



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

**Estudo da Eficácia e Eficiência do Uso de um
Ambiente de *Data Warehousing* para Aferição
da Qualidade Interna de Software: um Estudo
de Caso no TCU**

Autor: Pedro da Cunha Tomioka
Orientador: Prof. Msc. Hilmer Rodrigues Neri

Brasília, DF
2014



Pedro da Cunha Tomioka

**Estudo da Eficácia e Eficiência do Uso de um Ambiente
de *Data Warehousing* para Aferição da Qualidade Interna
de Software: um Estudo de Caso no TCU**

Monografia submetida ao curso de graduação
em Engenharia de Software da Universidade
de Brasília, como requisito parcial para ob-
tenção do Título de Bacharel em Engenharia
de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Msc. Hilmer Rodrigues Neri

Brasília, DF

2014

Pedro da Cunha Tomioka

Estudo da Eficácia e Eficiência do Uso de um Ambiente de *Data Warehousing* para Aferição da Qualidade Interna de Software: um Estudo de Caso no TCU/
Pedro da Cunha Tomioka. – Brasília, DF, 2014-

37 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Msc. Hilmer Rodrigues Neri

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2014.

1. Métricas de Código-Fonte. 2. *Data Warehousing*. I. Prof. Msc. Hilmer Rodrigues Neri. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Estudo da Eficácia e Eficiência do Uso de um Ambiente de *Data Warehousing* para Aferição da Qualidade Interna de Software: um Estudo de Caso no TCU

CDU 02:141:005.6

Pedro da Cunha Tomioka

Estudo da Eficácia e Eficiência do Uso de um Ambiente de *Data Warehousing* para Aferição da Qualidade Interna de Software: um Estudo de Caso no TCU

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, Ainda não se sabe:

Prof. Msc. Hilmer Rodrigues Neri
Orientador

Ainda não se sabe
Convidado 1

Ainda não se sabe
Convidado 2

Brasília, DF
2014

Este trabalho é dedicado à minha mãe, Elizabeth, e à minha irmã, Luisa.

Agradecimentos

Matheus Tristão, Nilton César Araruna, Guilherme Baufaker

Agradeço a meus amigos e companheiros de graduação: Pedro Henrique Potiguara, Aline Gonçalves, Hichemm Khalyd, Tiago Pereira, Thabata Granja, Guilherme Fay, Guilherme de Lima, Pedro Matias, Mayco Henrique, Carlos Filipe, Marcos Ronaldo, Rafael Ferreira, Fágner Rodrigues, Bruno Motta, Paulo Acés, Matheus Patrocínio, <equipe de mds>, Greg Ouyama, Eusyar Alves, João Pedro Carvalho, Artur Potiguara , Pedro Guilherme Moreira, Pedro Inazawa.

"The Truth is Out There"
(The X-Files)

Resumo

Palavras-chaves: Métricas de Código-Fonte. *Data Warehousing*. *Data Warehouse*

Abstract

Palavras-chaves: *Source Code Metrics. Data Warehousing. Data Warehouse*

Lista de ilustrações

Figura 1 – Elementos básicos de um data warehouse	19
Figura 2 – Uma matriz bidimensional	21
Figura 3 – Um cubo de dados tridimensional	21
Figura 4 – Um esquema estrela	23
Figura 5 – Um esquema floco de neve	23
Figura 6 – Projeto Físico do <i>Data Warehouse</i> por Rêgo (2014)	26
Figura 7 – Metadados do Data Warehouse por Rêgo (2014)	27
Figura 8 – Estrutura do Estudo de Caso	30

Lista de tabelas

Tabela 1 – Fatos e Dimensões do <i>Projeto de Data Warehouse</i> por Rêgo (2014) . . .	25
--	----

Lista de abreviaturas e siglas

ACC	<i>Afferent Connections per Class</i>
ACCM	<i>Average Cyclomatic Complexity per Method</i>
AMLOC	<i>Average Method Lines of Code</i>
ANPM	<i>Average Number of Parameters per Method</i>
CBO	<i>Coupling Between Objects</i>
CSV	<i>Comma-Separated Values</i>
DER	Diagrama Entidade Relacionamento
DIT	<i>Depth of Inheritance Tree</i>
DW	<i>Data Warehouse</i>
ETL	<i>Extraction-Transformation-Load</i>
FTP	<i>File Transfer Protocol</i>
GQM	<i>Goal-Question-Metric</i>
IEC	<i>International Electrotechnical Commission</i>
ISO	<i>International Organization for Standardization</i>
JSON	<i>JavaScript Object Notation</i>
LCOM4	<i>Lack of Cohesion in Methods</i>
LOC	<i>Lines of Code</i>
NPA	<i>Number of Public Attributes</i>
NOC	<i>Number of Children</i>
NOM	<i>Number of Methods</i>
OLAP	<i>On-Line Analytical Processing</i>
OLTP	<i>Online Transaction Processing</i>
RFC	<i>Response For a Class</i>

SCAM	<i>IEEE International Working Conference on Source Code Analysis and Manipulation</i>
SGBD	Sistema de Gerenciamento de Bancos de Dados
XML	<i>Extensible Markup Language</i>
YAML	<i>YAML Ain't Markup Language</i>
OSS	Sistemas de Fonte de Dados Operacionais (<i>Operational Source Systems</i>)
DSA	Área de Preparação dos Dados (<i>Data Staging Area</i>)
DPA	Área de Apresentação dos dados (<i>Data Presentation Area</i>)
DAT	Ferramentas de Acesso de dados (<i>Data Access Tools</i>)

Sumário

1	INTRODUÇÃO	15
1.1	Contexto	15
1.2	Problema	15
1.3	Questão de Pesquisa	15
1.4	Objetivos	15
1.5	Hipótese	15
1.6	Organização do Trabalho	15
2	MÉTRICAS DE SOFTWARE	16
3	DATA WAREHOUSING	17
3.1	Definições e terminologia	17
3.2	Características dos data warehouses	18
3.2.1	Componentes Básicos de um Data Warehouse	19
3.3	Modelagem de dados para data warehouses	20
3.3.1	Tipos de tabelas do modelo multidimensional	22
3.3.2	Esquemas multidimensionais	22
3.4	Solução de <i>Data Warehousing</i> para Métricas de Código-Fonte	23
3.4.1	Seleção do processo de negócio	24
3.4.2	Verificação da periodicidade de coleta de dados	25
3.4.3	Identificação dos fatos e das dimensões	25
3.5	Considerações finais do Capítulo	27
4	PROJETO DE ESTUDO DE CASO	28
4.1	Definição	28
4.2	Background	29
4.2.1	Outros Trabalhos	29
4.2.2	Problema	29
4.2.3	Questão de Pesquisa	29
4.3	Design	32
4.4	Seleção	32
4.5	Fonte e Método de Coleta de Dados	33
4.6	Análise	33
4.7	Validade	33
4.8	Considerações finais do Capítulo	34

5	CONCLUSÃO	35
	Referências	36

1 Introdução

1.1 Contexto

1.2 Problema

1.3 Questão de Pesquisa

1.4 Objetivos

- Aferir e avaliar a eficácia e eficiência da solução de data warehousing proposta por [Rêgo \(2014\)](#) quanto a qualidade interna de um produto de software;
- Definir o protocolo de estudo de caso;

1.5 Hipótese

1.6 Organização do Trabalho

- Capítulo [1](#) Introdução
- Capítulo [2](#) Métricas
- Capítulo [3](#) Data Warehousing
- Capítulo [4](#) Projeto de Estudo de Caso
- Capítulo [5](#) Conclusão

2 Métricas de Software

3 *Data Warehousing*

Este capítulo apresenta algumas definições necessárias para que se obtenha uma visão geral de data warehousing e OLAP. Primeiramente é definido o que é um data warehouse e quais são as terminologias relacionadas ao assunto. Em seguida é mostrado as características específicas de um data warehouse, procurando compara-lo aos banco de dados relacionais. Os componentes de um dw são definidos e apresentados e subsequente-mente a modelagem que se aplica a um dw é explicitada. Ao final é apresentado a solução de data warehousing para métricas de código-fonte e como ela foi pensada segundo os fundamentos explicados nos tópicos anteriores.

3.1 Definições e terminologia

[Inmon \(1992\)](#) define um data warehouse como uma coleção de dados orientada a assunto, integrada, não volátil, variável no tempo para o suporte às decisões da gerência.

[Hobbs et al. \(2011\)](#) afirma que data warehouse é um banco de dados que contém dados de múltiplos sistemas que foram consolidados, integrados, agregados e estruturados de modo que possam ser usados para apoiar o processo de análise e tomada de decisão de um negócio.

Os Data Warehouses oferecem armazenamento, funcionalidade e responsividade às consultas além das capacidades dos bancos de dados orientados à transação ([ELMASRI; NAVATHE, 2011](#)). São vários os tipos de aplicação aceitos para um data warehouse. Algumas delas serão definidas a seguir.

OLAP (online analytical processing — processamento analítico on-line)
É um conjunto de princípios que fornecem uma estrutura dimensional de apoio à decisão ([KIMBALL; ROSS, 2002](#)). As ferramentas OLAP empregam as capacidades de computação distribuída para análises que requerem mais armazenamento e poder de processamento do que pode estar localizado econômica e eficientemente em um desktop individual ([ELMASRI; NAVATHE, 2011](#)). Nestas aplicações dados sumariados e históricos são mais importantes que dados atômicos ([NERI, 2002](#)).

DSS (decision-support systems — sistemas de apoio à decisão) São sistemas que ajudam os principais tomadores de decisões de uma organização com dados de nível mais alto em decisões complexas e importantes ([ELMASRI; NAVATHE, 2011](#)). Segundo [Kimball \(2008\)](#) os DSSs têm como objetivo tornar a informação de uma organização acessível e consistente, prover uma fonte adaptável e resiliente de informações, e garantir a segurança aos dados para assim ser um sistema base para a tomada de decisão.

OLTP (online transaction processing) — processamento on-line de transações Bancos de dados tradicionais têm suporte para o OLTP, com suas operações de inserção, atualização e exclusão e também à consulta de dados. [Sharma \(2011\)](#) explica que sistemas OLTP usam tabelas simples para armazenar dados, estes que são normalizados, para se reduzir a redundância ou até mesmo eliminá-los, buscando sempre a garantia de sua consistência.

3.2 Características dos data warehouses

[Elmasri e Navathe \(2011\)](#) diferencia um Data warehouse de uma estratégia de multibancos de dados afirmando que o primeiro possui um armazenanento em um modelo multidimensional, assim dados de múltiplas fontes são integrados e processados para tal e sendo assim contrário ao segundo, que oferece acesso a bancos de dados disjuntos e normalmente heterogêneos. A vantagem aqui se nota pela adoção de um padrão de design mais conciso, que pode levar a uma menor complexidade de implantação.

Diferentemente da maioria dos banco de dados transacionais, data warehouses costumam apoiar a análise de série temporal e tendência, ambas exigindo mais dados históricos do que geralmente é mantido nos bancos de dados transacionais ([ELMASRI; NAVATHE, 2011](#)).

Em comparação com os bancos de dados transacionais, os data warehouses são não-voláteis ([ELMASRI; NAVATHE, 2011](#)). Sendo assim, suas informações possuem a caracterísca de mudarem de uma forma muito menos frequente, portanto dificilmente seria enquadrada como uma informação em tempo real, e sim como uma de atualização periódica.

Segundo [Elmasri e Navathe \(2011\)](#) data warehouses são otimizados para recuperação de dados, e não para processamento de rotina, uma característica intrínica à sistemas OLTP.

[Elmasri e Navathe \(2011\)](#) ordena as seguintes características diferenciadoras de data warehouses, proveniente da lista original de [Codd, Codd e Salley \(1993\)](#) sobre OLAP:

- Visão conceitual multidimensional.
- Dimensionalidade genérica.
- Dimensões e níveis de agregação ilimitados.
- Operações irrestritas entre dimensões.
- Tratamento dinâmico de matriz esparsa.
- Arquitetura de cliente-servidor.

- Suporte para múltiplos usuários.
- Acessibilidade.
- Transparência.
- Manipulação de dados intuitiva.
- Desempenho de relatório consistente.
- Recurso de relatório flexível.

3.2.1 Componentes Básicos de um Data Warehouse

A figura 1 representa a estrutura básica de um DW segundo Kimball e Ross (2002), os componentes serão definidos a seguir.

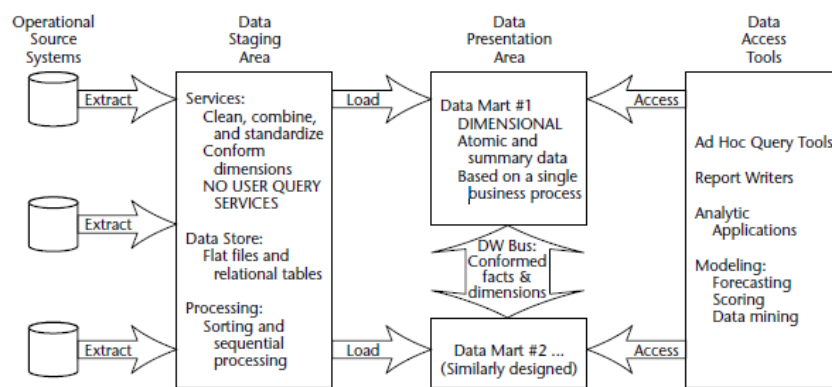


Figura 1 – Elementos básicos de um data warehouse

Sistemas de Fonte de Dados Operacionais (OSS - *Operational Source Systems*): Representa os sistemas que irão prover informações para o DW. Seus dados podem ser provenientes de outros sistemas OLTPs, planilhas e etc, que compoem o negócio a ser tratado. Devem ser pensados como fora do data warehouse porque, temos pouco ou nenhum controle sobre o conteúdo e formato dos dados presentes nestes tipos de sistemas legados (KIMBALL; ROSS, 2002).

Área de Preparação dos Dados (DSA - *Data Staging Area*): É onde irá ocorrer a possível limpeza e reformatação dos dados antes que sejam carregados no *data warehouse* (ELMASRI; NAVATHE, 2011), processo esse conhecido como extração transformação e carga (ETL).

O passo de extração consiste em ler os dados das fontes operacionais e copiar o que for necessário para o DW na área de preparação dos dados.

Transformação é o passo que irá tratar os dados agora presente na área de preparação dos dados. Segundo Kimball e Ross (2002) existem diversos tipos de transformações

que podem ocorrer, como a limpeza dos dados (correção de erros ortográficos, resolução de dados conflitantes com o domínio, tratamento de dados que estão em falta, ou o *parse* de dados para um certo padrão), e também ações como combinar dados de várias fontes, e por fim pode-se atribuir chaves para os dados que irão para o DW.

O terceiro passo do processo de ETL, a carga, é onde os dados que foram tratados, limpo, transformados e padronizados dentro da área de preparação dos dados serão carregados nas tabelas do data warehouse.

Área de Apresentação dos dados (DPA - *Data Presentation Area*): Representa a área onde os dados são organizados, armazenados e disponibilizados para consulta direta pelos usuários, autores de relatórios, e outras aplicações. Os dados da área de apresentação devem ser dimensionais e atômicos e devem estar de acordo com a arquitetura do DW ([KIMBALL; ROSS, 2002](#)).

Ferramentas de Acesso de dados (DAT - *Data Access Tools*):

Metadados: Definido como toda a informação no ambiente de data warehouse que não são os dados em si ([KIMBALL; ROSS, 2002](#)). Os metadados num ambiente DW podem apontar para dados sobre tabelas do sistema, índices, relacionamentos, e etc. [Kimball \(1998\)](#) recomenda que a arquitetura de um DW seja orientada a metadados, devido a seu papel crítico de prover informações e parâmetros que permitem que as aplicações executarem suas tarefas com um controle maior sobre os dados provenientes das fontes de dados e outros elementos fundamentais para sua execução.

3.3 Modelagem de dados para data warehouses

Modelos multidimensionais tiram proveito dos relacionamentos inerentes nos dados para preencher os dados em matrizes multidimensionais, chamadas cubos de dados ([ELMASRI; NAVATHE, 2011](#)).

Uma planilha simples contendo uma matriz bidimensional poderia representar as vendas regionais por produto para um determinado período (Figura 2). Poderia então ser acrescentado a dimensão de tempo para a relação, assim teríamos o trimestre fiscal para os produtos da região. Isto produziria uma matriz tridimensional, representada por um cubo (Figura 3).

		Região			
		Reg 1	Reg 2	Reg 3	...
Produto	Prod 1				
	Prod 2				
	Prod 3				
	...				

Figura 2 – Uma matriz bidimensional

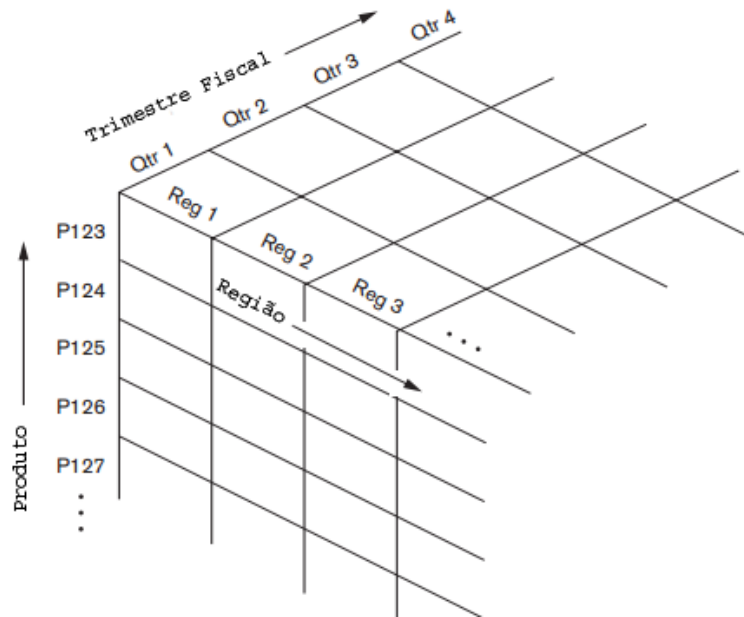


Figura 3 – Um cubo de dados tridimensional

O cubo da Figura 3 organiza os dados de produto por trimestres fiscais e por região. Cada célula representa a venda de um certo produto, num certo trimestre fiscal em uma certa região. Assim a modelagem permite adicionar mais dimensões, criando-se um hipercubo que combinaria todas as outras anteriores, o único problema é que não seria facilmente visualizada a sua forma gráfica. Porém diversas ferramentas de acesso de dados são capazes de apresentar tais dimensões de acordo com a escolha de combinação do usuário.

As operações possíveis de serem aplicadas a um cubo OLAP são:

- **Pivoting:** Também conhecido como *rotation* (rotação), nessa técnica, o cubo de dados pode ser imaginado girando para mostrar uma orientação diferente dos eixos (ELMASRI; NAVATHE, 2011).
- **Roll-up:** *Drilling* em modelagem multidimensional significa ir de um nível hierárquico a outro (BALLARD, 2006). Portanto *roll-up* ou *drill-up* consiste em subir na hierarquia, agrupando em unidades maiores ao longo de uma dimensão (ELMASRI; NAVATHE, 2011).

- **Drill-Down:** É o inverso de *roll-up*, busca alcançar a informação em um nível maior de detalhamento quanto a sua dimensão.
- **Slice:** Termo usado para definir um membro ou um grupo de membros que será separado, *Slice*, (de todas as outras dimensões) e em seguida, avaliado através de todas as outras dimensões (BALLARD, 2006).
- **Dice:** Similar ao *slice*, Ballard (2006) caracteriza que esta operação busca combinar membros de diferentes dimensões a partir da escolha de um ou mais membros de uma mesma dimensão, para que assim a interrelação entre elas possa ser analisada.

3.3.1 Tipos de tabelas do modelo multidimensional

O modelo multidimensional apresenta dois tipos de tabelas: a tabela de dimensão e fato. Uma tabela de dimensão contém os descritores textuais do negócio (KIMBALL; ROSS, 2002), ou seja, consiste em tuplas de atributos da dimensão (ELMASRI; NAVATHE, 2011). Uma tabela de fato possui tuplas para cada fato registrado. Um fato é uma medida de desempenho do negócio, tipicamente numérico e aditivo (KIMBALL; ROSS, 2002). Assim a tabela de fatos contém também as dimensões que são indicadas pelos fatos, isto é representado por ponteiros para tabelas de dimensão (um para cada variável de fato observado).

3.3.2 Esquemas multidimensionais

Dois esquemas comuns são o esquema estrela e o esquema floco de neve. O esquema estrela consiste em uma tabela de fatos com uma única tabela para cada dimensão (Figura 4) (ELMASRI; NAVATHE, 2011). O esquema floco de neve é resultado da normalização e expansão das tabelas de dimensão do esquema estrela (BALLARD, 2006) (Figura 5). Segundo Kimball e Ross (2002) o floco de neve não é recomendado em uma ambiente de data warehouse, pois quase sempre faz com que a apresentação dos dados ao usuário seja mais complexa, além do impacto negativo que causa sobre o desempenho da navegação.

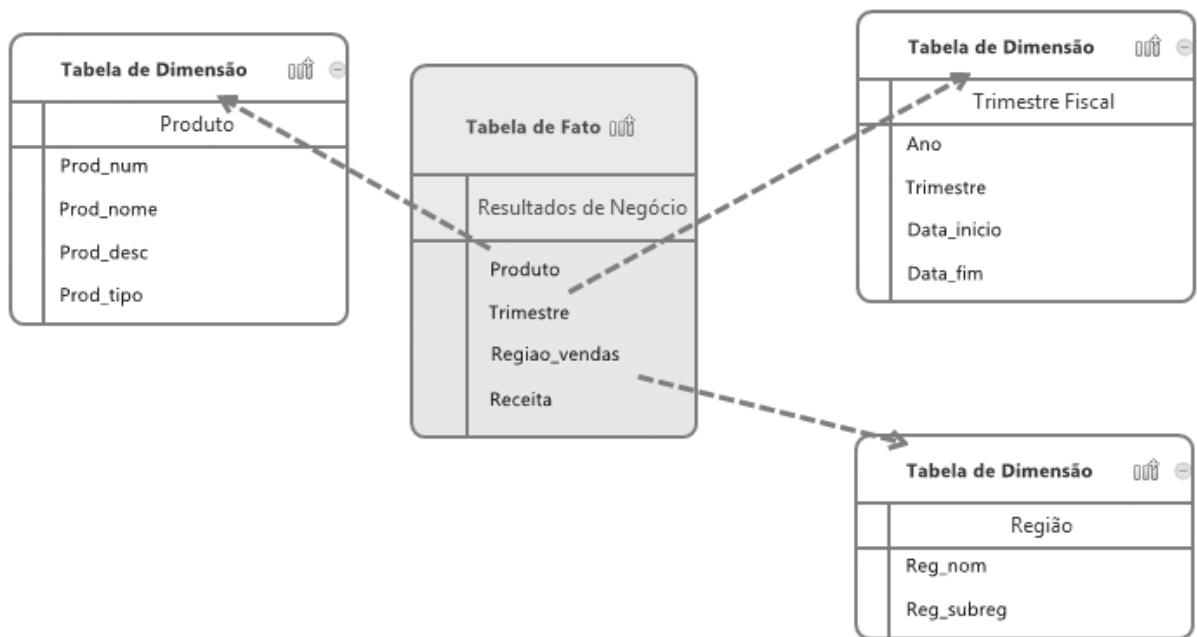


Figura 4 – Um esquema estrela

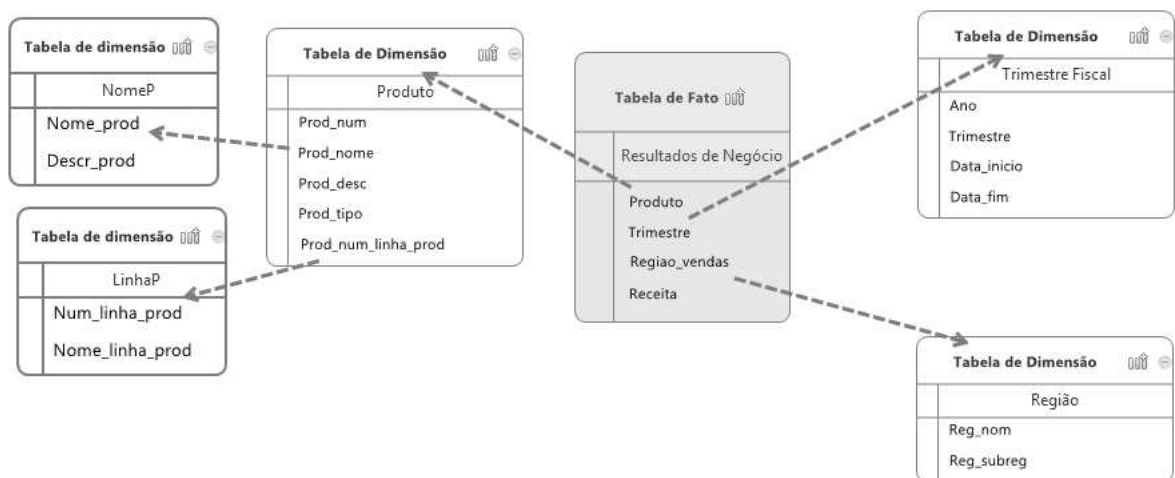


Figura 5 – Um esquema floco de neve

3.4 Solução de *Data Warehousing* para Métricas de Código-Fonte

Seguindo a metodologia de [Kimball e Ross \(2002\)](#) para projetar um data warehouse, [Rêgo \(2014\)](#) projetou uma solução de data warehousing para métricas de código-fonte seguindo quatro passos da metodologia, definidos em cada sub-seção a seguir.

3.4.1 Seleção do processo de negócio

[Rêgo \(2014\)](#) buscou indentificar os processos de negócio e seus respectivos requisitos. Assim foram levantados 15 requisitos de negócio, sendo que os 8 primeiros se referem a avaliação dos valores percentis das métricas de código-fonte e o restante a avaliação de cenários de limpeza e taxa de aproveitamento de oportunidades de melhoria de código-fonte. Os requisitos levantados por [Rêgo \(2014\)](#) foram os seguintes:

RN1: Visualizar o intervalo qualitativo obtido para cada métrica de código-fonte em uma determinada release do projeto para a configuração *Open JDK8 Metrics*.

RN2: Comparar o intervalo qualitativo obtido para cada métrica de código-fonte ao longo de todas as releases de um projeto para a configuração *Open JDK8 Metrics*.

RN3: Visualizar o o valor pecentil obtida para cada métrica de código-fonte em uma determinada release do projeto para a configuração *Open JDK8 Metrics*.

RN4: Comparar o o valor pecentil a para cada métrica de código-fonte ao longo de todas as releases para a configuração *Open JDK8 Metrics*.

RN5: Visualizar o intervalo qualitativo obtido para cada métrica de código-fonte em uma determinada release do projeto para a configuração *Tomcat Metrics*.

RN6: Comparar o intervalo qualitativo obtido para cada métrica de código-fonte ao longo de todas as releases de um projeto para a configuração *Tomcat Metrics*.

RN7: Visualizar a medida obtida para cada métrica de código-fonte em uma determinada release do projeto para a configuração *Tomcat Metrics*.

RN8: Comparar o valor percentil obtido para cada métrica de código-fonte ao longo de todas as releases para a configuração *Tomcat Metrics*.

RN9: Visualizar a quantidade de cenários de limpeza identificados por tipo de cenários de limpeza de código-fonte em cada classe ao longo de cada *release* de um projeto.

RN10: Comparar a quantidade de cenários de limpeza por tipo de cenários de limpeza de código-fonte em uma release de um projeto.

RN11: Visualizar o total de cenários de limpeza em uma determinada release de um projeto.

RN12: Visualizar cada uma das classes com um determinado cenário de limpeza de código-fonte ao longo das *releases* do projeto.

RN13: Visualizar as 10 classes de um projeto com menor número de cenários de limpeza identificados.

RN14: Visualizar as 10 classes de um projeto com maior número de cenários de limpeza identificados.

RN15: Acompanhar a Taxa de Aproveitamento de Oportunidades de Melhoria de Código-Fonte que é a divisão do total de cenários de limpeza identificados em uma release e o o número total de classes da mesmarelease de um projeto.

3.4.2 Verificação da periodicidade de coleta de dados

A identificação da periodicidade de coleta dos dados é essencial para que esta seja realizada de maneira correta, além de viabilizar a agregação dos dados em níveis ou hierarquias (RêGO, 2014). Assim a peridicidade foi identificada como as *releases* do software.

3.4.3 Identificação dos fatos e das dimensões

Rêgo (2014) identificou os fatos e dimensões a partir dos termos mais relevantes dos requisitos de negócios levantados. O relacionamento entre fato, natureza do fato e dimensão se mostra na Tabela 1.

Fato	Natureza do Fato	Dimensões
Valor Percentil	Não Aditivo	<ul style="list-style-type: none"> • Projeto • Métrica • Configuração • Qualidade • <i>Release</i> • Tempo
Quantidade de Cenários de Limpeza	Aditivo	<ul style="list-style-type: none"> • Projeto • Cenário de Limpeza • Classe • <i>Release</i>
Taxa de Aproveitamento de Oportunidade de Melhoria de Código-Fonte	Não Aditivo	<ul style="list-style-type: none"> • Projeto • <i>Release</i>

Tabela 1 – Fatos e Dimensões do *Projeto de Data Warehouse* por Rêgo (2014)

Assim Rêgo (2014) elaborou o modelo físico para o data warehouse:

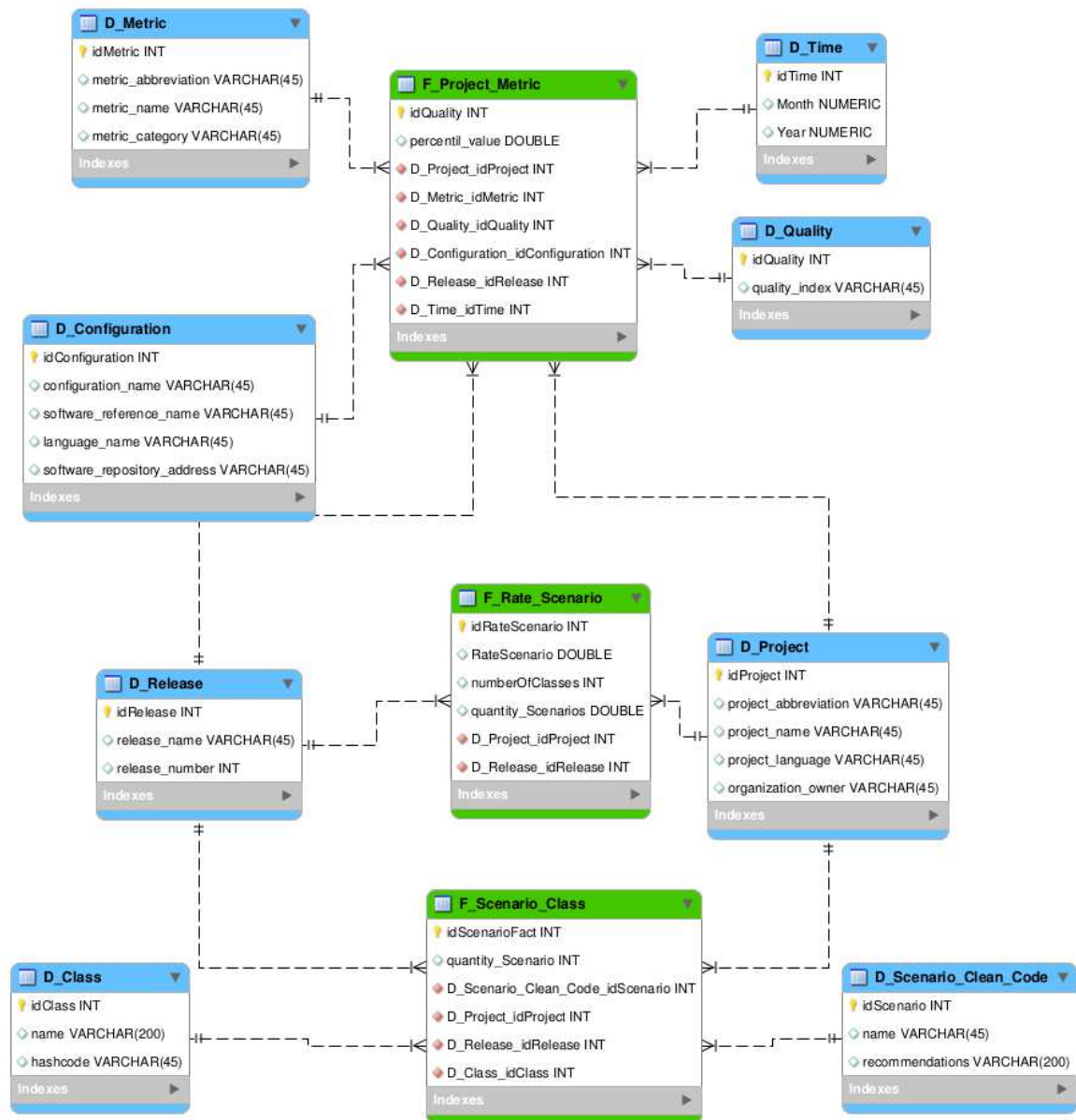
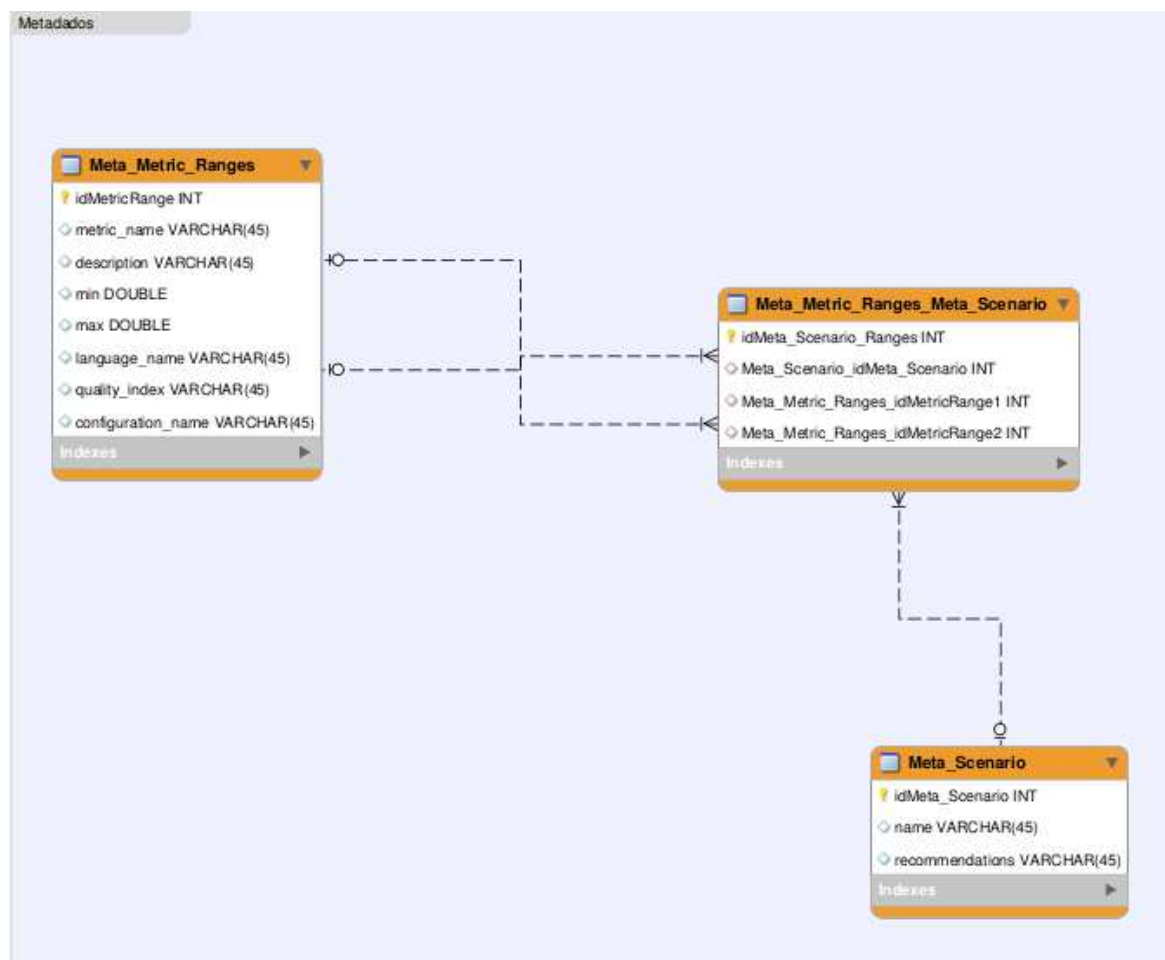


Figura 6 – Projeto Físico do *Data Warehouse* por Rêgo (2014)

Rêgo (2014) criou uma área de metadados para facilitar o processo de ETL. A Figura 7 mostra a modelagem dos metadados com suas chaves, atributos e relacionamentos. A tabela “Meta_Metric_Ranges” contém os intervalos qualitativos para cada um das métricas de código-fonte. A tabela “Meta_Scenario” contém os cenários de limpeza do código-fonte e também as recomendações de limpeza para cada cenário de limpeza previsto.

A tabela “Meta_Metric_Ranges_Meta_Scenario” possui duas chaves da tabela “Meta_Metric_Ranges”, pois Rêgo (2014) observou a que cada cenário de limpeza identificado deve conter dois intervalos qualitativos de métricas de código-fonte.

Figura 7 – Metadados do Data Warehouse por [Rêgo \(2014\)](#)

3.5 Considerações finais do Capítulo

—Considerações finais do Capítulo

4 Projeto de Estudo de Caso

Este capítulo apresenta o protocolo do estudo de caso a ser realizado. A sua definição e os requisitos necessários para sua execução são identificados e explicados.

4.1 Definição

Um estudo de caso consiste no estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento (GIL, 2008), assim representa uma maneira de se investigar um tópico empírico seguindo-se um conjunto de procedimentos pré-especificados (YIN, 2001). Uma vantagem dos estudos de caso é que eles são mais fáceis de planejar e são mais realistas, mas as desvantagens são que os resultados são difíceis de generalizar e mais difícil de interpretar, ou seja, é possível mostrar os efeitos de uma situação típica, mas requer mais análise para que se possa generalizar a outras situações (WOHLIN, 2012).

A estrutura do protocolo de estudo de caso proposta se baseia em Brereton, Kitchenham e Budgen (2008) e se apresenta dividido nos seguintes tópicos, cada um com seu objetivo específico:

- **Background - Seção 4.2:** Identificar outros estudos acerca do tópico, define a questão de pesquisa principal e suas proposições derivadas que serão abordado por este estudo.
- **Design - Seção 4.3:** Identificar se é um caso único ou múltiplo e seu propósito geral.
- **Seleção - Seção 4.4:** Critérios para a seleção do caso e descrição do objeto de estudo.
- **Fonte e Método de Coleta de Dados - Seção 4.5:** Identificar os dados que serão coletados, definindo um plano para a coleta e como a informação será armazenada.
- **Análise - Seção 4.6:** Identificar os critérios para interpretação dos resultados do estudo de caso, relacionar os dados com a questão de pesquisa e elaborar a explicação do encontrado.
- **Validade - Seção 4.7:** Se baseia nos tipos de validades aplicáveis a um estudo de caso por Yin (2001), sendo elas: constructo, interna, externa e confiabilidade.

4.2 Background

4.2.1 Outros Trabalhos

O uso do monitoramento de métricas de código-fonte num ambiente de data warehousing foi proposto e desenvolvido no trabalho de [Rêgo \(2014\)](#).

4.2.2 Problema

Dificuldade em analisar e interpretar dados numéricos a respeito da qualidade do código fonte, fator essencial na tomada de decisão quanto a refatoração

4.2.3 Questão de Pesquisa

A questão de pesquisa deriva do problema identificado, e busca definir o propósito da medição, o objeto que será medido, a questão a se tratar, e o ponto de vista que se delimita a questão ([BASILI; CALDIERA; ROMBACH, 1996](#))

Para responder a questão de pesquisa foi construído um GQM (*Goal-Question-Metric*) ([BASILI; CALDIERA; ROMBACH, 1996](#)) representado na Figura 8. O GQM possui objetivos específicos para a questão de pesquisa, cada um tem suas questão específicas que busca responder com a coleta de métricas a questão maior de pesquisa e assim atender o problema do estudo de caso. Foi definido um objetivo para a questão de pesquisa, e para ele foram levantadas nove questões. A definição dessa estrutura com a caracterização de cada item está a seguir.

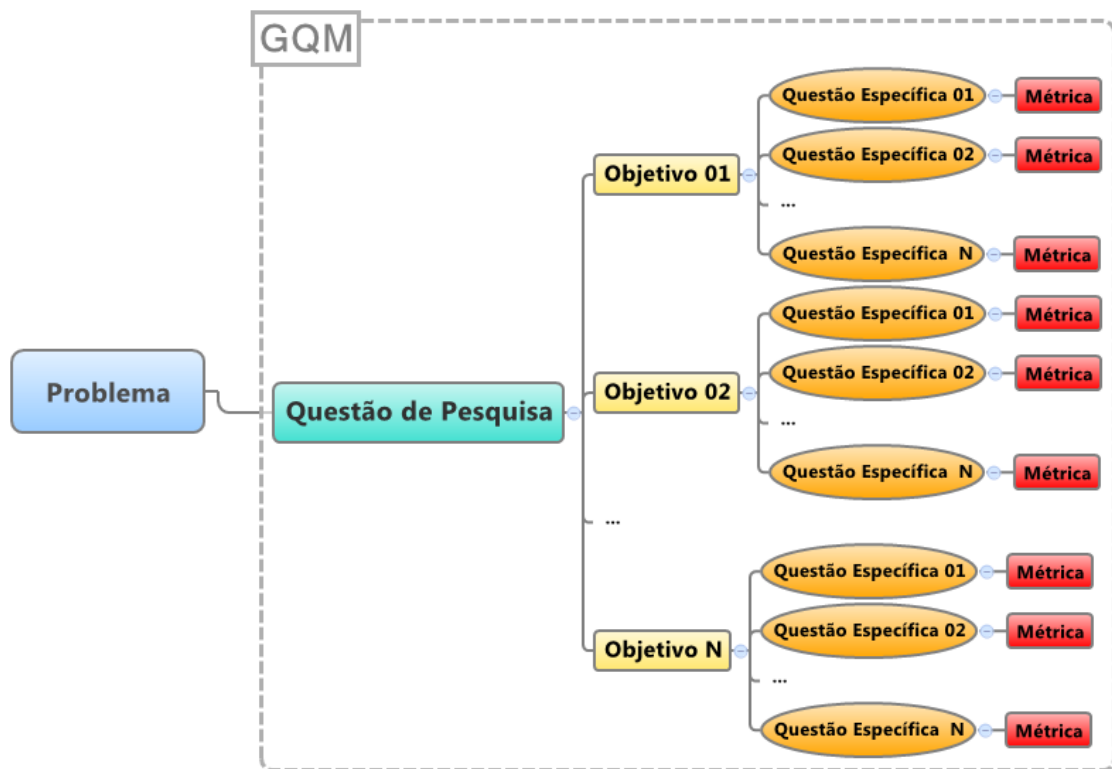


Figura 8 – Estrutura do Estudo de Caso

Questão de Pesquisa: *A solução de data warehousing é eficaz e eficiente no monitoramento de métricas de código-fonte para o apoio à decisões de refatoração do ponto de vista da equipe de qualidade?*

Objetivo 01: Avaliar a eficácia e eficiência da solução de *Data Warehousing* do ponto de vista da <equipe de qualidade> no contexto de desenvolvimento de software.

Questão 01: Quantas tomadas de decisão foram realizadas pela equipe baseado no uso da solução?

Fonte: Observação em campo.

Métrica: número de decisões tomadas.

Questão 02: Quantas tomadas de decisão foram realizadas pela equipe baseado no uso da solução em um <período de tempo>?

Fonte: Observação em campo (áudio/vídeo).

Métrica: número de decisões tomadas/tempo.

Questão 03: Qual a avaliação da equipe de qualidade quanto a detecção de ce-

narios de limpeza de código?

Fonte: Questionário com a equipe de qualidade.

Métrica: muito bom, bom, regular, ruim, muito ruim.

Questão 04: Com que frequência a equipe de qualidade encontra falhas relacionados à utilização da ferramenta?

Fonte: Observação em campo (áudio/vídeo).

Métrica: quantidade de falhas.

Questão 05: Com que frequência a equipe de qualidade encontra falhas relacionados à utilização da ferramenta por tempo?

Fonte: Registro de Observação em campo (áudio/vídeo).

Métrica: quantidade de falhas / tempo.

Interpretação do valor da métrica: Quanto mais próximo de zero melhor. $X \geq 0$.

Questão 06: Qual a proporção de uso da ferramenta para a tomada de decisões?

Fonte: Questionário com a equipe de qualidade.

Métrica: número de decisões tomadas / número de vezes que a solução foi usada

Questão 07: Qual a quantidade de cenários que foram corrigidos após utilização da solução?

Fonte: Código fonte.

Métrica: números de cenários corrigidos por release / número de cenários encontrados por release.

Questão 08: Qual o nível de satisfação do uso da solução em comparação à solução anterior?

Fonte: Equipe de Qualidade.

Métrica: muito satisfeito, satisfeito, neutro, insatisfeito, muito insatisfeito.

Questão 09: Qual a taxa de oportunidade de melhoria de código da solução em um intervalo de tempo (sprint, release)?

Fonte: Código-fonte.

Métrica: A Taxa de Aproveitamento de Oportunidades de Melhoria de Código, em uma release, é calculada como:

$$T_r = \frac{\sum_{i=1}^n Ce_i}{\sum_{i=1}^n Cl_i}$$

onde Ce é o total de cenários de limpezas identificados e Cl é total de classes em uma release.

Interpretação do valor da métrica:

- **Número de classes crescente e constante, número de oportunidade de melhoria estável:** Projeto cresceu mais dos que o cenários, cenários podem ou não estar sendo tratados.
- **Número de classes crescente e constante, número de oportunidade de melhoria crescendo:** Projeto cresce junto com cenários que não são tratados com eficiência ou não são tratados
- **Número de classes crescente ou não, mas constante, número de oportunidade de melhoria diminuindo:** Projeto cresce e cenários são tratados com eficiência.

4.3 Design

[Stake \(1995\)](#) Identifica três modalidades de estudo de caso: intrínseco, instrumental e coletivo. Dentre essas definições, este estudo de caso se caracteriza com instrumental, pois se busca um entendimento geral sobre o problema de pesquisa a partir do estudo de um caso particular. Tal característica se assemelha ao tipo de estudo de caso definido como exploratório por [Yin \(2001\)](#), onde a visão acurada de um caso particular poderá fornecer uma visão geral do problema considerado.

A justificativa para este estudo de um caso particular se dá pelo motivo de que o tema abordado foi pouco explorado até o momento. Assim espera-se que este caso único componha um estudo de múltiplos casos, pois para [Gil \(2008\)](#) considera que esta abordagem possa inserir as evidências coletadas aqui em diferentes contextos, com o intuito de elaborar uma pesquisa de melhor qualidade.

4.4 Seleção

A organização selecionada para este estudo de caso foi o TCU. O software que terá seu código-fonte analisado foi desenvolvido segundo a metodologia , com uma equipe de .

4.5 Fonte e Método de Coleta de Dados

Os dados serão coletados no órgão por meio de entrevistas informais, questionários, análise de documentos da organização, observação em campo e do código-fonte do software disponibilizado. Toda a informação coletada será categorizada e armazenada em meios eletrônicos.

Os questionários serão construídos a partir das questões específicas do processo levantado neste estudo de caso.

4.6 Análise

O processo de análise é composto pelas seguintes etapas:

- **Categorização:** Organização dos dados em duas categorias: Qualitativos e quantitativos. Os dados qualitativos referem-se aos questionários realizados. Os dados quantitativos, por sua vez, referem-se aos valores numéricos da solução de DW para monitoramento de métricas
- **Exibição:** Consiste na organização dos dados coletados para serem exibidos através de gráficos, tabelas e texto para poderem ser analisados.
- **Verificação:** Atestar padrões, tendências e aspectos específicos dos significados dos dados. Procurando assim gerar uma discussão e interpretação de cada dado exibido.
- **Conclusão:** Agrupamento dos resultados mais relevantes das discussões e interpretações dos dados anteriormente apresentados.

4.7 Validade

Para se julgar a qualidade de um projeto de pesquisa, [Yin \(2001\)](#) propõem quatro testes que devem ser considerados ao longo das fases de um estudo de caso, sendo eles:

- **Validade do Constructo:** Está presente na fase de coleta de dados onde deve ser evidenciado as múltiplas fontes de evidência e a coleta de um conjunto de métricas para que se possa saber exatamente o que medir e quais dados são relevantes para o estudo, de forma a responder as questões de pesquisa ([YIN, 2001](#)) buscou garantir a validade de construção ao se definir objetivos com evidências diferentes. Estas por sua vez estão diretamente relacionadas com os objetivos do estudo de caso e os objetivos do trabalho
- **Validade interna:** Para [Yin \(2001\)](#), o uso de várias fontes de dados e métodos de coleta permite a triangulação, uma técnica para confirmar se os resultados de

diversas fontes e de diversos métodos convergem. Dessa forma é possível aumentar a validade interna do estudo e aumentar a força das conclusões. A triangulação de dados se deu pelo resultado da solução de DW que utiliza o código-fonte e foi explicada no capítulo 3, de base de documentos, de questionários e de entrevistas para coleta de dados. A triangulação de métodos ocorreu pelo uso de métodos de coleta quantitativos e qualitativos

- **Validade externa:** Por este ser um caso único a generalização do estudo de caso se dá de maneira pobre (YIN, 2001), assim é necessário a utilização do estudo em múltiplos casos para que se comprove a generalidade dos resultados. O fato deste trabalho ser o primeiro a utilizar a solução para o estudo de caso no órgão, faz com que não seja possível correlacionar os resultados obtidos a nenhum outro estudo.
- **Confiabilidade:** Sobre isso, Yin (2001) associa à repetibilidade, desde que seja usada a mesma fonte de dados. Nesse trabalho o protocolo de estudo de caso apresentado nessa seção garante a repetibilidade desse trabalho e consequentemente a validade relacionada a confiabilidade

4.8 Considerações finais do Capítulo

5 Conclusão

Referências

- BALLARD, C. (Ed.). *Dimensional modeling: in a business intelligence environment*. 1st ed. ed. San Jose, Calif.: IBM International Technical Support Organization, 2006. (IBM redbooks). ISBN 0738496448. Citado 2 vezes nas páginas 21 e 22.
- BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. *The Goal Question Metric Approach*. [S.l.]: Encyclopedia of Software Engineering, 1996. Citado na página 29.
- BRERETON, P.; KITCHENHAM, B.; BUDGEN, D. Using a protocol template for case study planning. In: *Proceedings of EASE 2008*. [S.l.]: BCS-eWiC, 2008. Citado na página 28.
- CODD, E.; CODD, S.; SALLEY, C. *Providing OLAP (On-line Analytical Processing) to User-analysts: An IT Mandate*. [S.l.]: Codd & Associates, 1993. Citado na página 18.
- ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. Sao Paulo (SP): Pearson Addison Wesley, 2011. Citado 6 vezes nas páginas 17, 18, 19, 20, 21 e 22.
- GIL, A. C. *Como elaborar projetos de pesquisa*. Sao Paulo: Atlas, 2008. Citado 2 vezes nas páginas 28 e 32.
- HOBBS, L. et al. *Oracle 10g Data Warehousing*. Elsevier Science, 2011. (Oracle9iR2 Data Warehousing Series). Disponível em: <<http://books.google.com.br/books?id=EzpC7cuOGqUC>>. Citado na página 17.
- INMON, W. H. *Building the Data Warehouse*. New York, NY, USA: John Wiley & Sons, Inc., 1992. Citado na página 17.
- KIMBALL, R. *The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses*. [S.l.]: Wiley. com, 1998. Citado na página 20.
- KIMBALL, R. (Ed.). *The data warehouse lifecycle toolkit*. 2nd ed. ed. Indianapolis, IN: Wiley Pub, 2008. Citado na página 17.
- KIMBALL, R.; ROSS, M. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2nd. ed. New York, NY, USA: John Wiley & Sons, Inc., 2002. Citado 5 vezes nas páginas 17, 19, 20, 22 e 23.
- NERI, H. R. *Análise, Projeto e Implementação de um Esquema MOLAP de Data Warehouse utilizando SGBD-OR Oracle 8.1*. Universidade Federal da Paraíba - UFPB: [s.n.], 2002. Citado na página 17.
- RÊGO, G. B. Monitoramento dea métricas de código-fonte com suporte de um ambiente de data warehousing: um estudo de caso em uma autarquia da administração pública federal. 2014. Disponível em: <<http://bdm.unb.br/handle/10483/8069>>. Citado 9 vezes nas páginas 9, 10, 15, 23, 24, 25, 26, 27 e 29.
- SHARMA, N. *Getting started with data warehousing*. [S.l.]: IBM Redbooks, 2011. Citado na página 18.

STAKE, R. E. *The art of case study research*. Thousand Oaks: Sage Publications, 1995. ISBN 0803957661. Citado na página 32.

WOHLIN, C. *Experimentation in software engineering*. New York: Springer, 2012. Citado na página 28.

YIN, R. K. *Estudo de caso: planejamento e metodos*. Porto Alegre: Bookman, 2001. Citado 4 vezes nas páginas 28, 32, 33 e 34.