

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

**ABNER LUIS SANTOS ROLIM
PEDRO ERNESTO FERRONATO LIMA**

SACAD - SISTEMA ACADÊMICO DE APOIO A DECISÃO

Desenvolvimento de sistema de apoio a redução de evasão nas Instituições de Ensino Superior

Florianópolis

2012.

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

**ABNER LUIS SANTOS ROLIM
PEDRO ERNESTO FERRONATO LIMA**

SACAD - SISTEMA ACADÊMICO DE APOIO A DECISÃO

Desenvolvimento de sistema de apoio a redução de evasão nas Instituições de Ensino Superior

Trabalho apresentado à Universidade Federal de Santa Catarina, como requisito para a conclusão do Curso de Graduação de Sistemas de Informação.

Orientador: Prof. Dr. Renato Cislaghi.

Florianópolis

2012.

**ABNER LUIS SANTOS ROLIM
PEDRO ERNESTO FERRONATO LIMA**

SACAD - SISTEMA ACADÊMICO DE APOIO A DECISÃO

Desenvolvimento de sistema de apoio a redução de evasão nas Instituições de Ensino Superior

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Dr. Renato Cislaghi
Orientador

Banca Examinadora

Prof. Dr. José Leomar Todesco

Bel. Esp. José Marcos da Silva

AGRADECIMENTOS

Eu Abner, agradeço aos amigos, pelo incentivo, mesmo que muitas vezes ignorado. Em especial a Thaís, por me emprestar, na reta final, o notebook de sua filha, Sofia, a qual agradeço também. Agradeço ao armazenamento em nuvem, particularmente o Google Drive, sem o qual teríamos perdido o trabalho de tantos anos ao roubarem meu notebook. Agradeço aos envolvidos neste trabalho, principalmente pela paciência e resiliência. Aos professores, pelo apoio dado e pelo conhecimento transmitido.

Eu Pedro, agradeço à minha namorada e amigos que sempre me incentivaram e apoiaram quando necessário. Agradeço minha família, pois sem eles nada disso teria sido possível desde o princípio. Por fim, agradeço aos envolvidos nesse trabalho pela imensa paciência e apoio que nunca foi negado. Aos professores pelo conhecimento e valores transmitidos nesses anos.

Resumo

As instituições de ensino podem ser consideradas um modelo de negócio, visto que os mesmos fatores atuam na sua existência: competitividade e necessidade de manter a saúde financeira. Em contrapartida, nota-se um lapso na utilização de métodos e sistemas nas instituições de ensino. Enquanto grandes empresas utilizam Business Intelligence (BI), as instituições de ensino se restringem a sistemas operacionais. Para atender esse aspecto, o sistema SACAD foi proposto e este trabalho descreve a codificação desse sistema.

O objetivo deste trabalho é codificar um sistema de apoio à decisão (SAD) para responder três perguntas estratégicas propostas por Cislighi (2006):

Quais disciplinas têm maior repercussão no tempo médio para a conclusão de um curso? Quais alunos estão com perspectiva de serem jubilados? Qual é a expectativa de sucesso de um determinado aluno que se matricula em uma determinada carga (conjunto de disciplinas)?

O sistema tem duas grandes partes, o ETL, cuja responsabilidade é receber e importar os arquivos com os dados necessários para o funcionamento do sistema e uma aplicação web, responsável por calcular e apresentar as respostas às perguntas citadas acima.

Dentro do escopo dos cursos de Ciência da Computação e Sistemas de Informação da UFSC, foram importados 189.338 registros na etapa de ETL, tendo um tempo total de 22 minutos. O pré-processamento do relatório de jubramento gerou 632 registros para 2 semestres, processando 316 alunos em 15 segundos. O tempo médio de resposta da funcionalidade de chances de aprovação foi de 812 milissegundos e o tempo médio de resposta da funcionalidade do algoritmo de jubramento foi de 1130 milissegundos.

O sistema foi codificado com sucesso e pode ser utilizado como está por qualquer instituição de ensino superior que exporte os arquivos no layout solicitado pelo sistema ou ainda pode ser utilizado como referência para ser reescrito e/ou melhorado. A documentação do sistema encontra-se no apêndice deste documento.

Conteúdo

1. Introdução	10
1.1. Evasão das Instituições de Ensino Superior	10
1.2. Motivação	11
1.3. Objetivos	11
1.3.1. Objetivos Específicos	11
1.4. Delimitação de Escopo	12
1.5. Limitações	12
2. Fundamentação Teórica	12
2.1. Sistemas de Apoio a Decisão	12
2.2. Data Warehouses e Data Marts	13
3. Trabalhos Relacionados	14
3.1. A aplicação do balanced scorecard para gerenciamento estratégico de uma instituição de ensino superior privada com a utilização de Data Warehouse	14
3.2. Um modelo de sistema de gestão do conhecimento em um framework para a promoção da permanência discente no ensino de graduação	14
4. Desenvolvimento	15
4.1. Coleta e Armazenamento de Dados	15
4.1.1. Tecnologias Envolvidas	15
4.1.2. Ciclo de Vida do Projeto do Data Mart	15
4.2. Aplicação	20
4.2.1. Tecnologias envolvidas	20
4.2.2. Arquitetura da Aplicação	21
4.2.3. Aplicação Web	21
4.2.4. Perguntas Estratégicas	22
5. Resultados	27
5.1. ETL	27
5.1.1. Performance	27
5.1.2. Notificação	28
5.1.3. Relatórios de Monitoração da Importação	29
5.2. Rotina de pré-processamento	30
5.3. Chances de aprovação	30
5.4. Jubilamento detalhado	31
6. Conclusões e trabalhos futuros	32

6.1. Conclusões.....	32
6.2. Trabalhos futuros.....	32
7. Bibliografia.....	34
8. Apêndices.....	36
8.1. Manual Técnico.....	36
8.2. Artigo.....	100

Lista de Figuras

Figura 1 - Ciclo de Vida Dimensional do Negócio (Kimball, 1998)	16
Figura 2 - Ciclo de vida Aplicado ao Projeto - (Autoria própria, adaptação de Kimball, 1998)	16
Figura 3 - Modelagem Macro da Arquitetura - (Autoria própria)	17
Figura 4 - Fluxo de Execução do ETL (Autoria Própria).....	19
Figura 5 - Relatório de Desempenho da Disciplina.....	22
Figura 6 – Relatório de alunos com perspectiva de serem jubilados.....	23
Figura 7 – Relatório detalhado do algoritmo de jubramento	23
Figura 8 - Módulo de chances de aprovação	25
Figura 9 – E-mail de erro na importação (Autoria Própria)	28
Figura 10 – E-mail de sucesso da importação (Autoria Própria)	29
Figura 11 - Relatório de Resumo da Importação (Autoria Própria)	29
Figure 12 - Relatório de Resumo da Importação Detalhado (Autoria Própria)	30

Lista de Tabelas

Tabela 1 - <i>Taxonomy of knowledge (Zeleny 1987, pag.60)</i>	13
Tabela 2 - Importação de Arquivos (Autoria Própria)	27

1. Introdução

A abertura de mercados, fomentada pela globalização, trouxe para os modelos de negócios a necessidade de criar mecanismos eficientes de gerência de seus processos e produção. Em um modelo competitivo, ganha quem apresenta maior eficiência e controle sobre seus custos, antevendo possíveis fatores e interferências. Dado a dinamicidade das variáveis envolvidas, é natural que a computação, com ênfase na área de sistemas de informação, atue como ferramental para a tomada de decisão.

A instituição de ensino, com o objetivo de lucro ou não, pode ser considerada um modelo de negócio, visto que os mesmos fatores atuam na sua existência: competitividade e necessidade de manter a saúde financeira, justificando o investimento realizado. Porém em contraste com instituições comerciais, nota-se a um lapso na utilização e modernização de métodos, sistemas e softwares que auxiliam na gestão da saúde de uma IES. Enquanto em uma empresa a utilização de Business Intelligence (BI) é praxe, nas instituições de ensino o enfoque é diferente, como comentado por Veloso (2001):

[...] o Sistema de Informação Acadêmica, da forma que está construído, caracteriza-se como um aplicativo operacional; cumpre apenas um papel de registro da vida acadêmica do estudante, sem no entanto subsidiar a Instituição, e, mais diretamente, os Coordenadores dos cursos de graduação, para as análises necessárias e estudos de acompanhamento da vida acadêmica dos estudantes. (p. 135)

1.1. Evasão das Instituições de Ensino Superior

Segundo a pesquisa sobre evasão no ensino superior brasileiro, realizada em 2007 pelo Instituto Lobo para o Desenvolvimento da Educação, da Ciência e da Tecnologia, a taxa anual média de evasão no ensino superior brasileiro foi de 22%, com pouca oscilação, mas mostrando tendência de crescimento. A preocupação é refletida por ações do próprio governo que tem como uma das metas globais, aumentar a taxa de conclusão média para 90% nas IES federais, através do Programa de Reestruturação e Expansão das Universidades Federais, Reuni, lançado em 2007.

Tal atenção é cabível, uma vez que um aluno desistente representa um custo. Para as IES de duas formas, a vaga ociosa e o investimento perdido. Para o mercado de trabalho, perde ou adia a entrada de um potencial recurso. O impacto econômico pode ser considerado

também na visão do aluno, que tarda o possível aumento na sua renda, uma vez que a entrada no mercado de trabalho com titulação será adiada.

1.2. Motivação

Considerando os aspectos expostos acima, foi proposto por Cislaghi (2006) um Sistema Acadêmico de Apoio a Decisão, SACAD, descrito pelo autor como um instrumento de informação, produtividade e gestão; com o qual coordenadores (gestores) e discentes possam manusear. Um SAD especializado em fornecer apoio para diminuir a evasão nas IES. Na concepção do autor, este sistema de apoio se baseia em responder três questões básicas (Cislaghi,2006):

Quais disciplinas têm maior repercussão no tempo médio para a conclusão de um curso? Quais alunos estão com perspectiva de serem jubilados? Qual é a expectativa de sucesso de um determinado aluno que se matricula em uma determinada carga (conjunto de disciplinas)?

O presente trabalho descreve o desenvolvimento do sistema proposto pelo artigo, SACAD, nos cursos de Sistemas de Informação e Ciências da Computação.

1.3. Objetivos

Codificar e documentar a produção de um sistema que forneça dashboards, gráficos e projeções que permita responder as questões sugeridas no artigo o qual este sistema se baseia.

1.3.1. Objetivos Específicos

Será codificado um módulo de ETL com layout de entrada definido para permitir que, não só a UFSC, mas qualquer universidade interessada nas visões descritas no artigo e que possua métricas qualitativas e divisões temporais semelhantes, possa, respeitando o layout, importar dados e utilizar o sistema, sem qualquer ajuste.

Também codificaremos uma aplicação Web cuja função será apresentar aos usuários informações que ajudem a responder as três perguntas citadas anteriormente nesta seção.

Finalmente apresentaremos uma documentação compreensiva com o objetivo de explicar o funcionamento interno de todos os módulos do sistema desenvolvido e servir de base para uma implantação em produção

1.4. Delimitação de Escopo

O universo de dados testados e homologados virá de um Data Mart (DM) com informações adquiridas do CAGR, limitando-se aos cursos de Ciências da Computação e Sistema de Informação e considerando o ano 2001/2 como inicial. Serão feitos testes e a homologação será feita pelo orientador, finalizando o escopo deste trabalho. Para um futuro desenvolvimento na UFSC, utilizando tecnologia e processos de desenvolvimento homologado por seu NPD, o sistema contará com um Manual Técnico, onde estarão relatados todos os passos de desenvolvimento e explicações detalhadas da arquitetura, com o intuito de que o futuro desenvolvimento seja apenas uma tradução da aplicação desenvolvida aqui.

1.5. Limitações

Para utilizar esta solução, as IES interessadas devem ter métricas semelhantes a adotadas na UFSC no corrente ano. Ou seja, seus índices de aproveitamento do aluno, notas e outros dados quantitativos devem seguir o mesmo tipo de dado adotado na UFSC. A divisão temporal também deve se manter em dois semestres ao ano. O layout de entrada dos arquivos deve obedecer ao padrão exposto no manual de utilização, presente no apêndice deste trabalho.

2. Fundamentação Teórica

2.1. Sistemas de Apoio a Decisão

Os Sistemas de Apoio a Decisão, SAD, são descritos desde a década de 70, impulsionados por o aumento de competitividade entre as empresas, bem como disponibilidade do processamento de informação (hardware e softwares). Basicamente permitem uma análise estratégica dos dados coletados, fornecendo ferramentas e métodos para apoio a decisão. Naturalmente, embora não obrigatoriamente, se associam ao conceito de Data Warehouse (DW), onde os dados são organizados para permitir uma visão analítica do assunto. Em uma visão macro, segundo Toledo (2005), os Sistemas de Apoio a Decisão são estratégicos e táticos, e os sistemas Transacionais (OLTP) são operacionais. Zeleny (1987) descreve em

sua tabela “Taxonomia do conhecimento”, SAD como sistemas capazes de gerar conhecimento, como podemos observar na tabela 1, representado com sua sigla em inglês, DSS.

	Technology	analogy	Management	Metaphor
Data	EDP	Elements: H ² O, yeast bacteria, starch molecules	Muddling through	KNOW-NOTHING
Information	MIS	Ingredients: Flour, sugar, spices, fixed recipe for bread only (OR/MS) type	Efficiency (Measurement + search)	KNOW-HOW
Knowledge	DSS, ESS, AI	Choose among different recipes of bread	Effectiveness (decision making)	KNOW-WHAT
Wisdom	HSM, MSS	Why bread and not croissant	Explicability (judgment)	KNOW-WHY

Tabela 1 - Taxonomy of knowledge (Zeleny 1987, pag.60)

Podemos, então, afirmar que o uso de uma ferramenta On-line Analytical Processing, OLAP, acessando dados dispostos em um DW, se bem construído, permite o apoio a tomada de decisão, constituindo um SAD.

2.2. Data Warehouses e Data Marts

Em uma definição formal (Inmon, 2002), um DW "é uma coleção de dados orientados à assuntos, integrados, variáveis no tempo e não voláteis que dão suporte ao processo de tomada de decisão (pag. 31)". Em uma definição mais prática e comercial, Kimball (1998) resume como um recurso de apresentação dos dados de uma empresa que permite a formulação de consultas. Destaca também que estes dados não devem ser organizados em torno de um modelo de entidades relacionais, pois visa a legibilidade da informação e performance. Kimball, com sua interpretação bottom-up da constituição de um Data Warehouse, também resume um Data Warehouse como a união de todos Data Marts constituintes.

Um DM, por sua vez, é um subconjunto de um DW, geralmente projetado para servir especificamente um grupo de interessados e tem como fonte um assunto/setor também específico. Na visão de Kimball, é a base para constituir o Data Warehouse. Porém, para Inmon, os DM seriam alimentados por sumarizações dos dados disponíveis em um Data Warehouse (visão top-down). Portanto, um DM pode depender ou não de um DW para existir. Em nosso trabalho, utilizamos a abordagem bottom-up, criando um DM inicial, dado que nosso objetivo era bastante específico.

Implementado geralmente em uma modelagem dimensional de dados, onde as medidas relativas à um assunto são representada por uma tabela fato (vendas, por exemplo) que possui chaves primárias para suas dimensões, tabelas que representam as entidades envolvidas (cliente, dimensão temporal, produto, etc.). Nas tabelas dimensionais, também é possível encontrar atributos qualitativos, com o intuito de facilitar o agrupamento. A junção

entre fato/dimensão produz grânulos quantitativos, agrupados pelas hierarquias das dimensões envolvidas. Por exemplo, o total de vendas (tabela fato) agrupado por mês, de um determinado tipo de produto (perecíveis, não-perecíveis) em relação à estação do ano (verão, outono, inverno e primavera).

Utilizando uma função simples de avaliação da adaptação de determinado indivíduo ao meio ambiente, era possível selecionar os mais adaptados para reprodução, similar ao processo que acontece naturalmente. A partir deste algoritmo era possível resolver problemas complexos de uma forma relativamente simples, e assim como a natureza, o algoritmo não faz ideia do tipo de problema que está sendo resolvido.

3. Trabalhos Relacionados

3.1. A aplicação do balanced scorecard para gerenciamento estratégico de uma instituição de ensino superior privada com a utilização de Data Warehouse

O trabalho propõe o uso de um modelo de balanced scorecards (BS) para constituir um SAD para gestão de uma instituição de ensino privada. O trabalho propõe a estrutura do BS, elaboração de um Data Mart que armazene as informações necessárias e um sistema para controle de indicadores que não estejam presentes nos sistemas legados da instituição.

3.2. Um modelo de sistema de gestão do conhecimento em um framework para a promoção da permanência discente no ensino de graduação

A tese de doutorado trabalha a ideia da construção de um framework completo que promova a permanência discente, utilizando KPI's, sensores e procedimentos levantados estudando as causas da evasão através de abrangentes teorias e modelos de desgaste.

4. Desenvolvimento

4.1. Coleta e Armazenamento de Dados

Para prover as informações a serem analisadas pela proposta do trabalho, escolhemos aplicar a metodologia bottom-up, desenvolvendo um DM que pudesse prover a estrutura necessária à resposta. Os arquivos necessários para montagem do banco, bem como explicação detalhada do funcionamento dos passos do ETL estão contidos no Manual Técnico, na sessão de Apêndice.

4.1.1. Tecnologias Envolvidas

A ferramenta de ETL escolhida para o projeto foi o Pentaho Data Integration (PDI), anteriormente chamado de Kettle, na versão 4.1 Community. O PDI é open source e freeware. A escolha se deu também por conhecimento prévio na utilização da ferramenta.

Em todo o projeto, foi utilizado o MySQL 5.5.16 para o SGDB, incluindo na camada de stage do ETL. A modelagem do banco foi feita utilizando o MySQL Workbench 5.2.35 Community Edition.

A administração do ETL, bem como visualização de relatórios de importação, é feita através do Pentaho BI Server, na sua versão Community 3.10.0. Dessa forma é possível ter uma solução integrada e permite que mais relatórios ou novas visões dimensionais sejam adicionadas.

4.1.2. Ciclo de Vida do Projeto do Data Mart

No desenvolvimento de nosso DM, seguimos algumas orientações propostas por Kimball em seu livro *The Data Warehouse Life Cycle Toolkit* (1998), porém, por escopo de desenvolvimento e dimensão da equipe, não houve o paralelismo proposto por ele. Algumas fases foram resumidas, pois grande parte do levantamento de requisitos ou expectativas já era definida pelo artigo o qual o trabalho se baseia. Ou são abreviadas pelo tamanho do mesmo. O ciclo de vida de um DM, segundo Kimball, é definido por as seguintes fases, dispostas no diagrama da Figura 1.

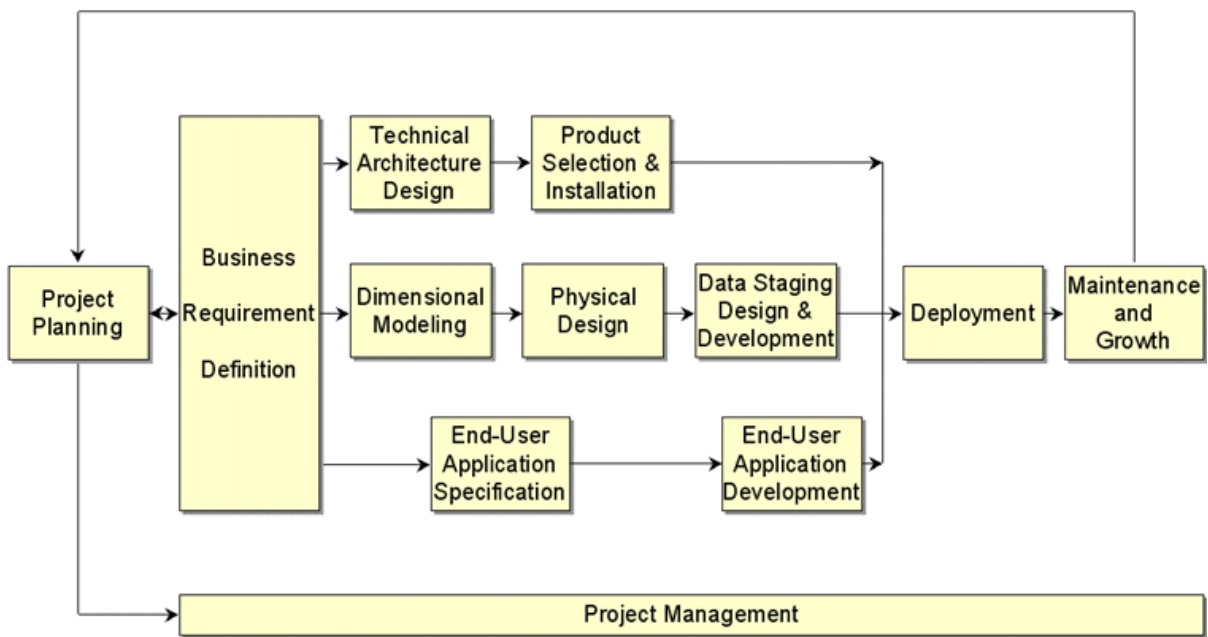


Figura 1 - Ciclo de Vida Dimensional do Negócio (Kimball, 1998)

No entanto, o ciclo de vida de nosso DM baseou-se no seguinte diagrama da Figura

2.

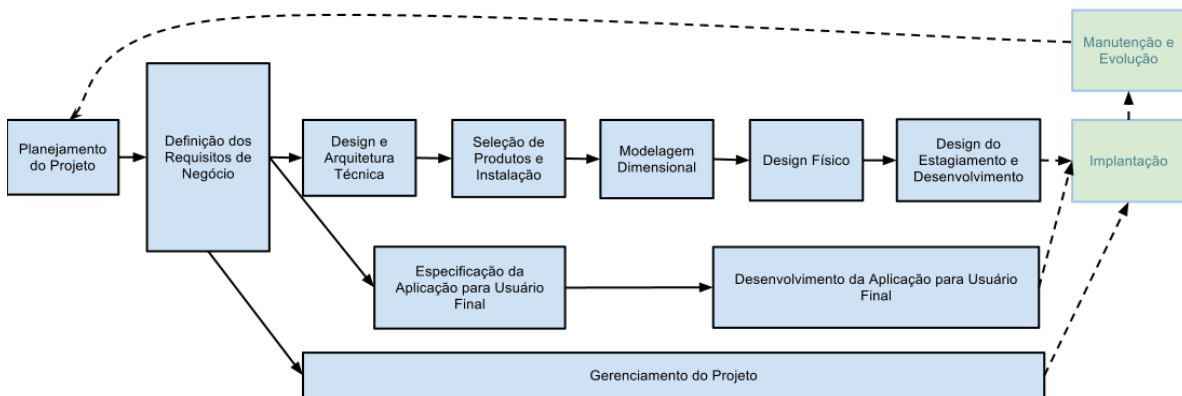


Figura 2 - Ciclo de vida Aplicado ao Projeto - (Autoria própria, adaptação de Kimball, 1998)

a) Planejamento do Projeto: o planejamento do projeto envolveu o autor do artigo SACAD, os autores deste trabalho e o gestor da ferramenta CAGR, de onde viriam as informações. Nesta fase, os papéis já estavam pré-definidos e a motivação, benefícios entre outras justificativas, explícita pelo embasamento teórico do trabalho. Dado ao tamanho da equipe e o tempo de desenvolvimento, excluiu-se outras universidades e foi definido o escopo para os dois cursos mencionados. O critério de sucesso seria conseguir alimentar o DM e prover as análises mencionadas no artigo-base, provendo tecnologia e conhecimento para uma futura implantação na UFSC. Como risco mapeado, levantou-se o fato dos

- Relatórios de Suporte: dois relatórios de suporte ao ETL seriam desenvolvidos e sendo acessíveis através de um filtro por data da importação. O primeiro representaria uma sumarização da importação e o segundo um detalhamento das linhas importadas (*drill-down*). Estes relatórios podem ser observados no Manual Técnico, no apêndice.

Um detalhamento maior da arquitetura selecionada pode ser observado no Manual Técnico.

d) Seleção de Produtos e Instalação: não houve estudo sobre outros produtos, a equipe já era familiarizada aos produtos da suíte Pentaho e Mysql. A curva de aprendizado e o tempo para pesquisa de outras ferramentas não compensaria o possível ganho.

e) Modelagem Dimensional: esta etapa não apresentou grandes problemas, visto que o time desenvolvedor tinha certa experiência em modelagem. Houve a preocupação da conformidade na modelagem das dimensões para possíveis integrações com outros DM's. Cada mudança do currículo de um curso ou mudança de curso pelo aluno é tratado como uma entidade nova, portanto não foi necessário utilizar nenhuma técnica de *slowly changing dimension*. O modelo completo da modelagem dimensional pode ser visto no Manual Técnico.

f) Design Físico: como agregações, para otimizar as consultas, os fatos que representavam um aluno foram agregados em desempenho por disciplina e desempenho por semestre. O fatos de um currículo também foi agregado por curso e especializado por disciplina. Particionamento das tabelas, tabelas temporárias e plano de indexação não foram identificados como necessários, porém certamente é uma sessão a ser revista no futuro, se em produção. Nesta etapa também criou-se tabelas de meta-dados, para auxiliar nas consultas. Nomeadas como Tabelas de Mapeamento, indicam como uma disciplina se relaciona com outra, em determinado currículo. Esses relacionamentos podem ser de equivalência, pré-requisito, cursadas em conjunto, entre outros.

g) Design da Staging Area e Desenvolvimento: para a staging area, produziu-se tabelas de logs. Estas tabelas foram nomeadas com o sufixo matriz e representam o exato conteúdo do arquivo lido. Há uma tabela matriz para cada arquivo. Além das colunas do arquivo, cada tabela contém um id de importação, que agrega os arquivos importados conjuntamente, uma coluna de id de status e uma coluna da descrição deste status. Nestas duas últimas colunas é feito todo controle de validação léxico-sintática e semântica. Não foi definido rotinas de expurgo, já que um dos desejos era servir de reconstrução para o DM, se necessário. Cabe aqui comentar que poderia ter sido utilizado tabelas de meta-dados, para rastreamento da origem e operações aplicadas nos dados, mas pelo escopo definido, foi criado somente duas

tabelas de meta-dados, a de importação, com dados gerais dos arquivos envolvidos, data e usuário que executou a importação e uma da relação arquivo-tabela, para auxiliar nos relatórios de controle de importação. Por ter uma fonte única de dados, não foi necessário adequação de siglas, códigos, entre outros. Para evitar linhas duplicadas, foi criada uma validação na staging area que permitiu identificar a linha duplicada, registrando na sua coluna de controle como erro de duplicidade. Com o modelo dimensional previamente definido, a carga das dimensões não representou problemas. Houve a necessidade de criar uma verificação se a entidade da dimensão, referida no arquivo à ser importado, já existia. Dessa forma seria possível informar corretamente o motivo de não importar esta entrada. Em linhas gerais, o fluxo de execução é representado pela figura 4.

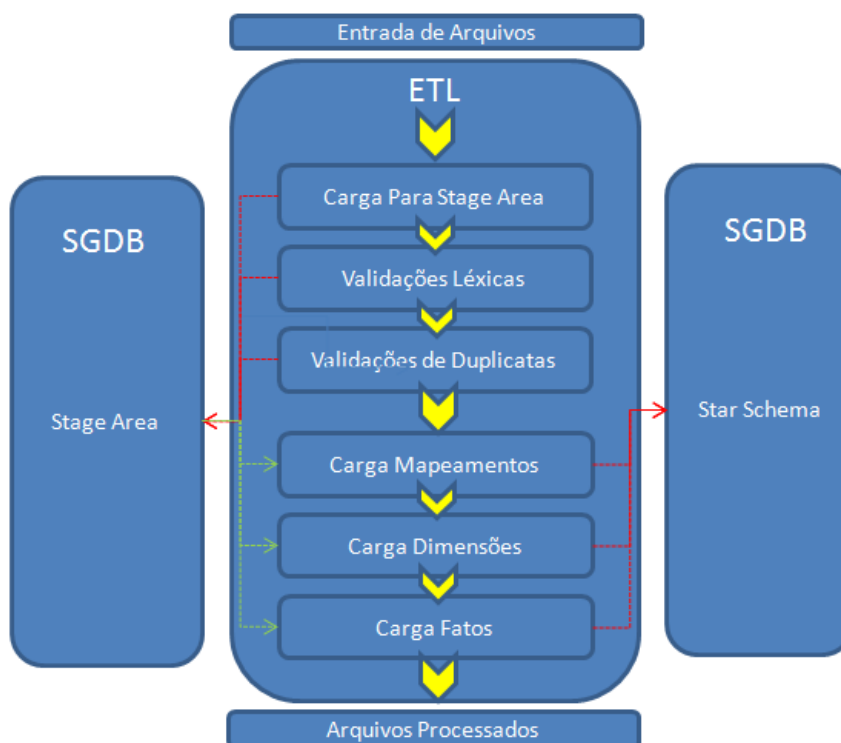


Figura 4 - Fluxo de Execução do ETL (Autoria Própria)

Todos os passos do ETL e explicações detalhadas das estratégias aplicadas estão aprofundados no Manual Técnico, disponível na sessão de Apêndice.

h) Especificação da Aplicação para Usuário Final e Desenvolvimento da Aplicação para Usuário Final: estas duas fases são descritas pelo capítulo 3.2 deste trabalho.

i) Implantação e Manutenção e Evolução: estas duas fases não fizeram parte do escopo deste trabalho.

j) Gerenciamento do Projeto: talvez a fase mais importante, porém subestimada pelos autores. A gerência se deu em reuniões entre os *sponsors* e os autores do trabalho. Porém nenhum documento foi gerado destas reuniões e os prazos frequentemente eram estourados.

4.2. Aplicação

A documentação técnica complementar da aplicação do usuário final está disponível no Manual Técnico. A seguir será apresentado um resumo das tecnologias envolvidas e arquitetura adotada.

4.2.1. Tecnologias envolvidas

A plataforma de desenvolvimento escolhida foi a Microsoft .NET, sendo utilizado o .NET Framework na sua versão 4.0. A linguagem de programação escolhida foi o C# 4.0. A especificação do C# é aberta e o .NET Framework é gratuito.

A IDE utilizada para desenvolver a aplicação é o Visual Studio 2010 Ultimate, da Microsoft, no sistema operacional Windows 7 Enterprise, ambos obtidos através de licença acadêmica disponibilizada pela UFSC.

O SGBD escolhido foi o MySQL 5.5.16. A ferramenta de acesso ao banco de dados para administração e consultas ad-hoc utilizada foi o MySQL Workbench 5.2.35 Community Edition. Ambos são gratuitos.

Para acesso ao SGBD através da aplicação foi utilizado o componente Dot Connect for MySql 6.80 Express, que é gratuito.

A escolha das ferramentas se deu por afinidade dos envolvidos com as mesmas, com o objetivo de focar o desenvolvimento nas regras de negócio do sistema e não em aprender novas tecnologias. Foram utilizadas licenças acadêmicas disponibilizadas pela UFSC, quando aplicável.

4.2.2. Arquitetura da Aplicação

A parte da solução que é responsável por exibir as informações ao usuário final consiste em uma aplicação web desenvolvida no framework web ASP.NET versão 4.0 na linguagem C#. O serviço responsável pelo pré-processamento de informações é um executável cujo ambiente de execução é o .NET 4.0.

4.2.3. Aplicação Web

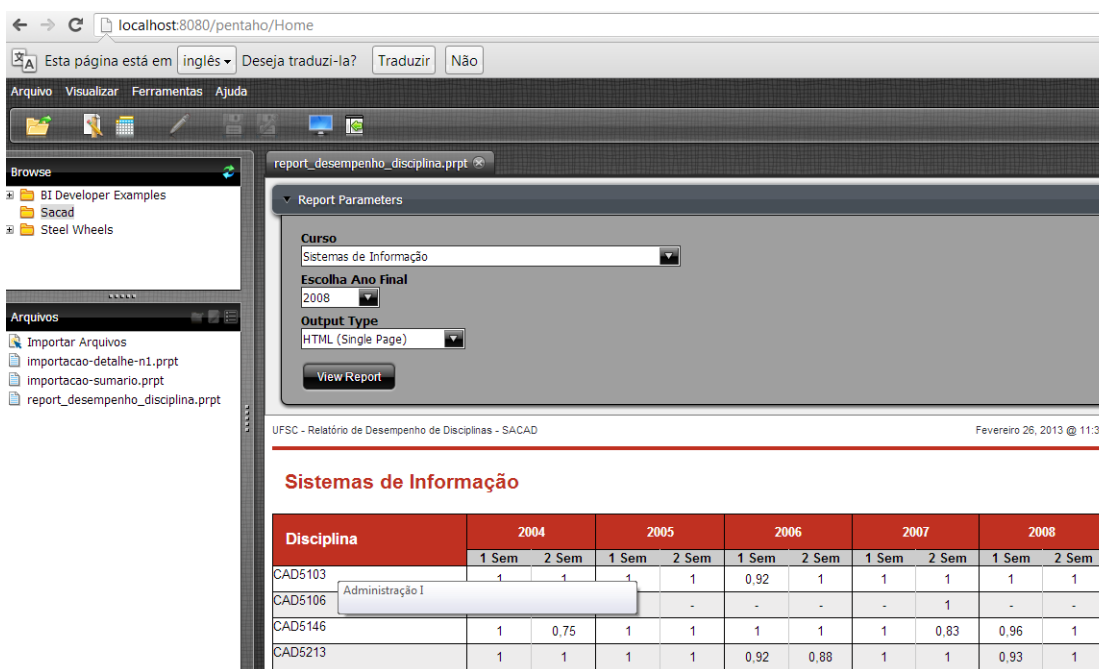
Para o desenvolvimento da aplicação web utilizou-se o padrão MVC para separação da lógica de interface, controle e modelo. Os testes unitários e a camada de pré-processamento não fazem parte da aplicação web.

4.2.4. Perguntas Estratégicas

Conforme já citado, o sistema se propõe a responder três perguntas estratégicas a respeito da gestão acadêmica. Nesta seção será detalhado como se deram a concepção e codificação dos algoritmos para responder as três perguntas.

a) Primeira Pergunta: *Quais disciplinas têm maior repercussão no tempo médio para a conclusão de um curso?*

Essa pergunta tem como chave para a resposta o IAD de cada disciplina, atualmente calculado levando em consideração o percentual de alunos matriculados que são aprovados, excluindo-se os desistentes. Como essa métrica é toda calculada pelo ETL, foi colocado um relatório diretamente no Pentaho Bi Server para exibir este relatório. Como filtro do relatório temos o curso e maior ano que queremos analisar, em um período de 5 anos (ano selecionado - 5). Abaixo podemos observar o relatório exibido na figura 5.



UFSC - Relatório de Desempenho de Disciplinas - SACAD

Fevereiro 26, 2013 @ 11:35

Sistemas de Informação

Disciplina	2004		2005		2006		2007		2008	
	1 Sem	2 Sem	1 Sem	2 Sem	1 Sem	2 Sem	1 Sem	2 Sem	1 Sem	2 Sem
CAD5103	1	1	1	1	0,92	1	1	1	1	1
CAD5106	Administração I		-	-	-	-	1	-	-	-
CAD5146	1	0,75	1	1	1	1	1	0,83	0,96	1
CAD5213	1	1	1	1	0,92	0,88	1	1	0,93	1

Figura 5 - Relatório de Desempenho da Disciplina

b) Segunda Pergunta: *Quais alunos estão com perspectiva de serem jubilados?*

Essa pergunta é respondida combinando duas etapas no sistema. A primeira, chamada de pré-processamento, consiste em realizar a execução completa do algoritmo do jubramento para cada aluno e persistir informações importantes que depois servirão de entrada para um relatório que consolida e destaca os alunos com situação mais crítica,

conforme exemplo na Figura 6. A segunda compreende o detalhamento da saída do algoritmo de jubramento para cada aluno, conforme exemplo na Figura 7.

SACAD							
Jubilamento Consolidado		Jubilamento Individual	Índice Aprovação de Disciplinas	Chances de Aprovação Aluno			
Sistemas de Informação <input type="text"/> Pesquisar							
Aluno	Média Créditos p/ Semestre	Créditos Restantes (obrigatórias)	Créditos Restantes (optativas)	Semestre Provável de Formatura	Prazo p/ Formatura	Previsão de Jubilar	
0	161	16	16	2034/1	2017/2	Sim	detalhes...
1	155	16	16	2032/2	2016/2	Sim	detalhes...
6	101	16	16	2018/2	2015/2	Sim	detalhes...
8	100	8	8	2016/1	2014/2	Sim	detalhes...
7	99	12	12	2017/2	2015/2	Sim	detalhes...
6	99	14	14	2018/1	2015/2	Sim	detalhes...
6	99	16	16	2018/1	2015/1	Sim	detalhes...
6	98	12	12	2017/2	2015/1	Sim	detalhes...
6	97	14	14	2018/1	2014/1	Sim	detalhes...
6	94	8	8	2017/2	2015/1	Sim	detalhes...
6	92	16	16	2017/2	2014/2	Sim	detalhes...
7	91	2	2	2016/2	2014/2	Sim	detalhes...
5	87	16	16	2018/1	2012/2	Sim	detalhes...
8	85	12	12	2016/1	2015/2	Sim	detalhes...
7	82	12	12	2016/2	2014/2	Sim	detalhes...
8	82	10	10	2016/1	2013/2	Sim	detalhes...
8	77	4	4	2015/1	2014/2	Sim	detalhes...
9	77	14	14	2016/1	2015/2	Sim	detalhes...
8	77	16	16	2015/2	2014/2	Sim	detalhes...

Figura 6 – Relatório de alunos com perspectiva de serem jubilados

SACAD					
Jubilamento Consolidado		Jubilamento Individual	Índice Aprovação de Disciplinas	Chances de Aprovação Aluno	
Histórico					
Tipo	Semestre	Cr.	Disciplina	IAD	
Obr	2008/1	4	Introdução à Informática	0,94	
Obr	2008/1	4	Teoria Geral de Sistemas	1	
Obr	2008/1	4	Fundamentos Matemáticos da Informática	1	
Obr	2008/1	2	Aspectos Comportamentais do Empreendedor	1	
Obr	2008/1	6	Introdução à Programação Orientada a Objetos	1	
Obr	2008/2	6	Desenvolvimento de Sistemas Orientados a Objetos I	1	
Obr	2008/2	2	Marketing Pessoal em Informática	1	
Obr	2008/2	4	Organização e Arquitetura de Computadores	0,83	
Obr	2008/2	4	Administração I	1	
Obr	2008/2	4	Probabilidade e Estatística	0,81	
Ritmo Aluno					
Tipo	Semestre	Cr.	Disciplina	IAD	
Obr	2012/1	4	Técnicas Estatísticas de Predição	0,92	
Obr	2012/1	6	Estruturas de Dados	1	
Obr	2012/2	3	Programação Econômica e Financeira	0,71	
Obr	2012/2	3	Geração de Idéias e Criatividade em Informática	1	
Obr	2013/1	3	Gestão Estratégica de Tecnologia, Informação e Comunicação	1	
Obr	2014/2	2	Informática e Sociedade	0,78	
Opt	2016/2	2	Tópicos Especiais em Software Aplicativo I	1	
Opt	2016/2	2	Temas em psicologia: Orientação e Planejamento de Carreira	1	
Opt	2016/2	4	Comércio Eletrônico	1	
Opt	2017/1	4	Tecnologia da Informação e Governança	0,38	
Opt	2017/1	4	Qualidade da Informação	1	

Figura 7 – Relatório detalhado do algoritmo de jubramento

O cálculo do jubramento é feito nos seguintes passos:

- 1 A partir de um aluno são buscadas as disciplinas cursadas e as disciplinas restantes de acordo com o currículo do aluno;
- 2 Com as disciplinas cursadas é verificada qual a média de créditos aprovados por semestre;
- 3 A partir das disciplinas restantes é criada uma simulação onde o aluno cursaria uma quantidade de créditos por semestre correspondente a sua própria média, sempre obedecendo aos pré-requisitos de cada disciplina;
- 4 Tendo essa simulação criada já se tem qual o semestre provável de formatura e pode-se responder se esse aluno tem perspectiva de ser jubilado;
- 5 Também é simulado um caminho ideal, o mais rápido possível, obedecendo à cadeia de pré-requisitos;
- 6 Com todas essas informações um resumo é montado indicando: créditos restantes para obrigatórias e optativas, média de créditos por semestre, prazo real para se formar, o semestre provável de formatura e dessa maneira indica se o aluno tem perspectiva de jubilar ou não.

c) Terceira pergunta: *Qual é a expectativa de sucesso de um determinado aluno que se matricula em um determinado conjunto de disciplinas?*

Para responder essa pergunta foi criada uma página na aplicação web que receberá como entrada a matrícula de um aluno e, opcionalmente, as disciplinas desejadas para um determinado semestre.

O caso de receber somente a matrícula serve para situações onde a integração com o sistema acadêmico vigente for fraca, ou seja, o aluno informará manualmente sua matrícula em dois sistemas diferentes ou ainda o sistema acadêmico enviaria somente a matrícula como um parâmetro em um hyperlink. Quando somente a matrícula for informada o aluno deve selecionar as disciplinas desejadas.

O caso de já receber o conjunto de disciplinas escolhidas no semestre tem como premissa uma integração mais completa entre o sistema SACAD e o sistema acadêmico em utilização e exigirá uma provável customização dos dois sistemas. A Figura 8 mostra um exemplo do módulo em funcionamento.

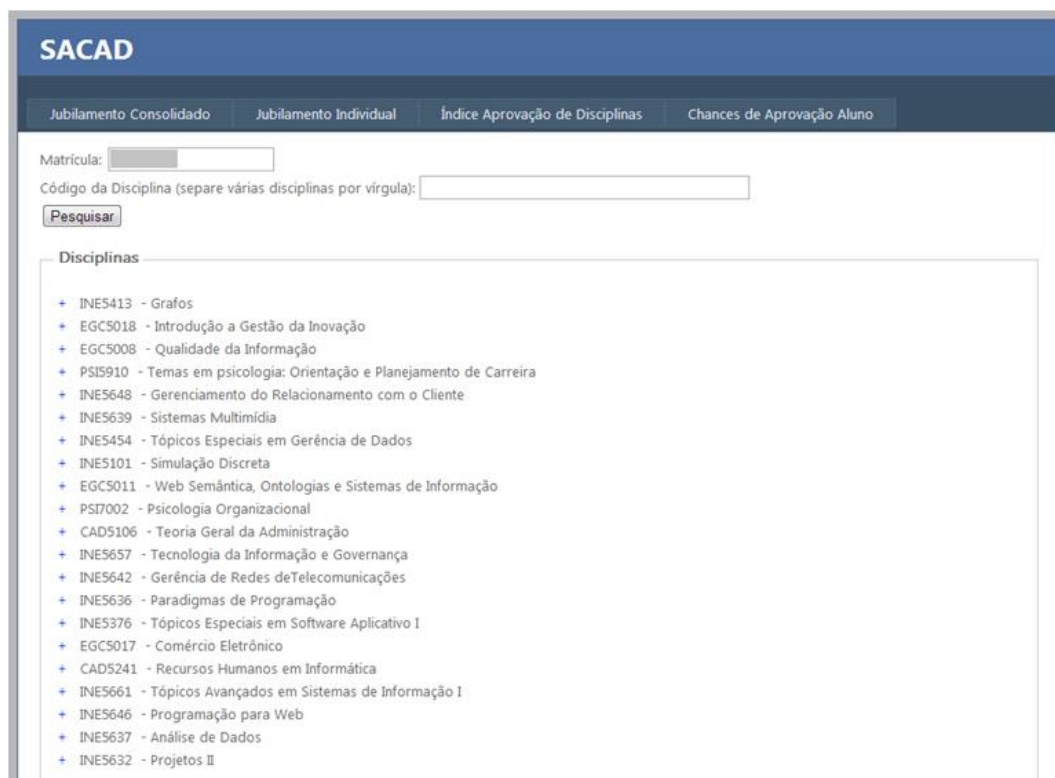


Figura 8 - Módulo de chances de aprovação

A resposta será dada, em alto nível, pelos seguintes passos:

- 1 A partir de um aluno e um conjunto de disciplinas, busca-se no histórico outros alunos em que as mesmas disciplinas tenham sido cursadas em um mesmo semestre;
- 2 Separa-se desse primeiro grupo os alunos cujo IAA esteja num intervalo entre mais e menos cinco por cento em relação ao IAA do aluno sendo avaliado;
- 3 Caso o número de casos encontrados não seja satisfatório, sendo este valor parametrizável pelo administrador do sistema, os procedimentos alternativos chamados de prótese e desbaste devem ser utilizados (explicados adiante);
- 4 Tendo separados os casos satisfatórios, devem-se calcular dois índices:
 - a O percentual de aprovações considerando o conjunto de disciplinas selecionadas em relação a todos os casos encontrados;
 - b O percentual de aprovações para cada disciplina isoladamente.
- 5 Os resultados são mostrados tanto individualmente (para cada disciplina) quanto para o conjunto delas.

Conforme dito no passo 3, os seguintes procedimentos alternativos são tomados caso não sejam encontrados casos suficientes para a execução do algoritmo. Segundo Cislaghi(2006), sua definição é dado por:

Prótese: localizar alunos que tenham cursado num mesmo semestre o mesmo número de disciplinas pretendidas, sendo que somente $n-1$ delas correspondem às disciplinas selecionadas. A seguir, para cada caso encontrado, substituir a disciplina que não pertence à seleção considerada pelo IAD da disciplina faltante, correspondente aquele determinado semestre. Este mecanismo foi denominado de prótese.

Desbaste: localizar alunos que tenham cursado num mesmo semestre $n+1$ disciplinas, sendo que somente n delas correspondem às disciplinas selecionadas. A seguir, desde que o aluno que a cursou tenha sido aprovado nesta disciplina, neste mesmo semestre (posição conservadora), considerar o caso como válido para os demais passos, embora desconsiderando para efeito de tratamento a disciplina não pertencente ao conjunto originalmente considerado para a análise. Este mecanismo foi denominado de desbaste.

A interface do sistema permitirá a seleção das disciplinas desejadas, excetuando-se as disciplinas já cursadas e aprovadas. O sistema nessa funcionalidade desempenha uma atitude mais passiva e não faz validações como cadeias de pré-requisitos no momento da seleção das disciplinas ou se a carga horária mínima ou máxima do curso é excedida, esse tipo de regra de negócio compete ao sistema acadêmico efetivo da universidade.

5. Resultados

5.1. ETL

A fase de ETL foi cumprida com sucesso, gerando uma estrutura robusta e modularizada. Houve uma preocupação de separar as responsabilidades e reusar transformações em comum em todo o desenvolvimento. A utilização de tecnologias de uso gratuito possibilita a inclusão de mais IES que venham a ter interesse em utilizar a ferramenta.

5.1.1. Performance

Os sete arquivos definidos como entrada (ver Manual Técnico) forneceram, limitados ao escopo deste trabalho, 189.338 registros, sendo importados em 22 minutos e 22 segundos. As medidas foram tomadas em um computador com a seguinte configuração: Intel Core i3-M330 com 4 GB de memória ram a 1066 MHz e um HD Sata II 7200 RPM. O detalhamento desta importação pode ser visto abaixo, na tabela 2.

Arquivo	Linhas	OK	Erros
Requisitos_13-11-2012.txt	2.926	2.900	26
DisciplinasEquivalentes_13-11-2012.txt	2.107	813	1.294
DadosAlunoCurso_13-11-2012.txt	6.055	5.703	352
Curriculo_13-11-2012.txt	3.308	3.220	88
AlunoDesempenhoSemestre_09-11-2012.txt	172.940	162.882	10.058
CurriculoParametros_13-11-2012.txt	34	25	9
DisciplinasRelacionadas_13-11-2012.txt	1.968	1.968	0

Tabela 2 - Importação de Arquivos (Autoria Própria)

Os registros de erro se referem a entidades duplicadas, identificadores não referenciados em outros arquivos e erros léxicos.

5.1.2. Notificação

Simulou-se uma exceção na execução do ETL, como conexão do SGDB perdida, para observar o fluxo de notificação de erro. A simulação gerou na caixa de e-mail configurada no arquivo de configurações, o seguinte e-mail de erro, observado na figura 9.



Figura 9 – E-mail de erro na importação (Autoria Própria)

Para os casos de sucesso, o sistema informou que houve erros nas validações, mas que a importação dos arquivos foi concluída com sucesso. O e-mail também contém o hiperlink para o relatório de importação. Na figura abaixo (figura 10) observamos que a mensagem não respeitou a formatação de quebra de linha, no entanto considerou-se o item concluído.



Importação 2013/02/17 19:01:28.000 finalizada com Sucesso

1 mensagem

SACAD Importação <sacad.importacao@gmail.com>
Responder a: sacad.importacao@gmail.com:
Para: sacad.importacao@gmail.com:

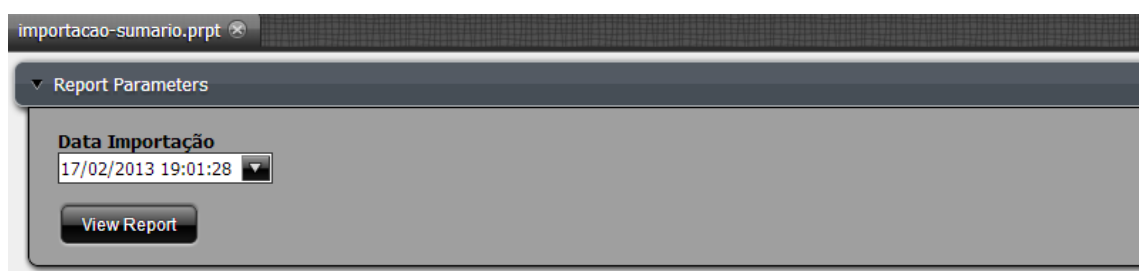
17 de fevereiro de 2013 19

A importação ocorreu sem erros bloqueantes. Porém é possível que erros de duplicação ou léxicos ocorram. Verifique o relatório de importação e os possíveis erros abaixo: **Resumo Importação**
----- Houve registros duplicados em outras importações. Arquivos importados e movidos para: C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solutions\SACAD\ETL\ArquivosImportados.....
Relatório acessível em: **Relatório Integração 2013/02/17 19:01:28.000** Message date: 2013/02/17 19:23:52.525 Contact information : Person to contact : Abner Luis Telephone number : [EMAIL.FONERESPSTA]

Figura 10 – E-mail de sucesso da importação (Autoria Própria)

5.1.3. Relatórios de Monitoração da Importação

Como ferramenta de verificação da importação, foram criados dois relatórios. Um sumarizado, com o resumo de cada arquivo de uma determinada importação e outro o detalhamento das linhas que resultaram em erro ou sucesso. Abaixo podemos ver o primeiro relatório de importação. Como filtro, temos as datas das importações realizadas, como podemos ver na figura 11.



UFSC - Importação Cubo SACP

Fevereiro 19, 2013 @ 09:35

Resumo Importação

Data da importação: 17/02/2013 19:01:28

Arquivo	Diretório Origem	Linhas	OK	Erros	Autor
Requisitos_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	2.926	2.900	26	-
DisciplinasEquivalentes_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	2.107	813	1.294	-
DadosAlunoCurso_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	6.055	5.703	352	-
Curriculo_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	3.308	3.220	88	-
AlunoDesempenhoSemestre_09-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	172.940	162.882	10.058	-
CurriculoParametros_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	34	25	9	-
DisciplinasRelacionadas_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	1.968	1.968	-	-

Figura 11 - Relatório de Resumo da Importação (Autoria Própria)

O relatório detalhado é acessível através dos consolidados das colunas OK e Erros, utilizando a técnica de drill-down. Abaixo podemos ver este detalhamento na figura 12.

Resumo Importação

Matriz referência: requisitosmatriz

Id Matriz	Id Histórico	Linha Arquivo	Tipo Status	Descrição do Status
15	375	15	4	Registros duplicado em outras importações
91	375	91	7	Referência inválida. Uma das chaves não foi previamente cadastrada. (iddisciplina=ECV5352 em map_disciplina_requisito)
99	375	99	4	Registros duplicado em outras importações
162	375	162	4	Registros duplicado em outras importações
238	375	238	4	Registros duplicado em outras importações
254	375	254	4	Registros duplicado em outras importações
290	375	290	4	Registros duplicado em outras importações
736	375	736	4	Registros duplicado em outras importações
744	375	744	4	Registros duplicado em outras importações
825	375	825	4	Registros duplicado em outras importações
1.130	375	1.130	4	Registros duplicado em outras importações
1.223	375	1.223	4	Registros duplicado em outras importações
1.477	375	1.477	4	Registros duplicado em outras importações
1.623	375	1.623	4	Registros duplicado em outras importações
1.761	375	1.761	7	Referência inválida. Uma das chaves não foi previamente cadastrada. (iddisciplina=EEL5166 em map_disciplina_requisito)

Figure 12 - Relatório de Resumo da Importação Detalhado (Autoria Própria)

5.2. Rotina de pré-processamento

Conforme comentado anteriormente, o relatório consolidado de alunos em perigo de jubramento precisa ser pré-processado para todos os alunos, pois seria inviável em termos de desempenho processar o algoritmo de jubramento todas as vezes que fosse necessário visualizar esse relatório.

Assim, a rotina de pré-processamento processou 2 semestres dos 316 alunos disponíveis para testes em 15 segundos. Esse pré-processamento é realizado apenas uma vez, e o relatório que o consome leva em média 340 milissegundos para processar e exibir os dados.

Essas medições foram tomadas em um computador com a seguinte configuração: Intel Core i5-560m com 4 GB de memória ram a 1066 MHz e um HD Sata II 7200 RPM.

5.3. Chances de aprovação

Para medir se a funcionalidade de chances de aprovação estava com um desempenho aceitável, foi criado o seguinte caso de teste: separa-se uma matrícula aleatória, e para essa matrícula é simulado a inscrição em 4 disciplinas, em seguida 5 disciplinas e por fim 3 disciplinas. É feito uma média do tempo de processamento das três interações. Colhemos 50 amostras desse teste e se obteve o tempo médio de resposta de 812 milissegundos. O tempo de processamento é medido através do log de requisições do IIS 7, servidor web no qual o sistema roda.

Essas medições foram tomadas em um computador com a seguinte configuração: Intel Core i5-560m com 4 GB de memória ram a 1066 MHz e um HD Sata II 7200 RPM.

5.4. Jubilamento detalhado

De maneira semelhante a funcionalidade de chances de aprovação, sorteamos uma matrícula e abrimos o relatório, medindo o tempo de processamento. Com 50 amostras, obteve-se a média de 1130 milissegundos por requisição.

Essas medições foram tomadas em um computador com a seguinte configuração: Intel Core i5-560m com 4 GB de memória ram a 1066 MHz e um HD Sata II 7200 RPM.

6. Conclusões e trabalhos futuros

6.1. Conclusões

O sistema foi desenvolvido conforme especificado por Cislaghi (2006). O funcionamento do sistema foi validado utilizando dados dos cursos de Sistemas de Informação e Ciência da Computação da UFSC. Com o DM populado, a instituição de ensino que utilizar o SACAD terá uma ferramenta para a gestão do problema de evasão nas IES, podendo ser ampliada com novas visões no futuro e integrada a outros DM's. Com alertas proativos a figura do coordenador de curso terá mais informação para identificar e tratar os casos de alunos e disciplinas problemáticas, embasando uma ação em conhecimento. Espera-se também um melhor rendimento do aluno já que o sistema o ajudará a se matricular em disciplinas adequadas ao seu perfil.

O sistema como está pode ser implantado na UFSC ou qualquer instituição de ensino, desde que esta gere os arquivos necessários para alimentar o sistema, sendo esses arquivos, juntamente com seu layout, relacionados no apêndice correspondente neste trabalho. Para o caso específico da UFSC, em conversa com o setor responsável pelo desenvolvimento de sistemas da universidade, foi definido que eles reescreverão o sistema na plataforma Java, pois a política interna do setor prefere trabalhar com somente um tipo de tecnologia para aproveitar o conhecimento interno de seus funcionários. Assim, o sistema servirá como referência direta.

6.2. Trabalhos futuros

Esta seção reúne possíveis melhorias para o sistema, levantadas durante o desenvolvimento, as quais vamos relacionar aqui para futura referência.

- Criar interface REST para permitir integração direta com outros sistemas. Isso permite que as funcionalidades do sistema criado sejam incorporadas nos sistemas da universidade sem que o usuário final perceba que está utilizando outro sistema. Para o caso específico da UFSC, essa integração pode dar no sistema CAGR, onde no

momento da matrícula semestral o aluno poderia já visualizar suas chances de aprovação ao escolher as disciplinas.

- Melhoria do índice IAD. Hoje a fórmula é simples e poderiam ser acrescentadas outras variáveis, como a carga horária da disciplina e quantidade de disciplinas que a tem como pré-requisito.
- O sistema foi desenvolvido para considerar cursos de 2 semestres anuais. Para cursos com uma divisão anual diferente é necessário fazer algumas adaptações nos algoritmos bem como no ETL.
- Tornar a staging area, hoje no formato de log, também dimensional.
- Ajustes visando desempenho (tuning) na base de dados, considerando o futuro uso em produção.

7. Bibliografia

- CISLAGHI, Renato; SILVA FILHO, Vasco; ALBUQUERQUE, Almir S.; SARTORI, Rejane. SACAD - Construção de um Data Mart para apoio aos coordenadores e acadêmicos de cursos de graduação. In: **VI Colóquio Internacional sobre Gestão Universitária na América do Sul**, 2006, Blumenau. VI Colóquio Internacional sobre Gestão Universitária na América do Sul - Mudanças e Perspectivas na Gestão Universitária, 2006.
- VELOSO, Tereza C. M. A. **A Evasão nos Cursos de Graduação da Universidade Federal de Mato Grosso, campus Universitário de Cuiabá - 1985/2 a 1995/2 um processo de exclusão**. 2001. 193p. Dissertação. (Mestrado), Programa de Pós-Graduação em Educação. Universidade Federal de Mato Grosso, UFMT, Cuiabá, 2001
- BRASIL. Governo Federal. Ministério da Educação. **Reuni - Reestruturação e Expansão das Universidades Federais: Diretrizes Gerais**. Documento Elaborado pelo Grupo Assessor nomeado pela Portaria nº 552 SESu/MEC, de 25 de junho de 2007, em complemento ao art. 1º §2º do Decreto Presidencial nº 6.096, de 24 de abril de 2007.. Disponível em:
<<http://portal.mec.gov.br/sesu/arquivos/pdf/diretrizesreuni.pdf>>. Acesso em: 15 nov. 2012.
- INMON, W. H.. **Building the Data Warehouse**. 3. ed. Usa: John Wiley & Sons Inc, 2002.
- KIMBALL, Ralph; REEVES, Laura; ROSS, Margy. **The Data Warehouse Lifecycle Toolkit :Expert Methods for Designing, Developing, and Deploying Data Warehouses**. Nova York: John Wiley & Sons Inc, 1998
- PENTAHO COMMUNITY (Usa). Organization (Comp.). **Latest Pentaho Data Integration (aka Kettle) Documentation. Documentação da Ferramenta de ETL Pentaho Data Integration**. Disponível em:
<<http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation>>. Acesso em: 20 fev. 2013.
- ALBUQUERQUE, A. R. P. L. ; ZANCARLI, Felipe Marcel. **Modelagem de um Sistema de Informação para Gerenciamento Estratégico de uma Instituição de Ensino**

Superior com Aplicação do Balanced Scorecard. Em: VII SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INTERNACIONAIS (SIMPOI), 2004, São Paulo (SP). Anais do VII Simpoi, v. VII, p. 1-12, 2004.

CISLAGHI, Renato. **Um modelo de sistema de gestão do conhecimento em um framework para a promoção da permanência discente no ensino de graduação.** 2008. 253f. Tese (doutorado) Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-graduação em Engenharia e Gestão do Conhecimento. 2008.

8. Apêndices

8.1. Manual Técnico

SUMÁRIO

1.	Solução Adotada	38
2.	Carga dos Dados (ETL)	38
2.1.	Tecnologias Envolvidas	38
2.2.	Arquitetura do ETL	39
2.2.1.	Configuração do ETL.....	40
2.2.2.	Layout de Entrada	42
2.2.3.	Estrutura de Estagiamento	44
2.2.4.	Carga dos Arquivos.....	46
2.2.5.	Validações.....	49
2.2.5.1.	Validações Léxicas	50
2.2.5.2.	Validações de Duplicidade	50
2.2.6.	Dicionário de Dados do Cubo.....	51
2.2.6.1.	Dimensões.....	51
2.2.6.2.	Fatos.....	52
2.2.6.3.	Mapeamentos	55
2.2.7.	ER do Cubo SACAD	56
2.2.8.	Carga do Cubo SACAD	56
2.2.8.1.	Carga das Dimensões.....	56
2.2.8.2.	Carga dos Mapeamentos.....	59
2.2.8.3.	Carga dos Fatos.....	61
2.2.9.	Tratamento de Exceções	63
2.2.9.1.	Variáveis de Controle de Erro	64
2.2.9.2.	Envio de E-mail	65
2.2.9.3.	Filtro de Referências Inválidas	66
2.2.10.	Área Administrativa	67
2.2.10.1.	Relatórios de Importação.....	68
3.	Aplicação	70
3.1.	Tecnologias envolvidas	70
3.2.	Arquitetura da aplicação.....	70

3.2.1.	Aplicação Web	71
3.2.1.1.	Camada de Apresentação	71
3.2.1.2.	Camada de Controle	71
3.2.1.3.	Camada de Modelo.....	71
3.2.2.	Rotina de Pré-Processamento	71
3.2.3.	Estrutura do Código.....	71
3.3.	Detalhamento dos Algoritmos	72
3.3.1.	Primeira Pergunta: Quais disciplinas têm maior repercussão no tempo médio para a conclusão de um curso?	72
3.3.2.	Segunda Pergunta: Quais alunos estão com perspectiva de serem jubilados? 73	
3.3.2.1.	Rotina de Pré-Processamento.....	75
3.3.2.2.	Algoritmo de Jubilamento	76
3.3.3.	Terceira pergunta: Qual é a expectativa de sucesso de um determinado aluno que se matricula em um determinado conjunto de disciplinas?	80
3.4.	Documentação das classes do sistema	81
3.4.1.	Classes	81
3.4.2.	Detalhamento das classes	84

Solução Adotada

A arquitetura envolvida na solução adotada pode ser vista no diagrama da figura 1.

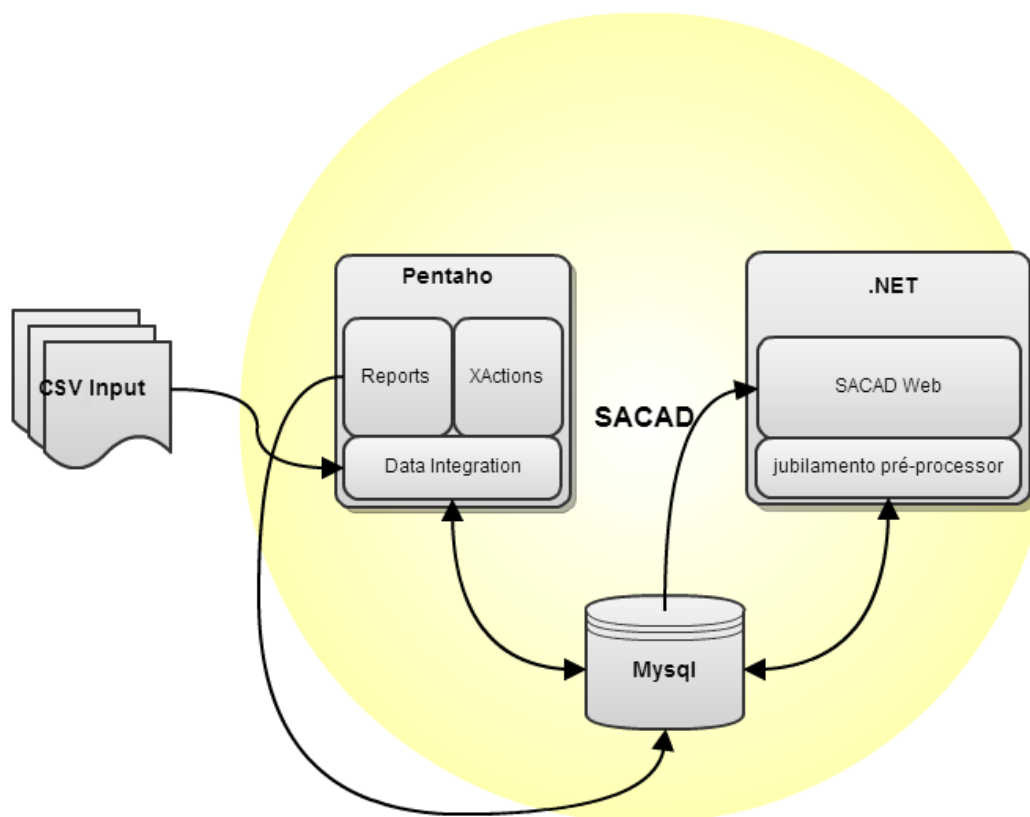


Figura 1 – Arquitetura Envolvida na Solução

Na figura podemos observar a entrada de dados (arquivos CSV), que é lido pela engine de ETL, Pentaho Data Integration (antigo Kettle). Esta engine realiza todo o tratamento de dados e carrega as dimensões no Data Mart (Mysql). Ainda na tecnologia Pentaho, temos Reports, que fornecem a monitoração das importações e as chamadas XActions, que invocam a engine de ETL, iniciando uma importação. Do lado .NET temos uma rotina de processamento (*batch*), que faz o pré-processamento dos jubileamentos e a aplicação SACAD web. Esta aplicação lê os dados pré-processados pelo batch e também o Data Mart populado para fornecer os relatórios e projeções idealizados pelo sistema.

Carga dos Dados (ETL)

Tecnologias Envolvidas

A ferramenta de *ETL* escolhida para o projeto foi o *Pentaho Data Integration (PDI)*, anteriormente chamado de *Kettle*, na versão 4.1 *Community*. O *PDI* é *open source* e *freeware*. A escolha se deu também por conhecimento prévio na utilização da ferramenta.

Em todo o projeto, foi utilizado o *MySQL 5.5.16* para o SGDB, incluindo na camada de *stage* do ETL. A modelagem do banco foi feita utilizando o *MySQL Workbench 5.2.35 Community Edition*.

A administração do *ETL*, bem como visualização de relatórios de importação, é feita através do *Pentaho BI Server*, na sua versão *Community 3.10.0*. Dessa forma é possível ter uma solução integrada e permite que mais relatórios ou visões do cubo sejam adicionados em trabalhos futuros.

Arquitetura do ETL

O *ETL* foi desenvolvido em módulos, que são executados sequencialmente. Primeiramente os arquivos são carregados para a área de estagiamento, depois eles são validados, registro a registro, quanto ao conteúdo da informação (léxico). Os registros não condizentes são marcados e o erro informado. Após a validação léxica, é feita uma validação quanto à duplicidade do registro no arquivo e também em importações anteriores. Os registros duplicados ou já importados também são marcados como inválidos e não serão carregados no cubo. Esta sequência pode ser exemplificada na figura 2:

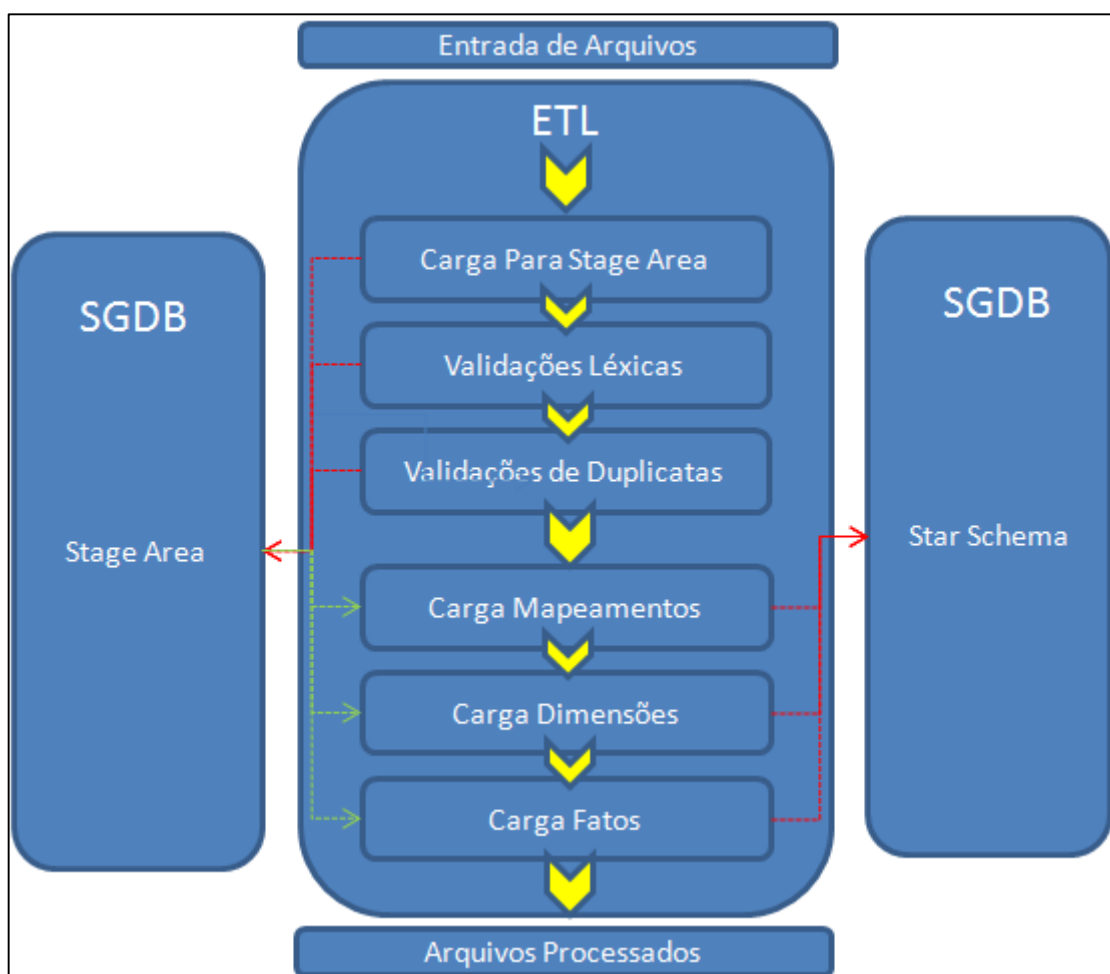


Figura 2 - Visão geral do ETL

Terminado as validações, os registros válidos são importados para as dimensões, mapeamentos e, finalmente, para os respectivos fatos. Durante essa fase ainda é feita uma validação referencial, sob a chave natural das dimensões envolvidas: caso elas não estejam previamente cadastradas, o registro que faz a referência não é importado e um erro é informado no registro de origem, na área de estagiamento.

Como último passo, os arquivos importados são copiados da pasta de entrada para um diretório de importados.

As fases da integração são coordenadas pelo job de nome IniciaIntegracao.kjb. Ele é responsável por iniciar algumas variáveis de controle de erros, validar o XML de configuração e chamar o coreIntegracao.kjb, que possui o sequenciamento de ações da integração. A estrutura do IniciaIntegracao pode ser visualizada na figura 3 e o sequenciamento de ações do coreIntegracao.kjb, na figura 4.

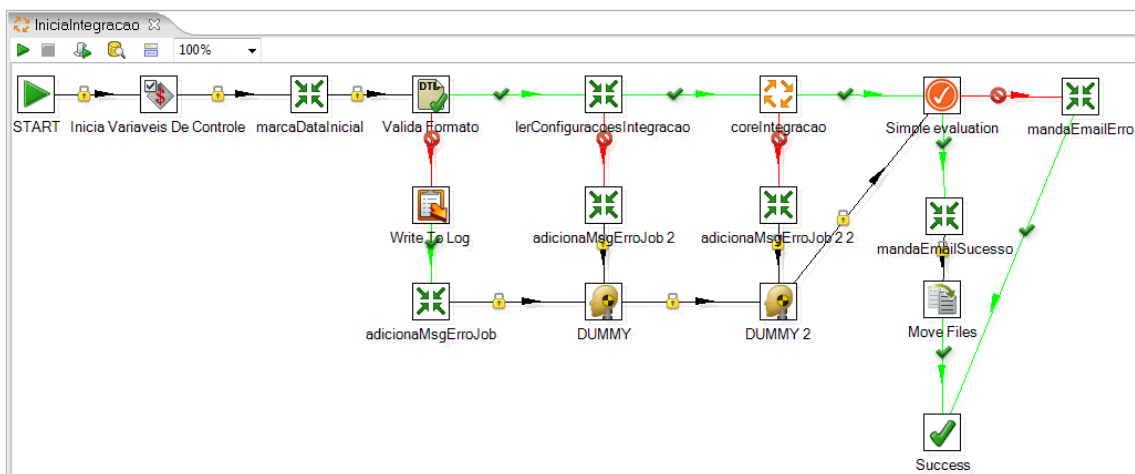


Figura 3 - Inicia Integração

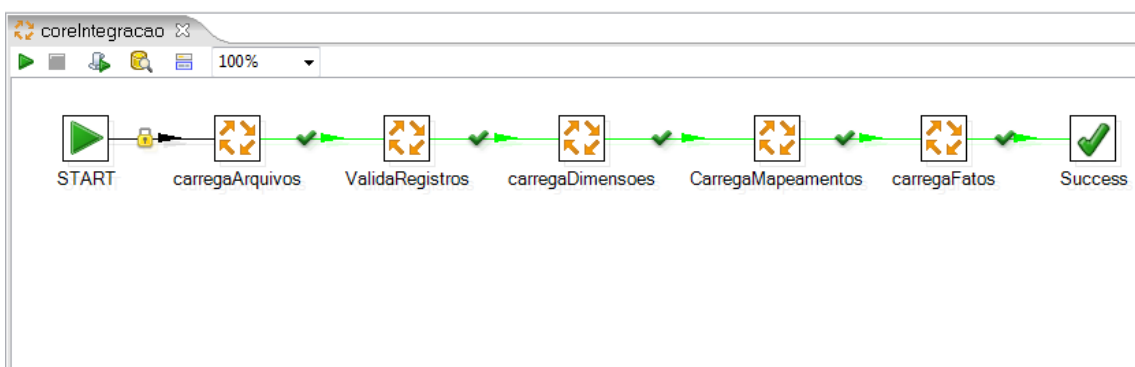


Figura 4 – core Integração

Configuração do ETL

O ETL foi desenvolvido para ser parametrizável através do arquivo config.xml presente no diretório de integração, ou seja, a pasta onde os arquivos do PDI estão. Este arquivo possui as seguintes configurações:

- **DiretorioEntrada:** Diretório onde os arquivos de entrada serão lidos;
- **DiretorioImportados:** Diretório onde os arquivos já processados serão movidos;
- **DiretorioLog:** Diretório de logs;
- **StageAreaJNDIName:** JNDI configurado no Pentaho BI Server para SGDB de de estagiamento;
- **SACADCubeJNDIName:** JNDI configurado no Pentaho BI Server para SGDB do datamart;
- **EndDestino:**Endereço de *e-mail* do responsável pela monitoração;
- **EndRemetente:** Endereço remetente do e-mail;
- **NomeRemetente:** Nome do remetente;
- **ServidorSMTP:** nome/*IP* do servidor de *SMTP*;
- **PortaServidor:** porta do servidor de *SMTP*;
- **Login:** logino do usuário do e-mail remetente;
- **Senha:** senha do usuário do e-mail remetente;
- **EndResposta:** endereço para resposta, caso o responsável pela monitoração deva entrar em contato com alguém;
- **NomeContatoResposta:** nome do contato de resposta, caso o responsável pela monitoração deva entrar em contato com alguém;
- **FoneResposta:** telefone para contato, caso o responsável pela monitoração deva entrar em contato com alguém.

Este arquivo de configuração é descrito por um *DTD* interno e validado pela integração quanto ao formato esperado do mesmo. O processo de *ETL* não inicia caso o arquivo seja inválido ou não exista. O *ETL* também valida alguns aspectos léxicos quanto ao preenchimento dessas informações. Após validar, as configurações são atribuídas em variáveis para serem consumidas pelo *ETL*. A transformação responsável por este processo é *lerConfiguracoesIntegracao.ktr*, exibido na figura 5.

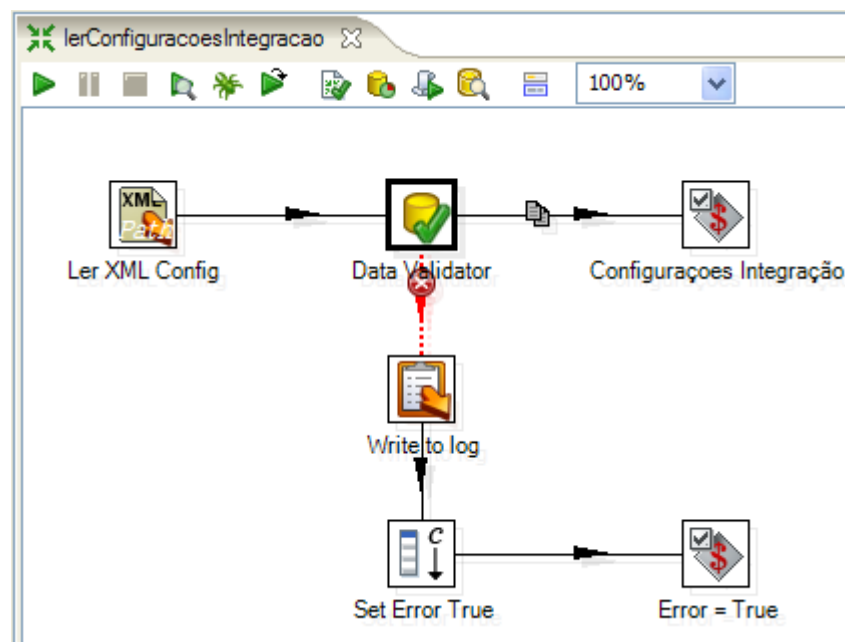


Figura 5 - Carga das Configurações

Layout de Entrada

Para entrada das informações, o ETL desenvolvido espera receber os arquivos no padrão de nome e conteúdo da Tabela 1.

Curriculo_[DATA].txt		
Nome Colunas	Tipo	Formato
ano_semestre	Integer	ANO+SEMESTRE
id_curriculo	Integer	ID_CURSO+ANO+SEMESTRE
inicio_curriculo	Integer	ANO+SEMESTRE
id_curso	Integer	
id_disciplina	String	
cred_disciplina	Integer	
carga_horaria	Integer	
disciplina_optativa	Boolean	1=optativa; 0=obrigatória
fase_padrao_disciplina	Integer	0 para não determinado
DisciplinasEquivalentes_[DATA].txt		
Nome Colunas	Tipo	Formato
id_curriculo	Integer	ID_CURSO+ANO+SEMESTRE
id_disciplina	String	
id_disciplina_eqv	String	
cd_tipo_tcj	List	EQUAL OR AND
RequisitosConjuntos_[DATA].txt		
Nome Colunas	Tipo	Formato
id_curriculo	Integer	ID_CURSO+ANO+SEMESTRE
id_disciplina	String	
id_disciplina_conjunto	String	
DadosAlunoCurso_[DATA].txt		
Nome Colunas	Tipo	Formato
id_aluno	Integer	Matricula
nome_aluno	String	
data_ingresso_aluno	Integer	ANO+SEMESTRE
situacao_aluno	Boolean	regular=1;trancado=0
data_situacao	Integer	ANO+SEMESTRE
id_curso	Integer	
nome_curso	String	
id_coordenadoria_do_curso	String	
nome_coordenadoria_do_curso	String	
nm_centro_do_curso	String	
id_curriculo	Integer	ID_CURSO+ANO+SEMESTRE
IAA	Integer	Índice de aproveitamento do Aluno
AlunoDesempenhoSemestre_[DATA].txt		

Nome Colunas	Tipo	Formato
id_aluno	Integer	Matricula
id_disciplina_dis	String	
nome_disciplina	String	
nota_aluno_disciplina	Double	
frequencia_aluno_disciplina	Boolean	1=FS;0=FI
ano_semestre	Integer	ANO+SEMESTRE
aluno_aprovado_disciplina	Boolean	1=SIM;0=Não
id_curriculo	Integer	ID_CURSO+ANO+SEMESTRE
IAA	Integer	Índice de aproveitamento do Aluno
Requisitos_[DATA].txt		
Name	Type	Format
id_curriculo	Integer	ID_CURSO+ANO+SEMESTRE
id_disciplina	String	
id_disciplina_req	String	
cd_tipo_tcj	List	EQUAL OR AND
CurriculoParametros_[DATA].txt		
Name	Type	Format
ano_semestre	Integer	ANO+SEMESTRE
id_curriculo	String	
inicio_curriculo	Integer	ANO+SEMESTRE
id_curso	String	
prazo_max_conclusao_curso	Integer	Semestres
cred_minimos_curso_sem	Integer	Créditos Semanais
cred_max_curso_sem	Integer	Créditos Semanais
carga_horaria_min_curso	Integer	Carga Horária Mínima
relacao_carga_horaria_creditos	Integer	Número de horas equivalente a 1 crédito
carga_horaria_media_curso	Integer	
carga_horaria_optativa	Integer	Carga Horária Optativas
DisciplinasRelacionadas_[DATA].txt		
Name	Type	Format
id_disciplina_dis	Integer	id natural da disciplina
nome_disciplina	String	Nome
cred_disciplina	Integer	quantidade de créditos

Tabela 1 - Layout de arquivos de entrada

É importante notar que os identificadores das entidades (alunos, disciplinas, currículos, etc.) estão definidos como tipo String. Tal medida é para abranger outras faculdades que possuam identificadores alpha-numéricos ou outros tipos de identificadores. Para o sistema, só é necessário que eles sejam iguais em todos os arquivos, já que a partir da leitura dos arquivos eles passam a ter chaves artificiais.

Atualmente, o layout definido não é seguido pela exportação dos arquivos da UFSC, gerada pelo CAGR. Para não influenciar no tempo de desenvolvimento, foi desenvolvida uma transformação de conversão dos atuais arquivos para o modelo sugerido. Esta transformação está no diretório principal, ETL, com o nome de **UFSC_Conversor_Arquivos.ktr**. Esta transformação deve ser executada na mesma pasta dos arquivos gerados pelo CAGR, observando as regras de entrada do nome do arquivo. Na saída teremos arquivos convertidos pelo layout esperado, já com o nome correto. Este processo de “tradução” tem de ser executado até que o CAGR passe a gerar os arquivos de acordo com o layout proposto.

Estrutura de Estagiamento

Para realizar o estagiamento na carga do cubo, foi criada uma base de dados chamada de *stageareaintegration* e adotado a seguinte estrutura representada no diagrama da **Figura** .

conteúdo exato do arquivo, com metadados gerados pela integração. Os seguintes metadados estão presentes em todas as tabelas matriz:

- idHistoricoIntegracao: id da tabela historicointegracao, que permite mapear qual integração e arquivo que originou o registro;
- statuspendente: o status atual do registro (ver tabela abaixo “status de controle”);
- Statusdescription: descrição detalhada do status atual do registro;
- linhaarquivoorigem: linha que gerou o registro em questão.

Os valores de statuspendente representam que estágio de importação o registro está. Todos os status são atualizados em uma fase específica do ETL e seu significado é dado pela Tabela 2.

Statuspendente	Statusdescription	Significado	Ocorrências
1	Pendente de validação	O dado foi recém importado do arquivo, tal qual está no arquivo	Steps de leitura do arquivo (cvsLerArquivo___)
2	Valido léxicamente e pendente de validação de duplicidade	ado validado quando ao conteúdo léxico, formatos aceitos, etc	Steps de validação (valida___)
3	N/A - depende do erro, cada erro contém sua descrição	Dado não é válido, algum campo contém um valor não aceito.	Steps de validação (valida___)
4	Registros duplicado em outras importações	O registro já foi importado previamente (duplicidade em stageArea)	Steps de validação de duplicidade ((valDup___)
5	Validado e pendente de cadastro	Registro válido a ser importado para o cubo	Steps de validação de duplicidade (valDup___)
6	Registros importados com sucesso	Registro importado para o cubo	Steps de validação de duplicidade ((valDup___)
7	Referência inválida. Uma das chaves não foi previamente cadastrada	Um registro contém uma referência para uma entidade (dimensão) não cadastrada previamente.	Steps de inserção genérica (subTranInserRef-___)

Tabela 2 - Mapeamento de Status da Importação

Há também uma tabela para representar a relação arquivo/tabela e agregar informação nos relatórios da importação. Esta tabela possui o nome de importacaometadata.

Carga dos Arquivos

A carga dos arquivos foi desenvolvida prevendo diferentes arquivos de entrada ou banco de dados. Na elaboração do TCC limitou-se para tratar arquivos CSV, porém a estrutura permite que novos formatos de arquivos sejam tratados, bastando desenvolver o módulo de leitura. O job carregaArquivos.kjb executa a leitura e carga desses arquivos, atualmente CSV, como observado na figura 7.

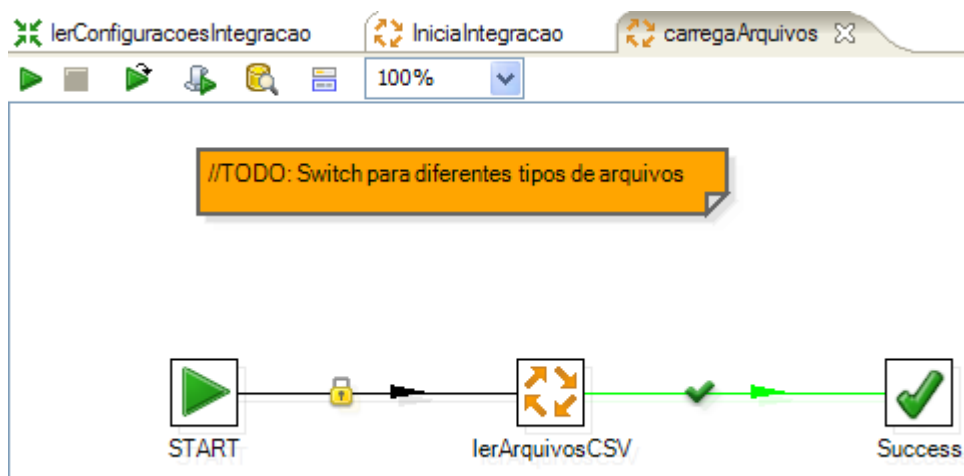


Figura 7 - Job de carga de arquivos

Na Tabela 3 podemos ver a relação entre os arquivos e suas respectivas tabelas.

Arquivo de Origem	Tabela Destino
Curriculo_[DATA].txt	curriculomatriz
DisciplinasEquivalentes_[DATA].txt	disciplinasequivalentesmatriz
RequisitosConjuntos_[DATA].txt	requisitosconjuntosmatriz
DadosAlunoCurso_[DATA].txt	dadosalunocursomatriz
AlunoDesempenhoSemestre_[DATA].txt	alunodesempenhosemestrematriz
Requisitos_[DATA].txt	requisitosmatriz
CurriculoParametros_[DATA].txt	curriculoparametrosmatriz
DisciplinasRelacionadas_[DATA].txt	disciplinasrelacionadasmatrix

Tabela 3 - Mapeamento de arquivos origem para tabelas destino

A leitura dos arquivos CSV se dá pelo job lerArquivosCSV.kjb, que marca a data inicial da integração e inicia as transformações responsáveis por cada um dos arquivos. No caso de ocorrer qualquer tipo de erro, um erro é adicionado ao log, no diretório configura no config.xml e o job abortado, caindo para o fluxo alternativo do job principal, IniciaIntegracao.kjb, onde é enviado o e-mail de erro. Este job, lerArquivosCSV.kjb, pode ser observado na figura 8.

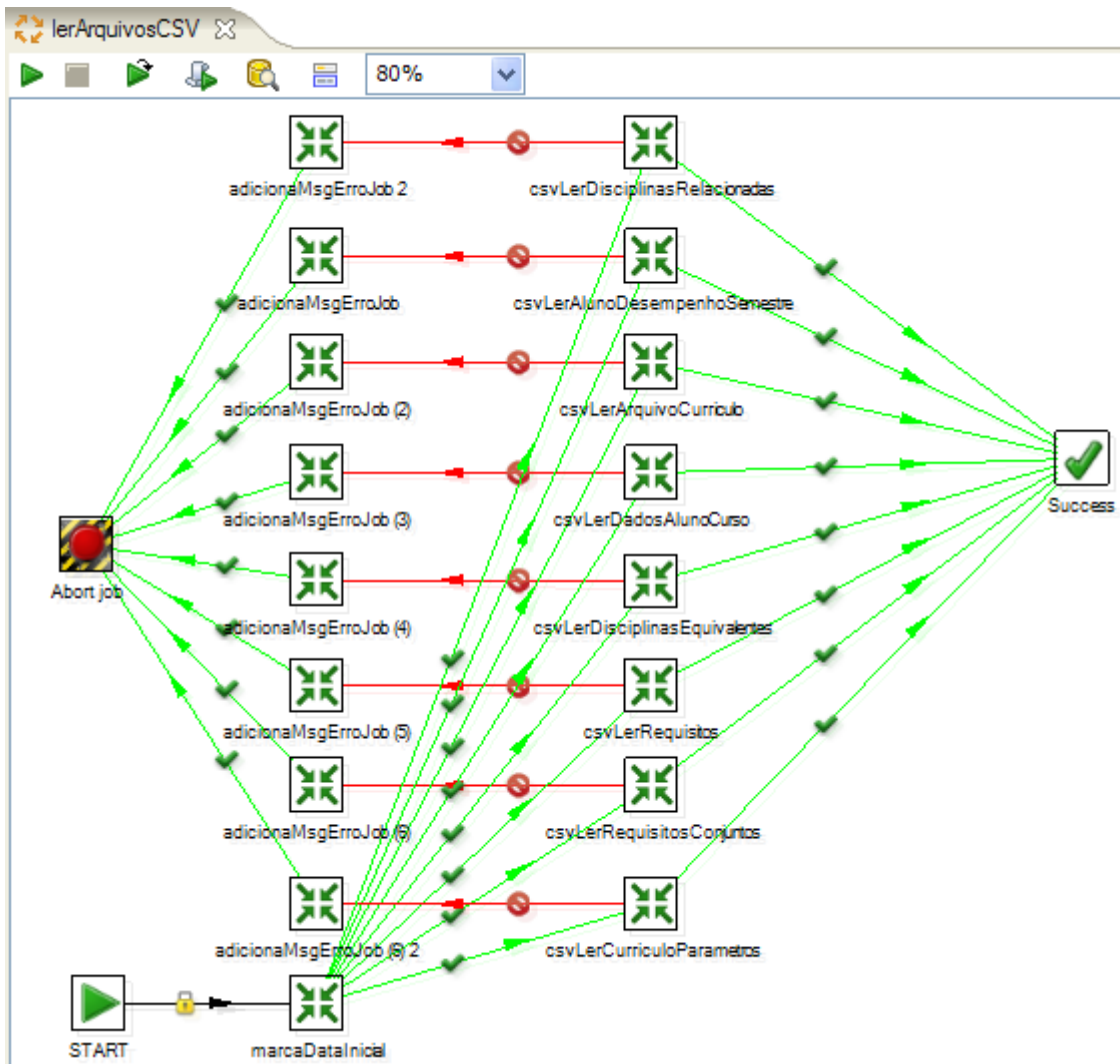


Figura 8 - Leitura de arquivos CSV

Cada uma das transformações de leitura dos arquivos segue o mesmo padrão. Primeiro algumas informações sobre o arquivo a ser lido são coletadas, como nome, data de modificação, etc. É lido também a data inicial da importação, fornecida pelo job carregaArquivos.kjb. Essas informações serão gravadas na tabela de controle da integração, HistoricoIntegracao. Após registrar a importação do arquivo, o conteúdo do arquivo é lido e gravado na sua respectiva tabela, com status de importação igual a 1 (ver cap. “Controle de Erros”). Outros metadados são formados quando o arquivo é processado, como a quantidade de linhas. Estas informações são gravadas no HistoricoIntegracao ao finalizar o processamento do arquivo. Este processo pode ser observado na figura 9.

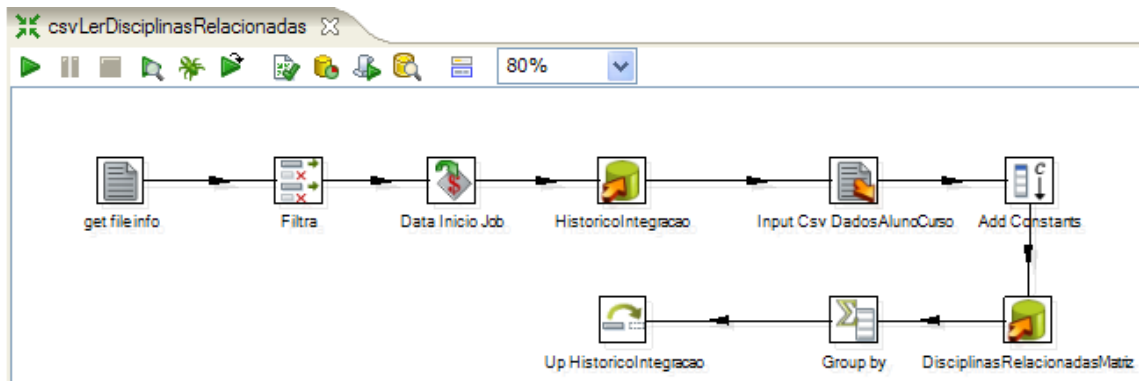


Figura 9 - Transformação de carga de arquivo CSV

Validações

Após a carga, os registros recém importados dos arquivos são validados. No caso de erro, será atualizado seu status de importação e uma mensagem de erro inserida explicando o erro. Ocorrem dois tipos de validações nesse estágio, validação léxica e de duplicidade. As validações são gerenciadas pelo job ValidaRegistros.kjb, onde são disparadas sequencialmente. Estas validações não abortam os Jobs, são capturadas e registradas na linha correspondente da Matriz. No entanto, exceções da execução são capturadas e abortam o Job (queda de conexão, etc.). Este tratamento pode ser observado na figura 10.

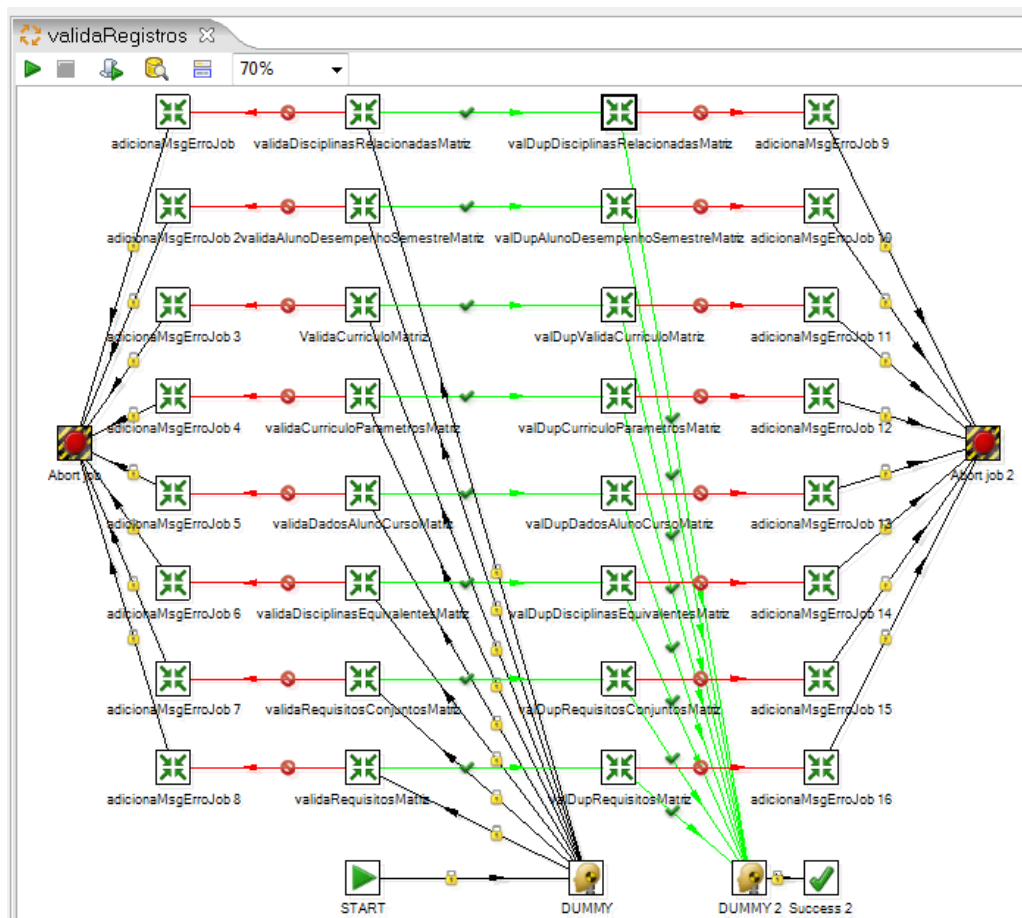


Figura 10 - Validação de Registros Importados

Validações Léxicas

As validações léxicas foram desenvolvidas em um padrão para todos os arquivos. Sua nomenclatura é iniciada por valida acrescido do nome do arquivo em questão (e.g. validaDisciplinasRelacionadasMatriz). São lidos todos os registros presentes na tabela matriz relativa ao arquivo que estão com status igual a 1. Caso o formato do dado não seja o esperado pela validação, um erro é gerado e a linha deste registro atualizada com o status igual 3. Para registros validados, o status é atualizado para 2. A validação léxica é genericamente apresentada pela figura 11.

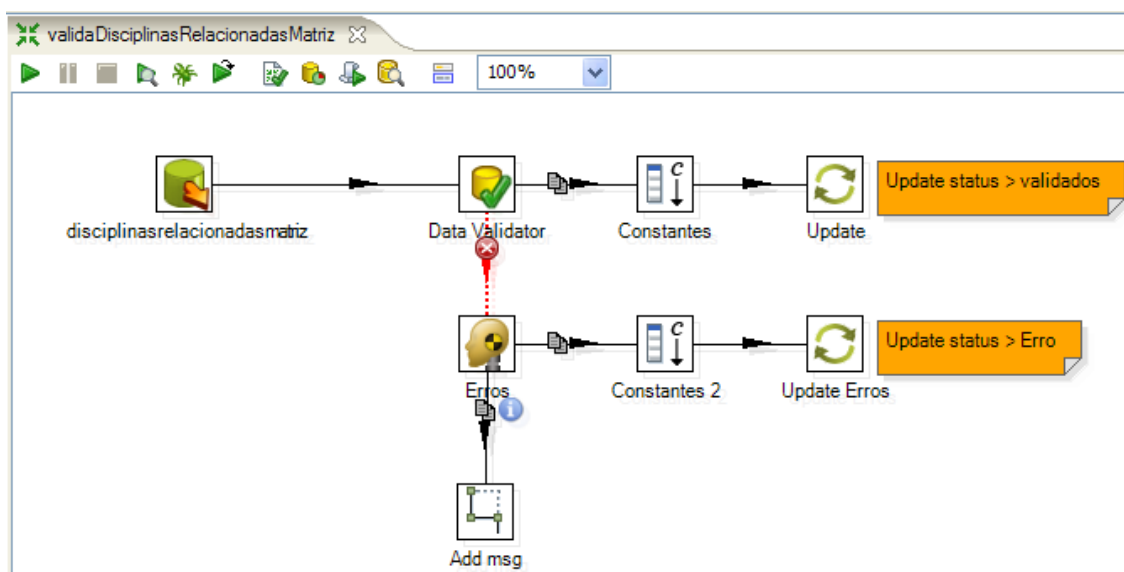


Figura 11 - Validação Léxica

Validações de Duplicidade

As validações de duplicidade permitem que o mesmo arquivo seja reprocessado, caso necessário e impede também que arquivos com linhas duplicadas causem problemas ao cubo do SACAD. Foi estabelecido um padrão de nomenclatura formado pelo prefixo valDup acrescidos da tabela matriz em questão (e.g. valDupDisciplinasRelacionadasMatriz). As validações são basicamente iguais no seu processo, diferenciando apenas pelas tabelas envolvidas e chaves utilizadas para verificação.

Inicialmente são lidos todos registros da matriz que estejam com status 2. Após a leitura, é utilizado um *step* de *Unique Hash Set*, para eliminar duplicatas da mesma importação. Este *step* permite que o registro-cópia seja tratado como erro, onde é atualizado para status 4, indicando um erro por duplicidade. Os registros que são únicos na importação, ou seja, passaram pelo *Unique Hash Set*, são validados contra importações prévias (*step* de *database lookup*), buscando um registro de uma importação anterior que tenha sido corretamente importado. Dessa forma também é garantido que registros não serão reimportados, evitando gerar informações errôneas nas tabelas fato ou uma política de controle de dimensão (*slowly changing dimensions*). Os registros que possuem uma cópia de outra importação, verificado pelo passo de *lookup*, também são atualizados com status 4. Os registros válidos, recebem status 5, indicando que estão elegíveis a serem importados para o cubo. Este processo é representado pela figura 12.

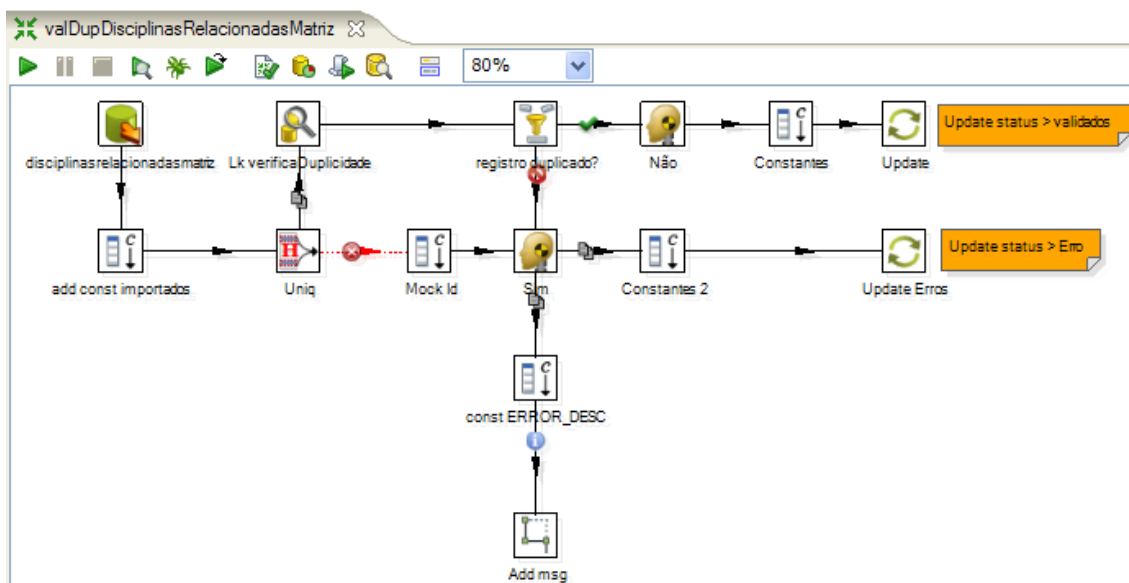


Figura 12 - Validações de Duplicatas

Dicionário de Dados do Cubo

Para facilitar as consultas analíticas, foi adotado o modelo estrela, em uma base de dados chamada de cubo_sacad. Adicionalmente a este modelo, introduziram-se algumas tabelas de mapeamento, que não se encaixavam na estrutura tradicional de modelagem de um cubo. Estes mapeamentos permitem consultar a correlação entre disciplinas e serão explicados mais adiante no documento.

Dimensões

As dimensões são precedidas pelo prefixo “dim”, o que facilita a identificação nas consultas. Foi identificada a necessidade de cinco dimensões no nosso *datamart*, verificadas nas tabelas abaixo (4, 5, 6, 7 e 8).

dim_aluno	
representa um aluno em um determinado curso e currículo	
CAMPO	COMENTARIOS
id	chave artificial
idaluno	matricula do aluno
nome	nome do aluno
ingresso	data que ingressou no curso

Tabela 4 – Dimensão Aluno

dim_curriculo	
representa um determinado currículo de um curso	
CAMPO	COMENTARIOS
id	chave artificial
idcurriculo	id natural do curriculo

nome	nome amigável do currículo. Caso não exista, é repetido idcurrículo
inicio_vigencia	data inicial de vigência do currículo

Tabela 5 – Dimensão Currículo

dim_curso	
representa um curso	
CAMPO	COMENTARIOS
id	chave artificial
id_curso	id do curso no sistema importado
nome	nome do curso
id_coordenadoria	id da coordenadoria do curso
nome_coordenadoria	coordenadoria do curso (ou equivalente)
centro	centro do curso (ou equivalente)

Tabela 6 – Dimensão Curso

dim_disciplina	
representa uma disciplina	
CAMPO	COMENTARIOS
id	chave artificial
iddisciplina	id da disciplina no sistema importado
nome	Nome da disciplina

Tabela 7 – Dimensão Disciplina

dim_tempo	
representa o tempo em que o registro ocorreu. Em nossos cubos, a menor unidade é dada por um semestre	
CAMPO	COMENTARIOS
id	chave artificial
ano	inteiro representando o ano
semestre	semestre do ano
semestre_ano	Inteiro representando ano+semestre, para facilitar consultas

Tabela 8 – Dimensão Tempo

No modelo adotado, as mudanças de currículo em um curso e possíveis mudanças do aluno (trancamento, etc) são registradas ou deduzidas nos fatos. Por esta abordagem, não foi necessário adotar nenhuma técnica de *slowly changing dimensions*.

Fatos

Para a necessidade do sistema, foram criados cinco fatos. Foi adotado o sufixo “fato” para facilitar a identificação dos mesmos. Abaixo temos uma descrição do que cada fato registra no sistema (tabelas 9, 10, 11, 12 e 13)

Fato_curriculo_curso	
Registra as configurações de um determinado currículo em um curso.	
CAMPO	COMENTARIOS
dim_curriculo_id	Curriculo
dim_curso_id	Curso
dim_data_id	dimensão degenerada de semestre, inteiro representando o semestre que ocorreu o fato
creditos_min	quantidade mínima de créditos durante a semana a serem feitos pelo aluno
creditos_max	quantidade máxima de créditos durante a semana a serem feitos pelo aluno
carga_horaria_min	número de horas/aula mínimas para se formar no curso
relacao_cred_carga	relação entre valor de crédito e carga horária;
prazo_max_conclusao	número de semestres máximo para a conclusão do curso
carga_horaria_optativa	carga horária de disciplinas optativas a ser cumprida

Tabela 9 – Fato Curriculo Curso

fato_curriculo_disciplina	
Registra as configurações de cada disciplina em um determinado currículo de um curso	
CAMPO	COMENTARIOS
dim_curriculo_id	Curriculo
dim_curso_id	Curso
dim_disciplina_id	Disciplina
dim_data_id	dimensão degenerada de semestre, inteiro representando o semestre que ocorreu o fato
creditos	créditos de uma disciplina neste currículo
optativa	se é optativa neste currículo
semestre_padrao	semestre padrão para cursar a disciplina neste currículo
carga_horaria	carga horária da disciplina neste currículo

Tabela 10 – Fato Currículo Disciplina

fato_desempenho_aluno_disciplina	
Representa o desempenho de um aluno em uma determinada disciplina no semestre	
CAMPO	COMENTARIOS
dim_curriculo_id	Curriculo
dim_aluno_id	Aluno
dim_tempo_id	dimensão degenerada de semestre, inteiro representando o semestre que ocorreu o fato
dim_disciplina_id	Disciplina

frequencia	54urrículo54 do aluno em uma determinada disciplina neste semestre
nota_final	nota final do aluno em uma determinada disciplina neste semestre
aluno_aprovado_disciplina	Se aluno foi ou não aprovado nesta disciplina

Tabela 11 – Fato Desempenho Aluno Disciplina

fato_desempenho_aluno_semestre	
Representa o desempenho do aluno no semestre inteiro	
CAMPO	COMENTARIOS
dim_curriculo_id	Curriculo
dim_aluno_id	Aluno
dim_tempo_id	Dimensão degenerada de semestre, inteiro representando o semestre que ocorreu o fato
media_geral	Media geral das notas do aluno nas disciplinas neste semestre
media_relativa	Media das notas do alunos ponderando créditos das disciplinas calculadas
IA	Índice de aproveitamento (gerado pela instituição) do aluno neste semestre
frequencia_geral	Frequência geral do aluno nas disciplinas neste semestre
frequencia_relativa	Frequência do aluno ponderando o crédito das disciplinas neste semestre.

Tabela 12 – Fato Desempenho Aluno Semestre

fato_desempenho_disciplina	
Representa o desempenho da disciplina no semestre	
CAMPO	COMENTARIOS
dim_disciplina_id	Disciplina
dim_tempo_id	dimensão degenerada de semestre, inteiro representando o semestre que ocorreu o fato
dim_curso_id	Curso
media	media das notas dos alunos da disciplina neste semestre
media_pond	media ponderada por quantidade de alunos da disciplina neste semestre
qtd_alunos	quantidade de alunos da disciplina neste semestre
qtd_reprovados	quantidade de alunos reprovados da disciplina neste semestre
qtd_desistentes	quantidade de alunos que não atingiram a 54urrículo54 mínima
IAD	calculo de IAD, métrica sugerida no artigo SACAD

Tabela 13 – Fato Desempenho Disciplina

Mapeamentos

Devido a necessidade de registrar alguns mapeamentos necessários para as consultas, foram criados três mapeamentos de disciplinas. Estes mapeamentos são exclusivamente para atender algumas queries do sistema, vistos nas tabelas 14,15 e 16.

Map_disciplina_equivalencia	
mapeamento de correspondência (equivalência) entre disciplinas nos diferentes currículos do curso	
CAMPO	COMENTARIOS
id	chave artificial
dim_curriculo_id	Curriculo
dim_disciplina_id	id da disciplina que substituiu outra neste currículo
dim_disciplina_id_eqv	id da disciplina que foi substituída por outra neste currículo
tipo_eqv	AND, EQUAL, OR. Informa se a correspondência deve considerar outra disciplina (AND), qualquer uma das disciplinas (OR) ou se a correspondência é de um para um (EQUAL)

Tabela 14 - Mapeamento de equivalências entre disciplinas

map_disciplina_requisito	
mapeamento de requisito entre disciplinas em um mesmo currículo	
CAMPO	COMENTARIOS
id	chave artificial
dim_curriculo_id	Curriculo
dim_disciplina_id	id da disciplina que é trancada por outra disciplina
dim_disciplina_id_req	id da disciplina que tranca outra disciplina
tipo_requisito	Atualmente somente pré-requisitos são preenchidos (EQUAL)

Tabela 15 - Mapeamento de pré-requisitos

map_disciplina_requisitos_conjuntos	
mapeamento de disciplinas que devem ser cursadas conjuntamente no mesmo currículo	
CAMPO	COMENTARIOS
id	chave artificial
dim_curriculo_id	Curriculo
dim_disciplina_id	id da disciplina que deve ser cursada em conjunto
dim_disciplina_conj_id	id da disciplina que deve ser cursada em conjunto

Tabela 16 - Mapeamento de Requisitos Conjuntos

ER do Cubo SACAD

O esquema estrela do cubo SACAD pode ser representado pelo diagrama ER da **Figura 2**.

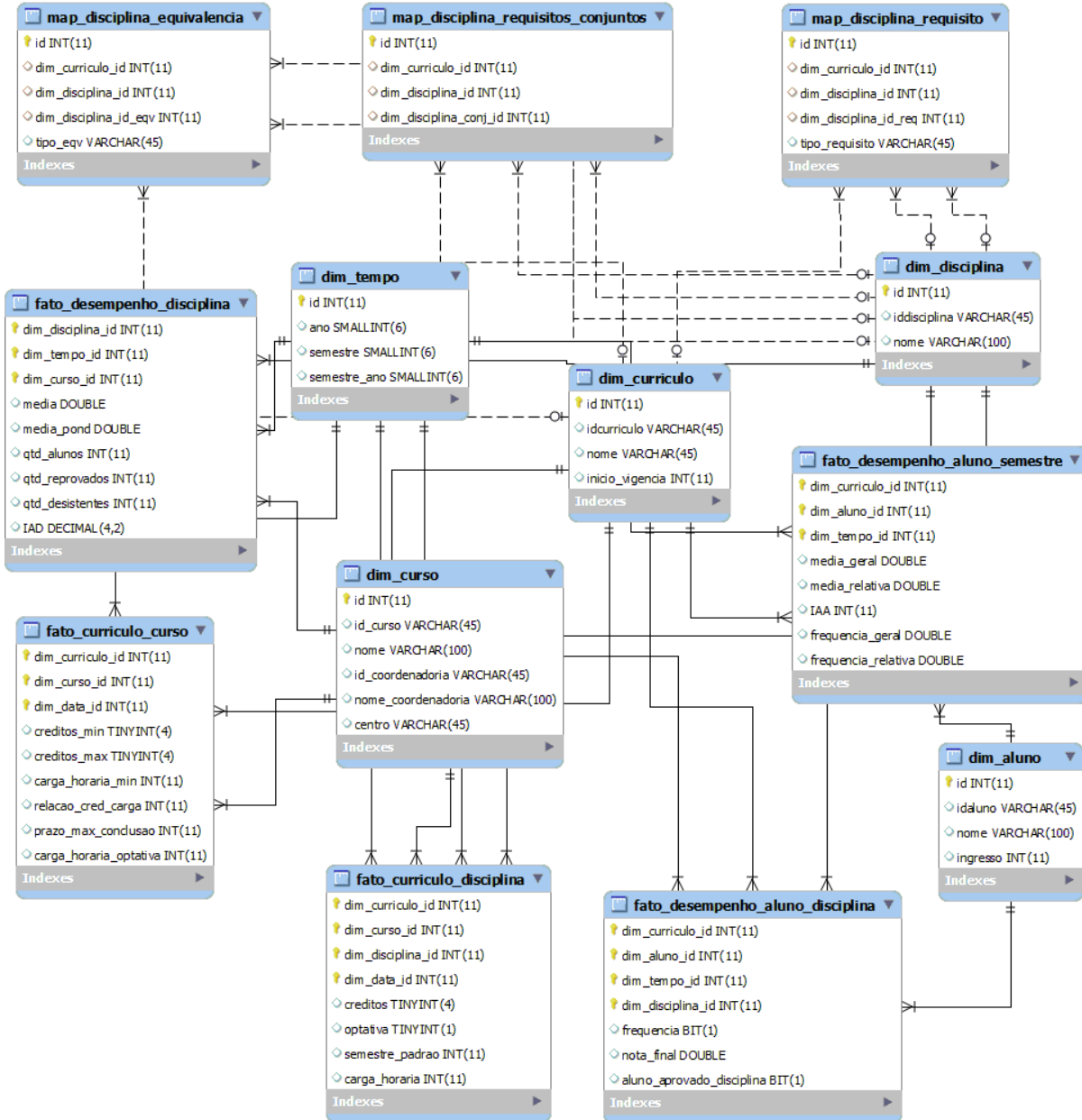


Figura 2 - Diagrama ER do Cubo SACAD

Carga do Cubo SACAD

Carga das Dimensões

Uma vez validados, os registros são primeiramente importados para as dimensões, pois necessitam estar previamente cadastrados para que sejam referenciados nas tabelas de mapeamentos e nas tabelas fatos. As dimensões adotadas na solução são bastante simples e, devido ao tratamento de duplicidade e validação léxica, sem necessidade de grandes transformações. A carga das dimensões é coordenada pelo *job* cargaDimensoes.kjb. Como

padronização, foi adotada a nomenclatura formada pelo prefixo “carrega” acrescido da dimensão a ser preenchida, como pode ser visto na figura 14.

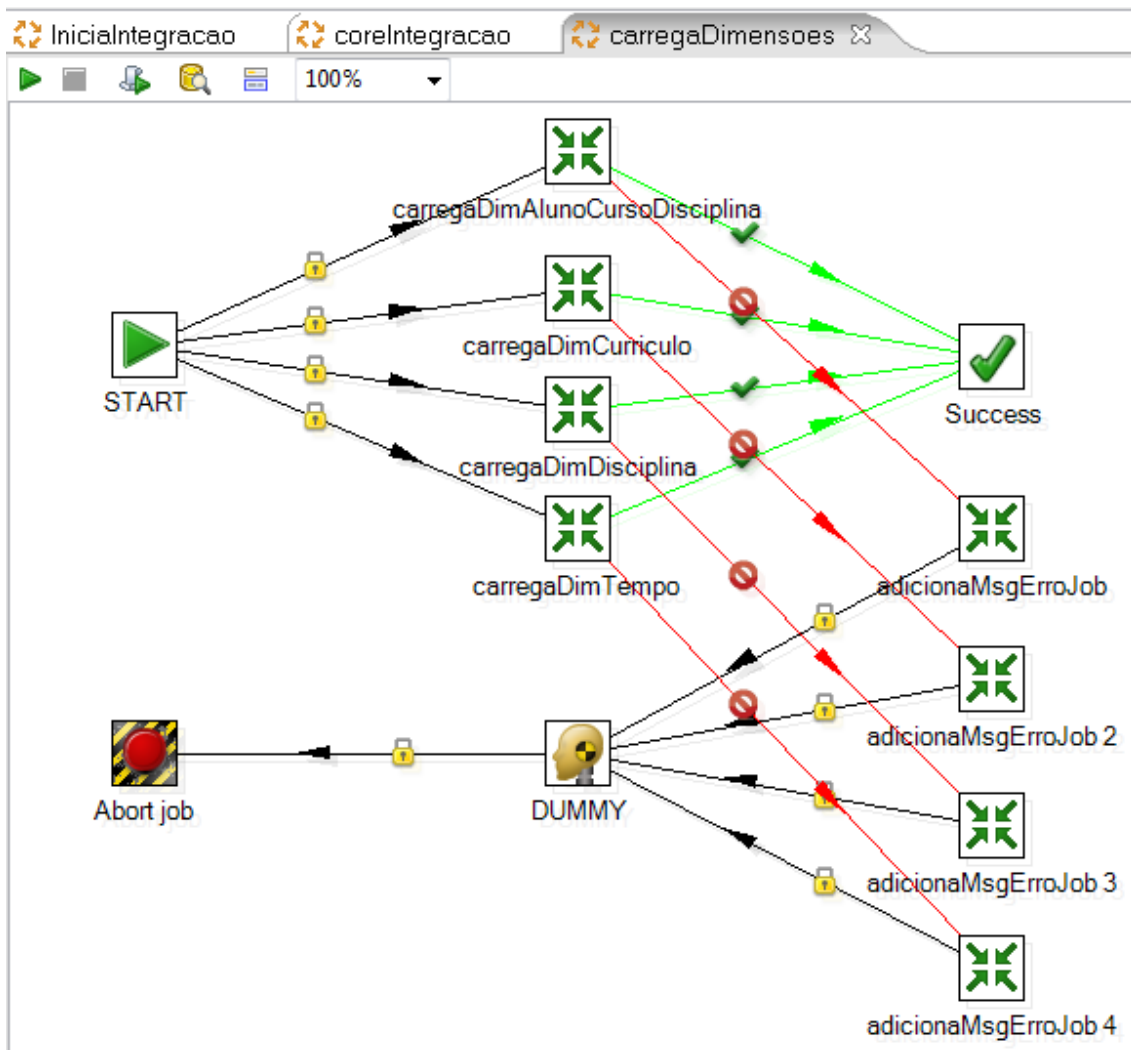


Figura 3 - Carga das Dimensões

A dimensão tempo (dim_tempo) é gerada via cálculo e abrange os semestres desde o ano 1900 até 2150. No entanto a carga verifica se há registros nessa dimensão e não efetua o preenchimento desta dimensão caso ela já tenha sido preenchida anteriormente. O job desta dimensão pode ser visto na figura 15.

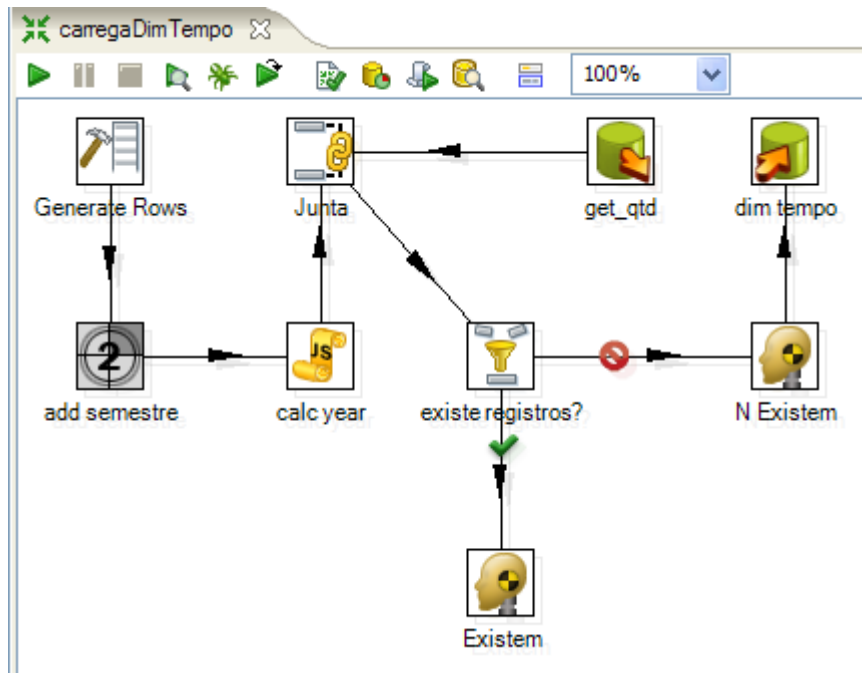


Figura 4 - Carga da Dimensão Tempo

As demais cargas são tipicamente exemplificadas pela figura abaixo, que retrata a carga da dimensão disciplina. A leitura é feita na matriz e inserida na dimensão. Por precaução, é utilizado o *step* de *insert/update*, para verificar se o registro já existe na tabela. Ao final do processamento, é atualizado o status do registro advindo da matriz, informando que foi importado com sucesso, como visto na figura 16.

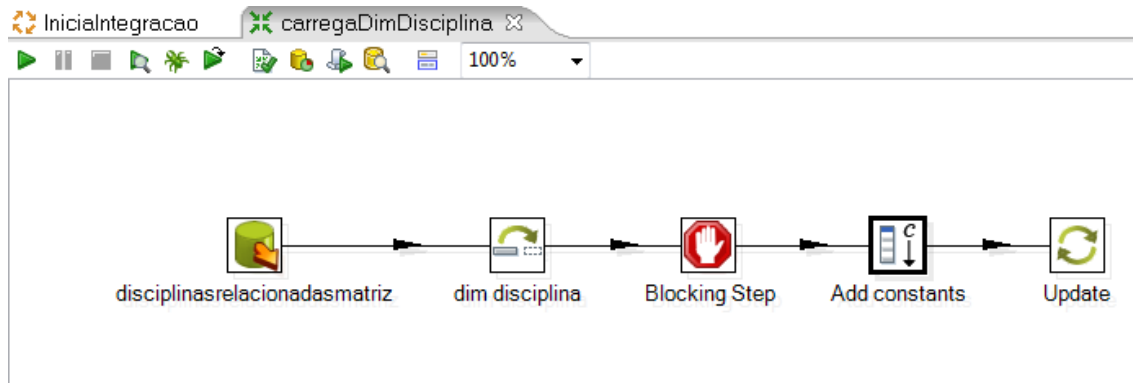


Figura 5 - Carga Típica de Dimensões

As dimensões têm por origem as tabelas matriz, obedecendo ao mapeamento da Tabela 17.

Mapeamento Tabelas Matriz - Dimensões		
Dimensão	Tabela Matriz Origem	Observação
dim_curso	dadosalunocursomatriz	Inserido somente registros distintos
dim_aluno	dadosalunocursomatriz	
dim_curriculo	curriculomatriz	
dim_disciplinas	disciplinasrelacionadasmatriz	
dim_tempo	N/A	dimensão calculada

Tabela 17 - Mapeamento Origem das Dimensões

Carga dos Mapeamentos

Após a carga das dimensões, os mapeamentos são cadastrados. Para padronizar, os carregamentos possuem a nomenclatura do sufixo `carregaMapeamento` acrescido pelo mapeamento em questão. O job que coordena essa carga é o `CarregaMapeamentos.kjb`, como observado na figura 17.

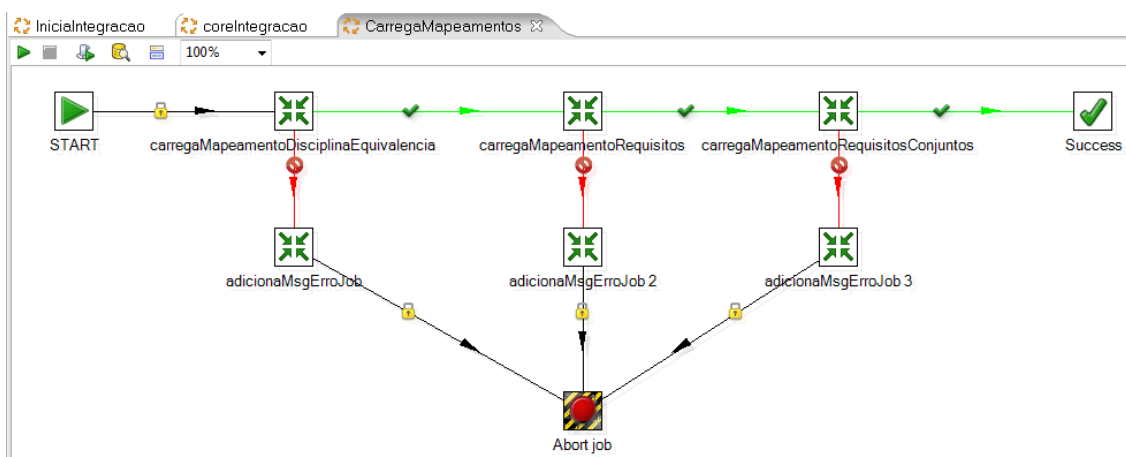


Figura 6 - Carga de Mapeamentos

Os mapeamentos também foram construídos de forma padronizada, modificando apenas a origem dos dados e o destino. Como se trata de uma transformação mais complexa, a Figura 7 destaca alguns trechos, que serão explicados a frente.

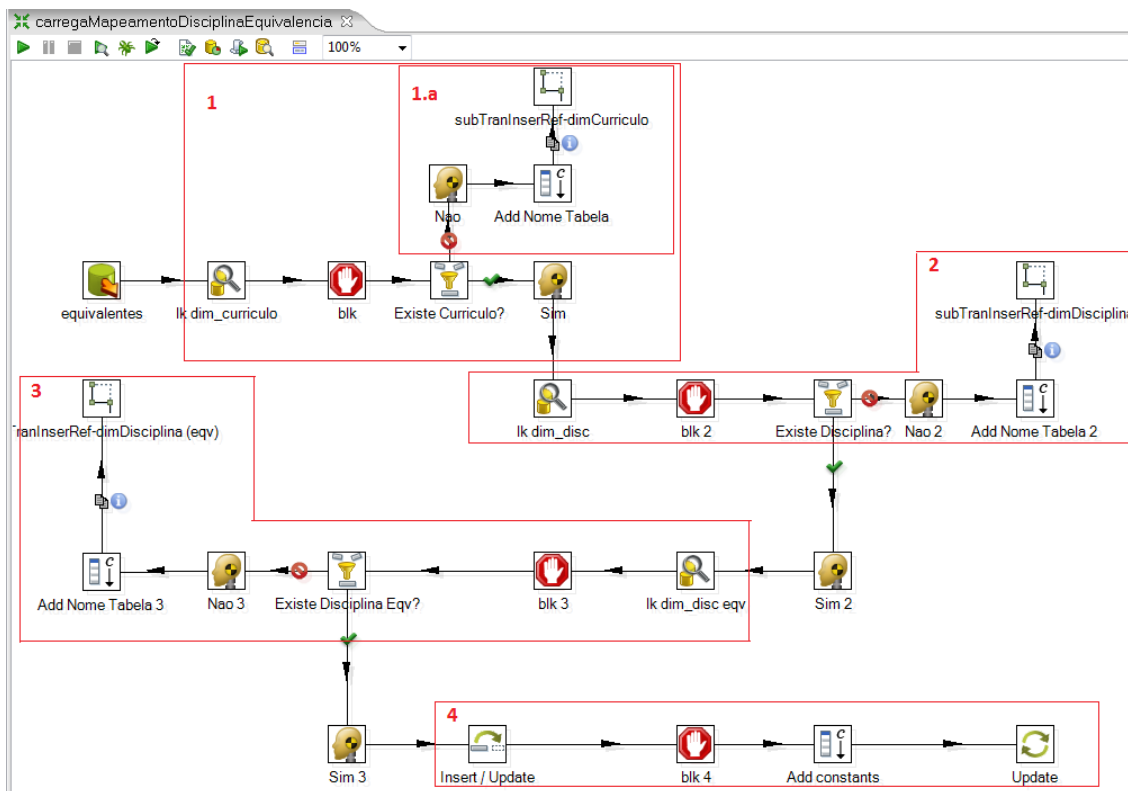


Figura 7 - Carga de Mapeamentos

Após a leitura dos dados provenientes da matriz relacionada, a zona demarcada como 1 (um) na figura 18 é executada. Primeiramente é feito um *lookup* para pegar a referência da dimensão currículo (id artificial) utilizando a chave natural advinda do arquivo. O *step* de bloqueio (*blk*) é utilizado para garantir que não haverá problemas com *deadlock* nas consultas ao banco. Caso o registro daquela dimensão não é encontrado no cubo *sacad*, o filtro “Existe Currículo?” fará com que o registro passe para a execução da área 1.a, demarcada na figura 18. Esta área é responsável por chamar uma *subtransformation* que atualiza o registro na tabela de origem com o erro 7 (ver tabela “Mapeamento de Status da Importação”). O funcionamento destas *subtransformations* é explicado mais adiante, em “Tratamento de Exceções”.

Para os casos de sucesso, o registro passará para as áreas 2 e 3, onde é feito exatamente o mesmo tratamento para a área 1, porém considerando as chaves das disciplinas (disciplina e disciplina equivalente). Nesse ponto também há a possibilidade de não haver a disciplina referenciada na dimensão disciplina, sendo chamado as *subtransformations* responsáveis por marcar o registro com erro 7.

A quarta (4) área trata do *insert* nas tabelas de mapeamento e posterior marcação do registro com status 6, importado com sucesso, na tabela matriz.

Para os mapeamentos, temos a relação de tabela de origem dos dados descrita na Tabela 18.

Mapeamento Tabelas Matriz - Mapeamentos

Mapeamentos	Tabela Matriz Origem
map_disciplina_equivalencia	disciplinasequivalentesmatriz
map_disciplina_requisito	requisitosmatriz
map_disciplina_requisitos_conjuntos	requisitosconjuntosmatriz

Tabela 18 - Relação Dados de Origem-Mapeamentos

Carga dos Fatos

A carga dos fatos é a transformação mais complexa dentro do ETL desenvolvido. Para padronizar, foi adotada a nomenclatura “carrega” acrescido do nome do fato a ser carregado. O *job* que coordena a carga dos fatos é o *carregaFatos.kjb* como observado na figura 19.

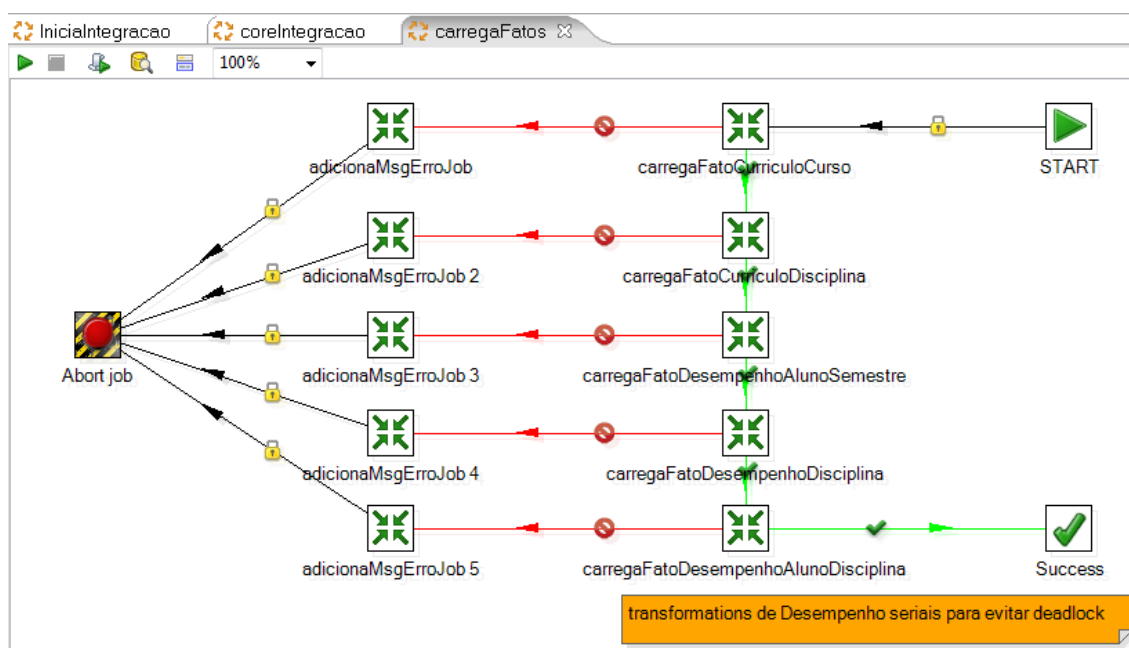


Figura 19 - Carrega Fatos

Os fatos que possuem a mesma tabela matriz como origem foram dispostos sequencialmente para reforçar a necessidade de execução serial, apesar do *PDI* tornar essa execução, paralela ou serial, configurável mesmo que visualmente as transformações estejam paralelas. Porém optou-se por deixar explícito visualmente e com uma nota de atenção.

A carga dos fatos se assemelha com a carga de mapeamentos. A lógica sequencial é a mesma: busca-se o id referente à chave na respectiva dimensão, filtrando e marcando com erro as inexistentes, insere os registros que possuem todas as referências válidas e atualiza-se a origem como registro importado (statuspendente igual a 6).

Como visão geral, será demonstrada uma dessas cargas do fato, explicando as fases envolvidas e depois abordaremos somente os cálculos realizados na carga de alguns fatos, que tornam a carga. Na **Figura** temos uma carga típica de um fato do cubo SACAD.

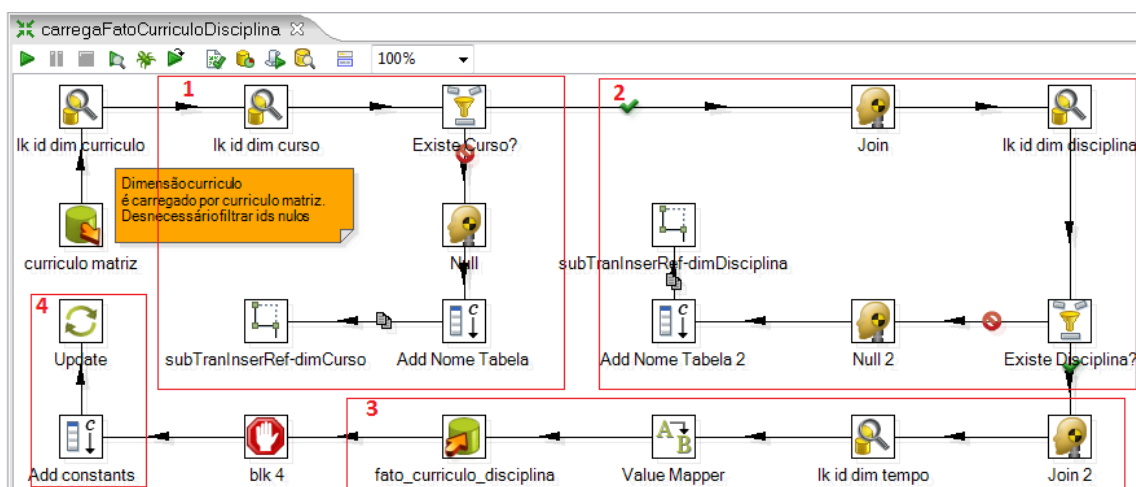


Figura 20 - Carga Fato Currículo Disciplina

Na **Figura**, marcado com o quadrado vermelho 1, temos o *lookup* da referência do curso na dimensão Curso e posterior tratamento para o caso de a referência não tiver sido previamente cadastrada. O tratamento dado a este registro, sem referências, é o mesmo do mapeamento: acionar a subtransformação relativa à dimensão e cadastrar o registro na matriz com statuspendente sete. O mesmo ocorre na área demarcada com dois (2), porém com a entidade disciplina. Na área três temos a carga do fato em si e na área quatro a atualização do registro original como importado.

Observando a **Figura 20**, podemos verificar que duas dimensões não sofreram verificações para identificar se foram previamente cadastradas: dimensão currículo e dimensão tempo. O motivo de não verificarmos a dimensão currículo é que o arquivo de origem deste fato é o mesmo que faz a carga da dimensão currículo. Para a dimensão tempo, não é necessário a verificação, pois é uma dimensão gerada, com um espectro de datas muito amplo, tornando virtualmente impossível não haver uma referência válida.

Na carga do fato `fato_desempenho_aluno_semestre` temos alguns cálculos realizados, demonstrados pela figura 21.

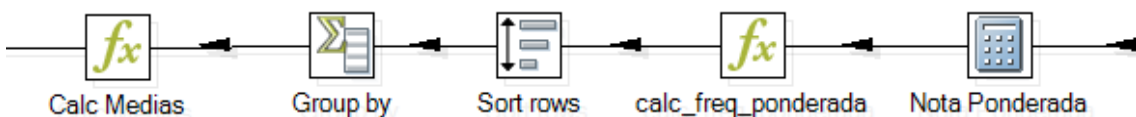


Figura 8 - Cálculos do Fato Desempenho Aluno Semestre

A nota ponderada é calculada pelo step **Nota Ponderada** e é resultante da multiplicação dos créditos pela nota do aluno na disciplina. A frequência ponderada é igual os créditos da disciplina, se o aluno possui frequência suficiente, e igual aos créditos negativos (-

[creditos]), se o aluno possui FI. Após este cálculo, utiliza-se um *step* de *group by*, para que os registros sejam agregados por aluno e semestre. Somente após esta agregação é realizado o cálculo das médias, sendo a média a soma das notas divididas pelo número de disciplinas e a média ponderada a soma das notas ponderadas divididas pela soma dos créditos. Também são calculadas a frequência média (soma da frequência dos alunos pelo número de alunos) e a frequência relativa, que é a frequência ponderada dividida pelo número de alunos. Estes cálculos são tratados para divisão por zero. É importante ressaltar que alguns desses cálculos foram adicionados para enriquecer o datamart, não sendo diretamente utilizados na solução desenvolvida.

Na carga do fato_desempenho_disciplina, é calculado também algumas métricas, como visto no detalhe da figura 22.

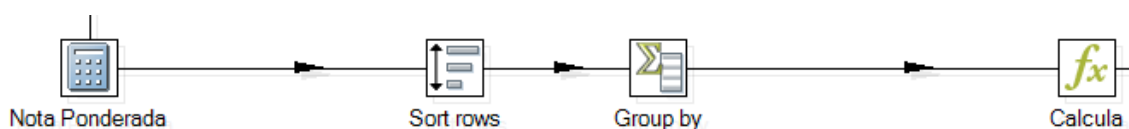


Figura 22 - Cálculo Fato Desempenho Disciplina

O cálculo da Nota Ponderada na figura 22 é o mesmo da figura 21. No *step* Calcula, os seguintes cálculos são aplicados:

- media: soma das notas na disciplinas dividido pela quantidade de alunos;
- media_pond: soma das notas ponderadas dividido pela soma dos créditos. Se os créditos forem 0, o cálculo de média é aplicado.;
- qtd_desistentes: quantidade de alunos menos a soma dos alunos frequentes;
- qtd_reprovados: quantidade de alunos menos quantidade de aprovados.
- IAD:Quantidade de alunos aprovados dividido pelo número de alunos não-desistentes (frequência diferente de FI).

Para os fatos, temos como origem as tabelas matriz relacionadas na Tabela 19.

Mapeamento Tabelas Matriz - Fatos		
Fatos	Tabela Matriz Origem	Observação
fato_curriculo_curso	curriculoparametrosmatriz	
fato_curriculo_disciplina	curriculomatriz	
fato_desempenho_aluno_semestre	alunodesempenhosemestrematriz	
fato_desempenho_disciplina	alunodesempenhosemestrematriz	
fato_desempenho_aluno_disciplina	alunodesempenhosemestrematriz	

Tabela 19 - Mapeamentos Matriz Origem-Fatos

Tratamento de Exceções

Para tratar e identificar possíveis exceções na importação dos arquivos, foram adotados as seguintes abordagens:

- **Arquivo de Log:** os Jobs possuem arquivo de log definido. Utiliza-se steps do tipo “Write to Log” para acrescentar mensagens que ajudem na identificação do erro. Os arquivos de log são gravados na pasta definida pela entrada DiretorioLog, no arquivo config.xml (ver “Configuração do ETL”);
- **Variáveis de Controle de Erro:** a integração utiliza duas variáveis para controlar os erros, JOB.CONTROLE.HASERRO que informa se houve um erro ao executar a importação e JOB.CONTROLE.MSGERRORS, que armazena todas as mensagens de erro adicionadas durante a importação. Com essas variáveis, é possível determinar ao final da importação se houve algum problema e quais mensagens de erros a importação gerou.
- **Envio de E-mail:** a integração é capaz de enviar um e-mail ao endereço configurado no arquivo config.xml (ver “Configuração do ETL”) ao final da importação.
- **Filtro de Referências Inválidas:** ao carregar os mapeamentos e fatos, foi adotado uma verificação se a referência explicitada no arquivo foi previamente cadastrada nas dimensões. Podemos tomar como exemplo um registro (id_disciplina) de disciplina que exista no arquivo Requisitos_[DATA].txt mas não exista no arquivo responsável pela carga da dimensão disciplina, DisciplinasRelacionadas_[DATA].txt

Variáveis de Controle de Erro

Para centralizar o controle da lógica de adicionar mensagens na variável JOB.CONTROLE.MSGERRORS foi criada uma transformação de nome adicionaMsgErroJob.ktr. Esta transformação recebe o parâmetro MSG_ADICIONAL. Esta transformação é mostrada na figura 23.

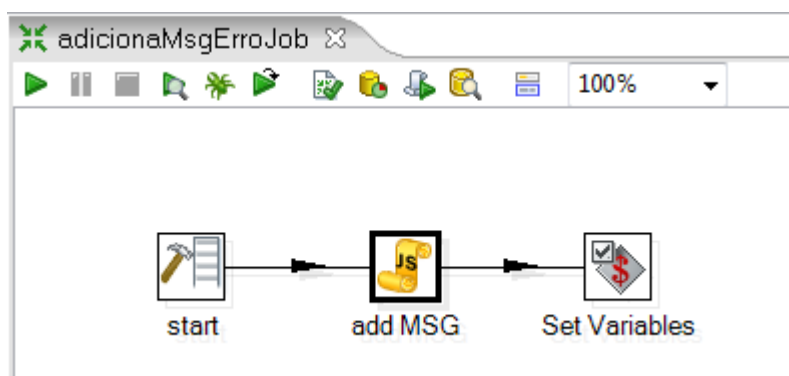


Figura 23 – Adicionar mensagem de erro

O funcionamento desta transformação é simples. No *step* add MSG é resgatado o valor atual de JOB.CONTROLE.MSGERRORS e concatena-se com o parâmetro MSG_ADICIONAL. A variável de controle JOB.CONTROLE.HASERRO é atualizada para “true” e as duas variáveis são novamente atualizadas no *step* “Set Variables”.

Esta abordagem permite que esta transformação seja chamada em qualquer job, bastando configurar a mensagem customizada a ser atualizada no parâmetro MSG_ADICIONAL. Abaixo vemos uma chamada a esta transformação no job lerArquivosCSV.kjb, na figura 24.

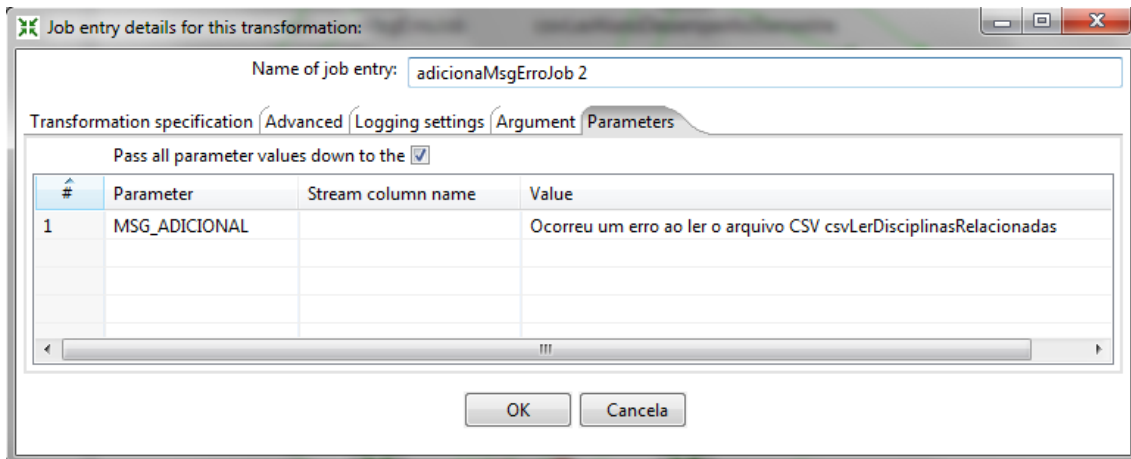


Figura 24 - Chamada a Transformação adicionamensagemerrojob.ktr

Envio de E-mail

O controle de envio de e-mail se dá no job principal, IniciaIntegracao.kjb. Qualquer exceção nos Jobs-filhos que não seja tratada, cairá no fluxo alternativo deste job principal, direcionando para o final da integração, onde há uma avaliação das variáveis de erro.

Caso execute com sucesso, a avaliação direcionará para o envio de e-mail de sucesso, como podemos observar na figura 25.

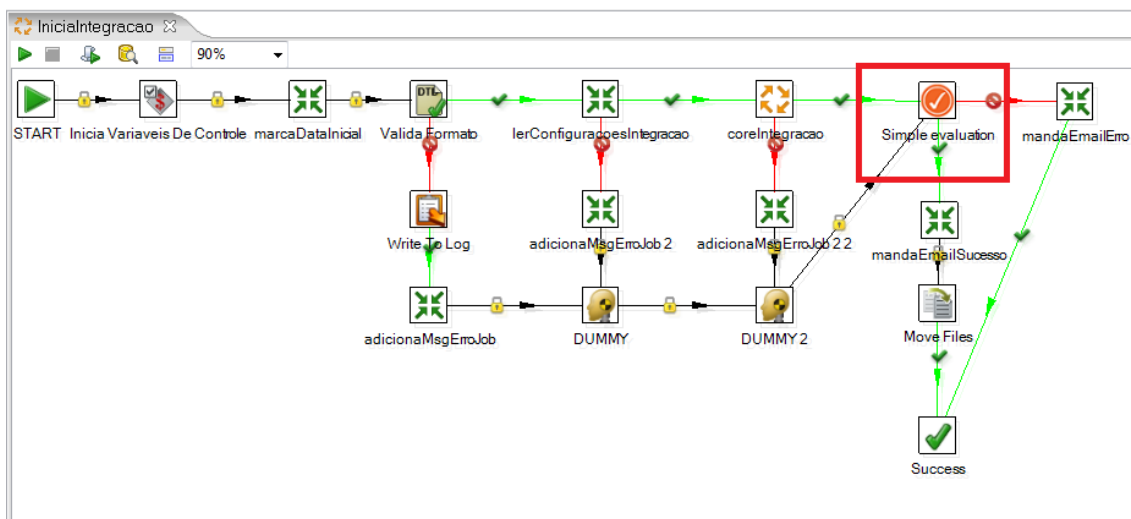


Figura 25 – Avaliação de Erros para Envio de E-mail

O envio de e-mail é configurado pelas informações no config.xml, mencionados no cap. 2.2.1 Configuração do ETL. Um exemplo de e-mail de sucesso pode ser observado na figura 26 e um de erro na figura 27.



Importação 2013/02/17 19:01:28.000 finalizada com Sucesso

1 mensagem

SACAD Importação <sacad.importacao@gmail.com>
Responder a: sacad.importacao@gmail.com:
Para: sacad.importacao@gmail.com:

17 de fevereiro de 2013 19

A importação ocorreu sem erros bloqueantes. Porém é possível que erros de duplicação ou léxicos ocorram. Verifique o relatório de importação e os possíveis erros abaixo: Resumo Importação
----- Houve registros duplicados em outras importações Arquivos importados e movidos para: C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solutions\SACAD\ETL\ArquivosImportados\.....
Relatório acessível em: Relatório Integração 2013/02/17 19:01:28.000 Message date: 2013/02/17 19:23:52.525 Contact information : Person to contact : Abner Luis Telephone number : {EMAIL.FONERESPOSTA}

Figura 26 – E-mail de Sucesso



Erro na Importação

1 mensagem

SACAD Importação <sacad.importacao@gmail.com>
Responder a: sacad.importacao@gmail.com:
Para: sacad.importacao@gmail.com:

Ocorreu um erro bloqueante na importação. Por favor, verifique os erros abaixo e contate o administrador.
Resumo Importação

Houve registros duplicados em outras importações
Ocorreu um erro ao carregar o ktr carregaFatoDesempenhoAlunoSemestre
Houve algum erro no core da importação. Verifique o log e o relatório para identificar.

Message date: 2013/02/17 17:45:21.490

Contact information :

Person to contact : Abner Luis
Telephone number : {EMAIL.FONERESPOSTA}

Figura 27 – E-mail de Erro

Filtro de Referências Inválidas

Para o tratamento de referências inválidas, foram criadas subtransformações, que são transformações que podem ser invocadas dentro de uma transformação. Dessa forma podemos chamar a mesma lógica nos diversos pontos necessários, parametrizando o que for característico da chamada. Essas subtransformações possuem a nomenclatura subTranInserRef- acrescido da dimensão a qual ela se refere. As mesmas são praticamente iguais, modificando apenas as tabelas de referência. Abaixo temos uma dessas subtransformação exemplificada, na figura 25.

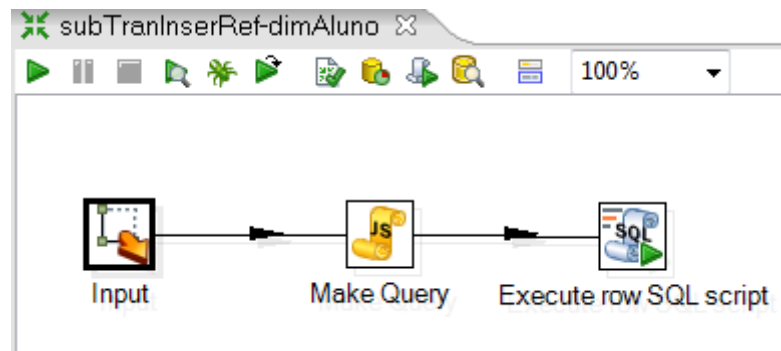


Figura 25 - Subtransformações de Tratamento de Erros de Referência

Como entrada, temos os seguintes parâmetros:

- Idaluno: id que não possui referência na dimensão aluno
- idTabelaOrigem: chave artificial da matriz que contém o registro de origem;
- nomeTabelaOrigem: nome da tabela matriz que contém o registro de origem;
- nomeTabDestino: nome da tabela de mapeamento ou fato que seria preenchida.

Com base nesses parâmetros é formado um comando SQL no step “Make Query”.

Esta concatenação é a seguinte:

```
var sql = "UPDATE "+nomeTabelaOrigem+" SET
StatusPendente=7,StatusDescription='Referência inválida. Uma das chaves não foi
previamente cadastrada.(idaluno="+idaluno+" em "+nomeTabDestino+" )' where id
="+idTabelaOrigem;
```

Não houve a preocupação com SQL Inject, já que a passagem dos parâmetros é feita pelo próprio ETL, sem interação humana.

Esta abordagem permitiu flexibilizar a chamada desta lógica em diversos pontos diferentes da aplicação. Outros tratamentos podem ser adicionados mais tarde nestas subtransformações, isolando a lógica de tratativa de problemas com referências inválidas.

Área Administrativa

O sistema de importação (ETL) foi desenvolvido para ser executado no Pentaho BI Server 4.1.0. Para isso, basta instalá-lo e jogar a pasta SACAD dentro do diretório pentaho-solutions, realizando um refresh do repositório logo após (consulte a documentação).

Uma vez realizado o deploy, é possível acessar a área administrativa da importação, observando as seguintes funções mostradas na figura 26.

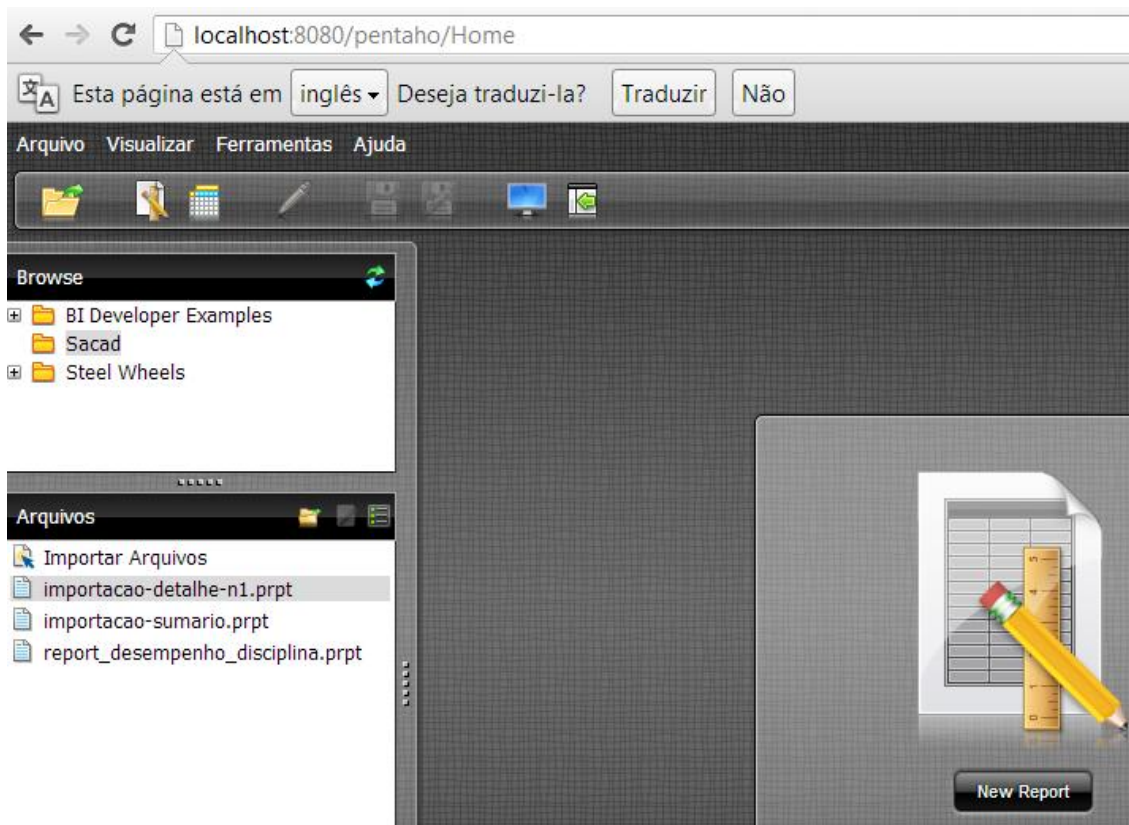


Figura 26 – Área Administrativa

Acessando a navegação do diretório Sacad, podemos visualizar 4 arquivos, que possuem diferentes funções, comentadas na listagem abaixo:

- Importar Arquivos: dispara o job de importação do ETL;
- importacao-detalle-n1.prpt: relatório acessível a partir do importacao-sumario.prpt, não deve ser executado isoladamente;
- importacao-sumario.prpt: relatório que sumariza a importação, utilizando um filtro de data da importação. Explicado no próximo tópico;
- report_desempenho_disciplina.prpt: relatório que responde a primeira pergunta proposta: Quais disciplinas têm maior repercussão no tempo médio para a conclusão de um curso?. Utiliza como filtro o curso e o último ano que se quer observar (período de -5 anos a partir do último ano).

Relatórios de Importação

Ao realizar uma importação, o administrador da ferramenta pode visualizar os resultados através de relatórios. O filtro para determinar a integração é a data da mesma, que é utilizada como chave para identificar quais arquivos foram carregados na mesma integração.

O relatório principal faz uma sumarização da importação selecionada e possui a seguinte estrutura da figura 27.

importacao-sumario.prpt

Report Parameters

Data Importação
17/02/2013 19:01:28

View Report

Resumo Importação

Data da importação: 17/02/2013 19:01:28

Arquivo	Diretório Origem	Linhas	OK	Erros	Autor
Requisitos_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	2.926	2.900	26	-
DisciplinasEquivalentes_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	2.107	813	1.294	-
DadosAlunoCurso_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	6.055	5.703	352	-
Curriculo_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	3.308	3.220	88	-
AlunoDesempenhoSemestre_09-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	172.940	162.882	10.058	-
CurriculoParametros_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	34	25	9	-
DisciplinasRelacionadas_13-11-2012.txt	C:\biserver-ce-3.10.0-stable\biserver-ce\pentaho-solu..	1.968	1.968	-	-

Figura 97 - Relatório Sumarizado da Importação

A coluna Autor é o usuário logado no Pentaho BI Server que executou a transformação. As colunas OK e Erros possuem hiperlink para a visualização *in loco* dos registros na respectiva matriz. Dessa forma é possível identificar as mensagens de erro/sucesso de cada caso e tomar a devida providência, como observado na imagem 28.

Resumo Importação

Matriz referência: requisitosmatriz

Id Matriz	Id Histórico	Linha Arquivo	Tipo Status	Descrição do Status
15	375	15	4	Registros duplicado em outras importações
91	375	91	7	Referência inválida. Uma das chaves não foi previamente cadastrada. (iddisciplina=ECV5352 em map_disciplina_requisito)
99	375	99	4	Registros duplicado em outras importações
162	375	162	4	Registros duplicado em outras importações
238	375	238	4	Registros duplicado em outras importações
254	375	254	4	Registros duplicado em outras importações
290	375	290	4	Registros duplicado em outras importações
736	375	736	4	Registros duplicado em outras importações
744	375	744	4	Registros duplicado em outras importações
825	375	825	4	Registros duplicado em outras importações
1.130	375	1.130	4	Registros duplicado em outras importações
1.223	375	1.223	4	Registros duplicado em outras importações
1.477	375	1.477	4	Registros duplicado em outras importações
1.623	375	1.623	4	Registros duplicado em outras importações
1.761	375	1.761	7	Referência inválida. Uma das chaves não foi previamente cadastrada. (iddisciplina=EEL5166 em map_disciplina_requisito)

Figura 108 - Relatório Detalhado da Importação

Aplicação

Tecnologias envolvidas

A plataforma de desenvolvimento escolhida foi a *Microsoft .NET*, sendo utilizado o *.NET Framework* na sua versão 4.0. A linguagem de programação escolhida foi o *C# 4.0*. A especificação do *C#* é aberta e o *.NET Framework* é gratuito. A escolha dessas tecnologias se deu pela familiaridade que os envolvidos têm com as mesmas, pois isso daria maior produtividade.

A *IDE* utilizada para desenvolver a aplicação é o *Visual Studio 2010 Ultimate*, da *Microsoft*, no sistema operacional *Windows 7 Enterprise*, ambos obtidos através de licença acadêmica disponibilizada pela UFSC.

O SGBD escolhido foi o *MySQL 5.5.16*. A ferramenta de acesso ao banco de dados para administração e consultas *ad-hoc* utilizada foi o *MySQL Workbench 5.2.35 Community Edition*. Ambos são gratuitos.

Para acesso ao SGBD através da aplicação foi utilizado o componente *Dot Connect for MySql 6.80 Express*, que é gratuito.

Arquitetura da aplicação

A parte da solução que é responsável por exibir as informações ao usuário final consiste em uma aplicação *web* desenvolvida no *framework web ASP.NET* versão 4.0 na linguagem *C#*. O serviço responsável pelo pré-processamento de informações é um executável cujo ambiente de execução é o *.NET 4.0*, como observado na figura 29.

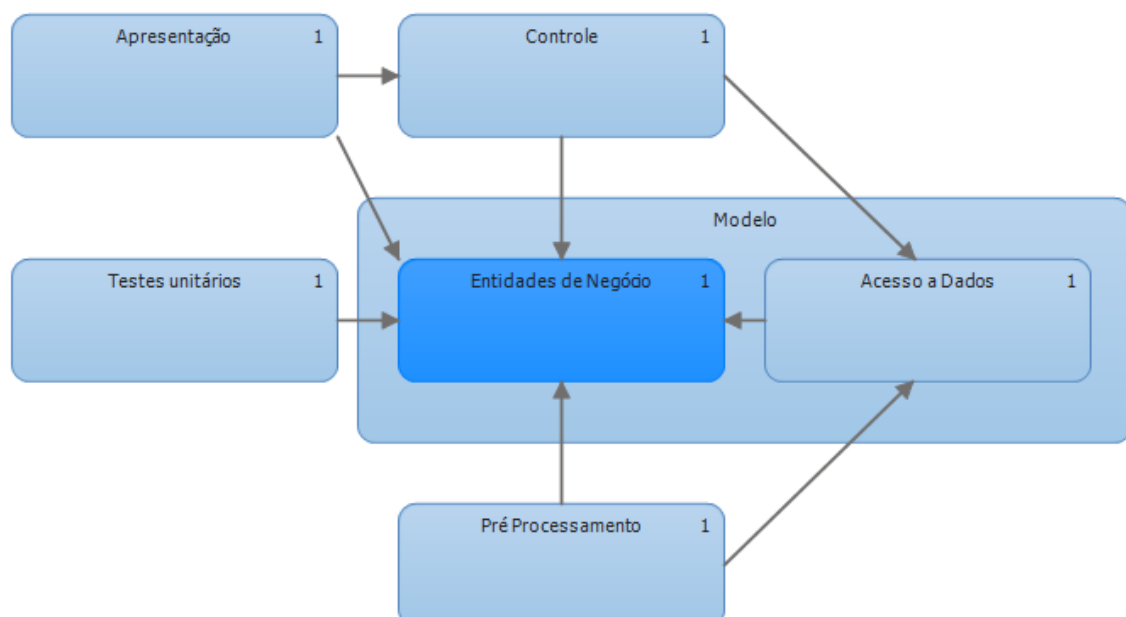


Figura 29 - Arquitetura lógica da aplicação

Aplicação Web

Para o desenvolvimento da aplicação web utilizou-se o padrão *MVC* para separação da lógica de interface, controle e modelo. A Figura mostra a arquitetura lógica da aplicação. Os testes unitários e a camada de pré-processamento não fazem parte da aplicação *web*.

Camada de Apresentação

A camada de apresentação da aplicação é um projeto do tipo *ASP.NET Web Forms 4.0*, estrutura que permite criar páginas *HTML* com componentes estáticos e dinâmicos.

Camada de Controle

A camada de controle da aplicação é um projeto do *Class Library*, estrutura que permite criar código que será empacotado em um arquivo *.dll*, este que é referenciado na camada de apresentação.

Camada de Modelo

A camada de modelo da aplicação é composta de duas camadas menores, sendo estas a camada de Entidades de Negócio e Acesso a Dados. Ambas as camadas estão em seu próprio projeto do tipo *Class Library* e têm responsabilidades distintas.

A camada de Entidades de Negócio é responsável por realizar os cálculos necessários para responder as perguntas que esse trabalho se propõe a responder. A inteligência da aplicação está nessa camada. Essa camada também é responsável por criar objetos de estrutura mais simples que serão enviadas pela camada de controle para a camada de visão, sendo estes objetos chamados de *Data Transfer Object*, ou, simplesmente, *DTO*.

A camada de Acesso a Dados é responsável por se conectar ao SGBD ou qualquer que seja a fonte de dados para alimentar as entidades de negócio.

Rotina de Pré-Processamento

O relatório de jubileamento consolidado é pré-processado através da aplicação *Sacad.PreProcessor*.

Estrutura do Código

O Visual Studio organiza o código em uma *Solution*, os *Projects* dessa *Solution* e os artefatos de código e configuração que pertencem a cada *Project*. A Figura 30 mostra os projetos como são exibidos dentro do *Visual Studio*.

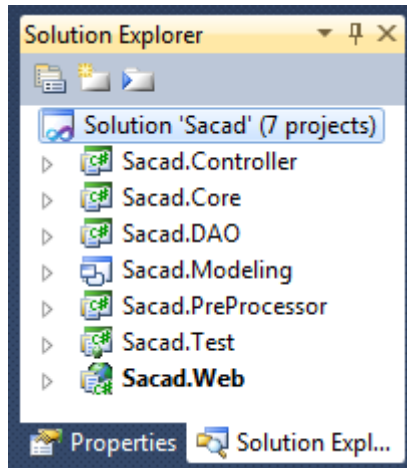


Figura 30 - Solution "Sacad"

- *Sacad.Controller* é um projeto do tipo *Class Library* e corresponde à camada controladora da aplicação *web*;
- *Sacad.Core* é um projeto do tipo *Class Library* e corresponde às entidades de negócio e outras classes de apoio;
- *Sacad.DAO* é um projeto do tipo *Class Library* e corresponde à camada de acesso a dados;
- *Sacad.Modeling* é um projeto do tipo *Modeling* e corresponde aos diagramas do sistema, alguns deles *UML*. Os diagramas *UML* são gerados a partir do código;
- *Sacad.PreProcessor* é um projeto do tipo *Console Application* e corresponde ao pré-processamento dos dados para o relatório de jubramento consolidado;
- *Sacad.Web* é um projeto do tipo *Web Application* e corresponde ao sistema *web* que será acessado pelos coordenadores de curso e alunos.

Detalhamento dos Algoritmos

Nesta seção será detalhada a codificação dos algoritmos para responder as três perguntas.

Primeira Pergunta: Quais disciplinas têm maior repercussão no tempo médio para a conclusão de um curso?

Essa pergunta tem como chave para a resposta o IAD¹ de cada disciplina, atualmente calculado levando em consideração o percentual de alunos matriculados que são aprovados, excluindo-se os desistentes.

O cálculo desse índice é realizado durante o processo de carga dos fatos, mais especificamente do fato de desempenho da disciplina, cuja tabela é a *fato_desempenho_disciplina*, conforme modelo ER na Figura .

¹ Índice de Aproveitamento da Disciplina

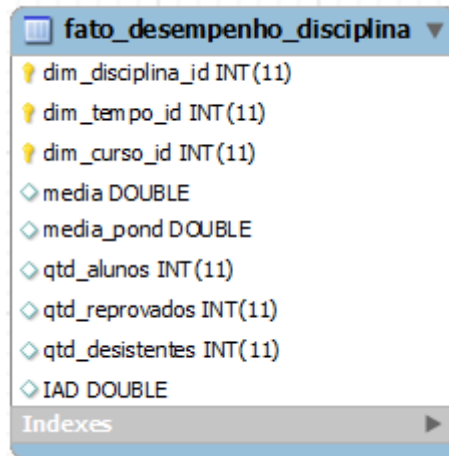


Figura 31 - Tabela de fato do desempenho das disciplinas

Segunda Pergunta: Quais alunos estão com perspectiva de serem jubilados?

Essa pergunta é respondida combinando duas etapas no sistema. A primeira, o pré-processamento, cujos dados serão exibidos na página *Jubilamento.aspx* no projeto *Sacad.Web* e a segunda que é o relatório detalhado do jubramento, acessado através da página *JubilamentoDetalhe.aspx* no projeto *Sacad.Web*.

O cálculo do jubramento é feito nos seguintes passos:

1. A partir de um aluno são buscadas as disciplinas cursadas e as disciplinas restantes de acordo com o currículo do aluno;
2. Com as disciplinas cursadas é verificada qual a média de créditos aprovados por semestre;
3. A partir das disciplinas restantes é criada uma simulação onde o aluno cursaria uma quantidade de créditos por semestre correspondente a sua própria média, sempre obedecendo aos pré-requisitos de cada disciplina;
4. Tendo essa simulação criada já se tem qual o semestre provável de formatura e pode-se responder se esse aluno tem perspectiva de ser jubilado;
5. Também é simulado um caminho ideal, o mais rápido possível, obedecendo à cadeia de pré-requisitos;
6. Com todas essas informações um resumo é montado indicando: créditos restantes para obrigatórias e optativas, média de créditos por semestre, prazo real para se formar, o semestre provável de formatura e dessa maneira indica se o aluno tem perspectiva de jubilar ou não.

Esse algoritmo envolve os casos de uso da Figura . Como visto nesse diagrama, o relatório consolidado depende do pré-processamento dos dados, sendo que este executa o algoritmo do jubramento para todos os alunos tendo como saída os dados necessários para o relatório consolidado. Da mesma forma, ao consultar dentro do relatório consolidado os detalhes de algum aluno específico, o cálculo é novamente feito para que maiores detalhes sejam apresentados ao coordenador. Essa decisão foi tomada porque persistir todas as informações necessárias para exibir os detalhes ocuparia muito espaço de armazenamento. Em contrapartida, a execução do algoritmo para um único aluno é rápida e pode ser calculada toda vez que se deseja extrair essas informações.

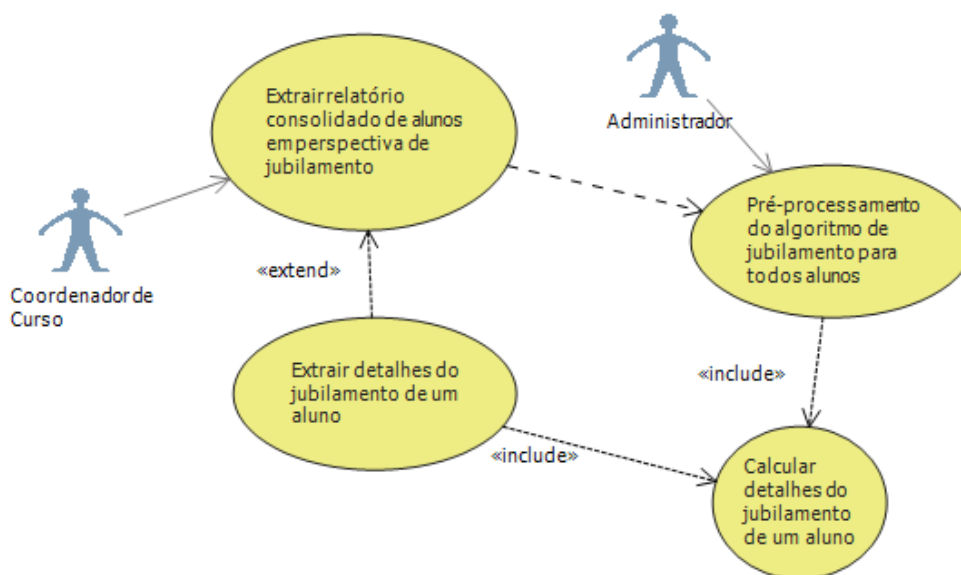


Figura 34 - Casos de uso do algoritmo de jubramento

No código, existem os seguintes componentes e/ou classes responsáveis pelo ciclo de vida do algoritmo de jubramento:

- As páginas “Jubilamento.aspx” e “JubilamentoDetalhe.aspx” que contém o *HTML* necessário para exibir os dados dos relatório consolidado e relatório detalhado, respectivamente;
- A classe “JubilamentoController”, responsável por consultar as entidades de negócio e enviar os resultados para as páginas *ASPX* correspondentes;
- A classe “Simulador”, responsável por efetivamente realizar os cálculos do algoritmo;
- A classe abstrata “ItemAbstrato” e suas derivadas, responsáveis por guardar os dados das disciplinas já cursadas, a serem cursadas e simuladas pelo algoritmo;
- A classe “DisciplinaResumida”, responsável por armazenar dados de contexto sobre as disciplinas do aluno para apoiar o cálculo realizado pela classe “Simulador”;
- A classe “ResultadoJubilamento”, responsável por agrupar todas as saídas geradas pelo algoritmo;
- A classe “ResultadoJubilamentoLinha”, responsável por guardar as informações necessárias de cada linha do relatório consolidado, sendo que cada linha tem como grão um aluno;
- A classe “JubilamentoBatch” responsável por instanciar a classe “Simulador” para cada aluno e persistir os resultados através de instâncias da classe “ResultadoJubilamento”;
- O programa “Sacad.PreProcessor”, responsável por instanciar a classe “JubilamentoBatch”.

O diagrama da Figura mostra em alto nível os componentes citados acima e seus relacionamentos.

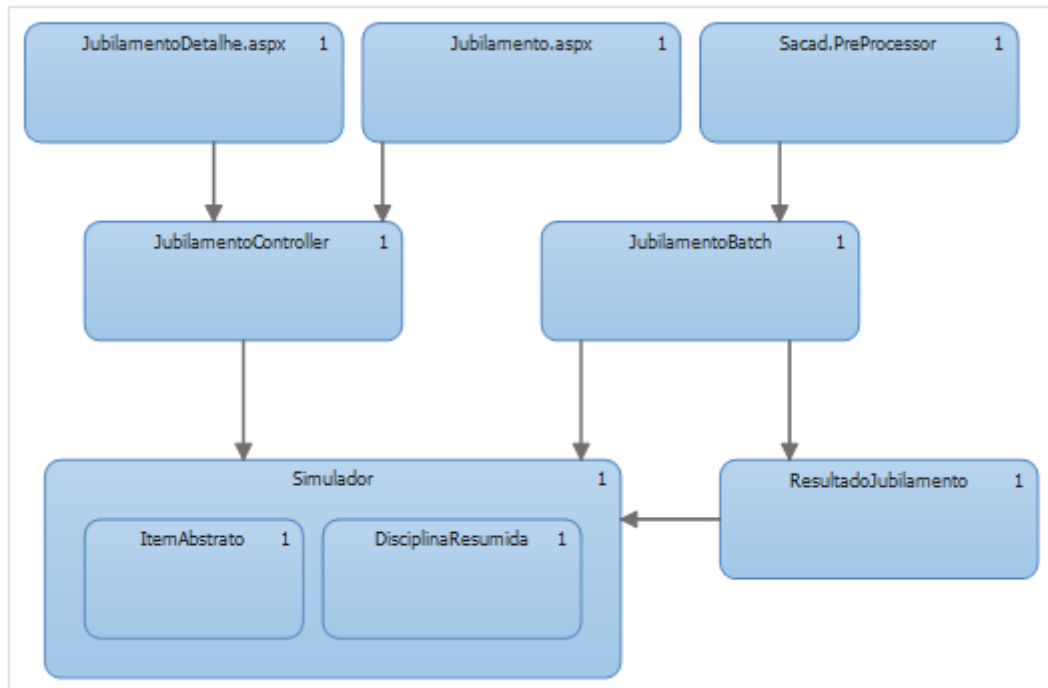


Figura 35 - Principais componentes do algoritmo de jubramento

Rotina de Pré-Processamento

A visão geral da rotina é simples: acessa as entidades de negócio que calculam o risco de jubramento para cada aluno, extrai desse resultado informações essenciais e as persiste para posterior consulta. A **Figura 11** mostra um diagrama de sequência simplificado para entendimento geral do processo.

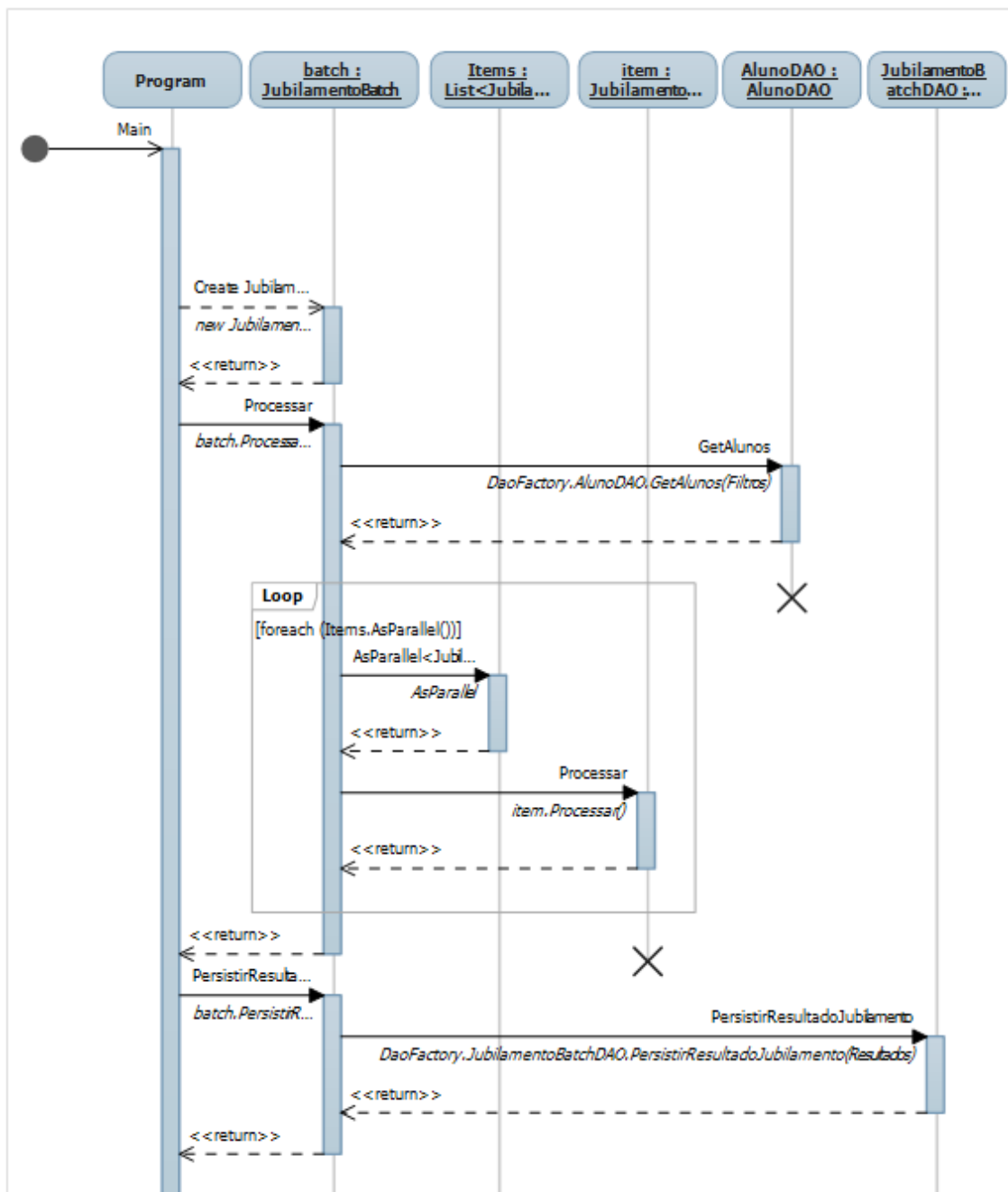


Figura 11 - Diagrama de seqüência simplificado da rotina de pré-processamento

O pré-processamento é um programa do tipo executável, chamado *Sacad.PreProcessor*.

Algoritmo de Jubilamento

Para calcular o resultado do jubilamento tudo gira em torno da classe *Simulador*, no *namespace* *Sacad.Core.Jubilamento*.

Para a exibição dos detalhes do algoritmo na página JubilamentoDetalhe.aspx, a classe Simulador é instanciada dentro da classe controladora JubilamentoController, no método GetDadosJubilamento que recebe como parâmetro a matrícula do aluno. Nesse método, a partir da matrícula, são carregados do banco de dados o objeto Aluno e duas listas, uma com instâncias da classe HistoricoItem e outra com objetos CurriculoItem. A classe Simulador é então instanciada e o método CalcularResultado é chamado. A Figura 13 mostra o código responsável pela explicação acima.

```
public ResultadoJubilamento GetDadosJubilamento(string matricula)
{
    Aluno aluno = DaoFactory.AlunoDAO.GetByMatricula(matricula);
    if (aluno == null)
        throw new ArgumentException("Aluno inexistente: " + matricula, "matricula");
    List<HistoricoItem> historico = DaoFactory.JubilamentoDAO.GetHistoricoAluno(aluno);
    List<CurriculoItem> curriculo = DaoFactory.JubilamentoDAO.GetCurriculoRestanteAluno(aluno);

    var simulador = new Simulador(aluno, curriculo, historico);
    return simulador.CalcularResultado();
}
```

Figura 13 - Código para calcular os detalhes do jubilamento

A Tabela 20 detalha as propriedades da classe Simulador. Todas elas são privadas e servem para o funcionamento interno da classe. O resultado do algoritmo é exposto pelo método CalcularResultado que retorna um objeto da classe ResultadoJubilamento, tendo relacionadas na Figura 12 suas propriedades.

Classe Simulador	
Propriedade privadas	Função
Aluno	Representa o aluno sendo avaliado pelo algoritmo.
Curriculo	Lista com objetos CurriculoItem, representa as disciplinas obrigatórias que ainda precisam ser cursadas.
Optativas	Lista com objetos CurriculoItem, representa as disciplinas optativas disponíveis.
Historico	Lista com objetos HistoricoItem, representa as disciplinas cursadas pelo aluno.
ProximoSemestre	É o semestre que está para ser cursado pelo aluno.
CargaHorariaPorSemestre	É a média de carga horária semestral que o aluno consegue cursar com aprovação, portanto, seu ritmo.
HistoricoSimulado	Lista com objetos SimulacaoItem, representa uma projeção da execução de cada disciplina restante pelo aluno em questão baseado no ritmo do aluno. É um dos principais produtos do algoritmo.
CaminhoMaisRapido	Lista com objetos SimulacaoItem, representa uma projeção de execução das disciplinas restantes, considerando que o aluno fez todas as disciplinas possíveis por semestre. É produto do algoritmo e tem como finalidade indicar a um aluno em situação de risco uma maneira de cumprir o currículo sem jubilar.

SemestreProvavelDe Formatura	Um dos resultados do algoritmo, essa propriedade informa em qual semestre o aluno iria se formar mantendo o ritmo em que está.
Prazo	Prazo para o aluno se formar, baseado no currículo do curso e também em possíveis trancamentos de matrícula (que postergam a data de jubramento).
Resultado	Objeto da classe ResultadoJubilamento, agrega todas as informações produzidas pelo algoritmo. O método CalcularResultado da classe Simulador expõe essa propriedade.
CargaHorariaRestante Obrigatorias	Quantidade de horas restantes de disciplinas obrigatórias para integralização curricular.
CargaHorariaRestante Optativas	Quantidade de horas restantes de disciplinas optativas para integralização curricular.

Tabela 20 - Propriedades da classe Simulador

O método `CalcularResultado` é o corpo do algoritmo e chama dois outros métodos principais, o `CriarSimulacao` e o `CriarCaminhoMaisRapido`.

O método `CriarSimulacao` tem por objetivo criar um suposto caminho de disciplinas que o aluno faria, considerando seu ritmo de carga horária semestral para as disciplinas em que houve aprovação. O método armazena no final uma lista com objetos da classe `SimulacaoItem`. Essa classe herda da classe `ItemAbstrato`, cuja função é representar uma disciplina (propriedade `Disciplina`) num dado momento do tempo (propriedade `Semestre`) para um determinado aluno (propriedade `Aluno`). A classe `SimulacaoItem` não adiciona nenhum tipo de comportamento a classe `ItemAbstrato`, mas tem o propósito de deixar claro sua semântica: é um objeto que representa a simulação do cumprimento de uma disciplina, num dado semestre para um determinado aluno. No diagrama da **Figura 12** a classe `ItemAbstrato` e suas classes filhas são mostradas.

O roteiro do método `CriarSimulacao` é explicado de forma resumida no diagrama de atividades da **Figura 14**.

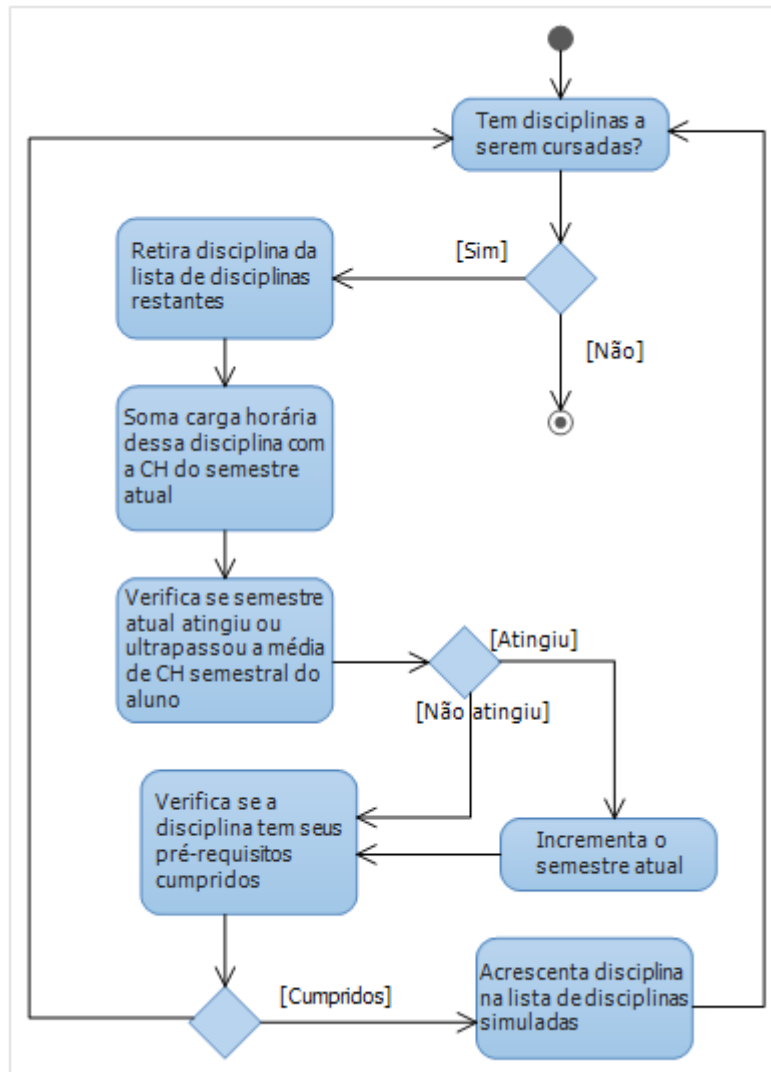


Figura 14 - Diagrama de atividades do método CriarSimulacao

Terceira pergunta: Qual é a expectativa de sucesso de um determinado aluno que se matricula em um determinado conjunto de disciplinas?

A página *ChancesAprovacao.aspx* é a responsável por apresentar os dados e chamar as classes correspondentes para realizar os cálculos.

O sistema nessa funcionalidade não faz validações como cadeias de pré-requisitos no momento da seleção das disciplinas ou se a carga horária mínima ou máxima do curso é excedida.

O diagrama de casos de uso na Figura 15 mostra as possibilidades dessa funcionalidade.

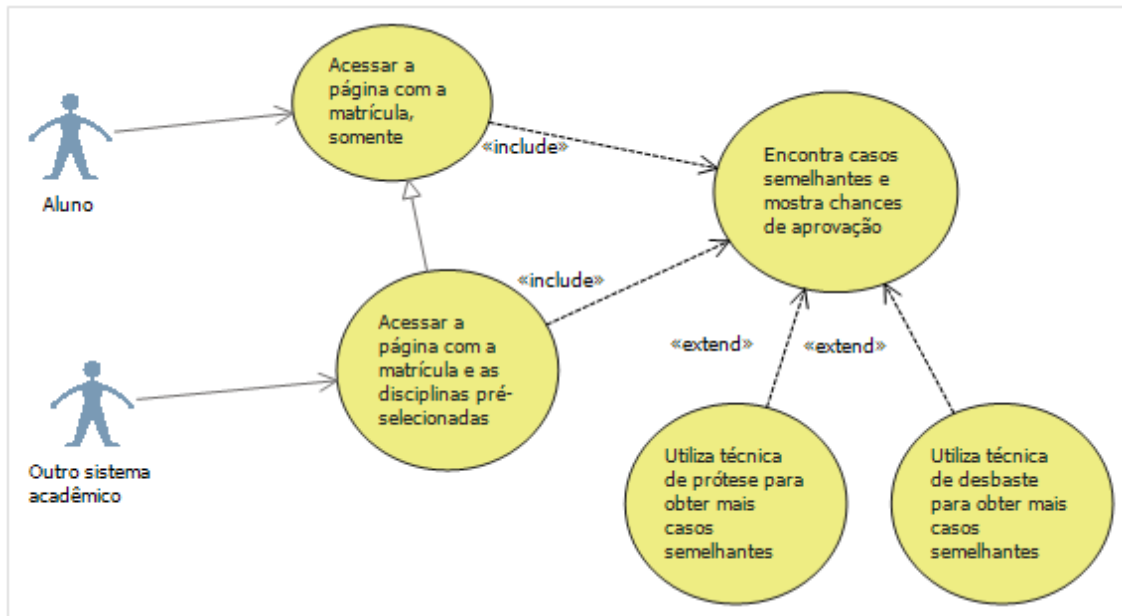


Figura 15 - Diagrama de casos de uso para o algoritmo de chances de aprovação

No código, existem os seguintes componentes e/ou classes responsáveis pelo ciclo de vida do algoritmo de chances de aprovação:

- A página “ChancesAprovacao.aspx” que contém o *HTML* necessário para exibir o formulário de entrada de dados e as saídas correspondentes;
- A classe “ChancesAprovacaoController”, responsável por consultar as entidades de negócio e enviar os resultados para as páginas *ASPX* correspondentes;
- A classe “CalculaChances”, responsável por efetivamente realizar os cálculos do algoritmo;
- A classe “CandidatoProteseDesbasteDTO” responsável por guardar os dados das candidatas aos procedimentos de prótese e desbaste;
- A classe “ResultadoChanceAprovacaoDTO”, responsável por agrupar todas as saídas geradas pelo algoritmo;
- A classe “ChancesAprovacaoDAO”, responsável por realizar as consultas ao banco de dados.

Documentação das classes do sistema

Utilizando a ferramenta *Sandcastle*, foi gerado um arquivo com a documentação do sistema. Esse arquivo bem como o que é necessário para gerá-lo estão junto ao código fonte do sistema.

Classes

A seguir estão relacionadas todas as classes utilizadas para executar os algoritmos das três perguntas.

Classe	Descrição
ChancesAprovacaoController	Controller da página chances de aprovação
ControllerBase	Classe controladora base
JubilamentoController	Controller das páginas de jubramento
Aluno	Classe que representa um aluno em um determinado momento do tempo
MapRequisitos	Classe estática de referência que mapeia todos os requisitos de todas as disciplinas existentes no sistema. Seus dados são carregados na inicialização do sistema e servem como um cache para informações consultadas repetidamente.
TimeProvider	Abstração do relógio do sistema para permitir realizar testes sem alterar o relógio do computador
Semestre	Representa um semestre de um curso. Contém a informação do ano e se é o primeiro ou o segundo semestre.
CalculaChances	Classe responsável pelo algoritmo de calcular as chances de aprovação de um aluno
CandidatoProteseDesbasteDTO	DTO responsável por trafegar os dados de uma disciplina para ser utilizada no algoritmo de chances de aprovação
DisciplinaDTO	DTO que representa uma disciplina
ResultadoChanceAprovacaoDTO	DTO responsável por trafegar os dados referente ao resultado do algoritmo chances de aprovação
CurriculoItem	Representa uma disciplina a ser cursada para um determinado aluno em um determinado semestre
CursoDTO	DTO que representa um curso
DisciplinaResumida	DTO que representa uma disciplina, de forma resumida com dados essenciais para o cálculo do jubramento




HistoricoItem	Disciplina do histórico de um determinado aluno
ItemAbstrato	Classe abstrata que representa uma disciplina em um dado semestre para um determinado aluno no algoritmo de jubramento
RelatorioJubilamentoLinha	DTO que representa uma linha no relatório de jubramento consolidado
ResultadoJubilamento	Persistirá essas informações pra serem pesquisadas pelo coordenador
SimulacaoItem	Representa uma disciplina simulada pelo algoritmo de jubramento
Simulador	Classe principal para o cálculo do algoritmo do jubramento
AlunoDAO	Classe de acesso a dados para assuntos relacionados a alunos
ChancesAprovacaoDAO	Classe de acesso a dados de assuntos relacionados a funcionalidade chances de aprovação
DAO	Classe base de acesso a dados
DAOFactory	Classe Factory responsável por instanciar classes DAO
DisciplinaDAO	Classe de acesso a dados para consultas de assuntos relacionados a disciplinas
JubilamentoBatchDAO	Classe responsável ao acesso a dados para o pré-processamento do relatório de jubramento
JubilamentoConsolidadoDAO	Classe responsável para fornecer os dados para relatório de jubramento consolidado
JubilamentoDAO	Classe de acesso a dados para o cálculo do algoritmo de jubramento
FiltroAluno	Classe que representa os filtros preenchidos pelo usuário na tela de chances de aprovação

Detalhamento das classes


A seguir estão relacionados os membros (métodos e propriedades) das classes acima citadas.

ChancesAprovacaoController

Métodos


	Name	Description
	Calcula	Delega a chamada do cálculo para a classe correspondente
	DisciplinasAluno	Retorna as disciplinas disponíveis para o aluno em questão
	DisciplinaUnica	Retorna uma disciplina de acordo com o critério de pesquisa na tela

Atributos


	Name	Description
	DaoFactory	(Herdado de ControllerBase .)

ControllerBase

Métodos




	Name	Description
	InicializaConteudo	Faz operações de configuração, no momento apenas carrega o mapeamento de requisitos das disciplinas numa cache.

Atributos


	Name	Description
	DaoFactory	Para ter acesso aos objetos DAO

JubilamentoController

Métodos

	Name	Description
	GetCursos	Retorna os cursos existentes no sistema para mostrar na página
	GetDadosJubilamento	Exibe os dados do relatório detalhado para um determinado aluno
	GetDadosRelatorio	Exibe os dados do relatório consolidado para um determinado curso

Atributos

	Name	Description
	DaoFactory	(Herdado de ControllerBase .)

Aluno

Atributos




	Name	Description
--	------	-------------

	Ativo	Se o aluno está ainda ativo
	CargaHorariaMinima	Carga horária mínima de disciplinas obrigatórias para o aluno se formar
	CargaHorariaMinimaOptativas	Carga horária mínima de disciplinas optativas para o aluno se formar
	CreditoSemestralMaximo	Máximo de créditos por semestre
	CreditoSemestralMinimo	Mínimo de créditos por semestre
	Id	Chave do banco de dados
	IdCurriculo	Chave do currículo no banco
	IdTempo	Chave da dimensão tempo no banco
	Ingresso	Data de ingresso do aluno
	Matricula	Matrícula do aluno
	Nome	Nome do aluno
	Prazo	Prazo para conclusão do curso

MapRequisitos


Métodos

	Name	Description
--	------	-------------

	GetRequisitosDe	Retorna lista com os pré-requisitos de uma determinada disciplina
	PreRequisitoDe	Retorna se uma disciplina é pré-requisito de outra
	SetMapRequisitos	Faz o setup inicial dos dados




TimeProvider

Atributos






	Name	Description
	Now	Retorna a “data atual”, que pode ser verdadeiramente a data do sistema ou qualquer data que o programador deseje durante o desenvolvimento da aplicação.

Semestre

Construtores

	Name	Description
	Semestre(DateTime)	Considera o dia de hoje para criar um objeto Semestre que represente o próximo semestre letivo
	Semestre(String)	Recebe um parâmetro string no formato AAAA/MM
	Semestre(Int32, Int32)	Recebe um parâmetro para o ano e outro para o semestre

Métodos

	Name	Description
	AdicionaSemestres	Acrescenta ao semestre atual o número de semestres desejado e retorna uma nova instância com o novo valor. A instância atual não é modificada
	DaData	Cria um Semestre baseado num objeto DateTime
	Equals(Semestre)	Verifica se dois semestres são iguais
	Proximo	Retorna uma nova instância que representa o próximo semestre
	ToString	(Sobrescreve ValueType.ToString() .)


Atributos

	Name	Description
	DataBase	Objeto DateTime que representa o Semestre

CalculaChances








Métodos

	Name	Description
--	------	-------------

	Calcular	Método principal da classe, executa o algoritmo em si (consulte o manual técnico para detalhes do algoritmo)
---	-----------------	--

CandidatoProteseDesbasteDTO



Atributos

	Name	Description
	Aprovado	Se o aluno foi aprovado naquela disciplina naquele semestre
	Disciplina	Disciplina em questão
	FrequenciaSuficiente	Se o aluno possuiu frequência suficiente
	Matricula	Matrícula do aluno
	Nota	Nota na disciplina
	Quantidade	Quantidade de ocorrências dessa disciplina no universo avaliado
	Semestre	Semestre cursado

DisciplinaDTO





Atributos

	Name	Description
--	-------------	--------------------

	CodigoDisciplina	Código da disciplina
	Nome	Nome da disciplina




ResultadoChanceAprovacaoDTO

Atributos

	Name	Description
	CodigoDisciplina	Código da disciplina
	Percentual	Representação visual do atributo PercentualValor
	PercentualValor	Valor da chance do aluno ser aprovado na disciplina
	QuantidadeAmostras	Quantidade de amostras avaliadas

CurriculoItem



Atributos

	Name	Description
	Aluno	(Herdado de ItemAbstrato.)
	CargaHoraria	(Herdado de ItemAbstrato.)
	Disciplina	(Herdado de ItemAbstrato.)

	Semestre	(Herdado de ItemAbstrato .)
---	-----------------	---







CursoDTO


Atributos

	Name	Description
	Id	Código do curso no banco de dados
	Nome	Nome do curso

DisciplinaResumida







Atributos

	Name	Description
	Codigo	Código da disciplina
	Creditos	Créditos da disciplina
	IAD	Índice de Aproveitamento Académico da disciplina
	Id	Chave no banco de dados da dimensão disciplina
	Nome	Nome da disciplina
	Optativa	Se é optativa ou não

	SemestrePadrao	Fase ideal do curso para cursar a disciplina, sensível ao contexto do semestre atual
---	----------------	--



HistoricoItem



Atributos

	Name	Description
	Aluno	(Herdado de ItemAbstrato.)
	Aprovado	Se o aluno foi aprovado na disciplina
	CargaHoraria	(Herdado de ItemAbstrato.)
	Disciplina	(Herdado de ItemAbstrato.)
	Nota	A nota do aluno na disciplina no semestre
	Semestre	(Herdado de ItemAbstrato.)

ItemAbstrato

Atributos

	Name	Description
	Aluno	O aluno que cursou a disciplina
	CargaHoraria	A carga horária da disciplina

	Disciplina	A disciplina cursada
	Semestre	O semestre em que a disciplina foi cursada










RelatorioJubilamentoLinha

Atributos

	Name	Description
	CreditosObrigatoriasRestantes	Créditos restantes nas disciplinas obrigatórias
	CreditosOptativasRestantes	Créditos restantes nas disciplinas optativas
	CreditosPorSemestre	Média de créditos por semestre
	Matricula	Matrícula do aluno
	NomeAluno	Nome do aluno
	Prazo	Prazo para conclusão do curso
	PrevisaoDeJubilar	Se o aluno está com risco de jubilar (True/False)
	PrevisaoDeJubilarTexto	Se o aluno está com risco de jubilar (Sim/Não), útil para apresentar no relatório
	SemestreProvavelDeFormatura	Semestre simulado em que o aluno se formaria

ResultadoJubilamento

Atributos

	Name	Description
	Aluno	Aluno sendo avaliado
	CaminhoMaisRapido	Cadeia de disciplinas que faria o aluno se formar em menos tempo
	CreditosObrigatoriasRestantes	Créditos restantes nas disciplinas obrigatórias
	CreditosOptativasRestantes	Créditos restantes nas disciplinas optativas
	CreditosPorSemestre	Média de créditos por semestre
	DentroDoPrazo	Se o aluno está dentro do prazo para se formar
	Historico	Lista de disciplinas já cursadas
	HistoricoSimulado	Simulação de disciplinas a serem cursadas pelo aluno, baseado em seu ritmo
	Prazo	Prazo para concluir o curso
	SemestreProvavelDeFormatura	Semestre simulado em que o aluno se formaria



SimulacaoItem

Atributos

	Name	Description
	Aluno	(Herdado de ItemAbstrato.)
	CargaHoraria	(Herdado de ItemAbstrato.)
	Disciplina	(Herdado de ItemAbstrato.)
	Semestre	(Herdado de ItemAbstrato.)


Simulador




Métodos

	Name	Description
	CalcularResultado	Executa o algoritmo principal do jubramento
	PreRequisitosCumpridos	Retorna True caso os pré-requisitos para determinada disciplina tenham sido cumpridos, de acordo com o histórico do aluno em questão

AlunoDAO




Métodos

	Name	Description
	ExecuteNonQuery	(Herdado de DAO.)

	ExecuteReader	(Herdado de DAO .)
	GetAlunos	Retorna uma lista de alunos baseado num filtro passado
	GetByMatricula	Retorna um objeto Aluno buscando pela matrícula


ChancesAprovacaoDAO


Métodos

	Name	Description
	CandidatosProteseDesbaste	Retorna uma lista com as disciplinas candidatas a prótese e desbaste baseado em um aluno e as disciplinas escolhidas
	ExecuteNonQuery	(Herdado de DAO .)
	ExecuteReader	(Herdado de DAO .)

DAO




Métodos

	Name	Description
	ExecuteNonQuery	Executa um comando SQL no banco de dados sem esperar uma tabela como retorno. Retorna o número de linhas afetadas

	ExecuteReader	Executa um comando SQL no banco de dados esperando um objeto da interface <code>IDataReader</code> , este que permite iterar entre as linhas do resultado recebido
---	-------------------------------	--



DisciplinaDAO

Métodos

	Name	Description
	ExecuteNonQuery	(Herdado de DAO .)
	ExecuteReader	(Herdado de DAO .)
	GetCadeiaDePreRequisitos	Retorna cadeia de pré-requisitos de todas as disciplinas do sistema
	GetDisciplina	Retorna uma disciplina baseada no seu código

JubilamentoBatchDAO


Métodos

	Name	Description
	ExecuteNonQuery	(Herdado de DAO .)
	ExecuteReader	(Herdado de DAO .)

	PersistirResultadoJubilamento	Responsável por gravar no banco de dados o resultado do pré-processamento
---	---	---




JubilamentoConsolidadoDAO



Métodos

	Name	Description
	ExecuteNonQuery	(Herdado de DAO.)
	ExecuteReader	(Herdado de DAO.)
	Relatorio	Retorna as linhas do relatório de jubilamento consolidado

JubilamentoDAO



Métodos

	Name	Description
	ExecuteNonQuery	(Herdado de DAO.)
	ExecuteReader	(Herdado de DAO.)
	GetCurriculoRestanteAluno	Retorna disciplinas restantes para a conclusão do curso

	GetCursos	
	GetHistoricoAluno	Retorna disciplinas pertencentes ao histórico de um determinado aluno

FiltroAluno

Atributos

	Name	Description
	Cursos	Cursos filtrados na tela
	Matriculas	Matrículas filtradas na tela

8.2. Artigo

SACAD - Desenvolvimento de sistema de apoio a redução de evasão nas Instituições de Ensino Superior

Abner L. S. Rolim, Pedro E. F. Lima, Renato Cislighi, José L. Todesco, José M. da Silva

Departamento de Informática e Estatística
Universidade Federal de Santa Catarina/CTC/INE
Florianópolis, Brasil

{abner, pedro.lima, cislighi, tite}@inf.ufsc.br, jmarcos@npd.ufsc.br

Abstract - This paper presents the conception and development of a decision support system for education institutions concerned with the problem of student evasion. Educational institutions can be considered a business model, but unlike enterprises, they lack a set of proper tools to address the several management problems such institutions face. The system developed proposes to answer three strategic questions: Which courses have more repercussion in the average time to a graduation conclusion? Which students are in perspective of being expelled due to a time limit constraint? What is the success rate of a determined student who enrolls in a determined set of courses?

Keywords: *Datamart; support decision system.*

INTRODUÇÃO

A abertura de mercados, fomentada pela globalização, trouxe para os modelos de negócios a necessidade de criar mecanismos eficientes de gerência de seus processos e produção. Em um modelo competitivo, ganha quem apresenta maior eficiência e controle sobre seus custos, antevendo possíveis fatores e interferências. Dado a dinamicidade das variáveis envolvidas, é natural que a computação, com ênfase na área de sistemas de informação, atue como ferramental para a tomada de decisão.

A instituição de ensino, com o objetivo de lucro ou não, pode ser considerada um modelo de negócio, visto que os mesmos fatores atuam na sua existência: competitividade e necessidade de manter a saúde financeira, justificando o investimento realizado. Porém em contraste com instituições comerciais, nota-se a um lapso na utilização e modernização de métodos, sistemas e softwares que auxiliam na gestão da saúde de uma IES. Enquanto em uma empresa a utilização de Business Intelligence (BI) é praxe, nas instituições de ensino o enfoque é diferente [1]. O Sistema de Informação Acadêmica, da forma que está construído, caracteriza-se como um aplicativo operacional; cumpre apenas um papel de registro da vida acadêmica do estudante, sem no entanto subsidiar a Instituição, e, mais diretamente, os Coordenadores dos cursos de graduação, para as análises

necessárias e estudos de acompanhamento da vida acadêmica dos estudantes [1].

Segundo a pesquisa sobre evasão no ensino superior brasileiro, realizada em 2007 pelo Instituto Lobo para o Desenvolvimento da Educação, da Ciência e da Tecnologia, a taxa anual média de evasão no ensino superior brasileiro foi de 22%, com pouca oscilação, mas mostrando tendência de crescimento. A preocupação é refletida por ações do próprio governo que tem como uma das metas globais, aumentar a taxa de conclusão média para 90% nas IES federais, através do Programa de Reestruturação e Expansão das Universidades Federais, Reuni, lançado em 2007 [2].

Tal atenção é cabível, uma vez que um aluno desistente representa um custo: para as IES de duas formas, a vaga ociosa e o investimento perdido e para o mercado de trabalho, perde ou adia a entrada de um potencial recurso. O impacto econômico pode ser considerado também na visão do aluno, que tarda o possível aumento na sua renda, uma vez que a entrada no mercado de trabalho com titulação será adiada.

Considerando os aspectos expostos acima, foi proposto [3] um Sistema Acadêmico de Apoio a Decisão, SACAD, descrito pelo autor como um instrumento de informação, produtividade e gestão; com o qual coordenadores (gestores) e discentes possam manusear. Um SAD especializado em fornecer apoio para diminuir a evasão nas IES. Na concepção do autor, este sistema de apoio se baseia em responder três questões básicas [3]: *Quais disciplinas têm maior repercussão no tempo médio para a conclusão de um curso? Quais alunos estão com perspectiva de serem jubilados? Qual é a expectativa de sucesso de um determinado aluno que se matricula em uma determinada carga (conjunto de disciplinas)?*

A contribuição deste trabalho é a codificação e documentação de um sistema de apoio a decisão, o SACAD, que forneça dashboards, gráficos e projeções que permita responder as questões sugeridas no artigo o qual este sistema se baseia.

O universo de dados testados e homologados virá de um Data Mart (DM) com informações adquiridas do CAGR, limitando-se aos cursos de Ciências da Computação e Sistema de Informação e considerando o ano 2001/2 como inicial. Serão feitos testes e a homologação será feita pelo orientador, finalizando o escopo deste trabalho. Para um futuro desenvolvimento na UFSC, utilizando tecnologia e processo de desenvolvimento homologado por seu NPD, o sistema contará com um Manual Técnico, onde estarão relatados todos os passos de desenvolvimento e explicações detalhadas da arquitetura, com o intuito de que o futuro desenvolvimento seja apenas uma tradução da aplicação desenvolvida aqui.

Para utilizar esta solução, as IES interessadas devem ter métricas semelhantes a adotadas na UFSC no corrente ano. Ou seja, seus índices de aproveitamento do aluno, notas e outros dados quantitativos devem seguir o mesmo tipo de dado adotado na UFSC. A divisão temporal também deve se manter em dois semestres ao ano e o layout de entrada dos arquivos deve obedecer o padrão exposto no manual de utilização, presente no apêndice deste trabalho.

TRABALHOS RELACIONADOS

Este trabalho como um todo se baseia em um artigo de Cislaghi [3], que propõe e define um *Datamart* de apoio a decisão para coordenadores de curso e alunos. As três perguntas estratégicas já citadas foram propostas no artigo de Cislaghi.

SISTEMAS DE APOIO A DECISÃO E DATAMARTS

Em uma definição formal [4], um *Data Warehouse* (DW) "é uma coleção de dados orientados a assuntos, integrados, variáveis no tempo e não voláteis que dão suporte ao processo de tomada de decisão". Em uma definição mais prática e comercial, Kimball [5] resume como um recurso de apresentação dos dados de uma empresa que permite a formulação de consultas. Destaca também que estes dados não devem ser organizados em torno de um modelo de entidades relacionais, pois visa a legibilidade da informação e performance. Kimball, com sua interpretação *bottom-up* da constituição de um *Data Warehouse*, também resume um *Data Warehouse* como a união de todos *Data Marts* constituintes.

Um DM, por sua vez, é um subconjunto de um DW, geralmente projetado para servir especificamente um grupo de interessados e tem como fonte um assunto/setor também específico. Na visão de Kimball, é a base para constituir o *Data Warehouse*. Porém, para Inmon [4], os DM seriam alimentados por sumarizações dos dados disponíveis em um *Data Warehouse* (visão *top-down*). Portanto, um DM pode depender ou não de um DW para existir. Em nosso trabalho, utilizamos a abordagem *bottom-up*, criando um DM inicial, dado que nosso objetivo era bastante específico.

DESENVOLVIMENTO

Para prover as informações a serem analisadas pela proposta do trabalho, escolhemos aplicar a metodologia *bottom-up*, desenvolvendo um DM que pudesse prover a estrutura necessária à resposta.

A ferramenta de ETL escolhida para o projeto foi o *Pentaho Data Integration* (PDI), anteriormente chamado de *Kettle*, na versão 4.1 *Community*. O PDI é *open source* e *freeware*. A escolha se deu também por conhecimento prévio na utilização da ferramenta.

Em todo o projeto, foi utilizado o *MySQL 5.5.16* para o SGDB, incluindo na camada de *stage* do ETL. A modelagem do banco foi feita utilizando o *MySQL Workbench 5.2.35 Community Edition*.

A administração do ETL, bem como visualização de relatórios de importação, é feita através do *Pentaho BI Server*, na sua versão *Community 3.10.0*. Dessa forma é possível ter uma solução integrada e permite que mais relatórios ou novas visões dimensionais sejam adicionadas.

Para apresentação das respostas e interação com os usuários finais do sistema, optou-se por uma aplicação *web*, sendo que a plataforma de desenvolvimento escolhida foi a *Microsoft .NET*, sendo utilizado o *.NET Framework* na sua versão 4.0. A linguagem de programação escolhida foi o *C# 4.0*. A especificação do *C#* é aberta e o *.NET Framework* é gratuito.

O planejamento do projeto envolveu o autor do artigo SACAD, os autores deste trabalho e o gestor da ferramenta CAGR, de onde viriam as informações. Nesta fase, os papéis já estavam pré-definidos e a motivação, benefícios entre outras justificativas, explícita pelo embasamento teórico do trabalho.

Definir os requisitos foi uma fase tranquila, pois a expectativa já estava definida e o acesso ao mantenedor do sistema-fonte (CAGR) era feito de forma direta, com reuniões se necessário. Deste modo, não foi preciso adequar a linguagem, dado o perfil técnico dos envolvidos. Os critérios de negócio estavam explícitos pelo escopo e pelo próprio artigo. O desempenho exigido não era requisito, pois o processamento aconteceria ao fim do semestre, podendo durar o tempo necessário.

A arquitetura técnica foi simplificada consideravelmente por não haver implantação em produção, no escopo deste trabalho. Não foi preciso dimensionar a infraestrutura. Acordou-se que os dados seriam exportados pelo CAGR conforme o layout que a equipe escolhesse. Como diagrama básico de arquitetura, chegou-se ao seguinte modelo da figura 1.

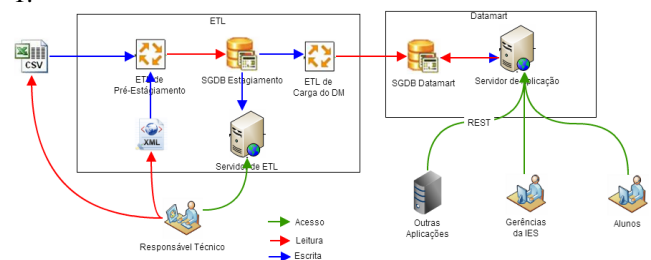


Figura 1. Arquitetura técnica da solução

importados em 22 minutos e 22 segundos. As medidas foram tomadas em um computador com a seguinte configuração: *Intel Core i3-M330* com 4 GB de memória *ram* a 1066 MHz e um *HD Sata II 7200 RPM*.

Como ferramenta de verificação da importação, foram criados dois relatórios. Um sumarizado, com o resumo de cada arquivo de uma determinada importação e outro o detalhamento das linhas que resultaram em erro ou sucesso.

O relatório consolidado de alunos em perigo de jubramento precisa ser pré-processado para todos os alunos, pois seria inviável em termos de desempenho processar o algoritmo de jubramento todas as vezes que fosse necessário visualizar esse relatório.

Assim, a rotina de pré-processamento processou 2 semestres dos 316 alunos disponíveis para testes em 15 segundos. Esse pré-processamento é realizado apenas uma vez, e o relatório que o consome leva em média 340 milissegundos para processar e exibir os dados.

Para medir se a funcionalidade de chances de aprovação estava com um desempenho aceitável, foi criado o seguinte caso de teste: separa-se uma matrícula aleatória, e para essa matrícula é simulado a inscrição em 4 disciplinas, em seguida 5 disciplinas e por fim 3 disciplinas. É feito uma média do tempo de processamento das três interações. Neste teste 50 amostras foram colhidas e se obteve o tempo médio de resposta de 812 milissegundos. O tempo de processamento é medido através do log de requests do *IIS 7*, servidor web no qual o sistema roda.

De maneira semelhante a funcionalidade de chances de aprovação, uma matrícula foi sorteada e o relatório foi aberto para medir o tempo de processamento. Para 50 amostras, obteve-se a média de 1130 milissegundos por requisição.

Essas medições foram tomadas em um computador com a seguinte configuração: *Intel Core i5-560m* com 4 GB de memória *ram* a 1066 MHz e um *HD Sata II 7200 RPM*.

CONCLUSÕES E TRABALHOS FUTUROS

Conclusões

O sistema foi desenvolvido conforme especificado por Cislaghi [3]. O funcionamento do sistema foi validado utilizando dados dos cursos de Sistemas de Informação e Ciência da Computação da UFSC. Com o DM populado, a instituição de ensino que utilizar o SACAD terá uma ferramenta para a gestão do problema de evasão nas Instituições de Ensino Superior, podendo ser ampliada com novas visões no futuro e integrada a outros DM's. Com alertas pró-ativos a figura do coordenador de curso terá mais informação para identificar e tratar os casos de alunos e disciplinas problemáticas, embasando uma ação em conhecimento. Espera-se também um melhor rendimento do aluno já que o sistema o ajudará a se matricular em disciplinas adequadas ao seu perfil.

O sistema como está pode ser implantado na UFSC ou qualquer instituição de ensino, desde que esta gere os arquivos necessários para alimentar o sistema, sendo esses

arquivos, juntamente com seu *layout*, relacionados no apêndice correspondente neste trabalho.

Trabalhos Futuros

A seguir relacionamos possíveis melhorias para o sistema levantadas durante o desenvolvimento.

Criar interface *REST* para permitir integração direta com outros sistemas. Isso permite que as funcionalidades do sistema criado sejam incorporadas nos sistemas da universidade sem que o usuário final perceba que está utilizando um outro sistema. Para o caso específico da UFSC, essa integração pode se dar no sistema CAGR, onde no momento da matrícula semestral o aluno poderia já visualizar suas chances de aprovação ao escolher as disciplinas.

Melhoria do índice de aproveitamento da disciplina. Hoje a fórmula é simples e poderiam ser acrescentadas outras variáveis, como a carga horária da disciplina e quantidade de disciplinas que a tem como pré-requisito.

O sistema foi desenvolvido para considerar cursos de 2 semestres anuais. Para cursos com uma divisão anual diferente é necessário fazer algumas adaptações nos algoritmos, bem como no ETL.

Tornar a área de estagiamento, hoje no formato de *log*, também dimensional.

Ajustes visando *performance (tuning)* na base de dados, considerando o futuro uso em produção.

REFERÊNCIAS

- VELOSO, Tereza C. M. A. A Evasão nos Cursos de Graduação da Universidade Federal de Mato Grosso, campus Universitário de Cuiabá - 1985/2 a 1995/2 um processo de exclusão. 2001. 193p. Dissertação. (Mestrado), Programa de Pós-Graduação em Educação. Universidade Federal de Mato Grosso, UFMT, Cuiabá, 2001
- BRASIL. Governo Federal. Ministério da Educação. Reuni - Reestruturação e Expansão das Universidades Federais: Diretrizes Gerais. Documento Elaborado pelo Grupo Assessor nomeado pela Portaria nº 552 SESu/MEC, de 25 de junho de 2007, em complemento ao art. 1º §2º do Decreto Presidencial nº 6.096, de 24 de abril de 2007.. Disponível em: <<http://portal.mec.gov.br/sesu/arquivos/pdf/diretrizesreuni.pdf>>. Acesso em: 15 nov. 2012.
- CISLAGHI, Renato; SILVA FILHO, Vasco; ALBUQUERQUE, Almir S.; SARTORI, Rejane. SACAD - Construção de um Data Mart para apoio aos coordenadores e acadêmicos de cursos de graduação. In: VI Colóquio Internacional sobre Gestão Universitária na América do Sul, 2006, Blumenau. VI Colóquio Internacional sobre Gestão Universitária na América do Sul - Mudanças e Perspectivas na Gestão Universitária, 2006.
- INMON, W. H.. Building the Data Warehouse. 3. ed. Usa: John Wiley & Sons Inc, 2002.
- KIMBALL, Ralph; REEVES, Laura; ROSS, Margy. The Data Warehouse Lifecycle Toolkit :Expert Methods for Designing, Developing, and Deploying Data Warehouses. Nova York: John Wiley & Sons Inc, 1998
- PENTAHO COMMUNITY (Usa). Organization (Comp.). Latest Pentaho Data Integration (aka Kettle) Documentation. Documentação da Ferramenta de ETL Pentaho Data Integration. Disponível em:

<<http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation>>. Acesso em: 20 fev. 2013.

ROLIM, Abner; LIMA, Pedro. SACAD - SISTEMA ACADÊMICO DE APOIO A DECISÃO: Desenvolvimento de sistema de apoio a redução de evasão nas Instituições de Ensino Superior