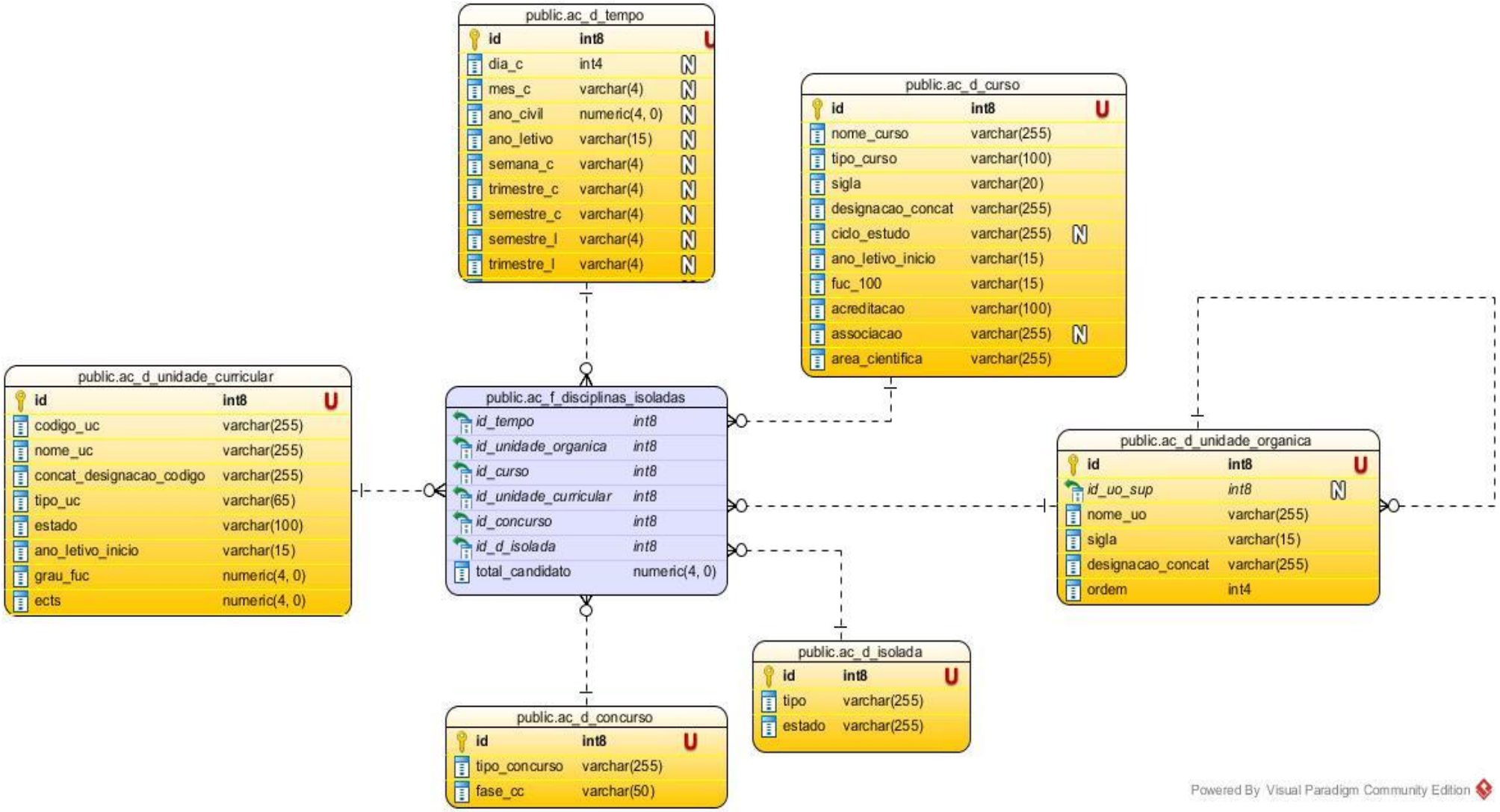


Figura 17 - Parte 3 - Modelo de dados: Candidaturas a disciplinas isoladas.

Este âmbito é constituído por 4 estrelas como foi ilustrado anteriormente. Estas estrelas permitem responder aos indicadores correspondentes às candidaturas do concurso nacional de acesso, concurso via escola e a candidatos a disciplinas isoladas. No entanto, para dar resposta aos indicadores da "Taxa de preenchimento de vagas" e da "Taxa de procura", são efetuadas *drill-across* entre as tabelas de factos "ac_f_candidatura" e "ac_f_vagas" para obter o número de estudantes candidatos/colocados e as vagas do respetivo curso. Também para o indicador da "Percentagem de estudantes captados de entre os 25% de melhores candidatos no concurso nacional de acesso", é efetuado o *drill-across* entre as tabelas de factos "ac_f_candidatura" e "ac_f_melhores_25_candidatos-dges", com o intuito de obter o número de estudantes identificados de entre os 25% melhores que escolheram a Universidade de Coimbra e o 25% dos melhores candidatos no concurso nacional de acesso.

Existem algumas preocupações que se devem ter em consideração no desenvolvimento de uma *data mart*. É frequente existirem dimensões com necessidade de atualização dos dados ao longo do tempo, para esses casos é necessário armazenar o histórico dessas atualizações, como é o caso das dimensões do curso, unidade curricular e estudante. Estas dimensões são designadas como *slowly changing dimension* – E o que as torna *slowly changing dimension?* – Em relação à dimensão curso, o campo "fuc_100"[14] pode variar ao longo dos anos letivos assim como o campo "grau_fuc"[15] da dimensão da unidade curricular. No caso da dimensão do estudante são os seguintes campos: o estado civil, a escolaridade do pai e da mãe, e a situação profissional do pai e da mãe. Como é que o modelo de dados lida com as *slowly changing dimension*? Quando são identificados estes casos, é inserido um novo registo na tabela de dimensão e na tabela de factos com o objetivo de não perder o histórico. Obviamente que, se isto acontecer com muita frequência as estrelas vão crescer exponencialmente e terá um impacto negativo no desempenho das consultas.

Quanto ao tipo de factos presentes nas quatros estrelas, são do tipo aditivos, sendo estes somados por todas as dimensões que constituem cada estrela.

Entretanto, a figura 18 ilustra o modelo em estrela que dá resposta aos indicadores do âmbito da frequência.

[14] Ficha de unidade curricular: indica o número de unidades curriculares cujo grau de preenchimento da informação relativa a ficha de unidade curricular é 0 % ou 100%.

[15] Indica o grau de preenchimento da informação relativa a ficha de unidade curricular pertencente a escala de 0 % até 100%.

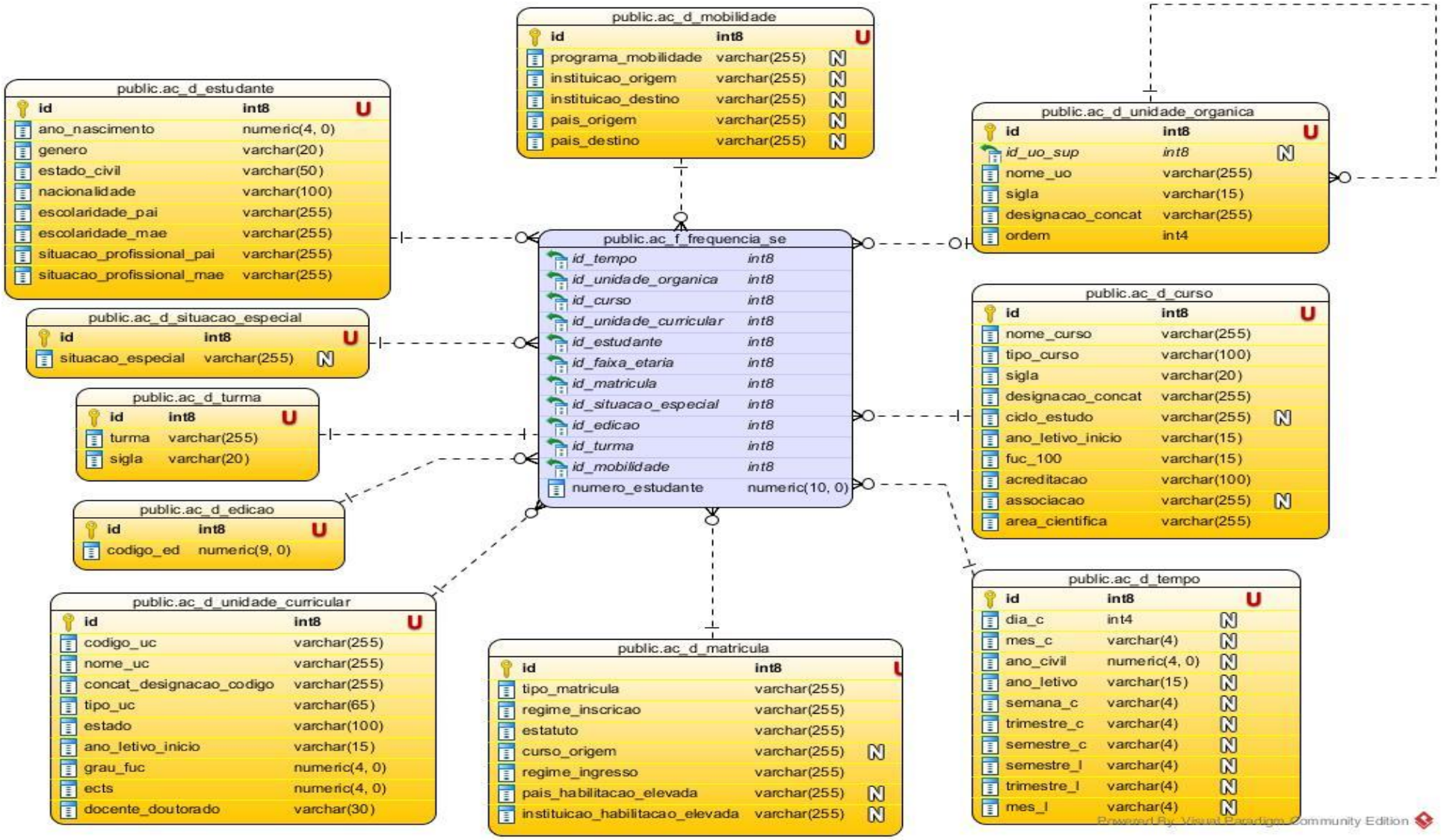


Figura 18 - Modelo de dados: Frequência

A estrela apresentada na figura 18, também tem dimensões *slowly changing dimension,* como é o caso das dimensões da matrícula, estudante, curso e unidade curricular. O tipo de matrícula e o regime de inscrição tornam a dimensão "ac_d_matricula" *slowly changing dimension,* porque pode ser possível que o estudante efetue um tipo de matrícula diferente que efetuou num ano anterior, assim como o regime de inscrição. Por exemplo, um estudante que está em mobilidade efetua, uma matrícula do tipo *incoming* num ano letivo e no ano letivo a seguir decide continuar na universidade e efetua uma outra matrícula do tipo normal. Tal como acontece nas outras dimensões descritas anteriormente, os estados dos carregamentos são mantidos permitindo assim manter o histórico.

Pode notar-se que a tabela de factos "ac_f_frequencia_se" tem o facto "numero_estudante". Este é um identificador único de cada estudante, uma vez que o facto deve refletir o número distinto por pessoa inscrita na Universidade de Coimbra. O facto é uma dimensão degenerativa (*degenerate dimension*), sendo uma chave e um facto para esta tabela. Isto permite evitar uma junção desnecessária entre a tabela de factos e a dimensão se esta fosse criada, tendo um impacto negativo nas consultas. Mas o principal motivo que se teve em consideração ao seguir esta abordagem foi o facto da tabela de factos armazenar informação ao nível das turmas. Isto permite que, existam muitos registos duplicados, por exemplo, se um estudante se inscrever numa unidade curricular e esta tiver uma turma teórica e prática, este estudante terá duas entradas na tabela de factos e assim sucessivamente. Estes foram os motivos para a adição do número de estudante na tabela de factos. Assim, caso se pretenda saber o número de estudantes inscritos na UC é efetuado um "*distinct count*" do número de estudantes para obter o valor correto. As restantes 15 estrelas do modelo de dados encontram-se descritas no anexo (F).

5.4.3. Espaço ocupado pela *data mart*

A tabela que se segue apresenta individualmente o espaço ocupado por cada tabela preenchida na *data mart,* assim como os índices. Os índices utilizados nas tabelas de factos são do tipo "*GIN*" (*Generalized Inverted Index*). Como o *PostgreSQL* não suporta índices do tipo *bitmap* optou-se por utilizar o *GIN* que tem as mesmas características de funcionamento, sendo frequentemente utilizados em tabelas de factos. Já as tabelas de dimensão utilizam índices do tipo *b-tree*. A tabela não contém informação dos espaços ocupados pelas vistas criadas para otimizar as consultas, uma vez que o *Mondrian* faz toda essa gestão através do cubo.

Tabelas de Factos	Espaço ocupado tabela	Espaço ocupado índice	Dimensões	Espaço ocupado tabela	Espaço ocupado índice
ac_f_candidatura	14 MB	11 MB	ac_d_candidatura	3576 KB	5376 KB
ac_f_curso	280 KB	352 KB	ac_d_concurso	8192 KB	80 KB
ac_f_disciplinas_isolada	192 KB	268 KB	ac_d_edicao	1544 KB	800 KB
ac_f_frequencia	170 MB	-	ac_d_estudante	1320 KB	2656 KB
ac_f_grau_satisfacao	64 KB	120 KB	ac_d_faixa_etaria	8192 KB	48 KB

ac_f_melhores_25_candidatos-dges	8182 KB	-	ac_d_info_grau_satisfacao	8192 KB	48 KB
ac_taxa_empregabilidade	32 KB	48 KB	ac_d_instituicao_parceira	8192 KB	48 KB
ac_f_unidade_curricular	4272 KB	5664 KB	ac_d_curso	144 KB	424 KB
ac_f_vagas	312 KB	328 KB	ac_d_isolada	8192 KB	48 KB
-	-	-	ac_d_lingua_ensino	8192 KB	48 KB
-	-	-	ac_d_matricula	816 KB	1064 KB
-	-	-	ac_d_mobilidade	136 KB	376 KB
-	-	-	ac_d_situacao_especial	8192 KB	48 KB
-	-	-	ac_d_tempo	8192 KB	48 KB
-	-	-	ac_d_turma	272 KB	104 KB
-	-	-	ac_d_unidade_curricular	3304 KB	3456 KB
-	-	-	ac_d_unidade_organica	32 KB	48 KB
Total:	215 MB		Total	60 MB	

Tabela 18 - Espaço ocupado pela *data mart*

Capítulo 6
Implementação

Este capítulo descreve a fase de implementação do projeto. O objetivo é descrever com detalhe as partes mais importantes de toda a implementação. Neste sentido, foi dividida nas seguintes fases: extração, transformação e carregamento (*ETL*), criação de cubos e criação dos *dashboards*, conforme descrito no subcapítulo 5.1 da arquitetura do sistema.

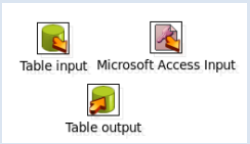
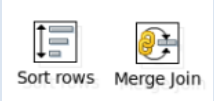
6.1. *ETL* (extração, transformação e carregamento)

O presente subcapítulo descreve com detalhe as transformações *ETL* utilizadas assim como os *jobs*. É nas transformações que é efetuado todo o processo de limpeza ou tratamento dos dados (por exemplo, identificar erros, uniformizar os dados, remover duplicados e fazer agregados). Por outro lado, os *jobs* permitem agendar as execuções das transformações e/ou outros *jobs*. Isto permite que sejam carregados e atualizados os dados numa *data mart* sem intervenção humana.

Entretanto, alguns âmbitos desenvolvidos neste projeto utilizam as mesmas tabelas na área de estágio, como é o caso das candidaturas do concurso nacional de acesso e via escola. Com o objetivo de não misturar dados de diferentes âmbitos na área de estágio, optou-se por extrair e carregar os dados na *data mart* por âmbito de indicadores.

6.1.1. Componentes das transformações

O *Pentaho Data Integration* utiliza um grande leque de componentes (do inglês *steps*) que são normalmente usados nas transformações do processo *ETL*. A tabela 19 faz a descrição de todos os componentes utilizados no processo *ETL* do projeto.

Componentes	Descrição
	O componente *table input* lê os dados de uma tabela de base de dados SQL assim como o *Microsoft Access input* lê os dados de tabelas de base de dados Access. Já o *table output* armazena os dados do fluxo da transformação numa tabela de base de dados SQL.
	O componente é usado para selecionar valores pretendidos de um fluxo assim como a sua precisão para o caso dos campos de tipo decimal.
	O componente *sort rows* é usado para ordenar um fluxo de dados em casos específicos. Por exemplo, o *Merge join* faz a junção de dois fluxos de dados mas precisam estar ordenados antes da junção. Nestes casos são utilizados o *sort rows*.
	O *switch/case* é um componente que separa o fluxo de dados em vários fluxos, consoante a condição. Já o *dummy* é usado para receber dados que não são importantes.

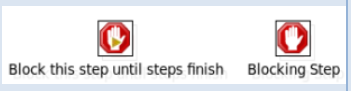
	O *block this step until steps finish* bloqueia um componente de ser executado até um outro terminar previamente definido pelo componente. O segundo bloqueia um step até receber todos os dados do fluxo.
	Este componente foi utilizado em algumas limpezas de dados. Permite alterar valores de campos de um fluxo de dados assim como atribuir novos.
	Este componente fornece várias funções de operações numéricas. Também permite, a extração do ano numa determinada data podendo ainda fazer arredondamentos.
	Este componente é utilizado quando se pretende extrair dados no sistema fonte, inicialmente os anos letivos para a extração estão como um array e é necessário separa-los individualmente.
	O primeiro componente é o usado para o carregamento das tabelas de factos. Antes de fazer qualquer operação o componente verifica: se o novo registo não existe na tabela, então insere. Se o novo registo existir e os campos para a atualização forem os mesmos, não faz nada. Mas se os campos não forem iguais ele atualiza. Já o *update* tem a função apenas de atualizar os dados.
	Este componente pode executar várias consultas SQL separadas por ponto e vírgula. Foi frequentemente usado para criar e apagar índices, apagar dados em tabelas, carregar tabelas de agregados, etc.
	Este componente permite eliminar linhas duplicadas num determinado fluxo dados. Estes não precisam ser ordenados previamente porque o próprio componente o faz.
	Este componente foi utilizado no preenchimento das tabelas de dimensões deste projeto. Antes dele inserir um novo registo, verifica se existe. Se não existir insere, caso contrário não faz nada.
	O primeiro componente permite fazer uma consulta numa tabela e devolve valores dos campos. O segundo tem a mesma função do primeiro com a diferença da consulta ser feita entre dois fluxos de dados. O último componente também permite procurar registos numa tabela. Estes foram os componentes utilizados para obter os identificadores das dimensões com o objetivo de serem inseridos nas tabelas de factos.
	Este componente permite gerar a relação entre pai e filho que existe numa determinada tabela. Foi o utilizado para preencher a tabela closure da unidade organica.

Tabela 19 - Descrição dos componentes usados do *Pentaho Data Integration*

6.1.2. Preenchimento da área de estágio

O preenchimento das tabelas da área de estágio foi conseguida através da combinação de transformações do *Pentaho Data Integration* e 11 funções desenvolvidas em *PLSQL*. Estas transformações são na grande maioria simples e parecidas, porque fazem a extração dos dados nos sistemas fonte e a seguir são armazenados nas tabelas da área de estágio. Entre a extração e o armazenamento é efetuado um conjunto de passos de limpeza e normalização dos dados, que começa com o componente *modified javascript value* e termina em funções *PLSQL*. Desta forma, a limpeza dos dados não afetará muito o tempo de execução de todo *ETL*.

A seguir é descrito o preenchimento das tabelas da área de estágio da candidatura do concurso nacional de acesso.

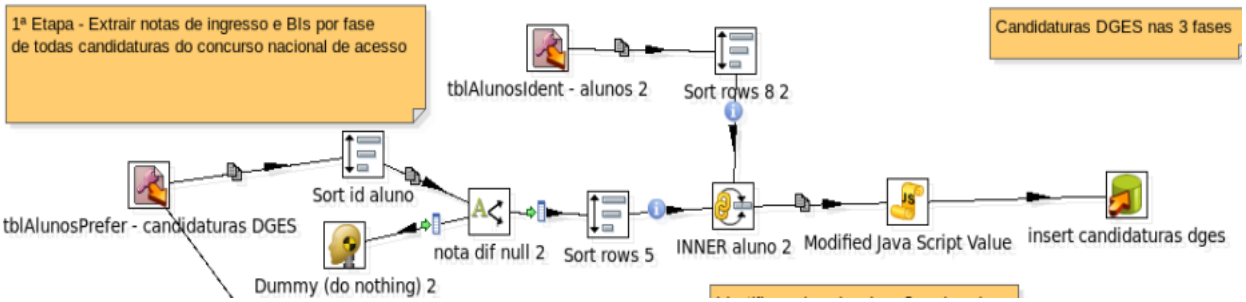


Figura 19 – 1ª Transformação: Parte 1: Preenchimento da área de estágio: DGES: 25 melhores candidatos

A figura 19 mostra que são extraídos dados de todas candidaturas do concurso nacional de acesso na tabela "tblAlunosPrefer - candidaturas DGES" e só serão considerados candidatos que têm notas diferentes de vazio, esta condição é imposta através do componente "nota diff null 2". Depois é feita uma junção do tipo "*INNER JOIN*" com a tabela "tblAlunosIdent - alunos 2" para obter o número do cartão de cidadão dos candidatos. Por fim, é armazenada a nota de candidatura, o número do BI e a fase do concurso na tabela "insert candidaturas dges". Esta informação vai permitir obter os melhores 25% candidatos do concurso nacional de acesso e identificar os que escolheram a UC.

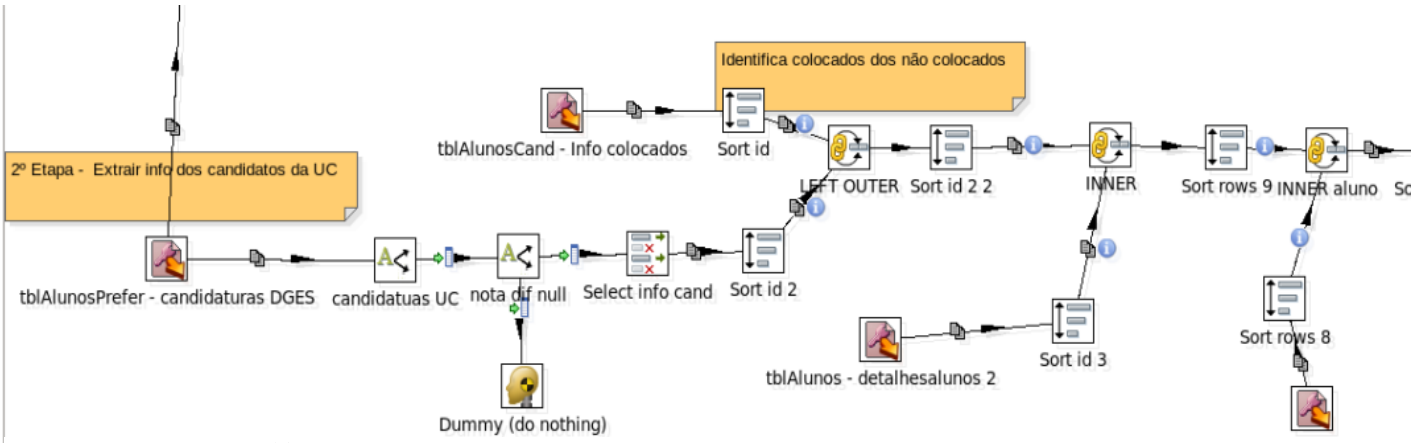


Figura 20 – 1ª Transformação: Parte 2: Preenchimento da área de estágio: DGES: Candidatos da Universidade de Coimbra

A figura 20 mostra como são extraídos os dados de todas as candidaturas da tabela "tblAlunosPrefer - candidaturas DGES" do concurso nacional de acesso. Destes, o componente designado como "candidaturas UC" deixa passar apenas os candidatos que escolheram os cursos da Universidade de Coimbra e o componente "nota diff null" descarta os candidatos cuja a nota é vazia. De seguida, são identificados os candidatos colocados dos não colocados através da junção do tipo "*left join*" desta tabela com a "tblAlunosCand - Info

colocados" (que contém a informação dos candidatos colocados). Os outros componentes permitem completar as informações em falta do candidato, como a data de nascimento, género, nome, o número do BI,etc.

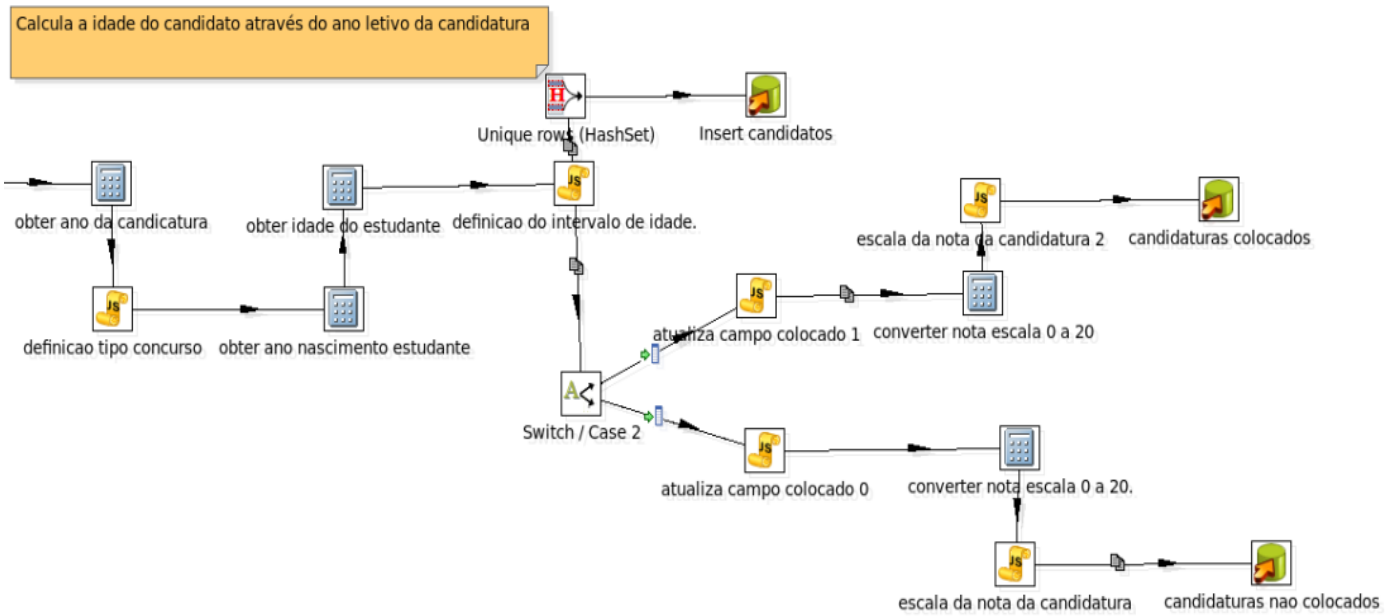


Figura 21 – 1ª Transformação: Parte 3: Preenchimento da área de estágio: DGES: Inserção das vagas

Como se pode visualizar através da figura 21, é obtido o ano letivo a partir da data da candidatura no componente "obter ano da candidatura", depois é também extraído o ano de nascimento através da data de nascimento, para se poder obter a idade atual dos candidatos (isto é feito no componente "obter idade do estudante" que recebe o ano da candidatura e o ano de nascimento do candidato para efetuar o cálculo). A informação dos candidatos é armazenada no componente "insert candidatos" e por fim, são identificados os colocados dos não colocados, e a nota de ingresso é convertida na escala de 0 a 20 valores e armazenada na tabela das candidaturas.

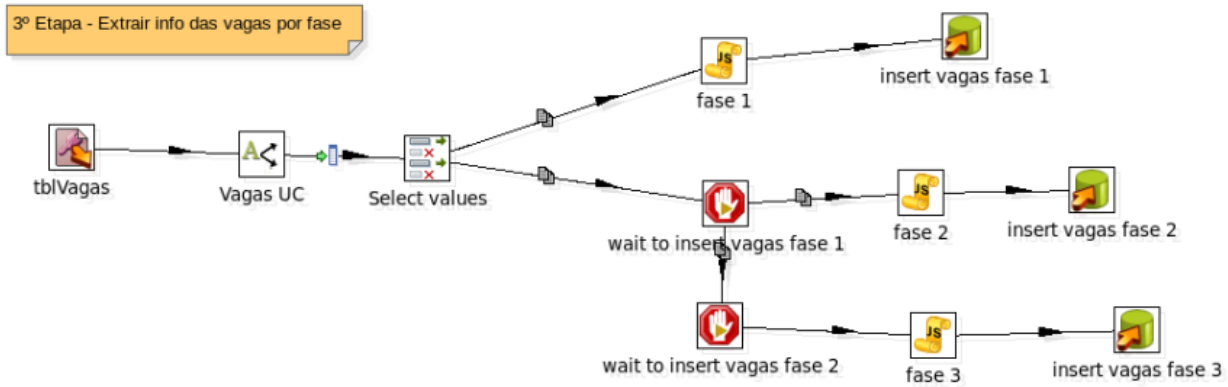


Figura 22 – 1ª Transformação: Parte 4: Preenchimento da área de estágio: DGES: Inserção dos Candidatos

A figura 22 mostra que são extraídas as informações do número de vagas por curso e por fase de candidatura. Primeiro é armazenada a informação da 1ª fase, e as outras ficam em espera até esta terminar (isto é feito através do componente "wait to insert vagas fase 1"). De seguida o fluxo da fase 3 também fica em espera que o da fase 2 termine para poder executar.

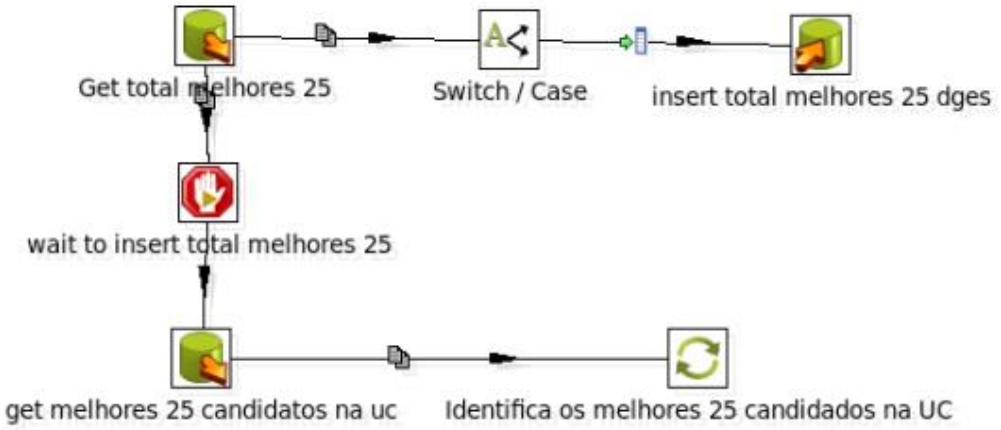


Figura 23 – 2ª Transformação: Parte 5: Preenchimento da área de estágio: DGES: Identificação dos melhores 25% que escolheram a UC

Inicialmente, a figura acima mostra como são obtidos em número absoluto os 25 melhores candidatos por fase do concurso nacional de acesso de um determinado ano letivo, e de seguida são armazenados. O segundo fluxo fica á espera até o "insert total melhores 25 dges" termine. Por fim, dos estudantes colocados na Universidade de Coimbra, identificam-se os que fazem parte dos 25% de melhores candidatos do concurso nacional de acesso. Depois de identificados é feita a atualização da informação desses candidatos na tabela "candidatura".

Dos candidatos colocados é necessário identificar os que efetuaram matrícula. A transformação apresentada na figura a seguir faz esse trabalho.

Figura 24 – 3ª Transformação: Parte 6: Preenchimento da área de estágio: DGES: Identificação dos candidatos que realizaram matrícula.

Como se pode verificar na figura 24, primeiro é obtido o ano letivo da candidatura no componente "get ano letivo na area de estagio", logo de seguida é feita uma consulta na vista "MVIEW_DEMOGRAFIA_MATRICULA" do NÓNIO em "read nonio". Da informação extraída são normalizados alguns campos no componente designado como "Normalizacao dos dados". O campo da nacionalidade tem valores do país do candidato, então esses valores são convertidos para "Nacionais" se o país for Portugal e "Estrangeiros" se o país for diferente de Portugal e "Desconhecida" se o campo estiver a vazio. Também é normalizado o estado civíl e género para "Indefinido" caso os campos estejam vazios. O resto da normalização dos campos é feita na função "atualizar_dados_demograficos_estudante()". Esta função tem como principal objetivo identificar os candidatos colocados que efetuaram matrícula. Isto é feito, através do número do BI (bilhete de identidade). Depois, é verificado, se o número de BI dos estudantes extraídos na vista existe na tabela das candidaturas, caso existam, o candidato colocado é identificado como matriculado. A partir deste momento as tabelas do "estudante" e da "matricula" têm toda a informação completa. As restantes transformações *ETL* para o preenchimento da área de estágio encontram-se no anexo J.

6.1.3. Preenchimento das dimensões

Com a área de estágio preenchida e com os dados devidamente tratados e consistentes é altura de efetuar o carregamento da *data mart*. Esta é a fase do preenchimento das dimensões, que é feita através do componente *Combination lookup/update* conforme descrito na secção 6.1.1. Este preenchimento é feito sem complexidade, porque os dados são carregados das tabelas da área de estágio e inseridos nas dimensões. As figuras a seguir demonstram o funcionamento completo dos mesmos.

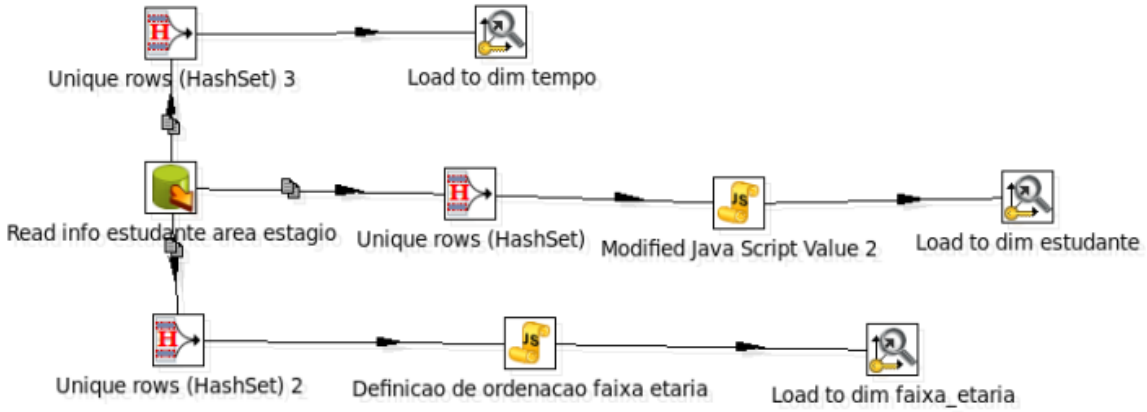


Figura 25 - Preenchimento das dimensões do tempo, estudante e faixa etária

Inicialmente é carregada toda a informação da demografia do estudante da tabela "estudante" da área de estágio. Antes de preencher as dimensões é utilizado o componente *"Unique rows (HashSet)"* com o intuito de eliminar registos repetidos. Apesar do componente *"Combination lookup/update",* usado no preenchimento das dimensões, também consegue eliminar registos repetidos. A utilização do primeiro componente *"Unique rows (HashSet)"* serve para reduzir o volume de dados desnecessários e desses já filtrados são enviados para o segundo componente, que voltará a verificar se os novos dados já existem ou não, no caso de não existirem são inseridos na dimensão. Portanto, antes da inserção dos dados na dimensão "ac_d_estudante", é feita uma última verificação em alguns campos (através do "*Modified java script value*") garantindo que estão consistentes e prontos para serem armazenados na dimensão "ac_d_estudante" através do "Load to dim estudante". Já o último fluxo da figura 25, depois dos dados serem filtrados, é criado o campo "ordem" no componente "Definicao de ordenacao faixa etaria" para ditar a ordem pela qual a faixa etária é apresentada no *dashboard* e de seguida são armazenadas através do "Load to dim faixa_etaria".

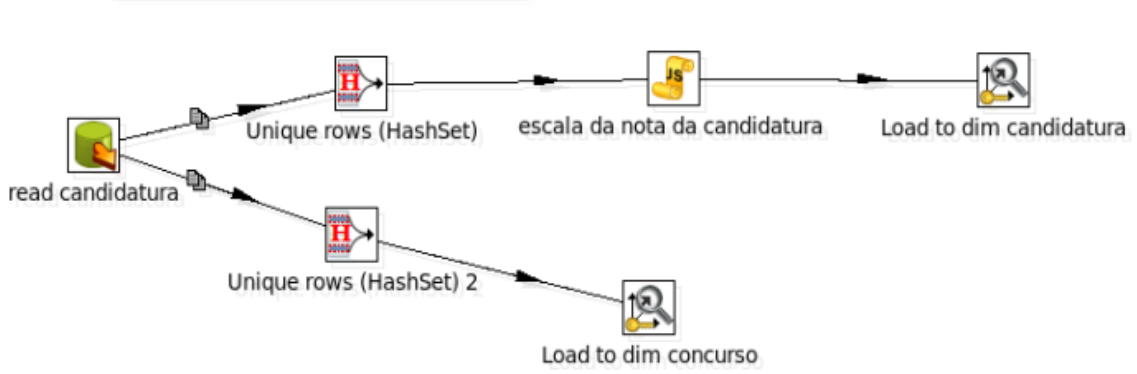


Figura 26 - Preenchimento das dimensões da candidatura e do concurso

O preenchimento da figura 26 é muito semelhante com o da figura 25. São carregadas as informações das candidaturas no componente "read candidatura" da área de estágio, de seguida são eliminados os registos repetidos e no componente "escala da nota da candidatura" é definida uma nova escala para as notas conforme definida na ficha de indicador das candidaturas. Depois os dados são armazenados na dimensão "ac_d_candidatura". Já o segundo fluxo, contém os dados devidamente tratados do concurso, que de seguida são armazenados na dimensão "ac_d_concurso".

6.1.4. Preenchimento das tabelas de factos

A fase do preenchimento das tabelas de factos é mais complexa. Porque é muito comum durante o carregamento das tabelas de factos existirem perdas de registos, causadas por diversos factores. Por vezes, o erro pode estar na consulta efetuada que carrega os dados da área de estágio durante alguma junção das várias tabelas ou mesmo a má configuração do componente responsável por preencher a tabela de factos.

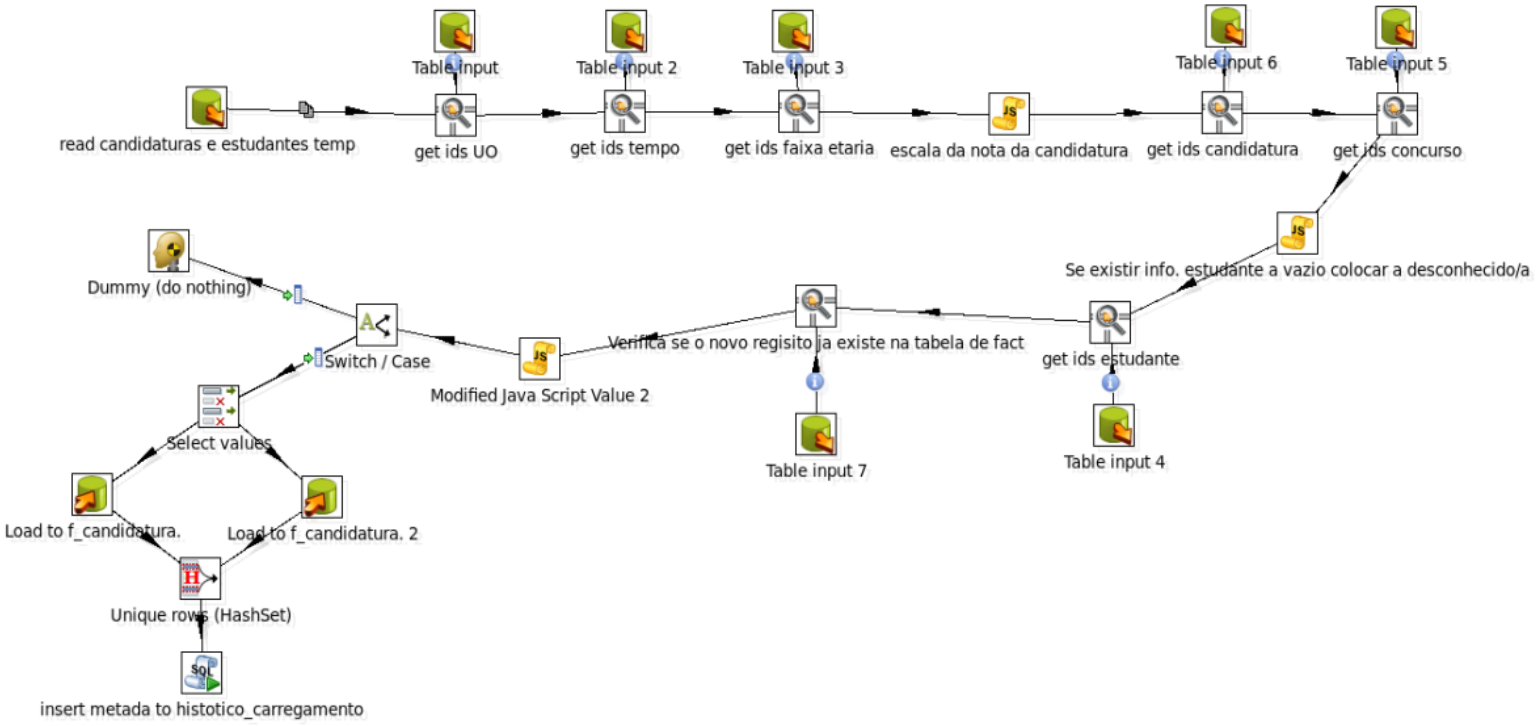


Figura 27 - Preenchimento da tabela de factos das candidaturas do concurso nacional de acesso

A figura 27 mostra a transformação utilizada para preencher a tabela de factos das candidaturas do concurso nacional de acesso. Inicialmente os dados são carregados da área de estágio através do componente "read candidaturas e estudantes temp". Nesta consulta é feita a junção dos dados da tabela da candidatura com a do estudante através do identificador em comum. Nestas consultas são efetuados agregados dos factos no nível de granularidade mais baixo que é o curso. Isto também só é possível porque os dados armazenados na dimensão do estudante são demográficos, o que permite a criação de melhores agregados. O segundo passo é obter os identificadores das seguintes dimensões: unidade orgânica, tempo, faixa etária, candidatura e estudante. Depois de obter todos os identificadores são verificados se estes existem na tabela de factos das candidaturas através do componente "Verifica se o novo registo ja existe na tabela de fact". Caso exista ou não a sequência procurada na tabela, o componente "Modified

java script value 2" cria a variável "**existe**" que vai assumir o valor vazio caso a consulta anterior retorne "**null**" ou "**1**", se a consulta devolver valores.

O componente "Switch/Case" recebe a variável "**existe**" e decide se os registos do fluxo serão descartados e redirecionados para o componente "Dummy (do nothing)" ou são enviados para os componentes "Load to f_candidatura" para serem armazenados.

O último passo é armazenar a data do carregamento dos dados, o ano letivo correspondente aos dados e a designação do âmbito através do componente "insert metada to historico_carregamento". Esta informação poderá ser utilizada em casos de auditoria na *data mart*, também é pretendido apresentar esta informação no *dashboard* para o conhecimento da data em que os dados foram carregados. Por fim, esta tabela é utilizada para definir uma janela temporal quando for necessário extrair novos dados e atualizar os antigos na *data mart*.

6.1.5. Componentes dos *jobs*

O *Pentaho Data Integration* é composto por transformações e *jobs*. Os *jobs* permitem que o processo *ETL* seja automatizado, dado isso, são descritas de forma breve, as funções de cada componente usado neste projeto.

Componentes	Descrição
	O primeiro componente serve para iniciar a execução de um *job* e o segundo permite terminar a execução, em caso de não ocorrerem erros.
	O componente permite executar uma transformação *ETL* de um determinado ficheiro.
	O componente permite executar outro *job*.
	O componente permite efetuar várias consultas SQL, foi normalmente utilizado na criação e remoção dos índices, etc.
	O componente permite suspender a execução de uma transformação ou *job* quando houver erros.

Tabela 20 - Descrição dos componentes usados nos *jobs*

6.1.6. *Jobs*

Um dos objetivos deste projeto é ter todo processo *ETL* automático. Isto, é feito através de um *job* principal, responsável por executar outros *jobs* com o intuito de extrair e atualizar os dados na *data mart*.

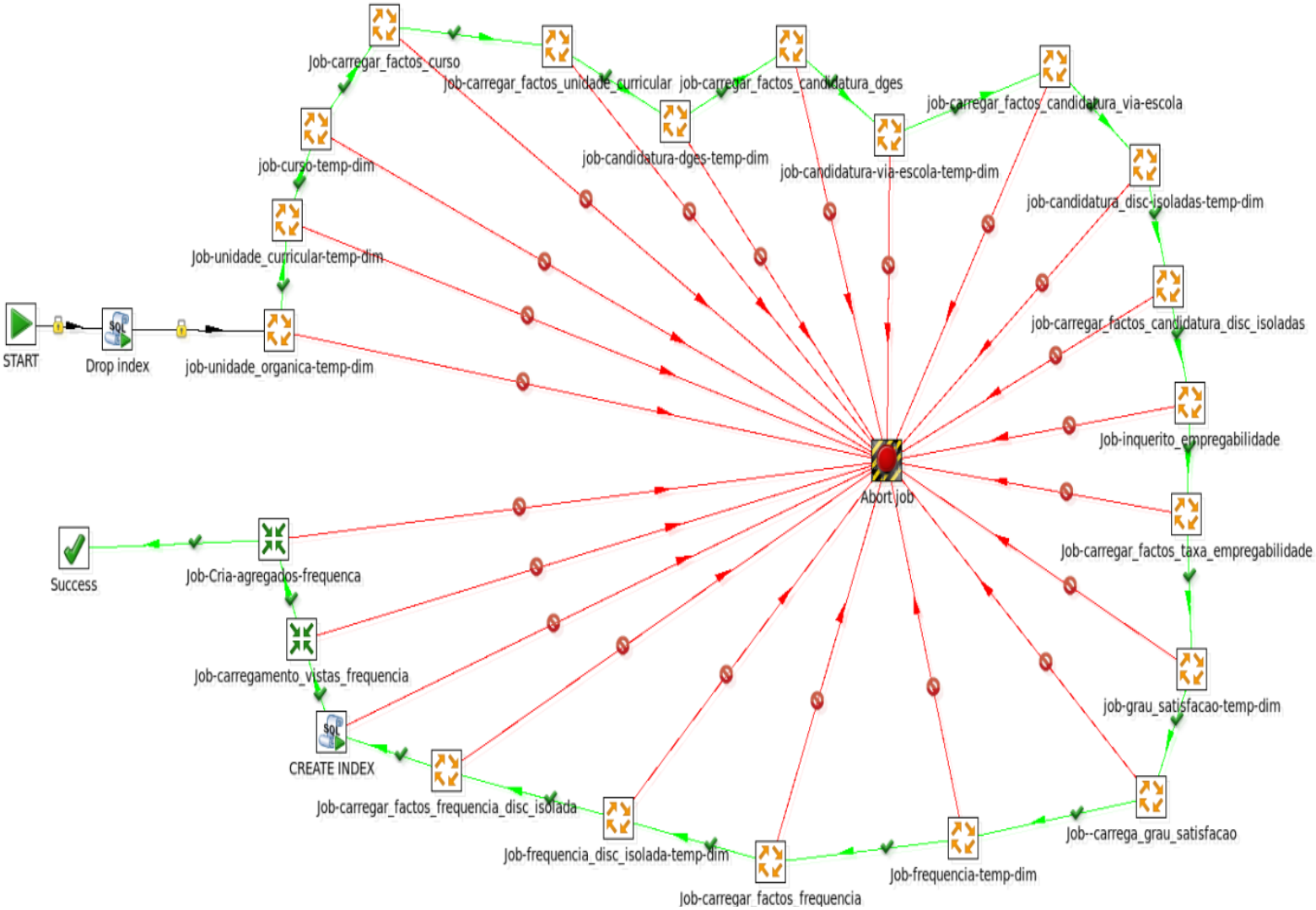


Figura 28 - Job responsável por extraír e atualizar a data mart

Como já foi dito, alguns âmbitos de indicadores utilizam as mesmas tabelas da área de estágio. Por isso, após o preenchimento da área de estágio de um determinado âmbito é de seguida preenchida a *data mart* como pode ser visualizado na figura acima. Quando o *job* principal é executado, começa-se por apagar todos os índices existentes na *data mart* através do componente "*Drop index*", de seguida, é preenchida a área temporária e as respetivas dimensões através dos componentes *jobs* cuja designação termina por "temp-dim" e das tabelas de factos são efetuados com os *jobs* cuja designação tem a palavra "factos". Assim que a última tabela de factos for preenchida, são criados os índices de toda *data mart* através do componente "*CREATE INDEX*". De seguida, são criadas três vistas materializadas através da tabela de facto da frequência e por fim, são criados os agregados para a mesma tabela com o objetivo de otimizar as consultas.

O primeiro carregamento da *data mart* foi de 42h03 minutos que corresponde a dados de 7 anos letivos (dos indicadores do concurso nacional de acesso, concurso via escola, curso,

unidades curriculares e frequência), 6 anos letivos (da taxa de empregabilidade), 5 anos letivos do indicador de grau de satisfação e 2 anos letivos das candidaturas a unidades curriculares isoladas. No entanto, 97,62% do tempo gasto no *ETL* é no âmbito da frequência visto que esta tabela de factos armazena inscrições dos estudantes ao nível das turmas. Já os outros 2,38% do tempo foram utilizado nos restantes âmbitos.

Durante todo desenvolvimento do *ETL*, tive a preocupação de experimentar várias abordagens com o intuito de obter melhores tempos de execuções. Mas o *bottleneck* no âmbito da frequência manteve-se, devido à extração de grande volume de dados (na ordem dos 3 milhões) relativos ao 7 anos letivos. É de realçar, que os próximos carregamentos vão demorar aproximadamente 9 horas.

6.2. Cubos *OLAP* (*OnLine analytical processing*)

O *Schema Workbench* é uma ferramenta que permite criar cubos *OLAP*. Estes permitem aceder à informação armazenada no modelo multidimensional de forma transparente e fácil. É composto por dimensões e *measures*. As *measures,* são factos de uma tabela de factos que estão associados a uma função de agregação, e estas podem ser contagens, contagens de valores distintos, somas, médias, valores mínimos e máximos. O cubo também permite criar *measures* que não existem na tabela de factos através de um "*Calculated member*". Um "*Calculated member*" é basicamente uma fórmula aplicada na coluna da tabela de factos com o intuito de criar uma nova *measure*. Foi usada esta abordagem por exemplo, para se saber a taxa de crescimento do número de estudantes a frequentar a UC. Primeiro, é necessário obter o número de estudantes inscritos do ano letivo anterior e depois o do ano letivo atual para se poder obter a taxa de crescimento.

A seguir são apresentados na tabela 21 os cubos criados neste projeto:

Nome do cubo	Tabela de factos
Candidaturas	ac_f_candidaturas
Vagas	ac_f_vagas
Melhores_25_candidatos_dges	ac_f_melhores_25_candidatos-dges
Disciplinas_isoladas	Ac_f_disciplinas_isoladas
Frequencia	ac_f_frequencia
Curso	ac_f_curso
Unidade_curricular	ac_f_unidade_curricular
TaxaEmpregabilidade	ac_f_taxa_empregabilidade
GrauSatisfacao	ac_f_grau_satisfacao

Tabela 21 - Cubos *OLAP* e as respetivas tabelas usados neste projeto

O cubo *OLAP* facilita o acesso da informação armazenada no modelo multidimensional. Este acesso, é feito através de consultas em *MDX* (*Multidimensional Expressions*). Esta linguagem aparenta ser um pouco similar com a linguagem *SQL* mas a sua sintaxe é muito diferente.

Sendo uma linguagem nova e diferente foi necessário recorrer ao *plugin Saiku Analytics* para auxiliar no desenvolvimento das consultas para os *dashboards*. À medida que se avançava no desenvolvimento foi ficando mais evidente que o *MDX* tem algumas limitações, pois, esta não permite por exemplo, fazer subconsultas que teriam sido úteis em algumas situações. A seguir é apresentada um exemplo de uma consulta em *MDX* utilizada neste projeto.

```
select NON EMPTY {[Measures].[candidato_colocado]} ON COLUMNS,
  NON EMPTY {[ac_d_curso_desagregacao.hie_designacao_concat].[designacao_concat].Members} ON ROWS
from [Candidaturas]
where Crossjoin(Crossjoin(Crossjoin(Crossjoin({[d_tipo_curso.hie_tipo_curso].[Conferente de grau]},
{[ac_d_tempo.hie_ano_letivo].[2008/2009]}), {[ac_d_unidade_organica_filtro.hie_ac_d_unidade_organica_filtro].[FCTUC].[DEI]}),
{[d_tipo_concurso.hie_tipo_concurso].[dges]}), {[d_fase_concurso.hie_fase_concurso].[1ª fase]})
```

Figura 29 - Exemplo de uma consulta *MDX*

A primeira linha da consulta devolve o número total de candidatos colocados, a segunda designada como "*ON ROWS*" devolve a designação dos cursos através do cubo Candidaturas. Por último, a informação é filtrada em 1ª fase do concurso nacional de acesso do ano letivo 2008-2009 e por Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologias através da cláusula "*where*". Ou seja, a consulta apresentada na figura anterior mostra o número total de candidatos colocados na 1ª fase do concurso nacional de acesso do ano letivo 2008-2009, nos Cursos do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologias.

6.3. Servidor *OLAP*

O servidor *OLAP* deste projeto é o *Pentaho BI Server*, foi onde foram desenvolvidos os *dashboards*. O desenvolvimento dos mesmos teve o auxílio do *plugin CDE* (*Community Dashboard Editor*) com suporte a *HTML, CSS e JavaScript*.

Inicialmente o *plugin* parece confuso, visto que é preciso perceber o funcionamento de cada componente, assim como a relação entre os diferentes componentes (*layout panel, components panel e data sources*). É no componente *layout panel* que foi desenvolvido todo o *design* dos *dashboards* através de *HTML e CSS* com ajuda da *framework Bootstrap*. Também foi utilizada uma outra biblioteca externa para criar os gráficos dos *dashboards,* uma vez que permite uma melhor personalização em relação aos gráficos nativos do *CDE,* o *Highcharts*.

É no *components panel* que foram criados os formulários/filtros assim como os componentes dos gráficos, tabelas e *scripts*. Toda a iteração existente nos *dashboards* é gerida por ele através de *JavaScript*. O *components panel* recebe as instruções vindas do *dashboard*, constrói as respetivas consultas em *MDX* e envia-as para os *data sources,* que são responsáveis por criar ligações aos dados para as consultas serem executadas. Quando o *data source* obtiver os resultados, estes são enviados para o *componente panel* onde são formatados consoante as séries dos gráficos ou tabelas e por sua vez são reenviados para serem apresentados no *dashboard*. Dado isso, o objetivo do desenvolvimento é os *dashboards* serem dinâmicos assim como a construção das consultas em *MDX*. Neste sentido, foram desenvolvidas várias funções em *JavaScript* para tornar os *dashboards* o mais dinâmico quanto possível.

6.4. Otimização

Durante o desenvolvimento dos *dashboards* foi tida em consideração que a má programação do *Java Script* traz um impacto negativo no desempenho dos mesmos. Por isso, foi evitado ao máximo código *Java Script* desnecessário e foi garantido que o código está otimizado. A outra preocupação desde o início, foi garantir que as *queries* usadas nos *dashboards* eram as mais eficazes. Apesar destas considerações, alguns *dashboards* continuavam com baixo desempenho, por exemplo, o da frequência no primeiro acesso demorava 5 minutos a apresentar os dados e aproximadamente 30 minutos quando se aplicava uma desagregação.

Dado isso, foi desenvolvido um plano de 3 passos para otimizar a *data mart,* o primeiro passo, foi criar índices *b-tree* nas chaves primárias das dimensões e em alguns campos. Nas chaves estrangeiras das tabelas de factos foram criados índices do tipo "*GIN*" (*Generalized Inverted Index*). O *PostgreSQL* não suporta índices do tipo *bitmap* mas suporta o "*GIN*" que tem o mesmo funcionamento. Este tipo de índice consegue melhores desempenhos e ocupa menos espaço que a *btree* nas tabelas de factos porque estas armazenam muitos dados duplicados. Os índices melhoraram o desempenho dos *dashboards,* à exceção do âmbito da frequência que continuava com baixo desempenho.

O segundo passo foi criar três vistas materializadas da tabela de factos da frequência e índices. Isto tornou as consultas muito rápidas, inicialmente as consultas demoravam em média 5 minutos, e com esta otimização as consultas passaram a demorar 5 segundos. Entretanto, ainda existiam algumas consultas lentas, que demoravam cerca de 20 segundos no nível de granularidade das unidades orgânicas. Estas consultas foram identificadas e o terceiro passo foi criar alguns agregados sobre essas.

6.5. Produto final

Na fase de definição dos indicadores, foram definidos 42 indicadores. Dos quais 18 correspondem a prioridade elevada, 14 de prioridade média e 8 de prioridade baixa.

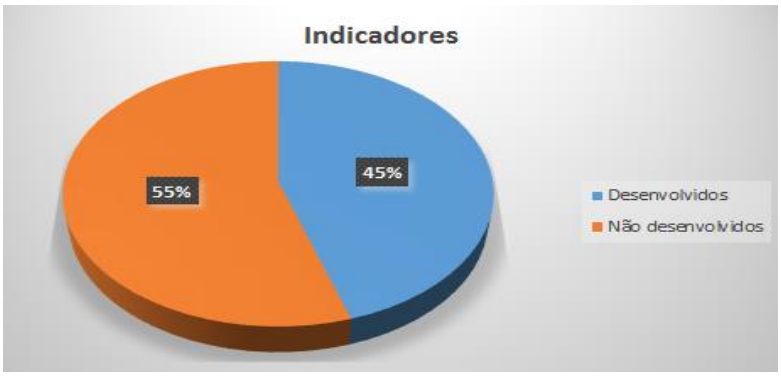


Figura 30 - Relação dos indicadores desenvolvidos dos não desenvolvidos

A figura 30 mostra que foram desenvolvidos 45% dos indicadores. Esta percentagem corresponde a prioridade elevada. Os restantes 55% correspondem aos indicadores de prioridade média e baixa que não foram desenvolvidos.

Entretanto, no início do desenvolvimento pensei em desenvolver um *dashboard* para cada indicador. Mas muito rapidamente notei que os indicadores do mesmo âmbito, tinham por vezes os mesmos filtros mas, com pequenas diferenças. Neste sentido, foram desenvolvidos

6 *dashboards* consoante o âmbito dos indicadores. Isto permitiu uma melhor reutilização do código criado e também evitar a criação de *dashboards* desnecessários. O primeiro *dashboard* desenvolvido foi do âmbito da candidatura. Composto por 12 indicadores relativos às candidaturas do concurso nacional de acesso, concurso via e as candidaturas as unidades curriculares isoladas. O segundo *dashboard* foi o da frequência, que é composto por 2 indicadores. O terceiro foi da unidade curricular com 1 indicador. O quarto foi o curso com 1 indicador. O quinto foi o da taxa de empregabilidade com 1 indicador. O sexto e último foi o do grau de satisfação também com 1 indicador.

O produto final apresenta algumas diferenças em comparação aos protótipos. Por exemplo, quando é aplicada uma desagregação no *dashboard*, esta só se refletia no gráfico de *snapshot* (ver subcapítulo 4.2). Já a versão final, apresenta a desagregação nos dois gráficos e tabelas. Também foi melhorado alguns aspetos de *design* da aplicação. Dado isto, a figura a seguir mostra o *dashboard* com a evolução da taxa de preenchimento de vagas do concurso nacional de acesso na Universidade de Coimbra nos últimos 7 anos letivos (gráfico da esquerda). E o gráfico à direita (*snapshot*) mostra a taxa de preenchimento de vagas no ano letivo 2014-2015. O gráfico de *snapshot* tem o objetivo de apresentar a informação com maior nível de detalhe do que o gráfico de evolução

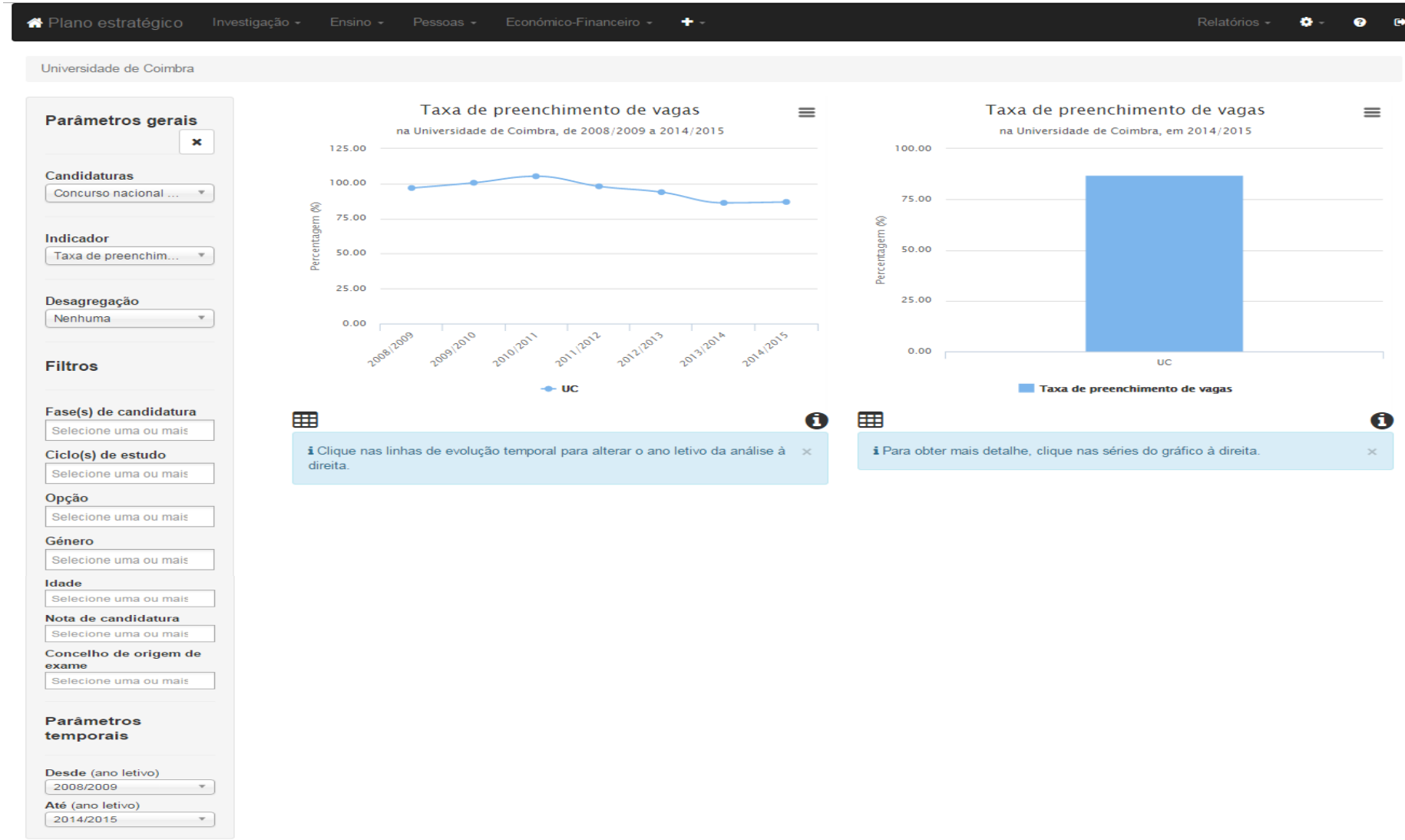


Figura 31 - *Dashboard* com a taxa de preenchimento de vagas do concurso nacional de acesso

Capítulo 7
Validação

A validação é uma das fases mais importantes para qualquer projeto. É nesta fase que são validados os dados e normalmente identificadas algumas incoerências de implementação do produto final em relação à especificação, faltando por vezes algumas funcionalidades presentes na especificação. O presente capítulo descreve todo processo utilizado na validação das funcionalidades e dos dados.

Validação interna

Durante o desenvolvimento do *ETL* preocupei-me em garantir que os dados da *data mart* correspondiam aos dados dos sistemas fonte. Depois disso, o segundo passo foi validar os dados apresentados nos *dashboards*. Como a maioria dos indicadores desenvolvidos são relativos a dados públicos, encontrei rapidamente documentos oficiais e relatórios de gestão de contas disponíneis na internet.

Os dados relativos ao concurso nacional de acesso foram validados através dos dados estatísticos disponibilizados pela Direcção Geral do Ensino Superior[16]. Os outros dados relativos às candidaturas do concurso via escola, frequência, unidades curriculares e cursos foram validados através do relatório de gestão de contas da Universidade de Coimbra do ano de 2013[17].

Antes dos *dashboards* serem disponibilizados para validação, foi criado um documento com testes funcionais. Foram criados 312 casos de testes (consultar anexo (K)) para garantir que todas as funcionalidades foram implementadas e corrigir eventuais erros que foram surgindo nos *dashboards* e ainda foram efetuados vários outros testes funcionais pelos colegas da equipa.

Validação externa

Antes dos *dashboards* serem enviados para os *stakeholders* responsáveis pela a validação do módulo académico, foram criadas fichas de validação para cada âmbito de indicadores com três objetivos: validar a usabilidade da aplicação em geral, correspondência com os requisitos e por último, e mais importante, a validação dos dados apresentados nos indicadores. A Dr.ª Helena Galante da Divisão de Avalição e Melhoria Contínua ficou responsável por validar este módulo.

No dia 22 de junho foram enviadas as fichas de validação (consultar o anexo (L)) para a Dr.ª Helena Galante. Mas o processo de validação ainda não foi concluído e continua a decorrer. No entanto, já permitiu identificar e corrigir problemas no sistema desenvolvido e, mais importante ainda, nos sistemas fonte.

[16] http://www.dges.mec.pt/guias/detcursopi.asp?frame=1&codc=9119&code=0501#lev4
[17] http://www.uc.pt/dpgd/doc_gestao/relatorio_gestao_contas_UC_2013.pdf

Capítulo 8
Metodologia de teste

Esta fase é uma das mais importantes, uma vez que permite identificar erros do sistema e avaliar a qualidade da informação apresentada pelo mesmo. Neste sentido, foram efetuados dois tipos de testes: de unidades e *black-box-testing*.

Durante o desenvolvimento do *ETL* foram efetuados vários testes de unidades com o intuito de garantir a qualidade dos dados e o correto funcionamento do mesmo. Uma vez que o objetivo é o processo *ETL* funcionar de forma automática.

Já o *front-end* (*dashboard*), foi testado através das duas abordagens mencionadas anteriormente. Numa primeira fase, o *front-end* foi testado com testes de unidade porque foram frequentemente verificados erros de programação durante todo o processo de desenvolvimento dos *dashboards*. Não foi uma tarefa fácil efetuar estes, uma vez que o *BI Server* tem uma forma de fazer *debugging* do sistema muito limitado. Por forma a ultrapassar esta limitação foi necessário recorrer à ferramenta *Firebug* [18].

Numa segunda fase, os testes seguiram a abordagem *black-box-testing*, onde o utilizador testa o sistema inserindo apenas um *input* e espera que este devolva um *output*. As funcionalidades são determinadas observando as saídas das entradas introduzidas.

O programador criou casos de testes consoante os requisitos funcionais implementados e presentes no subcapítulo 4.3. Aos testes efetuados foram tidas em conta todas as operações de *slice and dice, drill-down* e *roll-up*. Os casos de testes foram especificados como demonstrado na tabela 22, onde cada linha da tabela corresponde a um caso de teste. Também serviram de testes de aceitação que foram enviados para o cliente validar o projeto.

ID	Descrição	Ação	Resultado Esperado	Resultado Obtido	Validação
TF_AC_CC_YY	O utilizador consegue fazer login?	Fazer login (Credenciais corretas)	Entrar no sistema		Sim/Não

Tabela 22 - Exemplo de caso de teste - Abordagem *black-box-testing*

Legenda:

TF: Teste Funcional.
AC: Académicos;
CC: Categoria (Gerais, indicadores);
YY: número teste

Entretanto, à medida que o desenvolvimento dos *dashboards* fosse concluído, eram criados os respetivos casos de testes, pois através destes era possível verificar o comportamento completo das funcionalidades e se estas correspondiam ao esperado. Ao seu todo foram criados 312 casos de testes que garantiram o correto funcionamento dos *dashboards*.

[18] É uma ferramenta *open source* que facilita efetuar testes em tempo real de uma página web desenvolvida com *HTML*, *CSS*, *JavaScript*, etc.

Capítulo 9
Conclusão

O presente projeto apresentou grandes desafios durante todo o desenvolvimento. Inicialmente com o objetivo de perceber melhor o projeto, efetuei um estudo das soluções de *BI* disponíveis atualmente assim como a efetuação de testes ao módulo "Sucesso escolar" presente na plataforma web da *data warehouse* da UC. Depois foram definidos os indicadores de desempenho da aplicação com a ajuda de protótipos rápidos desenvolvidos. Estes foram analisados pelo grupo operacional e validados pela Vice-reitora Prof.ª Dr.ª Madalena Alarcão responsável. Com a fase anterior concluída, foram definidos os modelos de dados para a área de estágio e a *data mart,* é de salientar que se conseguiu cumprir com sucesso o planeamento do primeiro semestre. A fase da implementação, por um lado foi complexa e por outro, interessante tanto no desenvolvimento do *ETL* como dos *dashboards*. Porque foi muito exigente em termos de capacidade e esforço. Apesar do planeamento do segundo semestre sofrer alguns reajustes durante o desenvolvimento e a validação dos *dashboards* ainda não estarem concluídos, o balanço do segundo semestre também foi positivo porque conseguiu-se cumprir com o objetivo principal: implementação dos indicadores de prioridade elevada.

O sistema desenvolvido ajuda os *stakeholders* da Universidade de Coimbra a acompanhar a evolução das atividades académicas, ao longo dos anos. Também permite lhes obter indicadores de desempenho em tempo útil. Assim como, saber se estão a conseguir alcançar os objetivos definidos no plano estratégico.

Quanto ao trabalho futuro, podia-se criar um *back-end* que permiterá ao administrador do sistema remover ou adicionar novos filtros. Isto permitirá acrescentar mais análises sobre os indicadores. É preciso migrar as ferramentas *Schema Workbench* e *bi-server* para novas versões já disponíveis, com o intuito de ultrapassar as limitações que estas têm. Apesar da plataforma web apresentar bom desempenho e de existir agregados no âmbito da frequência, ainda é necessário criar agregados para as restantes tabelas de factos da *data mart*.

No entanto, o projeto permitiu consolidar e aprofundar os conhecimentos de engenharia de *software* e especificamente a área de *business intelligence*. Portanto, posso concluir que o contacto com os clientes nas reuniões da definição dos indicadores e os desafios travados durante o desenvolvimento do projeto me permitiram adquirir competências que não tinha antes, crescer a nível profissional e ter uma melhor preparação para o mundo de trabalho.

Anexos

[A] Documento do plano estratégico utilizado na definição dos indicadores:
- Plano Estrate¦ügico e de Ac¦ºa¦âo UC 2011-2015 (final).pdf

[B] Pasta que contém os indicadores:
- Indicadores

[C] Protótipos rápidos produzidos que ajudou a melhorar a definição dos indicadores. encontram-se em:
- prototipo_academicos.vp

[D] Imagem do modelo de dados da área de estágio:
- UC-NUM _modelo_area_estagio.jpg

[E] Imagem do modelo de dados multidimensional:
- UC-NUM _modelo_em _estrela.jpg

[F] Documento que descreve o modelo de dados multidimensional:
- UC-NUM_descricao_modelo_em_estrela.pdf

[G] Documento que descreve as vistas materializadas do NÓNIO:
- UC-NUM_REQ_AC_documento_das_vistas_materializadas_2015-02-26.pdf

[H] Documento que descreve a replica da base de dados do sistema Request tracker:
- UC-NUM_REQ_AC_documento_da_replica_bd_rt_2015-02-16.pdf

[I] O modelo da área de estágio no formato SQL:
- UC-NUM_AC_modelo_area_estagio_SQL.sql

[J] Documento com a descrição completa das transformações ETL para o preenchimento da área de temporária e *data mart*:
- UC-NUM-AC_descricao_ETL.pdf

[K] Documento com os testes funcionais:
- UC-NUM_TST_testes-funcionais_Carlos.pdf

[L] Documentos designados como fichas de validação, que foram utilizados na validação da plataforma:
- UC-NUM_TST_AC_ficha_validacao_candidaturas.pdf
- UC-NUM_TST_AC_ficha_validacao_cursos.pdf
- UC-NUM_TST_AC_ficha_validacao_estudante.pdf
- UC-NUM_TST_AC_ficha_validacao_frequencia.pdf
- UC-NUM_TST_AC_ficha_validacao_unidades_curriculares.pdf

Referências

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

[2] Pentaho Kettle Solutions, Building Open Source *ETL* Solutions with Pentaho Data Integration, disponível em:
http://dl.e-book-free.com/2013/07/pentaho_kettle_solutions.pdf

[3] Descrição completa do MULE ESB, disponível em:
http://www.mulesoft.org/what-mule-esb

[4] O que podes fazer com o Muler? Documentação disponível em::
http://www.mulesoft.org/what-mule-do. [Accessed 13 Mar 2014].

[5] Descrição completa do Google Refine, disponível em:

http://en.wikipedia.org/wiki/OpenRefine

[6] Informação completa da metodologia Scrum, disponível em:

http://pt.wikipedia.org/wiki/Scrum#Pap.C3.A9is_principais

[7] Estudos de comparação entre o SGBD Postgresql vs MySQL:

https://www.techunblocked.org/2015/01/postgresql-vs-mysql.html

[8] TPC-H utilizado para avaliar o desempenho das bases de dados relacionais:

http://www.pilhokim.com/index.php?title=Project/EFIM/TPC-H

[9] Dados estatísticos do concurso nacional de acesso, deponibilizado pela Direção Geral de Ensino Superior em:

http://www.dges.mec.pt/guias/detcursopi.asp?frame=1&codc=9119&code=0501#lev4

[10] Relatório de gestão de contas da Universidade de Coimbra do ano 2013, diponível em:

http://www.uc.pt/dpgd/doc_gestao/relatorio_gestao_contas_UC_2013.pdf