



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS



ANÁLISE, PROJETO E IMPLEMENTAÇÃO DE UM *DATA WAREHOUSE* PARA AMBIENTES DE TESTES DE *SOFTWARE*

RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA (PIBIC/CNPq/INPE**)**

Leandro Evaristo de Oliveira (Faculdade de Tecnologia (Fatec) Professor
Jessen Vidal, Bolsista PIBIC/CNPq)
E-mail: leandro.oliveira5@fatec.sp.gov.br

Nandamudi Lankalapalli Vijaykumar (LAC, Orientador)
E-mail: vijay@lac.inpe.br

COLABORADORES

MSc. Érica Ferreira de Souza (CAP)

Julho de 2012

Ficha será revisada pelo SID.

Dados Internacionais de Catalogação na Publicação

Oliveira, Leandro E. ; Vijaykumar, Nandamudi L.
ANÁLISE, PROJETO E IMPLEMENTAÇÃO DE UM DATA
WAREHOUSE PARA AMBIENTES DE TESTES DE SOFTWARE -
São José dos Campos: INPE, ano da publicação.
i + 0p. ; (aa/bb/cc/dd-TDI)

Relatório final de projeto de iniciação científica - Instituto
Nacional de Pesquisas Espaciais, São José dos Campos, 2012
Orientador: Nandamudi Lankalapalli Vijaykumar.

1. Data Warehouse. 2. Ontologia. 3. Teste de software
I. Análise, projeto e implementação de um Data Warehouse para
ambientes de testes de software.

CDU

Copyright 2012 do MCT/INPE. Nenhuma parte desta publicação pode ser reproduzida, armazenada em um sistema de recuperação, ou transmitida sob qualquer forma ou por qualquer meio, eletrônico, mecânico, fotográfico, reprográfico, de microfilmagem ou outros, sem a permissão escrita do INPE, com exceção de qualquer material fornecido especificamente no propósito de ser entrado e executado num sistema computacional, para o uso exclusivo do leitor da obra.

Copyright 2012 by MCT/INPE. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, microfilming or otherwise, without written permission from the INPE, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use of the reader of the work.

FOLHA DE APROVAÇÃO

CONFECCIONADA PELO SPG E INCLUÍDA PELO SID.

“Procure ser um homem de valor, em vez de ser um homem de sucesso”.

Albert Einstein

AGRADECIMENTOS

Agradeço ao CNPq pela oportunidade da bolsa PIBIC, ao orientador do projeto Nandamudi Lankalapalli Vijaykumar e a colaboradora Érica Ferreira de Souza por todo apoio, paciência e tempo empregado.

Agradeço ao Instituto Tecnológico de Aeronáutica - ITA; a Agência Nacional de Águas - ANA; a Financiadora de Estudos e Projetos - FINEP; e a Fundação Casimiro Montenegro Filho – FCMF e ao Coordenador Técnico no ITA Professor Dr. Adilson Marques da Cunha por disponibilizar os dados do Projeto FINEP 5206/06 para realização deste trabalho.

RESUMO

Dado o crescimento exponencial de volume de dados vindos de diversas fontes de conhecimento dentro das organizações, torna-se necessária a automatização das tarefas de aquisição, processamento, análise e disseminação do conhecimento. Projetos de engenharia de *software* também não fogem destes cenários gerando um elevado número de informações. Os envolvidos no projeto passam a enfrentar problemas, tais como: dificuldade de sistematizar as informações geradas; dificuldade para reutilizar o conhecimento por falta de vocabulário comum; perda de capital intelectual da organização; e a não representação do conhecimento. Diante deste contexto, o objetivo deste trabalho consiste em dois aspectos: i) realização de uma pesquisa sobre o estado da arte em *Data Warehouse*; e ii) levantamento e análise de requisitos de um projeto lógico (a modelagem) do *Data Warehouse* e sua implementação. A partir dessas etapas, informações serão armazenadas no *Data Warehouse* para que possam ser investigadas técnicas de pesquisa, análise e preparação de tais informações. Os dados utilizados para criação do *Data Warehouse* serão baseados em dados de testes gerados a partir de um projeto de teste de *software*. A modelagem do *Data Warehouse* será construída a partir de uma ontologia que descreve o processo de teste. Tal ontologia está sendo desenvolvida em um trabalho de pesquisa em doutorado no curso de Computação Aplicada (CAP) no Instituto Nacional de Pesquisas Espaciais (INPE).

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1 - Exemplo de Esquema Estrela para o processo de pedidos.....	6
Figura 2 – Estrutura da ontologia de processo de teste de <i>software</i>	11
Figura 3 – Propriedades <i>data type</i> da ontologia.	12
Figura 4 – Propriedades <i>object</i> da ontologia.	12
Figura 5 – Pseudo código de conversão dos dados da ontologia para um formato simplificado de XML. 13	13
Figura 6 – Modelo estrela do <i>Data Warehouse</i> para o processo de teste de <i>software</i>	16
Figura 7 – Processo de ETL com a ferramenta Pentaho Data Integration.	17
Figura 8 - Estrutura proposta para a solução.	18
Figura 9 - Gráfico de quantidade de testes por perfil.	19
Figura 10 - Gráfico de quantidade de testes por severidade do defeito.	20
Figura 11 - Gráfico de quantidade de testes por resolução.	20
Figura 12 - Gráfico de quantidade de testes por importância do caso de teste.	21

LISTA DE TABELAS

Pág.

Tabela 1 - Sistemas Operacional VS Sistemas Analítico	4
---	---

LISTA DE SIGLAS E ABREVIATURAS

DW	<i>Data Warehouse</i>
ETL	<i>Extract Transform and Load</i>
GC	Gestão do Conhecimento
INPE	Instituto Nacional de Pesquisas Espaciais

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO.....	1
2 <i>DATA WAREHOUSE</i>	3
2.1. Conceito e características	3
2.2. Sistemas de Informação: Operacional x Analítico	3
2.3. Fatos, dimensões e medidas	4
2.4. Teste de <i>Software</i> e Qualidade de Processo	6
2.5. Ontologia.....	7
2.5.1. Classificação	8
2.5.2. Linguagens de Representação	8
2.5.3. Ferramentas.....	9
2.5.4. Áreas aplicadas	9
3 ONTOLOGIA DE PROCESSO DE TESTE DE <i>SOFTWARE</i>	11
3.1.1. Ferramentas.....	13
3.1.2. Jena	13
4 MODELO DO <i>DATA WAREHOUSE</i>	15
4.1. Carga dos dados no <i>Data Warehouse</i> através do processo de Extract Transform and Load (ETL) 17	17
4.2. Visão geral da solução.....	18
5 VALIDAÇÃO DA ESTRUTURA PROPOSTA	19
5.1. Perguntas	19
5.1.1. Qual a quantidade de execuções dos testes por perfil da equipe?	19
5.1.2. Qual a quantidade de testes por severidade do defeito?	20
5.1.3. Qual a quantidade de testes por resolução do defeito?.....	20
5.1.4. Qual a quantidade de testes por importância do caso de teste?.....	21
6 CONCLUSÃO.....	23
REFERÊNCIAS BIBLIOGRÁFICAS	24
APÊNDICE A - COMANDOS SQL DE CONSULTA AO <i>DATA WAREHOUSE</i>	25
A.1 Comando SQL quantidade de testes executados por perfil.....	25
A.2 Comando SQL quantidade de testes por severidade.....	25
A.3 Comando SQL quantidade de testes por resolução.....	25
A.4 Comando SQL quantidade de testes por importância do caso de teste	25

1 INTRODUÇÃO

Dado o crescimento exponencial de dados vindos de diversas fontes de conhecimento dentro das organizações, torna-se necessário a automatização das tarefas de aquisição, processamento, análise e disseminação do conhecimento. As organizações necessitam gerenciar de forma efetiva as informações geradas em seu ambiente de produção para promover a melhoria contínua dos processos utilizados e gerar conhecimento que de suporte às decisões futuras.

Diante deste contexto, a Gestão do Conhecimento (GC) tem se posicionado para exercer um papel fundamental para as organizações. A GC tem a capacidade de gerir e reter conhecimento, tornando-se um dos fatores mais importantes no desenvolvimento de soluções. O objetivo da GC é promover o surgimento de conhecimento novo, seu armazenamento e compartilhamento por toda a organização (O'Leary, 2001).

Projetos de engenharia de *software* geram um elevado número de informações. Os envolvidos no projeto passam a enfrentar problemas, tais como: dificuldade de sistematizar as informações geradas; dificuldade para reutilizar o conhecimento por falta de vocabulário comum; perda de capital intelectual da organização; e a não representação do conhecimento. Para apoiar as atividades de tratar as informações e codificar o conhecimento, para torná-lo acessível e tratável, diversas tecnologias têm sido aplicadas, como Banco de Dados, Máquinas de Busca, Internet, Ontologia, dentre outras (O'Leary, 1998).

Dado este contexto, o objetivo deste trabalho consiste em dois aspectos: i) realização de uma pesquisa sobre o estado da arte em *Data Warehouse*; e ii) levantamento e análise de requisitos de um projeto lógico (a modelagem) do *Data Warehouse*, a sua implementação num sistema gerenciador de banco de dados. A partir dessas etapas, informações serão armazenadas no *Data Warehouse* para que possam ser investigadas técnicas de pesquisa, análise, preparação de tais informações. Os dados utilizados para criação do *Data Warehouse* serão baseados em dados de testes gerados a partir de um projeto

de teste de *software*. A modelagem do *Data Warehouse* será construída partir de uma ontologia que descreve o processo de teste e está sendo desenvolvida em um trabalho de pesquisa em doutorado no curso de Computação Aplicada (CAP) no Instituto Nacional de Pesquisas Espaciais (INPE).

2 DATA WAREHOUSE

Em uma definição formal, (JENSEN et al., 2010) “*Um Data Warehouse é um grande repositório de dados integrados obtidos de diversas fontes em uma empresa com a finalidade específica de análise de dados*”.

2.1. Conceito e características

Data Warehouse é uma base de dados criada para agrupar um grande volume de dados de maneira centralizada permitindo o fácil acesso a estes. Sob essa perspectiva pode assumir diferentes modelos sendo os mais comuns: Modelo Estrela, Modelo Floco de neve (ADAMSON, 2010) , (JENSEN et al., 2010).

Data Warehouse é utilizado como repositório dos dados, utilizando-se do processo de ETL (*Extract Transform Load*) é feita a extração e a transformação dos dados de acordo com as regras estabelecidas, essa etapa pode ser bem complexa e é crucial para manter a consistência e logo após essa etapa os dados são carregados efetivamente no *Data Warehouse*.

Visto que seu uso é exclusivo para análise dos dados, ele deve ser modelado de maneira que as consultas que normalmente são complexas tenham um desempenho otimizado assim aumenta a agilidade na apresentação dos dados, isso é possível com a técnica de modelagem dimensional.

2.2. Sistemas de Informação: Operacional x Analítico

A diferença mais marcante entre os modelos está na maneira como os dados estão organizados dentro do banco de dados, em quanto no modelo relacional as tabelas estão relacionadas entre si, no modelo dimensional os dados são distribuídos em dimensões o que permite a visualização dos dados de diferentes perspectivas.

Abaixo a Tabela 1, faz uma comparação dos dois modelos, mostrando a diferença entre eles:

	Sistema Operacional	Sistema Analítico
Propósito	Execução de um processo do negócio.	Avaliação de um processo do negócio.
Escopo de utilização	Inserção, Atualização, Remoção e Consulta	Consulta
Dados temporais	Atual	Atual e histórico
Escopo de interação	Transação individual	Transações agrupadas
Modelo de Otimização	Atualização concorrente	Alto desempenho em consulta
Princípio do Modelo	Entidade-Relacionamento (ER)	Dimensional (Esquema Estrela ou Cubo)

Tabela 1 - Sistemas Operacional VS Sistemas Analítico

Fonte: Adaptado de Adamson (2010).

2.3. Fatos, dimensões e medidas

Os fatos são as representações de algo do mundo real que a empresa deseja avaliar, a dimensão é o aspecto pelo qual os fatos podem ser avaliados. Na hora de realizar a consulta as dimensões fornecem o contexto utilizado para filtrar ou agrupar as medições que serão avaliadas. (ADAMSON, 2010) , (JENSEN et al., 2010).

Medidas é um conjunto de atributos numéricos usados para medir os fatos, tornando-os mensuráveis por tanto de grande importância do ponto de vista analítico.

Segundo Jensen et al. (2010) as medidas podem ser classificadas em valores aditivos, semi-aditivos e não-aditivos. Os Aditivos são os atributos que podem ser somados em qualquer dimensão e ainda assim mantêm sentido, os semi-aditivos mantêm sentido apenas com cálculo de média em determinadas dimensões e os não-aditivos são os atributos que não tem contexto quando somado em nenhuma dimensão.

- Exemplo de valor aditivo: soma das vendas de um determinado departamento.
- Exemplo de valor semi-aditivo: média de estoque de produtos.
- Exemplo de valor não aditivo: registro de uma foto.

O modelo estrela é composto por uma tabela central, a qual representa os fatos de um *Data Warehouse*. Conectada à tabela de fatos estão as dimensões, como mostra a Figura 1.

Segundo Adamson (2010), a distinção entre o modelo estrela e o cubo está na arquitetura do Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado, quando o modelo dimensional é implementado em um banco de dados relacional será chamado de Estrela e quando implementado no banco de dados multidimensional será chamado de Cubo.

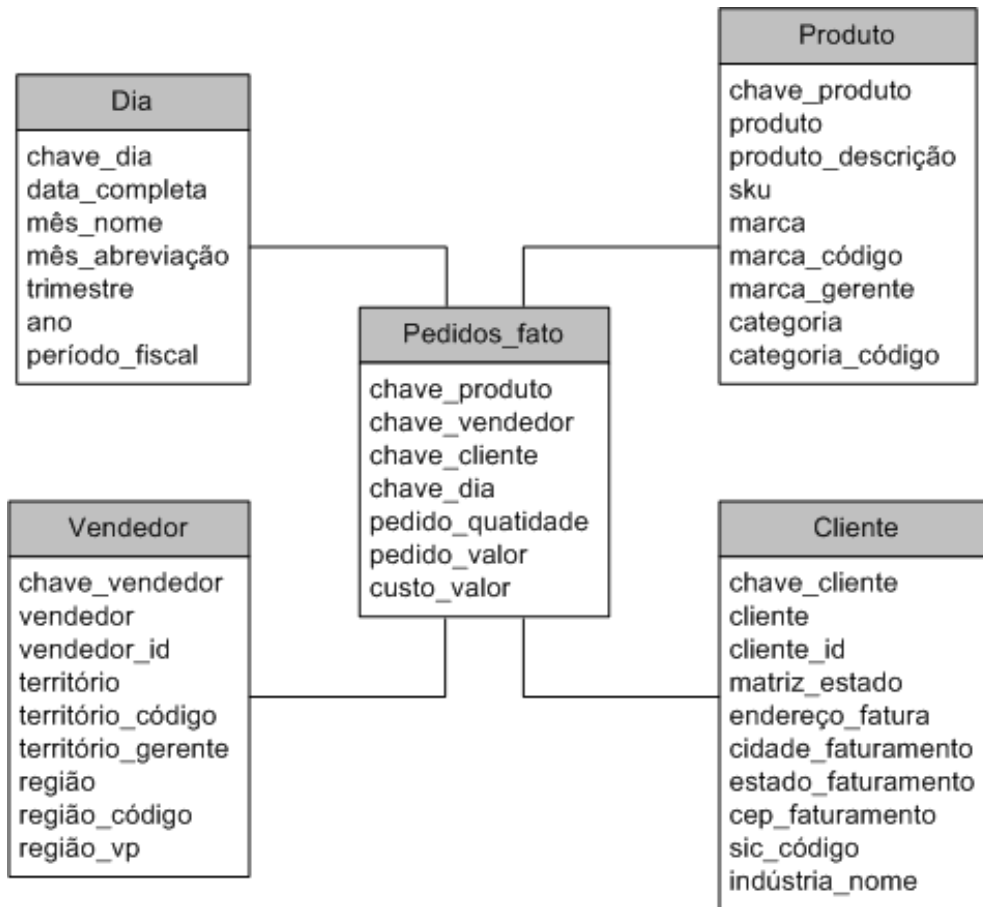


Figura 1 - Exemplo de Esquema Estrela para o processo de pedidos.

Fonte: Adaptado de Adamson (2010).

2.4. Teste de Software e Qualidade de Processo

Baseado em BASTOS et al. (2007) diversos tipos de testes podem ser utilizados conforme a necessidade de testar determinada funcionalidade ou parte específica do sistema, pode ser que determinado sistema não utilize todos estes tipos, mas certamente alguns deles serão perfeitamente aplicáveis quando deseja-se garantir o mínimo de qualidade agregada ao *software*.

Veja alguns dos tipos de testes e qual sua finalidade:

- Testes de estresse: este tipo de teste consiste em deixar o sistema sob um estado crítico verificando como será o comportamento do sistema trabalhando

com recursos que vão próximos ou além de sua capacidade (BASTOS apud SAMS, 2005).

- Testes de execução: de maneira geral este tipo de teste busca verificar a desempenho do sistema, qual seu tempo de resposta e se o tempo atende ao esperado quando executado no ambiente de produção.

- Testes de recuperação: este teste busca verificar o tempo que será necessário para que o sistema recupere-se em caso de falha, isso diz respeito principalmente aos dados armazenados pelo sistema e suas respectivas transações sobre estes.

- Testes de segurança: este teste verifica se a proteção dos dados estão de acordo com as normas da organização, garantindo que a informação não esteja acessível a pessoas externas ou mesmo a determinadas pessoas internas da organização.

- Testes de requisitos: este teste procura identificar se as funcionalidades do sistema estão de acordo com os requisitos dos usuários.

- Testes de tratamento de erros: este teste visa verificar se o sistema é capaz de tratar uma condição inesperada e executar o devido tratamento a esta condição, sem isto interrompa o seu funcionamento.

2.5. Ontologia

De acordo com (Gruber, 1993), a ontologia é uma especificação formal explícita de uma conceituação compartilhada. Ontologias envolvem a descrição de conceitos em um determinado domínio de conhecimento, com suas propriedades e restrições. A ontologia fornece uma descrição exata do conhecimento em uma linguagem formal para facilitar a comunicação, integração, busca, armazenamento, compartilhamento, reutilização e representação do conhecimento (O'Leary, 1998).

2.5.1. Classificação

São várias as classificações de ontologia encontradas na literatura. Guarino (1998) define a seguinte classificação:

- Ontologias Genéricas: Também chamada ontologia de fundamentação ou ontologia de nível superior, é uma conceituação abstrata sobre elementos genéricos para fazer parte de um domínio específico, como, por exemplo, espaço, tempo, matéria, objeto, evento e ação.
- Ontologias de Domínio: Descrevem o vocabulário relacionado a conceituações de domínios particulares, como, por exemplo, medicina, direito, dentre outros.
- Ontologias de Tarefa: Similar a uma ontologia de domínio, porém, ao invés de mapear os conceitos de um domínio particular, mapeia conceitos de uma tarefa ou atividade específicas, como, por exemplo, diagnóstico, venda e matrícula.
- Ontologias de Aplicação: Descrevem conceitos que são dependentes de um domínio e de uma tarefa particular e, assim, combina especializações de conceitos presentes nas ontologias de domínio e de tarefa.

2.5.2. Linguagens de Representação

Para representar uma conceitualização compartilhada, é necessária uma linguagem de representação. Existem hoje muitas linguagens de representação definidas e a grande maioria é baseada na sintaxe do XML, que tende a tornar-se a linguagem padrão para modelar as ontologias. Algumas das linguagens são:

- RDF/RDF Schema: A linguagem Resource Description Framework (RDF) foi criada com o objetivo de representar o conhecimento utilizando redes semânticas. São linguagens que permitem a representação de conceitos, taxonomias de conceitos e relações binárias.

•OIL: A Ontology Interchange Language (OIL), combina primitivas de modelagem das linguagens baseadas em frames com a semântica formal e serviços de inferência da lógica descritiva. Pode verificar classificação e taxonomia de conceitos.

•OWL: A Ontology Web Language (OWL) é uma linguagem de marcação semântica para publicação e compartilhamento de ontologias na WEB. desenvolvida como uma extensão do vocabulário da RDF.

2.5.3. Ferramentas

Existem diversas ferramentas de apoio à construção de ontologias, como, por exemplo, a Protége 2000, a OntoEdit e a Text-to-onto. Em geral, as ferramentas utilizam linguagem de representação para a construção das ontologias.

2.5.4. Áreas aplicadas

A ontologia têm sido aplicada em diversas áreas e muitos projetos têm sido desenvolvidos afim de criar ontologias genéricas que possam ser utilizadas por diversos sistemas em todo o mundo. Todas as ontologias criadas tem sempre o mesmo foco, criar facilidades para aquisição, manutenção e acesso a informações vindas de diversas fontes, promovendo a sua reutilização. Algumas áreas de aplicação da ontologia conhecidas são: Processamento de linguagem natural; Web Semântica; Comércio eletrônico; Sistemas de Informações Geográficas (SIG); e Pesquisa e Desenvolvimento (P&D).

Ontologias são apontadas, hoje, como sendo cruciais na GC em melhoria dos processos. Muitos dos trabalhos que exploram o uso da GC e ontologia se baseiam no conceito de experiências passadas. Em um ambiente de desenvolvimento de *software*, por exemplo, o reuso das experiências e conhecimento coletados podem ajudar a evitar que falhas se repitam e auxilia na solução de problemas recorrentes do processo aplicado, promovendo sua melhoria. Estas experiências (informações) podem ser incorporadas em

alguma base de conhecimento para facilitar a sua gestão e seu acesso, por exemplo, um *Data Warehouse*, de forma a auxiliar na tomada de decisão dentro da organização.

3 ONTOLOGIA DE PROCESSO DE TESTE DE SOFTWARE

Esta é uma ontologia de domínio, que descreve o desenvolvimento de *software*, particularmente o contexto do teste de *software*.

Como a ontologia utilizada ainda está em desenvolvimento ela poderá e provavelmente sofrerá algumas modificações, entretanto mesmo em fase inicial, esta já é capaz de descrever a execução de um teste, e é partir dessa premissa que o DW será criado, para armazenar os dados provenientes principalmente desta etapa do teste.

Veja na Figura 2 a estrutura da ontologia que descreve o processo de teste de *software*.

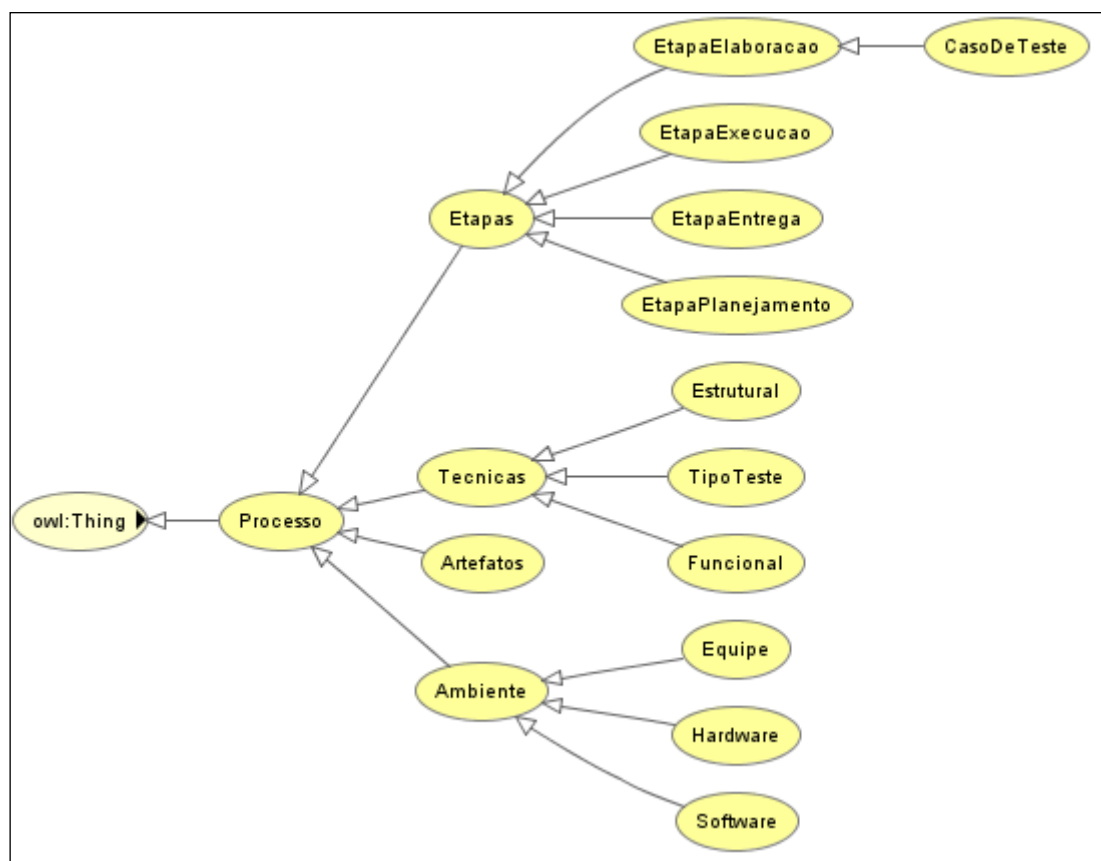


Figura 2 – Estrutura da ontologia de processo de teste de *software*.

Veja na Figura 3 com as propriedades *data type* da ontologia.

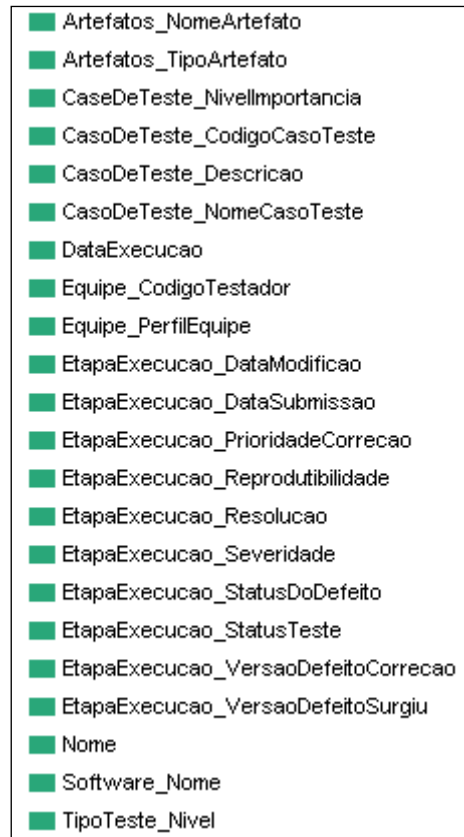


Figura 3 – Propriedades *data type* da ontologia.

Veja na Figura 4 as propriedades *object*

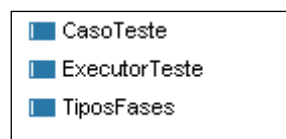


Figura 4 – Propriedades *object* da ontologia.

As instâncias criadas para ontologia foram extraídas de um projeto real desenvolvido no Instituto Tecnológico da Aeronáutica - Projeto FINEP 5206/06

- Projeto de Integração e Cooperação Amazônica para a Modernização do Monitoramento Hidrológico (ICA-MMH Parte B) (Cunha, 2010).

3.1.1. Ferramentas

A ontologia foi elaborada com o auxílio da ferramenta Protégé e o cadastro de instâncias também foi realizado com essa ferramenta.

Já para conversão dos dados da ontologias do arquivo rdf/xml em um formato de XML simples que será utilizado no processo de ETL (*Extract Transform and Load*) foi utilizada o *framework* Jena que possibilita a manipulação de dados contidos em arquivos nos formatos owl e rdf/xml.

3.1.2. Jena

A conversão dos dados foi realizada conforme mostra a Figura 5, com o pseudo código desenvolvido.

```
1  Abrir tag <ONTOLOGIA>
2      Para cada classe
3          Para cada instância da classe
4              Abrir tag <OBJETO>
5                  Gerar tag <URI> uri_do_recurso </URI>
6                  Gerar tag <CLASSE> nome_classe </CLASSE>
7                  Gerar tag <INSTANCIA> nome_instância </INSTANCIA>
8                  Para cada propriedade desta instância
9                      Se propriedade é um literal
10                         Gerar tag <nome_propriedade> valor_propriedade </nome_propriedade>
11                     Senão
12                         Gerar tag <nome_propriedade> uri_do_recurso </nome_propriedade>
13                     Fim Se
14                 Fim Para
15             Fechar tag </OBJETO>
16         Fim Para
17     Fim Para
18 Fechar tag </ONTOLOGIA>
19 Fim
```

Figura 5 – Pseudo código de conversão dos dados da ontologia para um formato simplificado de XML.

4 MODELO DO DATA WAREHOUSE

Para a criação do modelo multidimensional do DW foi escolhido o modelo Estrela, pois é implementado em banco de dados relacional, facilitando a implementação e também por utilizar linguagem SQL para a consulta dos dados.

O SGBD escolhido foi o MySQL Server 5.1 um banco de dados *opensource*, bastante utilizado no mercado.

Dentro do processo de testes optou-se pela criação de um modelo que pudesse armazenar os dados da etapa de execução de testes, contudo isso não impede a expansão do modelo abrangendo outras etapas dentro do processo, os dados utilizados para popular o DW, conforme citado anteriormente, são dados de um projeto real e foram obtidos de um projeto recentemente desenvolvido pelo Instituto Tecnológico de Aeronáutica (ITA).

Veja na Figura 6 o modelo criado para armazenar os dados da ontologia do processo de teste de *software*.

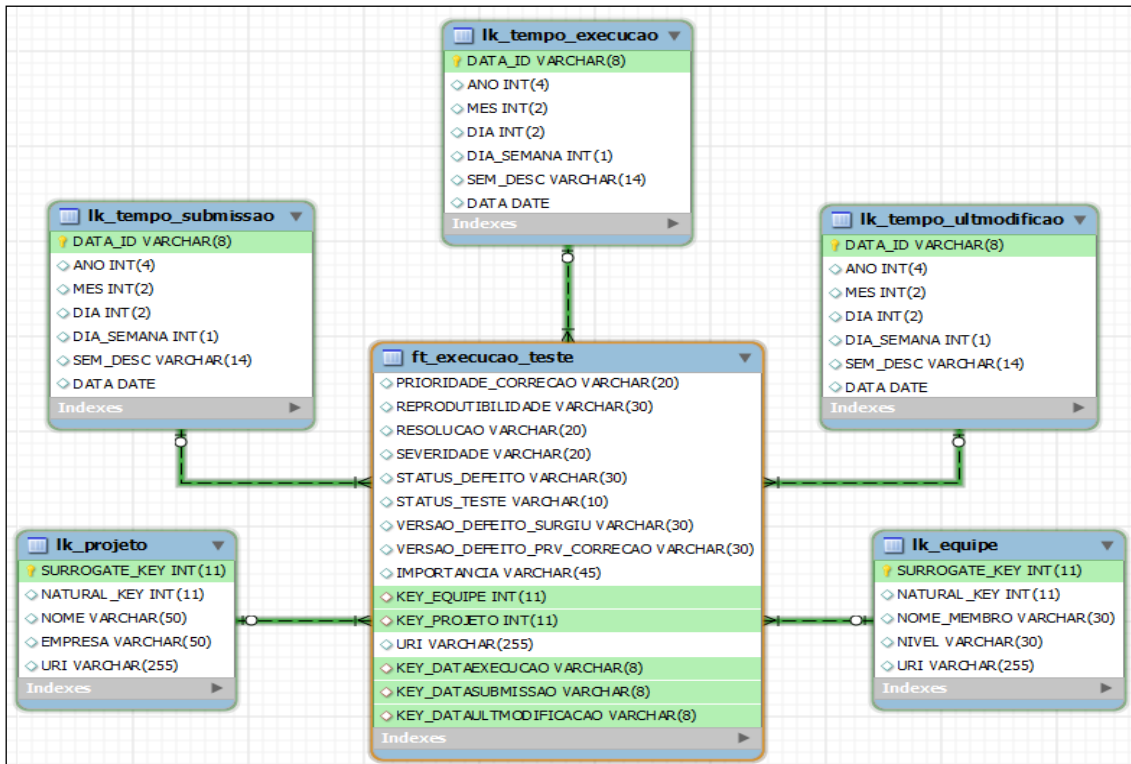


Figura 6 – Modelo estrela do *Data Warehouse* para o processo de teste de *software*.

Como visto na Figura 6 o modelo estrela criado é composto pela tabela *ft_execucao_teste* que é a tabela de fatos e por 5 dimensões, as dimensões são as seguintes:

- Três Dimensões de tempo, *lk_tempo_execucao* esta contém informação da data de execução de um teste, *lk_tempo_submissao* esta contém informação da data de submissão de um defeito, *lk_tempo_ultmodificao* esta contém informação da data que foi realizada a última modificação na solicitação de conserto de um bug.
- Uma dimensão *lk_equipe* esta contém informações da equipe de testes, como o nome do testador e seu nível dentro da equipe.
- Uma dimensão *lk_projeto* esta contém informações sobre os projetos cadastrados para os quais foram executados os testes.

4.1. Carga dos dados no *Data Warehouse* através do processo de Extract Transform and Load (ETL)

Após converter os dados da ontologia que antes estavam em formato owl em um formato simplificado de XML, estes dados serão processados pela ferramenta Pentaho Data Integration que irá extrair, transformar e finalmente carregá-los na base de dados do *Data Warehouse*.

A Figura 7 mostra a transformação criada para este processo.

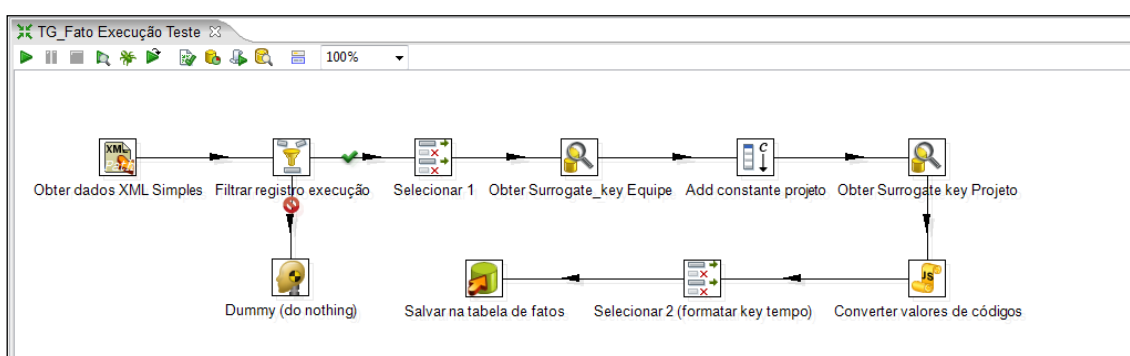


Figura 7 – Processo de ETL com a ferramenta Pentaho Data Integration.

O fluxo demonstrado na Figura 7 está descrito abaixo:

- 1º Obter os dados do arquivo XML simples.
- 2º Filtrar apenas linhas de execução de teste.
- 3º Selecionar somente os campos de execução de teste.
- 4º Obter a chave substituta no DW do testador que executou o teste.
- 5º Adicionar a constante de projeto, esta etapa foi necessária porque só existe um projeto.
- 6º Obter a chave substituta no DW do projeto que a execução de teste pertence.

- 7º Conversão de alguns valores que estão em formato de código, substituindo por seu valor real.
- 8º Selecionar alguns campos e formatar os campos de data para criar as chaves estrangeiras que criaram o relacionamento com as dimensões de tempo.
- 9º Exportar os dados para a tabela de fatos no DW.

4.2. Visão geral da solução

Como visto nos capítulos anteriores as etapas seguidas no desenvolvimento da solução, abordadas nos capítulos anteriores podem ser exemplificada com a Figura 8.

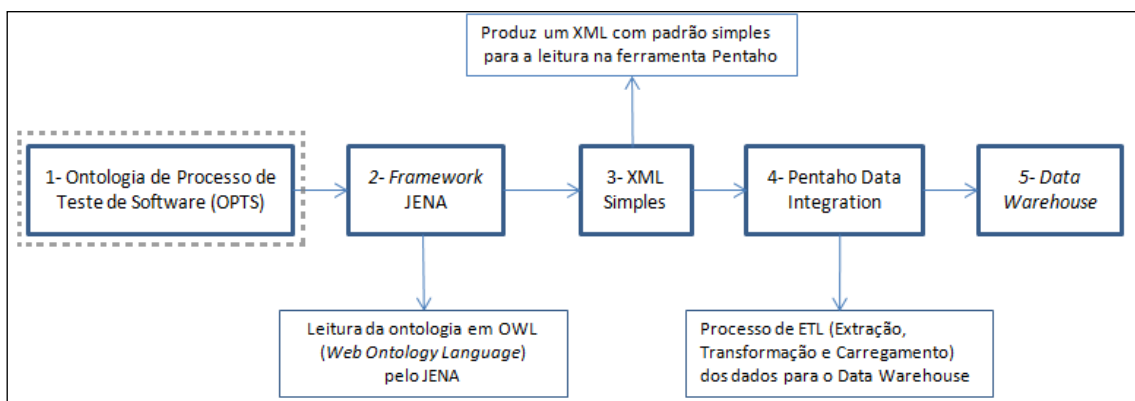


Figura 8 - Estrutura proposta para a solução.

5 VALIDAÇÃO DA ESTRUTURA PROPOSTA

Depois de realizado o processo completo da solução proposta, foram realizados algumas consultas na base de dados do *Data Warehouse* procurando exercitar a estrutura criada simulando seu uso em um ambiente real.

5.1. Perguntas

Como o objetivo de um *Data Warehouse* é procurar conhecimento útil na base de dados, para exercitar a estrutura foram realizadas consultas que pudessem responder algumas perguntas.

As *queries* SQL realizadas encontram-se no apêndice A.

5.1.1. Qual a quantidade de execuções dos testes por perfil da equipe?

O resultado obtido está em forma de gráfico na Figura 9.

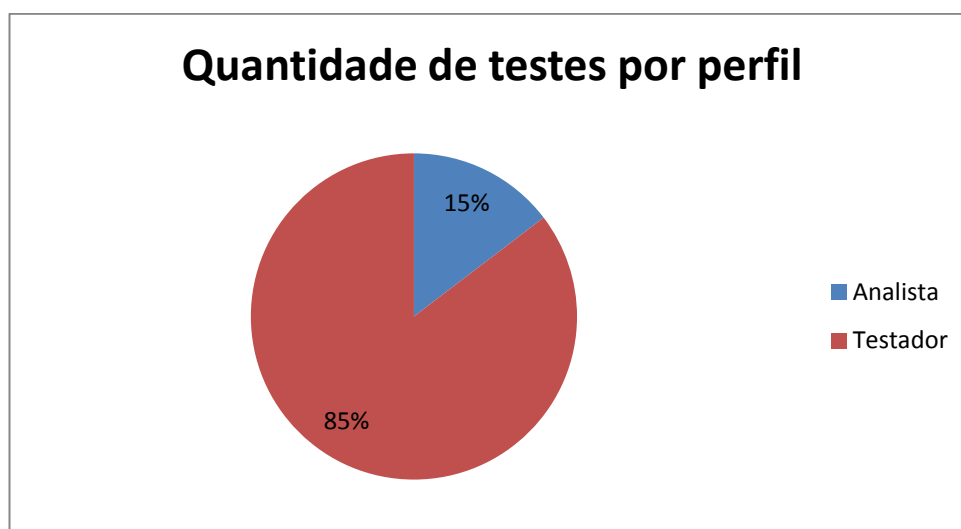


Figura 9 - Gráfico de quantidade de testes por perfil.

5.1.2. Qual a quantidade de testes por severidade do defeito?

O resultado obtido está na forma de gráfico na Figura 10.

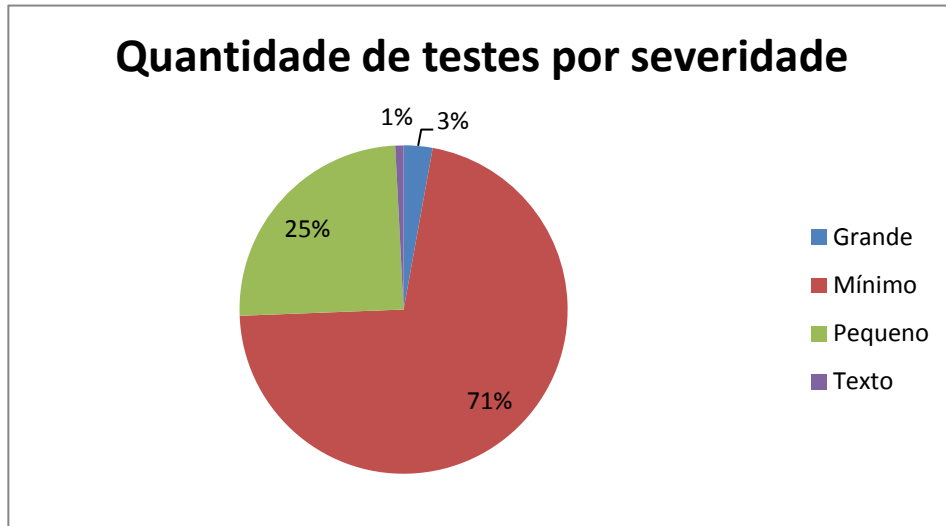


Figura 10 - Gráfico de quantidade de testes por severidade do defeito.

5.1.3. Qual a quantidade de testes por resolução do defeito?

O resultado obtido está na forma de gráfico na Figura 11.

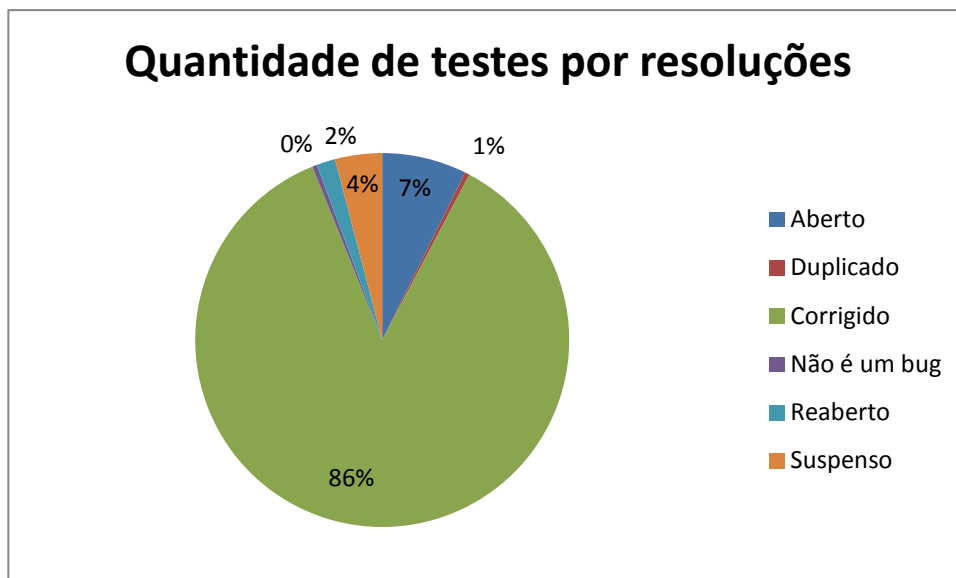


Figura 11 - Gráfico de quantidade de testes por resolução.

5.1.4. Qual a quantidade de testes por importância do caso de teste?

O resultado obtido está na forma de gráfico na Figura 12.

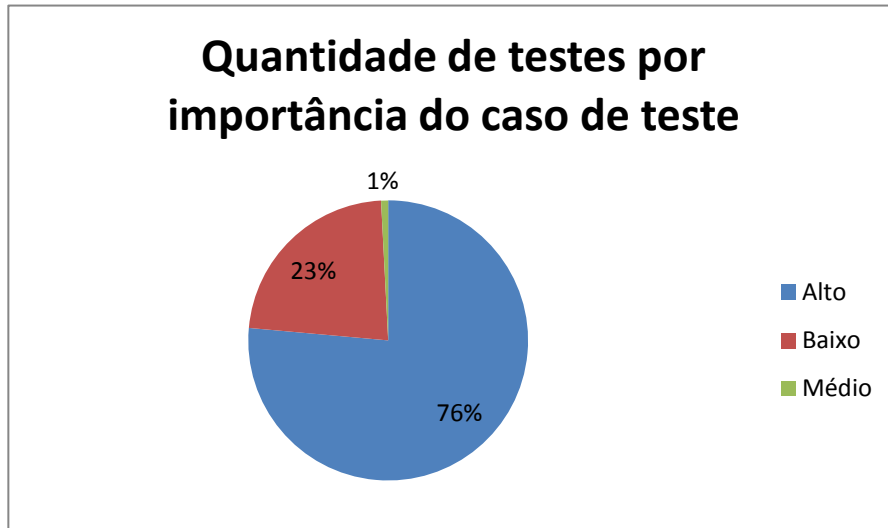


Figura 12 - Gráfico de quantidade de testes por importância do caso de teste.

6 CONCLUSÃO

O objetivo deste trabalho foi elaborar um projeto que conseguisse obter dados de uma ontologia e armazenar estes dados em um *Data Warehouse*, os dados obtidos correspondem ao processo de teste de *software*, o modelo foi criado em uma base MySQL. A partir dessa premissa as contribuições deste trabalho são:

- Criação de um *Data Warehouse* que armazena dados de execuções de casos de testes.
- Criação de um modelo expansível que pode abranger outras etapas dentro do processo de teste de *software*.
- Um modelo que facilita as consultas em uma base com grande volume de dados.

Analisando os resultados do capítulo 6, a solução do DW mesmo que específica para a etapa de execução dos testes, pode fornecer muitos detalhes para o gestor que poderão ser utilizados na melhoria do processo tanto de desenvolvimento quanto no próprio processo de teste de *software*, contudo o modelo pode ser enriquecido e fornecer ainda mais dados que possam ser utilizados na tomada de decisão do gerente ou responsável pelo projeto.

Sendo este trabalho apenas um esforço inicial a solução desenvolvida ainda pode ser melhorada. Alguns pontos que podem ser melhorados e abordados em trabalhos futuros incluem:

- Automatizar entrada de dados na ontologia obtendo dados de outras fontes.
- Expandir o modelo para armazenar dados de outras etapas do processo de teste.
- Integrar os dados de desenvolvimento com os de teste.

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMSON, C.; **Start Schema - The Complete Reference**. 1st ed. McGraw-Hill, 2010 ISBN: 978-0-07-174432-4.

BASTOS, A.; RIOS, E.; CRISTALLI, R. & MOREIRA, T. **Base de conhecimento em teste de software**. São Paulo, Martins Fontes, 2007 ISBN: 8599102893.

BOUMAN, R.; DOUGEN, J. van; **Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL**. 1ª Ed. Indianapolis: Wiley Publishing, 2010 ISBN: 978-0-470-48432-6.

CUNHA, A. M. Relatório Técnico do 5o Semestre do Projeto FINEP 5206/06. São José dos Campos, 2010.

GUARINO, N. *Formal Ontology and Information Systems*. In: **Proceedings of International Conference in Formal Ontology and Information Systems (FOIS'98)**, pp. 3-15, 1998.

GRUBER, T. R. **Toward principles for the design of ontologies used for knowledge sharing**. In *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Padova, Italy, 1993.

JENSEN, C. S.; PEDERSEN, T. B.; THOMSEN, C.; **Multidimensional Databases and Data Warehousing**. Morgan & Claypool, 2010 ISBN: 978-1608455379.

O'LEARY, D.; STUDER, R. **Knowledge management: An interdisciplinary approach**. IEEE Intelligent Systems, v. 16, No. 1, 2001.

O'LEARY, D. E. **Enterprise knowledge management**. IEEE Computer Magazine, p. 54-61, 1998.

APÊNDICE A - COMANDOS SQL DE CONSULTA AO DATA WAREHOUSE

A.1 Comando SQL quantidade de testes executados por perfil

```
SELECT eq.nivel as 'Cargo', count(*) as 'Quantidade de execução do teste'
FROM FT_EXECUCAO_TESTE ft, LK_EQUIPE eq
WHERE ft.key_equipe = eq.surrogate_key
GROUP BY (eq.nivel)
```

A.2 Comando SQL quantidade de testes por severidade

```
SELECT ft.severidade, count(*) as 'Quantidade de execução do teste'
FROM FT_EXECUCAO_TESTE ft, LK_TEMPO_EXECUCAO dim_exec,
LK_TEMPO_SUBMISSAO dim_sub
WHERE ft.key_dataexecucao = dim_exec.data_id AND ft.key_datasubmitissao
= dim_sub.data_id
GROUP BY (ft.severidade)
```

A.3 Comando SQL quantidade de testes por resolução

```
SELECT ft.resolucao, count(*) as 'Número relatado de defeitos'
FROM FT_EXECUCAO_TESTE ft
GROUP BY (ft.resolucao)
```

A.4 Comando SQL quantidade de testes por importância do caso de teste

```
SELECT ft.importancia, count(*) as 'Quantidade de testes executados'
FROM FT_EXECUCAO_TESTE ft
GROUP BY (ft.importancia)
```