

Bases Dados e *Data Warehousing*

UAb – Análise do abandono escolar e sucesso escolar
(Relatório Final)

Nuno José Alvela dos Santos (aluno nº 1200416)

Projeto Final do curso Licenciatura de Engenharia Informática

Trabalho orientado pelo professor Luis Cavique

Junho 2018

ÍNDICE

CONTEÚDO

Índice	i
Índice de Figuras	iii
Índice de Tabelas.....	v
Siglas	vi
1. Introdução	1
1.1. Introdução	1
1.2. Tema	1
1.3. Enquadramento	1
1.4. Objetivos	2
1.5. Planeamento.....	3
1.6. Organização do Relatório	4
2. Estratégia.....	5
2.1. Introdução	5
2.2. <i>Data Warehouse</i>	5
2.3. Ferramentas.....	9
3. Modelo OLTP.....	11
3.1. Introdução	11
3.2. Diagrama global	12
3.3. Modelo Moodle	13
3.4. Modelo Portal Académico.....	14
3.5. Modelo Portal UAb	19
3.6. Modelo Sitcon	22
4. Dados (carregamento)	25
4.1. Introdução	25
4.2. Modelo moodle	25
4.3. Modelo Portal UAb	26
4.4. Modelo Portal Académico.....	27
4.5. Modelo Sitcon	30
5. Modelo OLAP (DW)	30
5.1. Introdução	30
5.2. Dimensões e Factos (tabelas).....	31
5.3. Sucesso Escolar / Abandono Escolar (Definição).....	32
5.4. Estrela Aluno	32

5.5.	Estrela UC	35
5.6.	Estrela Curso	36
6.	ETL.....	38
6.1.	Introdução	38
6.2.	Definição dos componentes	39
6.3.	Tabelas de Dimensões.....	41
6.4.	Tabelas de Factos	60
7.	<i>Bussiness Intelligence</i>	74
7.1.	Introdução	74
7.2.	Preparação Inicial.....	74
7.3.	Métricas (DAX)	76
7.4.	Aluno	78
7.5.	Curso	83
7.6.	UC	86
7.7.	Publicação dos Relatórios	89
8.	Conclusões.....	90
8.1.	Trabalho realizado.....	90
8.2.	Trabalho futuro	91
	Anexo I – Código para popular bases de dados OLTP.....	92
	Anexo II – Manual de Instalação/utilização	110
	Referências bibliográficas	110

ÍNDICE DE FIGURAS

Figura 2.1 - Fluxograma: Inscrições E Avaliações	6
Figura 2.2 - Infraestrutura OLTP, OLAP, BI e Clientes	11
Figura 3.1 - Modelo Global.....	12
Figura 3.2 - Modelo lógico do subsistema Moodle	13
Figura 3.3 - Modelo Físico Do Subsistema Moodle	13
Figura 3.4 - Modelo lógico do subsistema Portal Académico.....	14
Figura 3.5 - Modelo físico do subsistema Portal Académico	15
Figura 3.6 - Modelo lógico do subsistema Portal UAb	19
Figura 3.7 - Modelo físico do subsistema Portal UAb	20
Figura 3.8 - Modelo lógico do subsistema SITCON	23
Figura 3.9 - Modelo físico do subsistema SITCON	24
Figura 5.1 - Estrela Aluno	33
Figura 5.2 - Estrela UC	35
Figura 5.3 - Estrela Curso	37
Figura 6.1 - <i>OLE DB Source</i>	39
Figura 6.2 - <i>Slowly Changing Dimension</i>	39
Figura 6.3 - <i>ADO NET Destination</i>	40
Figura 6.4 - <i>Task Factory Fact Table Destination</i>	40
Figura 6.5 - Merge Join	40
Figura 6.6 - Sort.....	40
Figura 6.7 - Ligação UAb-DW.....	41
Figura 6.8 - Fluxo De Controlo.....	42
Figura 6.9 - <i>Container "Portal Académico"</i>	43
Figura 6.10 - Fluxo Estado Civil	43
Figura 6.11 - Passo 1 do <i>OLE DD Source Editor</i>	44
Figura 6.12 - Passo 2 do <i>OLE DB Source Editor</i>	44
Figura 6.13 - Passo 3 do <i>OLE DB Source Editor</i>	45
Figura 6.14 - Passo 1 do SCD Wizard.....	46
Figura 6.15 - Passo 2 do SCD Wizard.....	46
Figura 6.16 - Passo 3 do SCD Wizard.....	47
Figura 6.17 - Passo 1 do <i>OLE DB Destination Editor</i>	48
Figura 6.18 - Passo 2 do <i>OLE DB Destination Editor</i>	48
Figura 6.19 - Passo 3 do <i>OLE DB Destination Editor</i>	49
Figura 6.20 - Fluxo Sexo	50
Figura 6.21 - Fluxo Estado Matricula	50
Figura 6.22 - Fluxo Ano Lectivo	51
Figura 6.23 - Fluxo Aluno	52
Figura 6.24 - Tipos de alterações dos atributos da tabela DimAluno.....	52
Figura 6.25 - <i>Container "Portal"</i>	53
Figura 6.26 - Fluxo Curso.....	53
Figura 6.27 - Tipos de alterações dos atributos da tabela DimCurso	54
Figura 6.28 - Fluxo Departamento.....	54
Figura 6.29 - Tipos de alterações dos atributos da tabela DimDepartamento	55
Figura 6.30 - Fluxo UC.....	55
Figura 6.31 - Tipos de alterações dos atributos da tabela DimUnidadeCurricular	56
Figura 6.32 - Fluxo Centro Local de Aprendizagem.....	57
Figura 6.33 - Tipos de alterações dos atributos da tabela DimUnidadeCurricular	57
Figura 6.34 - Fluxo Docente	58
Figura 6.35 - Tipos de alterações dos atributos da tabela DimDocente	58

Figura 6.36 - Container "Comuns"	59
Figura 6.37 - Fluxo Código Postal	59
Figura 6.38 - Tipos de alterações dos atributos da tabela DimCodigoPostal.....	60
Figura 6.39 - Container Factos.....	61
Figura 6.40 - Fluxo do processo ETL da tabela de Factos Aluno.....	61
Figura 6.41 - Junção das entidades [AvaliacaoUnidadeCurricular] e [Aluno]	63
Figura 6.42 - Junção do fluxo (4) e da tabela [Morada]	64
Figura 6.43 - Lookups da tabela FactosAluno	65
Figura 6.44 - Mapeamento das colunas de destino da tabela Factos Aluno.....	66
Figura 6.45 - Fluxo do processo ETL da tabela de Factos UC.....	67
Figura 6.46 - Lookups da tabela FactosUC	69
Figura 6.47 - Mapeamento das colunas de destino da tabela Factos UC.....	70
Figura 6.48 - Fluxo do processo ETL da tabela de Factos Curso.....	71
Figura 6.49 - Lookups da tabela FactosCurso	73
Figura 6.50 - Mapeamento das colunas de destino da tabela Factos Curso.....	74
Figura 7.1 - Ligação a origem de dados: Servidor	75
Figura 7.2 - Ligação a origem de dados: Credenciais.....	75
Figura 7.3 - Ligação a origem de dados: Base de Dados	76
Figura 7.4 - Dashboard Aluno: página Geral	79
Figura 7.5 - Painel de ferramentas.....	80
Figura 7.6 - Filtro dos Anos em análise	81
Figura 7.7 - Mapa	81
Figura 7.8- Taxas por [Ano] - Distrito.....	82
Figura 7.9 - Taxas por Ano - [Distrito]	82
Figura 7.10 - Vários indicadores.....	83
Figura 7.11 - Nº de UCs por Curso e Ano Lectivo.....	84
Figura 7.12 - Curso - Mapa	85
Figura 7.13 - Departamento - Mapa.....	85
Figura 7.14 - Departamento - Distrito e Ano Lectivo	86
Figura 7.15 - Taxas por Tipo Avaliação e Distrito.....	87
Figura 7.16 - Médias Notas e Reprovados.....	88
Figura 7.17 - Mapa Top 5 + Docentes	88

ÍNDICE DE TABELAS

Tabela 1.1 - Planeamento (esforço)	4
Tabela 2.1 – kpi	8
Tabela 2.2 - Estados vs kpi	8
Tabela 4.1 - Dados Tipo Avaliação	25
Tabela 4.2 - Dados Estado Civil	28
Tabela 4.3 - Dados Estado Matrícula	28
Tabela 4.4 - Dados Estado Perfil	29
Tabela 4.5 - Dados Sexo	30
Tabela 5.1 - Dimensões	31
Tabela 5.2 - Tabelas Dimensões vs Factos	32
Tabela 5.3 - Dimensões da Estrela Aluno.....	33
Tabela 5.4 - Chaves e fontes de dados da Estrela Aluno.....	34
Tabela 5.5 - Dimensões da Estrela UC.....	35
Tabela 5.6 - Chaves e fontes de dados da Estrela UC.....	36
Tabela 5.7 - Dimensões da Estrela Curso.....	37
Tabela 5.8 - Chaves e fontes de dados da Estrela Curso.....	38
Tabela 6.1 - <i>Derived Column</i>	40
Tabela 6.2 - Mapeamento Estado Civil	49
Tabela 6.3 - Mapeamento Sexo.....	50
Tabela 6.4 - Mapeamento Estado Matrícula	51
Tabela 6.5 - Mapeamento Ano Lectivo	51
Tabela 6.6- Mapeamento Aluno.....	53
Tabela 6.7 - Mapeamento Curso.....	54
Tabela 6.8 - Mapeamento Departamento.....	55
Tabela 6.9 - Tipos de alterações dos atributos da tabela DimUnidadeCurricular.....	56
Tabela 6.10 - Mapeamento Centro Local de Aprendizagem.....	58
Tabela 6.11 - Mapeamento Docente.....	59
Tabela 6.12 - Mapeamento Código Postal	60
Tabela 6.13 - <i>Derived Column</i> : adiciona campos Abandono, Insucesso e Sucesso.....	62

SIGLAS

ER	Entidade Relacionamento
BI	<i>Business Intelligence</i>
CLA	Centro Local de Aprendizagem
DAX	<i>Data Analysis Expressions</i>
DW	<i>Data Warehouse</i>
OLAP	Online Analytical Processing
ETL	<i>Extract, Transform and Load</i>
KPI	<i>Key Performance Indicators</i>
OLTP	<i>Online Transaction Processing</i>
SCD	<i>Slowly Changing Dimension</i>
UAb	Universidade Aberta
SSDT	<i>Sql Server Data Tools</i>
TF	<i>Task Factory</i>
UC	Unidade Curricular

1. INTRODUÇÃO

1.1. INTRODUÇÃO

A Universidade Aberta (UAb) tem como missão formar estudantes, que no seu tempo próprio não puderam encetar ou prosseguir estudos universitários. Por outro lado a Universidade disponibiliza formação a todos aqueles que já tendo uma formação superior querem reconvertê-la ou atualizá-la.

Por norma o público-alvo é maioritariamente adulto, com experiência de vida e normalmente com experiência profissional.

Sendo a formação disponibilizada *on-line* (*e-learning*), os alunos inscritos na Universidade estão distribuídos geograficamente por todo o mundo.

Neste contexto a UAb é uma organização que oferece os seus serviços a “clientes” com características diferentes daqueles que frequentam organizações de ensino mais “convencionais”.

Para garantir uma oferta de qualidade nos seus cursos, existe necessidade de conhecer em detalhe e periodicamente como se caracteriza o desempenho da UAb, e se necessário refletir sobre os resultados para planear estrategicamente os passos futuros.

1.2. TEMA

Com base nos dados registados nos vários subsistemas existentes na UAb pretende-se conhecer melhor o abandono escolar e o sucesso escolar.

Conhecendo melhor as variáveis inerentes ao abandono/sucesso escolar poderão os gestores atuar no sentido de proporcionar aos alunos mais e melhores recursos e paralelamente aferir a necessidade de reestruturação dos cursos, sempre com o objetivo de garantir/manter um serviço de qualidade.

1.3. ENQUADRAMENTO

Para conseguir atuar é necessário dotar os gestores (decisores) de ferramentas de apoio à decisão que lhes permitam facilitar e apurar a tomada de decisões.

A construção de um *Data Warehouse* é a ferramenta que melhor preconiza este objetivo.

Pretende-se, então, com base nos dados dos subsistemas existentes na UAb e por intermédio de processos ETL (*Extract, Transform and Load*) construir um *Data Warehouse* que permitirá a consulta com ferramentas de análise e reporte.

Os subsistemas existentes são:

Moodle:

- Conteúdo das várias Unidades Curriculares que o aluno está matriculado;
- Os professores disponibilizam em cada UC vários fóruns (notícias, ajuda, etc), Plano da Unidade Curricular (PUC), Informação diversa no âmbito do ensino da unidade curricular;
- Os professores disponibilizam vários módulos à medida que a matéria vai avançando. Dentro de cada módulo existe normalmente um fórum específico e atividades formativas dedicadas ao módulo. Tanto nos fóruns como nas atividades formativas os alunos são intervenientes. As atividades formativas podem ter notas, mas estas não contam para a nota final.
- Os professores disponibilizam os enunciados dos e-fólios, e os alunos entregam os seus e-fólios neste subsistema.
- Ficam registadas as notas dos e-fólios, p-fólios e exames (época normal e de recurso).

Neste subsistema existe interação entre professores e alunos, e entre alunos, e é onde na realidade acontece o ensino das Unidades Curriculares.

Portal Académico:

- Dados cadastrais do aluno (nome, morada, contactos, etc);
- Inscrições e matrículas;
- Pautas de inscrição e notas finais;
- Situação Financeira (conta corrente, referências MB, pagamentos, etc).

Sistema de gestão da Comunicação Online (SITCON):

Subsistema para envio de mensagens aos serviços Académicos (esclarecimentos de dúvidas, pedidos de documentos, etc).

Portal UAb:

- Dados sobre os Departamentos;
- Dados sobre os Cursos;
- Dados sobre os Docentes;
- Dados sobre as Unidades Curriculares;
- Dados sobre avaliações das Unidades Curriculares;
- Dados sobre os Centros Locais de Aprendizagem;

1.4. OBJETIVOS

Identificação das condições que podem influenciar o abandono escolar e o sucesso escolar nos cursos disponibilizados pela UAb.

Para tomar decisões existe necessidade de conhecer em detalhe os dados que caracterizam os alunos:

- Idade
- Sexo
- Existência de cursos superiores anteriores
- Estado civil
- Distribuição geográfica

Caracterização das disciplinas:

- Avaliação
- Cursos

Caraterização dos docentes.

- Relação entre disciplina versus avaliação

Caraterização dos Departamentos.

- Relação entre departamento versus sucesso/abandono.

Na construção do *Data Warehouse* irá ser descartado o subsistema sitcom uma vez que os dados existentes não influenciam o estudo do abandono escolar e/ou do sucesso escolar e vão ser trabalhados os dados dos subsistemas moodle, portal académico e portal UAb.

1.5. PLANEAMENTO

Atividade	Tarefa	Data Início	Data Fim	Duração	Concluído
Relatório Inicial	Construção do documento	19-03-2018	20-03-2018	2	Sim
	Revisão do documento	21-03-2018	03-04-2018	14	Sim
	Submissão do documento	04-04-2018	04-04-2018	1	Sim
Modelo de Dados OLTP	Desenho do modelo de dados	02-04-2018	10-04-2018	8	Sim
	Criação do modelo de dados	11-04-2018	22-04-2018	11	Sim
Dados	Criação de dados <i>dummy</i> / Carregamento de Dados	23-04-2018	29-04-2018	7	Não
Modelo de Dados OLAP (<i>Data Warehouse</i>)	Identificação das questões chave que DW deve responder	30-04-2018	02-05-2018	3	Não
	Identificação das tabelas de Factos e Dimensões	03-05-2018	07-05-2018	5	Não
	Criação do modelo de dados	08-05-2018	20-05-2018	12	Não

Relatório Intermédio	Construção do documento	15-04-2018	08-09-2018	24	Não
	Submissão do documento	09-05-2018	09-05-2018	1	Não
ETL	Criação de processos de extração	21-05-2018	28-05-2018	7	Não
	Criação de processo de transformação	28-05-2018	03-06-2018	7	Não
	Criação de processo de carregamento	04-06-2018	10-06-2018	7	Não
Business Intelligence	Identificação dos relatórios/visuais a disponibilizar	11-06-2018	14-06-2018	4	Não
	Criação dos relatórios	15-06-2018	24-06-2018	10	Não
Relatório Final	Construção do documento	04-06-2018	26-06-2018	23	Não
	Submissão do documento	27-06-2018	27-06-2018	1	Não

TABELA 1.1 - PLANEAMENTO (ESFORÇO)

Semana / Atividade	19/3	26/3	02/4	09/4	16/4	23/4	30/4	07/5	14/5	21/5	28/5	04/6	11/6	18/6	25/6
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Relatório Inicial															
Modelo de Dados OLTP															
Dados															
Modelo de Dados OLAP (Data Warehouse)															
Relatório Intermédio															
ETL															
Business Intelligence															
Relatório Final															

Tabela 2 – Planeamento (calendário)

1.6. ORGANIZAÇÃO DO RELATÓRIO

O trabalho encontra-se organizado em oito capítulos.

O presente capítulo pretende contextualizar o trabalho do projeto, dando um enquadramento ao tema proposto.

No segundo capítulo é identificada a abordagem estratégica para a definição do DW assim como os pressupostos para a escolha da infraestrutura onde assenta o trabalho e a plataforma de desenvolvimento.

No terceiro capítulo é feita uma abordagem à modelação lógica e física feita no desenvolvimento da base de dados dos vários subsistemas OLTP.

No quarto capítulo são referidas as técnicas para o carregamento de dados artificiais.

No quinto capítulo é abordado o desenvolvimento do modelo DW, seguindo o modelo em constelação, especificando as opções tomadas.

No sexto capítulo são descritos os passos para a construção dos processos de ETL.

No capítulo sete apresenta-se o final do processo do DW, que é a camada de apresentação para a tomada de decisões, isto é, os relatórios que permitem analisar as questões que se propuseram para este trabalho.

Por fim temos o capítulo oito que aborda os todos os passos feitos até então, e propõe-se a identificar conclusões sobre o âmbito do abandono escolar e do sucesso escolar.

2. ESTRATÉGIA

2.1. INTRODUÇÃO

Para a definição e construção do DW é necessário analisar e ponderar sobre os recursos existentes ao nível das fontes de dados, dos próprios dados, das questões que se querem ver respondidas e das técnicas que se podem aplicar para que o resultado final vá ao encontro das expectativas iniciais.

Este capítulo aborda a estratégia a seguir na definição do DW, e termina com os princípios que determinaram a escolha da plataforma de desenvolvimento, concretamente na definição da infraestrutura técnica utilizada no desenvolvimento deste projeto.

2.2. DATA WAREHOUSE

A estratégia de implementação do DW para análise do abandono escolar e do sucesso escolar terá de passar obrigatoriamente pelos dados existentes nos subsistemas de origem (obviamente), e tentar relacionar com fatores que possam influenciar o desempenho escolar, e que ajudem a caracterizar o abandono escolar e o sucesso escolar.

Dos dados existentes, temos 3 grandes áreas que nos podemos dedicar mais em detalhe: Alunos, Cursos e UCs.

Antes de mais temos que definir as métricas que caracterizam o abandono escolar e o sucesso escolar.

Segundo o documento das estatísticas da licenciatura de informática no âmbito das V Jornadas de Informática:

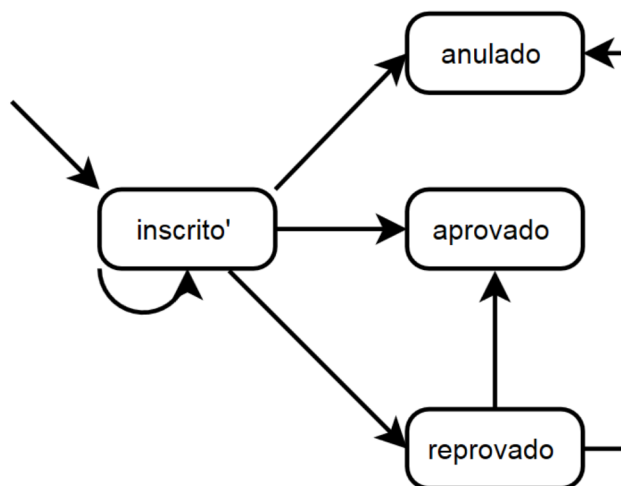


FIGURA 2.1 - Fluxograma: Inscritos E Avaliações

Inscritos = inscritos' (faltas) + anulados + aprovados + reprovados (todos os inscritos na UC incluindo anulados, aprovados e reprovados)

avaliados = aprovados + reprovados (estudantes que compareceram na prova presencial (p-fólio ou exame) com exclusão das faltas, mas incluindo notas iguais a zero)

Aprovados: com nota superior ou igual a 9,5.

Estudantes inscritos, avaliados e aprovados

$$taxa\ abandono\ (dropout) = 1 - \frac{avaliados}{inscritos}$$

$$taxa\ reprova\c{c}\tilde{a}\o\ (failure) = 1 - \frac{aprovados}{avaliados}$$

$$taxa\ sucesso = \frac{aprovados}{inscritos}$$

Na avaliação das Ucs na UAb existem 2 métodos possíveis que os alunos têm como opção: Avaliação Contínua e Exame.

Na Avaliação Contínua os alunos têm de realizar 2 ou 3 e-fólios durante o semestre e um p-fólio no final do semestre. Para permitir realizar o p-fólio, o aluno terá de ter no mínimo 3,5 valores no somatório dos resultados dos e-fólios, e para ser aprovado na UC terá de ter no mínimo 9,5 valores no somatório dos resultados dos e-fólios e p-fólios. O somatório

máximo dos valores dos e-fólios é de 8 valores e do p-fólio 12 valores (num total de 20 valores). Caso reprove no 1º e-fólio terá uma segunda oportunidade de ser aprovado na UC com a possibilidade de realizar um e-fólio de recurso.

Na modalidade de Exame o aluno realizará um exame com um máximo de 20 valores, e caso reprove tem uma segunda hipótese de realizar o exame de recurso.

Para a definição do abandono e o sucesso escolar terão de ser analisados os estados possíveis da matrícula da UC e das respetivas notas, sempre que existirem.

Os estados possíveis para uma UC foram definidos como:

A – Aprovado

I – Inscrito

N – Anulado

V - Avaliado

Sintetiza-se no quadro em baixo a definição do abandono e sucesso escolar com base nas variáveis estado, tipo de avaliação e notas:

A = Abandono I = Insucesso S = Sucesso

Estado	Tipo Avaliação	E-fólios	P-fólio	P-fólio recurso	Exame	Exame recurso	Final	A	I	S
A	Contínua	≥ 3.5	≥ 5.5				≥ 9.5	0	0	1
A	Contínua	≥ 3.5	*	≥ 5.5			≥ 9.5	0	0	1
A	Contínua					≥ 9.5	≥ 9.5	0	0	1
A	Exame				≥ 9.5		≥ 9.5	0	0	1
A	Exame				< 9.5	≥ 9.5	≥ 9.5	0	0	1
A	Exame					≥ 9.5	≥ 9.5	0	0	1
V	Contínua	≥ 3.5	< 5.5					0	1	0
V	Contínua	≥ 3.5	< 5.5	< 5.5				0	1	0
V	Contínua					< 9.5		0	1	0
V	Exame				< 9.5	< 9.5		0	1	0
V	Exame					< 9.5		0	1	0
V	Exame				< 9.5			0	1	0
I	Contínua	< 3.5						1	0	0
I	Contínua	≥ 3.5						1	0	0

I	Continua							1	0	0
I	Exame							1	0	0
N	Continua							1	0	0
N	Exame							1	0	0

TABELA 2.1 – KPI

* O somatório das notas dos e-fólios e do p-fólio é inferior a 9.5 valores.

Analisando a tabela anterior podemos desde já identificar uma correspondência entre os estados e os indicadores chave:

Estado	Abandono	Insucesso	Sucesso
A - Aprovado	0	0	1
V - Avaliado	0	1	0
N – Anulado	1	0	0
I - Inscrito	1	0	0

TABELA 2.2 - Estados vs kpi

Podemos agora transformar as fórmulas da fig. 2.1 com os novos dados:

$$taxa\ abandono\ (dropout) = 1 - \frac{Insucesso + Sucesso}{Abandono + Insucesso + Sucesso = Todos}$$

$$taxa\ reprova\c{a}\o\ (failure) = 1 - \frac{Sucesso}{Insucesso + Sucesso}$$

$$taxa\ sucesso = \frac{Sucesso}{Abandono + Insucesso + Sucesso = Todos}$$

Que nos permitirão parametrizar os kpi no modelo do DW e são as métricas de referência nas consultas a efetuar no contexto da exploração do DW.

Pegando novamente nos dados de origem, e identificando as questões que o DW deverá responder, pode-se fazer desde já o exercício da divisão das várias tabelas de factos que vão constituir a constelação do DW.

- Alunos:

O primeiro passo é a classificação dos alunos com abandono escolar e o sucesso escolar com base na entidade [AvaliacaoUnidadeCurricular], seguindo para a construção de uma tabela de factos para caracterizar os Alunos que se incluem neste grupo, e para podermos responder às questões:

- Distribuição geográfica (morada);
- Sexo;
- Idade;
- Estado Civil;
- Existência de curso anterior;
- Proximidade de CLA;
- Média;

Outra tabela de factos disponível para análise será a caracterização da UC individualmente.

As questões a colocar são:

- Identificar as UCs;
- Notas associadas;
- Distribuição geográfica (fazendo a ligação com o aluno);

Por fim, podemos caracterizar o curso em si relativamente ao abandono e sucesso escolar para a construção de outra tabela de factos:

- Identificação dos cursos;
- Média das notas
- Distribuição geográfica (fazendo a ligação com o aluno);

Claramente a variedade dos dados disponíveis dos alunos superam os restantes, e consequentemente o número de questões que se podem colocar e que o DW pode responder são influenciadas por este fator.

2.3. FERRAMENTAS

Na solução para este trabalho houve preocupação na escolha de ferramentas que fossem gratuitas e ao mesmo tempo flexíveis e robustas para responder aos objetivos propostos.

A análise das ferramentas disponíveis no mercado foi influenciada pelo conhecimento existente sobre algumas delas, o funcionamento e paralelamente conhecimento da performance das mesmas.

A construção da base de dados OLTP foi feita de raiz, criada num único motor de base de dados, com várias base de dados, uma por subsistema, pelo que não existem, como é normal em sistemas operacionais reais, uma dispersão de fontes de dados dos vários subsistemas, onde se podem incluir até "base de dados" mais rudimentares tipo folhas de cálculo...

Para o ambiente de desenvolvimento a escolha das ferramentas recaiu em ferramentas *Microsoft*, pelo princípio de gratuidade já referida atrás, mas tão importante como a gratuidade foi a potencialidade, usabilidade e flexibilidade das ferramentas nas várias vertentes da construção de um DW.

Outro aspeto a realçar será o fato de as ferramentas pertencerem a um mesmo fornecedor, onde a integração entre as várias interfaces é por princípio facilitada.

Todo o desenvolvimento assenta no sistema operativo *Windows 8.1 Enterprise (64 bits)*, sendo que para a construção das bases de dados transacionais e DW foi escolhido motor de base de dados *Microsoft SQL Server Express (64-bit)* versão 14.0.1000.169.

Para a construção dos processos de ETL foi utilizado o *Microsoft Visual Studio Tools for Applications 2015* (versão 14.0.23107.0 D14REL) com o produto *SQL Server Integration Services (Microsoft SQL Server Integration Services Designer* versão 14.0.1000.169).

Para trabalhar em conjunto com esta ferramenta e no sentido de simplificar a gestão dos carregamentos das tabelas de factos foi também utilizada o pacote "*TaskFactory*" mais concretamente o componente "*Fact Table Destination*" da empresa *Pragmatic Works* para trabalhar em conjunto com o *Sql Server Data Tools*. O *TaskFactory* é gratuito para trabalhar no SSDT, no entanto para trabalhar em modo comando é necessário adquirir uma licença.

No final, a ferramenta de exploração de dados escolhida é o *Microsoft Power BI Desktop* Versão: 2.57.5068.721 64-bit (abril de 2018).

Em baixo o diagrama de integração das ferramentas na construção do DW:

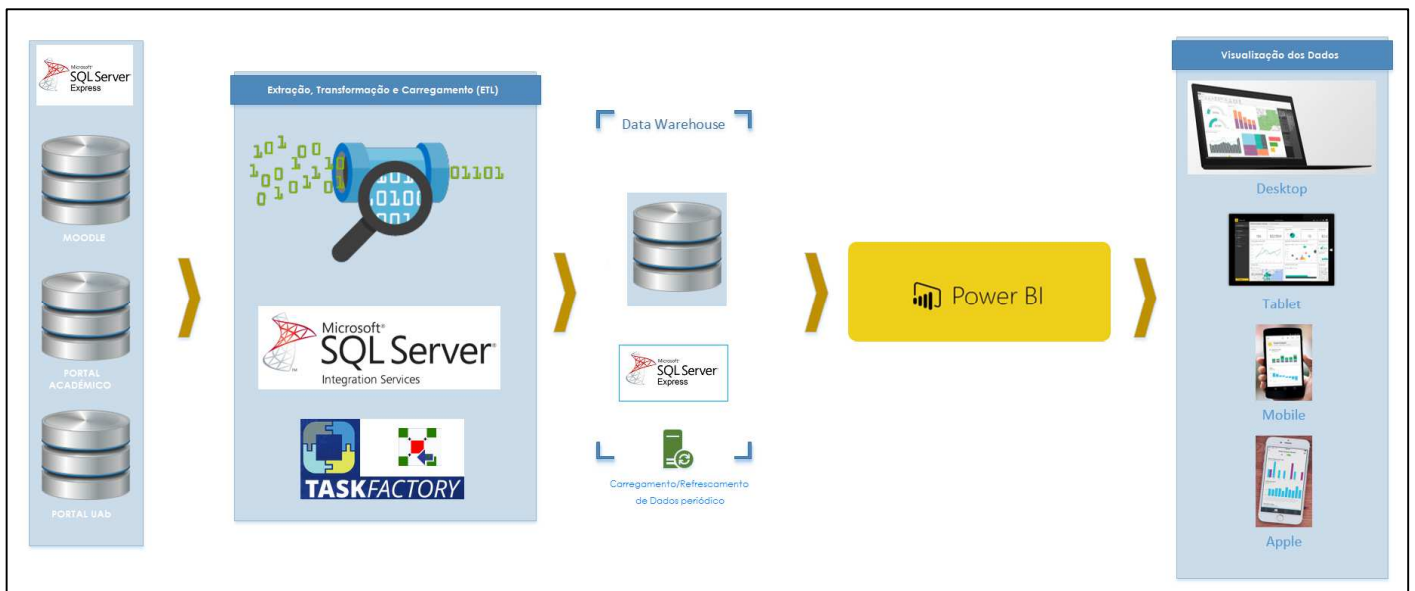


FIGURA 2.2 - Infraestrutura OLTP, OLAP, BI e Clientes

3. MODELO OLTP

3.1. INTRODUÇÃO

Para a construção do modelo transacional foram utilizadas técnicas de construção de modelos lógicos baseados no modelo entidade-relação (ER), mas a base do conhecimento que permitiu construir o “negócio” foi em grande medida fruto da experiência com a UAb durante a realização das restantes unidades curriculares da licenciatura de Informática (atual licenciatura em Engenharia Informática).

Claramente o modelo construído não reflete o modelo real, tanto para mais que não se conhece o modelo verdadeiro (ou modelos) da UAb, no entanto serve para o objetivo proposto da construção de um DW a partir de uma base de dados OLTP.

A figura a seguir representa o modelo global, seguido dos modelos em detalhe.

3.2. DIAGRAMA GLOBAL

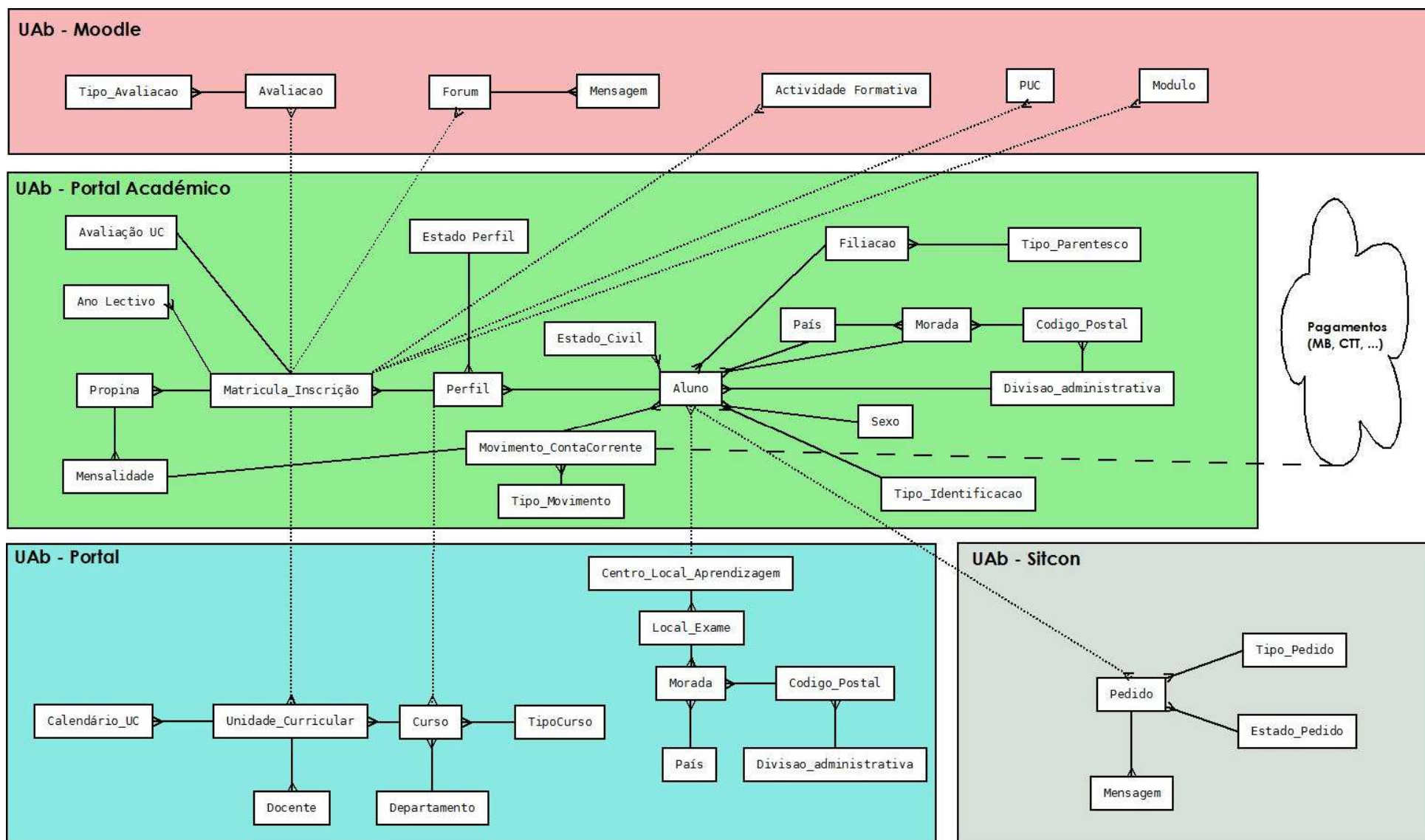


Figura 3.1 - Modelo Global

3.3. MODELO MOODLE

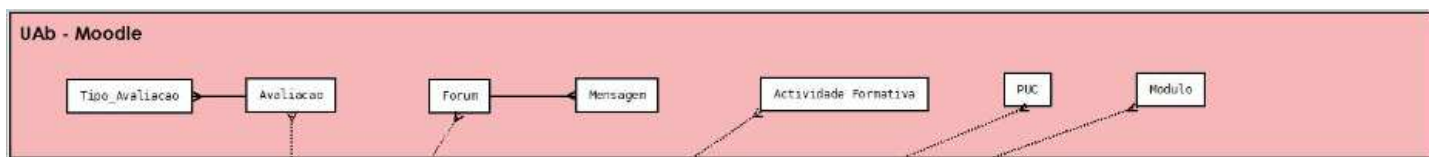


Figura 3.2 - Modelo lógico do subsistema Moodle

É neste subsistema onde existem as interações entre professor-aluno e aluno-aluno, digamos que é a “sala de aula”, e onde são registadas, entre outros, as várias avaliações do aluno ao longo do semestre, atividades formativas, disponibilização de materiais da unidade curricular, incluindo as provas de avaliação e-fólios, etc.

Da parte do objetivo traçado para o DW o importante neste subsistema será a componente das avaliações/notas.

A entidade de referência deste subsistema é a entidade “Matricula/Inscrição”, pertencente ao subsistema “Portal Académico”, sendo o subsistema Moodle consumidor dos dados dessa entidade/subsistema.

Com base na entidade “Matricula/Inscrição” temos acesso à relação Aluno e Unidade Curricular e a todos os atributos respetivos de forma a caracterizar as questões que o DW deve responder.

Transformando o modelo lógico no modelo físico ficamos com o seguinte modelo:

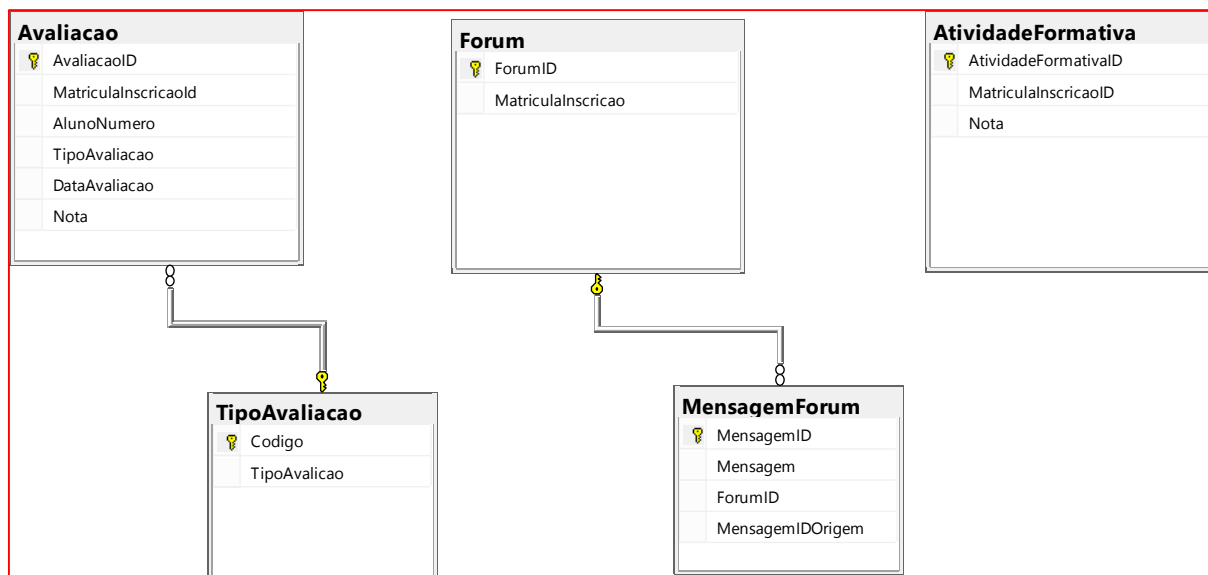


FIGURA 3.3 - Modelo Físico Do Subsistema Moodle

- **Avaliação:** registo das notas de avaliação dos e-fólios e p-fólios

```
[AvaliacaoID] [int] IDENTITY(1,1) NOT NULL
[MatriculaInscricaoId] [int] NOT NULL
[AlunoNumero] [int] NOT NULL
[TipoAvaliacao] [nvarchar](3) NOT NULL
```

[DataAvaliacao] [date] NOT NULL
[Nota] [float] NOT NULL

- **Tipo_Avaliacao:** tabela de parametrização com tipo de avaliação

[Codigo] [nvarchar](3) NOT NULL
[TipoAvaliacao] [nvarchar](32) NOT NULL

- **Forum:** registo do identificador de fórum das UCs

[ForumID] [int] IDENTITY(1,1) NOT NULL
[MatriculaInscricao] [int] NULL

- **Mensagem_Forum:** registo das mensagens dos fóruns

[MensagemID] [int] IDENTITY(1,1) NOT NULL
[Mensagem] [nvarchar](max) NOT NULL
[ForumID] [int] NOT NULL
[MensagemIDOrigem] [int] NULL

- **Actividade_Formativa:** registo das atividades formativas das UCs

[ActividadeFormativaID] [int] IDENTITY(1,1) NOT NULL
[MatriculaInscricaoID] [int] NOT NULL
[Nota] [float] NULL

IDENTITY(1,1): significa que o campo é gerado automaticamente com chaves de valor de tipo inteiro, com início no valor 1 (primeiro 1), e incrementado de um a um (segundo 1).

3.4. MODELO PORTAL ACADÉMICO

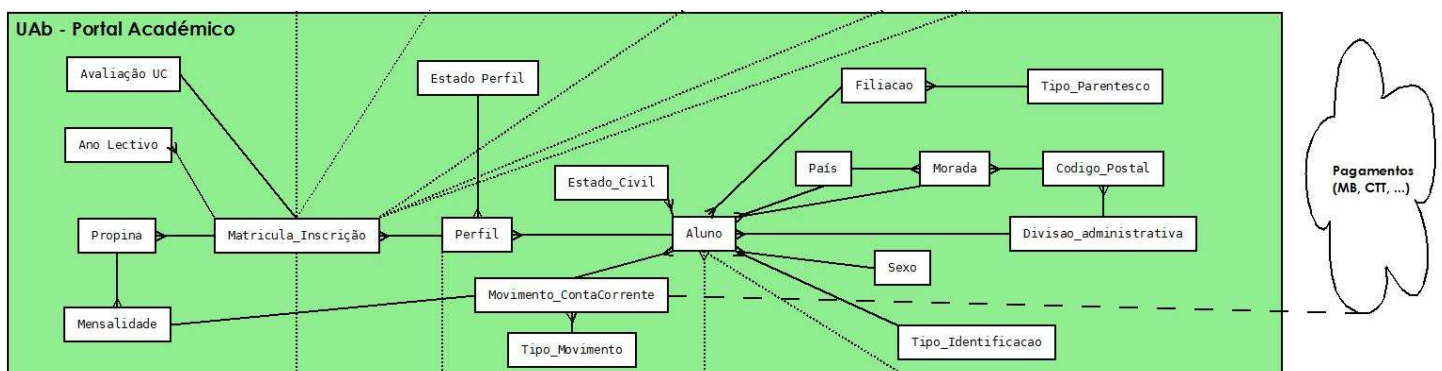


Figura 3.4 - Modelo lógico do subsistema Portal Académico

Subsistema onde ficam registados os dados:

Aluno: Nome, data de nascimento, morada, naturalidade, etc.

Unidade Curricular: inscrição/matricula nas disciplinas, notas finais, data de avaliação, etc.

[NaturalidadeFreguesia] [int] NULL
 [DataValidadeVacinaAntitetânica] [date] NOT NULL
 [MaeNome] [nvarchar](256) NULL
 [PaiNome] [nvarchar](256) NULL
 [NumeroMorada] [int] NOT NULL
 [Deslocado] [nvarchar](1) NULL
 [Telefone1] [nvarchar](16) NULL
 [Telefone2] [nvarchar](16) NULL
 [Telemovel] [nvarchar](16) NULL
 [Email] [nvarchar](64) NULL
 [PaisFiscal] [nvarchar](3) NULL
 [NumeroIdentificacaoFiscal] [int] NOT NULL
 [NIB] [nvarchar](21) NULL
 [IBAN] [nvarchar](25) NULL
 [CursosAnteriores] [int] NOT NULL
 [Activo] [nvarchar](1) NOT NULL
 [TrabalhadorEstudante] [bit] NOT NULL

- **AnoLectivo:** registo dos anos lectivos

[AnoLectivoID] [int] IDENTITY(1,1) NOT NULL
 [AnoLectivo] [nvarchar](32) NOT NULL
 [DataInicio] [date] NOT NULL
 [DataFim] [date] NOT NULL

- **AvaliacaoUnidadeCurricular:** tabela agregadora da informação sobre UC vs matricula

[MatriculaInscricaoId] [int] NOT NULL
 [CursoID] [int] NOT NULL
 [AnoLectivo] [int] NOT NULL
 [UnidadeCurricularCodigo] [int] NOT NULL
 [AlunoNumero] [int] NOT NULL
 [Estado] [nvarchar](1) NOT NULL
 [TipoAvaliacao] [nvarchar](1) NOT NULL
 [NotaEFolios] [decimal](18, 5) NULL
 [NotaPFolio] [decimal](18, 5) NULL
 [NotaPFolioRecurso] [decimal](18, 5) NULL
 [NotaExame] [decimal](18, 5) NULL
 [NotaExameRecurso] [decimal](18, 5) NULL
 [NotaFinal] [decimal](18, 5) NULL
 [DataEstado] [datetime2](7) NOT NULL

- **CodigoPostal:** registo dos códigos postais

[CodigoPostal] [int] NOT NULL
 [Localidade] [nvarchar](64) NOT NULL
 [Distrito] [int] NOT NULL
 [Concelho] [int] NOT NULL
 [Freguesia] [int] NOT NULL

- **DivisaoAdministrativa:** registo dos códigos das atividades administrativas

[Nivel_1] [int] NOT NULL,
 [Nivel_2] [int] NOT NULL,
 [Nivel_3] [int] NOT NULL,
 [Designacao] [nvarchar](64) NOT NULL,

- **EstadoCivil:** tabela de parametrização com tipo de estados civis

[EstadoCivilCodigo] [nvarchar](3) NOT NULL
 [EstadoCivilDescricao] [nvarchar](64) NOT NULL

- **EstadoMatricula:** tabela de parametrização com tipos de estados da matrícula

```
[Codigo] [nvarchar](1) NOT NULL  
[EstadoMatricula] [nvarchar](64) NOT NULL
```

- **EstadoPerfil:** tabela de parametrização com tipo de estados da tabela Perfil

```
[Codigo] [nvarchar](1) NOT NULL  
[Descricao] [nvarchar](64) NOT NULL
```

- **Filiacao:** tabela com os parentes do Aluno

```
[AlunoNumero] [int] NOT NULL  
[TipoParentescoID] [tinyint] NOT NULL  
[Naturalidade] [char](2) NULL  
[Nacionalidade] [char](3) NULL  
[FormacaoAcademica] [tinyint] NULL  
[Profissao] [char](4) NULL  
[SituacaoEmprego] [tinyint] NULL
```

- **HabilitacoesLiterarias:** tabela com os tipos de habilitações literárias

```
[HabilitacoesAcademicasID] [int] NOT NULL  
[Designacao] [nvarchar](64) NOT NULL
```

- **MatriculaInscricao:** tabela com os dados sobre as Matrículas/Inscrições

```
[MatriculaInscricaoId] [int] IDENTITY(1,1) NOT NULL  
[UnidadeCurricularCodigo] [int] NOT NULL  
[PerfilID] [int] NOT NULL  
[DataInscricao] [date] NOT NULL  
[Estado] [nvarchar](1) NOT NULL  
[DataEstado] [datetime2](7) NOT NULL  
[NotaFinal] [int] NOT NULL
```

- **Mensalidade:** tabela com os dados das mensalidades pagas ou por pagar

```
[MensalidadeID] [int] IDENTITY(1,1) NOT NULL  
[PropinaID] [int] NOT NULL  
[Valor] [float] NOT NULL  
[DataVencimento] [date] NOT NULL  
[Facturado] [nvarchar](2) NOT NULL  
[Pago] [nvarchar](2) NOT NULL  
[Acrescimos] [float] NULL  
[Descontos] [float] NULL  
[Iva] [float] NULL  
[Anulado] [nvarchar](2) NOT NULL  
[ReferenciaMB] [nvarchar](9) NULL  
[DataPagamento] [date] NULL
```

- **Morada:** tabela de moradas

```
[NumeroMorada] [int] NOT NULL
[PaisCodigo] [nvarchar](3) NOT NULL
[Artéria] [nvarchar](100) NOT NULL
[Localidade] [nvarchar](40) NOT NULL
[CodigoPostal] [int] NOT NULL
[Regiao] [nvarchar](40) NULL
[Endereco_linha_1] [nvarchar](50) NULL
[Endereco_linha_2] [nvarchar](50) NULL
[Endereco_linha_3] [nvarchar](50) NULL
```

- **MovimentoContaCorrente:** tabela com os dados sobre movimentos financeiros

```
[MovimentoID] [int] IDENTITY(1,1) NOT NULL
[AlunoNumero] [int] NOT NULL
[DataMovimento] [datetime2](7) NOT NULL
[TipoMovimento] [nvarchar](3) NOT NULL
[MovimentoIDOrigem] [int] NOT NULL
[Valor] [float] NOT NULL
[Origem] [nvarchar](3) NOT NULL
[Descricao] [nvarchar](128) NOT NULL
[DataValor] [datetime2](7) NOT NULL
```

- **OutraInformacao:** tabela com dados complementares dos alunos

```
[AlunoNumero] [int] NOT NULL
[Computador] [bit] NULL
[Internet] [bit] NULL
[NISS] [bigint] NULL
[NCGA] [bigint] NULL
[Regime] [tinyint] NULL
[InscritoAnoAnterior] [bit] NULL
[CursoInscritoAnoAnterior] [tinyint] NULL
[HabilitacoesAcademicasID] [int] NULL
```

- **Pais:** tabela com os países

```
[PaisCodigo] [nvarchar](3) NOT NULL
[PaisDesignacao] [nvarchar](64) NOT NULL
[PaisActivo] [nchar](1) NOT NULL
```

- **Perfil:** tabela com os perfis dos alunos (tipicamente quando têm mais do um curso inscrito)

```
[PerfilID] [int] IDENTITY(1,1) NOT NULL
[AlunoNumero] [int] NOT NULL
[CursoID] [int] NOT NULL
[DataCriacao] [date] NOT NULL
[Estado] [nvarchar](1) NOT NULL
[DataEstado] [datetime2](7) NOT NULL
```

- **Propina:** tabela com os dados sobre as propinas

```
[PropinaID] [int] IDENTITY(1,1) NOT NULL
```

```

[Descricao] [nvarchar](132) NOT NULL
[MatriculaInscricaoId] [int] NOT NULL
[DataPropina] [date] NOT NULL
[Desconto] [float] NULL
[NumeroMensalidades] [int] NOT NULL
[Estado] [nvarchar](3) NOT NULL
[DataEstado] [datetime2](7) NOT NULL

```

- **Sexo** tabela de parametrização do sexo

```

[SexoCodigo] [nvarchar](1) NOT NULL
[SexoDescricao] [nvarchar](32) NOT NULL

```

- **Tipoidentificacao** tabela de parametrização com tipo de identificação (BI, Cartão de Cidadão, ...)

```

[TipoIdentificacaoCodigo] [nvarchar](3) NOT NULL
[TipoIdentificacaoDesignacao] [nvarchar](64) NOT NULL

```

- **TipoMovimento**: tabela de parametrização com tipo de movimento financeiro

```

[Codigo] [nvarchar](3) NOT NULL
[Descricao] [nvarchar](32) NOT NULL

```

- **TipoMovimento**: tabela de parametrização com tipo de movimento financeiro

```

[ID] [tinyint] NOT NULL
[Nome] [varchar](20) NOT NULL
[Nome2] [varchar](20) NULL

```

3.5. MODELO PORTAL UAB

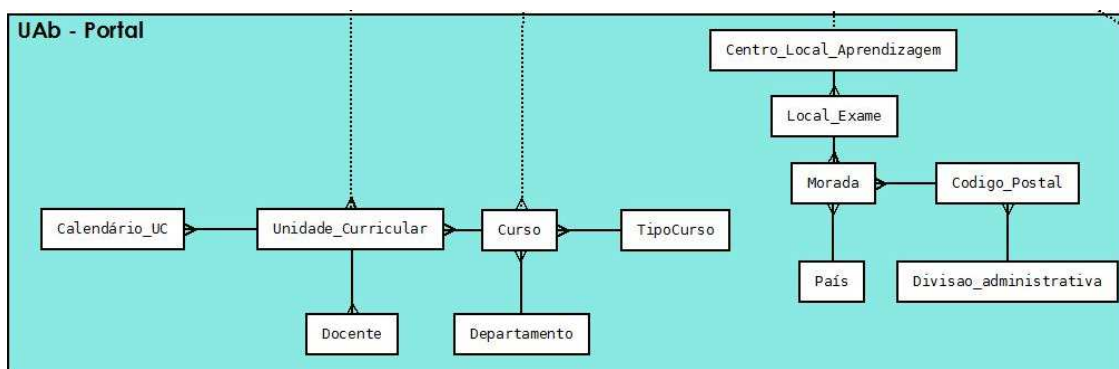


Figura 3.6 - Modelo lógico do subsistema Portal UAb

Subsistema que tem relação direta com a página oficial da UAb. Os dados presentes neste modelo foram recolhidos manualmente e diretamente da página oficial da UAb.

Os dados recolhidos dizem respeito aos docentes, cursos, departamentos, CLAs, UCs e as relações existentes entre as várias entidades.

O modelo físico que espelha o subsistema Portal UAb:

- **Curso:** tabela com os cursos da UAb (1º ciclo)

```
[CursoID] [int] IDENTITY(1,1) NOT NULL  
[Nome] [nvarchar](128) NOT NULL  
[TipoCurso] [int] NOT NULL
```

- **Departamento:** tabela com departamentos da UAb

```
[DepartamentoCodigo] [nvarchar](4) NOT NULL  
[Departamento] [nvarchar](64) NOT NULL
```

- **DivisaoAdministrativa:** tabela com os códigos das divisões administrativas

```
[Nivel_1] [int] NOT NULL  
[Nivel_2] [int] NOT NULL  
[Nivel_3] [int] NOT NULL  
[Designacao] [nvarchar](64) NOT NULL
```

- **Docente:** tabela com os docentes

```
[DocenteNumero] [int] IDENTITY(1,1) NOT NULL  
[Nome] [nvarchar](128) NOT NULL
```

- **DocenteUnidadeCurricular:** tabela com os docentes que ensinam as UCs

```
[DocenteNumero] [int] NOT NULL  
[UnidadeCurricularCodigo] [int] NOT NULL  
[Activo] [nvarchar](1) NOT NULL
```

- **Filiacao:** tabela com a identificação dos pais dos alunos

```
[AlunoNumero] [int] NOT NULL  
[TipoParentescoID] [tinyint] NOT NULL  
[Naturalidade] [char](2) NULL  
[Nacionalidade] [char](3) NULL  
[FormacaoAcademica] [tinyint] NULL  
[Profissão] [char](4) NULL  
[SituacaoEmprego] [tinyint] NULL
```

- **LocalExame:** tabela com os Locais de Exame das CLAs

```
[CentroLocalAprendizagemId] [int] NOT NULL  
[NumeroMorada] [int] NOT NULL  
[Local] [nvarchar](128) NOT NULL  
[Observacoes] [nvarchar](128) NULL
```

- **Morada:** tabela de moradas

```
[NumeroMorada] [int] NOT NULL  
[PaisCodigo] [nvarchar](3) NOT NULL  
[Artéria] [nvarchar](100) NOT NULL
```

```
[Localidade] [nvarchar](40) NOT NULL
[CodigoPostal] [int] NOT NULL
[Regiao] [nvarchar](40) NULL
[Endereco_linha_1] [nvarchar](50) NULL
[Endereco_linha_2] [nvarchar](50) NULL
[Endereco_linha_3] [nvarchar](50) NULL
```

- **Pais:** tabela de países

```
[PaisCodigo] [nvarchar](3) NOT NULL
[PaisDesignacao] [nvarchar](64) NOT NULL
[PaisActivo] [nchar](1) NOT NULL
```

RelDepartamentoCurso: tabela de relação entre as UCs e os departamentos

```
[CursoID] [int] NOT NULL
[DepartamentoCodigo] [nvarchar](4) NOT NULL
```

- **TipoParentesco:** tabela com os tipos de Parentesco

```
[ID] [tinyint] NOT NULL
[Nome] [varchar](20) NOT NULL
[Nome2] [varchar](20) NULL
```

- **TipoCurso:** tabela com os tipos de Curso

```
[ID] [int] NOT NULL
[TipoCurso] [nvarchar](32) NOT NULL
```

- **UnidadeCurricular:** tabela com as UCs

```
[UnidadeCurricularCodigo] [int] NOT NULL
[UnidadeCurricularDescricao] [nvarchar](128) NOT NULL
[ECTS] [int] NOT NULL
```

- **UnidadeCurricularCurso:** tabela com os cursos e respetivas UCs

```
[CursoID] [int] NOT NULL
[UnidadeCurricularCodigo] [int] NOT NULL
[Ano] [int] NOT NULL
[Semestre] [int] NOT NULL
[EstruturaCurricular] [nvarchar](128) NOT NULL
[ECTS] [int] NOT NULL
[AnoCurricular] [nvarchar](9) NOT NULL
[Obrigatorio] [nvarchar](1) NOT NULL
```

3.6. MODELO SITCON

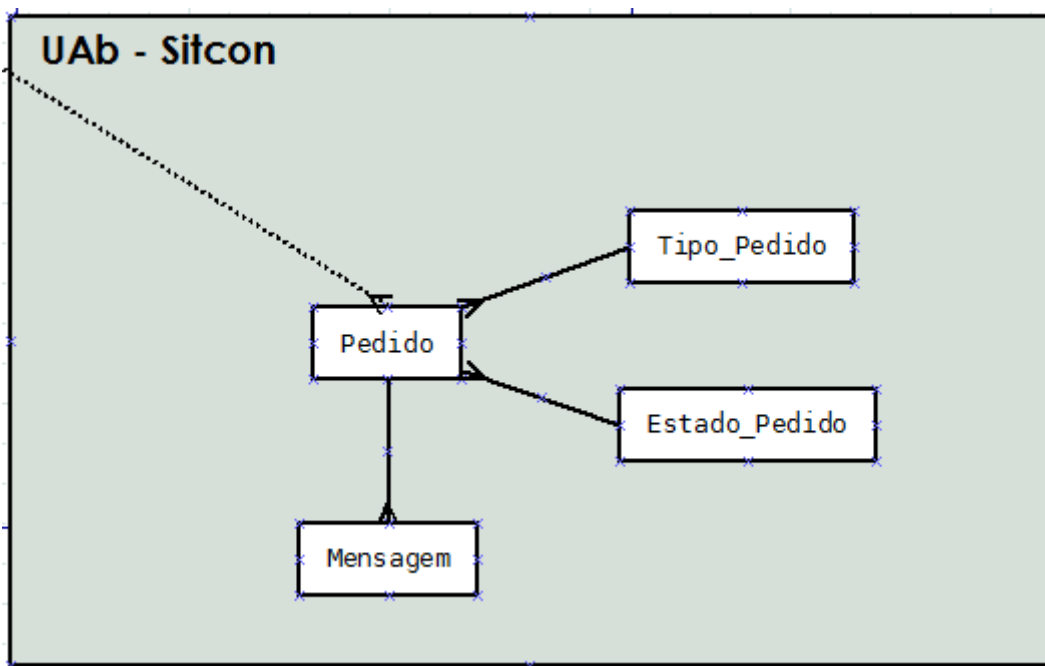


Figura 3.8 - Modelo lógico do subsistema SITCON

Neste subsistema são registados os pedidos dos alunos para os serviços da UAb. Os pedidos podem ser dos mais variados, desde dúvidas, a pedidos de documentos, pedidos de pagamentos de propinas de uma só vez com direito a desconto, etc.

Da análise inicial e dos objetivos propostos, verifica-se que este subsistema não está incluído no âmbito do projeto, pelo que não se vão popular as tabelas com dados, ficando apenas criado o modelo lógico e físico.

A entidade de referência deste subsistema é a entidade "Aluno" do subsistema Portal Académico.

O modelo físico que espelha o subsistema SITCON:

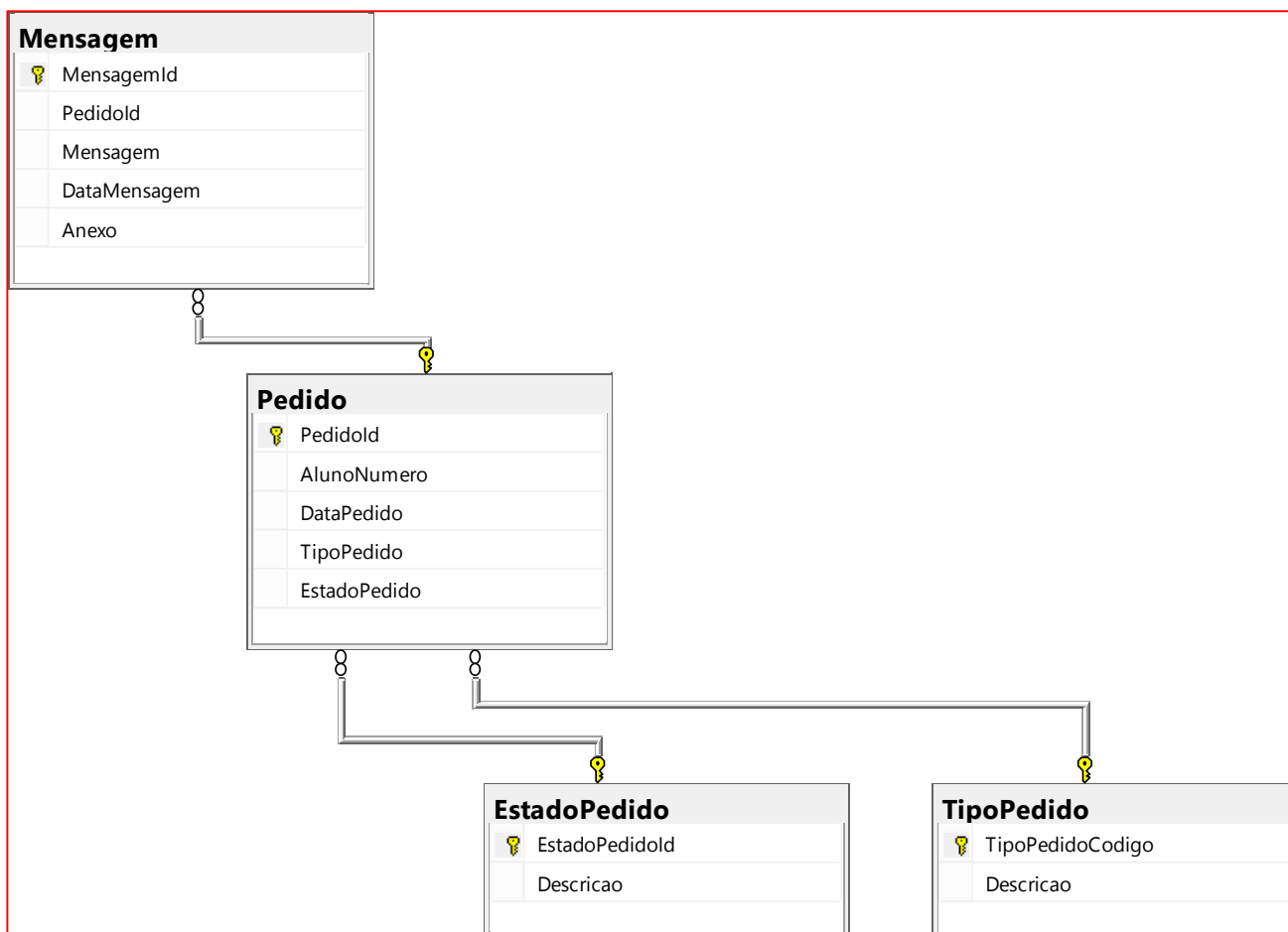


Figura 3.9 - Modelo físico do subsistema SITCON

- **EstadoPedido:** tabela de parametrização com estados de pedido

```
[EstadoPedidoId] [nvarchar](3) NOT NULL
[Descricao] [nvarchar](50) NOT NULL
```

- **Mensagem:** tabela com as mensagens dos pedidos

```
[MensagemId] [int] NOT NULL
[PedidoId] [int] NOT NULL
[Mensagem] [nvarchar](max) NOT NULL
[DataMensagem] [datetime2](7) NOT NULL
[Anexo] [image] NULL
```

- **Pedido:** tabela com os pedidos

```
[PedidoId] [int] NOT NULL
[AlunoNumero] [int] NOT NULL
[DataPedido] [datetime2](7) NOT NULL
[TipoPedido] [nvarchar](3) NOT NULL
[EstadoPedido] [nvarchar](3) NOT NULL
```

- **TipoPedido:** tabela de parametrização com tipos de pedido

[TipoPedidoCodigo] [nvarchar](3) NOT NULL
[Descricao] [nvarchar](50) NOT NULL

4. DADOS (CARREGAMENTO)

4.1. INTRODUÇÃO

Esta atividade revelou-se de uma complexidade maior do que a esperada inicialmente.

Na execução desta atividade, e com o objetivo da criação dos dados artificiais aleatórios, recorreu-se à função rand() e newid() do motor de base de dados SQL SERVER aliada à influência forçada por intermédio de introdução de probabilidades.

O objetivo final foi a criação de dados artificiais aleatórios, mas que de alguma forma representassem, no possível, a realidade. Por exemplo na criação das moradas dos alunos, a grande parte pertencem a Portugal, e a grande minoria são residentes no estrangeiro.

Relativamente aos dados sensíveis, nomeadamente os nomes e moradas dos alunos foram preenchidos com '**** ... aplicada máscara de dados *****', porque em sistemas de DW não é normal o estudo nominal dos factos, como no caso em concreto deste projeto.

Para os dados que se identificaram como irrelevantes para as questões que o DW deste projeto deverá responder, não foram alvo de carregamento, pois não iriam acrescentar valor.

Neste capítulo são apresentados os passos seguidos para a concretização da criação dos dados artificiais, o respetivo código encontra-se no anexo I.

4.2. MODELO MOODLE

Entidade **TipoAvaliacao**:

Foram carregados manualmente com os dados:

Codigo	TipoAvaliacao
EF	E-FÓLIO
EX	EXAME
EXR	EXAME RECURSO
PF	P-FÓLIO
PFR	P-FÓLIO RECURSO

TABELA 4.1 - Dados Tipo Avaliação

Entidade **Avaliacao**:

Foi criado um procedimento que percorre a tabela [MatriculaInscricao] do subsistema [UAb-PortalAcademico], e que cria os registos aleatoriamente mas com a seguinte influência ao nível das probabilidades forçadas:

Avaliação Contínua: 70% / Exame: 30%

Recurso – Sim: 70% / Não: 30%

Caso não exista recurso:

- Exame: a nota será de mesmo valor que a nota final

- Avaliação Contínua: a soma das notas dos e-fólios e do p-fólio será igual à nota final, respeitando a nota mínima de 3,5 valores na soma do e-fólios e 5,5 valores no p-fólio.

No caso de reprovar a nota final, sendo zero valores, tem como mínimo da soma dos e-fólios 3,5 (garante presença no p-fólio), mas a nota final é inferior a 9,5 valores.

No caso de existir recurso, o princípio é o mesmo, mas existe um exame ou um p-fólio prévio com valores que definem a reprovação: exame ou soma de e-fólios mais p-fólio inferior a 9,5 valores (dependendo da modalidade)

As restantes entidades não foram alvo de criação de dados artificiais:

- Forum
- MensagemForum
- AtividadeFormativa

4.3. MODELO PORTAL UAB

Entidade **CalendarioUnidadeCurricular:**

Entidade **CentroLocalAprendizagem:**

Entidade **Curso:**

Entidade **Departamento:**

Entidade **CentroLocalAprendizagem:**

Entidade **Docente:**

Entidade **DocenteUnidadeCurricular:**

Entidade **LocalExame:**

Entidade **Morada (CLA e Local Exame):**

Entidade **RelDepartamentoCurso:**

Entidade **TipoCurso:**

Entidade **UnidadeCurricular:**

Entidade **UnidadeCurricularCurso:**

Carregada com dados extraídos do site UAb.

Entidade **CodigoPostal:**

Carregada com dados a partir de CD dos CTT.

Entidade **DivisaoAdministrativa:**

Carregada com dados a partir do site www.ine.pt.

Entidade **Pais**:
www.ine.pt

4.4. MODELO PORTAL ACADÉMICO

Entidade **Aluno**:

Foi criado um procedimento que cria o número de registos correspondente ao parâmetro de entrada (que está definido como 10.000).

Com base, novamente, nas funções aleatórias e probabilidades (quando se aplica) temos os atributos:

[AlunoNumero] - tem início no número 1200000 incrementado de 1 a 1
[Nome] - '**** Nome aplicada máscara de dados *****'
[DataNascimento] - datas compreendidas entre 1955/01/01 e 1998/01/01
[Sexo] - Feminino: 65% / Masculino: 35%
[EstadoCivil] – Casado: 40% / Separado: 10% / Solteiro: 30% / União de facto: 15% / Viúvo: 5%
[Tipoidentificacao] - CARTÃO DE CIDADÃO: 75% / BI: 25%
[Numeroidentificacao] – número aleatório com 9 dígitos
[Nacionalidade] – 'PT'
[DataValidadeVacinaAntitetânica] – data de sistema + 1 ano
[MaeNome] - '**** MaeNome aplicada máscara de dados *****'
[PaiNome] - '**** PaiNome aplicada máscara de dados *****'
[NumeroMorada] – número incremental
[PaisMorada] – aleatório e de acordo com tabela de probabilidades construída a partir de informações extraídas no relatório de atividades da UAb de 2016
[Deslocado] – Não: 10% / Sim: 90%
[Telefone1] – número aleatório com 9 dígitos com 1º dígito = 2
[Telefone2] – número aleatório com 9 dígitos com 1º dígito = 2
[Telemovel] - número aleatório com 9 dígitos com 1º dígito = 9
[Email] - 'dummy@dummy.pt'
[NumeroidentificacaoFiscal] - número aleatório com 9 dígitos com 1º dígito = 1
[CursosAnteriores] – 2: 4% / 1: 11% / 0: 85%
[Activo] – Não: 5% / Sim: 95%
[TrabalhadorEstudante] – Não: 25% / Sim: 75%

Entidade **AnoLectivo**:

Foram carregados manualmente com os dados de 2000/2001 até 2018/2019.

Entidade **AvaliacaoUnidadeCurricular**:

Construído procedimento que percorre todos os registos da entidade [MatriculaInscricao] do subsistema [UAb-PortalAcademico], e que cria os registos agregando os dados das notas e estados.

[MatriculaInscricaoId] – [PortalAcademico].[MatriculaInscricao].[MatriculaInscricaoId]
 [CursoId] - [PortalAcademico].[Perfil].[CursoId]
 [AnoLectivo] - [PortalAcademico].[MatriculaInscricao].[AnoLectivo]
 [UnidadeCurricularCodigo] –
 [PortalAcademico].[MatriculaInscricao].[UnidadeCurricularCodigo]
 [AlunoNumero] - [PortalAcademico].[MatriculaInscricao].[AlunoNumero]
 [Estado] - [PortalAcademico].[MatriculaInscricao].[Estado]
 [TipoAvaliacao] - [PortalAcademico].[MatriculaInscricao].[TipoAvaliacao]
 [NotaEFolios] – [Moodle].[Avaliacao].[Nota]
 [NotaPFolio] - [Moodle].[Avaliacao].[Nota]
 [NotaPFolioRecurso] - [Moodle].[Avaliacao].[Nota]
 [NotaExame] - [Moodle].[Avaliacao].[Nota]
 [NotaExameRecurso] - [Moodle].[Avaliacao].[Nota]
 [NotaFinal] - [PortalAcademico].[MatriculaInscricao].[NotaFinal]
 [DataEstado] - [PortalAcademico].[MatriculaInscricao].[DataEstado]

Entidade **CodigoPostal:**

Carregada com dados a partir de CD dos CTT.

Entidade **DivisaoAdministrativa:**

Carregada com dados a partir do site www.ine.pt.

Entidade **EstadoCivil:**

Foram carregados manualmente com os dados:

EstadoCivilCodigo	EstadoCivilDescricao
C	Casado
J	Separado judicialmente de pessoas e bens
N	Não Definido
S	Solteiro
U	União de facto
V	Viúvo

TABELA 4.2 - Dados Estado Civil

Entidade **EstadoMatricula:**

Foram carregados manualmente com os dados:

Codigo	EstadoMatricula
A	Aprovado
I	Inscrito
N	Anulado
R	Reprovado

TABELA 4.3 - Dados Estado Matrícula

Entidade **EstadoPerfil:**

Foram carregados manualmente com os dados:

Codigo	Descricao
A	Ativo
I	Inativo
C	Concluído

TABELA 4.4 - Dados Estado Perfil

Entidade **MatriculaInscricao:**

Construído procedimento que percorre todos os registos da entidade [Perfil] do subsistema [UAb-PortalAcademico], e que cria os registos aleatoriamente mas com a influência ao nível das probabilidades forçadas.

Para o 1º Semestre e 1º Ano são inscritas aleatoriamente até 5 UCs do curso 1º semestre/1º ano.

Para o atributo [Estado] temos as seguintes probabilidades:

Ano de inscrição > 2017: Aprovado: 30% / Avaliado 50% / Inscrito: 20%

Ano de inscrição <= 2017: Aprovado: 45% / Avaliado: 10% / Inscrito 45%

Para os restantes semestres são consideradas inscrições aleatórias até 6 Ucs do curso, com um ciclo até a data de inscrição atingir o ano de 2018 ou ter atingido um total de 180 ECTS.

Distribuição das probabilidades para o atributo [Estado]:

1º Ano:

Ano de inscrição > 2017: Aprovado: 20% / Avaliado: 30% / Inscrito: 50%

Ano de inscrição <= 2017: Aprovado: 20% / Avaliado: 35% / Inscrito: 45%

2º Ano:

Ano de inscrição > 2017: Aprovado: 35% / Avaliado: 45% / Inscrito: 20%

Ano de inscrição <= 2017: Aprovado: 45% / Avaliado: 15% / Inscrito: 40%

3º Ano:

Ano de inscrição > 2017: Aprovado: 40% / Avaliado: 30% / Inscrito: 30%

Ano de inscrição <= 2017: Aprovado: 60% / Avaliado: 30% / Inscrito: 10%

Entidade **Morada:**

Procedimento construído com as seguintes características:

[NumeroMorada] – número incremental

[PaisMorada] – aleatório e de acordo com tabela de probabilidades construída a partir de informações extraídas no relatório de atividades da UAb de 2016

[Artéria] - '**** Nome aplicada máscara de dados *****'

[Localidade] e [CodigoPostal] - aleatório e de acordo com tabela de probabilidades construída a partir de informações extraídas no relatório de atividades da UAb de 2016

Entidade **Pais:**

www.ine.pt

Entidade **Perfil:**

Construído procedimento percorrendo todos os registos da entidade [Aluno] do subsistema [UAb-PortalAcademico] e com as seguintes características:

[CursoID] – Distribuição uniforme dos cursos: 10% (10 cursos do 1º ciclo)

[DataCriacao] – 2009: 20% / 2010: 25% / 2011: 25% / [2012,2017]: 30%

[Estado] – Ativo: 95% / Inativo 5%

No final é feita uma atualização para [Estado] = 'C' (Concluído) para todos os perfis cuja soma das UCs com estão Aprovado seja igual a 180 ECTS, e Data de Estado igual à data máxima da UC com Estado Aprovado.

Entidade **Sexo**:

Foram carregados manualmente com os dados:

SexoCodigo	SexoDescricao
F	Feminino
M	Masculino
N	Não Definido

TABELA 4.5 - Dados Sexo

As restantes entidades não foram alvo de criação de dados artificiais:

- Filiacao
- Mensalidade
- Movimento
- OutraInformacao
- Propina
- TipoFiliacao
- TipoMovimento

4.5. MODELO SITCON

As entidades não foram alvo de criação de dados artificiais:

- EstadoPedido
- Mensagem
- Pedido
- TipoPedido

5. MODELO OLAP (DW)

5.1. INTRODUÇÃO

Tal como referido anteriormente a construção do modelo do DW está condicionada em primeiro lugar pelos dados disponíveis nas bases de dados transacionais que vão

alimentar o DW, e em segundo lugar pelas questões que se pretendem ver respondidas pelo DW.

Neste capítulo são identificadas as tabelas de factos e dimensões necessárias para a criação do modelo DW, assim como as relações existentes na constelação.

Serão também identificadas individualmente as estrelas que constituem a constelação do DW, complementada com a informação das fontes de dados de cada estrela e das consultas possíveis em cada uma delas.

5.2. DIMENSÕES E FACTOS (TABELAS)

Para este DW e com base nos dados existentes nas bases de dados OLTP foram identificadas dimensões como *Slowly Changing Dimension* (SCD) do tipo 1. Isto é, sem registo de histórico:

Dimensão	Fonte de dados
DimAluno	[PortalAcademico].[Aluno]
DimAnoLectivo	Criados manualmente
DimCentroLocalAprendizagem	[PortalAcademico].[CentroLocalAprendizagem]
DimCodigoPostal	[PortalAcademico].[CodigoPostal]
DimCurso	[Portal].[Curso]
DimTempo *	Criados manualmente por procedimento
DimDepartamento	[Portal].[Departamento]
DimDocente	[Portal].[Docente]
DimEstadoCivil	Criados manualmente
DimSexo	Criados manualmente
DimUnidadeCurricular	Criados manualmente

TABELA 5.1 - Dimensões

* A dimensão "Tempo" é uma SCD do tipo 0 (depois de carregada não são necessárias mais atualizações).

E as tabelas de Factos:

- **Aluno:** Tabela que servirá de base ao objeto de estudo na visão Aluno. Os dados para análise serão baseados nas notas das UCs, sendo extraídas as métricas para o efeito. A tabela base será a [AvaliacaoUnidadeCurricular].
- **UC:** Esta tabela vai conter os dados relacionados com as Unidades Curriculares, sendo esta a referência das consultas.
- **Curso:** Esta tabela vai conter os dados relacionados com os Cursos.

Quadro identificativo das dimensões versus tabelas de Factos:

Dimensões \ Factos	Aluno	UC	Curso
DimAluno	X		
DimAnoLectivo	X		X
DimCentroLocalAprendizagem			
DimCodigoPostal	X	X	X
DimCurso			X
DimTempo	X	X	X
DimDepartamento			X
DimDocente		X	
DimEstadoCivil	X		
DimSexo	X		
DimUnidadeCurricular	X	X	X

TABELA 5.2 - Tabelas Dimensões vs Factos

5.3. SUCESSO ESCOLAR / ABANDONO ESCOLAR (DEFINIÇÃO)

Em termos de definição do Sucesso Escolar e Abandono Escolar no contexto da base de dados OLTP criada para este projeto, temos como chave a entidade [AvaliacaoUnidadeCurricular] e atributo [Estado] (da matricula).

Conforme já identificado no capítulo 2.2 o atributo [Estado] da entidade [AvaliacaoUnidadeCurricular] pode assumir 4 estados:

Inscrito: Para identificar a existência do registo da UC.

Avaliado: Para identificar que ocorreu existência de avaliação e implica que a nota está preenchida, isto é, não nula.

Aprovado: Indica que além de ter sido Avaliado teve uma nota superior ou igual a 10. Este estado corresponde diretamente ao **Sucesso**.

Anulado: Matricula que foram anuladas (**Abandono**)

5.4. ESTRELA ALUNO

O modelo definido para a estrela é o seguinte:

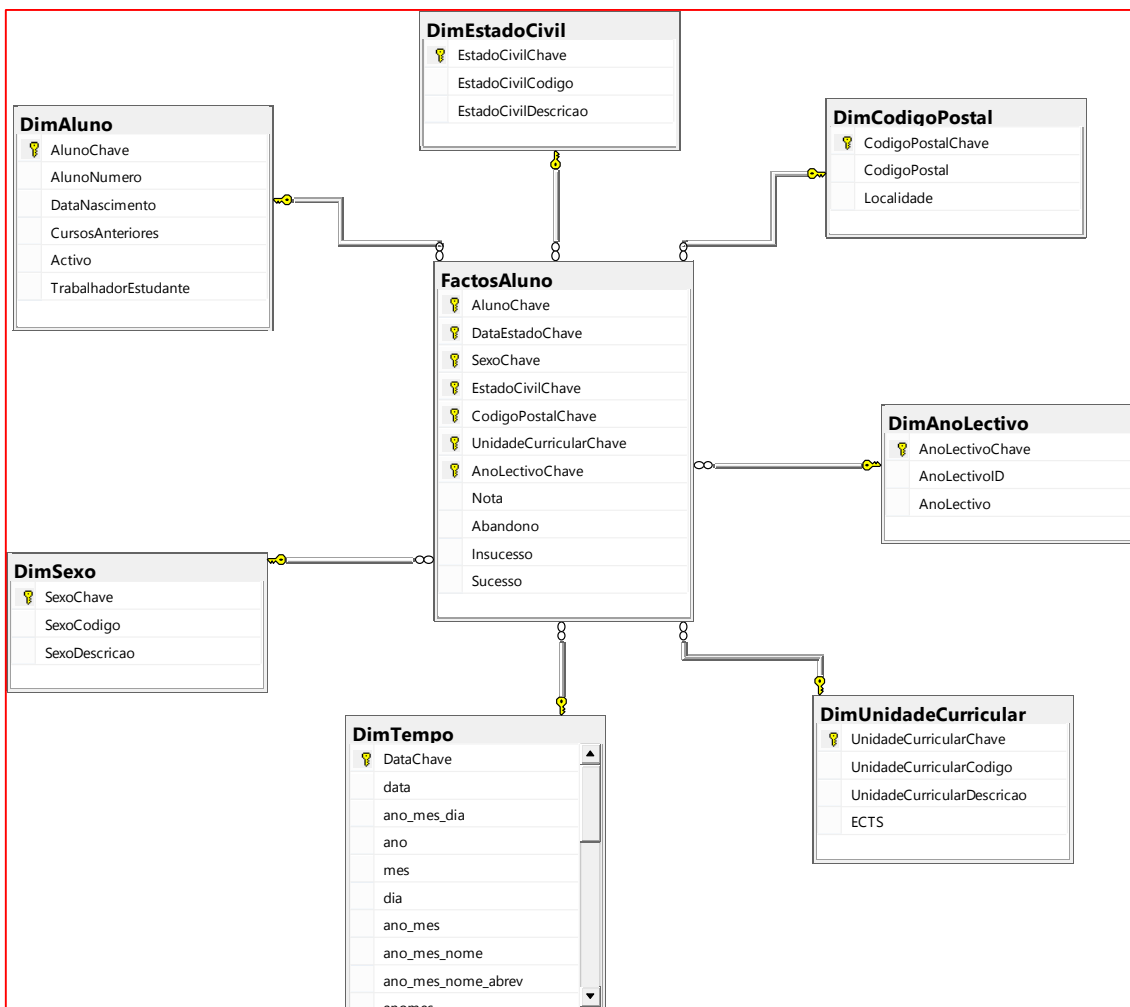


FIGURA 5.1 - Estrela Aluno

As fontes de dados para a estrela são obtidos da base dados transacional [PortalAcademico].

Composta pelas dimensões:

Dimensão	Tabela Transacional
DimAluno	[PortalAcademico].Aluno
DimTempo	Não aplicável
DimSexo	[PortalAcademico].Sexo
DimEstadoCivil	[PortalAcademico].EstadoCivil
DimCodigoPostal	[PortalAcademico].CodigoPostal
DimUnidadeCurricular	[Portal].UnidadeCurricular
DimAnoLectivo	[PortalAcademico].AnoLectivo

TABELA 5.3 - Dimensões da Estrela Aluno

O carregamento da tabela de factos é feito a partir do cruzamento das chaves de negócio da BD OLTP e as dimensões da BD OLAP, no sentido de obter a chave única da dimensão em tratamento.

Em baixo estão representadas as chaves de negócio usadas para obter as chaves das dimensões. Esta representação inclui as bases de dados e as tabelas de origem (negócio) utilizadas como fontes de dados neste processo:

Chave DW	Chave negócio
[AlunoChave]	[UAb-DW].[DimAluno].[AlunoNumero] = [PortalAcademico].[Aluno].[AlunoNumero]
[DataEstadoChave]	[UAb-DW].[DimTempo].[data] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[DataEstado]
[SexoChave]	[UAb-DW].[DimSexo].[SexoCodigo] = [PortalAcademico].[Aluno].[Sexo]
[EstadoCivilChave]	[UAb-DW].[DimEstadoCivil].[EstadoCivilCodigo] = [PortalAcademico].[Aluno].[EstadoCivil]
[CodigoPostalChave]	[UAb-DW].[DimCodigoPostal].[CodigoPostal] = [PortalAcademico].[CodigoPostal].[CodigoPostal] Join [PortalAcademico].[Morada].[CodigoPostal] = [PortalAcademico].[CodigoPostal].[CodigoPostal] Join [PortalAcademico].[Aluno].[Numero_Morada] = [PortalAcademico].[Morada].[NumeroMorada] Join [PortalAcademico].[Aluno].[AlunoNumero] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[AlunoNumero]
[UnidadeCurricularChave]	[UAb-DW].[DimUnidadeCurricular].[UnidadeCurricularCodigo] = [Portal].[UnidadeCurricular].[UnidadeCurricularCodigo]
[AnoLectivoChave]	[UAb-DW].[DimAnoLectivo].[DimAnoLectivoID] = [PortalAcademico].[AnoLectivo].[AnoLectivoID]

TABELA 5.4 - Chaves e fontes de dados da Estrela Aluno

Consultas que podem ser respondidas com a estrela Aluno:

- Taxa de Sucesso e Abandono (global e distribuída geograficamente)
- Distribuição de Sucesso e Abandono por género
- Distribuição de Sucesso e Abandono por estado civil
- Distribuição de Sucesso e Abandono por nº de cursos anteriores
- Distribuição de Sucesso e Abandono por idade
- Distribuição de Sucesso e Abandono por alunos com e sem estatuto de Trabalhador/Estudante

- Distância entre localidade da morada do Aluno e da localidade das CLAs

5.5. ESTRELA UC

O modelo definido para a estrela Unidade Curricular é o seguinte:

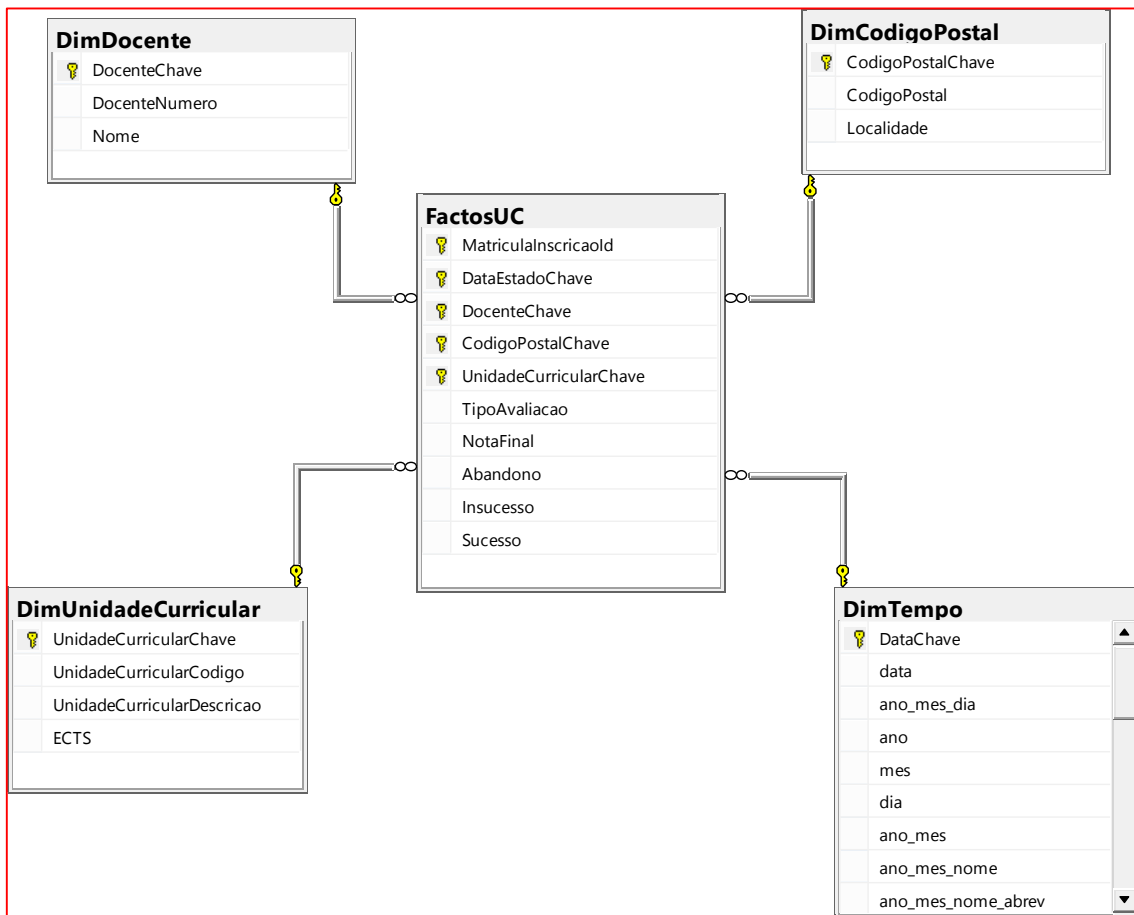


FIGURA 5.2 - Estrela UC

Composta pelas dimensões:

Dimensão	Tabela Transaccional
DimTempo	Não aplicável
DimDocente	[Portal].Docente
DimCodigoPostal	[PortalAcademico].CodigoPostal
DimUnidadeCurricular	[Portal].UnidadeCurricular

TABELA 5.5 - Dimensões da Estrela UC

Representação das chaves de negócio usadas para obter as chaves das dimensões:

Chave DW	Chave negócio
[DataEstadoChave]	[UAb-DW].[DimTempo].[data] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[DataEstado]
[DocenteChave]	[UAb-DW].[DimDocente].[DocenteNumero] = [Portal].[Docente].[DocenteNumero] Join [Portal].[DocenteUnidadeCurricular].[UnidadeCurricularCodigo]]= [PortalAcademico].[AvaliacaoUnidadeCurricular].[UnidadeCurricularCodigo]
[CodigoPostalChave]	[UAb-DW].[DimCodigoPostal].[CodigoPostal] = [PortalAcademico].[CodigoPostal].[CodigoPostal] Join [PortalAcademico].[Morada].[CodigoPostal] = [PortalAcademico].[CodigoPostal].[CodigoPostal] Join [PortalAcademico].[Aluno].[Numero_Morada] = [PortalAcademico].[Morada].[NumeroMorada] Join [PortalAcademico].[Aluno].[AlunoNumero] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[AlunoNumero]
[UnidadeCurricularChave]	[UAb-DW].[DimUnidadeCurricular].[UnidadeCurricularCodigo] = [Portal].[UnidadeCurricular].[UnidadeCurricularCodigo]

TABELA 5.6 - Chaves e fontes de dados da Estrela UC

Consultas que podem ser respondidas com a estrela UC:

- Tipo de avaliação associada ao Sucesso e Abandono Escolar
- Média das notas associada ao Sucesso Escolar
- Top 10 Docentes associadas às Taxas de Sucesso e Abandono Escolar
- Identificação das UCs e Taxas com avaliação Reprovado
- Distribuição do top 5 das melhores e piores médias das UCs geograficamente

5.6. ESTRELA CURSO

O modelo definido para a estrela Curso é o seguinte:

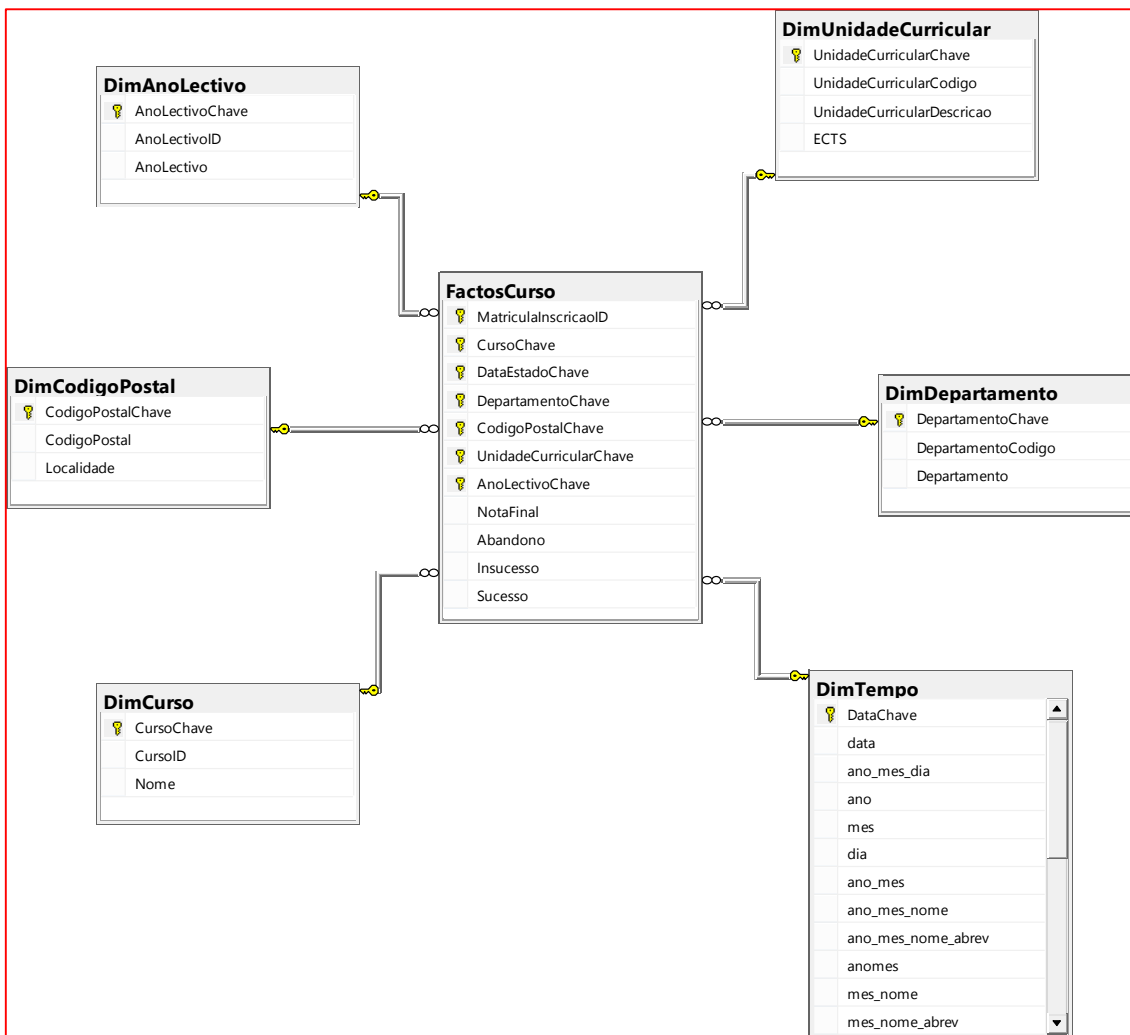


FIGURA 5.3 - Estrela Curso

Composta pelas dimensões:

Dimensão	Tabela Transaccional
DimCurso	[Portal].Curso
DimTempo	Não aplicável
DimDepartamento	[Portal].Departamento
DimCodigoPostal	[PortalAcademico].CodigoPostal
DimUnidadeCurricular	[Portal].UnidadeCurricular
DimAnoLectivo	[PortalAcademico].AnoLectivo

TABELA 5.7 - Dimensões da Estrela Curso

Tabela de Factos Curso:

Chave DW	Chave negócio
[CursoChave]	[UAb-DW].[DimCurso].[CursoID] = [Portal].[Curso].[CursoID] Join [Portal].[UnidadeCurricularCurso].[UnidadeCurricularCodigo] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[UnidadeCurricularCodigo]
[DataEstadoChave]	[UAb-DW].[DimTempo].[data] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[DataEstado]
[DepartamentoChave]	[UAb-DW].[DimDepartamento].[DepartamentoCodigo] = [Portal].[Departamento].[DepartamentoCodigo] Join [Portal].[RelDepartamentoCurso].[CursoID] = [Portal].[Curso].[CursoID] Join [Portal].[UnidadeCurricularCurso].[UnidadeCurricularCodigo] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[UnidadeCurricularCodigo]
[CodigoPostalChave]	[UAb-DW].[DimCodigoPostal].[CodigoPostal] = [PortalAcademico].[CodigoPostal].[CodigoPostal] Join [PortalAcademico].[Morada].[CodigoPostal] = [PortalAcademico].[CodigoPostal].[CodigoPostal] Join [PortalAcademico].[Aluno].[Numero_Morada] = [PortalAcademico].[Morada].[NumeroMorada] Join [PortalAcademico].[Aluno].[AlunoNumero] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[AlunoNumero]
[UnidadeCurricularChave]	[UAb-DW].[DimUnidadeCurricular].[UnidadeCurricularCodigo] = [PortalAcademico].[AvaliacaoUnidadeCurricular].[UnidadeCurricularCodigo]
[AnoLectivoChave]	[UAb-DW].[DimAnoLectivo].[AnoLectivoID] = [PortalAcademico].[AnoLectivo].[AnoLectivoID]

TABELA 5.8 - Chaves e fontes de dados da Estrela Curso

Consultas que podem ser respondidas com a estrela UC:

- Número de UCs associado Abandono Escolar por Ano Letivo
- Departamentos associados ao Sucesso e Abandono Escolar
- Identificação dos Cursos associadas ao Sucesso e Abandono Escolar por ano letivo
- Distribuição geográfica dos Cursos associada ao Sucesso e Abandono Escolar

6. ETL

6.1. INTRODUÇÃO

Neste capítulo descrevem-se os passos necessários para o carregamento das tabelas dimensões e de fatos do modelo DW criado e descrito no capítulo anterior.

Para o carregamento das dimensões foi utilizada a ferramenta da *Microsoft* “*Sql Server Data Tools*”, utilizando o componente específico “*Slowly Changing Dimension*” que permite de uma forma fácil, rápida e eficaz, a construção de um agente que gere o carregamento inicial de uma dimensão assim como das iterações posteriores.

Para as tabelas de factos foi utilizado o pacote “*TaskFactory*” mais concretamente o componente “*Fact Table Destination*” que à semelhança do “*Slowly Changing Dimension*” para o carregamento das dimensões permite a definição e carregamento das tabelas de fatos de uma forma quase intuitiva, assim como a gestão iterativa/periódica dos carregamentos.

Uma vez definido o modelo do DW e as regras de extração e transformação, estas ferramentas facilitam a criação dos processos ETL no sentido em que a gestão do processo é maioritariamente feito com base em elementos visuais, e os processos mais convencionais do tipo escrita de comandos e procedimentos ficam renegados para segundo plano, podendo não ser totalmente eliminados, mas ficam muito diminuídos na sua necessidade.

6.2. DEFINIÇÃO DOS COMPONENTES

Para a construção das dimensões foram usados os seguintes componentes da ferramenta SSDT:

OLE DB Source: Extrai dados de uma base de dados relacional. Foi utilizado para se conectar às BDs [UAb-Portal] e [UAb-PortalAcademico].

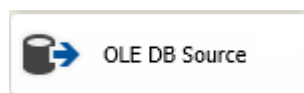


FIGURA 6.1 - OLE DB Source

Slowly Changing Dimension: Componente utilizado para gerir especificamente dimensões de um DW. Insere novos registos e atualiza registos existentes com uma gestão aplicada sobre regras previamente definidas aquando da criação e parametrização do componente.

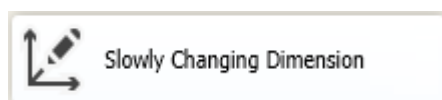


FIGURA 6.2 - Slowly Changing Dimension

ADO NET Destination: Componente cujo propósito serve para inserir registos numa tabela existente, mais concretamente numa tabela tipo dimensão, pois neste caso representa o *output* do componente Slowly Changing Dimension.



FIGURA 6.3 - ADO NET Destination

TF Fact Table Destination: Componente para gerir as tabelas de factos de Data Warehouses. Este componente é muito poderoso e fácil de usar no que diz respeito à gestão do carregamento de tabelas transacionais para tabelas de factos.



FIGURA 6.4 - Task Factory Fact Table Destination

Merge Join: Componente que serve para combinar 2 fluxos de dados ordenados num único fluxo de dados. Pode-se fazer a analogia com o comando *Join* (*full*, *left* ou *inner join*).



FIGURA 6.5 - Merge Join

Sort: Este componente é utilizado em conjunto com o componente *Merge*, uma vez que no *Merge* obriga a que os fluxos de dados estejam ordenados pelo mesmo atributo.



FIGURA 6.6 - Sort

Derived Column: Componente utilizado para criar novas colunas baseado em expressões ou valores de outras colunas.

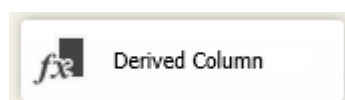


TABELA 6.1 - Derived Column

Para aceder às várias bases de dados temos de parametrizar as ligações necessárias para construir o processo ETL, concretamente as ligações às bases de dados OLTP e OLAP.

Por exemplo, para a ligação à base de dados UAb-DW foi parametrizada a seguinte ligação:

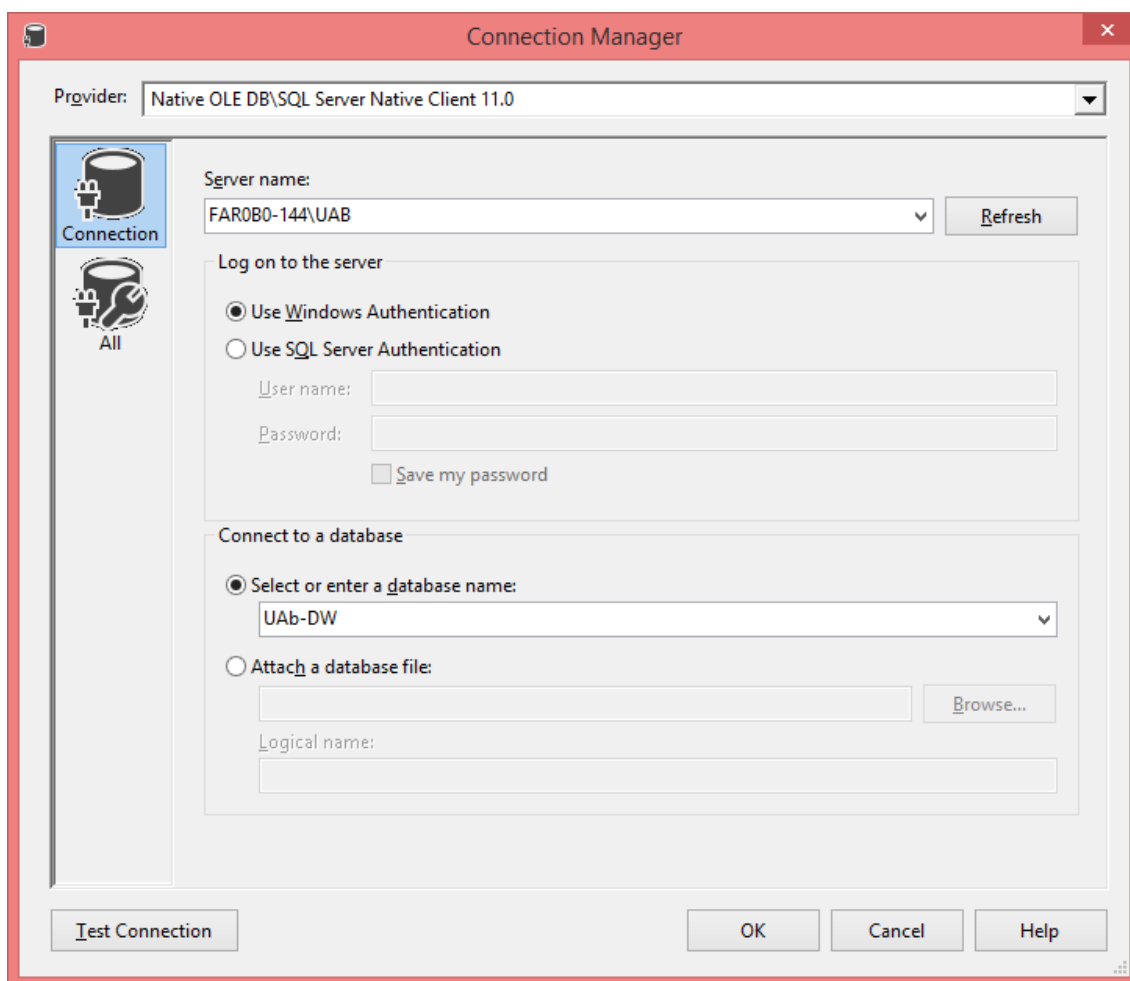


FIGURA 6.7 - Ligação UAb-DW

Para organizar o processo de carregamento e os respetivos fluxos foram criados vários *Containers* de componentes.

Container é uma “caixa” onde são organizadas lógica e fisicamente as várias tarefas do processo ETL e que pode ser executada isoladamente com todas as tarefas incluídas.

6.3. TABELAS DE DIMENSÕES

Para os carregamentos foi desenhado um diagrama para controlar o fluxo. Este fluxo é desenhado diretamente na ferramenta e dentro de cada componente do fluxo foram implementados os vários fluxos de dados para carregar as dimensões.

O fluxo de controlo total foi desenhado conforme ilustrado na figura seguinte:

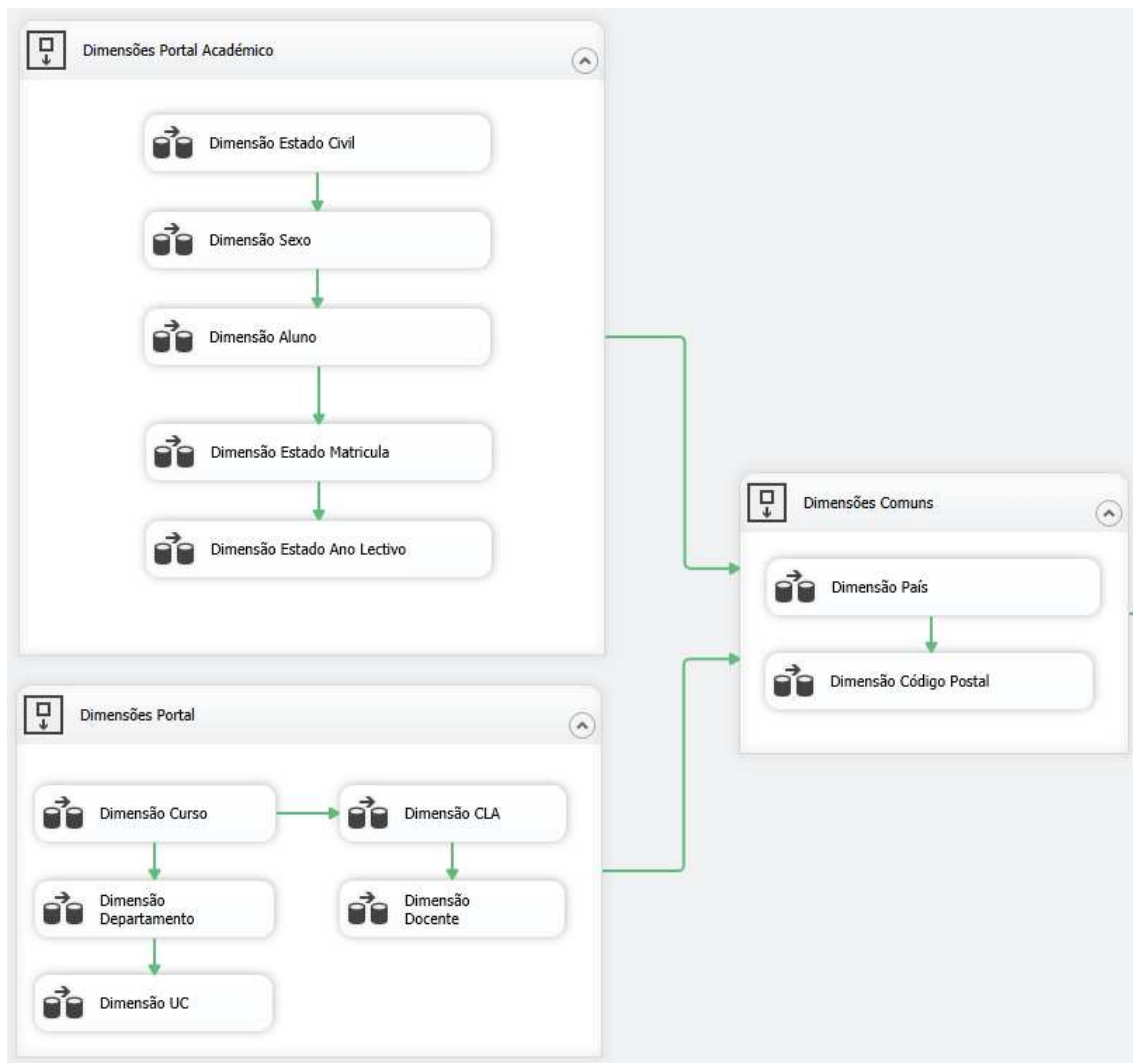


FIGURA 6.8 - Fluxo De Controllo

O processo global foi separado em 3 *Containers* lógicos/físicos, sendo que não existindo dependência entre eles, os mesmos podem ter execuções em paralelo nem que seja parcialmente para garantir um processamento mais rápido:

- *Container* Dimensões Portal Académico
- *Container* Dimensões Portal
- *Container* Dimensões Comuns

- Na construção do "*Container* Dimensões Portal Académico" foram definidos os seguintes carregamentos de dimensões:



FIGURA 6.9 - Container "Portal Académico"

- Dimensão Estado Civil:

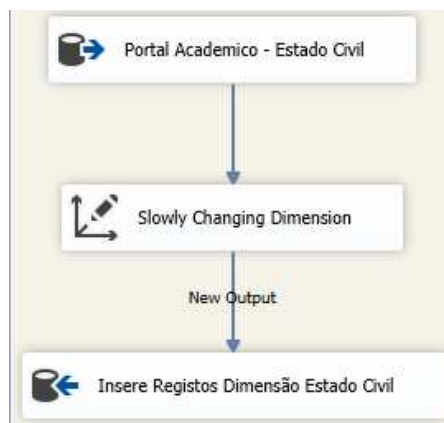


FIGURA 6.10 - Fluxo Estado Civil

Para este primeiro processo de fluxo de dados ilustram-se os parâmetros dos vários componentes.

OLE DB Source (Portal Académico - Estado Civil):

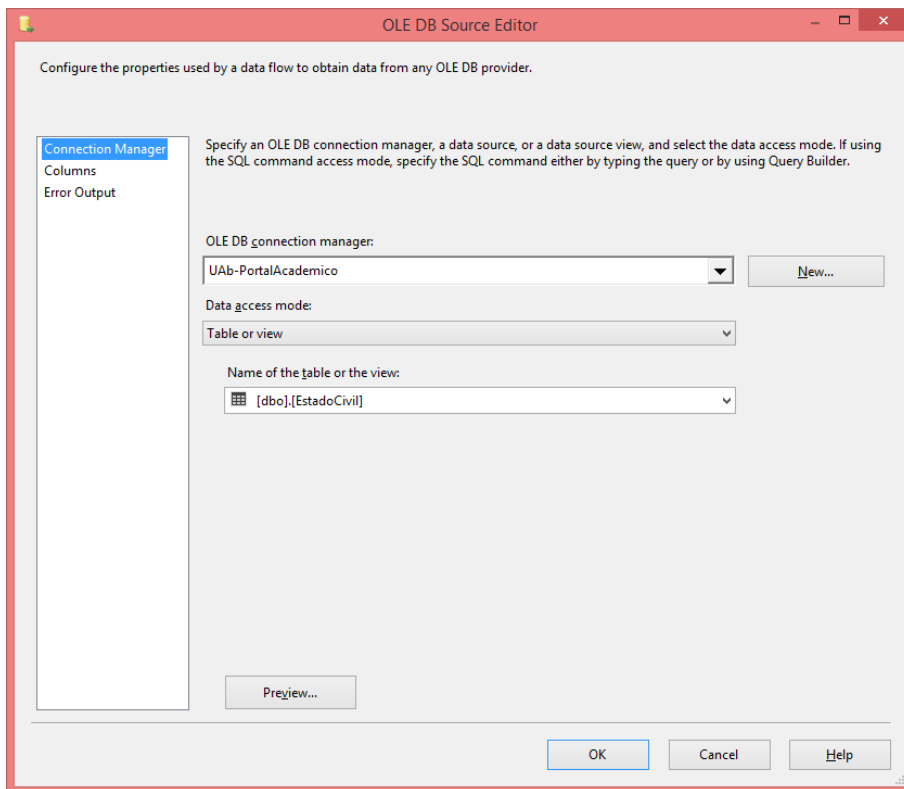


FIGURA 6.11 - Passo 1 do OLE DB Source Editor

Identifica-se a conexão e o nome da tabela de origem, neste caso a tabela EstadoCivil da base de dados UAb-PortalAcademico.

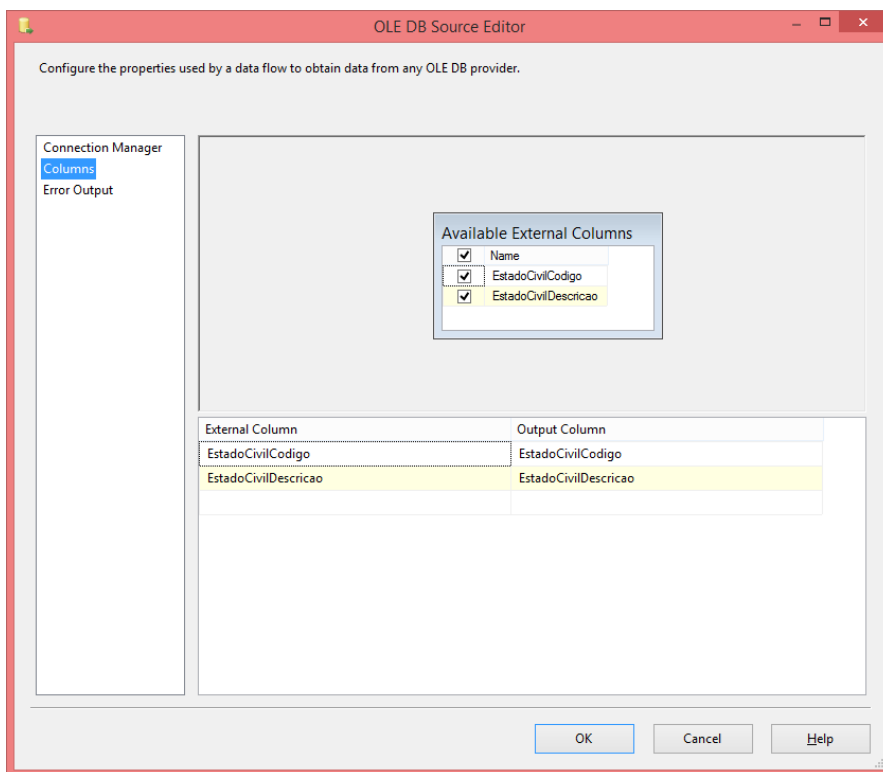


FIGURA 6.12 - Passo 2 do OLE DB Source Editor

Neste Passo escolhemos os atributos necessários para a construção da tabela dimensão. No caso em concreto foram escolhidos todos os atributos: EstadoCivilCodigo e EstadoCivilDescricao.

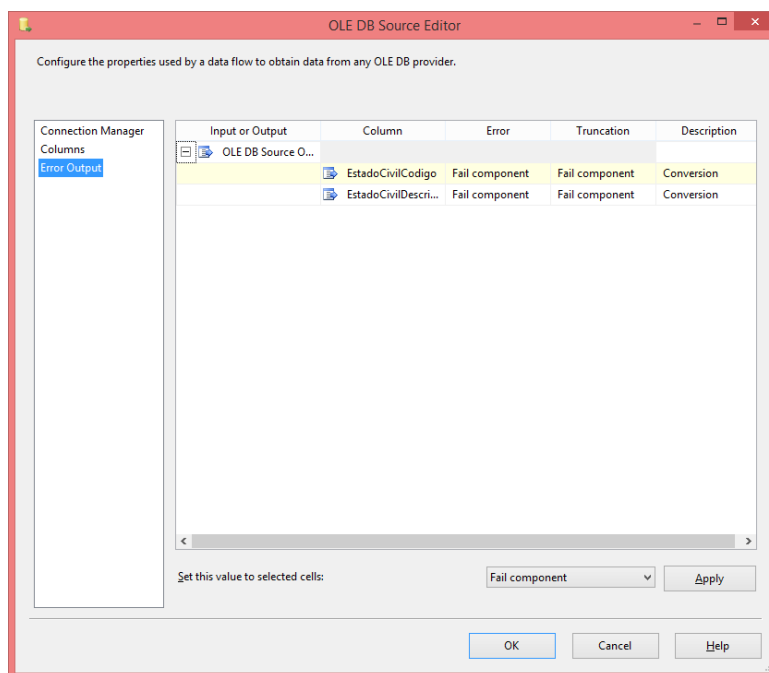


FIGURA 6.13 - Passo 3 do OLE DB Source Editor

Caso seja necessário tratamento de registos com erro, pode-se usar este passo para parametrizar o *output*. No caso em concreto não foi necessário configurar este passo.

Slowly Changing Dimension (Portal Académico - Estado Civil):

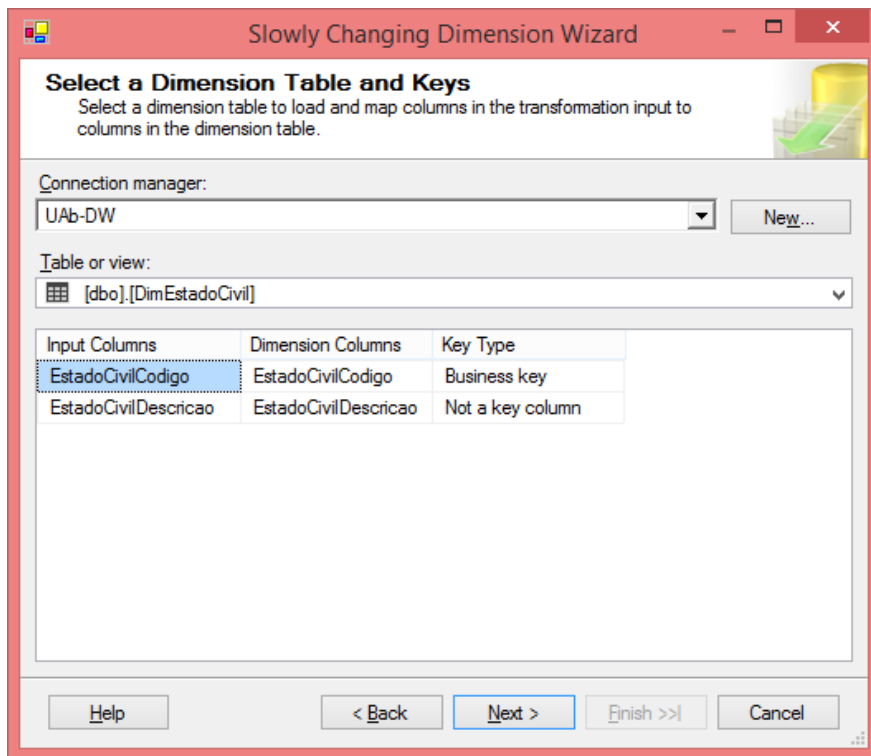


FIGURA 6.14 - Passo 1 do SCD Wizard

Neste passo identifica-se a conexão e a tabela de destino (DimEstadoCivil).

Identificam-se também as chaves de negócio da tabela de origem (neste caso EstadoCivilCodigo).

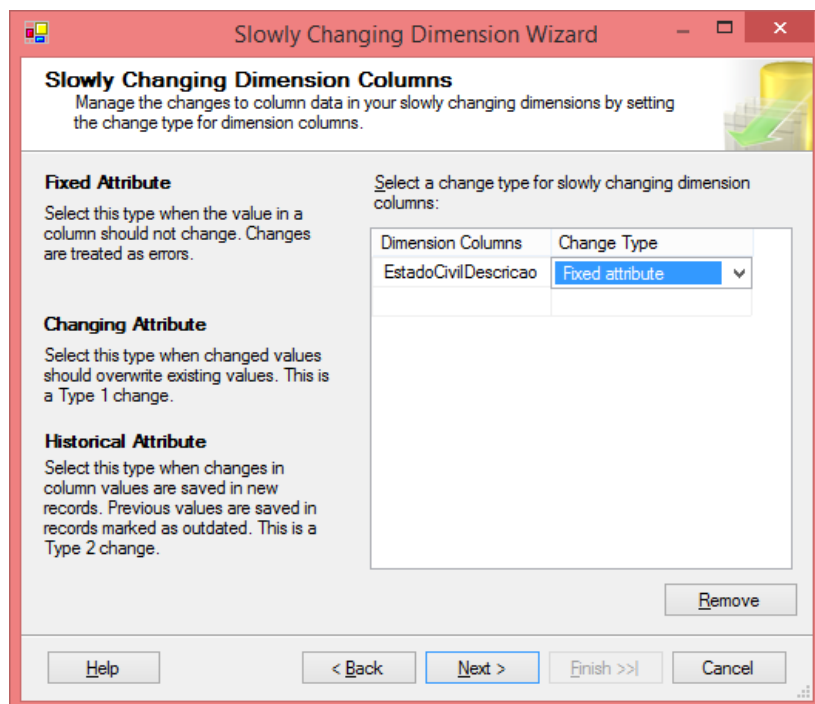


FIGURA 6.15 - Passo 2 do SCD Wizard

Neste passo identificam-se que tipos de alterações podem sofrer os restantes atributos. Este passo é importante para a gestão automática das alterações de dados da

dimensão. No caso em concreto parametrizámos o atributo EstadoCivilDescricao como "Fixed Attribute", ou seja, o atributo não sofrerá alterações. Outras opções disponíveis são:

- **Changing attribute:** atributo que sofrerá alterações e o componente tratará automaticamente das atualizações dos registos sempre que exista alterações deste atributo.

- **Historical attribute:** para os atributos desta natureza existe necessidade de criar 2 atributos adicionais do tipo "Data" na tabela dimensão. [Data Início] e [Data Fim], para que sempre que exista alteração ao conteúdo deste atributo o componente preencha a [Data Fim] com a data de sistema e criará novo atributo com [Data Início] igual à data de sistema e [Data Fim] igual a nulo. Desta forma conseguem-se identificar todas as alterações ao atributo cronologicamente.

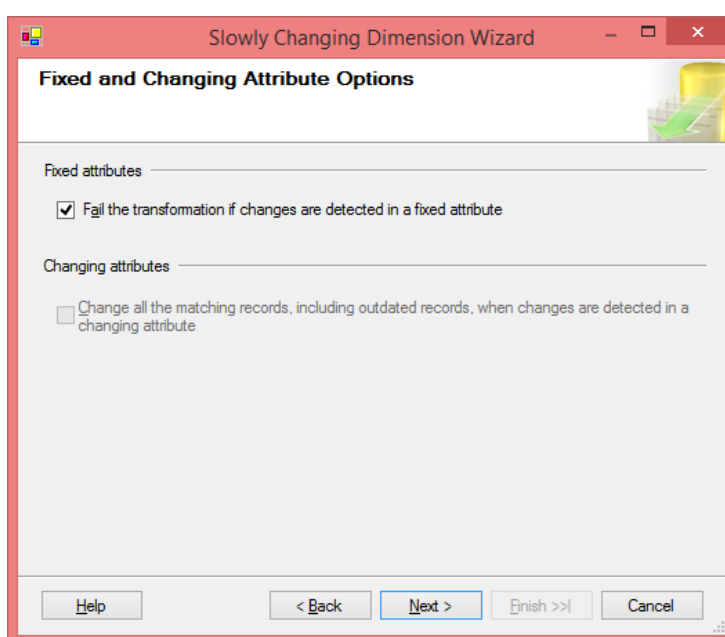


FIGURA 6.16 - Passo 3 do SCD Wizard

Caso existam alterações ao atributo que considerámos fixo (que não iria ter alterações), o processo será abortado.

ADO NET Destination (Portal Académico - Estado Civil):

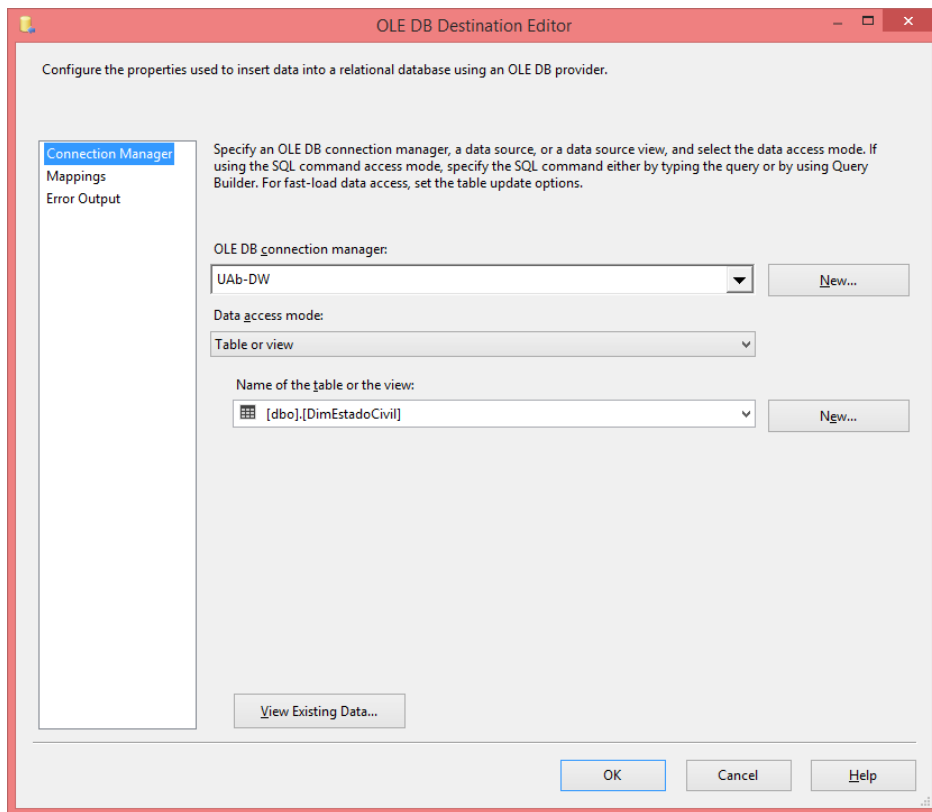


FIGURA 6.17 - Passo 1 do OLE DB Destination Editor

Identifica-se a conexão e o nome da tabela de destino, neste caso a tabela DimEstadoCivil da base de dados UAb-DW.

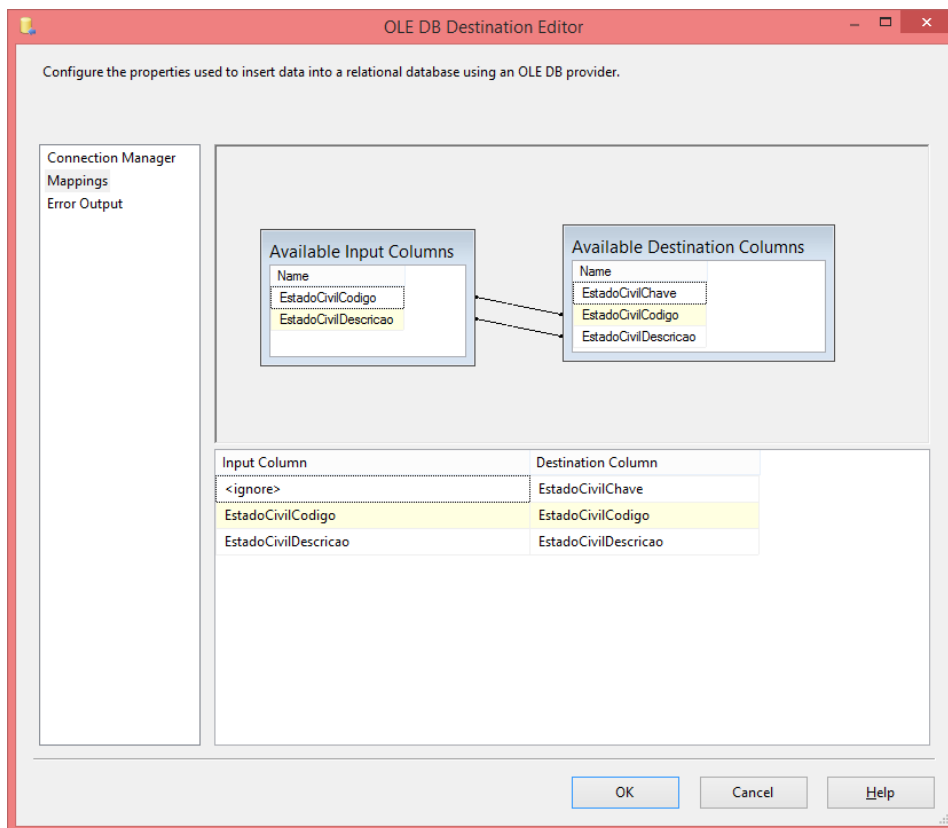


FIGURA 6.18 - Passo 2 do OLE DB Destination Editor

Neste passo definimos o mapeamento entre os atributos da tabela de origem e os atributos da tabela de destino.

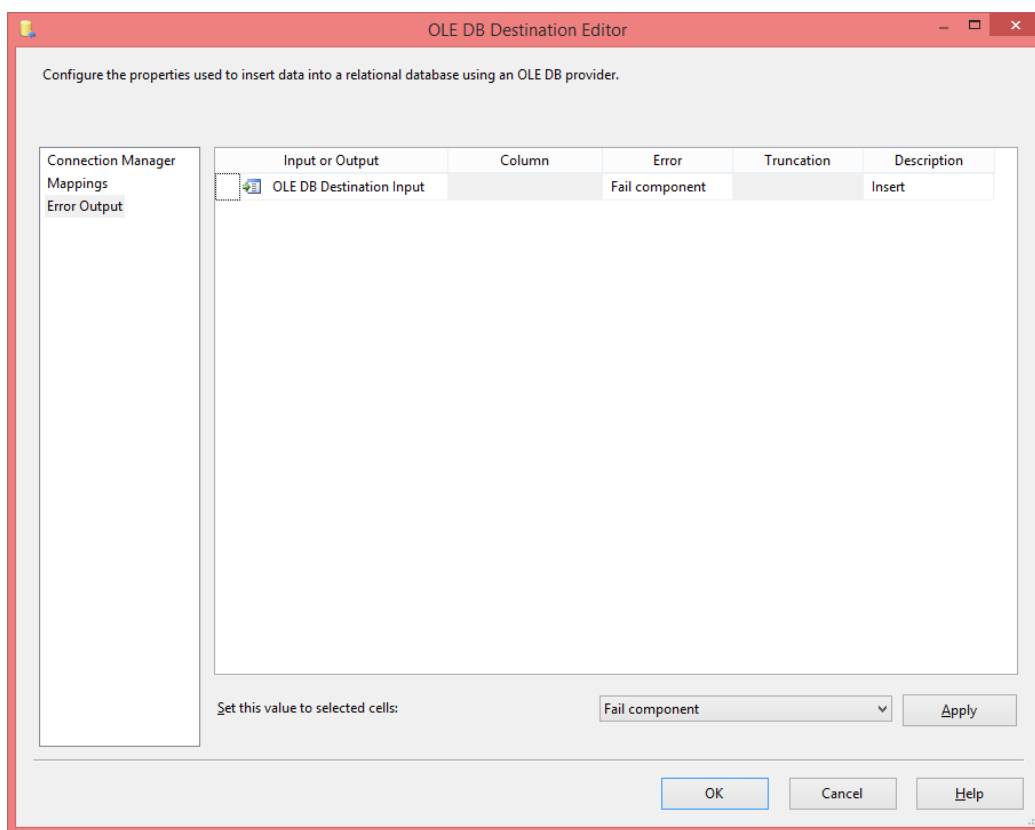


FIGURA 6.19 - Passo 3 do OLE DB Destination Editor

Caso seja necessário tratamento de registos com erro, pode-se usar este passo para parametrizar o *output*. No caso em concreto não foi necessário configurar este passo.

Em baixo descrevem-se os mapeamentos das tabelas por intermédio de uma tabela:

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimEstadoCivil].[EstadoCivilChave]
[EstadoCivil].[EstadoCivilCodigo]	[DimEstadoCivil].[EstadoCivilCodigo]
[EstadoCivil].[EstadoCivilDescricao]	[DimEstadoCivil].[EstadoCivilDescricao]

TABELA 6.2 - Mapeamento Estado Civil

Nas restantes dimensões, e como o método de parametrização é semelhante, procede-se à identificação dos carregamentos com base nas tabelas de mapeamentos.

Dimensão Sexo:

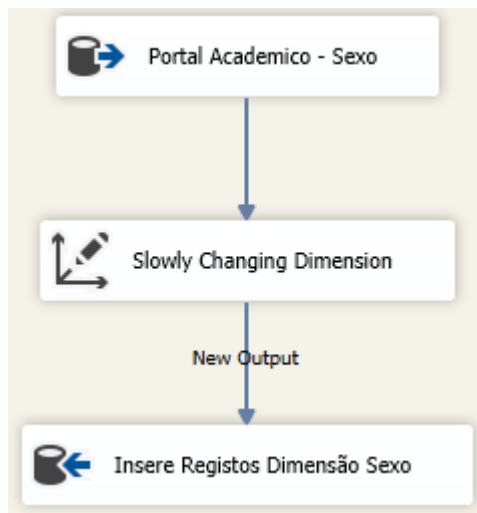


FIGURA 6.20 - Fluxo Sexo

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimSexo].[SexoChave]
[Sexo].[SexoCodigo]	[DimSexo].[SexoCodigo]
[Sexo].[SexoDescricao]	[DimSexo].[SexoDescricao]

TABELA 6.3 - Mapeamento Sexo

Dimensão Estado Matricula:

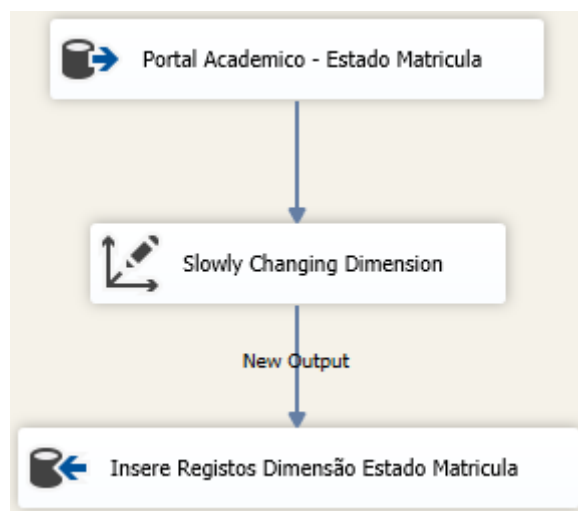


FIGURA 6.21 - Fluxo Estado Matricula

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
--	---

Não aplicável	[DimEstadoMatricula].[EstadoMatriculaChave]
[EstadoMatricula].[Codigo]	[DimEstadoMatricula].[EstadoMatriculaCodigo]
[EstadoMatricula].[EstadoMatricula]	[DimEstadoMatricula].[EstadoMatriculaDescricao]

TABELA 6.4 - Mapeamento Estado Matricula

Dimensão Ano Lectivo:

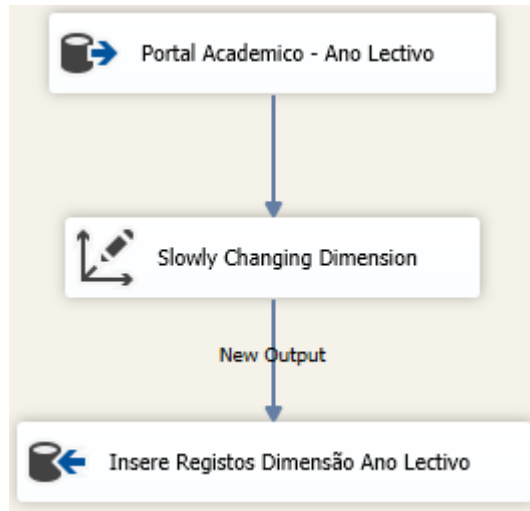


FIGURA 6.22 - Fluxo Ano Lectivo

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimAnoLectivo].[EstadoMatriculaChave]
[AnoLectivo].[Codigo]	[DimAnoLectivo].[EstadoMatriculaCodigo]
[AnoLectivo].[EstadoMatricula]	[DimAnoLectivo].[EstadoMatriculaDescricao]

TABELA 6.5 - Mapeamento Ano Lectivo

Dimensão Aluno:

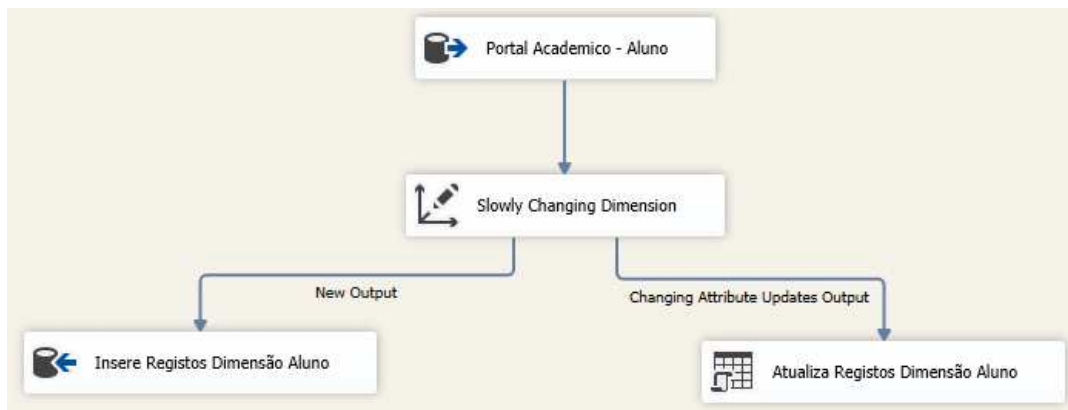


FIGURA 6.23 - Fluxo Aluno

A dimensão Aluno tem alguns atributos diferentes no que diz respeito ao tipo de alterações, neste caso existem atributos se pretendem que estejam sincronizados ao nível de conteúdo entre os sistemas transacionais e as respetivas dimensões.

No caso em concreto os atributos Activo, CursosAnteriores e TrabalhadorEstudante são do tipo alteráveis, conforme ilustrado na figura seguinte:

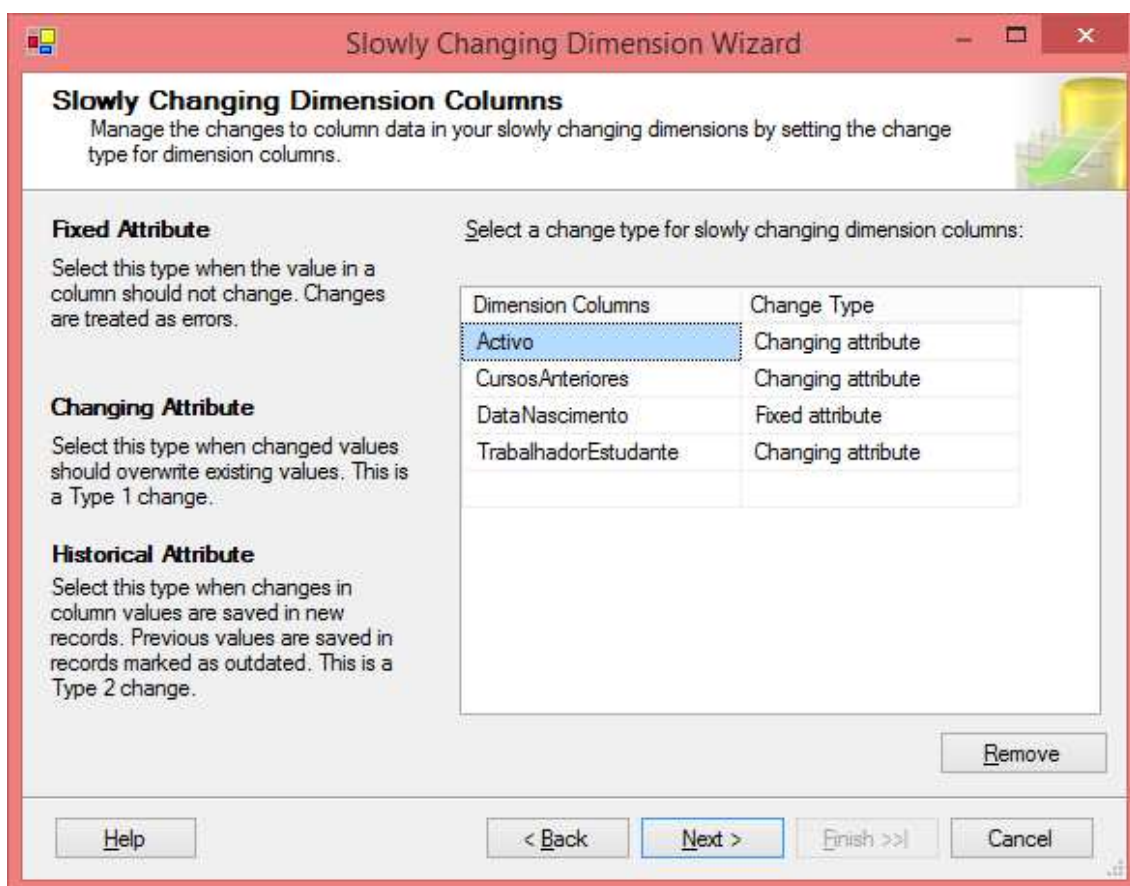


FIGURA 6.24 - Tipos de alterações dos atributos da tabela DimAluno

Pelo que o componente SCD tem 2 saídas de dados, uma para inserção e outra para alteração.

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimAluno].[AlunoChave]
[Aluno].[AlunoNumero]	[DimAluno].[AlunoNumero]
[Aluno].[DataNascimento]	[DimAluno].[DataNascimento]
[Aluno].[CursosAnteriores]	[DimAluno].[CursosAnteriores]
[Aluno].[Activo]	[DimAluno].[Activo]
[Aluno].[TrabalhadorEstudante]	[DimAluno].[TrabalhadorEstudante]

TABELA 6.6- Mapeamento Aluno

- De seguida apresentam-se as dimensões do Container "Portal"

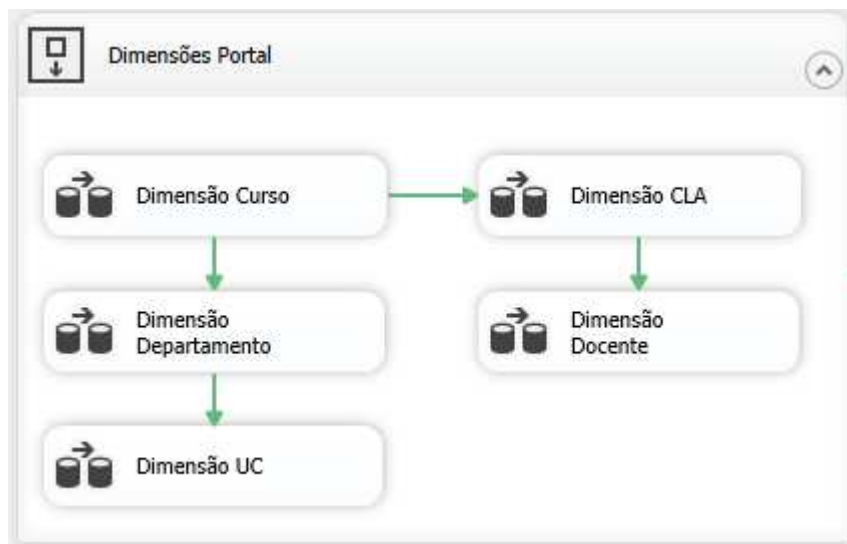


FIGURA 6.25 - Container "Portal"

Dimensão Curso:

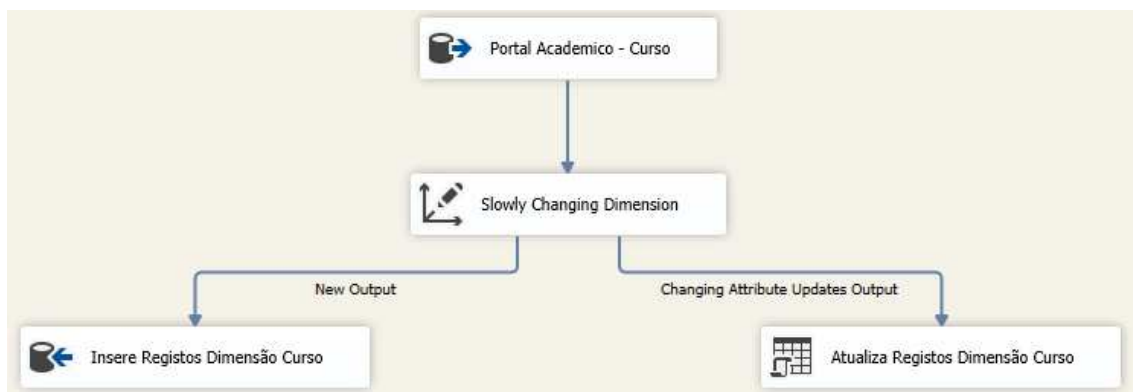


FIGURA 6.26 - Fluxo Curso

Atributo Nome alterável:

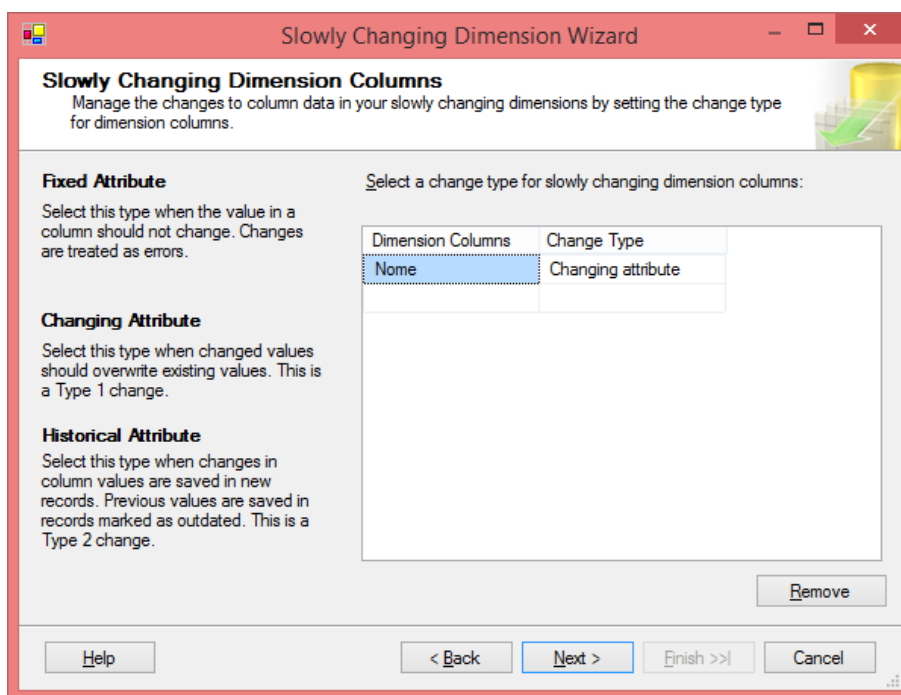


FIGURA 6.27 - Tipos de alterações dos atributos da tabela DimCurso

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimCurso].[CursoChave]
[Curso].[CursoID]	[DimCurso].[CursoID]
[Curso].[Nome]	[DimCurso].[Nome]

TABELA 6.7 - Mapeamento Curso

Dimensão Curso:

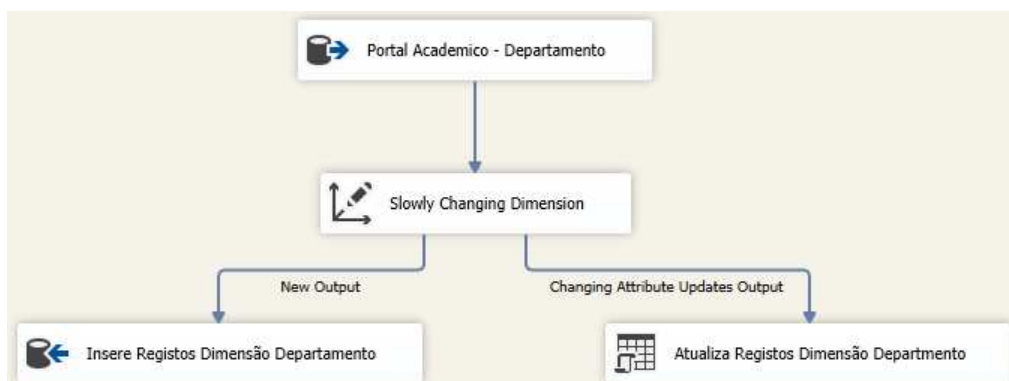


FIGURA 6.28 - Fluxo Departamento

Atributo Departamento alterável:

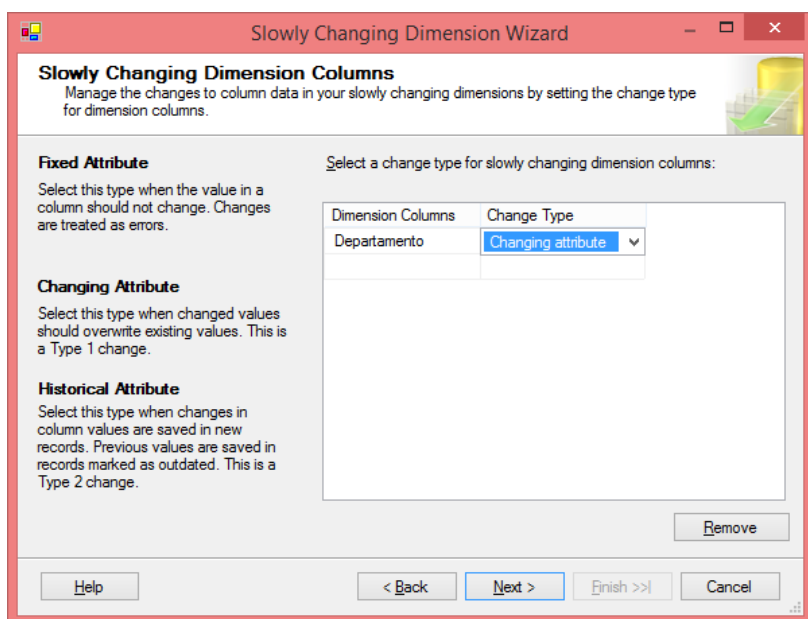


FIGURA 6.29 - Tipos de alterações dos atributos da tabela DimDepartamento

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimDepartamento].[DepartamentoChave]
[Departamento].[DepartamentoCodigo]	[DimDepartamento].[DepartamentoCodigo]
[Departamento].[Departamento]	[DimDepartamento].[Departamento]

TABELA 6.8 - Mapeamento Departamento

Dimensão Unidade Curricular:

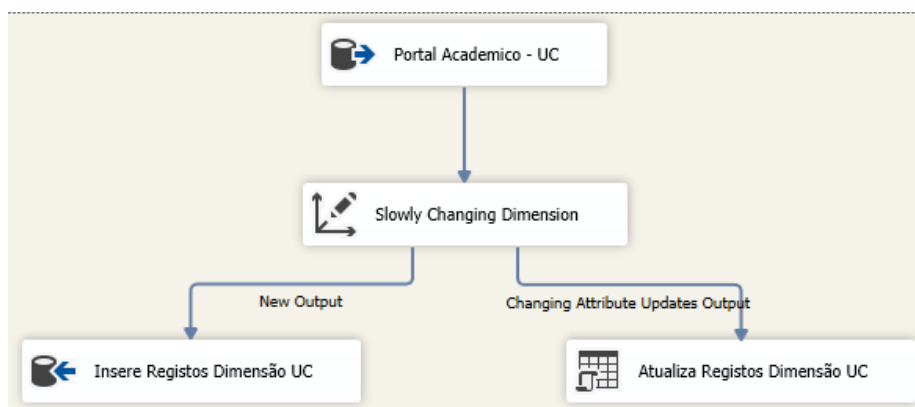


FIGURA 6.30 - Fluxo UC

Atributo UnidadeCurricularDescricao alterável:

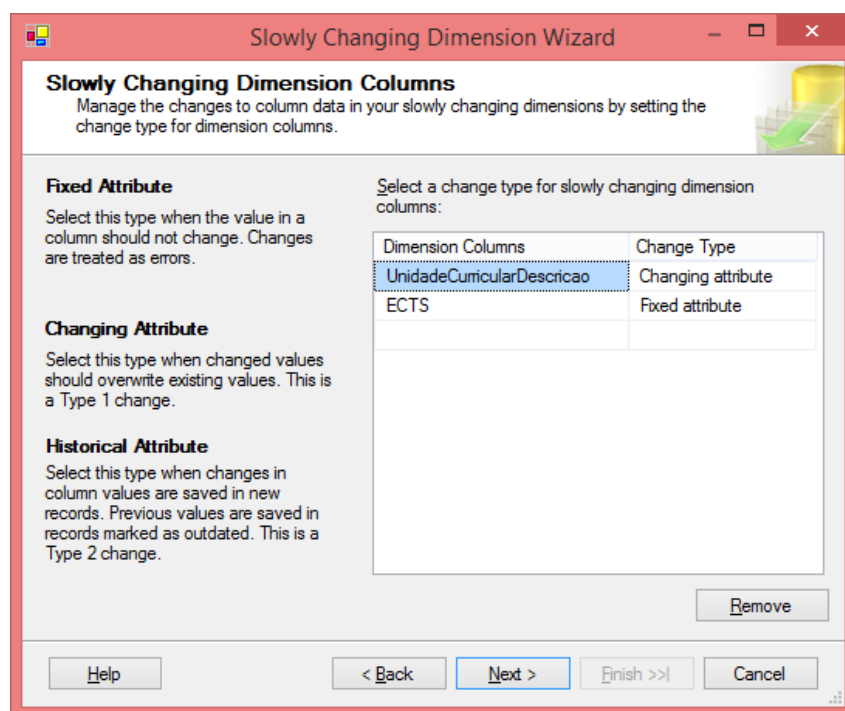


FIGURA 6.31 - Tipos de alterações dos atributos da tabela DimUnidadeCurricular

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimUnidadeCurricular].[UnidadeCurricularChave]
[UnidadeCurricular].[UnidadeCurricularCodigo]	[DimUnidadeCurricular].[UnidadeCurricularCodigo]
[UnidadeCurricular].[Descricao]	[DimUnidadeCurricular].[Descricao]
[UnidadeCurricular].[ECTS]	[DimUnidadeCurricular].[ECTS]

TABELA 6.9 - Tipos de alterações dos atributos da tabela DimUnidadeCurricular

Dimensão Centro Local de Aprendizagem:

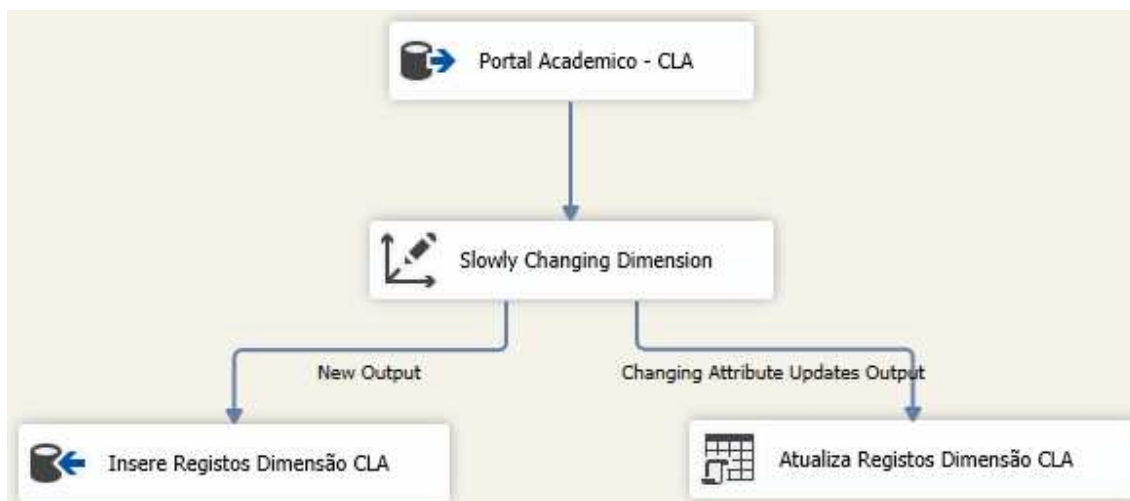


FIGURA 6.32 - Fluxo Centro Local de Aprendizagem

Atributo Local alterável:

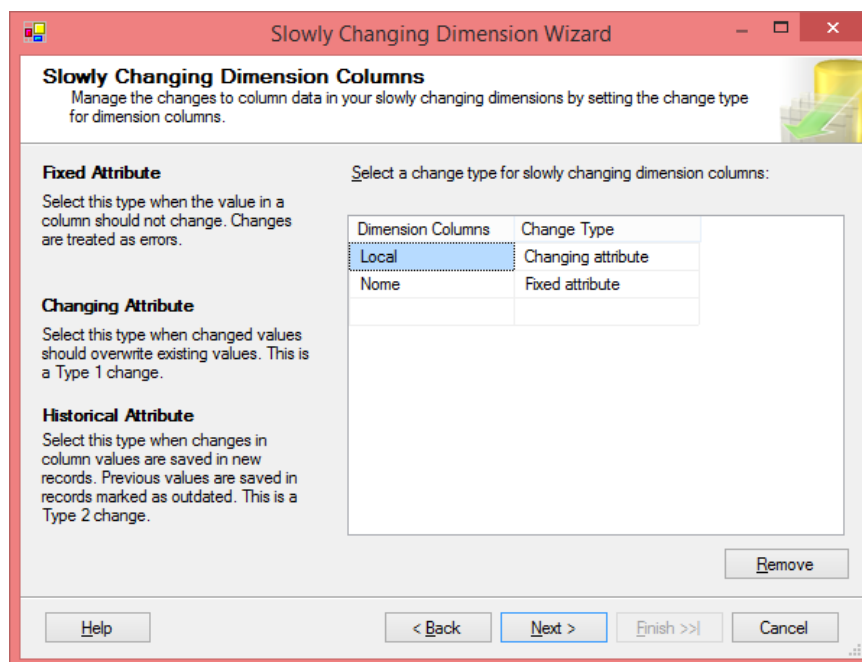


FIGURA 6.33 - Tipos de alterações dos atributos da tabela DimUnidadeCurricular

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimCentroLocalAprendizagem].[CentroLocalAprendizagemChave]
[CentroLocalAprendizagem].[CentroLocalAprendizagemId]	[DimCentroLocalAprendizagem].[CentroLocalAprendizagemId]
[CentroLocalAprendizagem].[Nome]	[DimCentroLocalAprendizagem].[Nome]

[CentroLocalAprendizagem].[Local]

[CentroLocalAprendizagem].[Local]

TABELA 6.10 - Mapeamento Centro Local de Aprendizagem

Dimensão Docente:

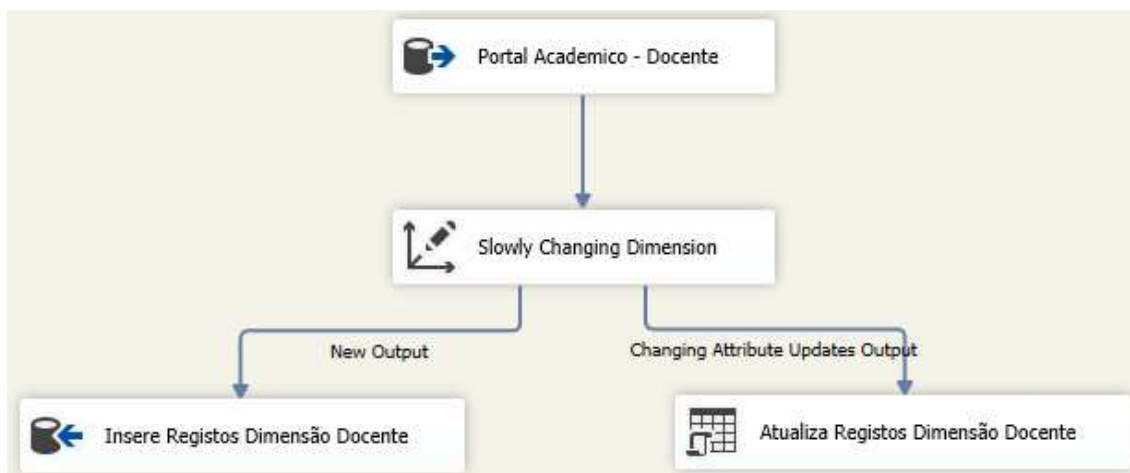


FIGURA 6.34 - Fluxo Docente

Atributo Nome alterável:

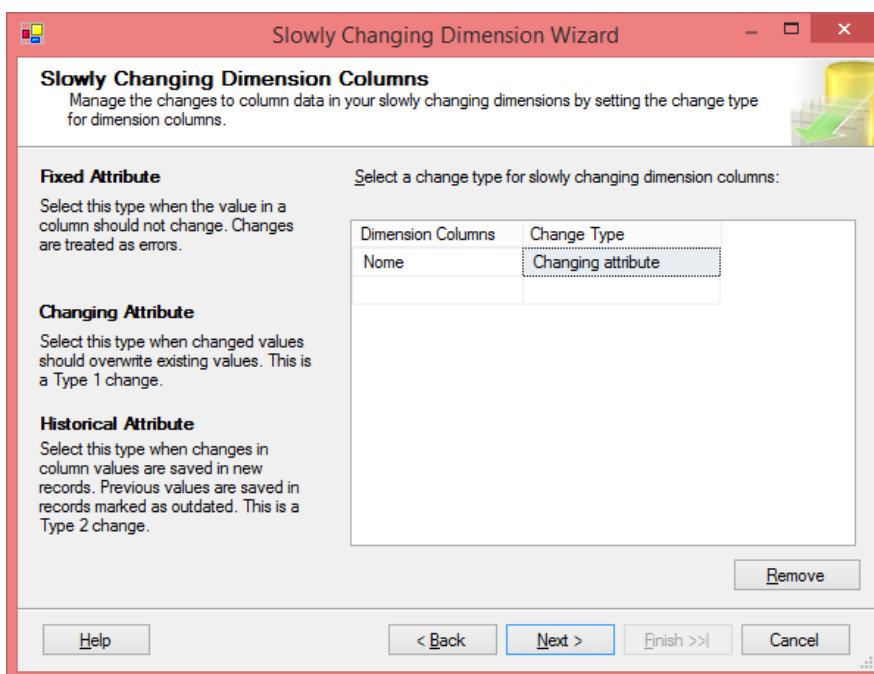


FIGURA 6.35 - Tipos de alterações dos atributos da tabela DimDocente

Atributo da tabela de origem
[UAB-[PortalAcademico]

Atributo da tabela de destino
[UAb-DW]

Não aplicável	[DimDocente].[DocenteChave]
[Docente].[DocenteNumero]	[DimDocente].[DocenteNumero]
[Docente].[Nome]	[DimDocente].[Nome]

TABELA 6.11 - Mapeamento Docente

➤ Container "Comuns"

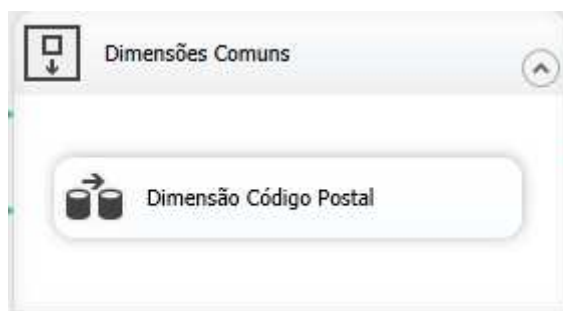


FIGURA 6.36 - Container "Comuns"

Dimensão Código Postal:

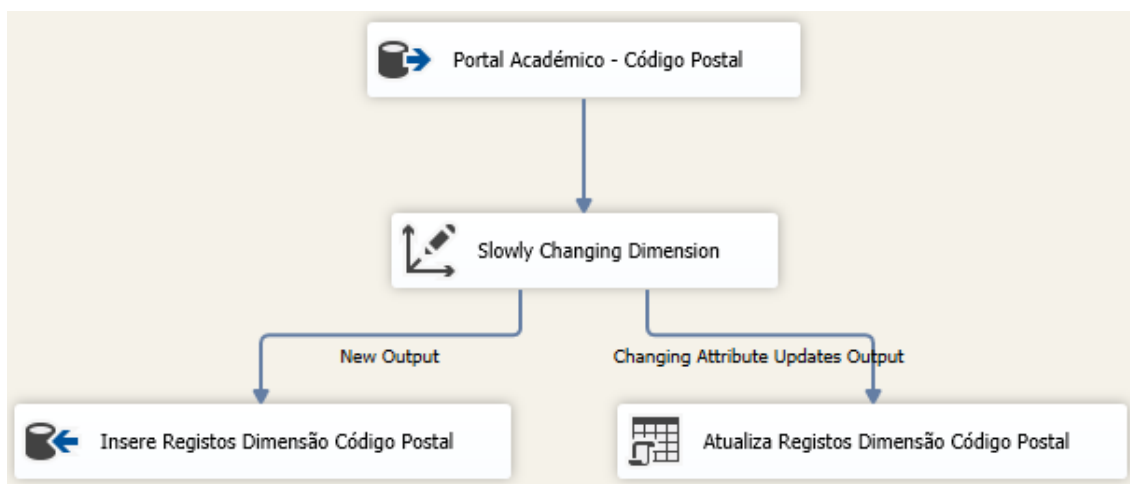


FIGURA 6.37 - Fluxo Código Postal

Atributo Localidade alterável:

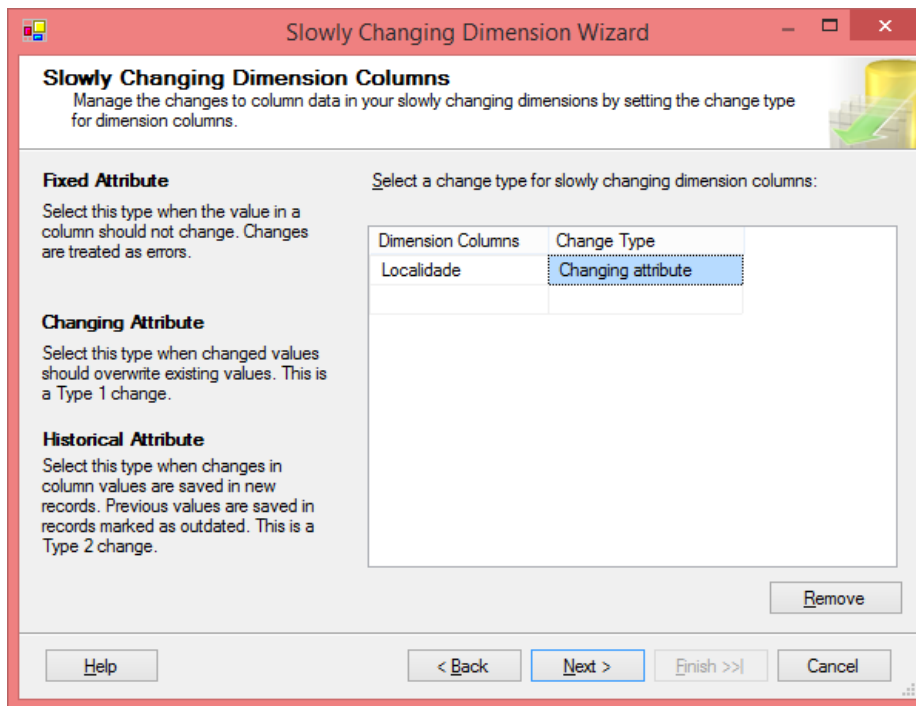


FIGURA 6.38 - Tipos de alterações dos atributos da tabela DimCodigoPostal

Atributo da tabela de origem [UAB-[PortalAcademico]	Atributo da tabela de destino [UAb-DW]
Não aplicável	[DimCodigoPostal].[CodigoPostalChave]
[CodigoPostal].[CodigoPostal]	[DimCodigoPostal].[CodigoPostal]
[CodigoPostal].[Localidade]	[DimCodigoPostal].[Localidade]

TABELA 6.12 - Mapeamento Código Postal

6.4. TABELAS DE FACTOS

Para o processo ETL das tabelas de Factos foi construído um *Container* próprio para a execução primeiro em série depois em paralelo do carregamento das 3 tabelas de factos:

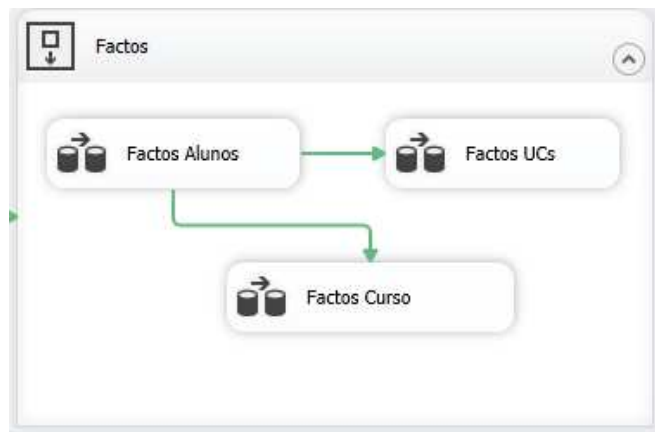


FIGURA 6.39 - Container Factos

Descrevendo em mais detalhe os processos de carregamento:

Tabela de Factos Alunos:

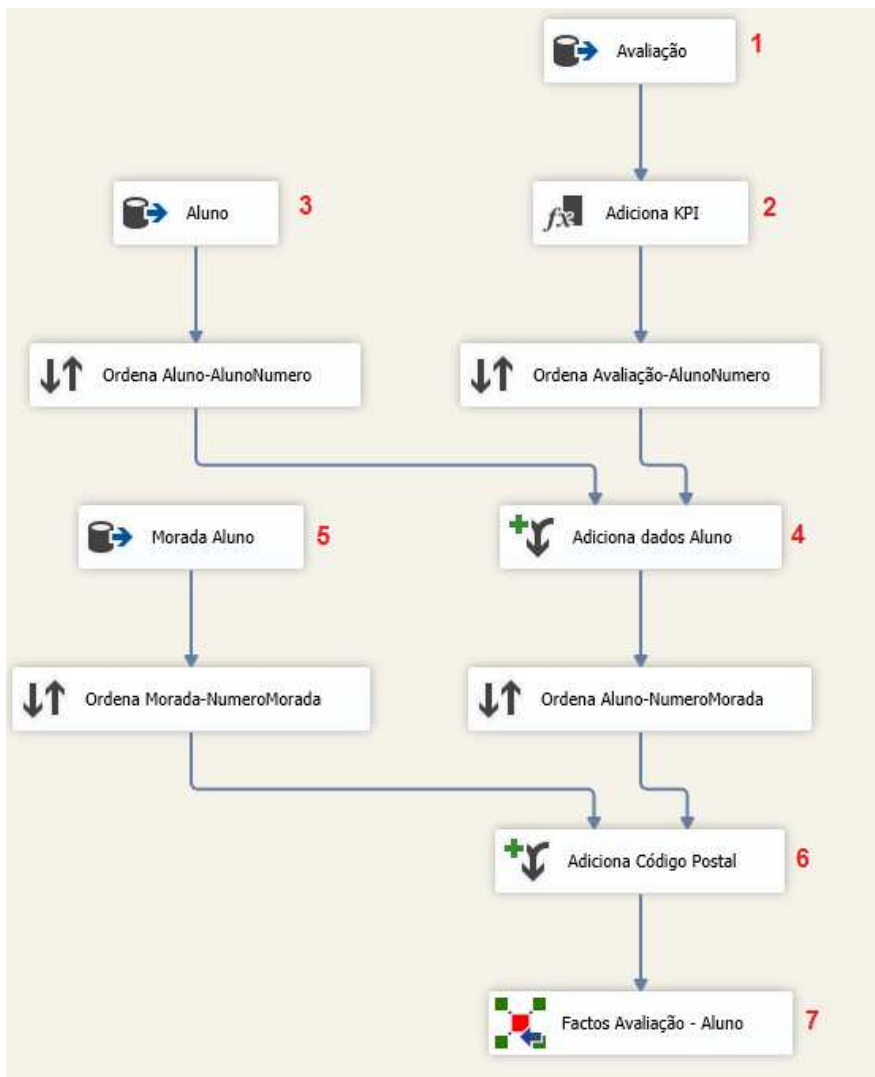


FIGURA 6.40 - Fluxo do processo ETL da tabela de Factos Aluno

(1) Obtidos todos os registos da tabela [PortalAcademico].[AvaliacaoUnidadeCurricular].

Atributos:

- MatriculaInscricaoId
- AlunoNumero
- UnidadeCurricularCodigo
- Estado
- DataEstado
- NotaFinal
- TipoAvaliacao
- AnoLectivo

(2) Analisar o campo Estado e adicionar os campos já preenchidos ('0' ou '1'):

- Abandono
- Insucesso
- Sucesso

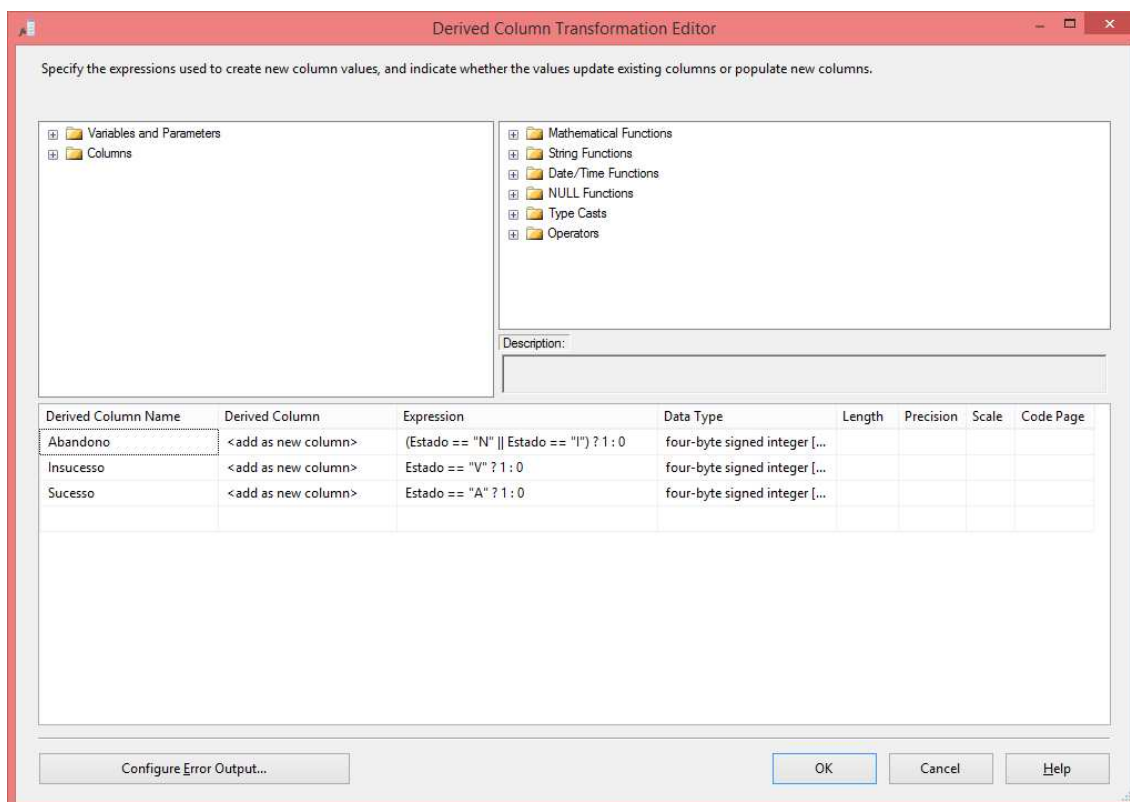


TABELA 6.13 - *Derived Column*: adiciona campos Abandono, Insucesso e Sucesso

(3) Obtidos todos os registos da tabela [PortalAcademico].[Aluno].

Atributos:

- AlunoNumero
- Sexo
- EstadoCivil
- NumeroMorada

(4) Ordenados os registos de (2) e (3) pelo atributo [AlunoNumero], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona dados Aluno]:

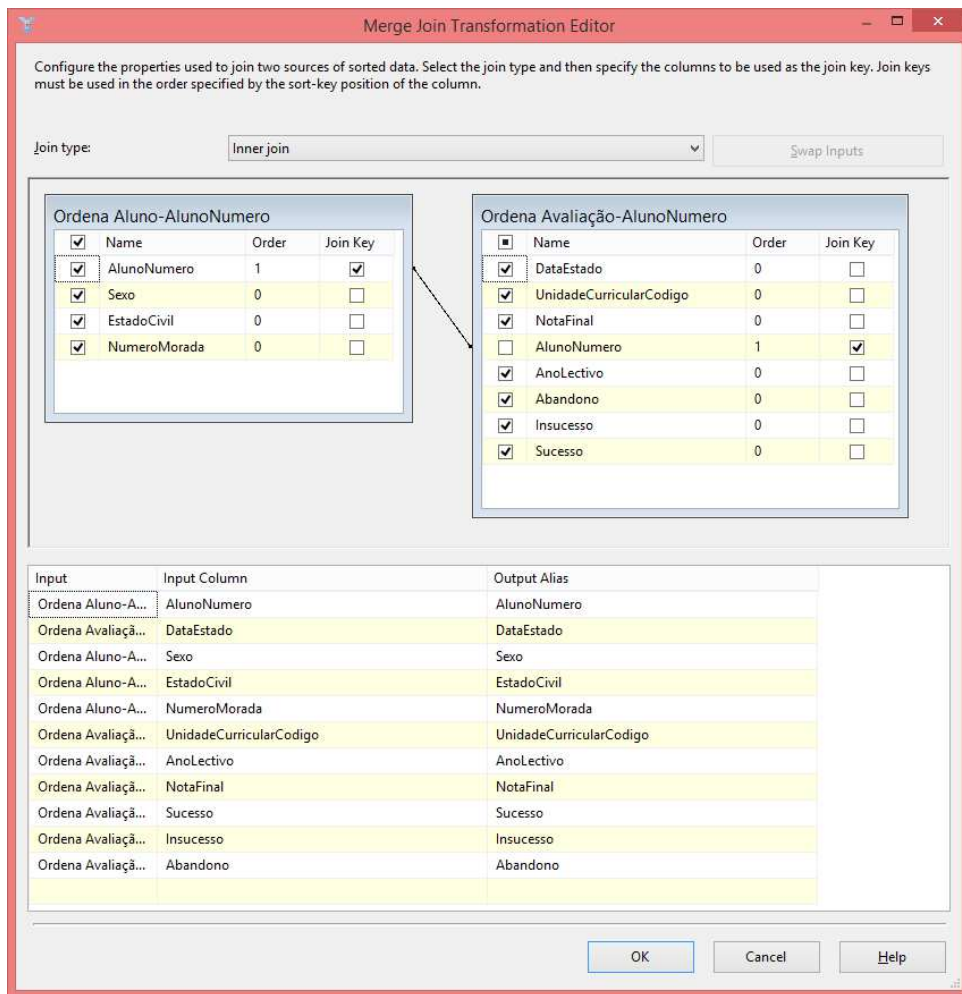


FIGURA 6.41 - Junção das entidades [AvaliacaoUnidadeCurricular] e [Aluno]

- (5) Obtidos todos os registos da tabela [PortalAcademico].[Morada].
Atributos:
- NumeroMorada
- CodigoPostal
- (6) Ordenados os registos de (4) e (5) pelo atributo [NumeroMorada], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona Código Postal]:

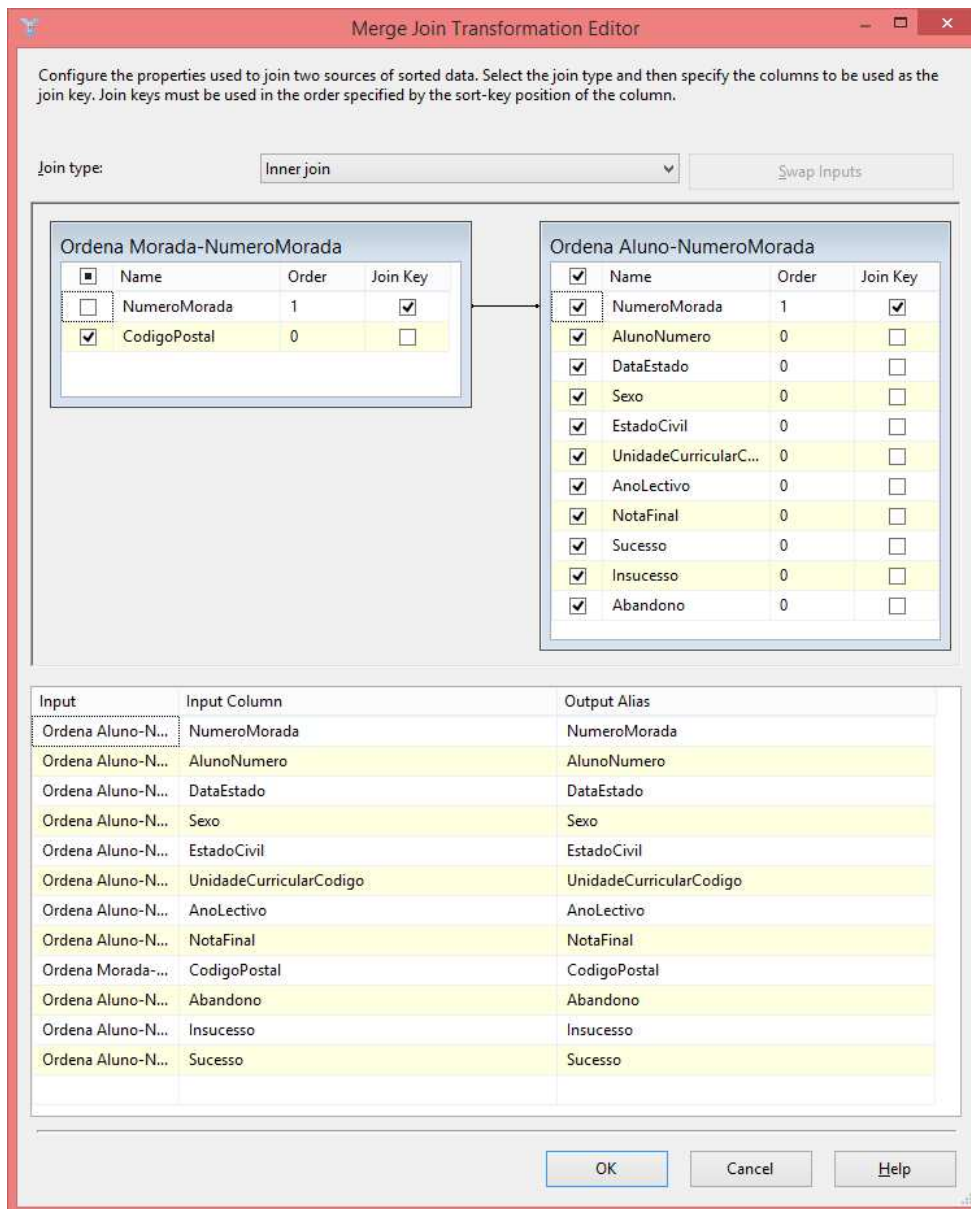


FIGURA 6.42 - Junção do fluxo (4) e da tabela [Morada]

- (7) Neste componente vai ser processado o carregamento da tabela de factos Aluno tendo como origem o fluxo de dados anteriormente processado (6).

Definem-se todos os *lookups (joins)* das tabelas dimensões que existem para o carregamento no componente *Fact Table Destination*:

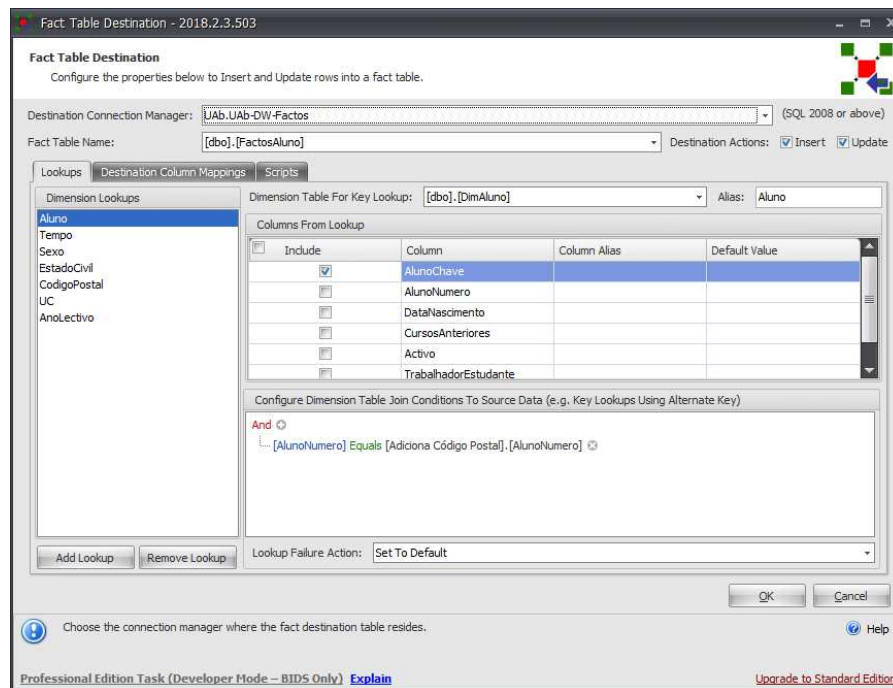


FIGURA 6.43 - Lookups da tabela FactosAluno

[Aluno] -> [AlunoNumero] = fluxo(6).[AlunoNumero]

[Tempo] -> [data] = fluxo(6).[DataEstado]

[Sexo] -> [SexoCodigo] = fluxo(6).[Sexo]

[EstadoCivil] -> [EstadoCivilCodigo] = fluxo(6).[EstadoCivil]

[CodigoPostal] -> [CodigoPostal] = fluxo(6).[CodigoPostal]

[UC] -> [UnidadeCurricularCodigo] = fluxo(6).[UnidadeCurricularCodigo]

[AnoLectivo] -> [AnoLectivoID] = fluxo(6).[AnoLectivo]

Com base nos lookups incluem-se os campos chave das dimensões e definem-se os campos chave e os campos factos da tabela de factos:

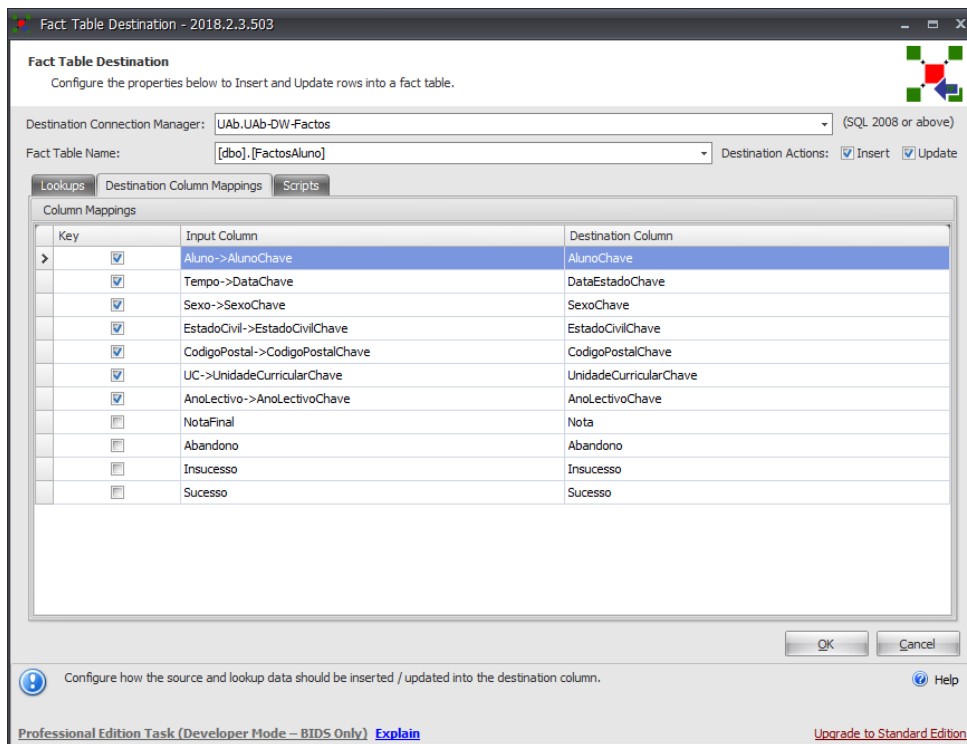


FIGURA 6.44 - Mapeamento das colunas de destino da tabela Factos Aluno

Depois da parametrização feita, o componente fica pronto para ser executado.

Pode-se concluir que com o desenho gráfico do processo e a parametrização do componente de carregamento da tabela de factos conforme identificado em cima, a atividade fica simplificada, comparativamente à criação de código em procedimentos com *joins* entre tabelas dimensões e factos.

Tabela de Factos UC:

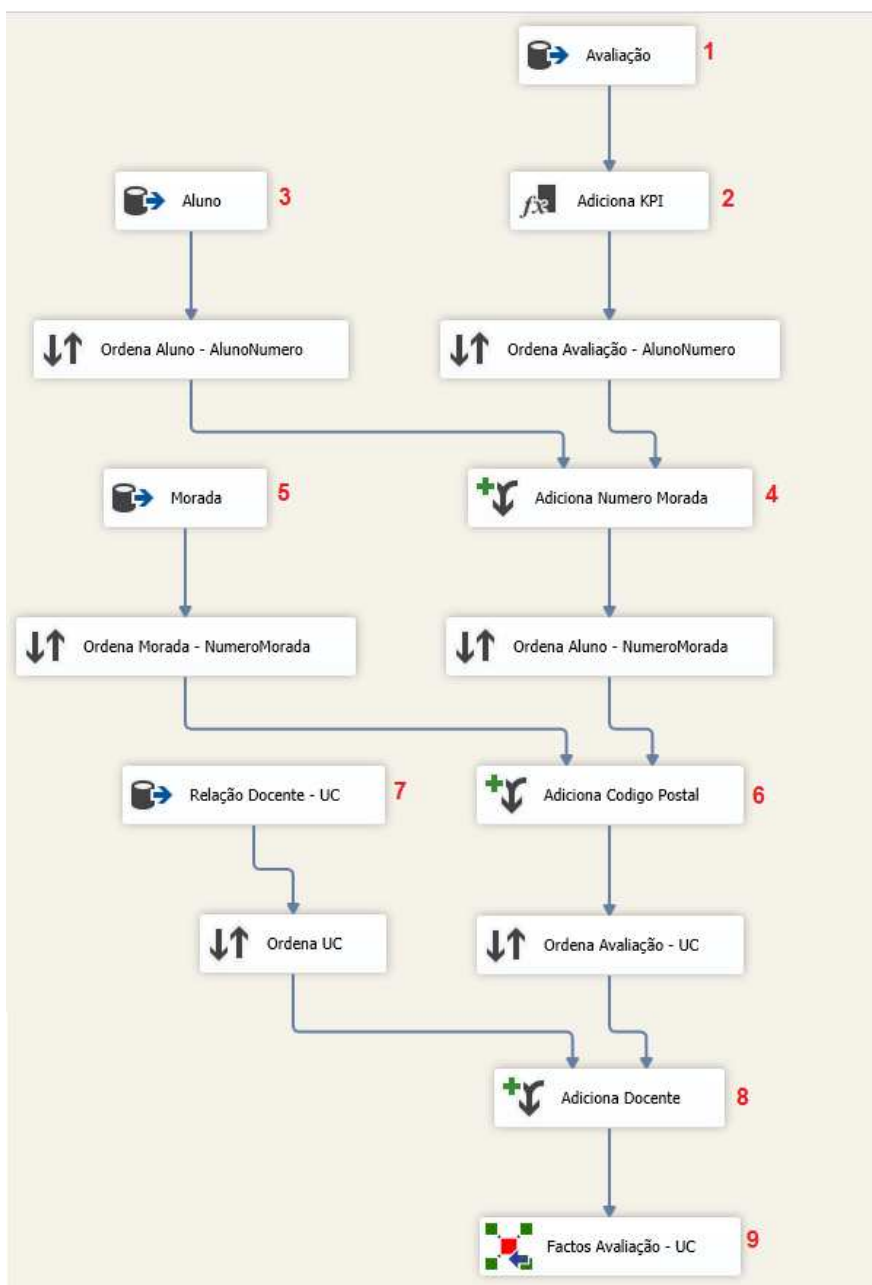


FIGURA 6.45 - Fluxo do processo ETL da tabela de Factos UC

(1) Obtidos todos os registos da tabela [PortalAcademico].[AvaliacaoUnidadeCurricular].

Atributos:

- MatriculaInscricaoId (necessário para criação de chave primária)
- AlunoNumero
- UnidadeCurricularCodigo
- Estado
- DataEstado
- NotaFinal
- TipoAvaliacao
- AnoLectivo

(2) Analisar o campo Estado e adicionar os campos já preenchidos ('0' ou '1'):

- Abandono
- Insucesso
- Sucesso

(3) Obtidos todos os registos da tabela [PortalAcademico].[Aluno].

Atributos:

- NumeroMorada

(4) Ordenados os registos de **(2)** e **(3)** pelo atributo [AlunoNumero], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona Numero Morada].

(5) Obtidos todos os registos da tabela [PortalAcademico].[Morada].

Atributos:

- NumeroMorada
- CodigoPostal

(6) Ordenados os registos de **(4)** e **(5)** pelo atributo [NumeroMorada], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona Codigo Postal].

(7) Obtidos todos os registos da tabela [Portal].[DocenteUnidadeCurricular].

Atributos:

- DocenteNumero
- UnidadeCurricularCodigo

(8) Ordenados os registos de **(6)** e **(7)** pelo atributo [UnidadeCurricularCodigo], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona Docente].

(9) Neste componente vai ser processado o carregamento da tabela de factos UC tendo como origem o fluxo de dados anteriormente processado **(8)**.

Definem-se todos os *lookups (joins)* das tabelas dimensões que existem para o carregamento:

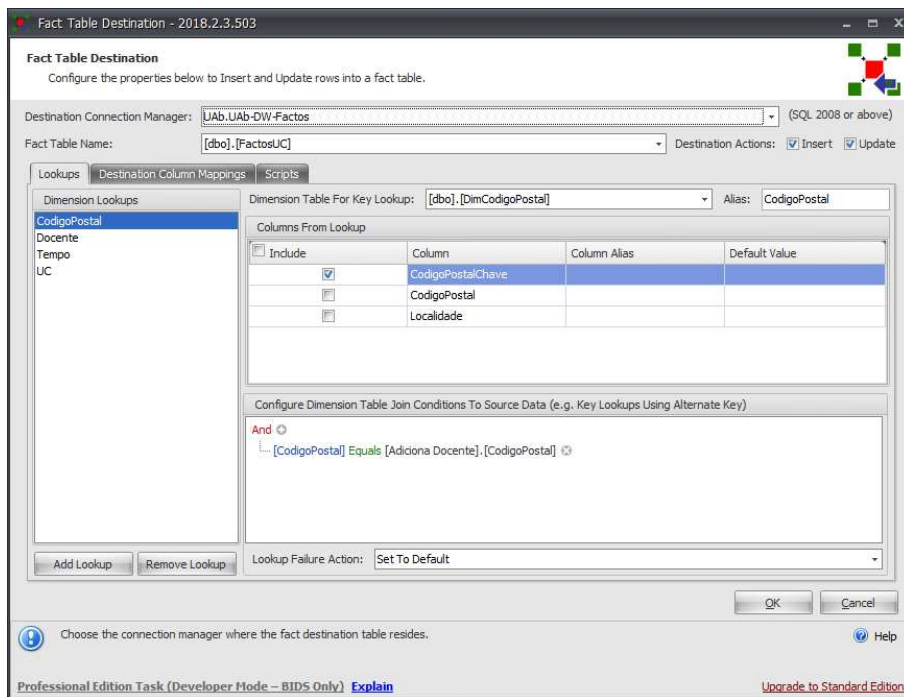


FIGURA 6.46 - Lookups da tabela FactosUC

[CodigoPostal] -> [CodigoPostal] = fluxo(8).[CodigoPostal]

[Docente] -> [DocenteNumero] = fluxo(8).[DocenteNumero]

[Tempo] -> [data] = fluxo(8).[DataEstado]

[UC] -> [UnidadeCurricularCodigo] = fluxo(8).[UnidadeCurricularCodigo]

Com base nos lookups incluem-se os campos chave das dimensões e definem-se os campos chave e os campos factos da tabela de factos:

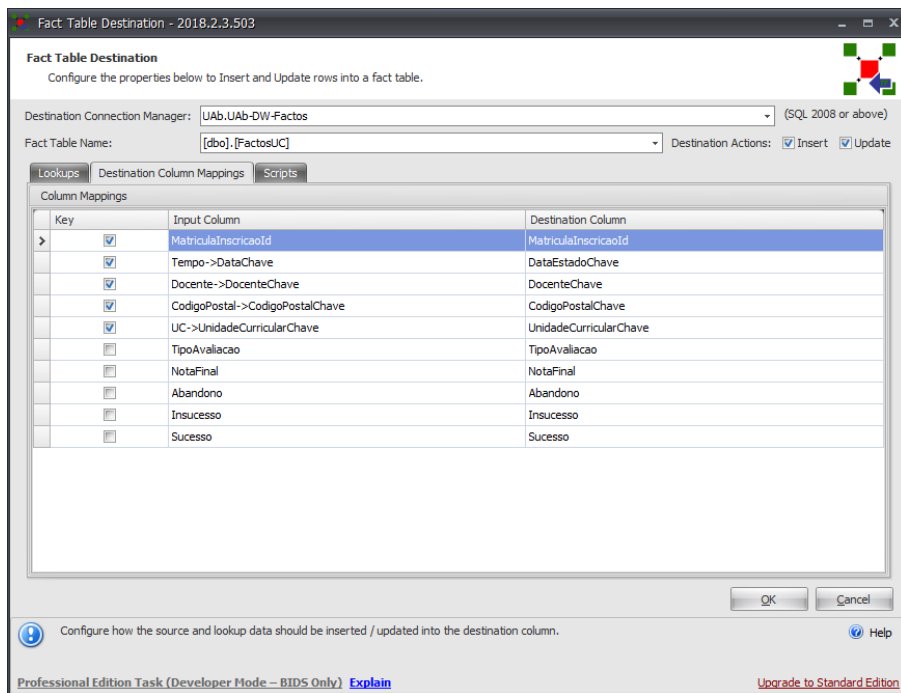


FIGURA 6.47 - Mapeamento das colunas de destino da tabela Factos UC

Tabela de Factos Curso:

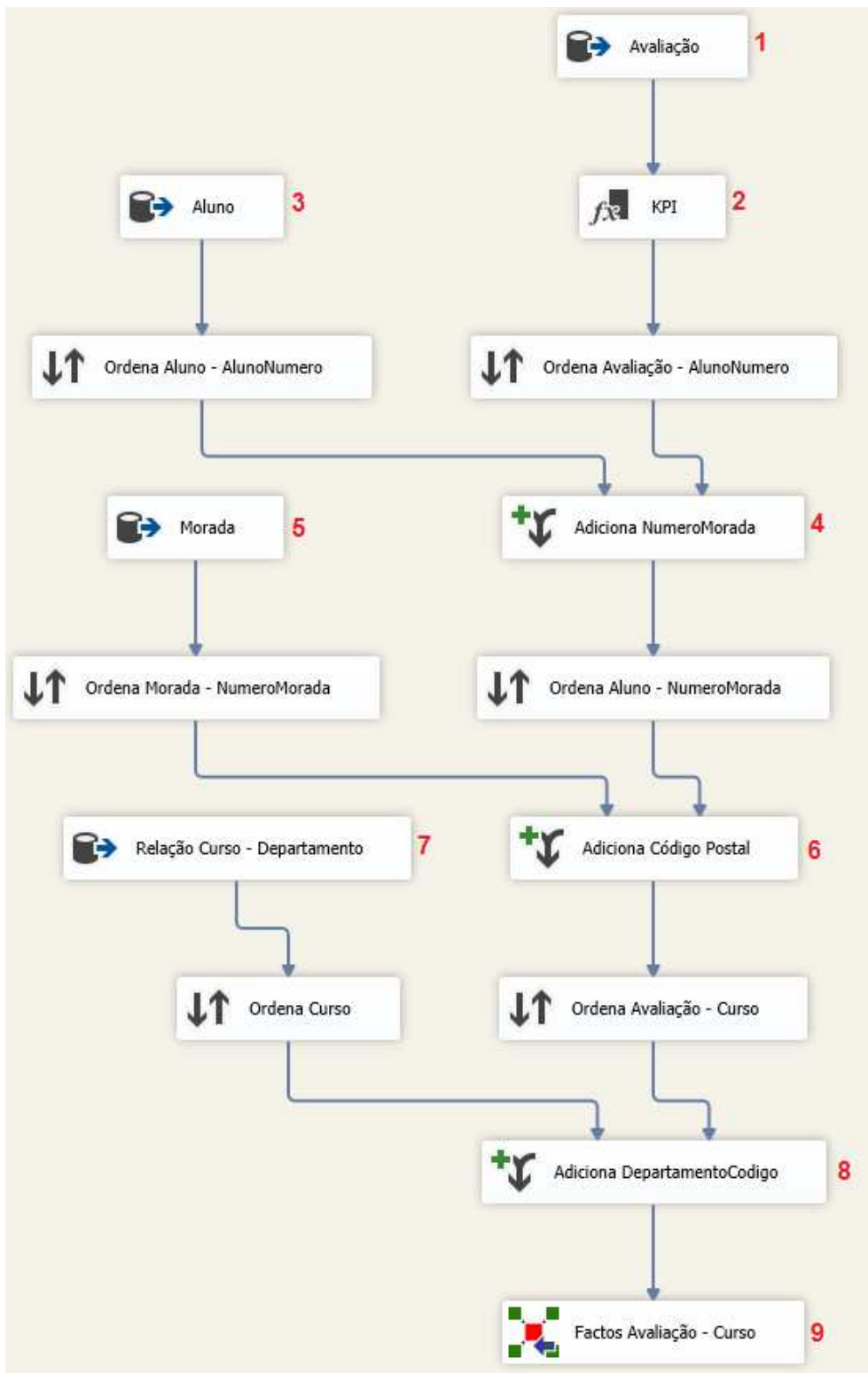


FIGURA 6.48 - Fluxo do processo ETL da tabela de Factos Curso

- (1) Obtidos todos os registos da tabela [PortalAcademico].[AvaliacaoUnidadeCurricular].
 Atributos:
 - MatriculaInscricaoId (necessário para criação de chave primária)
 - AlunoNumero
 - UnidadeCurricularCodigo

- Estado
- DataEstado
- NotaFinal
- AnoLectivo
- CursoID

- (2)** Analisar o campo Estado e adicionar os campos já preenchidos ('0' ou '1'):
- Abandono
 - Insucesso
 - Sucesso
- (3)** Obtidos todos os registos da tabela [PortalAcademico].[Aluno].
Atributos:
- NumeroMorada
- (4)** Ordenados os registos de **(2)** e **(3)** pelo atributo [AlunoNumero], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona Numero Morada].
- (5)** Obtidos todos os registos da tabela [PortalAcademico].[Morada].
Atributos:
- NumeroMorada
 - CodigoPostal
- (6)** Ordenados os registos de **(4)** e **(5)** pelo atributo [NumeroMorada], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona Codigo Postal].
- (7)** Obtidos todos os registos da tabela [Portal].[RelDepartamentoCurso].
Atributos:
- CursoID
 - DepartamentoCodigo
- (8)** Ordenados os registos de **(6)** e **(7)** pelo atributo [CursoID], o fluxo de dados fica constituído pela junção dos 2 fluxos de dados [Adiciona DepartamentoCodigo]:
- (9)** Neste componente vai ser processado o carregamento da tabela de factos Curso tendo como origem o fluxo de dados anteriormente processado **(8)**.

Definem-se todos os *lookups (joins)* das tabelas dimensões que existem para o carregamento:

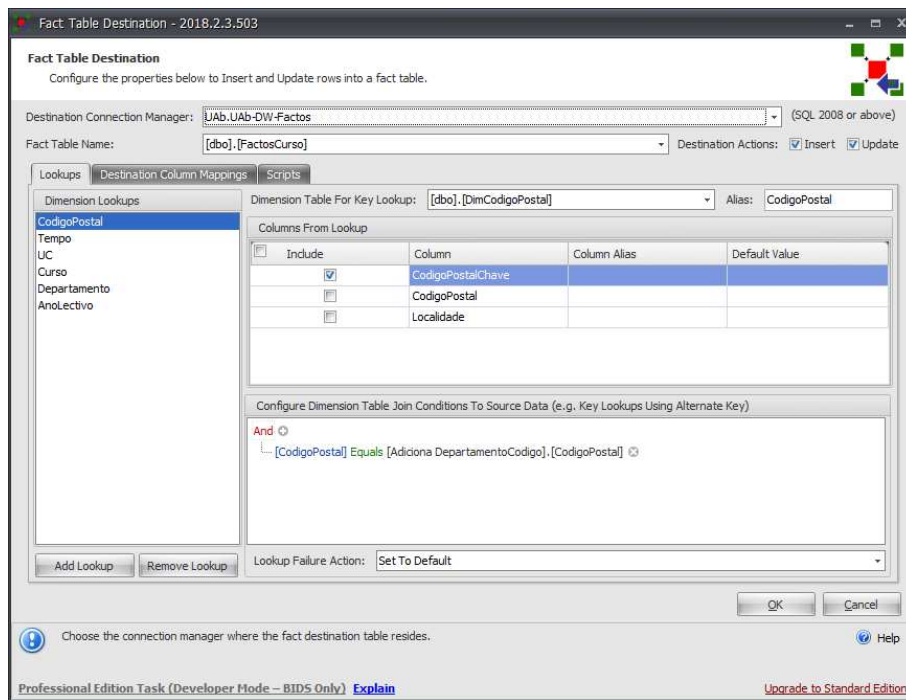


FIGURA 6.49 - Lookups da tabela FactosCurso

[CodigoPostal] -> [CodigoPostal] = fluxo(8).[CodigoPostal]

[Tempo] -> [data] = fluxo(8).[DataEstado]

[UC] -> [UnidadeCurricularCodigo] = fluxo(8).[UnidadeCurricularCodigo]

[Curso] -> [CursoID] = fluxo(8).[CursoID]

[Departamento] -> [DepartamentoCodigo] = fluxo(8).[DepartamentoCodigo]

[AnoLectivo] -> [AnoLectivoID] = fluxo(8).[AnoLectivo]

Com base nos lookups incluem-se os campos chave das dimensões e definem-se os campos chave e os campos factos da tabela de factos:

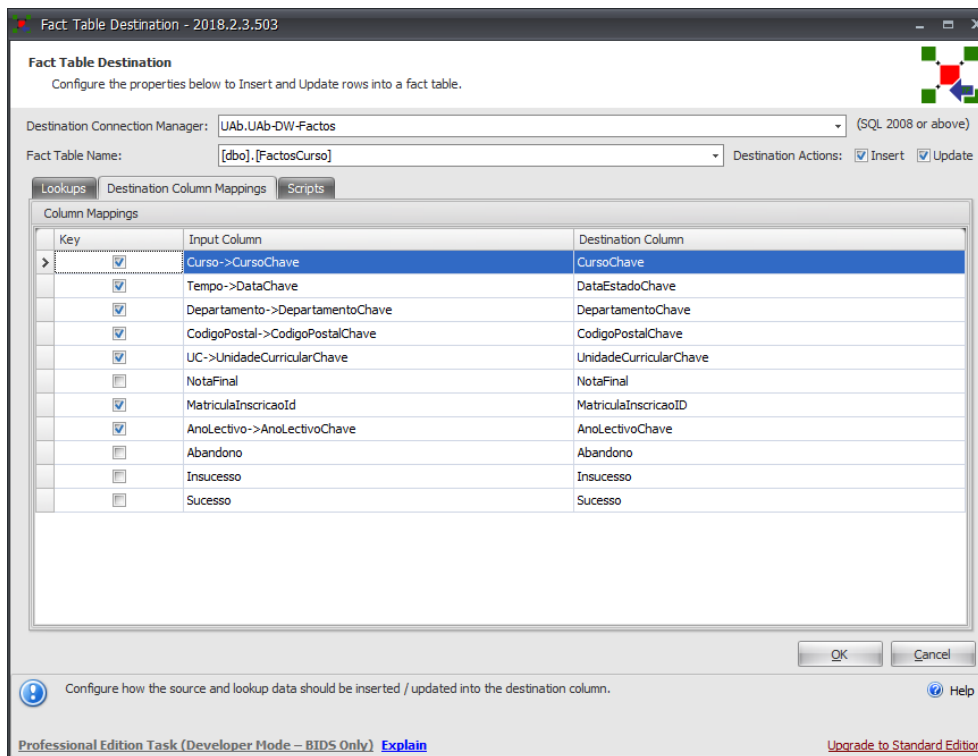


FIGURA 6.50 - Mapeamento das colunas de destino da tabela Factos Curso

7. BUSINESS INTELLIGENCE

7.1. INTRODUÇÃO

Neste capítulo serão apresentados os relatórios criados com a informação extraída do DW. Para o efeito foi utilizada a ferramenta *Power BI Desktop* versão de maio de 2018.

Para a resolução dos problemas de análise de dados e de cálculo básico, nomeadamente ao nível da criação das métricas adaptadas ao *Power BI* foi utilizada a linguagem DAX (Data Analysis Expressions).

A linguagem DAX ajuda a criar nova informação a partir de dados já existentes no modelo do DW.

7.2. PREPARAÇÃO INICIAL

No *Power BI* existe necessidade de ligação à origem de dados, e a ferramenta tem um leque muito variado de possibilidades de fontes de dados, desde simples documentos de texto, *MS Access*, *MS Excel*, *Web*, etc, até motores de bases de dados do tipo *SQL Server*, *Oracle*, *DB2*, *Sybase*, etc.

No caso em concreto deste projeto foi feita uma ligação ao SQL Server do nosso DW com as credenciais do sistema operativo:

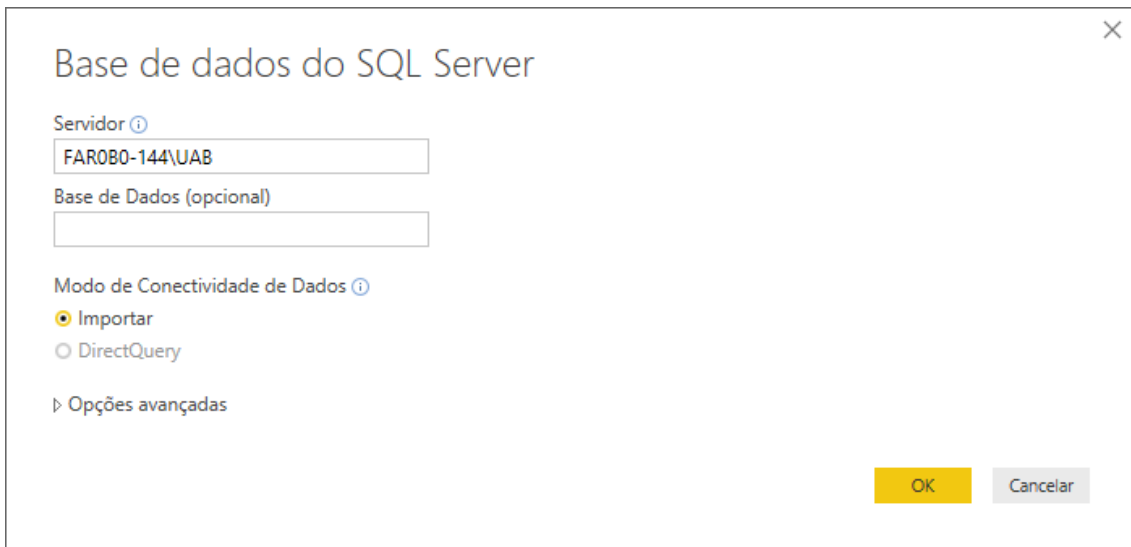


FIGURA 7.1 - Ligação a origem de dados: Servidor

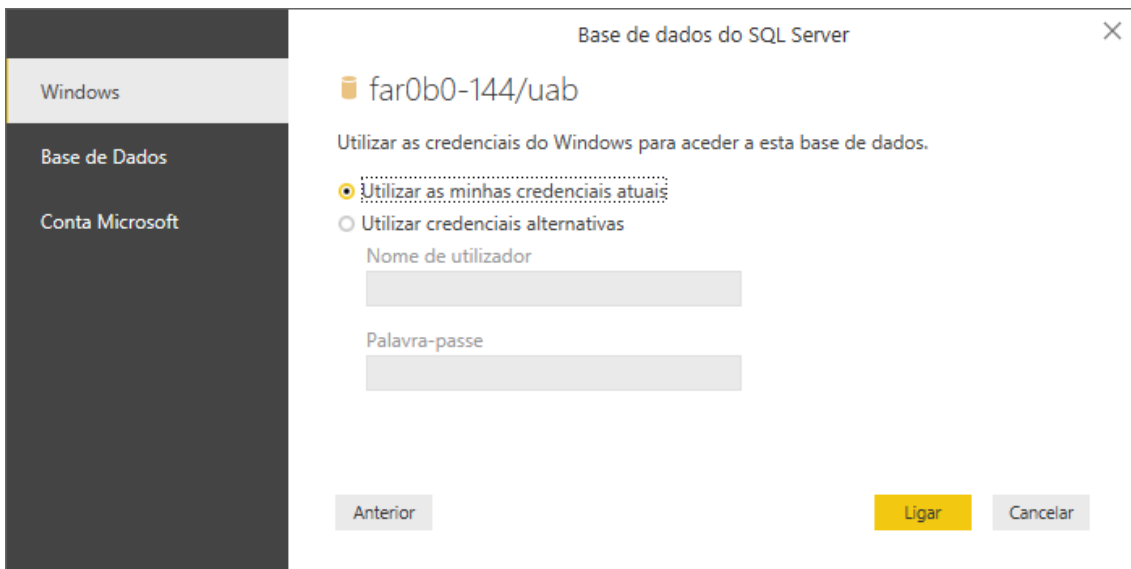


FIGURA 7.2 - Ligação a origem de dados: Credenciais

Após sucesso na ligação com o servidor de base de dados, o próximo passo é a seleção da BD onde está hospedado o DW, no caso em concreto [UAb-DW], que possibilitará a escolha das tabelas necessárias para trabalhar os relatórios.

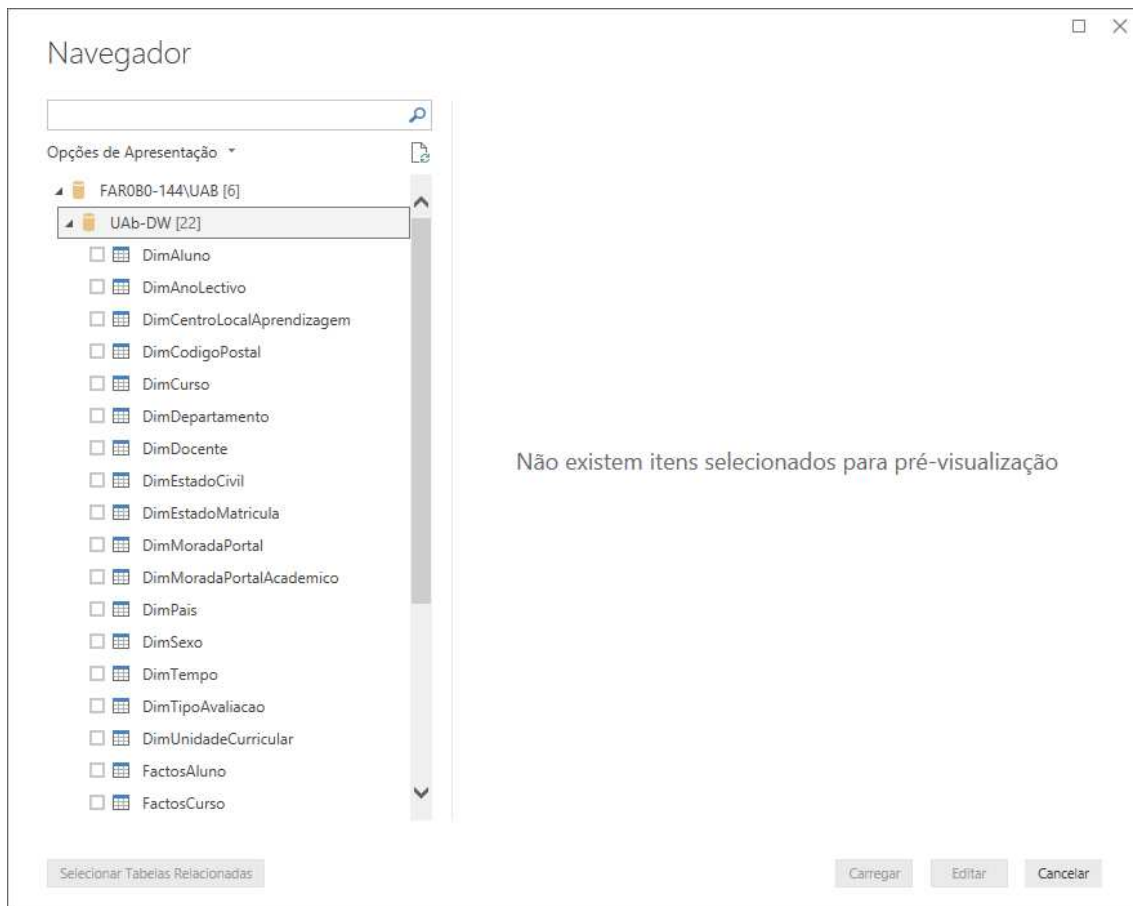


FIGURA 7.3 - Ligação a origem de dados: Base de Dados

Os relatórios criados foram organizados em conformidade com as estrelas criadas, isto é, 3 Dashboards: Aluno, Curso e UC. Em cada Dashboard foram criadas páginas, e em cada página foram criados os gráficos correspondentes às questões que nos propusemos responder no início deste documento.

7.3. MÉTRICAS (DAX)

Para a criação das métricas identificadas no capítulo 2.2 (relembradas em baixo):

$$taxa\ abandono\ (dropout) = 1 - \frac{Insucesso + Sucesso}{Abandono + Insucesso + Sucesso = Todos}$$

$$taxa\ reprovação\ (failure) = 1 - \frac{Sucesso}{Insucesso + Sucesso}$$

$$taxa\ sucesso = \frac{Sucesso}{Abandono + Insucesso + Sucesso = Todos}$$

Foram criadas para cada estrela a respetiva versão em DAX para auxiliar na criação dos relatórios. Em baixo identificam-se as fórmulas criadas para a estrela Aluno, sendo

que para as outras 2 estrelas o princípio é o mesmo, apenas alterando as tabelas de factos.

Relembra-se também que as estrelas contém os atributos [Abandono], [Sucesso] e [Insucesso], correspondentes aos kpi e são do tipo booleano e que estão identificadas nas fórmulas em baixo:

Taxa Abandono =

```
1 - (
    DIVIDE(
        CALCULATE(COUNTROWS(FactosAluno);FactosAluno[InSucesso])
        +
        CALCULATE(COUNTROWS(FactosAluno);FactosAluno[Sucesso])
    ;
        CALCULATE(COUNTROWS(FactosAluno))
    )
)
```

Taxa Reprovação =

```
1 - (
    DIVIDE(
        CALCULATE(COUNTROWS(FactosAluno);FactosAluno[Sucesso])
    ;
        CALCULATE(COUNTROWS(FactosAluno);FactosAluno[Insucesso])
        +
        CALCULATE(COUNTROWS(FactosAluno);FactosAluno[Sucesso])
    )
)
```

Taxa Sucesso =

```
DIVIDE(
    CALCULATE(COUNTROWS(FactosAluno);FactosAluno[Sucesso])
;
    CALCULATE(COUNTROWS(FactosAluno))
)
```

Significado das funções DAX:

- **CALCULATE(<expressão>,<filtro1>,<filtro2>...)**

expressão: A expressão a avaliar

filtro1,filtro2,...: (opcional) Uma lista de expressões Booleanas separadas por ponto e vírgula ou uma expressão de tabela que define um filtro.

Valor devolvido: o valor que é o resultado da expressão.

É uma função que avalia uma expressão no contexto modificado pelos filtros presentes no relatório. Por exemplo, se filtrar no relatório para um determinado ano, a função avalia apenas os dados com base no filtro do ano selecionado.

Por exemplo, no caso em concreto na métrica [Taxa Sucesso], a função CALCULATE tem como expressão contar registos da tabela [FactosAlunos], e tem como filtro (filtro1) para a contagem, o campo FactosAluno[Sucesso] = 'Verdadeiro'.

- **COUNTROWS(<tabela>)**

tabela: O nome da tabela que contém as linhas a serem contadas ou uma expressão que retorna uma tabela.

Valor devolvido: um número inteiro.

A função COUNTROWS conta o número de linhas na tabela especificada ou em uma tabela definida por uma expressão.

Por exemplo, no caso em concreto na métrica [Taxa Sucesso], a função COUNTROWS dentro da função CALCULATE, conta o número de linhas da tabela [FactosAlunos].

- **DIVIDE(<numerador>, <denominador> [,<resultado alternativo>])**

numerador: o dividendo ou o número a ser dividido.

denominador: o divisor ou o número pelo qual dividir.

resultado alternativo: (opcional) o valor retornado quando a divisão por zero resulta em um erro. Quando não fornecido, o valor padrão é BLANK().

A função DIVIDE efetua a divisão e retorna um resultado alternativo ou BLANK() na divisão por zero.

Por exemplo, no caso em concreto na métrica [Taxa Sucesso], a função DIVIDE, faz a operação de divisão do numerador "registos marcados como 'Verdadeiro' no atributo [Sucesso]" e o denominador "todos os registos da tabela [FactosAluno]".

7.4. ALUNO

Para o *Dashboard* Aluno foram criadas 5 páginas:

1. Geral: contém uma visão geral de Taxa de Sucesso, Taxa de Abandono e Taxa de Reprovação ao longo dos anos 2011 até 2018.
Responde às questões: Taxa de Sucesso e Abandono global.
2. Mapa: funcionalidade do Power BI, que por intermédio do motor de pesquisa 'Bing' e o campo Localidade (precedido do prefixo Portugal) consegue

identificar no mapa a localização. Nesta página foi colocado um filtro dinâmico por Distrito para identificar no mapa apenas os distritos que o utilizador quer ver representado.

Responde às questões: "Taxa de Sucesso e Abandono distribuída geograficamente e distância entre localidade da morada do Aluno e da localidade das CLAs".

3. Taxas por [Ano] – Distrito: distribuição das Taxas de Abandono e Sucesso por distrito. Contém uma matriz e um gráfico de barras com a distribuição por distrito. Possibilidade de filtrar por ano. Contém também um cartão com os indicadores globais das 2 taxas para fazer a comparação entre o global e as taxas detalhadas dos distritos.

Responde às questões: "Taxa de Sucesso e Abandono global distribuída por distrito".

4. Taxas por Ano - [Distrito]: gráfico de linhas com as taxas distribuídas por ano. Existe um filtro por distrito que possibilita a visualização das taxas distribuídas por ano e pelos filtros selecionados.

Responde às questões: "Taxa de Sucesso e Abandono global distribuída por distrito".

5. Vários Indicadores: vários gráficos, barras e linhas, correspondendo a vários indicadores.

Responde às questões: "Distribuição de Sucesso e Abandono por género", "estado civil", "nº de cursos anteriores", "idade" e "estatuto de trabalhador estudante".

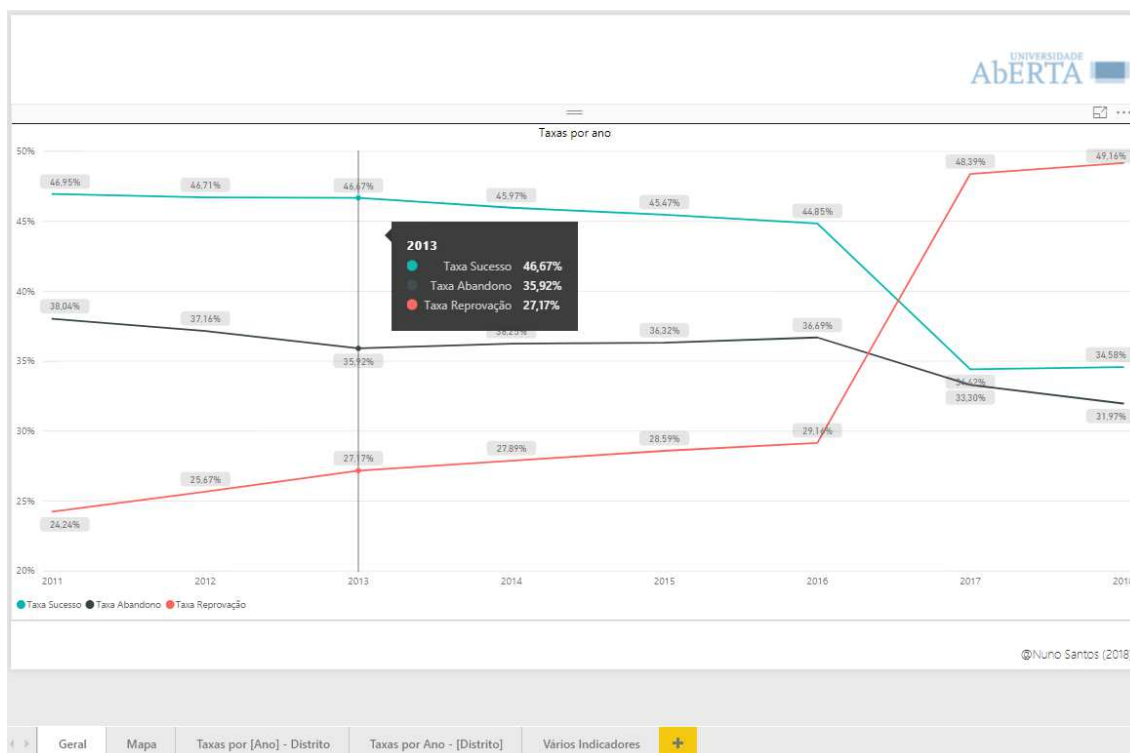


FIGURA 7.4 - Dashboard Aluno: página Geral

Para criar este relatório temos disponíveis 2 painéis de ferramentas: um painel dos parâmetros da Visualização selecionada (gráfico, matriz, etc) e outro painel com as tabelas/atributos do DW.

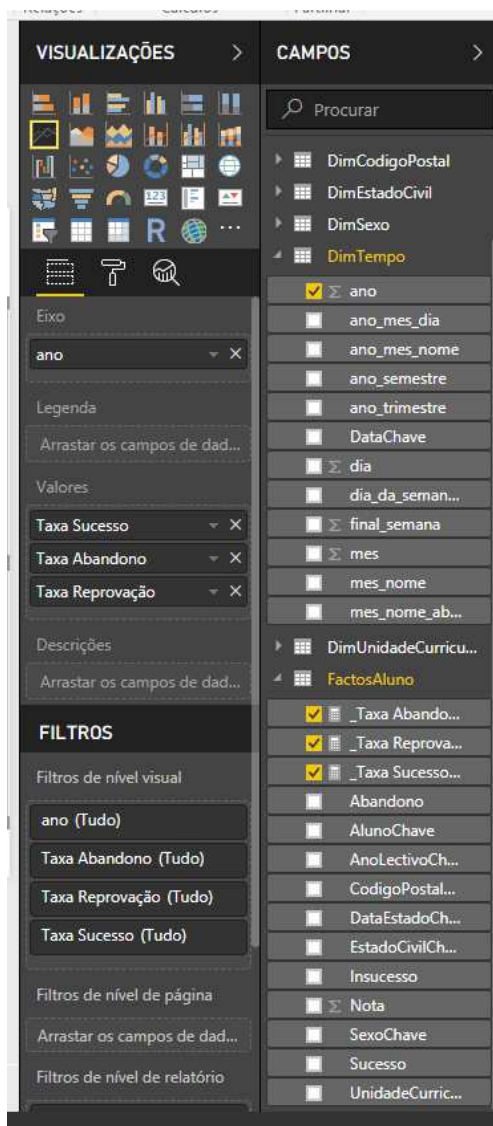


FIGURA 7.5 - Painel de ferramentas

Para a construção deste gráfico foi escolhido o visual “Gráfico de Linhas” colocando o atributo Tempo[Ano] no eixo X, os valores [Taxa Abandono], [Taxa Insucesso] e [Taxa Sucesso]. Ao nível dos filtros de relatório foram adicionados apenas os anos em análise [2011,2018]:

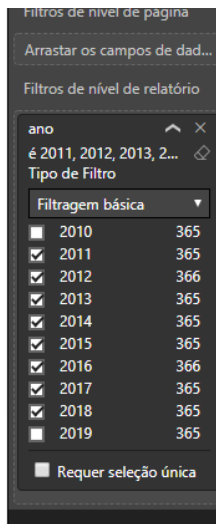


FIGURA 7.6 - Filtro dos Anos em análise

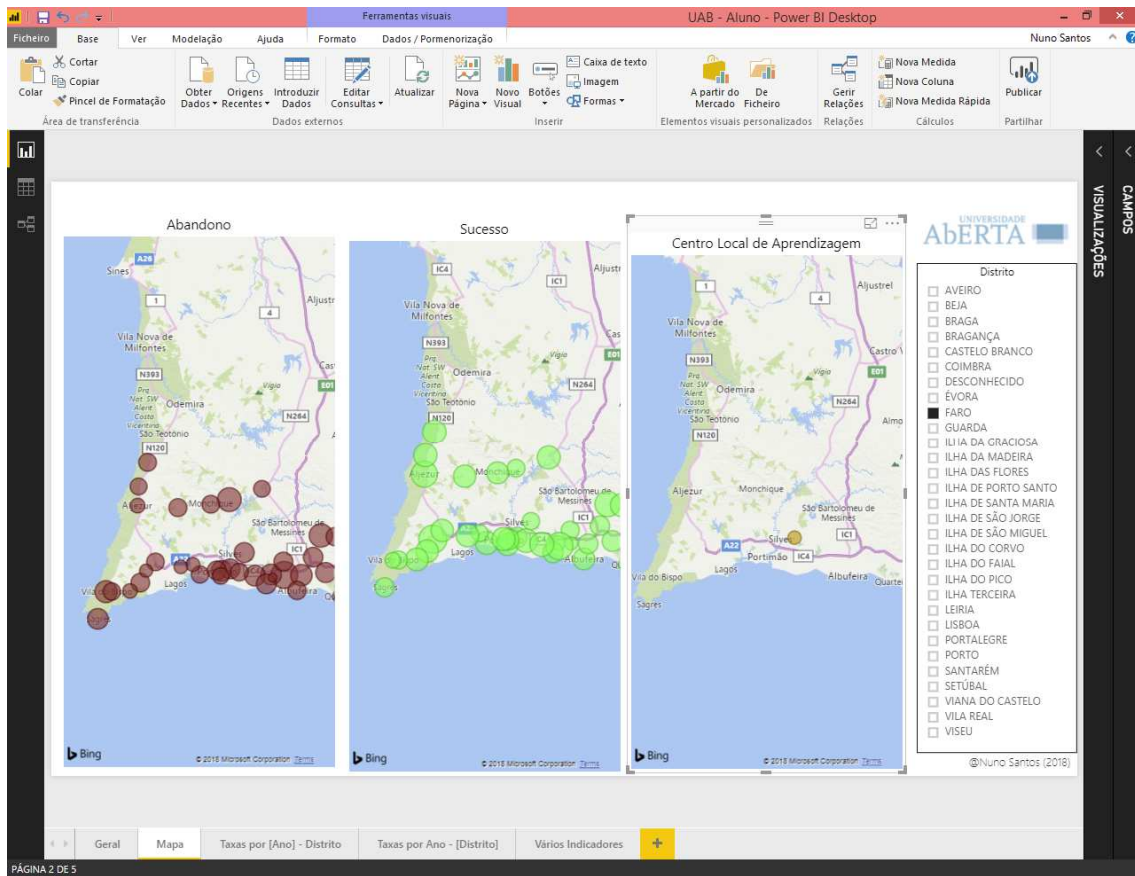


FIGURA 7.7 - Mapa

Nesta página foi utilizada a visualização geográfica 'Mapa', recorrendo a uma coluna criada no Power BI e consiste na adição do prefixo Portugal ao atributo [Localidade] da tabela [CodigoPostal], para que a descodificação da localidade no visual seja mais orientada. Não sendo uma técnica 100% exata, os resultados são suficientemente apurados para garantir a consistências dos dados. Podemos, com o filtro do lado direito, filtrar os distritos que queremos analisar.

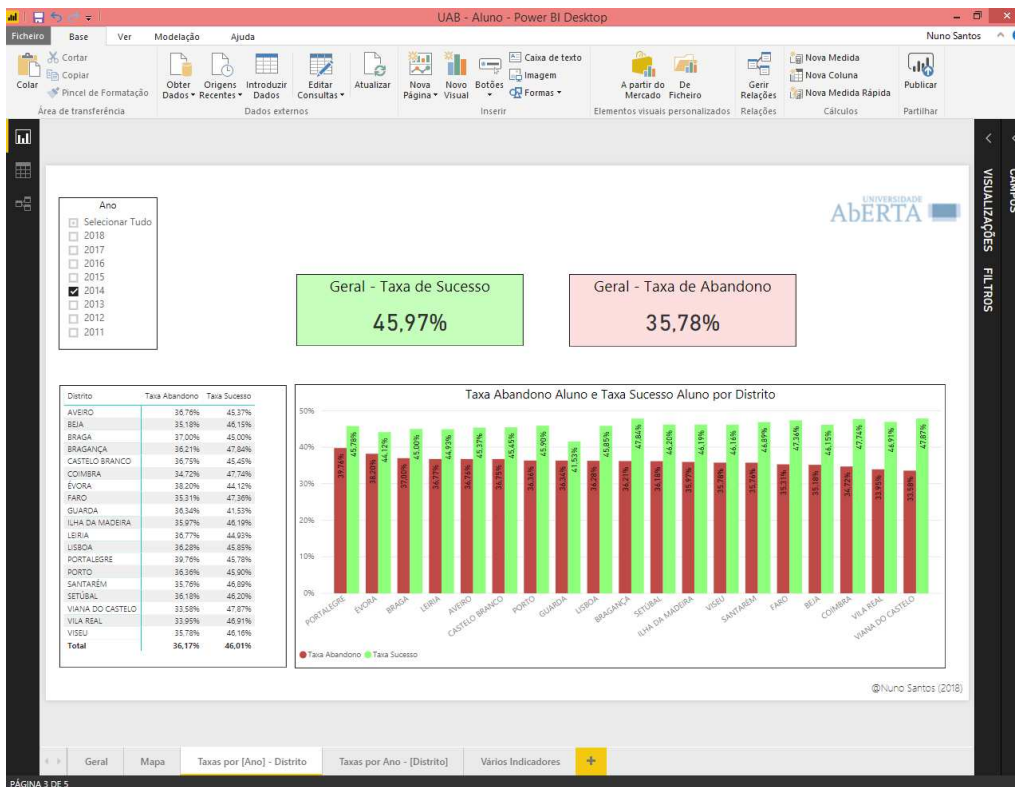


FIGURA 7.8- TAXAS POR [ANO] - DISTRITO

Nesta página foram utilizados os visuais Matriz e gráfico de barras para a visualização dos dados por distrito. Tem disponível um filtro por ano, caso seja necessário analisar um ou mais anos em detalhe. Em cima estão presentes 2 cartões com valores agregados das Taxas.

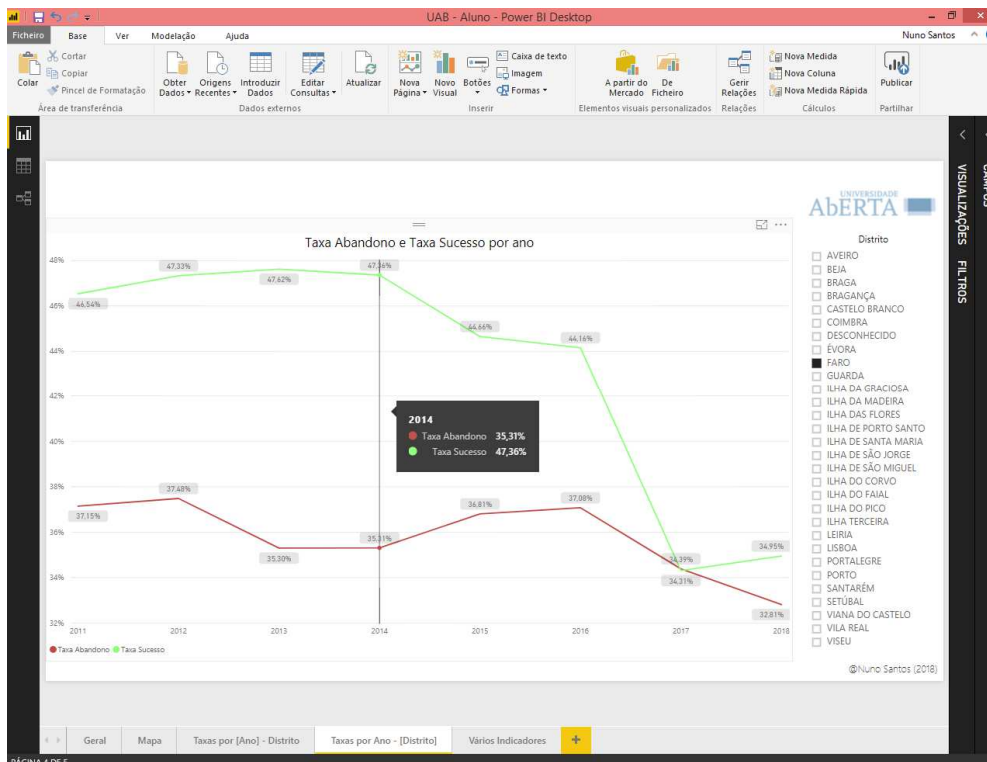


FIGURA 7.9 - TAXAS POR ANO - [DISTRITO]

Nesta página as Taxas de Abandono e Sucesso Escolar estão dispostas por Ano num gráfico de linhas. À direita existe um filtro por distrito para analisar os dados por um ou mais distritos em detalhe.

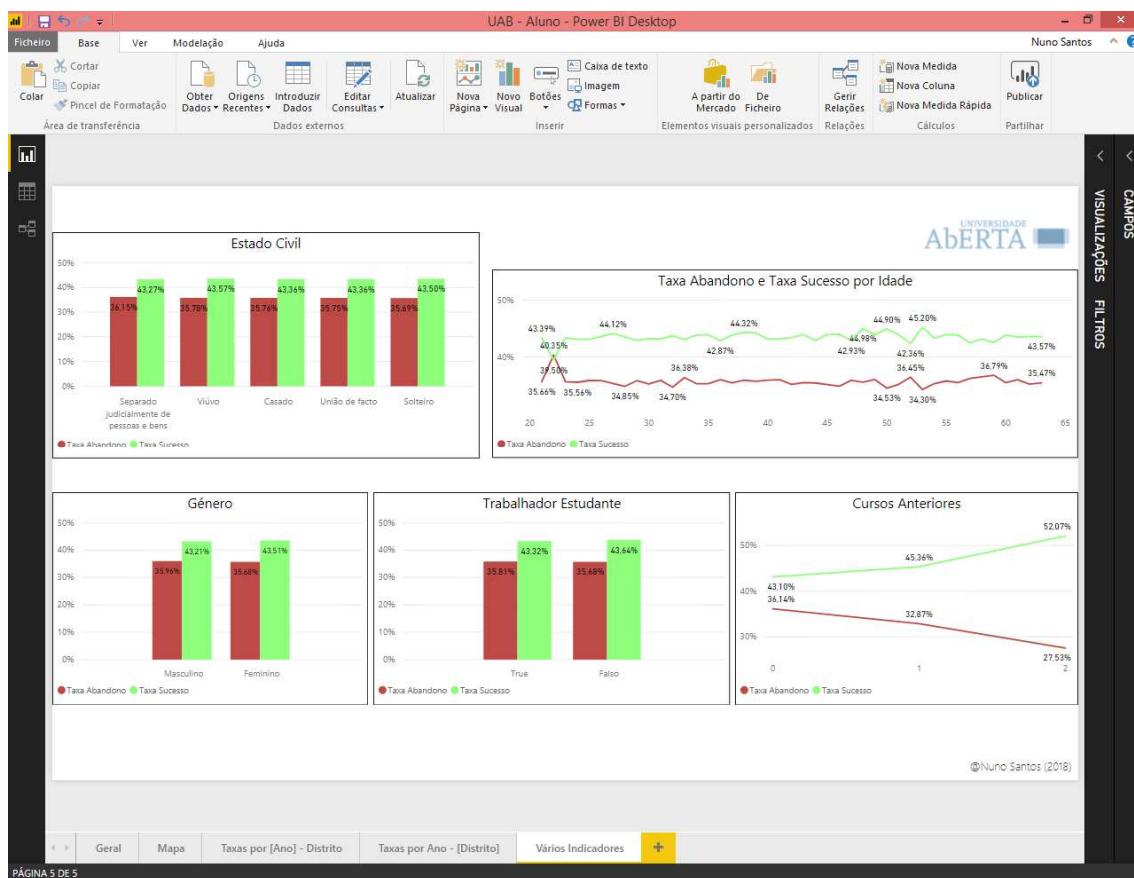


FIGURA 7.10 - Vários indicadores

Nesta última página dos relatórios da estrela Aluno temos vários gráficos de barras e linhas expondo os dados de vários indicadores: estado civil, género, idade, trabalhador estudante e nº de cursos anteriores.

Foram manipulados os dados dos gráficos da idade introduzindo uma perturbação nos valores da idade 21/22 anos, e no gráfico do nº de cursos anteriores foram manipulados os dados por forma a aparentar que quantos mais nº de cursos anteriores houver, maior é a Taxa de Sucesso e menor a Taxa de Abandono.

Como os dados foram criados por métodos aleatórios, em alguns casos influenciados em algumas variáveis, tem como consequência serem muito diferentes da realidade e serem em muitas situações muito equilibrados.

7.5. CURSO

Para o *Dashboard* Curso foram criadas 4 páginas:

1. Nº de UCs por Curso e Ano Letivo: página com gráfico de barras e matriz de Taxas por Curso. Existe também um gráfico de barras com o nº de UCs classificadas como abandono distribuídas por Curso. Responde às questões: "Número de UCs associado Abandono Escolar por Ano Letivo".
2. Curso – Mapa: utiliza os visuais 'Mapa' e permite localizar no mapa a distribuição geográfica dos valores das Taxa de Abandono e Sucesso Escolar. Tem como filtros o Distrito e Curso para possibilitar a análise mais em detalhe com combinação variáveis. Responde às questões: "Distribuição geográfica dos Cursos associada ao Sucesso e Abandono Escolar".
3. Departamento – Mapa: página com visuais semelhantes à página anterior mas com a variável Departamento em vez de Curso e com o filtro apenas do Distrito. Responde às questões: "Departamentos associados ao Sucesso e Abandono Escolar".
4. Departamento - Distrito e Ano Letivo: Utiliza gráfico de barras com distribuição por Departamento e recorre aos filtros Distrito e Ano Letivo para uma combinação de variáveis para análise mais detalhada. Responde às questões: "Departamentos associados ao Sucesso e Abandono Escolar".

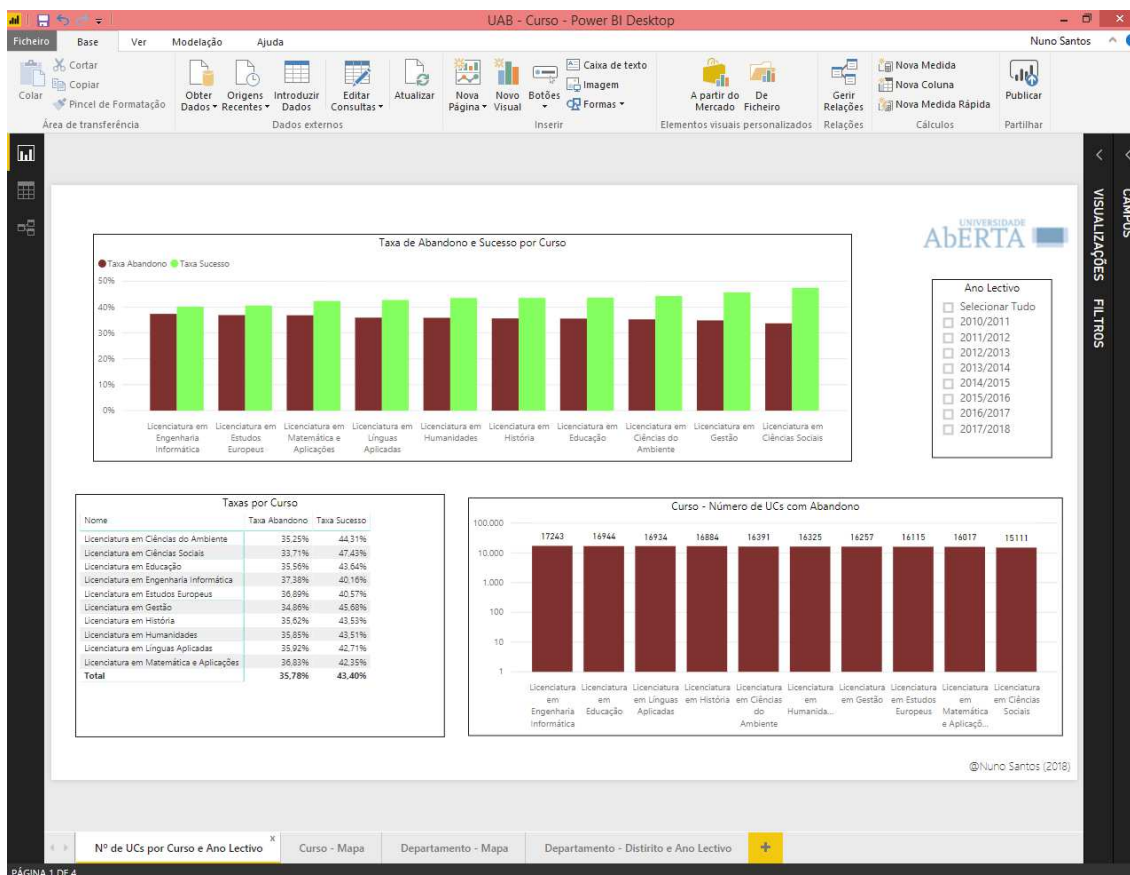


FIGURA 7.11 - Nº de UCs por Curso e Ano Letivo

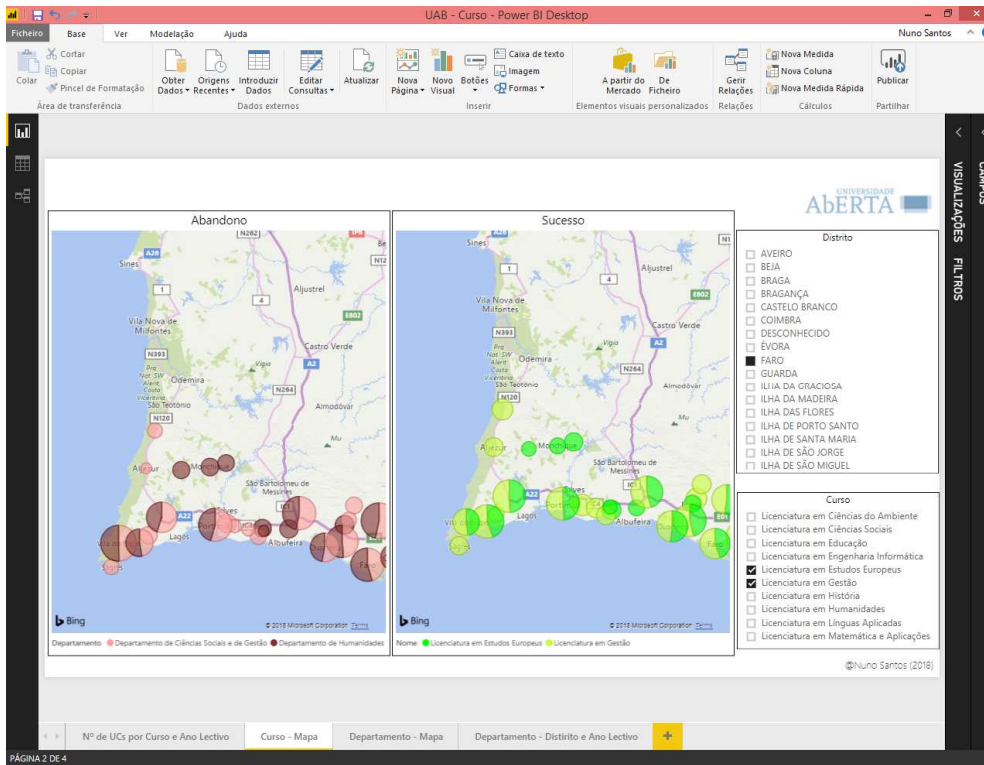


FIGURA 7.12 - Curso - Mapa

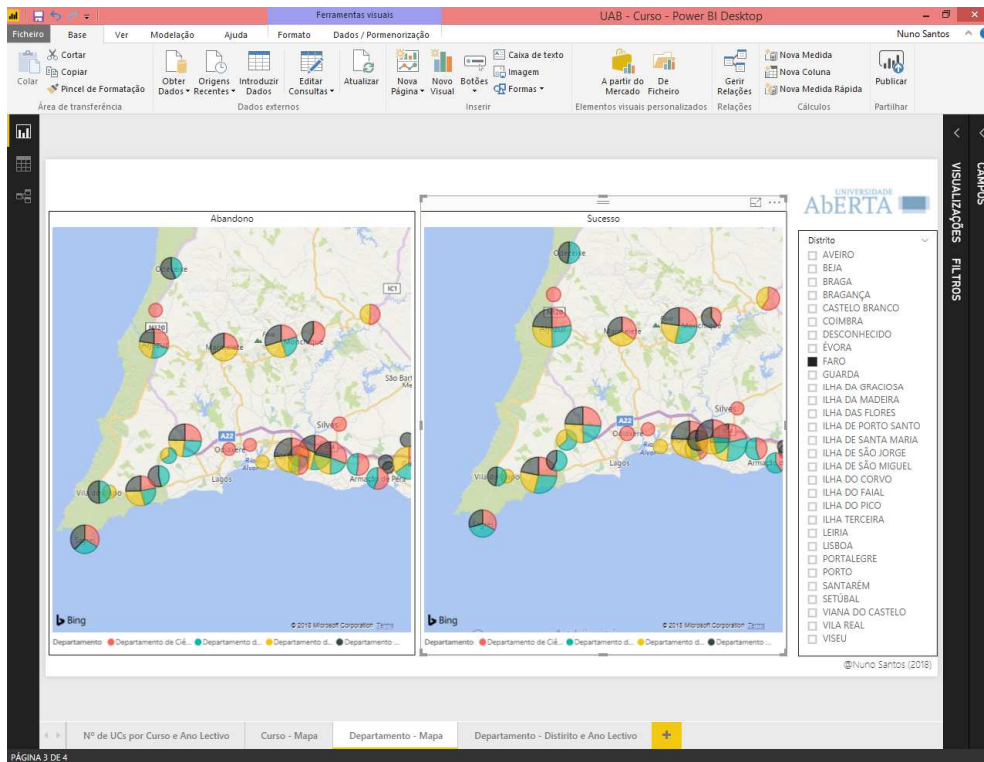


FIGURA 7.13 - Departamento - Mapa

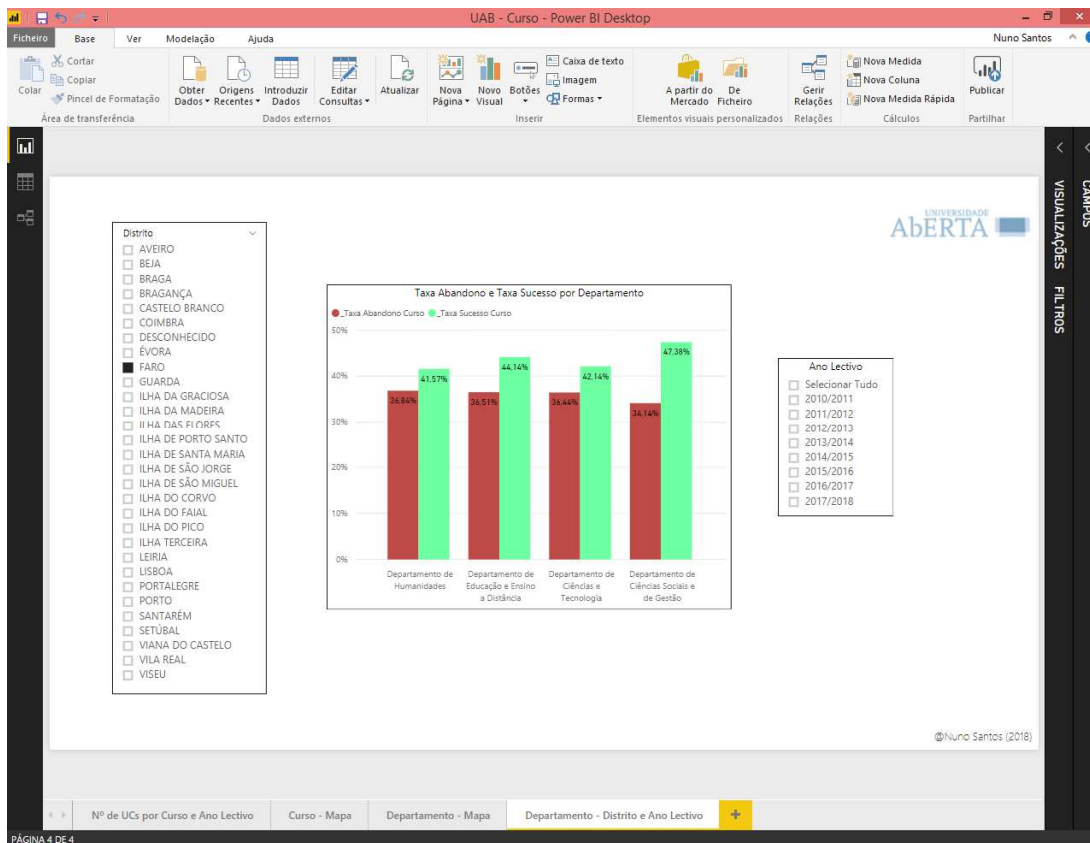


FIGURA 7.14 - Departamento - Distrito e Ano Lectivo

7.6. UC

No sentido de acautelar o correto valor das Taxas devido à relação de 1 para n entre as entidades [DimDocente] e [FactosUC] foi criada no *Power Bi* uma cópia da entidade [FactosUC] a que foi nomeada de [FactosUC Sem Docente] com os registos distintos da tabela original mas sem a chave referente à tabela [DimDocente]. Para utilizou-se o editor do *Power Bi* removendo a coluna do Docente e fazendo a seleção distinta dos registos.

Para o *Dashboard UC* foram criadas 3 páginas:

1. Taxas por Tipo Avaliação e Distrito: página com gráfico de barras com a distribuição de Taxas por Tipo de Avaliação. Existe também uma matriz com a mesma distribuição acrescida da variável distrito. Responde às questões: "Tipo de avaliação associada ao Sucesso e Abandono Escolar".
2. Médias Notas e Reprovados: constituída por 2 matrizes, uma com a média das notas associadas ao Sucesso Escolar e outra com a quantidade e taxa de reprovação de UCs. Tem associado um filtro por Ano. Responde às questões: "Média das notas associada ao Sucesso Escolar" e "Identificação das UCs e Taxas com avaliação Reprovado".

3. Mapa Top 5 + Docentes: página com 2 visuais do tipo 'Mapa'. Identifica o top 5 das UCs com melhor e pior média distribuídas geograficamente. Do lado esquerdo existem 2 matrizes com o top 10 dos docentes associados ao Sucesso Escolar e ao Abandono Escolar quantificado as taxas associadas às UCs por docente.

Responde às questões: "Top 10 Docentes associadas às Taxas de Sucesso e Abandono Escolar" e à "Distribuição do top 5 das melhores e piores médias das UCs geograficamente".

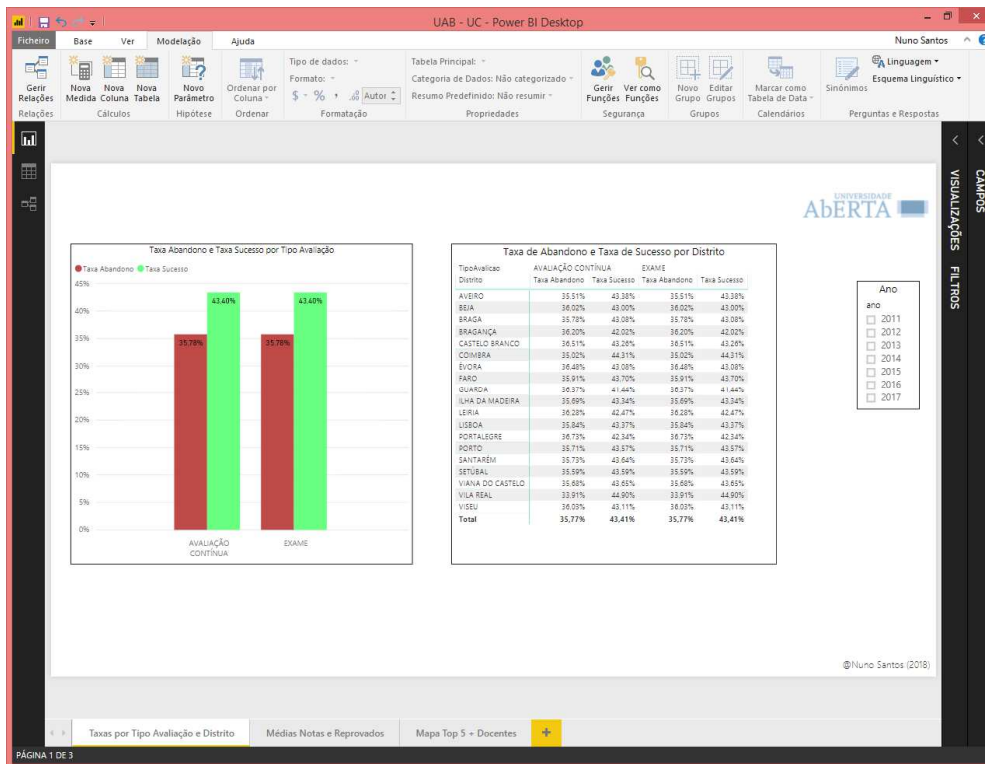


FIGURA 7.15 - Taxas por Tipo Avaliação e Distrito

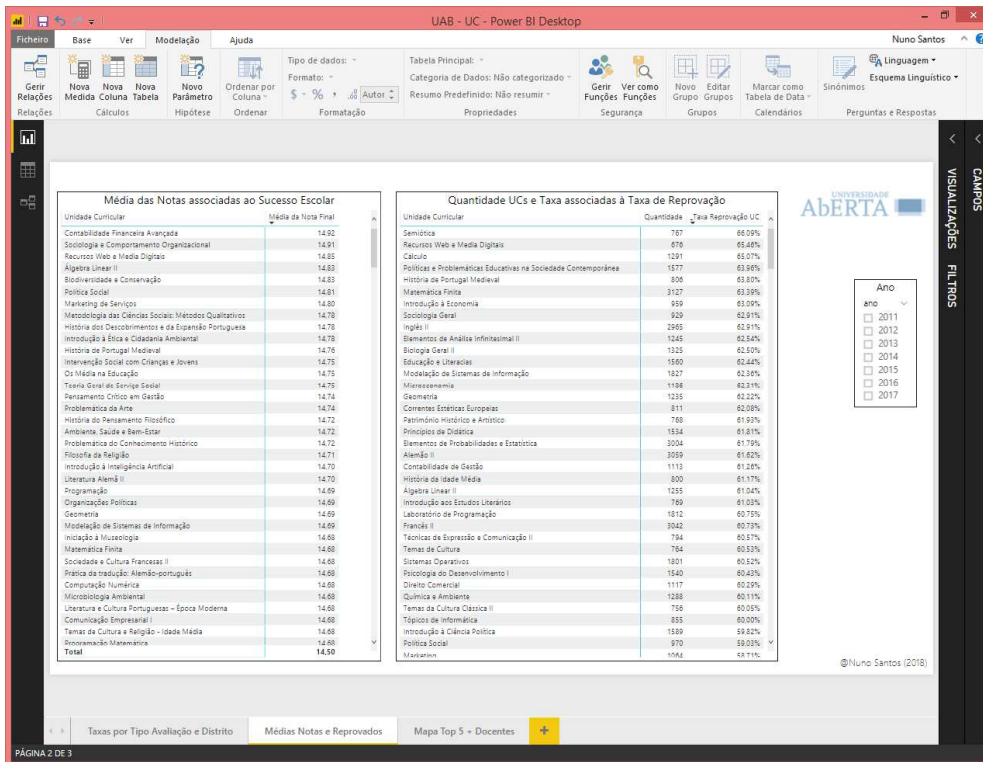


FIGURA 7.16 - Médias Notas e Reprovados

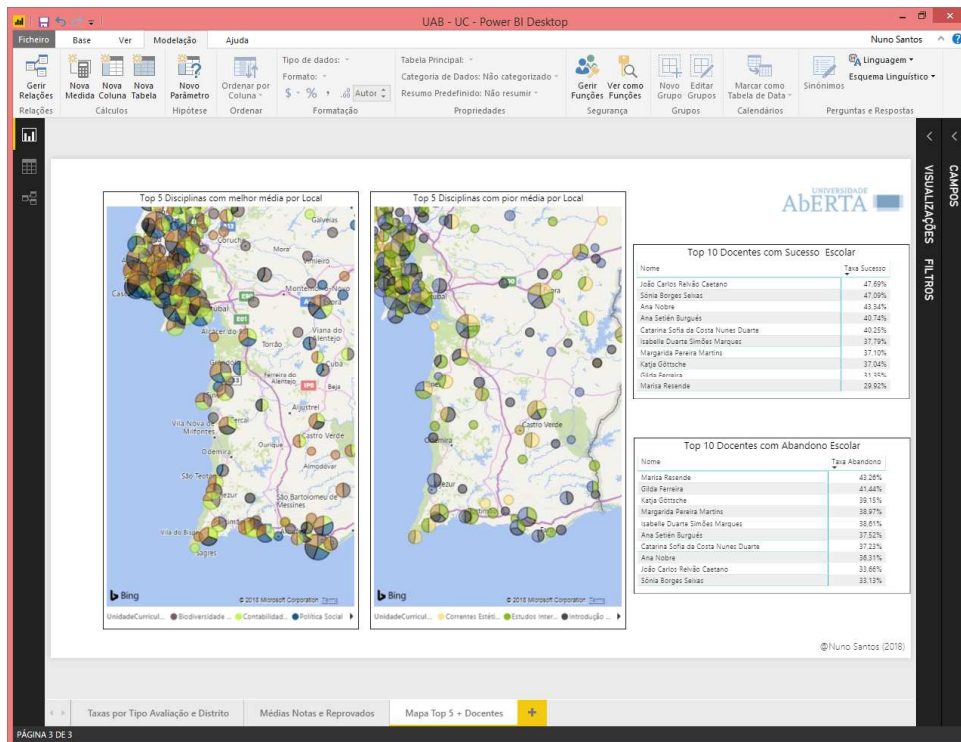


FIGURA 7.17 - Mapa Top 5 + Docentes

7.7. PUBLICAÇÃO DOS RELATÓRIOS

Existem 2 métodos para visualização dos relatórios.

- Na própria ferramenta *Power BI*, onde foram recolhidas as imagens que foram apresentadas atrás no documento, sendo que nesta ferramenta além de visualizar o relatório existe a possibilidade de trabalhar o relatório com outros atributos, pois estão disponíveis os dados para atingir esse fim. Desta forma o trabalho/análise nos relatórios fica potenciada devido à disponibilidade das ferramentas ser maior, no entanto obriga a que esteja instalado no cliente uma versão da aplicação.
- Outro método de visualizar os relatórios será fazer a publicação dos relatórios no servidor *Power BI* a partir da funcionalidade existente na ferramenta:

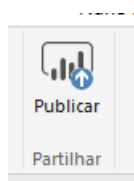


FIGURA 7.18 - Publicar

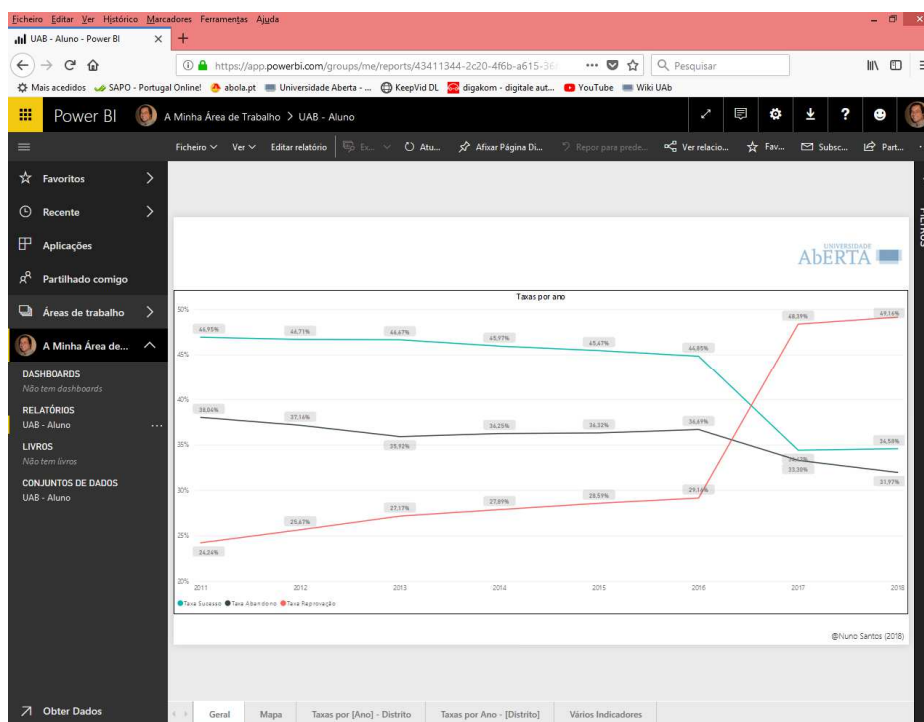


FIGURA 7.19 - Relatório no Servidor

Este método permite aceder aos relatórios em qualquer local e em qualquer altura, desde que o utilizador tenha acesso à internet, pois o portal do *Power BI* está sempre disponível.

Tratando-se de uma versão gratuita, os relatórios apenas estão acessíveis para o utilizador que os publicou (ou quem tenha acesso às credenciais deste) e num limite de 10 GB de dados.

Para uma partilha universal tem de ser adquirida uma licença de *Power BI Pro* (pode ser experimentada gratuitamente por 60 dias).

8. CONCLUSÕES

8.1. TRABALHO REALIZADO

A proposta para este trabalho visou a construção de uma base de dados transacional (OLTP) e de um *Data Warehouse*, incluindo os processos ETL e a componente de *Business Intelligence*.

O tema escolhido foi de uma Universidade, e a Universidade Aberta foi aquela que reuniu mais consenso devido à experiência mais recente nas interações com esta instituição.

Em termos de temas do DW escolhidos pelo professor orientador foram o Abandono Escolar e o Sucesso Escolar.

Numa fase preliminar estive em cima da mesa a possibilidade de trabalhar com dados reais anonimizados. A perspetiva afigurava-se como um cenário bastante aliciante, pois o resultado final incidiria sobre as técnicas utilizadas na construção das bases de dados e dos processos associados, mas a análise à luz de dados reais era bastante promissora no que diz respeito à análise real dos dados no contexto dos temas propostos.

O desafio proposto foi cumprido atendendo aos objetivos inicialmente propostos, sendo estes:

- Projetar e desenvolver bases de dados transacionais (OLTP) dos vários subsistemas da Universidade Aberta, sendo uma das bases principais a contribuir para a projeção a experiência empírica.
- Criação de processos de carregamento de dados aleatórios nas bases de dados transacionais dos vários subsistemas, aproveitando em paralelo dados reais extraídos do portal da UAb. Esta atividade revelou-se um desafio maior que o esperado e com um grau de complexidade grande. O resultado atingido não é perfeito, pois não permite aferir a realidade, mas permite demonstrar com o resultado final, que aplicando as técnicas certas de modelação multidimensional, se consegue fabricar ferramentas específicas para a tomada de decisões, concretamente no âmbito da atividade escolar e mais especificamente no estudo do Abandono e Sucesso Escolar.
- Projetar e desenvolver um DW no contexto dos temas propostos para o este projeto.
- Criação de processo ETL, selecionando e extraindo os dados das bases de dados transacionais.

- Criação de uma série de relatórios dentro do contexto dos temas propostos. A demonstração dos relatórios também foi potencializada pela ferramenta de exploração de dados *Power BI*. A modelação deve e tem de ser independente das ferramentas de *Business Intelligence*, mas o resultado final ficou acrescido de valor devido à ferramenta escolhida. A construção dos relatórios foi relativamente simples de construir dado que a base que a suporta ter sido estruturada para esse fim.

Pode-se concluir que para tomadas de decisão existe sempre necessidade de ter ferramentas à altura que ajudem a delinear o rumo a que a Universidade quer seguir, sendo a gestão dos recursos materiais e humanos aqueles que mais podem definir o nível da qualidade a oferecer aos seus clientes.

Os gestores/decisores não podem "navegar à vista", isto é, necessitam de ter uma linha base (*baseline*), e para identificar essa linha, é necessário, com base nos dados existentes, construir um primeiro Data Warehouse no sentido de conseguir ter dados, métricas reais sobre a situação real.

Após identificação da *baseline*, consegue-se analisar o curso dos factos com base nos vários carregamentos dos dados e continua análise dos dados.

Da análise de dados podemos chegar à conclusão que determinado CLA está longe/perto de determinados locais e que esse fato pode influenciar as Taxas de Abandono/Sucesso, que determinado escalão etário tem determinado comportamento, etc.

Os relatórios criados são apenas uma amostra daquilo que se pode fazer com base no modelo construído, isto é, tendo em mãos os vários atributos e dados, o utilizador com alguma experiência, consegue manipular as variáveis de forma a conseguir extrair informação que o ajude a ter uma visão mais concreta dos vários cenários a analisar.

8.2. TRABALHO FUTURO

Sabendo que os dados criados estão longe de representar a realidade dos factos, pode este trabalho ser aproveitado para, num ambiente com dados reais, fazer uma implementação de modo a ter uma ideia real da situação em que se encontra a UAb.

Com base em dados reais, outros parâmetros podem surgir e serem analisados estendendo as mesmas técnicas utilizadas neste trabalho e aprofundar o tema.

A implementação de um *DW* na UAb é importante. Importante também é o estudo de variáveis geográficas num contexto de uma Universidade cujo ensino é feito à distância, e a troca de experiências entre Instituições de ensino cujo método de ensino também seja feito à distância é importante para promover a melhoria dos métodos e recursos de ensino.

ANEXO I – CÓDIGO PARA POPULAR BASES DE DADOS OLTP

Código para popular bases de dados transacionais: Tabelas [Aluno] e [Morada]

```
-- =====
-- Autor:          Nuno Santos
-- Data Criação:   2018-05-25
-- Descrição:     Procedimento para criar dados anónimos e aleatórios nas tabelas
--               [Aluno] e [Morada]
-- =====
CREATE PROCEDURE [dbo].[CriaDadosAluno] (@p_nrRegistos int)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @v_AlunoNumero int = 1200000, @i int = 1
    DECLARE @v_CodigoPostal int, @v_Localidade nvarchar(64)
    DECLARE @v_Distrito int

    DECLARE @random_EstadoCivil int
    DECLARE @random_Sexo int
    DECLARE @random_TipoIdentificacao int
    DECLARE @random_Deslocado int
    DECLARE @random_CursosAnteriores int
    DECLARE @random_Activo int
    DECLARE @random_Pais float
    DECLARE @random_Distrito float
    DECLARE @random_TrabalhadorEstudante float

    DECLARE @v_pais nvarchar(3)

    DECLARE @v_dataNascimentoInicio AS date = '19550101', @v_dataNascimentoFim
AS date = '19980101'
    --idade entre 20 e 63 anos

    DELETE FROM Aluno
    DELETE FROM Morada WHERE NumeroMorada > 999

    WHILE @i < @p_nrRegistos
    BEGIN

        SET @random_EstadoCivil = (RAND() * 100) + 1
/* Estado Civil
    40% C Casado
    10% J Separado judicialmente de pessoas e bens
    0% N Não Definido
    30% S Solteiro
    15% U União de facto
    5% V Viúvo
*/

        SET @random_TipoIdentificacao = (RAND() * 100) + 1
/* Tipo Identificação
    25% BI BILHETE DE IDENTIDADE CIVIL
    75% CC CARTÃO DE CIDADÃO
*/

        SET @random_Deslocado = (RAND() * 100) + 1
/* Deslocado
    90% Não
    10% Sim
*/
```

```

        SET @random_Sexo = (RAND() * 100) + 1
/* Deslocado
   65% F
   35% M
*/

        SET @random_CursosAnteriores = (RAND() * 100) + 1
/* Cursos Anteriores
   85% 0
   12% 1
   3% 2
*/

        SET @random_Activo = (RAND() * 100) + 1
/* Activo
   95% Sim
   5% Não
*/

        SET @random_TrabalhadorEstudante = (RAND() * 100) + 1
/* TrabalhadorEstudante
   75% Sim
   25% Não
*/

        SET @random_Pais = round((RAND() * 100),2) + 1
        SELECT @v_pais=Pais FROM PercentagemPais WHERE Valor = (
            SELECT min(valor) FROM PercentagemPais WHERE valor > @random_Pais )
-- As percentagens de probabilidade estão na tabela PercentagemPais

        IF @v_pais <> 'PT'
        BEGIN
            SET @v_Localidade = '*****'
            SET @v_CodigoPostal = 9999999
        END
        ELSE
        BEGIN
            SET @random_Distrito = round((RAND() * 100),2) + 1
            SELECT @v_Distrito=Distrito FROM PercentagemDistrito WHERE Valor =
            (
                SELECT min(Valor) FROM PercentagemDistrito WHERE Valor >
                @random_Distrito )
            -- As percentagens de probabilidade estão na tabela PercentagemDistrito

            SELECT top 1 @v_CodigoPostal=CodigoPostal,
                @v_Localidade=Localidade
            FROM [UAb-PortalAcademico].dbo.CodigoPostal
            WHERE distrito = @v_Distrito ORDER BY NEWID()
        END

        INSERT INTO Aluno (
            [AlunoNumero],
            [Nome] ,
            [DataNascimento],
            [Sexo],
            [EstadoCivil],
            [TipoIdentificacao],
            [NumeroIdentificacao],
            [Nacionalidade],
            [NaturalidadeDistrito],
            [NaturalidadeConcelho],
            [NaturalidadeFreguesia],
            [DataValidadeVacinaAntitetânica],

```

```

[MaeNome],
[PaiNome],
[NumeroMorada],
[Deslocado],
[Telefone1],
[Telefone2],
[Telemovel],
[Email],
[PaisFiscal],
[NumeroIdentificacaoFiscal],
[NIB],
[IBAN],
[CursosAnteriores],
[Activo],
[TrabalhadorEstudante]
)
VALUES
(
@v_AlunoNumero + @i, --[AlunoNumero]
'**** Nome aplicada máscara de dados ****', -- [Nome]
dateadd(day, rand(checksum(newid()))*(1+datediff(day,
@v_dataNascimentoInicio, @v_dataNascimentoFim)), @v_dataNascimentoInicio), --
[DataNascimento]
CASE WHEN @random_Sexo > 65 THEN 'M'
ELSE 'F'
END, -- [Sexo]
CASE WHEN @random_EstadoCivil > 60 THEN 'C'
WHEN @random_EstadoCivil > 50 THEN 'J'
WHEN @random_EstadoCivil > 20 THEN 'S'
WHEN @random_EstadoCivil > 5 THEN 'U'
ELSE 'V'
END, --[EstadoCivil]
CASE WHEN @random_TipoIdentificacao > 75 THEN 'BI'
ELSE 'CC'
END, -- [TipoIdentificacao],
100000000 + floor(9999999 * RAND(convert(varbinary,
newid()))), -- [NumeroIdentificacao]
'PT', -- [Nacionalidade]
0 , -- [NaturalidadeDistrito],
0 , -- [NaturalidadeConcelho],
0 , -- [NaturalidadeFreguesia],
getdate()+365, -- [DataValidadeVacinaAntitetânica]
'**** MaeNome aplicada máscara de dados ****', --
[MaeNome]
'**** PaiNome aplicada máscara de dados ****', --
[PaiNome]
1000 + @i, -- [NumeroMorada]
CASE WHEN @random_Deslocado > 90 THEN 'S'
ELSE 'N'
END, -- [Deslocado],
200000000 + floor(9999999 * RAND(convert(varbinary,
newid()))), -- [Telefone1]
null, -- [Telefone2]
900000000 + floor(9999999 * RAND(convert(varbinary,
newid()))), -- [Telemovel]
'dummy@dummy.pt', -- [Email]
null, --[PaisFiscal]
100000000 + floor(9999999 * RAND(convert(varbinary,
newid()))), -- [NumeroIdentificacaoFiscal]
null, --[NIB]
null, --[IBAN]
CASE WHEN @random_CursosAnteriores > 96 THEN 2
WHEN @random_CursosAnteriores > 85 THEN 1
ELSE 0
END, -- [CursosAnteriores]

```

```

CASE WHEN @random_Activo > 95 THEN 'N'
      ELSE 'S'
END,      -- [Activo]
CASE WHEN @random_TrabalhadorEstudante > 75 THEN 0
      ELSE 1
END      -- [TrabalhadorEstudante]
)

INSERT INTO Morada (
    [NumeroMorada],
    [PaisCodigo],
    [Artéria],
    [Localidade],
    [CodigoPostal],
    [Regiao],
    [Endereco_linha_1],
    [Endereco_linha_2],
    [Endereco_linha_3]
)
VALUES
(
    1000 + @i,      --[NumeroMorada]
    @v_pais,      --[PaisCodigo]
    '**** Nome aplicada máscara de dados ****',
-- [Artéria]
    @v_Localidade,  --[Localidade]
    @v_CodigoPostal,  --[CodigoPostal]
    null,          --[Regiao]
    null,          --[Endereco_linha_1]
    null,          --[Endereco_linha_2]
    null           --[Endereco_linha_3]
)

        SET @i = @i + 1;
END -- WHILE
END

```

Código para popular bases de dados transacionais: Tabelas [Perfil], [MatriculaInscricao] e [Avaliacao]

```

-- =====
-- Autor:          Nuno Santos
-- Data Criação:  2018-05-25
-- Descrição:  Procedimento para criar dados anónimos e aleatórios nas tabelas
-- [Perfil], [MatriculaInscricao] e [Avaliacao]
-- =====
ALTER PROCEDURE [dbo].[CriaDadosAvaliacoes]
AS
BEGIN
    SET NOCOUNT ON;
print 'Inicio'
print getdate()

    delete from Perfil

    declare @v_AlunoNumero int, @v_Curso int
    declare @v_DataCriacao date, @v_DataEstado date
    declare @v_Estado char
    declare @v_ano int, @v_ects int, @v_aux int

```

```

declare @random_DataCriacao float
declare @random_Estado float

-- cursor para percorrer alunos
declare c_alunos cursor for
SELECT AlunoNumero
FROM Aluno

open c_alunos
fetch next from c_alunos into @v_AlunoNumero
while @@fetch_status = 0
begin

set @v_Curso = ROUND(RAND()*10+0.5,0)
-- distribuição uniforme dos 10 cursos de 1º ciclo

SET @random_DataCriacao = ROUND((RAND() * 100),0) + 1
/* Data Criação
5% [2012,2017] (30%)
20% 2009
25% 2010
25% 2011
*/

SET @random_Estado = (RAND() * 100) + 1
/* Estado
5% Inactivo
95% Activo
*/

select @v_DataCriacao = CASE WHEN @random_DataCriacao > 80 THEN
'20090601'
WHEN @random_DataCriacao > 55 THEN '20100601'
WHEN @random_DataCriacao > 30 THEN '20110601'
WHEN @random_DataCriacao > 25 THEN '20120601'
WHEN @random_DataCriacao > 20 THEN '20130601'
WHEN @random_DataCriacao > 15 THEN '20140601'
WHEN @random_DataCriacao > 10 THEN '20150601'
ELSE '20160601'
END

select @v_Estado = CASE WHEN @random_Estado > 5.0 THEN 'A'
ELSE 'I'
END

set @v_DataEstado = dateadd(day,(ROUND(RAND()*180,0) ),@v_DataCriacao )
-- acrescenta múltiplos aleatórios de 180 à data de criação
INSERT INTO [dbo].[Perfil]
([AlunoNumero]
,[CursoID]
,[DataCriacao]
,[Estado]
,[DataEstado])
VALUES
(@v_AlunoNumero, -- AlunoNumero
@v_Curso, -- CursoID
@v_DataCriacao, -- DataCriacao
@v_Estado, -- Estado
@v_DataEstado -- DataEstado
)

```

```

        fetch next from c_alunos into @v_AlunoNumero
    end
    close c_alunos
    deallocate c_alunos

-----
-- MatriculaInscricao --
-----

delete from MatriculaInscricao

declare @v_PerfilID int, @v_semestre int = 0
declare @v_DataInscricao date
declare @v_NotaFinal float
declare @v_UnidadeCurricularCodigo int
declare @v_i int
declare @v_AnoLectivo int

declare @random_UC int
declare @random_Nota int

-- cursor para percorrer perfis
declare c_perfis cursor for
SELECT AlunoNumero, PerfilID, CursoID, DataCriacao
FROM Perfil

open c_perfis
fetch next from c_perfis into @v_AlunoNumero, @v_PerfilID , @v_Curso,
@v_DataCriacao
while @@fetch_status = 0
begin
    set @v_ects = 0
    set @v_semestre = 0
    set @v_DataEstado = datefromparts(YEAR(@v_DataCriacao)+1,2,1) --
Fevereiro do próximo ano
    while @v_DataEstado < '20180101' and @v_ects < 180
    begin
        -----
        -- 1º SEMESTRE --
        -----
        if @v_semestre = 0
        begin
            set @v_semestre = 1
            set @random_UC = ABS(Checksum(NewID()) % 5) + 1 --
máximo 5 disciplinas mínimo 3 no primeiro ano/semestre

            declare c_UCs cursor for
            select X.* from (
            select distinct a.UnidadeCurricularCodigo from [UAb-
Portal].dbo.UnidadeCurricular a
            INNER JOIN [UAb-Portal].dbo.UnidadeCurricularCurso b
ON a.UnidadeCurricularCodigo = b.UnidadeCurricularCodigo
            WHERE b.CursoID = @v_Curso
            and b.Semestre = @v_semestre
            and ano = 1) X
            ORDER BY NEWID() -- Ordenação aleatória das
UCs

            set @v_i = 0

            open c_UCs
            fetch next from c_UCs into @v_UnidadeCurricularCodigo

```

```

while @@fetch_status = 0 and @v_i < @random_UC + 1--
só processa o número de UCs que foi obtido aleatoriamente
begin

SET @random_Estado = (RAND() * 100) + 1

/* Estado
   > 2017
   30% A - Aprovado
   50% V - Avaliado
   20% I - Inscrito

   < 2017
   45% A - Aprovado
   10% V - Avaliado
   45% I - Inscrito
*/

*/
if @v_DataCriacao > '20170101'
select @v_Estado = CASE WHEN
@random_Estado > 69 THEN 'A'

WHEN @random_Estado > 20 THEN 'V'

ELSE 'I'

END
else
select @v_Estado = CASE WHEN
@random_Estado > 54 THEN 'A'

WHEN @random_Estado > 45 THEN 'V'

ELSE 'I'

END

set @v_DataEstado =
datefromparts(YEAR(@v_DataCriacao)+1,2,1) -- Fevereiro do próximo ano
set @v_DataInscricao =
datefromparts(YEAR(@v_DataCriacao),8,1) -- Agosto do mesmo ano

if @v_Estado = 'A'
set @v_NotaFinal = ABS(Checksum(NewID()))
% 10) + 10 -- notas de 10 a 20 (Aprovado)
else if @v_Estado = 'V'
set @v_NotaFinal = ABS(Checksum(NewID()))
% 9) -- notas de 0 a 9 (Reprovado)
else
set @v_NotaFinal = -1 -- inscrito nota =
-1

SELECT @v_AnoLectivo = AnoLectivoID
FROM [UAB-PortalAcademico].[dbo].[AnoLectivo]
where dataFim = (select min(datafim) from [UAB-
PortalAcademico].[dbo].[AnoLectivo]
where dataFim > @v_DataEstado)

INSERT INTO [dbo].[MatriculaInscricao]
(
[AlunoNumero]
,[UnidadeCurricularCodigo]
,[PerfilID]
,[DataInscricao]
,[Estado]
,[DataEstado]
,[NotaFinal]
,[AnoLectivo])
VALUES

```

```

(
    -- AlunoNumero
    @v_AlunoNumero,
    -- UnidadeCurricularCodigo
    @v_UnidadeCurricularCodigo,
    -- PerfilID
    @v_PerfilID,
    -- DataInscricao
    @v_DataInscricao,
    -- Estado
    @v_Estado,
    -- DataEstado
    @v_DataEstado,
    -- NotaFinal
    @v_NotaFinal,
    -- AnoLectivo
    @v_AnoLectivo
)

    fetch next from c_UCs into
@v_UnidadeCurricularCodigo
    set @v_i = @v_i + 1
end
close c_UCs
deallocate c_UCs
set @v_DataCriacao = @v_DataEstado

end

-----
-- OUTROS SEMESTRES --
-----

-- troca de semestre em tratamento
if @v_semestre = 1
    set @v_semestre = 2
else
    set @v_semestre = 1

select @v_ects =SUM(ECTS) FROM [MatriculaInscricao] a
inner join [UAb-Portal].dbo.UnidadeCurricular b ON
a.UnidadeCurricularCodigo = b.UnidadeCurricularCodigo
WHERE perfilID=@v_PerfilID
AND estado = 'A'

if @v_ects is null
    set @v_ects = 0

if @v_DataEstado < '20180101' AND @v_ects <180
begin

/*
-- troca de semestre em tratamento
if @v_semestre = 1
    set @v_semestre = 2
else
    set @v_semestre = 1
*/

if 180 - @v_ects < 37
begin
    set @v_aux = round((180 - @v_ects)/6,0)
    if @v_aux <> 0
        set @random_UC = ABS(Checksum(NewID())) %
@v_aux) -- máximo 6 disciplinas após primeiro ano/semestre
    else
        set @random_UC =1

```

```

        end
    else
        set @random_UC = ABS(Checksum(NewID())) % 5) + 1
-- máximo 6 disciplinas após primeiro ano/semestre

        if @random_UC = 0 set @random_UC =1

        if @v_semestre = 1

            begin
                set @v_DataEstado =
datefromparts(YEAR(@v_DataCriacao)+1,2,1) -- Fevereiro do próximo ano
                set @v_DataInscricao =
datefromparts(YEAR(@v_DataCriacao),8,14) -- Agosto do mesmo ano
                end
            else
                begin
                    set @v_DataEstado =
datefromparts(YEAR(@v_DataCriacao),7,1) -- Julho do mesmo ano
                    set @v_DataInscricao =
datefromparts(YEAR(@v_DataCriacao),1,20) -- Janeiro do mesmo ano
                    end

                    declare c_UCs cursor for
                    select X.* from (
                    select distinct a.UnidadeCurricularCodigo, min(b.Ano)
Ano from [UAb-Portal].dbo.UnidadeCurricular a
                    INNER JOIN [UAb-Portal].dbo.UnidadeCurricularCurso b
ON a.UnidadeCurricularCodigo = b.UnidadeCurricularCodigo
                    WHERE b.CursoID = @v_Curso
                    and b.Semestre = @v_semestre
                    and a.UnidadeCurricularCodigo not in (select
UnidadeCurricularCodigo FROM [MatriculaInscricao] WHERE perfilID=@v_PerfilID AND
Estado='A')

                    group by a.UnidadeCurricularCodigo
                    ) X
                    ORDER BY NEWID() -- Ordenação aleatória das
UCs

                    set @v_i = 0

                    open c_UCs
                    fetch next from c_UCs into @v_UnidadeCurricularCodigo,
@v_ano

                    while @@fetch_status = 0 and @v_i < @random_UC+1 -- só
processa o número de UCs que foi obtido aleatoriamente
                    begin

                        SET @random_Estado = (RAND() * 100) + 1

/* Estado
1ºano
    > 2017
    20% A - Aprovado
    30% V - Avaliado
    50% I - Inscrito

    < 2017
    20% A - Aprovado
    35% V - Avaliado
    45% I - Inscrito
2º ano
    > 2017
    35% A - Aprovado
    45% V - Avaliado

```

```

20% I - Inscrito

< 2017
45% A - Aprovado
15% V - Avaliado
40% I - Inscrito

3º ano
> 2017
40% A - Aprovado
30% V - Avaliado
30% I - Inscrito

< 2017
60% A- Aprovado
10% V - Avaliado
30% I - Inscrito
*/

```

```

if @v_DataEstado > '20170101'
begin
if @v_ano = 1
select @v_Estado = CASE WHEN
@random_Estado > 80 THEN 'A'
WHEN @random_Estado > 50 THEN 'V'
ELSE 'I'
END
else if @v_ano = 2
select @v_Estado = CASE WHEN
@random_Estado > 64 THEN 'A'
WHEN @random_Estado > 20 THEN 'V'
ELSE 'I'
END
else
select @v_Estado = CASE WHEN
@random_Estado > 59 THEN 'A'
WHEN @random_Estado > 30 THEN 'V'
ELSE 'I'
END
end
else
begin
if @v_ano = 1
select @v_Estado = CASE WHEN
@random_Estado > 80 THEN 'A'
WHEN @random_Estado > 45 THEN 'V'
ELSE 'I'
END
else if @v_ano = 2
select @v_Estado = CASE WHEN
@random_Estado > 54 THEN 'A'
WHEN @random_Estado > 40 THEN 'V'
ELSE 'I'
END
else
select @v_Estado = CASE WHEN
@random_Estado > 40 THEN 'A'

```

```

WHEN @random_Estado > 30 THEN 'V'

ELSE 'I'

END

end
if @v_Estado = 'A'
    set @v_NotaFinal = ABS(Checksum(NewID()))
% 10) + 10 -- notas de 10 a 20 (Aprovado)
else if @v_Estado = 'V'
    set @v_NotaFinal = ABS(Checksum(NewID()))
% 9) -- notas de 0 a 9 (Reprovado)
else
    set @v_NotaFinal = -1 -- inscrito nota =
-1

--
--
if @v_NotaFinal < 3.5
    set @v_NotaFinal = 0

SELECT @v_AnoLectivo = AnoLectivoID
FROM [UAb-PortalAcademico].[dbo].[AnoLectivo]
where dataFim = (select min(datafim) from [UAb-
PortalAcademico].[dbo].[AnoLectivo]
where dataFim > @v_DataEstado)

INSERT INTO [dbo].[MatriculaInscricao]
(
    [AlunoNumero]
    , [UnidadeCurricularCodigo]
    , [PerfilID]
    , [DataInscricao]
    , [Estado]
    , [DataEstado]
    , [NotaFinal]
    , [AnoLectivo])
VALUES
(
    @v_AlunoNumero,
    @v_UnidadeCurricularCodigo,
    @v_PerfilID,
    @v_DataInscricao,
    @v_Estado,
    @v_DataEstado,
    @v_NotaFinal,
    @v_AnoLectivo
)

fetch next from c_UCs into
@v_UnidadeCurricularCodigo, @v_ano
set @v_i = @v_i + 1
end
close c_UCs
deallocate c_UCs
set @v_DataCriacao = @v_DataEstado
end
end

```

```

        fetch next from c_perfis into @v_AlunoNumero, @v_PerfilID,
@v_Curso, @v_DataCriacao
    end
    close c_perfis
    deallocate c_perfis

-- Eliminar UC para os cursos com + de 180 ECTS
delete from [MatriculaInscricao] where matriculaInscricaoId in
(select max(matriculaInscricaoId) AS matriculaInscricaoId from
[MatriculaInscricao]
where perfilID in
(
select perfilID FROM [MatriculaInscricao] a
inner join [UAb-Portal].dbo.UnidadeCurricular b ON a.UnidadeCurricularCodigo =
b.UnidadeCurricularCodigo
WHERE estado = 'A'
group by perfilID
having SUM(ECTS) > 180
)
AND estado = 'A'
group by perfilID)

print 'Fim inscricoes'
print getdate()
-----
-- Avaliacao --
-----

        delete from Avaliacao

        declare @v_TipoAvaliacao nvarchar(3)
        declare @v_DataAvaliacao date
        declare @v_Nota float
        declare @v_MatriculaInscricaoIdId int
        declare @v_NotaEfA float, @v_NotaEfB float, @v_NotaPF float

        declare @random_AvaliacaoContinua float
        declare @random_Recurso float

-- cursor para percorrer matriculas
declare c_matriculas cursor for
SELECT MatriculaInscricaoId
        ,DataEstado
        ,NotaFinal
        ,AlunoNumero
FROM MatriculaInscricao
where notafinal >-1

        open c_matriculas
        fetch next from c_matriculas into @v_MatriculaInscricaoIdId ,
@v_DataEstado, @v_NotaFinal, @v_AlunoNumero
        while @@fetch_status = 0
        begin

                set @random_AvaliacaoContinua = (RAND() * 100) + 1
/*
                70% continua
                30% exame
*/
                set @random_Recurso = (RAND() * 100) + 1
/*
                70% não
                30% sim
*/

```

```

if @random_AvaliacaoContinua > 30.0 -- contínua
begin
    update MatriculaInscricao set TipoAvaliacao = 'A' Where
MatriculaInscricaoId = @v_MatriculaInscricaoIdId
    set @v_NotaEfa = ROUND(@v_NotaFinal * 0.4,1)
    if @v_NotaEfa < 3.5
        set @v_NotaEfa = 3.5
    set @v_NotaPF = @v_NotaFinal - @v_NotaEfa
    set @v_NotaEfB = ROUND(@v_NotaEfa * 0.5,1)
    set @v_NotaEfa = @v_NotaEfa - @v_NotaEfB

    set @v_TipoAvaliacao = 'EF' -- EFOLIO A
    set @v_DataAvaliacao = dateadd(month,-5,@v_DataEstado)
    set @v_Nota = @v_NotaEfa

    exec InsereAvaliacao
    @v_MatriculaInscricaoIdId -- MatriculaInscricaoId
    ,@v_AlunoNumero -- AlunoNumero
    ,@v_TipoAvaliacao -- TipoAvaliacao
    ,@v_DataAvaliacao -- DataAvaliacao
    ,@v_Nota -- Nota

    set @v_TipoAvaliacao = 'EF' -- EFOLIO B
    set @v_Nota = @v_NotaEfB
    set @v_DataAvaliacao = dateadd(month,-4,@v_DataEstado)

    exec InsereAvaliacao
    @v_MatriculaInscricaoIdId -- MatriculaInscricaoId
    ,@v_AlunoNumero -- AlunoNumero
    ,@v_TipoAvaliacao -- TipoAvaliacao
    ,@v_DataAvaliacao -- DataAvaliacao
    ,@v_Nota -- Nota

if @random_Recurso > 30.0 -- sem recurso
begin
    set @v_TipoAvaliacao = 'PF' -- P-FOLIO
    set @v_Nota = @v_NotaPF
    set @v_DataAvaliacao = dateadd(month,-4,@v_DataEstado)

    exec InsereAvaliacao
    @v_MatriculaInscricaoIdId -- MatriculaInscricaoId
    ,@v_AlunoNumero --
AlunoNumero
    ,@v_TipoAvaliacao -- TipoAvaliacao
    ,@v_DataAvaliacao -- DataAvaliacao
    ,@v_Nota -- Nota
end
else -- com recurso
begin
    set @v_TipoAvaliacao = 'PF' -- P-FOLIO
    set @v_Nota = 9.5 - @v_NotaEfa - @v_NotaEfB
    set @v_DataAvaliacao = dateadd(month,-4,@v_DataEstado)

    exec InsereAvaliacao
    @v_MatriculaInscricaoIdId -- MatriculaInscricaoId
    ,@v_AlunoNumero --
AlunoNumero
    ,@v_TipoAvaliacao -- TipoAvaliacao
    ,@v_DataAvaliacao -- DataAvaliacao
    ,@v_Nota -- Nota

    set @v_TipoAvaliacao = 'PFR' -- P-FOLIO RECURSO
    set @v_Nota = @v_NotaPF
    if @v_Nota < 0 set @v_Nota = 0

```

```

set @v_DataAvaliacao = dateadd(month, -4, @v_DataEstado)

exec InsereAvaliacao
@v_MatriculaInscricaoIdId -- MatriculaInscricaoId
, @v_AlunoNumero --
AlunoNumero
, @v_TipoAvaliacao -- TipoAvaliacao
, @v_DataAvaliacao -- DataAvaliacao
, @v_Nota -- Nota

end

end
else -- exame
begin
update MatriculaInscricao set TipoAvaliacao = 'E' Where
MatriculaInscricaoId = @v_MatriculaInscricaoIdId
if @random_Recurso > 30.0 -- sem recurso
begin
set @v_TipoAvaliacao = 'EX' -- EXAME
set @v_Nota = @v_NotaFinal
set @v_DataAvaliacao = dateadd(month, -3, @v_DataEstado)
exec InsereAvaliacao
@v_MatriculaInscricaoIdId -- MatriculaInscricaoId
, @v_AlunoNumero --
AlunoNumero
, @v_TipoAvaliacao -- TipoAvaliacao
, @v_DataAvaliacao -- DataAvaliacao
, @v_Nota -- Nota

end
else -- com recurso
begin
set @v_Nota = ROUND(ABS(Checksum(NewID())) % 9) +
RAND(),1) -- notas de 0 a 9 (Reprovado)
set @v_TipoAvaliacao = 'EX' -- EXAME
set @v_DataAvaliacao = dateadd(month, -3, @v_DataEstado)

exec InsereAvaliacao
@v_MatriculaInscricaoIdId -- MatriculaInscricaoId
, @v_AlunoNumero --
AlunoNumero
, @v_TipoAvaliacao -- TipoAvaliacao
, @v_DataAvaliacao -- DataAvaliacao
, @v_Nota -- Nota

set @v_Nota = @v_NotaFinal
set @v_TipoAvaliacao = 'EXR' -- EXAME RECURSO
set @v_DataAvaliacao = @v_DataEstado

exec InsereAvaliacao
@v_MatriculaInscricaoIdId -- MatriculaInscricaoId
, @v_AlunoNumero --
AlunoNumero
, @v_TipoAvaliacao -- TipoAvaliacao
, @v_DataAvaliacao -- DataAvaliacao
, @v_Nota -- Nota

end

end
end

```

```

        fetch next from c_matriculas into @v_MatriculaInscricaoIdId ,
@v_DataEstado, @v_NotaFinal, @v_AlunoNumero
    end
    close c_matriculas
    deallocate c_matriculas

```

```

-----
-- Perfil Concluido  --
-----

```

```

print 'Fim Avaliações'
print getdate()

```

```

update Perfil set Estado = 'C' where PerfilID in (
select a.PerfilID from MatriculaInscricao a
inner join [UAb-Portal].dbo.UnidadeCurricular b on
a.UnidadeCurricularCodigo=b.UnidadeCurricularCodigo
where a.Estado = 'A'
group by PerfilID
having sum(ECTS) =180)

```

```

update [Perfil] set DataEstado =
(select max(DataEstado) from MatriculaInscricao b WHERE [Perfil].PerfilID =
b.PerfilID and b.Estado = 'A')
WHERE Estado = 'C'

```

```

print 'Fim'
print getdate()

```

END

Procedimento para inserir registo na tabela [Avaliacao]:

```

-- =====
-- Autor:          Nuno Santos
-- Data Criação:  2018-05-25
-- Descrição:     Procedimento para inserir registo na tabelas [Avaliacao]
-- =====

```

```

ALTER PROCEDURE [dbo].[InsereAvaliacao]
    @v_MatriculaInscricaoIdId int
    ,@v_AlunoNumero          int
    ,@v_TipoAvaliacao        nvarchar(3)
    ,@v_DataAvaliacao        date
    ,@v_Nota                 float

```

AS

BEGIN

SET NOCOUNT ON;

```

INSERT INTO [dbo].[Avaliacao]
([MatriculaInscricaoId]
,[AlunoNumero]
,[TipoAvaliacao]
,[DataAvaliacao]
,[Nota])

```

VALUES

```

(
    @v_MatriculaInscricaoIdId -- MatriculaInscricaoId
    ,@v_AlunoNumero           --
AlunoNumero
    ,@v_TipoAvaliacao         -- TipoAvaliacao
    ,@v_DataAvaliacao         -- DataAvaliacao

```

```

        ,@v_Nota
    )
-- Nota

```

END

Código para popular bases de dados transacionais: Tabela [AvaliacaoUnidadeCurricular]:

```

-- =====
-- Autor: Nuno Santos
-- Data Criação: 2018-06-02
-- Descrição: Procedimento para criar dados agregados das notas e estados das UCs
-- =====
ALTER PROCEDURE [dbo].[CriaDadosAvaliacoesUC]
AS
BEGIN
    SET NOCOUNT ON;
    print 'Inicio'
    print getdate()

    delete from [UAb-PortalAcademico].dbo.AvaliacaoUnidadeCurricular

    declare @v_AlunoNumero int, @v_MatriculaInscricaoId int, @v_AnoLectivo
int, @v_CursoID int
    declare @v_UnidadeCurricularCodigo int, @v_TipoAvaliacao nvarchar(1),
@v_contador int
    declare @v_DataEstado date
    declare @v_Estado char
    declare @v_NotaEFolios float, @v_NotaPFolio float, @v_NotaPFolioRecurso
float
    declare @v_NotaExame float, @v_NotaExameRecurso float, @v_NotaFinal float

-- cursor para percorrer matriculas
    declare c_matriculas cursor for
    SELECT M.MatriculaInscricaoId
           ,M.AlunoNumero
           ,M.UnidadeCurricularCodigo
           ,M.Estado
           ,M.DataEstado
           ,M.NotaFinal
           ,M.TipoAvaliacao
           ,M.AnoLectivo
           ,P.CursoID
    FROM MatriculaInscricao M
    inner join Perfil P on M.PerfilID = P.PerfilID

    open c_matriculas
    fetch next from c_matriculas into

@v_MatriculaInscricaoId

, @v_AlunoNumero

,@v_UnidadeCurricularCodigo

,@v_Estado
,@v_DataEstado
,@v_NotaFinal
,@v_TipoAvaliacao
,@v_AnoLectivo
,@v_CursoID

    while @@fetch_status = 0
    begin

```

```

set @v_NotaEFolios = null
set @v_NotaPFolio = null
set @v_NotaPFolioRecurso = null
set @v_NotaExame = null
set @v_NotaExameRecurso = null
set @v_contador = 0

select @v_contador = count(*)
from Avaliacao
Where MatriculaInscricaoId = @v_MatriculaInscricaoId

if @v_contador > 0
    select @v_NotaEFolios = SUM([NotaEFolios]),

@v_NotaPFolio = SUM([NotaPFolio]),

@v_NotaPFolioRecurso = SUM([NotaPFolioRecurso]),

= SUM([NotaExame]),

@v_NotaExameRecurso = SUM([NotaExameRecurso])
from (
Select
MatriculaInscricaoId, SUM(Nota) AS [NotaEFolios], null
[NotaPFolio], null [NotaPFolioRecurso], null [NotaExame], null [NotaExameRecurso]
from Avaliacao
Where TipoAvaliacao = 'EF'
group by MatriculaInscricaoId
union all
select MatriculaInscricaoId, null [NotaEFolios], null
[NotaPFolio], null [NotaPFolioRecurso], SUM(Nota) AS [NotaExame], null
[NotaExameRecurso]
from Avaliacao
Where TipoAvaliacao = 'EX'
group by MatriculaInscricaoId
union all
select MatriculaInscricaoId, null [NotaEFolios], null
[NotaPFolio], SUM(Nota) AS [NotaPFolioRecurso], null [NotaExame], null
[NotaExameRecurso]
from Avaliacao
Where TipoAvaliacao = 'PFR'
group by MatriculaInscricaoId
union all
select MatriculaInscricaoId, null [NotaEFolios], SUM(Nota) AS
[NotaPFolio], null [NotaPFolioRecurso], null [NotaExame], null [NotaExameRecurso]
from Avaliacao
Where TipoAvaliacao = 'PF'
group by MatriculaInscricaoId
union all
select MatriculaInscricaoId, null [NotaEFolios], null
[NotaPFolio], null [NotaPFolioRecurso], null [NotaExame], SUM(Nota) AS
[NotaExameRecurso]
from Avaliacao
Where TipoAvaliacao = 'EXR'
group by MatriculaInscricaoId
) a
where a.MatriculaInscricaoId = @v_MatriculaInscricaoId
--
where a.MatriculaInscricaoId = 2596822
group by MatriculaInscricaoId

INSERT INTO [UAb-
PortalAcademico].dbo.AvaliacaoUnidadeCurricular

```

```

(
[MatriculaInscricaoId],
[CursoID],
[AnoLectivo],
[UnidadeCurricularCodigo],
[AlunoNumero],
[Estado],
[TipoAvaliacao],
[NotaEFolios],
[NotaPFolio],
[NotaPFolioRecurso],
[NotaExame],
[NotaExameRecurso],
[NotaFinal],
[DataEstado]
)

VALUES

-- MatriculaInscricaoId
-- CursoID
-- AnoLectivo
-- UnidadeCurricularCodigo
-- AlunoNumero
-- Estado
-- TipoAvaliacao
-- NotaEFolios
-- NotaPFolio
-- NotaPFolioRecurso
-- NotaExame
-- NotaExameRecurso
-- NotaFinal
-- DataEstado

(
@v_MatriculaInscricaoId,
@v_CursoID,
@v_AnoLectivo,
@v_UnidadeCurricularCodigo,
@v_AlunoNumero,
@v_Estado,
@v_TipoAvaliacao,
@v_NotaEFolios,
@v_NotaPFolio,
@v_NotaPFolioRecurso,
@v_NotaExame,
@v_NotaExameRecurso,
@v_NotaFinal,
@v_DataEstado
)

fetch next from c_matriculas into

@v_MatriculaInscricaoId
,@v_AlunoNumero
,@v_UnidadeCurricularCodigo
,@v_Estado
,@v_DataEstado
,@v_NotaFinal
,@v_TipoAvaliacao
,@v_AnoLectivo
,@v_CursoID

end
close c_matriculas
deallocate c_matriculas

```

```
print 'Fim'  
print getdate()
```

END

ANEXO II – MANUAL DE INSTALAÇÃO/UTILIZAÇÃO

REFERÊNCIAS BIBLIOGRÁFICAS

Website da Universidade Aberta:

www.uab.pt

Website do instituto Nacional de Estatística:

www.ine.pt

Website dos CTT:

www.ctt.pt

Hybrid, End-to-End; Power BI, Azure SQL Database, Data Factory:

<http://radacad.com/hybrid-end-to-end-power-bi-azure-sql-database-data-factory>

InforAberta – V Jornadas de Informática – Estatísticas da Licenciatura em Informática, Luis Caviqne, Montijo, marco 2015

Data Warehouse – Tipos de dimensões - <https://rafaelpiton.com.br/data-warehouse-tipos-dimensoes/>

Noções básicas do DAX no Power BI Desktop:

<https://docs.microsoft.com/pt-pt/power-bi/desktop-quickstart-learn-dax-basics>

DIVIDE Function (DAX):

<https://msdn.microsoft.com/en-us/query-bi/dax/divide-function-dax>

CALCULATE Function (DAX):

<https://msdn.microsoft.com/en-us/query-bi/dax/calculate-function-dax>

COUNTROWS Function (DAX):

<https://msdn.microsoft.com/en-us/query-bi/dax/countrows-function-dax>

Modelos de dados disponibilizados pelo Professor Caviqne (Ministério da Educação):

- DADOS_PESSOAIS_RESUMIDO_ESIGO.pdf (SIGO: Sistema Integrado de Informação e Gestão da Oferta Educativa e Formativa)

- Diagrama_PublicasAlunos.docx (diagrama das escolas públicas alunos)