

**UNIVERSIDADE ESTADUAL DE GOIÁS**  
**UNIDADE UNIVERSITÁRIA SANTA HELENA DE GOIÁS**

**REFLEX: FERRAMENTA DE MODELAGEM, CRIAÇÃO E MIGRAÇÃO DE  
DADOS DE DATA WAREHOUSE**

**THYERRE RANGEL MORAIS DA SILVA**

**SANTA HELENA DE GOIÁS – GO.**

**2020**

**UNIVERSIDADE ESTADUAL DE GOIÁS**  
**UNIDADE UNIVERSITÁRIA SANTA HELENA DE GOIÁS**

**REFLEX: FERRAMENTA DE MODELAGEM, CRIAÇÃO E MIGRAÇÃO DE  
DADOS DE DATA WAREHOUSE**

**THYERRE RANGEL MORAIS DA SILVA**

Monografia apresentada à Banca Examinadora da  
Universidade Estadual de Goiás, Unidade Universitária  
Santa Helena de Goiás, como parte dos requisitos para  
obtenção do Título de Bacharel, no Curso de Sistemas de  
Informação, sob a orientação do Prof. Me. Edmar A.  
Yokome.

**SANTA HELENA DE GOIÁS – GO.**

**2020**

**UNIVERSIDADE ESTADUAL DE GOIÁS**

**UNIDADE UNIVERSITÁRIA SANTA HELENA DE GOIÁS**

## **FOLHA DE APROVAÇÃO**

Autor:Thyerre Rangel Moraes da Silva

Título: Ferramenta de modelagem, criação e migração de dados de Data warehouses:  
Desenvolvimento de uma ferramenta web, para a modelagem, criação e transferência  
de dados de um Data Warehouse/Data Mart.

Aprovada em Santa Helena de Goiás, .....de.....de 20.....

Banca Examinadora:

---

Edmar A. Yokome  
Orientador  
(UEG Câmpus Santa Helena de Goiás/Mestre em Ciência da Computação)

---

Pollyana de Queiroz Ribeiro  
UEG/UnU Santa Helena/ Mestra em Ciência da Computação

---

Ana Clara Araujo Gomes da Silva  
Mestra em Ciência da Computação

**SANTA HELENA DE GOIÁS**  
**2020**

Dedico este trabalho a minha família que sempre esteve presente nesse período importante da minha vida, especialmente a minha avó Iraci da Silva Ferreira, minha mãe Marcia Ferreira Silva e minha namorada Thainara das Neves Silva que sempre acreditaram em meu potencial e me apoiaram nessa trajetória.

## **AGRADECIMENTOS**

Agradeço a Deus por me conceder forças para a conclusão desta jornada tão importante em minha vida. Agradeço a Universidade Estadual de Goiás, Câmpus Santa Helena de Goiás, por promover esta oportunidade em que foi possível o aprimoramento na minha área de atuação profissional e agregação de conhecimento por meio deste curso. Agradeço ao meu orientador Edmar Yokome, por sua orientação durante esse projeto, além de todo o corpo docente que contribuiu disseminando conhecimento e experiências que muito contribuíram para a minha formação. Agradeço aos meus colegas de trabalho por compartilhar todo o conhecimento necessário para a realização desse projeto.

## **RESUMO**

DA SILVA, Thyerre Rangel Morais. REFLEX: Ferramenta de modelagem, criação e migração de dados de Data warehouses. 2020. p31 Monografia (Graduação em Sistemas de Informação) – Universidade Estadual de Goiás – UEG – Câmpus Santa Helena de Goiás, 2020.

Com o crescimento da internet, as empresas começaram a produzir um grande volume de dados que são armazenados em bancos de dados de forma a preservar as informações a serem utilizadas para auxílio na tomada de decisão de uma organização, através de relatórios, gráficos e outras formas de consumo a essas informações, a fim de melhorar diversas áreas dentro e fora da organização. Estima-se que em 2020 as organizações geraram 40 Zettabytes equivalente a 40 trilhões de *gigabytes* um volume que tende a subir com os anos seguintes, muitas dessas organizações possuem sua base de dados unificada em um mesmo cluster fazendo com que a demanda desse banco de dados seja alta podendo trazer quedas no desempenho dos sistemas que utilizam esse mesmo banco de dados, uma solução conhecida é a criação de data warehouse para a alocação dos dados referente a tomada de decisão de uma empresa. O presente trabalho tem como objetivo desenvolver uma ferramenta para auxiliar a visão da modelagem e criação de data warehouse de forma a preservar os princípios de integridade, redundância e segurança de um banco de dados em Mysql, a ferramenta também se encarrega da atualização periódica dos dados do data warehouse com a base principal de dados, podendo ser escolhido a periodicidade dessas atualizações. A ferramenta utiliza em seu *backend* o *framework Laravel* construído a partir da linguagem de programação *PHP*, em seu *frontend* é utilizado o *framework Angular* que é feito sobre as linguagens *TypeScript* *JavaScript*, *HTML* e *CSS* para compor a interface web, a ferramenta também irá utilizar uma base de dados para salvar as configurações de atualização dos dados, schema dos data warehouse e controle de usuário. O sistema auxilia na criação da estrutura do data warehouse e permite a configuração de atualização dos dados periodicamente, tendo um desempenho aceitável para clusters que possuem outras aplicações rodando.

**Palavras-chave:** Data Warehouse, Ferramenta web, Banco de dados, Data Mart.

#### **ABSTRACT**

With the growth of the internet, companies began to produce a large volume of data that is stored in databases in order to preserve the information to be used to assist in the decision-making process of an organization, through reports, graphs and other forms consumption to this information, in order to improve several areas inside and outside the organization. It is estimated that in 2020 organizations generated 40 Zettabytes, equivalent to 40 trillion gigabytes, a volume that tends to increase with the following years, many of these organizations have their database unified in the same cluster, making the demand for this database be high, which may bring down the performance of systems that use this same database, a known solution and the creation of a data warehouse for the allocation of data related to the decision-making process of a company. This work aims to develop a tool to assist the vision of modeling and creation of a data warehouse in order to preserve the principles of integrity, redundancy and security of a database in Mysql, the tool will also be in charge of periodic updating of data from the database. data warehouse with the main database, and the frequency of these updates can be chosen. The tool uses in its backend the *Laravel*

framework built from the PHP programming language, in its frontend the Angular framework is used that is made on the TypeScript JavaScript, HTML and CSS languages to compose the web interface, the tool will also use a database to save data update settings, data warehouse schema and user control. The system assists in the creation of the data warehouse structure and allows the configuration of data update periodically, having an acceptable performance for clusters that have other applications running.

**Keywords:** Data Warehouse, Web tool, Database, Data Mart.

## **SUMÁRIO**

1	LISTA DE FIGURAS	7
1	INTRODUÇÃO	8
1.1	Contextualização	8
1.2	Motivação	10
1.3	Objetivo	10
1.3.1	Objetivo Geral	10
1.3.2	Objetivo Específicos	10
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	Business Intelligence	10
2.2	Data Warehouse (DW) e Data Mart (DM)	12
2.3	Migração de Dados	14
2.4	Atualização de Dados por Pacotes	15
2.5	Banco de Dados	15
2.6	Sistemas de Gerenciamento de Banco de Dados	16
2.7	Banco de Dados Relacional	16
3	MATERIAIS E MÉTODOS	17
3.1	Levantamento de Requisitos	17
3.1.1	Requisitos Funcionais	18
3.1.2	Requisitos não Funcionais	18
3.2	Modelagem do Sistema	18
3.2.1	Diagrama de Caso de Uso	18
3.2.2	Desenvolvimento ágil Kanban	19
3.3	Tecnologias e Materiais	21
3.3.1	IDE	21
3.3.2	Modelagem do Banco de Dados	22
3.3.3	Frontend	24
3.3.4	Backend	26
3.3.5	Funcionamento	26
4	CONCLUSÃO	27
4.1	Resultados obtidos	27
4	Trabalhos futuros	29
	REFERÊNCIAS	31

7

## **LISTA DE FIGURAS**

Figura 1 - Modelo de ETL	11
Figura 2 - Modelo visual de um Data Warehouse e seus Data Marts.	12
Figura 3 - Modelo Estrela	13
Figura 4 - Modelo Floco de Neve	14
Figura 5 - Diagrama de Caso de Uso	19
Figura 6 – Quadro Kanban	20
Figura 7 - IDE Visual Studio Code	23
Figura 8 - Modelagem do banco de dados da ferramenta REFLEX	24
Figura 9 - Hierarquia de componentes e templates no Angular	25
Figura 10 - Monitor de recursos do linux	28
Figura 11 - Tela de seleção da estrutura do data warehouse/mart	29

8

## **1 INTRODUÇÃO**

### **1.1 Contextualização**

O avanço nas tecnologias proporcionou para as organizações uma grande

necessidade de informações a cada dia mais dinâmicas, seguras e consistentes, com a expansão da internet o volume de dados produzidos por essas empresas cresceu exponencialmente o estudo “*A Universe of Opportunities and Challenges*” feito pela consultoria EMC Corporation, aponta que até o ano de 2010 foi gerado um volume de dados digitais de 988 exabytes equivalente a 988 bilhões de gigabytes, esse mesmo estudo que foi produzido para o ano de 2020 cerca de 40 Zettabytes que é equivalente a 40 trilhões de gigabytes.

Com esse crescente volume de dados as organizações têm a necessidade de armazenar esses dados de forma eficiente, segura, consistente e acessível, para que os mesmos sejam manipulados de forma qualificada a fim de extrair informações cruciais que poderão ser utilizadas para diversos fins dentro e fora de uma organização, como o auxílio à tomada de decisões de uma empresa, mudanças operacionais, de produto e serviço, assim podendo otimizar processos, obter respostas mais ágeis e maximizar resultados.

Para a alocação desses dados brutos que irão se tornar futuras informações são usados os banco de dados, que são responsáveis por guardar esses recursos, para realizar a manipulação desses dados é necessário usar uma interface de acesso, que são os Sistemas de Gerenciamento de Banco de Dados (SGBD). Ao longo dos anos foram surgindo diversos SGBD's com objetivos semelhantes, porém com sistemáticas e custos variados como Oracle, Mysql, Postgresql, MongoDB, entre outros.

Existem modelos de SGBD's, alguns deles são: o modelo relacional que é consolidado a algum tempo no mercado, utilizam a linguagem SQL sigla para *Standard Query Language*, para realizar interações com os dados; e o modelo NoSQL ou não relacional, que vem cada vez mais conquistando mercado e espaço nas organizações, esse termo engloba diversas tecnologias diferentes que possuem características que definem o NoSQL.

Os SGBD's de modelo relacional sofre com o escalonamento de dados para grandes volumes de dados, isso pode prejudicar no desempenho dos sistemas que utilizem esse banco de dados para realizar as consultas, inserts e atualizações, trazendo lentidão e

9

travamentos prejudicando a usabilidade do sistema e a agilidade dos processos dentro de uma empresa.

As organizações demandam a necessidade de analisar e transformar dados brutos contidos nos bancos de dados em informações relevantes para uma organização, surgindo então a análise de dados que é o trabalho de entendimento em alto nível dos dados para a

obtenção de informações sobre vendas, marketing, clientes, entre outras possibilidades. Derivado dessa área criou-se um ramo chamado de BI (*Business Intelligence*) que são processos orientados pela tecnologia da informação e análise de dados, para apresentar informações acionáveis para auxílio a executivos, gerentes e outros agentes corporativos.

Uma forma de contornar o problema de desempenho e trabalhar com a análise de dados, consiste em criar uma *Data Warehouse*, fazendo com que rotinas específicas dos sistemas como os relatórios que utilizam o banco de dados principal, busquem as informações em outra base de dados menor e específica para essas rotinas, podendo estar em outra máquina, assim não sobrecarregando a máquina principal que está alocado o banco de dados principal, essa clusterização pode fazer com que o desempenho do banco principal fique disponível para as rotinas dos sistemas que não sejam relatórios evitando travamentos e lentidão.

O objetivo do *Data Warehouse* é armazenar os dados retirados de diversas fontes e facilitar a consulta para gerar documentos que auxilie na análise e *business intelligence* de uma organização, e segue os princípios de um SGBD's que são: independência de dados, compartilhamento, integridade dos dados, segurança e controle de redundância. Ele é projetado e construído por profissionais da área como DBA's (*Database Administrator*) e profissionais de BI, sendo geralmente adotado por organizações de grande porte.

As organizações estão investindo em diversas áreas de atuação capazes de trazer informações relevantes para uma tomada de decisão consistente e ágil. Pequenas e médias empresas estão começando a entender que a melhor maneira de faturar e melhorar seus produtos e serviços é por meio da análise de informações que é possível ser feita adotando a criação de *Data Warehouse*.

A construção da estrutura de um *data warehouse* é trabalhosa e pode levar as médias e pequenas empresas a não optarem por esse método, a solução seria criar ferramentas que facilitem a criação dessa estrutura e torne o modelo de análise de dados mais acessível, sem perder seus princípios. O trabalho em questão propõe a criação de uma ferramenta capaz de auxiliar a construção de um *data warehouse* e a atualização dos seus dados

10

periodicamente, fazendo com que o DBA não se preocupe com a codificação da estrutura e sim com a análise desses dados.

## **1.2. Motivação**



A motivação por trás da criação desse projeto, e a tentativa de fazer com que o profissional de banco de dados possa ter mais tempo com foco na modelagem da estrutura de um *data warehouse* automatizando a codificação e criação da estrutura relacional.

### **1.3 Objetivos**

#### **1.3.1. Objetivo Geral**

Desenvolver uma ferramenta capaz de auxiliar na criação da estrutura relacional de *Data Warehouses* e *Data Marts*.

#### **1.3.2. Objetivos Específicos**

- Facilitar a modelagem e criação de um *Data Warehouse*;
- Auxiliar os profissionais de DBA;
- Controlar a atualização dos dados;
- Minimizar o impacto na performance do servidor ao atualizar os dados.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Neste capítulo são apresentados os conceitos que conduzem o desenvolvimento desta monografia.

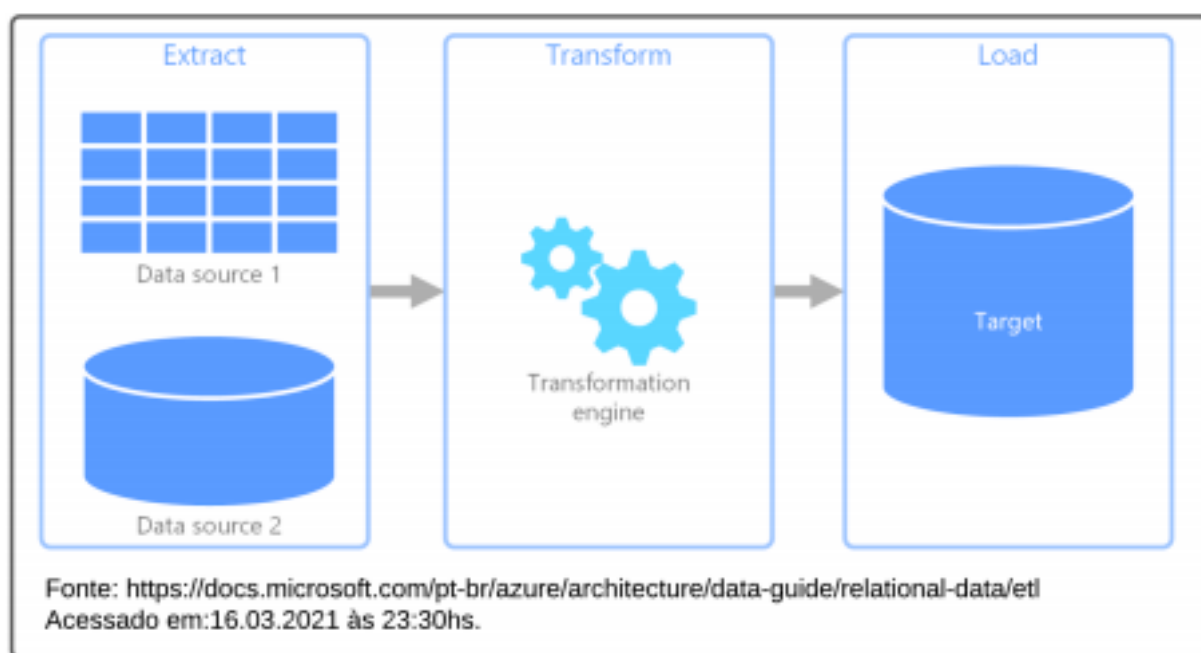
### **2.1 – Business Intelligence**

De acordo com PRIMAK (2008, p.02) “O termo Business intelligence surgiu na década de 80 no Gartner group e faz referência ao processo inteligente de coleta, organização,

11

análise, compartilhamento e monitoramento de dados contidos em *Data Warehouse* / *Data Mart*, gerando informações para o suporte a tomada de decisões no ambiente de Negócios”. Nesse processo de análise de dados são utilizadas diversas tecnologias, são elas: processo ETL

(*Extract Transform Load*) que consiste em na integração de dados em três etapas, a extração, que é a primeira tarefa a ser realizada e representa a obtenção de informações relevantes que pretende ser propagada para um data warehouse, a transformação, que constitui-se como a manipulação desses dados para que possam ser transformados para a readequação no banco de dados, e o carregamento (*Loading*) que é a inserção dos dados transformados em uma nova base de dados; o OLTP (*On-line Transaction Processing*) que utiliza o processamento de dados das rotinas que geram diariamente dados através de sistemas informatizados da organização dando suporte às funções de execução do negócio organizacional; o OLAP (*On-line Analytical Processing*) que consiste na capacidade de analisar grandes volumes de dados em várias visões dentro de um data warehouse; e os *Dashboards* que são visões gráficas e intuitivas que trazem informações relevantes para a análise da situação da organização como um todo ou setor específico. Esses *dashboards* são criados com os dados devidamente tratados, podendo ter em seu corpo, gráficos, planilhas, entre outros indicadores e suas métricas.



**Figura 1 - Modelo de ETL**

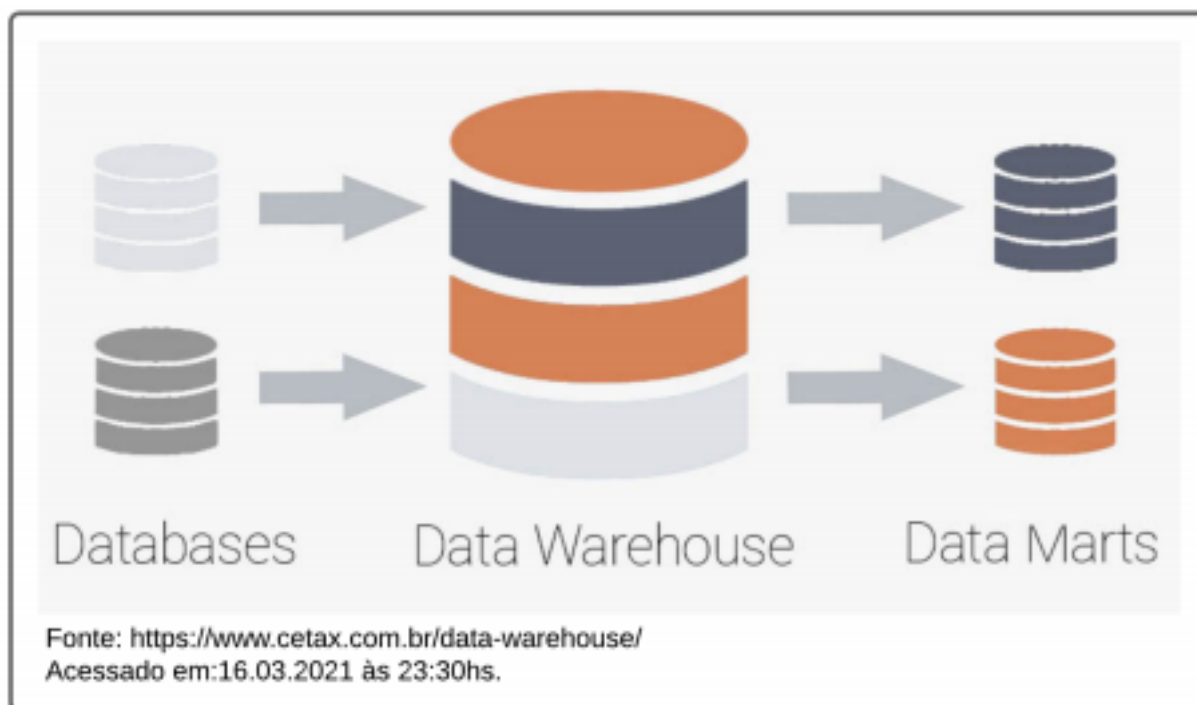
A Figura 1 apresenta o modelo visual de ETL, apresentando suas três etapas. Os objetivos do BI são diversos dentre eles podemos citar: o apoio a tomada de decisões gerenciais e administrativas, facilitar o acesso e compartilhamento de informações

dentro da organização, reduzir riscos e obstáculos, além de identificar oportunidades de vendas e obter informações valiosas sobre o comportamento dos seus clientes.

## 2.2 – Data Warehouse (DW) e Data Mart (DM)

*Data warehouse* “representa uma grande base de dados capaz de integrar de forma concisa e confiável, as informações de interesse para a empresa que se encontram espalhadas pelos sistemas operacionais e em fontes externas para posterior utilização nos sistemas de apoio à decisão” MACHADO (2013 p.22)

O objetivo do *data warehouse* é a centralização dos dados retirados de diversas fontes para assim facilitar a interpretação desses dados e a fim de criar e organizar relatórios, planilhas e gráficos por meio de históricos que ajudaram as organizações a obter *insights*<sup>1</sup> e auxílio à tomada de decisões importantes.



**Figura 2 - Modelo visual de um Data Warehouse e seus Data Marts.**

A Figura 2 apresenta a interação entre os bancos de dados principais com um *data warehouse* e sua estrutura composta por *datas marts*.

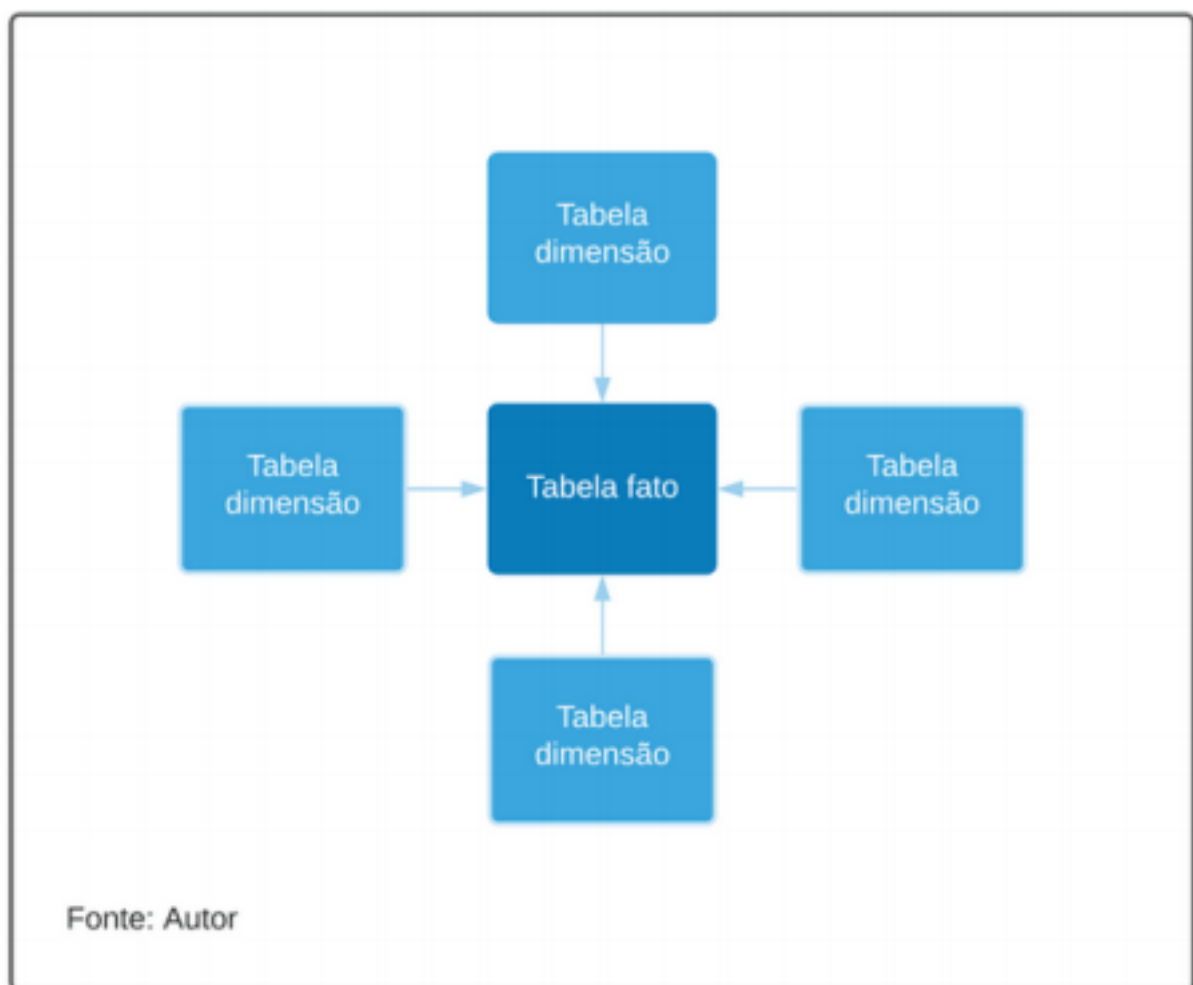
<sup>1</sup>*insights*: Insight é a compreensão de uma causa e efeito específicos dentro de um contexto particular.

As principais características de um *data warehouse* consistem na organização dos

dados, consistência, variedade de tempo e não-volatilidade. Seus dados são disponíveis apenas para leitura, assegurando a integridade do conteúdo alocado .

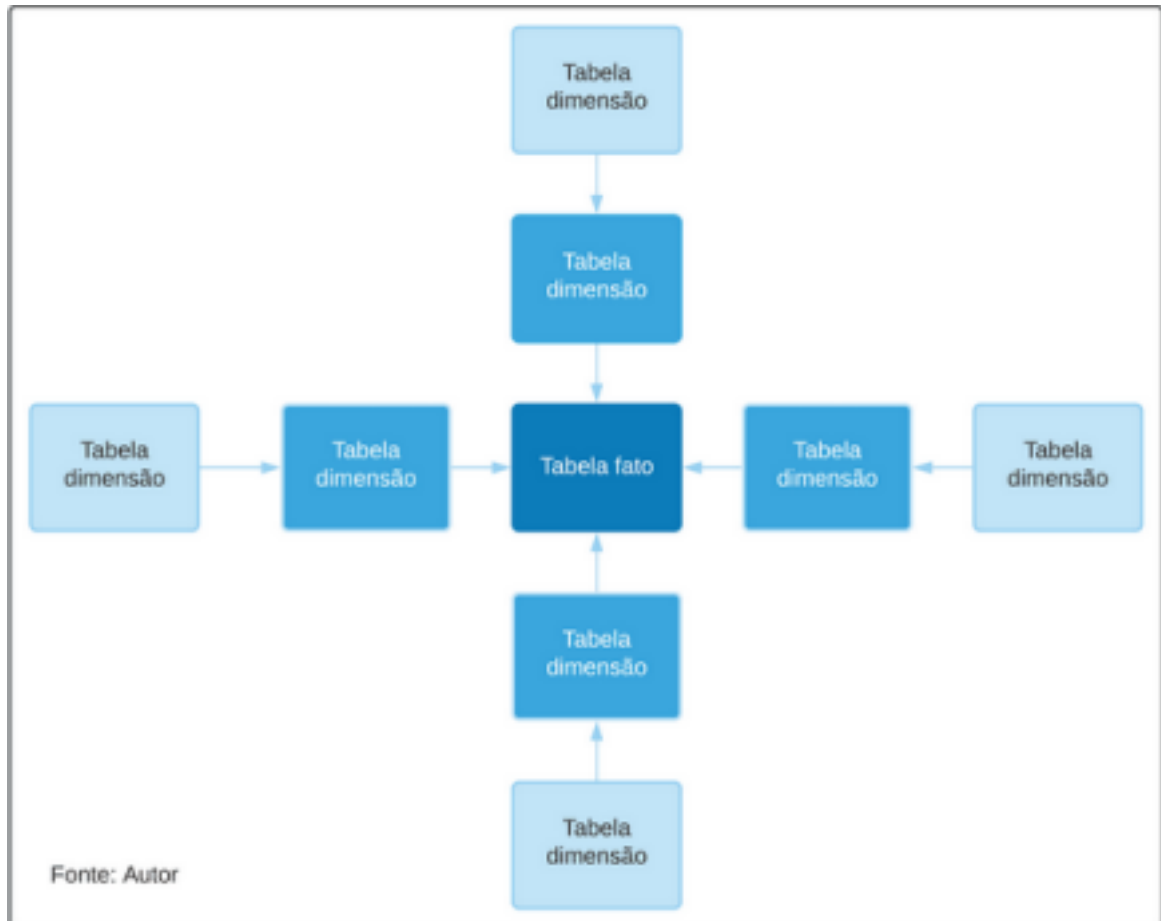
Os *data marts* são subconjuntos de dados de um *data warehouse*, como pequenas fatias que o compõem, normalmente organizados com a finalidade de auxiliar um departamento específico ou um processo de negócio.

Os *data warehouses* podem ser construídos utilizando a modelagem dimensional que engloba dois conceitos, são eles: o *star schema* ou esquema de estrela que consiste na técnica de Ralph Kimball, usando uma tabela principal chamada de tabela fato e outras tabelas ligada a tabela fato que são chamadas de tabelas dimensões, esse conceito possui somente duas camadas, conforme mostra a Figura 1; o *snowflake schema* ou esquema floco de neve é uma uma variação do esquema de estrela, também possui uma tabela principal chamada fato e uma camada de tabelas dimensões ligada a tabela fato, mas possuindo diversas tabelas dimensões ligadas entre si, podendo assim existir várias camadas.



**Figura 3 - Modelo Estrela**

Na Figura 3 é mostrado o modelo de estrela, nele pode haver várias tabelas, mas todas deverão estar ligadas com a tabela fato, e não possuindo redundâncias entre elas.



**Figura 4 - Modelo Floco de Neve**

Na Figura 4 é mostrado o modelo floco de neve, nesse modelo há várias tabelas e N dimensões, a primeira dimensão é ligada na tabela fato e as demais tabelas são ligadas em cadeia, o projeto utiliza esse modelo em suas construções.

### **2.3– Migração de Dados**

A migração de dados é uma metodologia que consiste em vários aspectos como, transferência de dados, mapeamento desses dados e a possibilidade de transformação desses dados entre a origem e o destino.

O processo de transferir dados de um sistema para outro é chamado de migração de dados, de acordo com a (IDC) *International Data Corporation*, a migração de dados faz

parte de 60% de todos os projetos de TI dentro de grandes organizações, a existência de

15

ferramentas de migração de dados possibilita ao profissional de TI gerenciar e acompanhar os dados migrados.

A ferramenta proposta neste trabalho usa da metodologia de migração de dados para a criação da estrutura relacional do *data warehouse* por meio da transferência das entidades e relacionamentos presentes no banco de dados principal, podendo haver a transferência completa da estrutura ou parcial.

## **2.4 – Atualização de Dados por Pacotes**

O método de atualização de dados por pacotes pode ser usado para diminuir a carga sobre o banco de dados fazendo com que os registros sejam limitados e ordenados para que ocorra várias transferências de dados pequenas de um servidor para outro, assim podendo evitar a perda de desempenho. Geralmente é utilizado quando há um grande volume de registro a ser transferido de uma base de dados para outra, a quantidade de registros por pacote enviado é definida de modo a manter o consumo de memória e processamento em níveis aceitáveis para o funcionamento dos demais sistemas presentes no cluster.

O modo de atualização por pacotes apresentado nesse projeto, consiste na ordenação dos registros de uma determinada tabela e após essa ordenação é feito a transferência limitando a quantidade de registros, esse limite é definido pelo usuário, após a transferência de um pacote e salvo o último valor da chave primária da tabela em questão, no próximo pacote será ordenado novamente e selecionado os registro que tenha a chave primária maior que a ultima salva no pacote anterior.

## **2.5 – Banco de Dados**

Os bancos de dados são definidos por ELMASRI e NAVATHE (2011, p.03) como: “uma coleção de dados relacionados. Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significado implícito. Por exemplo, considere os nomes, números de telefone e endereços das pessoas que você conhece. Você pode ter registrado esses dados em uma agenda ou, talvez, os tenha armazenado em um disco rígido, usando um computador

pessoal é um software como Microsoft Access ou Excel.

16

Essa coleção de dados relacionados, com um significado implícito, é um banco de dados”. Elmasri e Navathe também cita restrições ao termo banco de dados são ela: deve representar algum aspecto do mundo real chamado de mini-mundo ou de universo de discurso; seja uma coleção de dados logicamente coerente; e seja modelado, construído e populado com dados para alguma finalidade específica.

## **2.6 – Sistemas de Gerenciamento de Banco de Dados**

Os sistemas de gerenciamento de banco de dados (SGBD) são sistemas informatizados pelo qual é armazenado e manipulado os dados de modo que fiquem organizados, seguros e estruturados.

De acordo com ALVES (2014, p.17) “um SGBD deve ainda permitir que bancos de dados sejam excluídos ou que sua estrutura seja alterada, como adição de novas tabelas/arquivos. Alguns sistemas de bancos de dados são genéricos, ou seja, permitem que praticamente qualquer tipo de dado seja armazenado, uma vez que o próprio usuário define a estrutura do arquivo”.

Entre os SGBD’s “alguns possuem uma linguagem de programação própria, mas a maioria utiliza a já consagrada SQL (sigla de Structured Query Language, uma linguagem declarativa utilizada para manipulação de bancos de dados relacionais), que permite ao usuário criar rotinas específicas ou mesmo aplicativos completos” ALVES (2014, p17).

Esses sistemas funcionam como interfaces para a manipulação dos dados, existindo uma grande variedade deles, sendo eles: Oracle, Postgresql, MongoDB entre outros, sendo o foco desse projeto o Mysql, que é um SGBD de código aberto.

## **2.7 – Banco de Dados Relacional**

Existem diversos tipos de sistemas de banco de dados que possuem a mesma missão, que é o armazenamento de dados, mas de forma distintas, neste trabalho é abordado o modelo de banco de dados relacional. O modelo relacional incorpora o princípio da informação que é definido por DATE (2000, p.52) como: “todo conteúdo de informação do banco de dados é representado de um e somente de um modo, ou seja, como valores explícitos

em posições de colunas em linhas de tabelas”. Outra característica dos bancos de dados relacionais é possuírem cardinalidade que é definido por Elmasri e Navathe (2011, p.142),

17

como “o número máximo de instâncias de relacionamento em que uma entidade pode participar”, ou seja, a cardinalidade define a quantidade de itens que uma tabela no banco de dados pode ter relacionados com outra tabela.

Os bancos de dados relacionais geralmente utilizam a linguagem SQL que é a linguagem padrão para realizar queries, que possui um subconjunto de linguagens para poder realizar desde a criação de uma tabela até a inserção de um registro, essas sub linguagens são:

- (DML) Linguagem de manipulação de dados, encarregada de realizar as inclusões, alterações e exclusões dos registros dentro de uma tabela;
- (DDL) Linguagem de definição de dados que permite ao utilizador criar novas estruturas como tabelas, colunas e views;
- (DCL) Linguagem de controle de dados, encarregada das autorizações de usuário afim de definir quais usuários podem alterar determinadas estruturas do banco de dados;
- (DTL) Linguagem de transação de dados é usada para delimitar o começo e o fim de uma transação no banco de dados, podendo controlar a interrupção de uma transação caso necessário;
- (DQL) Linguagem de consulta de dados permite ao usuário consultar os dados salvos nas tabelas, essa linguagem possui apenas um comando, o select, e sendo ele o comando mais usado pelos DBA.

### **3 MATERIAIS E MÉTODOS**

O presente capítulo demonstra os materiais, métodos e técnicas para o desenvolvimento do sistema de informação proposto neste trabalho. A seção 3.1 apresenta o levantamento de requisitos, a seção 3.2 apresenta a modelagem do sistema, a seção 3.3 demonstra as tecnologias necessárias para o desenvolvimento da ferramenta.

#### **3.1 – Levantamento de Requisitos**

Os requisitos foram obtidos a partir de fontes com sites e livros junto com a ajuda



de pessoas da área de tecnologia da informação. Nessa seção são apresentados os requisitos funcionais e não funcionais requerido para que a ferramenta proposta no projeto cumpra o seu objetivo.

18

### 3.1.1 – Requisitos Funcionais

- Auxiliar na criação da estrutura de um *data warehouse*;
- Atualização periodicamente dos dados;
- Tela intuitiva para a criação da estrutura;
- Controle de desempenho na atualização dos dados;
- Usar bancos de dados relacionais;
- Conexão com *host*<sup>2</sup> e *server*<sup>3</sup>;

### 3.1.2 – Requisitos não Funcionais

- As solicitações serão feitas via a interface web;
- Interface com modo *dark*;

## 3.2 – Modelagem do Sistema

A modelagem do sistema consiste na criação de modelos que esclareçam o funcionamento do sistema, e sua estrutura, estes modelos são normalmente desenvolvidos utilizando notações gráficas, capazes de levar a percepção das funcionalidades existentes no sistema de forma simplificada, normalmente utilizando diagramas de casos de uso dentro da UML.

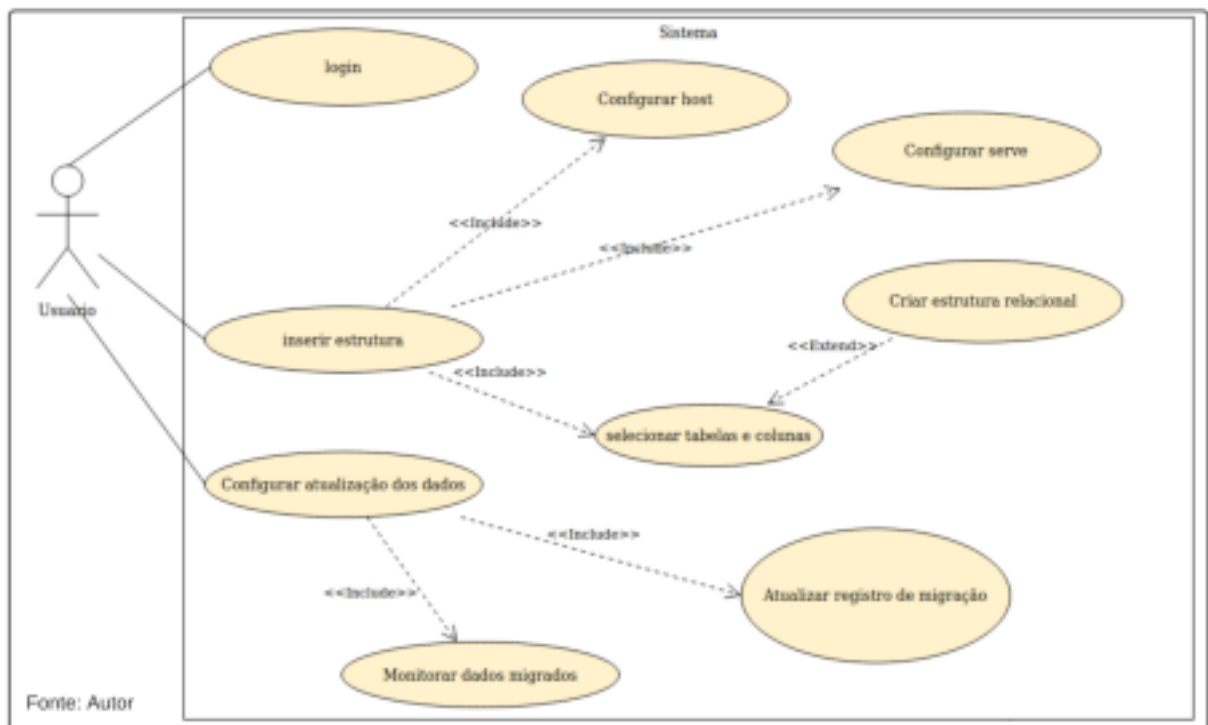
### 3.2.1 – Diagrama de Caso de Uso

De acordo com SCHAC (2016), o diagrama de caso de uso é a representação da interação de um agente com o sistema, que mostra a relação entre o usuário e os diferentes casos de uso, assim identificando diferentes tipos de usuários de um software, podendo haver casos de uso acompanhados por outros tipos de diagramas.

<sup>2</sup> *host*: Servidor hospedeiro

<sup>3</sup> *server*: Servidor receptor

19



**Figura 5 - Diagrama de Caso de Uso**

A Figura 5 apresenta o diagrama de caso de uso criado para exemplificação das funcionalidades presentes na ferramenta proposta neste projeto.

Com o auxílio do diagrama pode-se verificar as interações que o usuário tem ao utilizar a ferramenta. Após a efetuar o login o usuário entra na tela de inserção da estrutura do modelo para configurar a conexão com o *host* (banco de dados principal) e o *serve* (banco de dados criado para onde é migrado a estrutura e os dados do *data warehouse*), nessa tela também será feito a seleção das tabelas e colunas que serão usados, assim podendo criar a estrutura relacional. Em seguida, o usuário pode acessar outra tela para definir as configurações de atualização dos dados. Ao final, após a definição do registro de migração, o usuário pode monitorar o quantitativo de registros que foi migrado até o momento.

### 3.2.2 – Desenvolvimento ágil Kanban

De acordo com RODRIGUES (2008, p.42) “A metodologia ágil de desenvolvimento de software surgiu em contradição aos processos pesados de desenvolvimentos de software. Os métodos ágeis de desenvolvimento de software utilizam uma sistematização mais rápida e objetiva para se obter um software”.

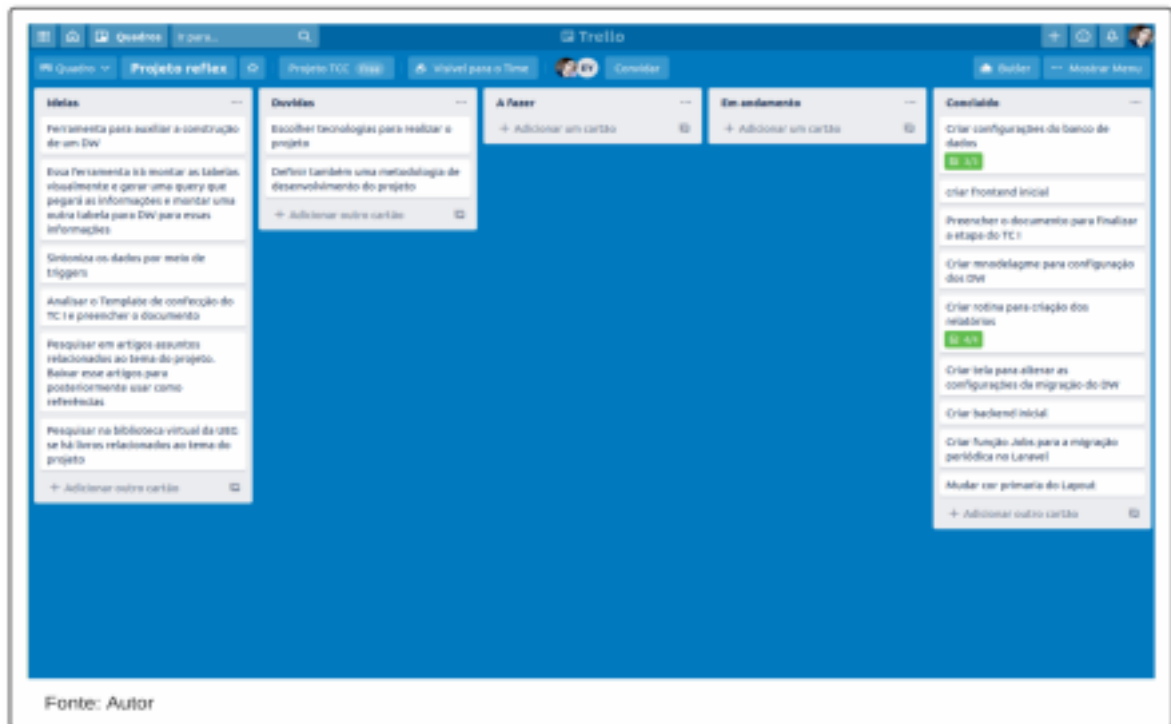
Existem diversos métodos de desenvolvimento ágeis como, *Scrum*, *eXtreme Programming* (XP), *Feature Driven Development* (FDD), *Microsoft Solutions Framework*

20

(MSF), *Dynamic System Development Model* (DSDM), *Lean* e o *Kanban* que foi o método utilizado para a construção e desenvolvimento desse projeto.

O *kanban* é um conceito de gerenciamento de projetos criado pela Toyota para, “gerencia o método JIT de produção. Em resumo, o sistema *kanban* é um sistema de informações que controla harmoniosamente as quantidades de produção em cada processo” MONDEN (2015, p.09). O modelo consiste na utilização de cartões e etapas, assim uma tarefa ou subtarefa passa por determinadas etapas até sua conclusão, essas etapas geralmente são compostas por: A fazer, onde estarão as tarefas que ainda não foram inicializadas; Em progresso, que ficam as tarefas que estão sendo feitas; e Concluída, onde estarão as tarefas que foram concluídas.

Na criação da ferramenta presente, foi utilizado o software Trello que faz o gerenciamento *Kanban* das tarefas e ideias no decorrer do desenvolvimento e construção do projeto. Essa ferramenta online possibilita que haja a interação de diversas pessoas envolvidas, trazendo clareza e objetividade ao projeto, podendo ser acessada de qualquer lugar.



**Figura 6 – Quadro Kanban**

Na Figura 6, pode se ver o quadro *Kanban* que foi utilizado para a organização das ideias e tarefas do projeto, também é possível acompanhar as etapas que foram percorridas por cada tarefa fazendo com que a entrega de diversos pacotes de atualizações

21

sejam frequentes, esse modelo de desenvolvimento é derivado da *continuous integration* (CI) ou integração contínua, possibilitada pelo uso desse método ágil.

### 3.3 – Tecnologias e Materiais

As tecnologias utilizadas para modelagem e construção dessa ferramenta apresentada no projeto, foram todas open source, o que significa que todas utilizam licenças de código aberto para seu uso. Os Materiais fazem referência à aparelhização informatizada que foi utilizada para a disponibilização, e alocação da ferramenta em ambiente de desenvolvimento e de produção.

#### 3.3.1 – IDE

As IDEs *Integrated Development Environment* do Inglês Ambiente de



funcionalidade que foi feita através dessa IDE foi o controle de versões pelo GIT, com a utilização do painel GIT é possível ver todos os arquivos que foram alterados e sua linha de tempo através dos *commits*, deixando assim o desenvolvedor situado de quais os arquivos foram alterados e quem os alterou.

### 3.3.2 – Modelagem do Banco de Dados

A modelagem do banco de dados é definida como “um detalhamento dos tipos de informações que devem ser armazenadas no banco de dados que se deseja projetar/construir. Esse detalhamento pode ser efetuado a partir de uma descrição textual ou por meio de gráficos. Esses métodos permitem a construção de um modelo de dados estruturado e

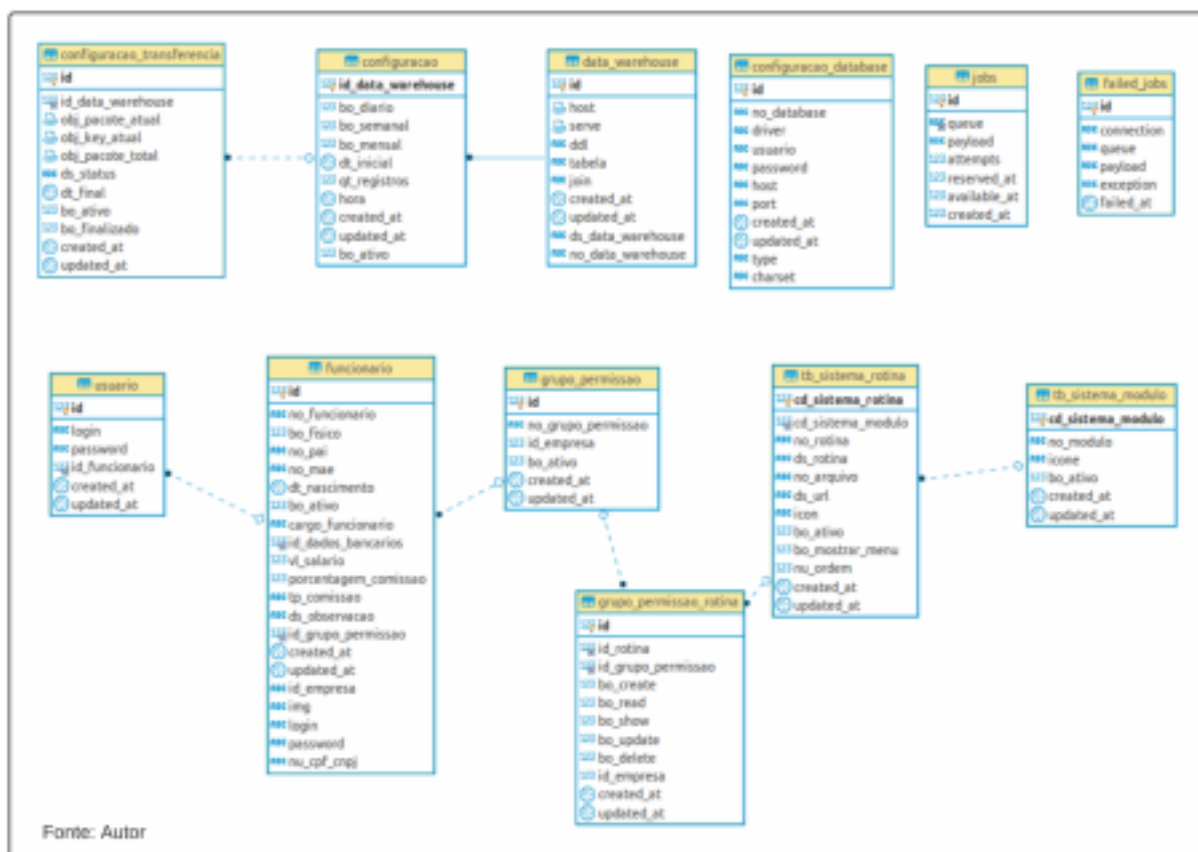
23

consistente, e são utilizados na representação desse modelo em níveis de abstração distintos” ALVES (2014, p.29).

O objetivo da criação dessa modelagem é permitir que diversos níveis de usuários tenham o entendimento adequado do desenvolvimento do banco de dados, para que isso ocorra existem níveis de abstração que podem ser divididos em modelo lógico e modelo conceitual.

O modelo conceitual é definido como “um conceito que descreve o que deve ser armazenado no banco de dados, independente do SGBD (Sistema Gerenciador de Banco de Dados) utilizado na implementação. Nele são representados os dados e suas relações que devem constar no banco de dados, sem se preocupar em descrever como eles devem ser armazenados” ALVES (2014, p.30).

O modelo lógico “possui um detalhamento do banco de dados próximo da visão de um especialista ou profissional de banco de dados. Sendo assim, ele depende do tipo de SGBD utilizado na implementação. Nesse modelo, as entidades representadas por retângulos no diagrama do modelo conceitual se tornam tabelas do banco de dados, com nome e definição das colunas que formam sua estrutura” ALVES (2014, p.30).



**Figura 8 - Modelagem do banco de dados da ferramenta REFLEX**

24

A Figura 8 apresenta o modelo lógico do banco de dados da ferramenta desenvolvida neste trabalho, esse modelo foi construído utilizando a ferramenta de modelagem Mysql Workbench. O modelo é constituído de 12 tabelas, podendo ser separadas em 3 campos, são eles: o campo de acesso, que consiste na parte de rotinas do sistema, usuário e permissão, são representadas pelas tabelas usuario, funcionario, grupo\_permissao, tb\_sistema\_rotina, tb\_sistema\_modulo e grupo\_permissao\_rotina; o campo de data warehouse, que é a parte de criação do data warehouse/mart e configuração de conexão com o host e o server, são compostos por duas tabelas, data\_warehouse, configuracao\_database; e o campo de transferência, que consiste na parte de atualização dos dados e controle de periodicidade da atualização, são compostos pelas tabelas configuracao\_transferencia, configuracao, jobs e failed\_jobs, sendo essas duas últimas tabelas criadas pelo framework Laravel para controle das queue.

### 3.3.3 – Frontend

O frontend de uma aplicação é a parte de interação com o usuário, atualmente à preocupação com a interação do usuário com a interface de um produto, colocando assim em destaque termos como: *UX User Experience*, que tem como objetivo melhorar a experiência de utilização de um produto, preocupando-se com cada etapa com a qual o usuário interage com o produto; *UI User Interface* que representa a interação do usuário com o produto, sendo assim responsável pela criação de interfaces funcionais e navegação intuitiva.

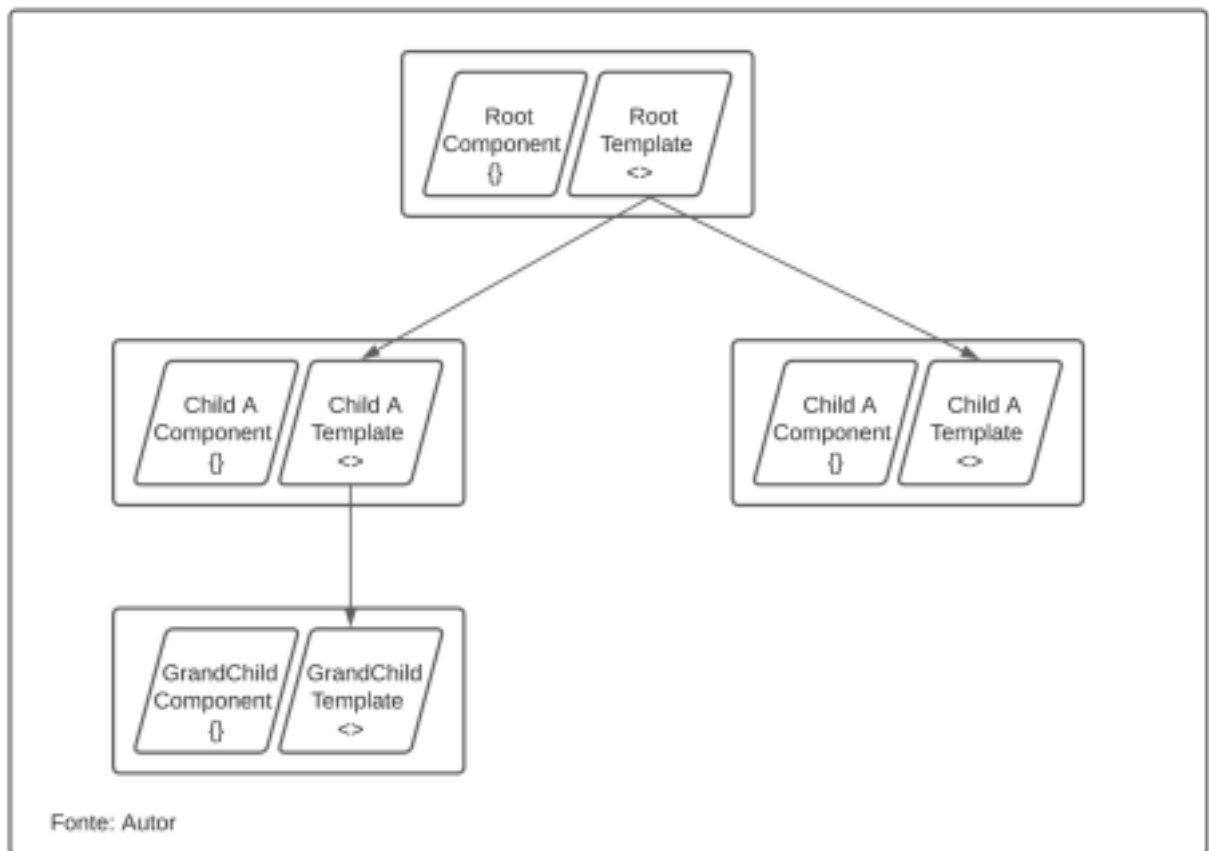
A ferramenta desenvolvida tem sua interface web construída usando o framework Angular, que possui características interessantes, como velocidade e desempenho escaláveis, reutilização de códigos para outras plataformas, produtividade, entre outras. O angular usa como linguagem padrão de desenvolvimento o typescript que possibilita a utilização da orientação a objeto em sua construção. No final o typescript é transpilado para javascript, que é a linguagem que os navegadores interpretam.

O Angular usa o método de componentes e templates em sua construção. Explicando de maneira simplificada, os componentes são os responsáveis por controlar os objetos e interações com a view, e os templates são as próprias view codificadas em HTML e CSS, permitindo interações com outros templates, esse modelo é mostrado na Figura 7, assim

25

uma tela pode ser constituída de vários conjuntos de componentes e templates, deixando cada um encarregado de uma função na composição total da *view*.





**Figura 9 - Hierarquia de componentes e templates no Angular**

A figura 9 mostra a hierarquia de componentes, a imagem é baseada na documentação online do Angular, essa hierarquia possibilita a construção fragmentada de uma tela, facilitando a manutenção e a correção de bugs no decorrer do desenvolvimento.

A estrutura de pastas usadas para constituir as view no frontend da ferramenta proposta no projeto é constituído pelos arquivos: CSS, que é a folha de estilização da view; o arquivo HTML que é composto pelas tags que criam a estrutura que compõem a view; o arquivo TS (*typescript*) que faz a função de controlador de interações da view, onde ficaram as funções chamadas pelo *onclick*, *onchange*, entre outras; o Module que é responsável pelos imports necessários para a constituição da *view*; o Service que faz a função de realizar as requisições no *backend*. Essa estrutura foi baseada na versão 8.0 do angular na qual o projeto foi construído.

26

### 3.3.4 – Backend

O *backend* da ferramenta REFLEX foi construído sobre o framework Laravel que utiliza a linguagem de programação PHP para sua construção. Um *backend* geralmente é onde

fica a regra de negócio de uma aplicação e onde se conecta diretamente ao banco de dados, portanto tende-se a ter uma segurança mais alta para que não ocorra inovações ao banco de dados e prejudique os dados contidos ali.

O Laravel usa a biblioteca Eloquent ORM que é um mapeador de objeto-relacional fazendo a interface da aplicação com o banco de dados ser mais agradável. Essa interface possui proteções contra ataques diretos ao banco de dados como SQL injection. Outro ganho ao se utilizar o Laravel é a praticidade de sua estrutura, podendo ser utilizada com diversos padrões como MVC, entre outros.

O framework permite a utilização tanto como plataforma web, podendo ser usado para regra de negócio e a renderização de views, como API HTTP na qual foi utilizada para a construção desse projeto, que é acessada por meio de requisições a partir do *client-side*.

As estruturas dos *data warehouse* foram criadas utilizando a classe Schema do Laravel, que permite de maneira agnóstica a manipulação da estrutura do banco de dados que estiver conectado, assim possibilitando a criação de tabelas, colunas e suas ligações.

O Laravel também conta com a criação de queue ou jobs, que são tarefas que podem ser executadas paralelamente e programáveis, com isso foi possível realizar a atualização dos dados assíncrona de uma base de dados.

### 3.3.5 – Funcionamento

Para que a ferramenta funcione é necessário usar um servidor HTTP, para os testes foi utilizado o Apache HTTP Server na sua versão 2.4, deve-se configurar duas rotas, uma rota para acessar o *backend* e outra para o *frontend*, assim é possível ter acesso a ferramenta e criar o *data warehouse*.

Após o usuário efetuar login, aparecerá a tela de seleção de tabelas, nessa tela é possível configurar a conexão do banco de dados principal, conexão do banco de dados onde será criado a estrutura do *data warehouse* e selecionar as tabelas e colunas presentes na modelagem. Ao gerar o *data warehouse* dois algoritmos fazem a criação das tabelas e a criação das ligações dessas tabelas, e por fim é criado um registro de configuração vazio.

27

Na tela de configuração pode ser selecionado a periodicidade, horário da atualização e configurar a quantidade de item por pacote que o sistema irá enviar, fazendo assim com que a carga de desempenho das atualizações sejam controladas a fim de não

prejudicar os demais sistemas presentes no cluster.

O monitoramento da quantidade de registros migrados também ocorre na tela de configuração, podendo ver quantos registros existem e quantos já foram enviados, além de apresentar esse quantitativo em porcentagem. O usuário pode alterar o período conforme desejado, podendo até mesmo interromper as atualizações de dados.

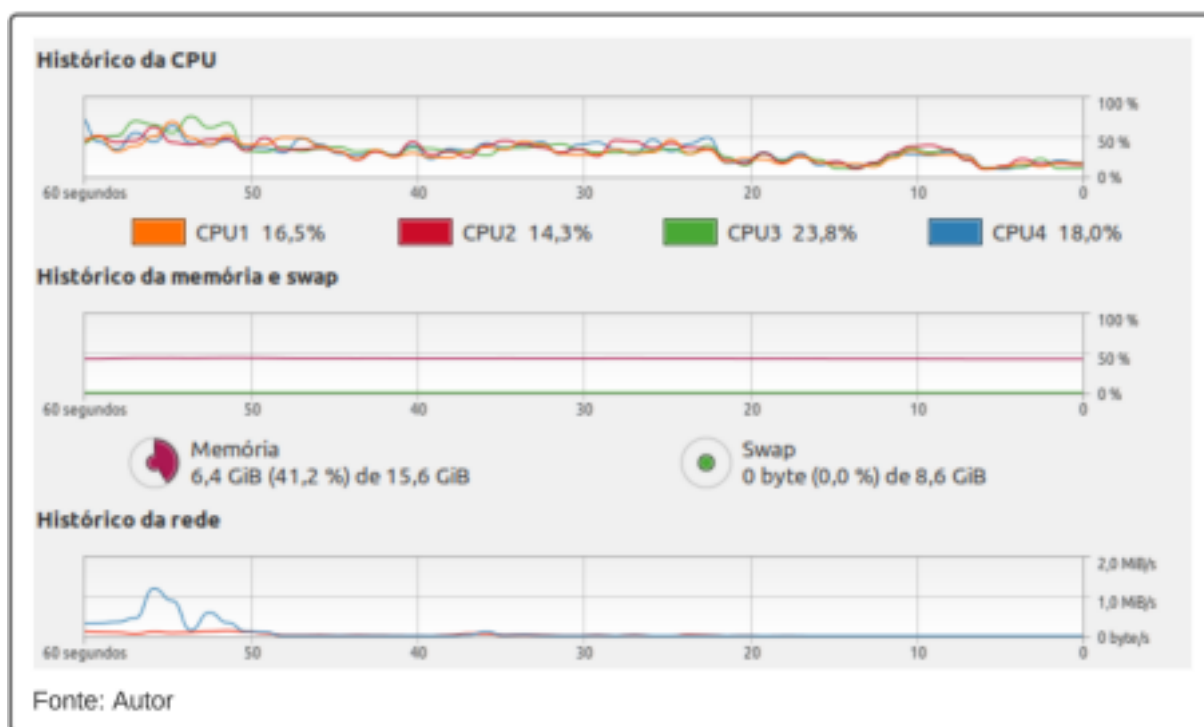
## 4 CONCLUSÃO

Neste capítulo são descritos os resultados obtidos e sugestões de trabalhos

### futuros. 4.1 – Resultados obtidos

Atualmente para a criação de um data warehouse ou data mart o profissional de banco de dados modela o *data warehouse* e cria esse modelo no banco de dados onde será implementado essa aplicação, após a criação o profissional deve considerar a utilização de alguma ferramenta para a atualização desses dados. O intuito da ferramenta proposta neste trabalho é facilitar a parte da criação da estrutura após a modelagem dos dados e auxiliar a atualização periódica dos dados da fonte principal para o data warehouse/mart.

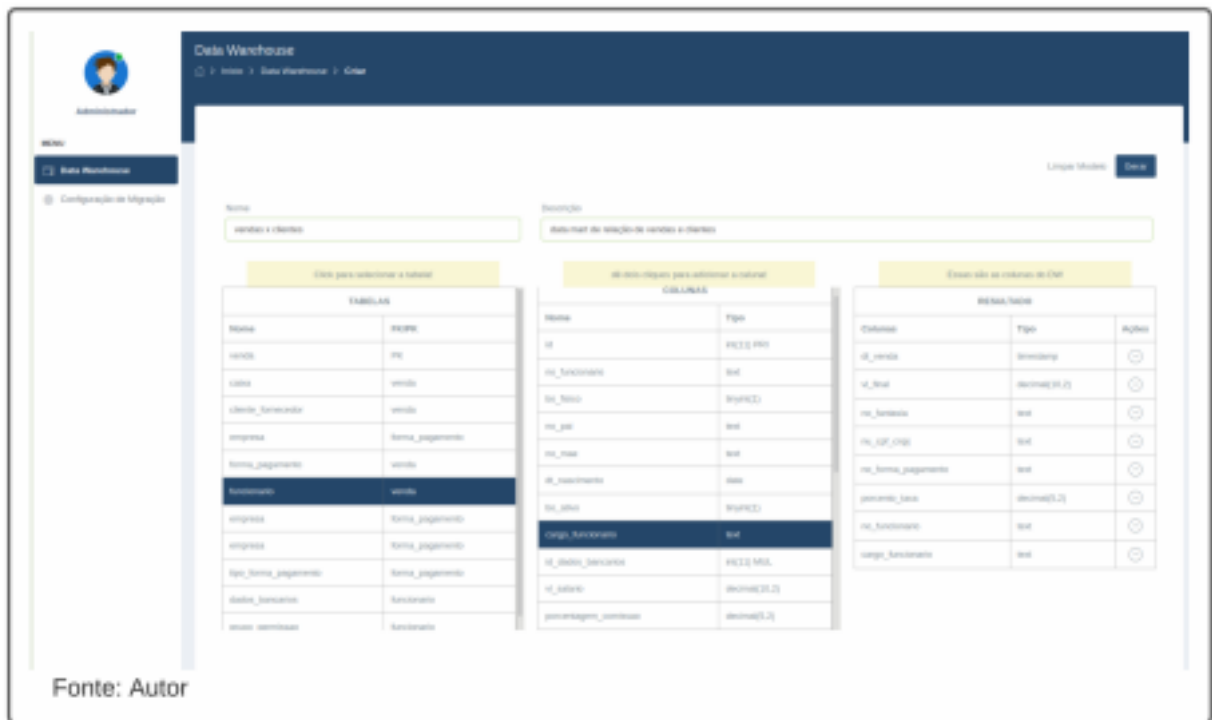
O teste principal para verificar o desempenho e a integridade da estrutura foi feito usando uma modelagem para a criação de um *data warehouse* sobre vendas, nessa modelagem retirada do banco de dados do sistema comercial SIAGESC, contia 5 tabelas, sendo elas, venda, cliente, forma\_pagamento, funcionario e nota, contabilizando cerca de 12000 registros, o teste foi feito utilizando um servidor local, com as configurações, 16 gigabytes de memória RAM quad Core com o sistema operacional linux. A estrutura foi criada buscando as cardinalidades no banco de dados principal, então obtendo êxito total em sua criação, migrando todas as primary key e foreign key dentre o conjunto de tabelas.



**Figura 10 - Monitor de recursos do linux**

Na Figura 10 é possível visualizar o desempenho do processador e memória RAM ao realizar a criação e a transferência de dados do teste principal, o desempenho foi verificado através do painel do monitoramento de sistema do linux que possibilitou obter as métricas de uso da CPU e memória RAM, no sistema foram configurados 500 registros por pacotes. antes de rodar a atualização dos dados os valores de memória estavam em 5.8 gigabytes, sendo usado pelo sistema e outras aplicações, a utilização da CPU estavam entre 10 por cento de uso, sendo assim o desempenho a mais causado ao rodar a aplicação nessas circunstância foi aceitável.

Os testes feitos obtiveram sucesso na criação da estrutura relacional, com todas as tabelas e colunas selecionadas, importando as cardinalidade no banco principal, a fim de preservar a integridade dos dados, usando um único tipo de SGBD o Mysql, a atualização desses dados por meio de pacotes que são configurados na tela de configuração de atualização da ferramenta também resultaram em sucesso, o estresse causado na máquina pela transferência desses dados pode ser controlado configurando a quantidade de registros por pacotes.



**Figura 11 - Tela de seleção da estrutura do data warehouse/mart**

A Figura 11 apresenta a tela de seleção de tabelas e colunas que irá existir na estrutura do data warehouse/mart, essa estrutura é derivada de um banco de dados principal, o layout da tela basicamente em 4 partes, o cabeçalho onde colocara o nome e descrição do data warehouse, a seção TABELAS para a seleção das tabelas, a seção COLUNAS para a seleção das colunas presentes na tabela selecionada e a seção RESULTADO que apresenta o conjunto de colunas selecionadas.

A ferramenta é recomendada para ser utilizada na criação de pequenos e simples data warehouse ou data mart, que utilizem um tipo de SGBD, no caso apresentado foi o Mysql, para a criação de data warehouses que tenham sua estrutura diferente que não compoñham a base principal não é recomendado a utilização desta aplicação, pois a mesma busca se basear na estrutura do banco de dados principal para compor o princípio da integridade dos dados.

## 4.2 – Trabalhos futuros

Como forma de melhorar a ferramenta proposta neste projeto, podemos sugerir atualizações e implementações futuras como: Integrar a ferramenta com outros SGBD permitindo a criação de data warehouse/mart entre SGBD diferentes; Alterar o nome das

tabelas na estrutura final do data warehouse, podendo assim se adequar a modelagens mais

30

avanzadas; Permitir a inserção e manipulação de arquivos (planilhas, csv) para a composição das bases de dados criadas.

31

## REFERÊNCIAS

PRIMAK, F. V.. Decisões com BI (Business Intelligence). Brasil, Ciência Moderna, 2008 152p.

NAVATHE, SHAMKANT B., Elmasri R.. Sistemas de banco de dados: fundamentos e aplicações. Brasil, LTC, 2002.

DATE, C. J.. INTRODUÇÃO A SISTEMAS DE BANCOS DE DADOS. 8. ed. Rio de Janeiro: Elsevier, 2003.

SCHAC, S. R. Engenharia de Software: os paradigmas clássico e orientado a objetos. 7. ed. Porto Alegre: Bookman, 2016.

RODRIGUES, E. J. L.. Curso de Engenharia de Software. ed. Universo dos Livros Editora, 2008.

MONDEN, Y.. Sistema Toyota de Produção: Uma Abordagem Integrada ao Just-in-time. 4. ed. Porto Alegre : Bookman, 2015

ALVES, W. P.. Banco de Dados. 1. ed. São Paulo : Érica, 2014

MACHADO, F. N. R.. Tecnologia e projeto de Data Warehouse. 6 ed. Érica, 2013.

MICROSOFT. ETL (extrair, transformar e carregar) Disponível em: <https://docs.microsoft.com/pt-br/azure/architecture/data-guide/relational-data/etl>. Acesso em: 16/03/2021.

CETAX. Data Warehouse: Tudo o que você precisa saber! Disponível em: <https://www.cetax.com.br/data-warehouse/>. Acesso em: 16/03/2021