Modelagem de um Data Mart para os Telescópios de Múons Tupi

Lucas Bertelli Martins*
Daniel Cardoso Moraes de Oliveira**

Resumo

Devido ao avanço tecnológico e a busca incessante pelo conhecimento, pesquisadores se esforçam cada vez mais para alcançar seus objetivos. As áreas do conhecimento se dialogam somando saberes na realização de suas tarefas. Os telescópios de múons são utilizados no estudo de eventos transientes solares. Alguns desses eventos podem afetar os modernos meios de comunicação e o clima da Terra. O múon é a única partícula carregada capaz de penetrar profundamente no subsolo terrestre. Os telescópios Tupi são exemplos de telescópios de múons, eles geram um grande volume de dados, que precisam ser consultados e agregados pelos pesquisadores. Tais telescópios são capazes de detectar múons, que representam 80% dos raios cósmicos energizados que atingem o nível do mar. Pesquisadores brasileiros construíram o telescópio para medir continuamente o fluxo de partículas derivadas da radiação solar, com o propósito de investigar possíveis relações entre o ciclo solar e variação climática. Entretanto, hoje os dados são armazenados em arquivos texto, o que dificulta as consultas. O objetivo deste trabalho é propor um *Data Mart* para os dados do Tupi, possibilitando aos físicos realizarem suas consultas de forma fácil e com desempenho aceitável.

Palavras-chave: Data Warehouse. Data Mart. Modelagem Dimensional. Big Data. Telescópios Tupi.

* Graduando em Sistemas de Informação; aluno do Instituto de Computação da Universidade Federal Fluminense (UFF), RJ; Universidade Federal Fluminense, Av. Gal. Milton Tavares de Souza, s/nº, Boa Viagem, *Campus* da Praia Vermelha, 24210-346 – Niterói, RJ; lucasbm@id.uff.br.
** Prof. D.Sc. Daniel Cardoso Moraes de Oliveira; Orientador; professor do Instituto de Computação da Universidade Federal Fluminense (UFF); RJ; Universidade Federal Fluminense, Av. Gal. Milton Tavares de Souza, s/nº, Boa Viagem, *Campus* da Praia Vermelha, 24210-346 – Niterói, RJ; danielcmo@ic.uff.br.

Trabalho aprovado em 19/07/2017.

1 **INTRODUÇÃO**

A partícula múon é a componente carregada mais abundante da radiação cósmica secundária ao nível do mar e a única partícula com carga elétrica capaz de penetrar profundamente no subsolo terrestre (VASCONCELOS *et al.*, 2015, p. 31). A medição do fluxo dessa partícula permite estudar eventos transientes solares, tais como: erupções solares, ejeções de massa coronal (EMC), choques interplanetários de várias origens, radiações e tempestades geomagnéticas (AUGUSTO; OJEDA, 2006). Alguns desses eventos podem causar consequências para os modernos meios de comunicação e clima da Terra (VASCONCELOS *et al.*,2015, p. 1; 21; 23).

Existem telescópios que fazem essa detecção, como por exemplo, os da classe Tupi. Esses telescópios geram um grande volume de dados diário, aproximadamente 48.000 leituras de múons. Atualmente, esses dados são armazenados em arquivos de texto plano com extensão do tipo .DAT (que são gerados automaticamente pelo telescópio). Cada arquivo possui uma leitura colhida a cada (aproximadamente) 2 segundos. São aproximadamente 48.000 entradas por arquivo, podendo ocorrer entradas com intervalos menores dependendo da configuração.

Os físicos necessitam consultar tais dados, além disso, precisam realizar agregações sobre esses dados (somatórios, médias, *etc*.), o que torna o trabalho tedioso e propenso a erros se feito de forma manual ou via *scripts*. Nota-se que esse problema é um clássico problema de *Big Data* e *Data Science*. O *Big Data* pode ser definido como um grande e complexo conjunto de dados, cujos métodos de processamento tradicionais seriam insuficientes para seu tratamento – que inclui processos como análise, captura, pesquisa, compartilhamento, armazenamento, transferência, visualização e segurança das informações (PASSOS, 2016, p. 392). Já a *Data Science* é descrita como a ciência responsável pela análise e utilização de dados que incorporam técnicas e teorias de diversas áreas, como lógica, matemática, estatística, computação, engenharia e economia.

Nos últimos anos diversas técnicas têm sido propostas para se trabalhar com *Big Data*, como a Modelagem Dimensional, o uso de *Data Warehouse* e *Data Marts* (INMON, 2002).

Devido à necessidade de se trabalhar com esse volume de dados surgiram os *Data Warehouse* (DW) como alternativa para solucionar a demanda. Um *Data Warehouse* é um grande banco de dados contendo dados históricos resumidos em diversos níveis de detalhamento (TREPPER, 2000, p. 289). Um *Data Warehouse* reúne e consolida informações de diversos *Data Marts* e sistemas da organização, consiste em uma coleção de dados orientada por assuntos, variante no tempo, e não volátil, que visa apoiar os processos de tomada de decisão (INMON, 1999, p. 375).

Os *Data Marts* são bancos de dados modelados de forma dimensional que visam atender os requisitos específicos de um departamento da empresa. Possuem dados sumarizados, por exemplo, dados agregados por mês, trimestre ou ano (INMON, 2002, p. 28-36). Assim como o DW, os *Data Marts* são desnormalizados, armazenam dados históricos, não são voláteis e auxiliam na tomada de decisão.

A modelagem dimensional é utilizada para desenvolvimento desses bancos de dados, essa modelagem estrutura os dados em tabelas: fatos e dimensões. A tabela fato é a principal tabela nesse modelo, pois ela armazena os dados relativos ao desempenho do negócio, como por exemplo, o número de vendas. As tabelas dimensões geralmente representam relacionamentos hierárquicos dos negócios, sendo responsáveis por possibilitar diferentes níveis de detalhamento dos fatos e permitir que se façam agregações destes (KIMBALL, 2002, p. 16-21).

Destacam-se algumas vantagens, observadas por Poe (1998) e Bispo (1998), do modelo dimensional em relação aos modelos de dados relacionais convencionais:

- Permite a criação de um projeto de banco de dados que fornecerá respostas rápidas, com menos tabelas e índices;
- Permite ao administrador do banco de dados trabalhar com projetos mais simples e assim produzir melhores planos de execução;
- Possui uma estrutura mais intuitiva, assemelhando o projeto do banco de dados com a forma como o usuário final pensa e usa os dados (FORTULAN, 2005).

O objetivo deste artigo é propor um *Data Mart* para os dados gerados pelos telescópios Tupi, utilizando-se de técnicas de modelagem dimensional, análises de granularidade e sumarização dos dados. De forma a oferecer um *Data Mart* escalável, seguro e simplificado, facilitando e otimizando a consulta aos dados.

2 **MODELAGEM DIMENSIONAL**

Segundo Kimball (1998), a modelagem dimensional é uma técnica de design de bancos de dados projetada para apoiar as consultas analíticas dos usuários finais. Tal tipo de modelagem é utilizada quando se deseja processar grandes volumes de dados, que faz uso de redundâncias planejadas dos dados para aumentar o desempenho das consultas (KIMBALL, 1998).

Conforme apresentado na introdução o modelo dimensional é composto pelas tabelas fato com suas respectivas dimensões. As dimensões podem ser compartilhadas por tabelas fato diferentes, porém cada tabela fato deve ser referente a um único assunto.

Existem dois modelos de implementação, o Modelo Estrela e o Modelo Floco de Neve. O Modelo Estrela recebe este nome por ser representado com a tabela fato centralizada e com as suas respectivas dimensões no seu entorno. Nesse modelo a tabela fato possui chaves estrangeiras para todas as suas dimensões, é um modelo desnormalizado, que favorece a extração de dados e que assemelha-se ao modelo de negócio, o que facilita a leitura e entendimento.

Já o Modelo Floco de Neve é uma variação do Modelo Estrela, no qual todas as dimensões são normalizadas, fazendo com que sejam geradas quebras na tabela original ao longo de hierarquias existentes em seus atributos. Recomenda-se utilizar esse modelo apenas quando a linha de dimensão possuir muitos atributos e começar a ser relevante do ponto de vista de armazenamento. Devido a essa estrutura o acesso aos dados é mais lento que no Modelo Estrela.

Um *Data Mart* é um subconjunto lógico do *Data Warehouse* completo. Sendo o *Data Warehouse* formado pela união de todos os *Data Marts* da empresa. Além dessa definição lógica simples, também podemos ver o *Data Mart* como um *Data Warehouse* restrito a um único processo do negócio ou a um grupo de processos do negócio de um determinado departamento da empresa. Eles são representados por um modelo dimensional, são baseados em dados granulares e podem ou não conter resumos para aprimorar o desempenho, ou seja, agregações pré-calculadas (KIMBALL, 1998, p. 1.4).

Como abordado anteriormente, um *Data Warehouse* é constituído pela união dos *Data Marts*. Tem o objetivo de integrar e consolidar as informações oriundas de

diversas fontes. Assim, como nos *Data Marts*, um *Data Warehouse* deve ser modelado de forma dimensional, pois em comparação com o modelo relacional, a modelagem dimensional produz modelos mais previsíveis e compreensíveis, facilitando a utilização e assimilação pelos usuários finais. Além de possibilitar consultas com alto desempenho (KIMBALL, 1998, p. 1.4;1.5;1.8;1.9).

Portanto, a modelagem dimensional possui uma estrutura simplificada, mais próxima da visão que o usuário tem do seu negócio, facilitando assim a compreensão, de forma que os próprios usuários possam criar suas consultas, por exemplo, utilizando-se de ferramentas de *Business Intelligence* (BI) conectadas a base de dados.

Comparando com a modelagem relacional tradicional, observa-se que é mais difícil o entendimento e a criação de consultas pelo usuário final, pois essa modelagem não é orientada por assuntos e geralmente possui mais tabelas, sendo necessário um maior conhecimento do esquema para se realizar as junções necessárias. Devido à necessidade de se realizar mais junções e não poder ter redundância de dados o desempenho das consultas é inferior na modelagem relacional tradicional. Entretanto, bancos de dados dimensionais podem sim serem modelados sobre sistemas de gerência de bancos de dados relacionais comuns como o Oracle ou o mySQL.

3 **OS TELESCÓPIOS TUPI**

Os telescópios Tupi são telescópios de múons em uma montagem equatorial, constituídos por dois detectores fixos e outros dois que podem ser orientados de modo a detectar partículas provenientes de uma determinada direção. Os telescópios utilizados para captura de dados desse artigo estão localizados no Instituto de Física da Universidade Federal Fluminense, cuja localização obtida via GPS é 22 ^{0} 54’ 33” de latitude Sul, 43^{0} 08’ 39” de longitude Oeste e no nível do mar (AUGUSTO; OJEDA, 2006).

Os telescópios são automatizados e funcionam continuamente, 24 horas por dia. Seus resultados ajudam a fomentar uma área emergente de estudos conhecida como clima espacial. Trabalham de forma sincronizada para medir continuamente o fluxo de partículas derivadas da radiação do Sol, investigando as possíveis relações

entre os ciclos solares e as variações climáticas da Terra (AUGUSTO; OJEDA, 2006).

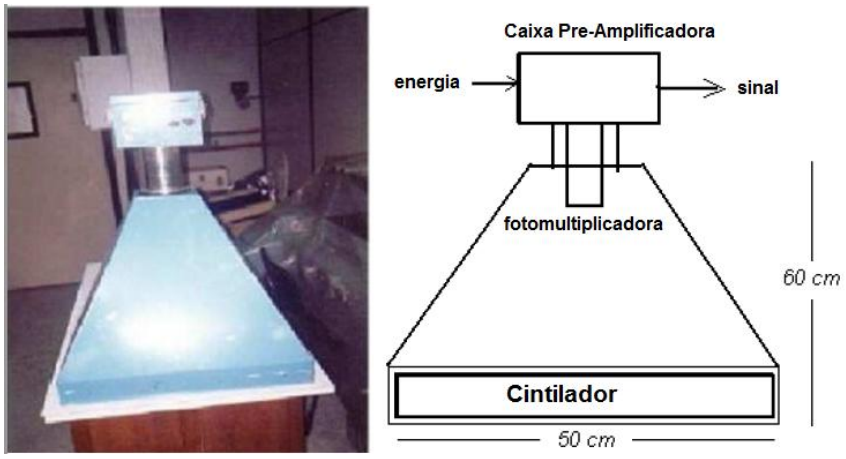

Figura 1. Unidade de detecção padrão de cada telescópio Tupi. Fonte: www.tupi.if.uff.br

A Figura 1 mostra os componentes dos detectores Tupi. Cada detector é composto por um cintilador colocado na base da caixa piramidal, uma fotomultiplicadora (PM) no vértice da pirâmide, cuja saída está conectada a um pré-amplificador (AUGUSTO; OJEDA, 2006).

Quando uma partícula carregada rápida, por exemplo um múon, atravessa o cintilador, este emite luz fluorescente que é captada pela fotomultiplicadora. A fotomultiplicadora converte a luz de baixa intensidade em um sinal elétrico, que é pré-amplificado até uma amplitude suficiente para facilitar uma posterior análise (AUGUSTO; OJEDA, 2006).

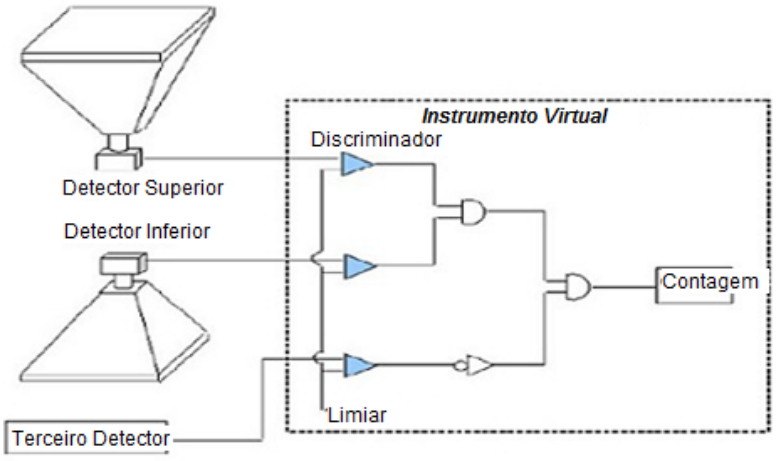

Figura 2. Estrutura padrão de cada telescópio Tupi. Fonte: www.tupi.if.uff.br

A Figura 2 mostra a disposição geral de cada telescópio, a lógica implementada na aquisição de dados, onde os sinais analógicos dos três detectores são digitalizados utilizando a técnica de instrumentos virtuais e as ferramentas do software Lab-VIEW (AUGUSTO; OJEDA, 2006).

Os telescópios contam o número de sinais coincidentes no detector superior e inferior, conforme mostrado na Figura 2. Além disso, cada telescópio usa um veto ou sistema de proteção anti-coincidência, que usa um terceiro detector perto dos dois outros detectores. Este sistema permite que seja feita a detecção de múons que viajam apenas perto do eixo estabelecido entre o detector superior e inferior (AUGUSTO; OJEDA, 2006).

A seguir, a Figura 3 apresenta a configuração atual dos telescópios Tupi. O telescópio vertical, o azul, e seis dos telescópios inclinados, os vermelhos, estão em funcionamento. Os outros telescópios ainda estão em construção e totalizarão quatorze telescópios.

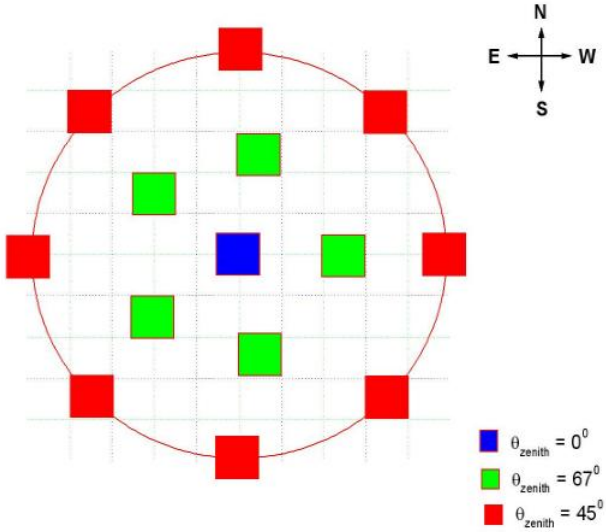

Figura 3. Configuração atual dos telescópios Tupi. Fonte: www.tupi.if.uff.br

4 **CENÁRIO ATUAL**

Atualmente, os dados astronômicos gerados pelos telescópios Tupi são armazenados automaticamente em um repositório remoto do *Google Drive*. Através do uso do aplicativo do *Google Drive* foi compartilhada a pasta no sistema operacional na qual os arquivos são gerados pelos telescópios e assim eles são carregados automaticamente para o *Google Drive*.

Os arquivos possuem a regra de nomenclatura DB_Tupi_”Ano”_”Mes”_”Dia”, na qual, DB_Tupi é um valor fixo e os valores entre aspas devem ser substituídos pelo ano, mês e dia da medição realizada, essa nomenclatura é feita de forma automática pelos telescópios. Além disso, as medições são fragmentadas em arquivos diários, com extensões .DAT (texto plano), a Figura 4 exemplifica a estruturação padrão dos arquivos.

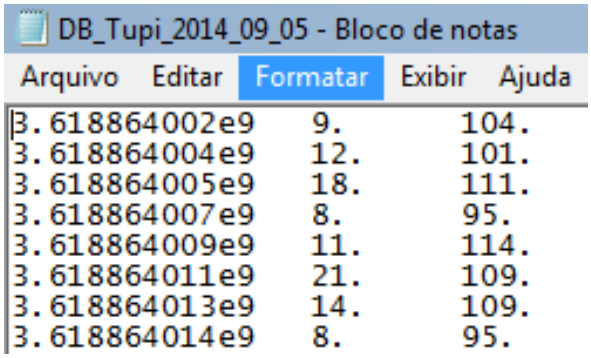

Figura 4. Estrutura padrão dos arquivos gerados pelos telescópios Tupi.

Na Figura 4 observa-se que o conteúdo é separado por tabulações, sendo a primeira tabulação correspondente ao momento da medição representado no formato de tempo universal, que são segundos decorridos a partir de 01/01/1900. A segunda tabulação corresponde às contagens para o telescópio vertical e a última tabulação corresponde a soma das contagens dos telescópios inclinados.

Cabe aos pesquisadores, localizar os arquivos desejados, realizar o *download* e realizar o agrupamento, de acordo com o período de tempo que se deseja analisar. Esse agrupamento hoje deve ser feito via script ou planilhas, o que não é escalável. A seguir, a Figura 5 apresenta uma visualização dessa armazenagem.

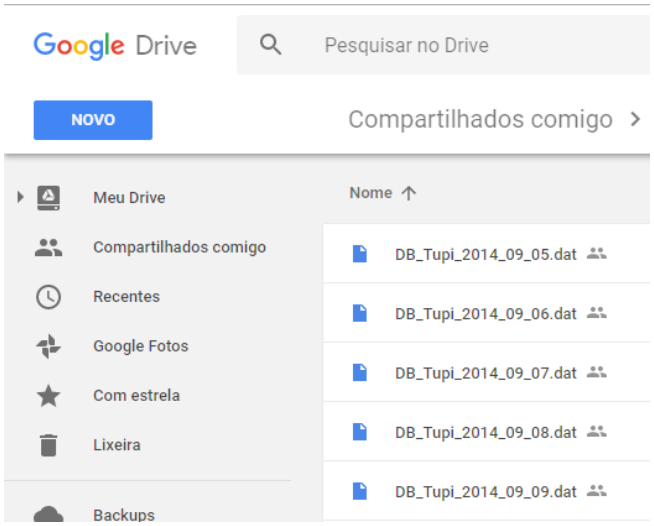

Figura 5. Tela de visualização de arquivos do repositório Tupi no Google Drive.

Diante desse cenário, observamos os seguintes problemas ou limitações causadas por essa forma de armazenamento:

1. Duplicidade de arquivos. Apesar de existir uma regra de nomenclatura, o *Drive* permite que existam arquivos com o mesmo nome, o que gera inconsistência, no sentido de quais são os dados corretos;

2. Pesquisadores tem o trabalho em agrupar o conteúdo dos arquivos para análise, pois eles estão fragmentados em arquivos diários;

3. Os próprios pesquisadores têm de construir os cálculos necessários em uma ferramenta de análise, ainda que sejam análises simples e recorrentes, como informações agrupadas por ano;

4. Cada pesquisador implementa da sua maneira, o que pode gerar inconsistência na análise dos resultados, ou seja, uma análise de um mesmo período de tempo pode chegar a resultados diferentes;

5. As ferramentas de análise muitas vezes não oferecem um desempenho aceitável ou não são capazes de trabalhar com a quantidade de registros desejada;

6. Não oferece apoio a pesquisas científicas que utilizam técnicas de maior poder computacional, como mineração de dados, simulações, aprendizado de máquina, entre outras;

7. Não oferece apoio a integração com sistemas que manipulem esses dados. Como consequência, também desestimula a criação de novos sistemas. Já que, a única forma de acesso direto é via API do Drive, e essa forma apenas permite manipular os arquivos e seus metadados, mas não o seu conteúdo;

8. Não é possível realizar buscas diretas no conteúdo dos arquivos e seus conteúdos não são indexados;

9. Por não existirem campos pré-calculados e o volume de dados ser grande, consultas recorrentes como agrupamentos por ano, mês, dia, hora, se tornam demoradas e, assim, inviabilizadas.

5 **ABORDAGEM PROPOSTA**

5.1 **Modelo de Dados**

A seguir, a Figura 6 apresenta o Modelo Lógico implementado no *Data Mart* Tupi. Utilizamos a modelagem dimensional estrela como técnica de design do banco de dados, sendo a tabela fato representada pela tabela FAT_SINAIS, responsável por armazenar os fatos, que no nosso contexto são contagens de sinais coincidentes dectados no detector vertical e no escaler, armazenados, respectivamente, nos atributos valor_vertical e valor_escaler. Ainda na FAT_SINAIS, os atributos id_tempo e id_telescopio são chaves estrangeiras para as dimensões tempo e telescópio respectivamente e também são a chave primária composta da tabela.

A tabela DIM_TEMPO representa a dimensão tempo, sendo responsável por armazenar todas as possíveis granularidades de tempo para um fato, no nosso contexto está modelada com o granularidade mínima de segundos, ou seja, o atributo dt_data_completa armazena a data, a hora, os minutos e os segundos. Os atributos num_ano, num_mes, num_dia, num_trimestre, num_semestre, num_hora, num_minuto, num_segundo são a representação numérica de partes da data completa. Ainda na DIM_TEMPO, o atributo id_tempo é a chave primária da tabela.

A tabela DIM_TELESCOPIO representa a dimensão telescópio, sendo responsável por armazenar todas os possíveis grupos de telescópios para um fato, foi modelada já pensando em armazenar dados de outros telescópios de múons sem ser os da classe Tupi, de forma a permitir comparações entre medições de diferentes grupos de telescópios múons. Porém, no momento contém apenas o grupo Tupi. Possui os atributos id_telescopio, nm_telescopio, nm_organizacao, nm_pais, nm_local, ds_local e zip_code, que armazenam, respectivamente, chave primária da tabela, nome do grupo de telescópio, nome da organização ao qual o telescópio pertence, o país sede, local dessa organizacao, endereço completo da organização e a zona de informação postal.

A tabela CONTROLE_CARGA é apenas uma tabela auxiliar utilizada pela aplicação Carga Drive-to-Tupi para saber quando foi realizada a última carga de dados. Possuindo os atributos id e data_ultima_carga, que armazenam

respectivamente a chave primária da tabela e a data completa da carga de dados feita no *Data Mart* Tupi.

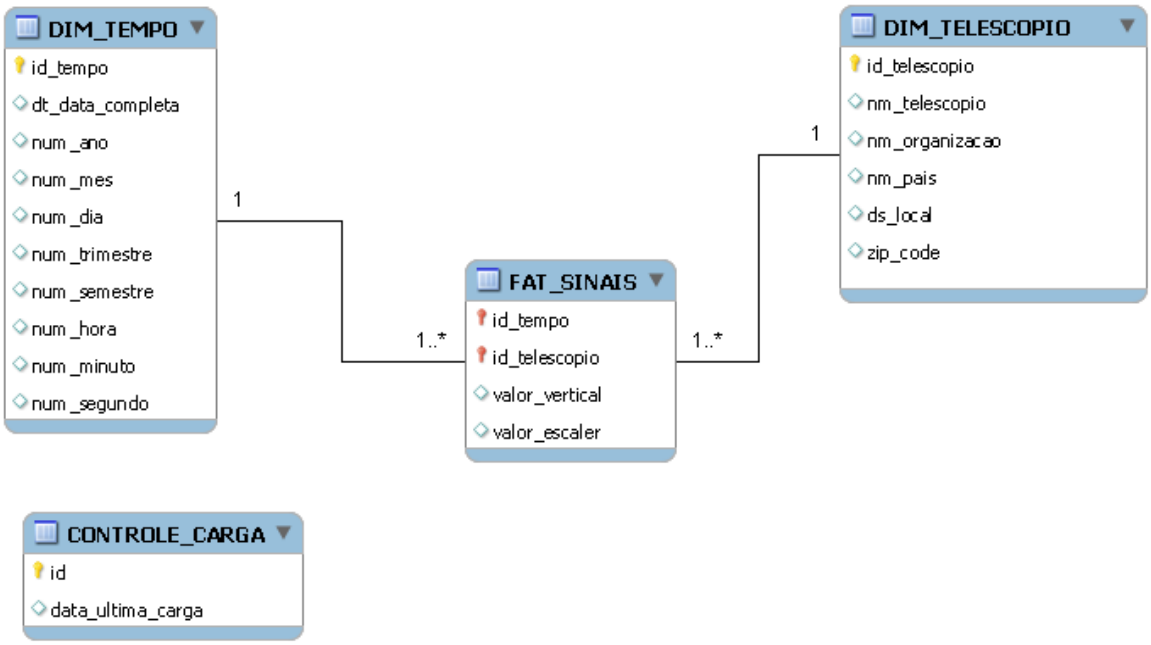

Figura 6. Modelo Lógico do Data Mart Tupi.

O Modelo Físico dos dados do *Data Mart* Tupi é mostrado na Figura 7 a seguir. Nela observamos os tipos utilizados para cada atributo.

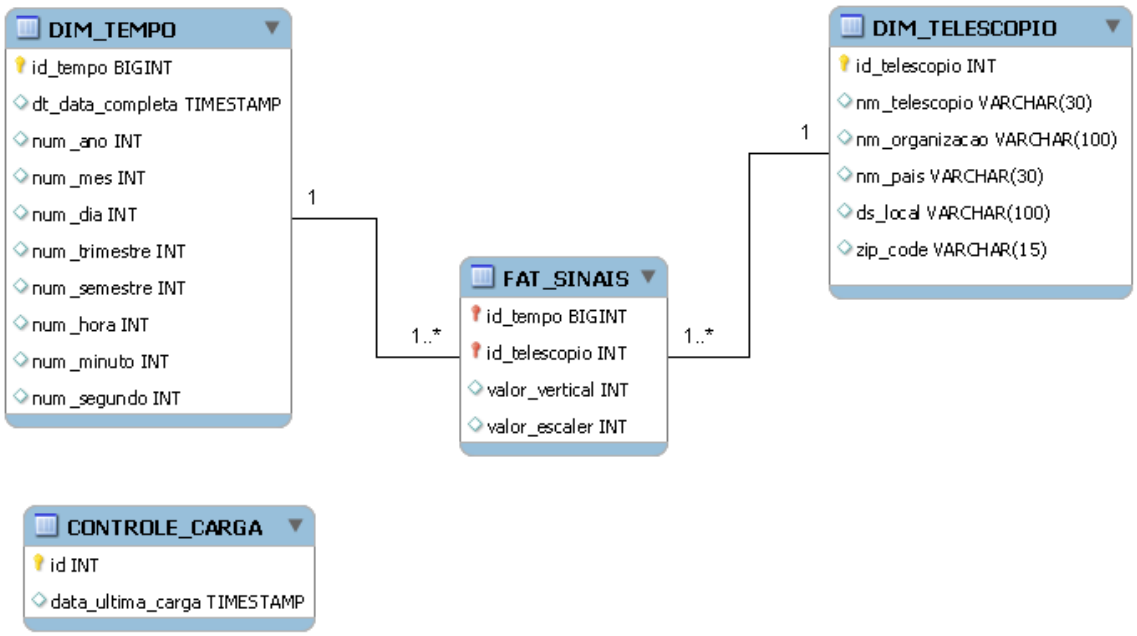

Figura 7. Modelo Físico do Data Mart Tupi.

Foram utilizados resumos para aprimorar o desempenho, ou seja, criadas agregações pré-calculadas para as consultas mais frequentes. As consultas mais frequentes identificadas foram as de somatório simples dos valores dos atributos valor_vertical e valor_escaler agrupadas por ano, mês, dia, hora, trimestre ou semestre.

Essas agregações estão implementadas no modelo na forma de tuplas da DIM_TEMPO com a valor *null* para todos os atributos da tabela, com exceção do id_tempo e dos atributos correspondentes as partições numéricas da data completa que se deseja agrupar. A Figura 8 busca esclarecer a forma descrita de implementação dos agregados.

Para agregados por Ano = 2017 e **Mês = Fevereiro:**

dt_data_completa	num_ano	num_mes	num_dia	num_trimestre	num_semestre	num_hora	num_minuto	num_segundos
null	2017	2	null	null	null	null	null	null

Para agregados por Ano = 2017 , Mês = 2 **e Dia = 5:**

dt_data_completa	num_ano	num_mes	num_dia	num_trimestre	num_semestre	num_hora	num_minuto	num_segundos
null	2017	2	5	null	null	null	null	null

Para agregados por Ano = 2017 , Mês = 2, Dia = 5 e **Hora = 16:**

dt_data_completa	num_ano	num_mes	num_dia	num_trimestre	num_semestre	num_hora	num_minuto	num_segundos
null	2017	2	5	null	null	16	null	null

Para agregados por **Trimestre = 3** do Ano = 2017:

dt_data_completa	num_ano	num_mes	num_dia	num_trimestre	num_semestre	num_hora	num_minuto	num_segundos
null	2017	null	null	3	null	null	null	null

Para agregados por **Semestre = 2** do Ano = 2017:

dt_data_completa	num_ano	num_mes	num_dia	num_trimestre	num_semestre	num_hora	num_minuto	num_segundos
null	2017	null	null	null	2	null	null	null

Figura 8. Exemplo de como são armazenados os agregados no Data Mart Tupi.

Dessa forma, para consultar fatos que não sejam agregações devemos fazer uma junção com a dimensão tempo na qual o atributo dt_data_completa é *null.* Logo, quando queremos consultar somente os valores dos agregados devemos realizar a junção na qual o dt_data_completa não é *null*.

Para encapsular essa lógica como os dados são armazenados e assim facilitar as consultas dos usuários foram criadas as funções:

- Consultas a agregações pré-calculadas:
  - get_agregados_por(ano);
  - get_agregados_por(ano, mês);
  - get_agregados_por(ano, mês, dia);
  - get_agregados_por(ano, mês, dia, hora);
  - get_agregados_trimestre(ano, num_trimestre);
  - get_agregados_semestre(ano, num_semestre).

- Consultas a fatos por período de tempo:
    - get_fatos_ano(ano);
    - get_fatos_mes(ano, mês);
    - get_fatos_dia(ano, mês, dia);
    - get_fatos_hora(ano, mês, dia, hora).

A tabela 1 exemplifica o uso de uma função de agregados em comparação com uma consulta equivalente sem utilizar as funções de agregados. Podemos observar que é mais simples para o usuário final realizar a consulta utilizando as funções de agregados.

Código SQL da Consulta	Utiliza uma função get_agregados?
select * from get_agregados_por(2017) as (valor_vertical bigint, valor_escaler bigint);	SIM
select valor_vertical, valor_escaler from FAT_SINAIS f inner join DIM_TEMPO t on f.id_tempo = t.id_tempo where num_ano = 2017 and dt_data_completa IS NULL and num_mes IS NULL and num_dia IS NULL and num_trimestre IS NULL and num_semestre IS NULL and num_hora IS NULL and num_minuto IS NULL and num_segundo IS NULL;	NÃO

Tabela 1. Comparação entre consultas utilizando ou não funções de get_agregados.

Para exemplificar uma consulta a fatos por período de tempo utilizando as funções criadas em comparação com uma consulta sem utilizar as funções, veja a Tabela 2. Nesta tabela podemos observar mais uma vez que é mais fácil para o usuário final realizar suas consultas utilizando-se das funções criadas para retornar fatos de um período de tempo especificado por parâmetro da função, do que

escrever o código da segunda consulta que não utiliza funções para retornar os fatos.

Além de mais fácil, realizar as consultas através das funções criadas também diminui as possibilidades de inserções de erros nas consultas.

Código SQL da Consulta	Utiliza uma função get_fatos?
select * from get_fatos_dia(2014, 9, 6) as (data_completa_utc timestamp, valor_vertical bigint, valor_escaler bigint);	SIM
select t.dt_data_completa AT time zone 'UTC', f.valor_vertical, f.valor_escaler from FAT_SINAIS f inner join DIM_TEMPO t on f.id_tempo = t.id_tempo where t.num_ano = 2014 and t.num_mes = 9 and t.num_dia = 6 and t.dt_data_completa IS NOT NULL;	NÃO

Tabela 2. Comparação entre consultas utilizando ou não funções de get_fatos.

Ao utilizar essas funções deve-se respeitar a ordem de passagem dos parâmetros e especificar corretamente as variáveis de retorno com seus respectivos tipos compatíveis com o retorno da função. Os tipos retornados são os mesmos utilizados no modelo físico, ver Figura 7.

5.2 **Arquitetura**

A Figura 9 a seguir apresenta a arquitetura da solução proposta, desde o processo de captação dos dados pelos telescópios até a carga no *Data Mart* proposto.

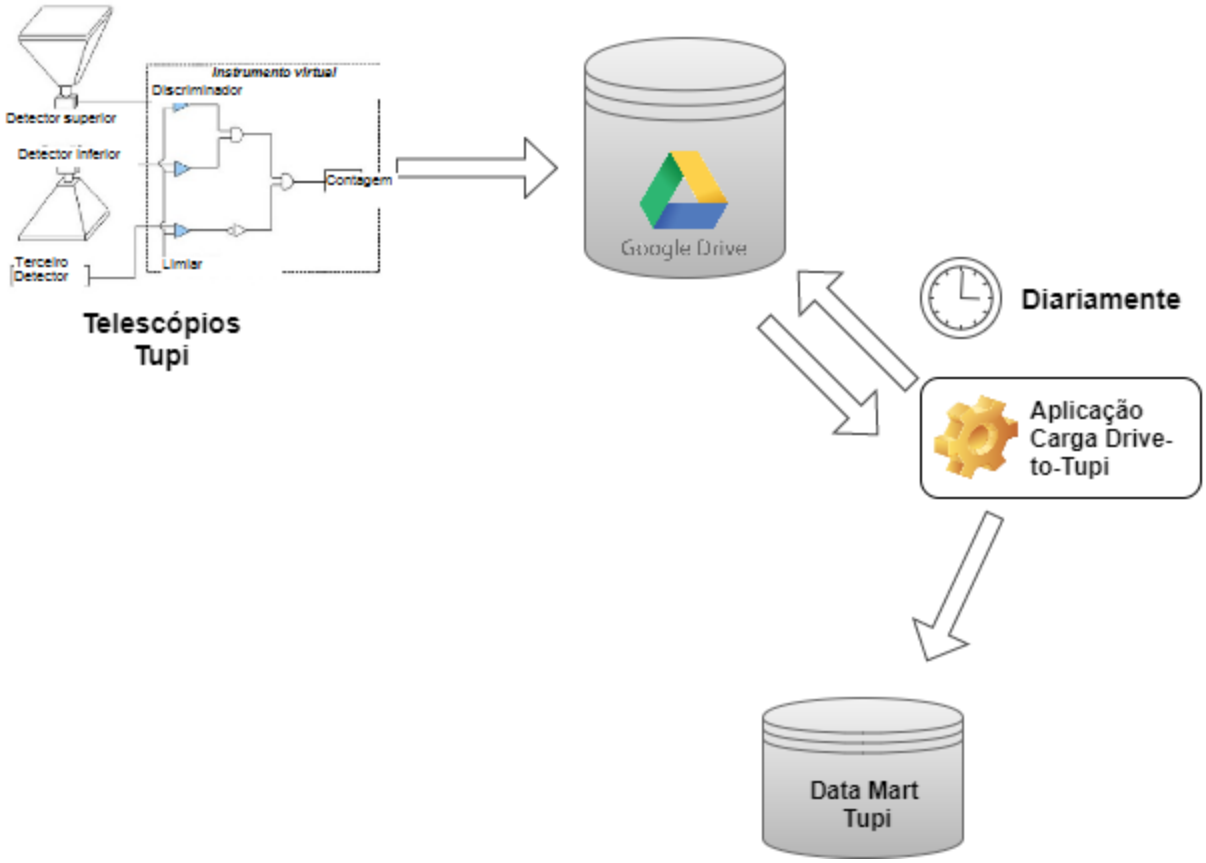

Figura 9. Arquitetura da solução proposta.

Os arquivos gerados pelos telescópios Tupi são armazenados no repositório remoto do *Google Drive*, que chamaremos somente de *Drive* neste artigo. Diariamente a aplicação "Carga Drive-to-Tupi" é executada verificando se existem atualizações ou novos arquivos no *Drive*, caso haja, ela realiza o processo de carga no *Data Mart* Tupi.

5.3 **Implementação**

Na implementação foi utilizada a modelagem dimensional estrela na construção do modelo de dados do *Data Mart* Tupi e criadas as agregações pré-calculadas mais frequentes, como foi detalhado no item 5.1 deste artigo. Esse modelo de dados foi implementado no SGBD PostgreSQL versão 9.6, instalado em um servidor com o sistema operacional Linux, distribuição Mint. Foi desenvolvida a aplicação "Carga Drive-to-Tupi" e utilizado o agendador de tarefas do sistema, o *cron*, para executar a aplicação diariamente.

A aplicação “Carga Drive-to-Tupi” foi desenvolvida na linguagem Java, utilizando recursos da *release* Java 8 e a API do Google Drive versão 3 rev76-1.22.0. A seguir, a Figura 6 exibe um pseudocódigo demonstrando o seu funcionamento.

```
INICIO
VARIAVEIS
novos_arquivos: List;
dt_ult_carga: ZonedDateTime;
dt_ult_carga = CONSULTA_DT_ULT_CARGA_DATAMART();
novos_arquivos = EXISTEM_NOVOS_ARQUIVOS_DO_TUPI (dt_ult_carga);
SE ( novos_arquivos ) FAÇA
      novos_arquivos = ORDENAR_DT_MODIFICA_CRESCENTE(novos_arquivos);
      BAIXAR_NOVOS_ARQUIVOS (novos_arquivos);
      EXCLUIR_INDICES ();
      PARA CADA arquivo EM novos_arquivos FAÇA
          SE( VERIFICA_SE_JA_EXISTE_NO_DATAMART(arquivo) ) FAÇA
               DELETAR_MEDICAO_DO_DATAMART(arquivo);
          FIM SE;
          PARA CADA linha EM arquivo FAÇA
               LEIA linha;
               TRANSFORMA_TU_EM_DATETIME(linha);
               INSERE_NO_DATA_MART(linha);
               ATUALIZA_AGREGADOS_AFETADOS(linha);
          FIM PARA;
      FIM PARA;
      INSERE_DATA_ATUAL_NA_CONTROLE_DE_CARGA();
      CRIAR_INDICES ();
FIM SE;
FIM
```

Tabela 3. Pseudocódigo da aplicação Carga Drive-to-Tupi.

No pseudocódigo, Tabela 3, observamos de forma simplificada os principais procedimentos realizados pela aplicação de carga. Primeiramente, a aplicação consulta na tabela CONTROLE_CARGA do *Data Mart* Tupi quando foi feita a última carga de dados. Utilizando-se da data retornada pela consulta, a aplicação busca no *Drive* por novos arquivos ou atualizações nos arquivos do Tupi, ou seja, ela acessa os metadados de cada arquivo, verifica se ele possui a nomenclatura correta e se os campos correspondentes à data de criação ou a data da última atualização são datas posteriores à data da última carga no *Data Mart* Tupi. Se houver resultados, os arquivos novos/atualizados são ordenados de forma crescente da data de atualização, para que ao serem baixados os arquivos mais atuais sobrescrevam os mais antigos na pasta e assim elimine os arquivos duplicados. Os índices envolvidos são excluídos para agilizar o processo de carga. Para cada arquivo é verificado se já existem medições para aquele dia no *Data Mart* Tupi, caso haja, todos os registros correspondentes ao dia que será inserido são excluídos, dessa forma garantimos que não há duplicidade de dados. Posteriormente, para cada linha contida nos arquivos novos é feita a transformação do valor correspondente ao "tempo universal" para um formato de data, hora, minutos e segundos. Depois da transformação a linha é inserida no *Data Mart* Tupi de acordo com a modelagem e os agregados afetados têm seus valores atualizados. Ao término da carga, os índices são recriados e é inserida a data, hora, minutos e segundos atual na tabela CONTROLE_CARGA, que é a tabela consultada pela função "EXISTEM_NOVOS_ARQUIVOS_TUPI()" para obter a data delimitadora de quais são os arquivos novos/atualizados contidos no *Drive*.

6 **AVALIAÇÃO EXPERIMENTAL**

Foi realizada uma avaliação experimental, na qual foi carregada uma amostra de 754 arquivos de leituras no Drive. Esses arquivos correspondem às medições diárias contidas no período dos anos de 2014 a 2017.

Após a execução da aplicação Carga Drive-to-Tupi, foram carregados apenas 750 arquivos no *Data Mart* Tupi, pois a aplicação descartou 4 arquivos duplicados, mantendo somente os mais recentes.

A tabela de fatos, a FAT_SINAIS, ficou com um total de 33.311.791 tuplas e a tabela da dimensão tempo, a DIM_TEMPO, ficou com um total de 33.311.791 tuplas. A dimensão telescópio, ou DIM_TELESCOPIO, não recebe inserções no processo de carga de dados e a CONTROLE_CARGA teve mais uma tupla inserida, totalizando duas tuplas.

O banco de dados Tupi sem índices ocupou um espaço em disco de aproximadamente 4 GB, e com índices, um espaço aproximado de 12 GB.

A seguir, a Tabela 4 exibe uma comparação entre o tempo de execução decorrido de consultas equivalentes, no caso, consultar o somatório dos atributos valor_vertical e valor_escaler no período de 2017.

Código SQL da Consulta	Tempo de Execução	Utiliza agregados?
select valor_vertical, valor_escaler from FAT_SINAIS f inner join DIM_TEMPO t on f.id_tempo = t.id_tempo where num_ano = 2017 and dt_data_completa IS NULL and num_mes IS NULL and num_dia IS NULL and num_trimestre IS NULL and num_semestre IS NULL and num_hora IS NULL and num_minuto IS NULL and num_segundo IS NULL;	352 msec	SIM
select sum(valor_vertical), sum(valor_escaler) from FAT_SINAIS f inner join DIM_TEMPO t on f.id_tempo = t.id_tempo where t.num_ano = 2017 and t.dt_data_completa IS NOT NULL;	5 min	NÃO

Tabela 4. Comparação do tempo de execução de consultas equivalentes utilizando ou não agregados.

Observamos na Tabela 4 que a primeira consulta teve um tempo de execução de 352 milissegundos, enquanto a consulta que não utiliza agregações, segunda consulta, demorou 5 minutos para terminar a execução. Ressalto ainda, que a amostra contém apenas 750 arquivos e logo esses tempos de execução tendem a

aumentar quando a base de dados estiver completa. Esta comparação justifica a decisão de se utilizar agregações pré-calculadas.

Por fim, realizamos uma consulta para retornar todos os fatos de um dia específico e observamos que teve um tempo de execução de 2 segundos.

Código SQL da Consulta	Tempo de Execução
select t.dt_data_completa AT time zone 'UTC', f.valor_vertical, f.valor_escaler from FAT_SINAIS f inner join DIM_TEMPO t on f.id_tempo = t.id_tempo where t.num_ano = 2014 and t.num_mes = 9 and t.num_dia = 6 and t.dt_data_completa IS NOT NULL;	2 seg

Tabela 5. Tempo de execução de uma consulta a todos os fatos de um dia específico.

6 **CONCLUSÃO**

O estudo apresentou os problemas e limitações da armazenagem atual utilizada em pesquisas nos telescópios de múons Tupi. Além disso, também apresentou os conceitos e técnicas existentes na literatura para lidar com grandes volumes de dados.

De um modo geral, os pesquisadores necessitam de ferramentas adequadas que facilitem seu trabalho, gerando resultados rápidos, precisos e satisfatórios.

Desta forma, foi proposto um *Data Mart*, utilizando-se das técnicas apresentadas, para solucionar os problemas. Foi realizada uma avaliação experimental da solução proposta, na qual, verificaram-se resultados satisfatórios. Evidenciando assim, os objetivos alcançados.

Através das consultas e análises foram confirmados os resultados, tanto na teoria quanto na prática, o que representa uma aceleração no processo de pesquisa.

A partir do estudo as pesquisas científicas na área poderão utilizar de um maior poder computacional, podendo economizar não só tempo, mas também alcançar novas perspectivas e descobertas.

Como sugestões para trabalhos futuros, pode-se apontar a elaboração de uma aplicação *Web* que visa oferecer visualizações sobre os dados do *Data Mart* Tupi, como gráficos de series temporais dinâmicos, de acordo com os filtros de tempo oferecidos. Além disso, uma aplicação *Web* também permitirá aos pesquisadores visualizar e analisar esses dados de qualquer lugar, como por exemplo, em uma reunião externa a universidade.

Outro trabalho futuro interessante seria o estudo de padrões nos dados do *Data Mart* Tupi, utilizando-se de técnicas de mineração de dados, o que pode gerar novas descobertas na área.

Modeling a Data Mart for Tupi Muons Telescopes

Abstract

Due to technological advancement and incessant search for knowledge, researchers are increasingly striving to achieve their goals. The knowledge areas dialogue by adding knowledge in the accomplishment of their tasks. The muons telescopes are used in the study of transient solar events. Some of these events can affect the modern media and climate of the Earth. The muon is the only charged particle capable of penetrating deep into the Earth's subsoil. Tupi telescopes are examples of muons telescopes, they generate a large volume of data, which need to be queried and aggregated by researchers. They are able to detect muons, which represent 80% of the charged cosmic rays that reach sea level. Brazilian researchers have constructed the telescope to continuously measure the influx of particles derived from solar radiation, in order to investigate possible association between the solar cycle and climatic variation. The objective of this paper is to propose a Data Mart for Tupi data, allowing the physicists to perform their queries in an easy way and with acceptable performance.

Keywords: Data Warehouse. Data Mart. Dimensional Modeling. Big data. Tupi Telescopes.

REFERÊNCIAS

AUGUSTO, Carlos Roberto A.; OJEDA, Carlos E. Navia. **Observação do excesso e deficit de muons no nível do mar em associação com eventos solares transientes**. 2006. Tese de Doutorado. Tese de Doutorado em Física, Universidade Federal Fluminense, Rio de Janeiro, RJ.

BISPO, C. A. F. **Uma análise da nova geração de sistemas de apoio à decisão.** 1998. 165 p. Dissertação (Mestrado em Engenharia de Produção) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 1998.

BISPO, C. A. F.; CAZARINI, E. W. A nova geração de sistemas de apoio à decisão. In: ENEGEP, 18, 1998, Niterói, Rio de Janeiro, Brasil. **Anais...** Niterói: ABEPRO, 1998.

FORTULAN, Marcos Roberto; GONÇALVES FILHO, Eduardo Vila. **Uma proposta de aplicação de Business Intelligence no chão-de-fábrica.** Gestão & Produção, v. 12, n. 1, p. 55-66, 2005.

INMON, William Harvey. **Building the Data Warehousing**. 2002.

INMON, William H.; WELCH, J. D.; GLASSEY, Katherine L. **Gerenciando data warehouse**. Makron Books, 1999.

KIMBALL, Ralph; ROSS, Margy. **The data warehouse toolkit: the complete guide to dimensional modeling**. 2 ed. John Wiley & Sons, 2002.

KIMBALL, Ralph. **The data warehouse lifecycle toolkit: expert methods for designing, developing, and deploying data warehouses.** John Wiley & Sons, 1998.

PASSOS, Danielle Sandler dos. **Big data, data science e seus contributos para o avanço no uso da open source intelligence.** NOVA Information Management

School, Instituto Universitário de Lisboa (ISCTE), Universidade Nova de Lisboa. Lisboa, 2016.

POE, V.; KLAUER, P.; BROBST, S. **Building a data warehouse for decision support.** 2 ed. Upper Saddle River - NJ: Prentice-Hall PTR, 1998. 285 p.

TREPPER, C. **Estratégias de e-commerce**. Rio de Janeiro: Campus, 2000.

VASCONCELOS, Débora Nunes Barros de et al. **Telescópio de múons para estudo da atividade solar.** 2015.

APÊNDICE A – Lista de ilustrações

APÊNDICE B – Lista de Tabelas