



UNIVERSIDADE DO SUL DE SANTA CATARINA

ISABELE AURORA CÂNDIDO VITORINO RAU

DATA LAKE: UMA NOVA ABORDAGEM PARA O ARMAZENAMENTO DE DADOS

Florianópolis

2021

ISABELE AURORA CÂNDIDO VITORINO RAU

DATA LAKE: UMA NOVA ABORDAGEM PARA O ARMAZENAMENTO DE DADOS

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de informação da Universidade do Sul de Santa Catarina como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Daniella Vieira, MEng.

Dedico este projeto à minha família e amigos que sempre estiveram presentes direta ou indiretamente em todos os momentos de minha formação.

AGRADECIMENTOS

Em primeiro lugar quero agradecer ao meu avô, Osvaldo, que sempre me incentivou os estudos, me deu suporte e me ensinou a ser uma pessoa correta e íntegra. E também a minha avó, Iva, que com certeza me ajudou lá de cima, nessa reta final!!

Ao meu marido, meu companheiro de vida, que esteve ao meu lado em todos os momentos ao longo desses anos de faculdade e, ingresso no mercado de trabalho. Fazendo questão de sempre me deixar segura em quaisquer decisões que eu precisasse tomar e, lembrando que o tempo de bonança iria chegar.

À minha irmã que me incentivou diversas vezes ao longo dessa jornada acadêmica, durante muitas noites. Ao meu tio Rodrigo, que me inspirou e direcionou minha carreira e, contribuiu muito para a ampliação dos meus conhecimentos.

À minha orientadora Daniella, que é uma figura inspiradora, exemplo vivo de que planejamento e execução nos levam aos caminhos que desejamos para nós. Ela foi fundamental desde o início deste projeto, tanto na idealização do tema como no desenvolvimento, tinha sempre uma visão mais realista e um olhar mais preciso. Tive o prazer de ser sua aluna em duas cadeiras neste curso e sua orientanda, podendo aprender além dos meios acadêmicos. Foi ela quem me ensinou que tudo na vida é uma questão de prioridade. Minha eterna gratidão e admiração à Prof. Daniella Vieira.

Ao meu colega, Marcus Bittencourt, que por algumas vezes dividiu comigo seus conhecimentos sobre a temática abordada neste trabalho.

Enfim, à minha família e amigos tão especiais que fizeram parte direta e indiretamente deste projeto, minha eterna gratidão, levarei todos em meu coração!

RESUMO

Com a evolução dos sistemas e a produção massiva de dados por parte destes, nasceu um novo mercado, de dados. Tendo estes se tornados bens valiosos para as organizações e podendo serem convertidos em vantagens competitivas para as mesmas. Sucintamente, esse foi um dos motivos para o surgimento da Era do *Big Data*. Entretanto, as abordagens tradicionais de armazenamento de dados, mostraram limitações ao lidar com altos volumes de dados. Desse modo, surgiu a necessidade de tecnologias mais aprimoradas para armazenar e processar dados. Uma das estratégias criadas para lidar com o *Big Data* foi o *Data Lake*. Essa terminologia é abordada no decorrer do presente trabalho, assim como o conceito de *Data Warehouse*. São explorados os conceitos de ambas tecnologias, com o intuito de elucidar suas diferenças e semelhanças e o melhor cenário para aplicá-las. Também são apresentados dois conjuntos de *software* capazes de auxiliar na implementação de um ambiente de *Data Lake*. Sendo também demonstrado, através de um experimento, a estruturação de um ambiente de *Data Lake* utilizando um destes conjuntos de *software*. Através destes insumos é possível detectar quais aspectos devem ser considerados pelas empresas na hora de escolher o tipo de armazenamento de dados que faz mais sentido para o seu negócio.

Palavras-chave: Data Lake. Apache Hadoop. Amazon AWS.

ABSTRACT

With the evolution of systems and the massive production of data by them, a new data market was born. These have become valuable assets for organizations and can be converted into competitive advantages for them. Briefly, this was one of the reasons for the emergence of the Age of Big Data. However, traditional approaches to data storage have shown limitations when dealing with high volumes of data. Thus, the need for more surgiuimproved technologies to store and process data emerged. One of the strategies created to deal with Big Data was the Data Lake. This terminology is discussed throughout this work, as well as the concept of Data Warehouse. The concepts of both technologies are explored in order to elucidate their differences and similarities and the best scenario to apply them. Also presented are two sets of software capable of assisting in the implementation of a Data Lake environment. It is also demonstrated, through an experiment, the structuring of a Data Lake environment using one of these software sets. Through these inputs, it is possible to detect which aspects should be considered by companies when choosing the type of data storage that makes the most sense for your business.

Keywords: Data Lake. Apache Hadoop. Amazon AWS.

LISTA DE ILUSTRAÇÕES

Figura 1 – Resultado da pesquisa de áreas temáticas para 'data lake' e 'big data'	17
Figura 2 – Publicações com referências à data lake e big data	17
Figura 3 – Exemplificação de tipos de dados	20
Figura 4 – Representação simplificada de um sistema de banco de dados relacional	22
Figura 5 – Representação de um sistema de banco de dados relacional	24
Figura 6 – Exemplo de banco de dados chave-valor	25
Figura 7 – Exemplo de banco de dados orientado à documentos	25
Figura 8 – Exemplo de banco de dados orientado à colunas	26
Figura 9 – Exemplo de um grafo simples	26
Figura 10 – Exemplo de orientação por assunto	29
Figura 11 – Exemplo de visão integrada do DW	29
Figura 12 – Exemplo de não volatilidade do DW	30
Figura 13 – Exemplificação de variantes do tempo	30
Figura 14 – Estágios para implementação de um Data Lake	33
Figura 15 – Atividades metodológicas	38
Figura 16 – Ecossistema Hadoop	39
Figura 17 – Arquitetura base do HDFS	40
Figura 18 – Aplicações que compõem o ecossistema Hadoop	41
Figura 19 – Ecossistema AWS	43
Figura 20 – Arquitetura da solução proposta	47
Figura 21 – Parte 1 da criação de conta na AWS	48
Figura 22 – Parte 2 de criação de conta na AWS	49
Figura 23 – Parte 3 da criação de conta na AWS	50
Figura 24 – Painel de gerenciamento de serviços da AWS	51
Figura 25 – Painel e criação de buckets do S3 da AWS	52
Figura 26 – Visualização da estrutura de diretórios no bucket do S3 da AWS	53
Figura 27 – Adição de banco de dados no Glue da AWS	53
Figura 28 – Parte 1 de criação de crawler no Glue da AWS	54
Figura 29 – Parte 2 de criação de crawler no Glue da AWS	55
Figura 30 – Parte 3 de criação de crawler no Glue da AWS	56
Figura 31 – Parte 1 da criação de um job no Glue na AWS	57
Figura 32 – Parte 2 da criação de um job no Glue na AWS	58
Figura 33 – Painel de trabalho e roteiro no Glue na AWS	59
Figura 34 – Parte 1 da criação de um gatilho (trigger) no Glue na AWS	60
Figura 35 – Parte 2 da criação de um gatilho (trigger) no Glue na AWS	60
Figura 36 – Exibição de consulta no Athena na AWS	61
Figura 37 – Painéis gerados através do QuickSight da AWS	62

Figura 38 – Resumo dos 5 V's do Big Data	70
--	----

LISTA DE GRÁFICOS

Gráfico 1 – Volume de dados/informações criados em todo o mundo de 2010 a 2024 . . .	16
--	----

LISTA DE TABELAS

Tabela 1 – Comparativo entre os conceitos ACID e BASE	27
Tabela 2 – Comparativo entre Sistemas transacionais e Sistemas de apoio à decisão . .	28
Tabela 3 – Comparativo entre Data Warehouse e Data Lake	34
Tabela 4 – Equivalências das ferramentas do Hadoop e AWS	46

LISTA DE ABREVIATURAS E SIGLAS

ABES	Associação Brasileira das Empresas de Software
ABNT	Associação Brasileira de Normas Técnicas
ACID	Atomicidade, Consistência, Isolamento e Durabilidade
BI	Business Intelligence
CAP	Consistency, Availability and Partition Tolerance
COMMIT	Envio do código modificado pelo desenvolvedor ao servidor de controle de versão
CTO	Diretor Técnico (Chief Technical Officer)
DL	Data Lake
DW	Data Warehouse
ER	Entidade-relacionamento (entity-relationship)
ETL	Extract, Transform and Load
GB	Gigabyte
IBM	International Business Machines
IDC	International Data Corporation
OLAP	Online Analytical Processing
SAD	Sistema de apoio a Decisão
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Structured Query Language
TCC	Trabalho de Conclusão de Curso
TI	Tecnologia da Informação
XXI	Século 21

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Problemática	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	Justificativa	15
1.4	Estrutura da monografia	18
2	CONCEITOS FUNDAMENTAIS	19
2.1	Dados, informação e conhecimento	19
2.1.1	Tipos de dados	20
2.2	Banco de dados	21
2.2.1	Banco de dados relacionais	23
2.2.2	Banco de dados não relacionais	24
2.3	Data Warehouse	27
2.4	Data Lake	32
2.5	Comparativo entre as tecnologias	33
3	MÉTODO DE PESQUISA	37
3.1	Classificação da pesquisa	37
3.2	Atividades metodológicas	38
3.3	Delimitações	38
4	FERRAMENTAS PARA A CRIAÇÃO DE UM DL	39
4.1	Apache Hadoop	39
4.2	Amazon AWS	42
4.2.1	Amazon S3	43
4.2.2	AWS Glue	44
4.2.3	Amazon Athena	45
4.3	Equivalência entre as ferramentas	45
5	ESTRUTURAÇÃO DE UM AMBIENTE DE DATA LAKE NA NUVEM DA AMAZON	47
5.1	Cenário	47
5.1.1	Criação de conta na Amazon	48
5.1.2	Configuração do Amazon S3	50
5.1.3	Configuração do AWS Glue	53
5.1.4	Consultas através do Athena	61

6	CONCLUSÃO	63
	REFERÊNCIAS	65
	APÊNDICES	69
	APÊNDICE A	70

1 INTRODUÇÃO

Com o advento de novas tecnologias e com a difusão de novos meios de comunicação observa-se que a geração de dados cresce de maneira exponencial (SOMASUNDARAM; SHRI-VASTAVA, 2011). Considerando que todo dispositivo eletrônico e *software* que ele manipule gera dados constantemente, estima-se atualmente que o volume de dados gerados no mundo esteja na casa dos *zettabytes*¹ (STATISTA, 2020).

Segundo Moresi (2000), há um consenso de que na sociedade pós-industrial a informação passou a ser considerada um capital precioso equiparando-se aos recursos de produção, material e financeiro. Diante disso, a exploração dos dados gerados pelas organizações mostrou-se muito atrativa. Neste contexto, nasceu a necessidade de dispor de tecnologias que fossem capazes de armazenar, gerenciar e manipular essa massiva quantidade de dados voláteis. Dentre as tecnologias disponíveis para compor repositórios de Big Data,² podem ser citadas: os *Data Warehouses* (DW) e *Data Lakes* (DL). Sabe-se que DW e DL tratam de soluções distintas e têm finalidades diferentes. Contudo, muitas empresas usam estas duas soluções para atender a necessidades específicas e alcançar determinadas metas.

Estudos da década de 90 estimavam que por volta de 2000, o mercado de '*Data warehousing*' seria multibilionário e que cresceria a uma taxa de 20% ao ano no século XXI. Consideravam ainda que 1 terabyte (TB) teria o custo de implantação de US\$3 milhões levando até dois anos para ser implementado (GRAY; ISRAEL, 1999). Contudo, de acordo com o *International Data Corporation - IDC*, o mercado de dados atingiu a marca de US\$203 bilhões em 2020 tendo uma taxa de crescimento de 11,7% ao ano (INTERNATIONAL DATA CORPORATION, 2020).

O uso de *Data Warehouses* são comprovadamente adequados em processos de tomada de decisão. Os dados produzidos nas organizações depois de armazenados nos seus *Data Warehouses*, constituem a base e a principal fonte de informação para os seus agentes de decisão (MORGADO, 2013). Todavia, apesar de os *Data Warehouses* convencionais serem capazes de lidar com vários tipos e formatos de dados, não são capazes de guardar e processar dados espaciais, não sendo, assim, possível, em alguns cenários aplicativos, retirar uma parte importante do conhecimento contido na informação recolhida pelas organizações (MORGADO, 2013).

Em 2010 os conceitos de *Data Lake* ou *Data Hubs* surgiram no mercado sendo este termo introduzido por James Dixon (DIXON, 2010). Um *Data Lake* se refere ao armazenamento massivo e escalável que suporta dados em seu formato nativo (como é) até que seja necessário processá-lo para que se possa usá-lo sem comprometer a estrutura do dado. Os *Data Lakes* são construídos para lidar com uma grande quantidade de dados não estruturados, em contraponto aos *Data Warehouses* que processam dados estruturados. Além disso, os *Data Lakes* são dinâmicos,

¹ 1 zettabyte equivalente à aproximadamente 10^{21} bytes.

² Descrito no Apêndice A

ao contrário dos *Data Warehouses*. Os dados em um *Data Lake* também são acessíveis no momento em que são criados, em contraponto aos *Data Warehouses* projetados para processar os dados a medida que são necessários (MILOSLAVSKAYA; TOLSTOY, 2016a).

Diante do exposto, o presente trabalho tem como propósito, inicialmente, trazer os princípios conceituais sobre a temática de DW, visando demonstrar as diferenças entre DW e DL. Esta contribuição se apresenta necessária para que se compreenda em quais cenários de aplicação elas devem ser utilizadas. Desta forma, a principal contribuição do presente trabalho é demonstrar os princípios arquiteturais para a formação de um DL.

1.1 Problemática

De acordo com a definição clássica de Laudon e Laudon (1999), Sistemas de Informação são definidos como: “um conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações, com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em organizações”. Corroborando esta definição O’Brien (2004) define sistemas de informação como: “Um conjunto organizado de pessoas, *hardware*, *software*, redes de comunicações e recursos de dados que coleta, transforma e dissemina informações em uma organização”.

Considerando as definições descritas, no âmbito corporativo, sabe-se que existem diversos sistemas de informação que atendem os mais variados objetivos. Nesse contexto muitos sistemas desenvolvidos por fornecedores distintos são construídos sob arquiteturas de informação nem sempre integradas. Contudo, tais sistemas manipulam dados em diversas grandezas.

As demandas por informações transparentes e transversais aos sistemas é crescente. Assim, arquiteturas de informação foram desenvolvidas para atender a estes cenários e compreender como e quando aplicá-las é essencial para fazer um bom uso da tecnologia. Por isso, o presente trabalho tem como propósito trazer os princípios conceituais sobre a temática de DW e DL de modo que se compreenda a diferença entre essas soluções, bem como demonstrar os princípios arquiteturais para a formação de um DL.

Conforme citado anteriormente, *Data Warehouse* é compreendido como depósito de dados que armazena as informações de empresas de forma consolidada. Com base nos dados produzidos a partir de uma base confiável é possível tomar decisões gerenciais assertivas amparadas por informações sistematizadas. Já o *Data Lake* é compreendido como um repositório que centraliza e armazena todos os tipos de dados gerados pela e para a empresa, sendo eles estruturados ou não. Eles são depositados em estado bruto, sem processamento e análise e até mesmo sem uma governança. A ideia é manter na organização dados que podem ser estrategicamente úteis, mesmo que eles, na realidade, não sejam requeridos em nenhum momento posterior. Dessa maneira o *Data Lake* seria o ambiente de armazenamento dessas informações.

De acordo com Khine e Wang (2018), se *Data Warehouse* é uma garrafa d’água pronta para o consumo, um *Data Lake* é um lago inteiro. Esta nova arquitetura é necessária para a

era do *Big Data* que requer técnicas e métodos adequados para armazenar e processar grandes volumes de dados. Contudo, nem todas as organizações demandam por tal modelo arquitetural. Neste sentido, existe no mercado uma percepção equivocada, muitas vezes, sobre o conceito e a aplicabilidade de cada uma destas arquiteturas.

Assim, o problema de pesquisa busca responder no âmbito teórico às seguintes questões: (1) Qual a diferença conceitual entre DW e DL?; (2) Em que contexto devem ser utilizadas as arquiteturas de *Data Warehouse* ou *Data Lake*?. Contudo, tendo em vista que o objetivo principal trata de um estudo mais aprofundado sobre os *Data Lakes*, a maior contribuição da pesquisa aplica-se à seguinte pergunta: (3) Quais as etapas necessárias para a criação de um *Data Lake*?

1.2 Objetivos

Nesta seção serão apresentados os objetivos geral e específico desta monografia.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é investigar em que contexto devem ser utilizadas as arquiteturas de *Data Warehouse* e *Data Lake*, demonstrando como a tecnologia de DL pode ser aplicada no âmbito corporativo, tendo em vista que este é o maior desafio tecnológico.

1.2.2 Objetivos Específicos

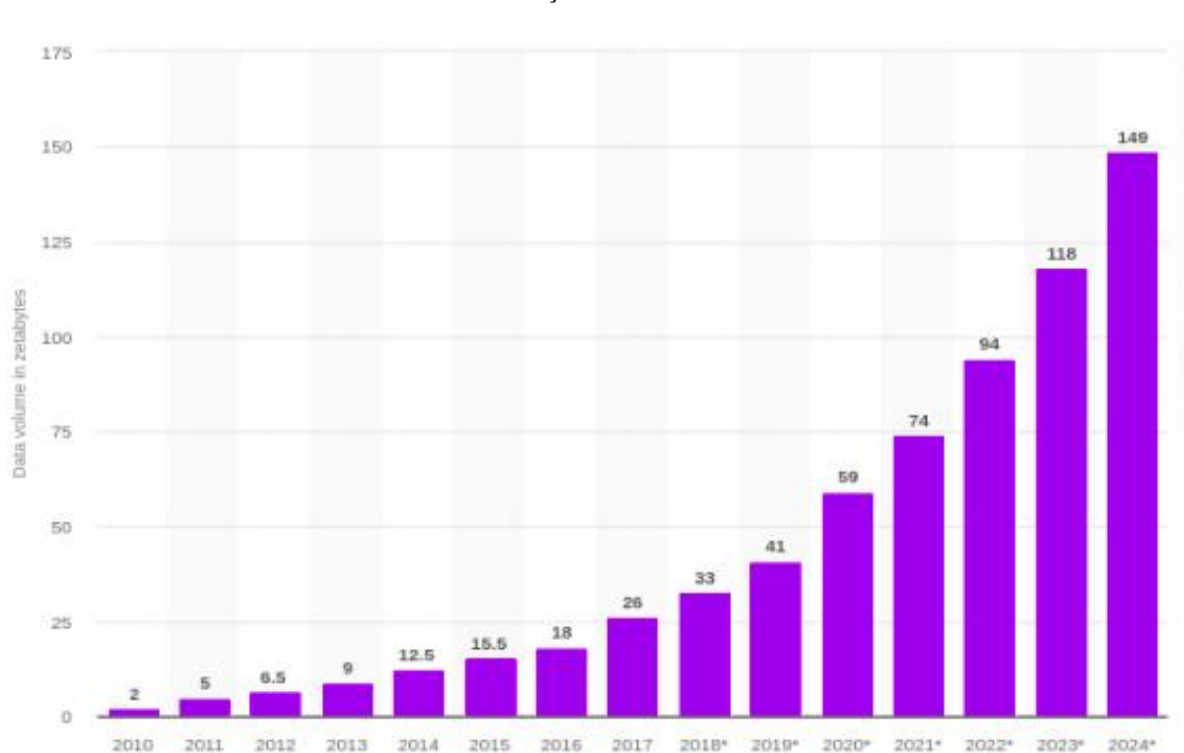
Os objetivos específicos deste trabalho são:

- a) Realização de uma pesquisa sobre o referencial teórico vinculado ao armazenamento de dados utilizando *Data Warehouse* e *Data Lake*;
- b) Realização de uma pesquisa sobre as arquiteturas de *Data Warehouse* e *Data Lake*, visando esclarecer as diferenças conceituais;
- c) Modelagem e estruturação de um ambiente de *Data Lake*, de modo a demonstrar como um ambiente corporativo pode construí-lo.

1.3 Justificativa

Segundo um estudo feito pela ABES (2020) - Associação Brasileira das Empresas de Software em parceria com o *International Data Corporation* - IDC, referente ao mercado brasileiro de *software*, o Brasil apresentou crescimento de 10,5% no segmento de TI em 2019. Considerando os mercados de *software*, *hardware* e exportações do segmento, o setor movimentou em 2019 a quantia de US\$ 44,3 bilhões. Além disso o Brasil representa 1,8% do mercado mundial de TI e 40,7% do mercado da América Latina (ABES, 2020). Com isso, empresas já consolidadas necessitam de alternativas que gerem valor, no sentido de agregar um diferencial competitivo a elas, para que se mantenham ativas no mercado e continuem crescendo.

Gráfico 1 – Volume de dados/informações criados em todo o mundo de 2010 a 2024



Statista, 2020.

Aliado a isso e conforme já mencionado anteriormente, a geração de dados cresceu de forma exponencial nos últimos anos e a estimativa é que de cresça ainda mais, como é possível observar no Gráfico 1. A perspectiva é que o volume de dados alcance a casa dos 149 *Zettabytes* em 2024.

Somasundaram e Shrivastava (2011) evidenciavam o crescimento estrondoso das informações nas empresas (ou dos dados gerados pelas empresas) e falavam sobre a importância do armazenamento, proteção, otimização e gerenciamento dessas informações. Aliado a isso, para Moresi (2000), há um consenso de que na sociedade pós-industrial, a informação passou a ser considerada um capital precioso equiparando-se aos recursos de produção, material e financeiro.

Em vistas disso, a exploração dos dados gerados pela própria organização tornou-se um bem muito valioso. Esse armazenamento e gerenciamento de dados tem se mostrado cada vez mais como uma vantagem competitiva para as empresas, pois, tendem a gerar informações estratégicas para o negócio. Tais informações estratégicas podem ajudar na previsão do futuro do negócio, na obtenção de informações relevantes para as tomadas de decisão ou ainda estimular tomadas de ações proativas e abrir caminho para melhores estratégias de mercado (ASAAD; AHMAD; ALI, 2021). Existem no mercado algumas tecnologias utilizadas para armazenar e processar grandes volumes de dados, como é o caso das tecnologias abordadas neste trabalho.

Durante a pesquisa para a definição da proposta temática do presente trabalho observou-se que há uma falta de compreensão conceitual dos termos DW e DL. O que motivou a proposta desta pesquisa.

A temática de DL tem ganho expressão, conforme pode ser observado na Figura 1. Na pesquisa à base de publicações *Web Of Science* em março de 2021, utilizando as palavras-chave 'data lake' e 'big data' na busca por termos, foram identificadas as áreas temáticas nas quais autores publicaram trabalhos de expressão internacional. Pode-se observar pela Figura 1 que o tema é totalmente convergente com o curso de Sistemas de Informação e Ciências da Computação.

Figura 1 – Resultado da pesquisa de áreas temáticas para 'data lake' e 'big data'



Autoria própria, 2021, adaptado de Web Of Science, 2021.

Nessa mesma busca, foram identificados até a presente data, 93 publicações do período de 2001 a 2020, conforme ilustra a Figura 2. Nota-se que a partir de 2016 há um aumento significativo nos trabalhos publicados nessas áreas temáticas.

Figura 2 – Publicações com referentes à data lake e big data



Web Of Science, 2021.

Dado o exposto, justifica-se o tema deste trabalho considerando que o mesmo está contido no domínio de conhecimento do curso de Sistemas de Informação, e, instiga a produção de trabalhos de pesquisa técnicos-científicos numa temática que está em crescente avanço.

1.4 Estrutura da monografia

O presente trabalho é composto por capítulos que demonstram a pesquisa sobre a temática de estudo. Neste sentido, no capítulo 1 - Introdução, são descritos de maneira breve os assuntos que serão abordados ao longo do texto, além dos objetivos, a problemática e a justificativa para o desenvolvimento da pesquisa.

No capítulo 2 - Revisão bibliográfica, são apresentados os resultados da pesquisa bibliográfica relacionada aos assuntos vinculados à temática do trabalho. Neste contexto, são apresentados os temas dados, informação e conhecimento, bem como, a fundamentação sobre banco de dados relacionais e não relacionais, visando dar ao leitor uma perspectiva sobre como tais conceitos contribuem para a formação conceitual do que são DW e DL. Na sequência são apresentados os resultados da pesquisa bibliográfica sobre a arquitetura de DW e DL.

No capítulo 3 - Metodologia de pesquisa, são descritos os tipos de pesquisa utilizados no presente trabalho, assim como as etapas metodológicas a serem seguidas no decorrer do desenvolvimento, bem como as delimitações desta monografia.

Na sequência, no capítulo 4 - Ferramentas para a criação de um DL, são apresentados dois conjuntos de ferramentas disponíveis no mercado atualmente para a construção e implementação de um DL. Bem como a equivalência entre essas ferramentas.

No capítulo 5 - Estruturação de um ambiente de um DL em nuvem, é detalhada a preparação de um ambiente de DL, desde a ingestão de dados até consultas.

No capítulo 6 - Conclusão, são descritas as conclusões realizadas pela autora com base na pesquisa e implementação contidas nesta monografia, como também são abordadas sugestões para trabalhos futuros.

2 CONCEITOS FUNDAMENTAIS

O presente capítulo tem por objetivo a contextualização sobre os assuntos que serão abordados no decorrer deste trabalho. Inicialmente serão abordados conceitos fundamentais, sobre a teoria de banco de dados, que fundamenta a formações de DW e DL. Na sequência serão aprofundados os conhecimentos sobre as duas arquiteturas visando esclarecer as diferenças conceituais entre elas.

2.1 Dados, informação e conhecimento

Segundo Elmasri e Navathe (2011), “Os dados são fatos que podem ser gravados e que possuem um significado implícito“. Já Date (2004) conceitua dado e informação como sinônimos. Contudo, o autor faz um parêntese à diferenciação feita por outros autores “[...] o termo ‘dado’ para se referir aos valores fisicamente registrados no banco de dados, e ‘informação’ para se referir ao significado desses valores para o usuário.” Ele pondera que a distinção é claramente importante, mas que deve ser explicitada apenas quando for relevante. Deste modo, é possível dizer que os dados são códigos que constituem a matéria-prima da informação, ou seja, é a informação não tratada que ainda não apresenta relevância. Para Silva (2007), os dados representam um ou mais significados de um sistema que isoladamente não podem transmitir uma mensagem ou representar algum conhecimento. De modo objetivo, pode-se dizer que os dados são elementos que representam eventos ocorridos na empresa ou circunstâncias físicas antes que tenham sido organizados ou arranjados de maneira que as pessoas possam entender e usar (ROSINI; PALMISANO, 2003). Ou ainda, conforme Oliveira (2018) pontua em sua obra, dado é qualquer elemento identificado em sua forma bruta, que por si só, não conduz a uma compreensão de determinado fato ou situação.

Diante desse pressuposto, Rosini e Palmisano (2003), afirmam que a informação é o dado configurado de forma adequada ao entendimento e à utilização pelo ser humano. Segundo Davenport e Prusak (2014) a informação tem por finalidade mudar o modo como o destinatário vê algo, exercer algum impacto sobre seu julgamento e comportamento e que diferentemente do dado, a informação tem significado, relevância e propósito. Além disso, os autores ressaltam que ao transformar dado em informação pode-se agregar valor de diversas maneiras, como:

- **Contextualizar:** uma vez que tomam conhecimento da finalidade dos dados coletados.
- **Categorizar:** descobrem-se as unidades de análise ou componentes essenciais dos dados.
- **Calcular:** pode-se analisar estatística e matematicamente os dados.
- **Corrigir:** pode-se eliminar os erros dos dados.
- **Condensar:** pode-se resumir os dados de maneira mais concisa.

Davenport e Prusak (2014) apresentam em sua obra uma definição que classificam como “definição funcional” do conhecimento:

Conhecimento é uma mistura fluida de experiência condensada, valores, informação contextual e insight experimentado, a qual proporciona uma estrutura para a avaliação e incorporação de novas experiências e informações. Ele tem origem e é aplicado na mente dos conhecedores. Nas organizações, ele costuma estar embutido não só em documentos ou repositórios, mas também em rotinas, processos, práticas e normas organizacionais. (DAVENPORT; PRUSAK, 2014).

Diante desse pressuposto conclui-se que o conhecimento é algo complexo, uma vez que, se dá pelo conjunto de diversos elementos, ao mesmo tempo, em que é fluido é também formalmente estruturado, além de ser intuitivo - o que dificulta seu entendimento lógico. Além disso o ativo humano tem uma boa parcela nessa etapa de transformação da informação em conhecimento. Davenport e Prusak (2014) afirmam que “[...] o conhecimento existe dentro das pessoas, faz parte da complexidade e imprevisibilidade humana [...]”. A Tabela 1 apresenta um resumo sobre os três conceitos abordados nesta seção.

Nesse momento é importante conceituar o que são metadados, pois eles aparecerão mais adiante, nada mais são do que informações sobre os dados, têm como objetivo catalogar e classificar informações sobre o mesmo, de modo que funcionem com o método de pesquisa (ENDEAVOR, 2015).

2.1.1 Tipos de dados

Um conceito que será visto com bastante frequência no presente trabalho é o de categorias de dados, por esta razão é importante conceituá-los. Observe a Figura 3:

Figura 3 – Exemplificação de tipos de dados



Conforme é possível observar na Figura acima e de acordo com SALESFORCE BRASIL (2020):

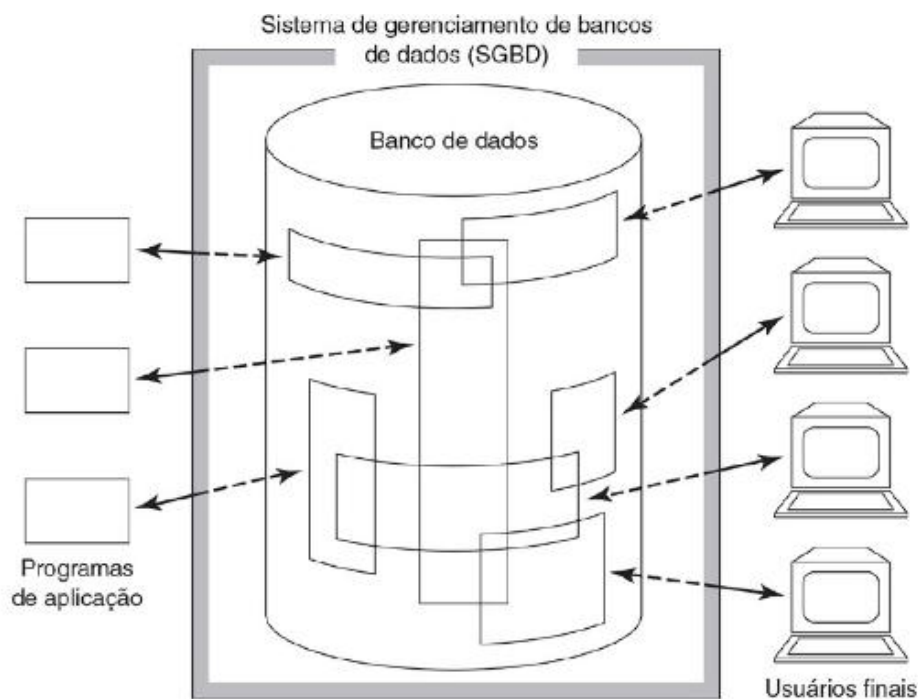
Dados estruturados, possuem maior organização, são formatados de acordo com parâmetros pré-estabelecidos, como, por exemplo, esquemas relacionais. Um dos principais formatos de dados estruturados são tabelas, pois elas os distribuem em linhas e colunas com valores pré-determinados. Outros exemplos de dados desse tipo são: planilhas eletrônicas e bancos de dados (arquivos do Excel, CSV, SQL, JSON, entre outros).

Dados semi estruturados, como sugere o nome, estão entre o meio termo. Não se encontram tão bem estruturados mas possuem uma certa organização. Como exemplos de dados deste tipo, pode-se citar: arquivos da web, como HTML, XML, OWL e outros.

Dados não estruturados, não possuem organização, nem hierarquia interna. É a categoria que abrange a maior parte dos dados da *web*, sendo a mais ampla entre elas. Como exemplos de dados assim, tem-se: arquivos texto (arquivos do Word, PDFs), arquivos mídia (imagem, áudio e vídeo), e-mails, mensagens de texto, dados de redes sociais, dispositivos móveis, Internet das Coisas (IoT), entre outros.

2.2 Banco de dados

Segundo Date (2004), banco de dados é um sistema computadorizado de manutenção de registros, em outras palavras, é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar. Ainda, de acordo com Elmasri e Navathe (2011), um banco de dados é uma coleção lógica e coerente de dados com algum significado inerente. Observe na Figura a seguir a representação de um sistema de banco de dados relacional:

Figura 4 – Representação simplificada de um sistema de banco de dados relacional

Date, 2004.

A Figura 4 representa a definição de Date (2004) de que “tal sistema envolve quatro componentes principais: dados, *hardware*, *software* e usuários”. Há uma forte integração entre esses quatro elementos. Os dados são armazenados num depósito de dados, podendo ser consultados e alterados através do *software*, por meio do *hardware*. Portanto, podemos comparar um banco de dados, como um grande ficheiro de registros de um determinado negócio, sendo ele computadorizado, facilitando sua manutenção, pesquisa e gerenciamento. Para isso, são necessários programas de gerenciamento de banco de dados, conhecidos como SGBD, conforme Elmasri e Navathe (2011). Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que permite aos usuários criar e manter um banco de dados. O SGBD é, portanto, um sistema de *software* de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento do banco de dados entre vários usuários e aplicações.

É um assunto sobre o qual poderíamos nos aprofundar bastante, citando suas muitas características e vantagens como, por exemplo: segurança, persistência e controle de redundância. Mas o objetivo aqui é a conceituação e introdução a este assunto, que servirá como base para os demais assuntos que abordaremos no decorrer deste trabalho. Conforme Elmasri e Navathe (2011), a principal classificação dos SGBDs é baseada no modelo de dados. Entre os mais utilizados atualmente devemos citar o modelo relacional e o modelo não relacional. Em um sistema relacional, o usuário vê os dados como tabelas e nada mais que tabelas. Em contraste, o usuário de um sistema não relacional vê outras estruturas de dados (DATE, 2004).

Outro ponto importante em relação ao banco de dados se refere às transações, Elmasri e Navathe (2011) conceituam-nas:

Uma transação é um programa em execução que inclui algumas operações de banco de dados, como fazer leitura do banco de dados ou aplicar inserções, exclusões ou atualizações a ele. Ao final da transação, ela precisa deixar o banco de dados em um estado válido ou consistente, que satisfaça todas as restrições especificadas no esquema de banco de dados. (ELMASRI; NAVATHE, 2011).

As propriedades dessas transações se distinguem de acordo com o modelo de dados adotado, veremos com mais detalhes essas propriedades nas seções seguintes.

2.2.1 Banco de dados relacionais

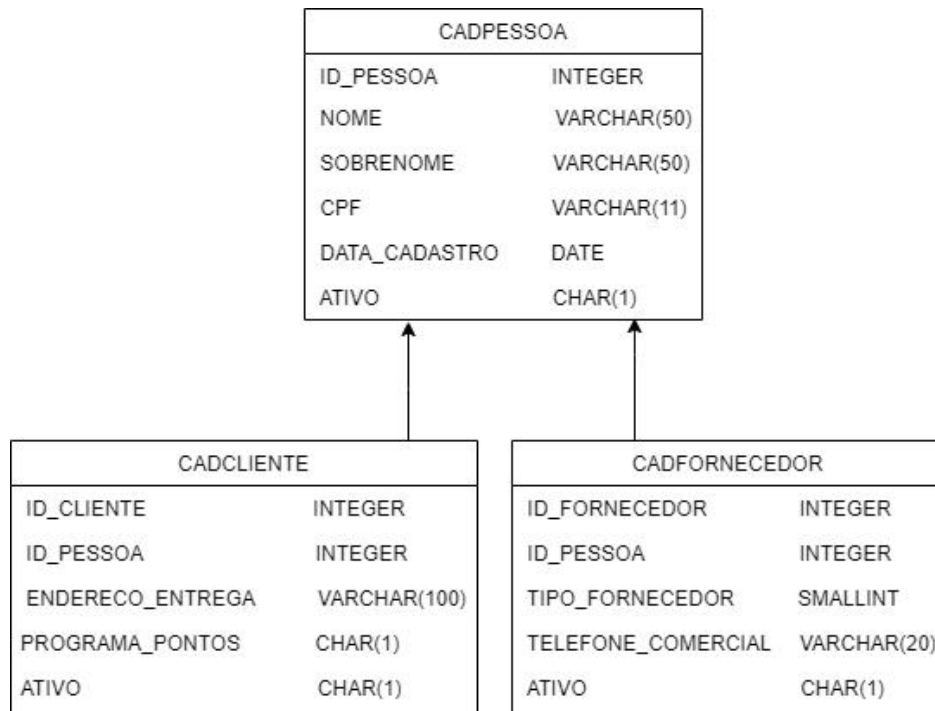
Baseados no modelo ER - entidade-relacionamento, os bancos de dados relacionais têm seus dados representados por um conjunto de relações. Essas relações ou tabelas são constituídas por uma série de elementos (atributos) que possuem características comuns (CÔRREA; ALMEIDA; GRAÇA NETO, 2017). São bancos amplamente conhecidos como bancos de dados *Structured Query Language* (SQL), por ser esta a sua linguagem de consulta.

Elmasri e Navathe (2011) explicam essa estrutura:

Quando uma relação é considerada uma tabela de valores, cada linha da tabela representa uma coleção de valores de dados relacionados. Uma linha representa um fato que normalmente corresponde a uma entidade ou relacionamento do mundo real. (ELMASRI; NAVATHE, 2011).

Seguindo uma definição informal da representação de um banco de dados relacional, pode-se afirmar que cada tabela contém um ou mais dados em colunas; cada linha, também chamada de registro, contém uma instância exclusiva de dados ou chave para os dados definidos pelas colunas. Além disso, cada tabela possui normalmente uma coluna de chave primária, que é um registro único dentro da tabela para identificação dos registros. É com base na chave primária das tabelas que são estabelecidos os relacionamentos entre elas através do uso de chaves estrangeiras (é um campo em uma tabela que se vincula à chave primária de outra tabela). Na Figura 5 é possível observar as entidades, os atributos de cada uma delas, além dos relacionamentos entre elas.

Figura 5 – Representação de um sistema de banco de dados relacional



Autoria própria, 2021, adaptado de Agüena, 2018.

No que tange as transações no modelo relacional, são seguidas as propriedades definidas como ACID - atomicidade, correção, isolamento e durabilidade. Conforme a definição de Date (2004):

- **Atomicidade:** Qualquer transação é uma proposição do tipo tudo ou nada.
- **Correção (também chamada consistência na literatura):** Qualquer transação transforma o estado correto do banco de dados em outro estado correto, sem necessariamente preservar a correção em todos os pontos intermediários.
- **Isolamento:** As atualizações em qualquer transação são ocultadas de todas as outras transações, até que determinada transação faça o COMMIT.
- **Durabilidade:** Quando determinada transação faz o COMMIT, suas atualizações sobrevivem no banco de dados, mesmo que haja uma falha subsequente no sistema.

Exemplos de bancos de dados relacionais: Oracle, SQL Server, MySQL, PostgreSQL.

2.2.2 Banco de dados não relacionais

Amplamente conhecidos como bancos de dados NoSQL, são bancos de dados não orientados à tabelas. Conforme Côrrea, Almeida e Graça Neto (2017), esse tipo de banco rompe os paradigmas propostos pelos bancos de dados relacionais, desse modo também não utiliza a linguagem *SQL*. Foram criados objetivando solucionar limitações do modelo relacional, como a escalabilidade horizontal, que no modelo anterior era difícil de ser implementada, e de alta performance, visto que no modelo relacional consultas à grandes bases de dados levava

tempo considerável. Além do mais, esse tipo de banco trabalha com dados estruturados, semi-estruturados e não estruturados.

Segundo Côrrea, Almeida e Graça Neto (2017), os bancos de dados não relacionais podem ser classificados quanto a sua estratégia de armazenamento de dados, veja a seguir essa classificação:

- **Chave-valor:** modelo simples que permite visualização do banco como uma grande tabela. O banco é composto por um conjunto de chaves que estão associadas. Exemplos de bancos chave-valor: Redis, Membase, Voldemort e MemcacheDB.

Figura 6 – Exemplo de banco de dados chave-valor

Chave (Campo)	Valor (Instância)
Nome	Hélio Rodrigues
Idade	45
Sexo	Masculino
Fone	99 999999999

Autoria própria, 2021, adaptado de Fernandes, 2013.

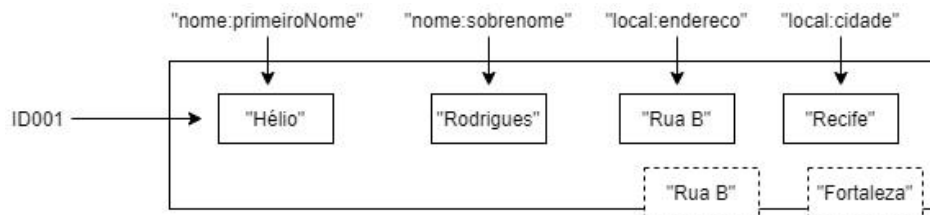
- **Orientado à documentos:** armazena uma coleção de documentos. O documento é um objeto com código único e um conjunto de campos que podem ser strings, listas ou documentos aninhados. É uma estrutura que se assemelha à chave-valor e não exige de uma estrutura fixa. Exemplos: MongoDB, CouchDB e RavenDB.

Figura 7 – Exemplo de banco de dados orientado à documentos

ID: P001
Assunto: "Eu gosto de laranja"
Autor: "Hélio"
Data: "27/01/2011"
Tags: ["laranjas", "suco", "plantas"]
Mensagem: "Hoje estou com vontade de tomar suco de laranja!"

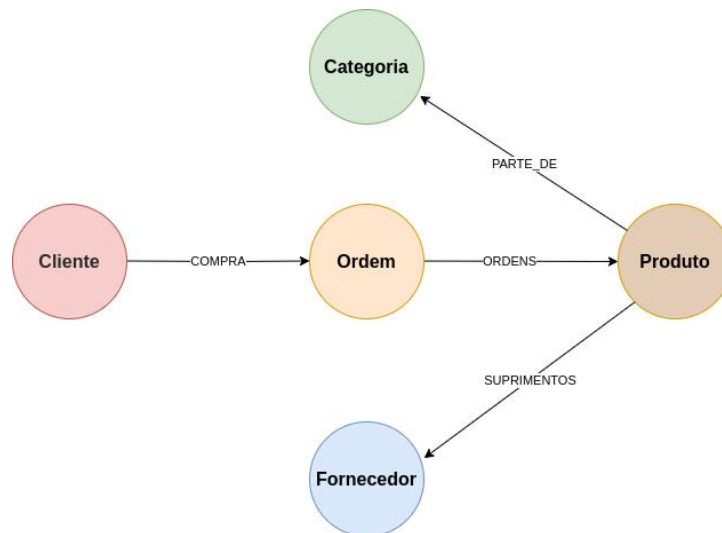
Autoria própria, 2021, adaptado de Fernandes, 2013.

- **Orientado à colunas:** maior complexidade, indexados por uma tripla (linha, coluna e *timestamp*), linhas e colunas são identificadas por chaves e o *timestamp* permite identificar versões diferentes de um mesmo dado. Esse banco permite o particionamento, tem forte consistência, mas não garante alta disponibilidade. Exemplos: BigTable, Hadoop/HBase, Cassandra e SimpleDB.

Figura 8 – Exemplo de banco de dados orientado à colunas

Autoria própria, 2021, adaptado de Fernandes, 2013.

- **Orientados à grafos:** apresenta três componentes básicos: nós/vértices representam as entidades, arestas representam os relacionamentos e os atributos/propriedades dos nós e relacionamentos. Exemplos: Neo4J, FlockDB e InfiniteGraph.

Figura 9 – Exemplo de um grafo simples

Autoria própria, 2021, adaptado de Neo4j, 2021.

Em relação às transações no modelo não relacional, é seguido o conceito BASE - basicamente disponível, estado leve e consistência eventual. Conforme descreve Fernandes (2013):

Uma aplicação funciona basicamente todo o tempo (basicamente disponível), não tem que ser consistente o tempo todo (estado leve) e o sistema torna-se consistente no momento devido (eventualmente consistente). (FERNANDES, 2013)

Na Tabela 1 é possível observar as principais diferenças entre os conceitos ACID, comuns ao modelo de banco de dados relacional e, do conceito BASE, comuns aos modelos de banco de dados não-relacionais.

Tabela 1 – Comparativo entre os conceitos ACID e BASE

ACID	BASE
Forte consistência	Fraca consistência
Isolamento	Foco em disponibilidade
Concentra-se em "commit"	Melhor esforço
Transações aninhadas	Respostas aproximadas
Disponibilidade	Mais simples e mais rápido
Conservador (pessimista)	Agressivo (otimista)
Evolução difícil (por exemplo, esquema)	Evolução mais fácil

Autoria própria, adaptado de Fernandes, 2013.

Além de trabalhar sobre as propriedades BASE, o NoSQL também trabalha sobre o Teorema CAP, proposto por Eric Brewer. Segundo Brewer (2000), um sistema computacional distribuído deve procurar obter consistência, disponibilidade e tolerância à falhas, contudo, através do Teorema CAP, Brewer demonstra que apenas duas dessas propriedades podem ser garantidas simultaneamente.

Para Côrrea, Almeida e Graça Neto (2017) essas são algumas vantagens dos modelos de dados não-relacional: ausência de esquema ou esquema flexível, horizontalmente escalável, baixo custo, complexidade, confiabilidade, coerência e desconhecimento da tecnologia.

Com base no exposto é possível compreender que essas teorias fundamentaram os princípios que nortearam a construção dos DWs e DLs. Portanto, nos tópicos seguintes serão abordados os conceitos arquiteturais dessas duas plataformas.

2.3 Data Warehouse

O termo remonta a década de 80 quando Barry Devlin e Paul Murphy, pesquisadores da IBM, o cunharam para designar sua proposta de sistema de informação integrado (HAYES, 2002). Os autores propuseram em seu artigo para o *IBM Systems Journal* um modelo de sistema de informações que atuasse de forma consistente e integrada com o objetivo de fornecer informações relevantes para os tomadores de decisão (HOJI, 2012). Foi na década de 90 com a publicação do livro *"Building the Data Warehouse"* de Willian Inmon que o termo ficou mais conhecido, fazendo com que o autor fosse reconhecido por muitos como "pai" e criador do termo.

A crescente utilização dessa tecnologia pelas organizações se deu em grande parte pela necessidade de possuírem informações estratégicas e meios de utilizá-las de modo a assegurar respostas e ações rápidas no ambiente altamente competitivo e mutável como o mundo dos negócios (HOJI, 2012). Corroborando essa afirmação, Kimball e Ross (2013) afirma que um dos ativos mais importantes de qualquer organização é a informação, sendo, este ativo, quase sempre utilizado para dois propósitos: manutenção de registros operacionais e análise para tomada

decisão. Alguns autores como Machado, afirmam que o *Data Warehouse* é uma evolução natural dos Sistemas de Apoio a Decisão (SAD), conforme apresentado anteriormente por Laudon e Laudon (1999).

Outra contribuição relevante de Kimball e Ross (2013) sobre o tema é em relação ao armazenamento da informação, ele afirma que a informação normalmente é armazenada de duas maneiras na organização, em sistemas transacionais e no *Data Warehouse*. O autor faz uma analogia para exemplificar essa afirmação, os usuários do sistema transacional ou de operações da organização seriam aqueles que fariam a “roda girar”, ao passo que os usuários do DW observariam a “roda girar”. Na Tabela 2 é possível visualizar algumas características desses sistemas.

Tabela 2 – Comparativo entre Sistemas transacionais e Sistemas de suporte à decisão

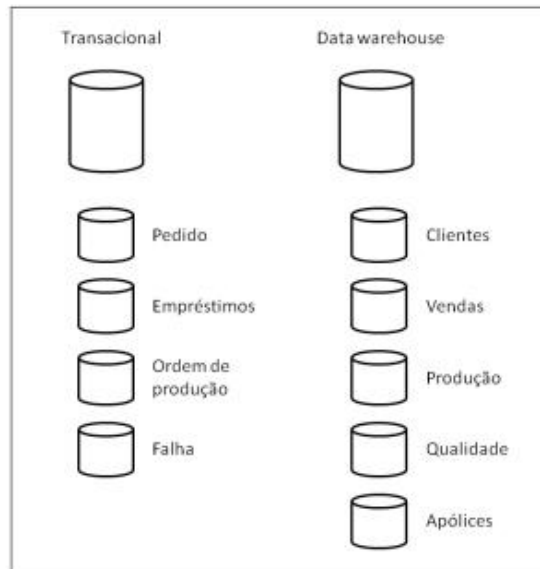
Sistemas transacionais	Sistemas de apoio à decisão (Data Warehouse)
Orientado por aplicação	Orientado por assunto
Dados detalhados	Dados detalhados e sumarizados
Dados precisos, no momento do acesso	Representa valores no tempo, fotografias
Atende a comunidade funcional	Atende a comunidade gerencial
Atualizável	Não atualizável
Alta disponibilidade	Razoável disponibilidade
Estrutura estática	Estrutura flexível
Alta probabilidade de acesso	Média/baixa probabilidade de acesso
Suporte a operações do dia-a-dia	Suporte a necessidades gerenciais
Não há redundância	Redundância faz parte do sistema
Acesso a um único registro	Acesso a múltiplos registros de uma só vez
Dirigido à transação - OLTP	Dirigido à análise - OLAP
Pequena quantidade de dados utilizada em um processo	Alta quantidade de dados utilizada em um processo

Autoria própria, 2021, adaptado de Inmon, 2004.

Date (2004) define *Data Warehouse* como um depósito de dados orientado por assunto, integrado, não volátil, variável com o tempo, para apoiar decisões gerenciais. Essas quatro características citadas por Date foram descritas por Inmon (2002), observe as definições:

Orientados para o assunto: eles podem analisar dados sobre um determinado assunto ou área funcional (como vendas).

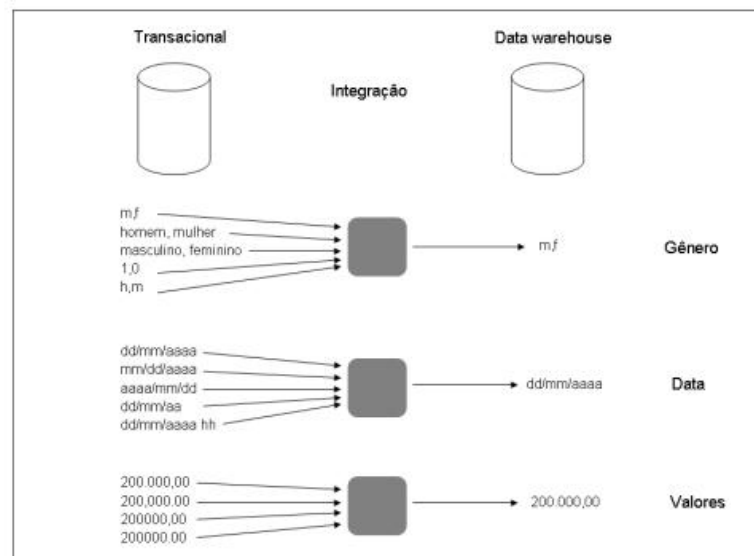
Figura 10 – Exemplo de orientação por assunto



Hoji, 2012.

Integrados: os *Data Warehouse* criam consistência entre diferentes tipos de dados de fontes distintas.

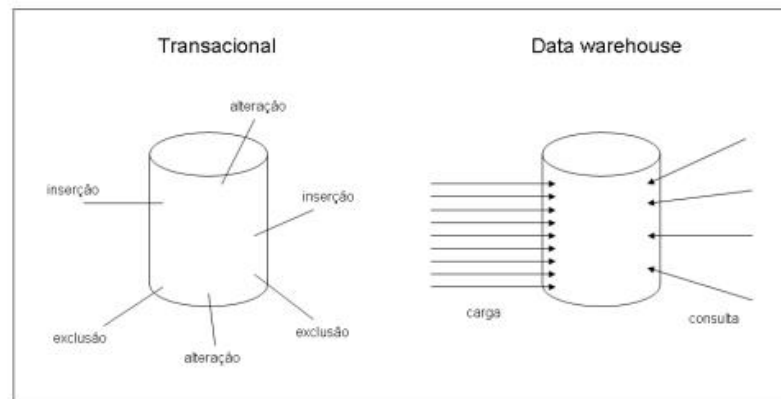
Figura 11 – Exemplo de visão integrada do DW



Hoji, 2012.

Não voláteis: quando os dados estão em um *Data Warehouse*, eles são estáveis e não mudam.

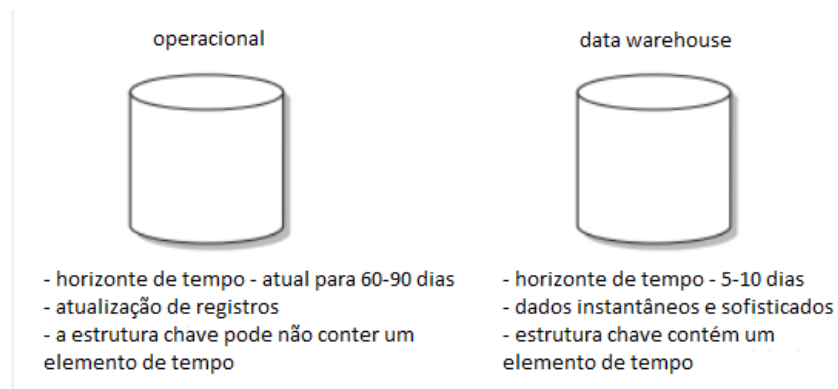
Figura 12 – Exemplo de não volatilidade do DW



Hoji, 2012.

Variáveis de acordo com o tempo: a análise de *Data Warehouse* analisa as mudanças ao longo do tempo.

Figura 13 – Exemplificação de variantes no tempo



Autoria própria, 2021, adaptado de Inmon, 2002.

Além disso, Inmon (2002) cita os principais problemas relacionados à existência de diversas bases e sistemas de dados isolados dentro de uma organização:

- Falta de credibilidade dos dados
- Problemas de produtividade
- Incapacidade de transformar dados em informações

Data Warehouses suportam fluxo de dados de múltiplos sistemas operacionais para sistemas de análise/solução, criando dessa maneira um repositório único de dados de diversas origens através de processos *ETL*. Desse modo, fica evidente o conceito do *Data Warehouse*, centralizar os dados retirados dos seus locais de origem (como planilhas, ERPs, CRMs, entre outros) e acomodá-los em um único local – o *Data Warehouse* propriamente dito. Dessa maneira, as informações passam a ficar concentradas em um só lugar, criado com foco na consulta.

Um *Data Warehouse* bem projetado realizará consultas muito rapidamente, fornecerá alta taxa de transferência de dados e dará flexibilidade suficiente para os usuários finais dividirem e organizarem ou reduzirem o volume de dados para um exame mais detalhado com o propósito de atender a uma variedade de demandas. Além disso, o *Data Warehouse* serve como uma base funcional para ambientes de *BI* de *middleware* fornecendo aos usuários finais relatórios, painéis e diversas outras interfaces.

Segundo Kimball e Ross (2013), os principais os objetivos dessa tecnologia são:

- Tornar as informações facilmente acessíveis;
- Apresentar as informações de forma consistente;
- Ser adaptável à mudanças;
- Apresentar as informações em tempo hábil;
- Ser um armazém de dados seguro, protegendo os ativos de informação;
- Servir como base autorizada e confiável para a tomada de decisão;

Pelas características apresentadas, por toda fundamentação feita e abordagens, o DW se mostra vantajoso para uma organização que deseja oferecer visões organizadas sobre o negócio, como ferramenta para auxiliar nas tomadas de decisões, sendo capaz proporcionar maior embasamento e clareza acerca das situações.

Há uma certa confusão no que diz respeito a arquiteturas e metodologias de implementação de um *Data Warehouse*. Visando a simplificação dessa temática Sá (2009) definiu arquitetura como uma descrição do que se deve construir, quais são os componentes, qual o papel de cada um deles e como eles interagem entre si e, definiu como metodologia a descrição do que se deve construir e implementar, focando nos resultados das atividades e técnicas.

De modo claro e objetivo, pode-se definir que uma arquitetura tradicional de *Data Warehouse* engloba:

- a) processos de ETL (extração, transformação e carga - que é a sistematização do tratamento e limpeza dos dados provenientes de dos sistemas da organização
- b) carga dos dados
- c) ferramentas de BI para análise desses dados.

Brito (2018) destaca que os aplicativos de BI são usados pelas empresas para análise dos dados e geração de *insights* sobre o negócio da empresa, ela ressalta a existência de uma vasta gama de categorias dessas ferramentas, tais como: relatórios, painéis, mineração de dados, OLAP e monitoramento do negócio.

Há duas abordagens referentes à metodologia e modelagem arquitetural de um *Data Warehouse* muito famosas e divergentes entre si, uma delas é a arquitetura bottom-up proposta por Kimball e a outra a arquitetura down-up proposta por Inmon.

Do ponto de vista estrutural do *Data Warehouse*, Kimball e Ross (2013) afirmam que a modelagem dimensional é amplamente aceita como técnica preferida para tratar dados analíticos por abordar dois requisitos simultâneos:

- a) Entregar dados compreensíveis para os usuários corporativos.
- b) Oferecer agilidade na consulta.

Segundo Kimball e Ross (2013) a simplicidade é fundamental, pois, garante que os usuários consigam entender facilmente os dados, além de permitir que o *software* entregue resultados com rapidez e eficiência.

2.4 Data Lake

Um *Data Lake* propõe um armazenamento de dados em seu formato nativo, dados volumosos e diversamente estruturados, em um local de armazenamento evolutivo que permite análises posteriores, com relatórios, painéis e visualizações a processamento de big data, análise em tempo real e aprendizado de máquina para orientar as melhores decisões (NOGUEIRA; ROMDHANE, 2018).

A denominação para esse repositório foi criada por James Dixon, CTO da Pentaho. É apropriado descrever esse tipo de repositório como um lago porque ele armazena um conjunto de dados em seu estado natural, como um corpo d'água que não foi filtrado ou contido. Os dados fluem de diversas fontes para o lago e são armazenados em seu formato original (KHINE; WANG, 2018).

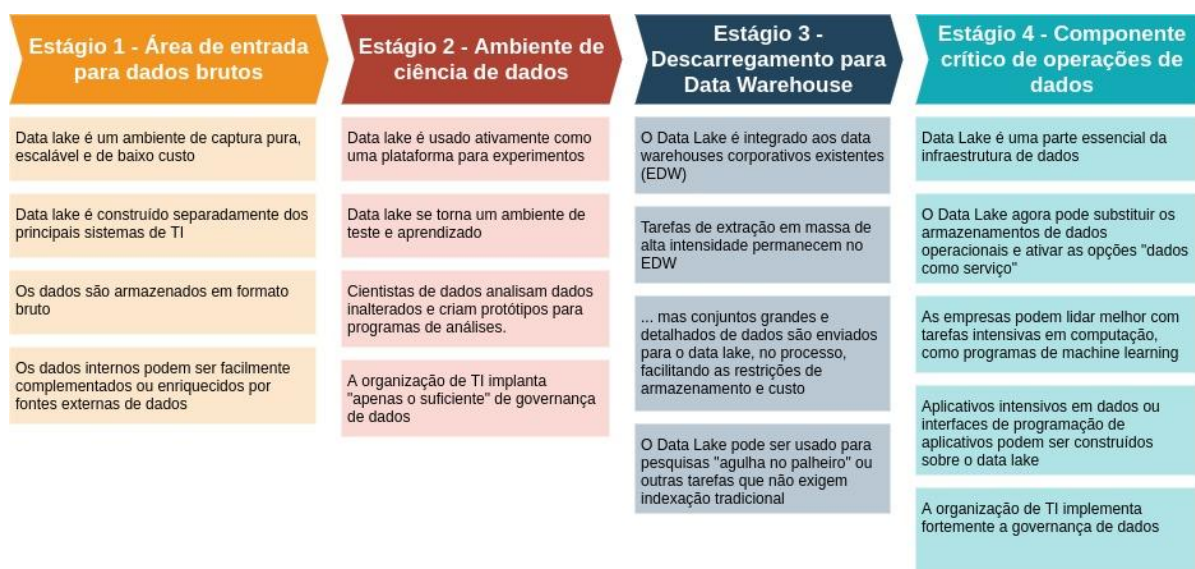
Segundo Khine e Wang (2018), a ideia básica do *Data Lake* é simples, todos os dados emitidos pela organização serão armazenados em uma única estrutura de dados chamada *Data Lake*. Os dados serão armazenados em seu formato original, dispensando a etapa de transformação durante o pré-processamento dos dados, agilizando o processo de disponibilidade dos dados após a inserção dos mesmos na base. Propondo um novo modelo de processamento, conhecido como ELT, que significa: extração, carregamento e transformação.

Um ponto interessante a ser mencionado, levando-se em consideração uma pesquisa feita pela empresa Aberdeen, esse tipo de repositório, além de ajudar as empresas a explorarem o potencial dos mais diversos tipos de dados, também pode ajudar a tornar os sistemas legados mais eficientes, transferindo a capacidade para infraestrutura mais nova e flexível (ABERDEEN, 2017).

Outro ponto a ser levado em consideração com o uso de Data Lake, de acordo com o Semantix (2019), é o papel da governança de dados. Com grandes volumes de dados é natural que não se compreenda todas as informações ali contidas, desta forma juntamente com implementação desta solução se faz necessário o planejamento de projetos de segurança e governança.

Etapas para a implementação de um Data Lake, segundo McKinsey&Company (apud HAGSTROEM et al., 2017):

Figura 14 – Estágios para implementação de um Data Lake



Autoria própria, 2020, adaptado de McKinsey&Company (apud HAGSTROEM et al., 2017).

As etapas detalhadas na Figura 14 serão mais exploradas na seção 5 desta monografia.

Quando o assunto é a ingestão de dados no DL, há dois modos de serem feitos, conforme é explicitado a seguir.

Carga em tempo real: o custo de implementação é maior e se faz necessário o uso de um serviço de mensageria.

Carga em batch ou lote: é um pré-agendamento das consultas às fontes de dados.

Pelas características apresentadas, por toda fundamentação feita e abordagens, o DL mostra-se vantajoso quando a necessidade do negócio é ter um processamento de dados variados, em tempo real e sem necessariamente ter um profissional especializado para fazer análises.

2.5 Comparativo entre as tecnologias

Segundo Khine e Wang (2018), eles diferem em muitos aspectos, desde conceitos, estruturas até implementação. Os Data Warehouse possuem funções reguladoras definidas e capacidade de armazenamento. Em teoria, Data Lake não tem limite para capacidade de armazenamento, qualquer tipo de dados com qualquer quantidade pode ser carregado no armazenamento do Data Lake repositório.

A Tabela 3 ilustra de maneira simplificada as principais diferenças entre essas duas abordagens:

Tabela 3 – Comparativo entre Data Warehouse e Data Lake

Comparação	Data Warehouse	Data Lake
Dados	Estruturados, dados processados	Estruturados/Semi-estruturados, não estruturados, dados não tratados, dados não processados
Processamento	Esquema na gravação	Esquema na leitura
Armazenamento	Caro, confiável	Baixo custo
Agilidade	Menos ágil, configuração fixa	Alta agilidade, configuração flexível
Segurança	Maduro	Amadurecendo
Usuários	Profissionais de negócio	Cientista de dados (especialistas)

Autoria própria, 2021, Adaptado de Khine e Wang, 2017.

Observe a seguir, as ponderações feitas por Khine e Wang acerca dessas diferenças: (KHINE e WANG, 2017, tradução da autora):

- a) **Dados** – dizem que no domínio comercial apenas 20% dos dados são estruturados, ou seja, o DW é a nata da colheita, uma vez que aceita apenas dados estruturados, extremamente resumidos e consolidados por ETL. Os outros 80% dos dados seriam semiestruturados e não estruturados. Não há uma conceituação exata dessa porcentagem, apenas se sabe que a maior porção equivale aos dados não estruturados e semiestruturados.
- b) **Processamento** - Como os dados armazenados no DW são cuidadosamente selecionados através dos processos ETL, eles acabam se limitados devido à natureza resumida e estruturada, desse modo eles não conseguem responder à perguntas prontas dos tomadores de decisão e à perguntas que necessitem extração de dados de transações combinados com dados não estruturados. Em contrapartida, o DL pode lidar com esse tipo de consulta, pois, somente quando o usuário consultar os dados é que eles serão transformados (Extrair-Carregar-Transformar), aplicando a abordagem “Esquema na leitura”. Isso dá

mais flexibilidade para o cientista de dados, familiarizado com dados inexplorados, dados brutos ou binários, combinados com dados estruturados.

- c) **Custo** - Muitas soluções de DL são implementadas em estruturas de código aberto e projetadas para servidores comuns. Portanto, se comparado às altas taxas de licenciamento de dados armazenamento em DW, é relativamente mais barato.
- d) **Agilidade** - o design do DW é feito antes do carregamento dos dados (esquema na gravação). Por definição, essa é uma abordagem altamente estruturada e com dados altamente controlados pela gestão. Embora seja possível alterar o design do DW, é algo muito custoso a se fazer, uma vez que toda a sua arquitetura foi planejada especificamente para os processos de negócio. Muitas vezes sendo necessária a reconsideração de todo o projeto, algo que eleva o custo em todas as esferas. O DL não possui uma estrutura explicitamente definida como o DW, portanto, eles são mais flexíveis e ágeis. Além disso, eles dão aos desenvolvedores e cientistas de dados a capacidade de configurar facilmente os modelos, as consultas e os aplicativos on-the-fly.
- e) **Segurança** - Os DW existem há décadas e têm uma definição bem concreta de segurança, em contrapartida, os dados no DL ainda são pouco explorados, sendo deixados como área de pesquisa aberta.
- f) **Usuários** - até o momento, o DL ainda é mais adequado para analistas e cientistas de dados, devido aos motivos mencionados anteriormente. Muitos especialistas no mercado atual estão mais familiarizados com os procedimentos de armazenamento de dados e consideram DL incompetente e pesado. No entanto, cientistas de dados que estão interessados nos conceitos de DL estão pesquisando e construindo DL (especialmente a construção de teste de DL em pequenas e médias empresas, ou construindo como protótipos de amostra). Eles são os principais recursos que têm fornecido feedbacks para a melhoria do mesmo.

Enquanto armazém de dados, DW necessita uma modelagem pré-definida dos dados para fazer um pré-processamento dos mesmos para uso posterior. Além do mais, é um recurso com necessidade de arquitetura pré-definida e alto investimento, sendo considerado um “*model-on-write*”.

Já um DL, conforme apresentado nos capítulos anteriores desta monografia, faz o carregamento dos dados independente de seu formato e efetua a transformação apenas no momento da consulta. Abordagem conhecida como “*model-on-ready*”, é um modelo que lida bem com altas cargas de dados, tanto de leitura, quanto de gravação.

Dado o exposto, nota-se que ambos possuem diferentes propostas. Ao ponto que DL é capaz de abrigar um elevado volume de dados em variados formatos e dar acesso a eles de forma ágil sem que se faça necessário grandes investimentos, a proposta do DW é apresentar visões

organizadas que possibilitem o direcionamento das tomadas de decisões com base nos dados, é um modelo que exige maiores investimentos, uma vez que necessita de planejamento arquitetural e modelagem antes mesmo da ingestão de dados.

Desta forma, a conclusão que se chega é que a determinação do uso de uma ou de outra tecnologia, irá depender de alguns fatores, como:

- quais fontes de informação estão disponíveis;
- como esses dados estão organizados;
- qual é a necessidade da empresa no uso dos dados; e,
- qual é a sua capacidade de investimento.

3 MÉTODO DE PESQUISA

O presente capítulo aborda o método de trabalho utilizado para responder às perguntas de pesquisa propostas. De acordo com Mota *et al.* (2016), a metodologia descreve os procedimentos metodológicos da pesquisa, ou seja, recursos foram utilizados para coletar informações na busca por respostas para o problema.

3.1 Classificação da pesquisa

Para Minayo (2014), a pesquisa é considerada uma atividade básica das ciências na sua indagação e descoberta da realidade. Silva e Menezes (2005) corroboram essa afirmação e a complementam afirmando que a pesquisa é uma atitude e uma prática teórica de constante busca que define um processo intrinsecamente inacabado e permanente.

Segundo Silva e Menezes (2005) há diversas formas de classificar as pesquisas, seguindo as definições clássicas os autores as classificam segundo a sua natureza e sua forma de abordagem do problema.

Quanto à natureza, Silva e Menezes (2005) classificam em:

- a) Pesquisa Básica: objetiva gerar conhecimentos novos úteis para o avanço da ciência sem aplicação prática prevista. Envolve verdades e interesses universais.
- b) Pesquisa Aplicada: objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos. Envolve verdades e interesses locais.

E quanto à abordagem do problema Silva e Menezes (2005):

- a) Pesquisa Quantitativa: considera que tudo pode ser quantificável, o que significa traduzir em números, opiniões e informações para classificá-las e analisá-las. Requer o uso de recursos e de técnicas estatísticas (percentagem, média, moda, mediana, desvio-padrão, coeficiente de correlação, análise de regressão, etc.).
- b) Pesquisa Qualitativa: considera que há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números. A interpretação dos fenômenos e a atribuição de significados são básicas no processo de pesquisa qualitativa. Não requer o uso de métodos e técnicas estatísticas. O ambiente natural é a fonte direta para coleta de dados e o pesquisador é o instrumento-chave. É descritiva. Os pesquisadores tendem a analisar seus dados indutivamente. O processo e seu significado são os focos principais de abordagem.

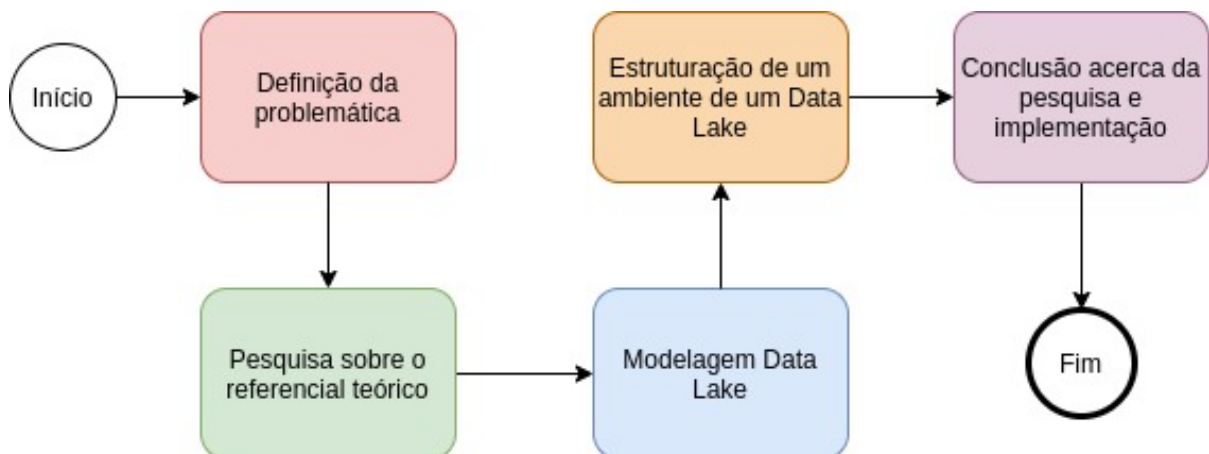
O método deste trabalho se caracteriza quanto à natureza como uma pesquisa aplicada, uma vez que o objetivo final é a produção de conhecimento para uma aplicação prática, que é a implantação de um Data Lake. Em relação aos procedimentos técnicos, se enquadra como uma pesquisa bibliográfica, por ser elaborada a partir de materiais já publicados (SILVA; MENEZES, 2005). Quanto a abordagem da problemática, classifica-se com uma pesquisa qualitativa, uma vez que interpretação será baseada nos resultados e métodos aplicados.

3.2 Atividades metodológicas

As atividades metodológicas a serem seguidas para a conclusão deste trabalho são descritas como:

- 1) Definição da problemática
- 2) Pesquisa sobre o referencial teórico vinculado ao armazenamento de dados utilizando *Data Warehouse* e *Data Lake*, visando esclarecer diferenças conceituais e de aplicação entres ambas.
- 3) Modelagem de um *Data Lake*.
- 4) Estruturação de um ambiente de um *Data Lake*.
- 5) Considerações finais sobre a temática.

Figura 15 – Atividades metodológicas



Autoria própria, 2021.

3.3 Delimitações

O presente trabalho limita-se a fazer uma pesquisa sobre o referencial teórico das arquiteturas de DL e DW e, demonstrar a estruturação de um ambiente de DL, propondo essa solução com uma ferramenta de gerenciamento de dados na esfera corporativa na área de TI. Portanto, como delimitações, destaca-se que inicialmente essas ferramentas não serão efetivamente implantadas no ambiente corporativo.

4 FERRAMENTAS PARA A CRIAÇÃO DE UM DL

Considerando os conceitos apresentados e compreendendo o significado e aplicação de DW e DL, no presente capítulo, será abordada a temática principal do trabalho que trata de dois conjuntos de ferramentas para montagem de um DL. Sendo um deles, *open source* e outro de *software* proprietário. São elas, respectivamente, Apache Hadoop e Amazon AWS.

4.1 Apache Hadoop

O Apache Hadoop que é mantido pela Apache Foundation, já é considerado um ecossistema devido à quantidade de aplicações que podem ser utilizadas em conjunto para se implementar um repositório de dados. Segundo o IPSENSE (2019), o objetivo principal desse conjunto de ferramentas é viabilizar alto processamento e armazenamento de dados de forma distribuída, através da utilização de *clusters* de *hardwares* de baixo custo e com alta tolerância à falhas.

Na Figura 16 é possível observar parte destas ferramentas.

Figura 16 – Ecossistema Hadoop



Apache, 2020.

Conforme explicitado na figura acima, há diversas ferramentas disponíveis para a implementação, consulta, análise e transformação dos dados utilizando o Apache Hadoop. Nesse momento serão detalhadas algumas destas aplicações, de acordo a documentação oficial do Hadoop, THE APACHE SOFTWARE FOUNDATION (2021).

Pode-se dizer que o Hadoop é o “*core*” da aplicação, pois, é a plataforma onde ocorre a ingestão dos dados. Essa plataforma possui dois componentes principais e módulos adicionais, são eles, respectivamente :

- **Hadoop Map Reduce:** é uma estrutura de *software* que possibilita implementar aplicações que processam grandes quantidades de dados em paralelo em grandes *clusters* de

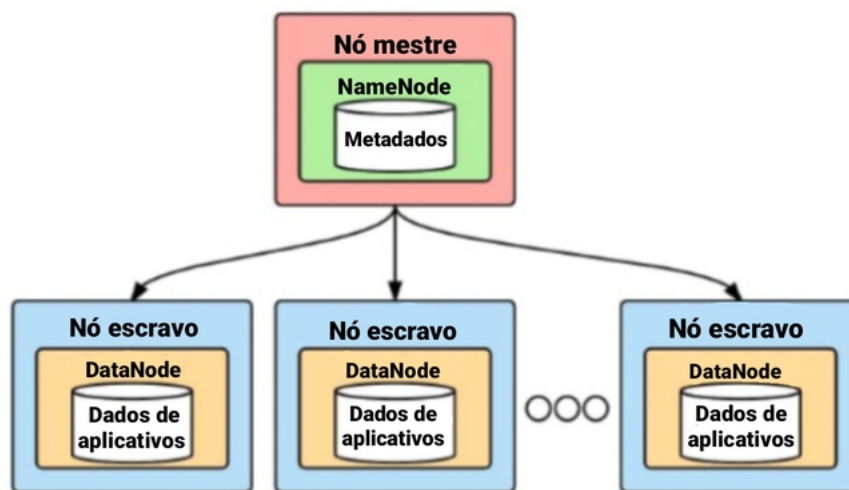
forma simples, de modo geral é a camada de processamento dos dados.

- **HDFS** (*Hadoop Distributed File System*): é um sistema de arquivos distribuídos que foi projetado para ser executado em *hardware* de baixo custo. Embora tenha muitas semelhanças com os sistemas de arquivos distribuídos existentes, este possui algumas particularidades, como alta tolerância à falhas. Além disso, fornece acesso de alto rendimento aos dados da aplicação, sendo adequado para aplicações que possuem grandes volumes de dados, é basicamente, a camada de armazenamento dos dados.

No cenário de implementação de um DL, o HDFS é onde ocorre a ingestão dos dados brutos, suportando todo o tipo de arquivo, além de permitir particionamento de dados, ele organiza os arquivos na rede e faz a replicação dos arquivos nos nós do *cluster*, deixando todo o processo transparente para o usuário.

Na Figura 17 é possível visualizar a arquitetura base do HDFS:

Figura 17 – Arquitetura base do HDFS



Autoria própria, 2021, adaptado de Oeiras, 2020.

O HDFS possui uma arquitetura mestre-escravo, como é possível observar na figura acima. De acordo com a THE APACHE SOFTWARE FOUNDATION (2021), um cluster HDFS possui um único *NameNode* mestre que faz a gestão do sistema de arquivos e gerencia o acesso à estes pelos clientes. Ademais, há diversos *DataNodes*, normalmente um por *Node* no *cluster* que gerenciam o armazenamento anexado aos *Nodes* em que são executados, eles também são responsáveis pelas ações de leitura e gravação solicitadas pelos clientes do sistema. Uma boa

vantagem é que tanto NameNode, quanto DataNode são projetados para rodar em máquinas convencionais (THE APACHE SOFTWARE FOUNDATION, 2021).

Para contextualizar sobre algumas expressões vistas na estrutura de um DL no ecossistema em questão, seguem suas definições:

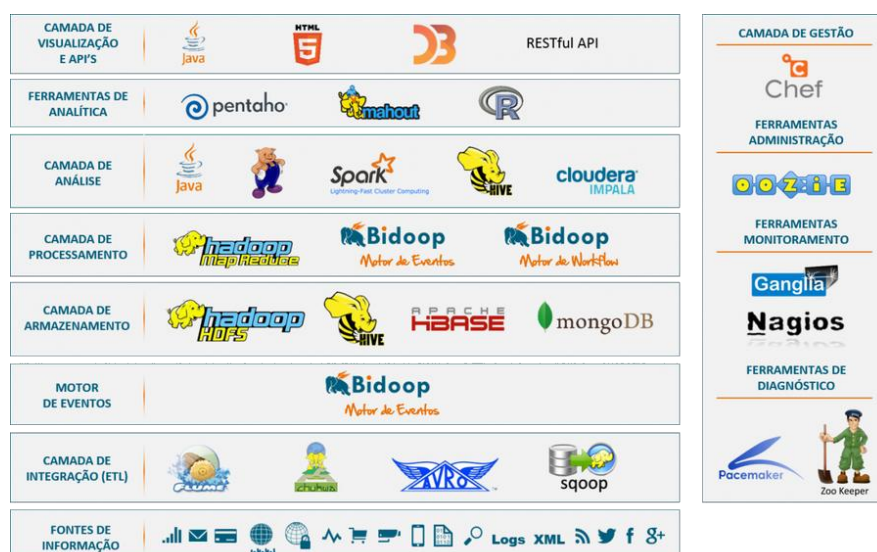
- *Cluster*: é um conjunto de computadores que trabalham de forma integrada a fim de processar grandes quantidades de dados.
- *Node* (Nó): como é chamado cada computador do *cluster*.
- *NameNode*: armazena os metadados e os organiza nos *DataNodes*, funciona como um *DNS*.
- *DataNode*: local onde os dados são armazenados e que provém os recursos para o processamento.
- *Edgenode*: é a orla à volta do Hadoop.
- *Arestas*: é como são chamados os pontos de comunicação no *cluster*.

Conforme explicitado anteriormente neste capítulo, seguem os módulos adicionais do Hadoop:

- **Hadoop Yarn**: arquitetura responsável pela gestão dos *clusters* e agendamento de tarefas, é conhecida como a camada de gerenciamento de recursos.
- **Hadoop Common**: inclui bibliotecas Java e utilitários necessários para outros módulos Hadoop.

A Figura 18 possibilita a visualização do sistema em camadas e especifica onde cada ferramenta pode atuar.

Figura 18 – Aplicações que compõem o ecossistema Hadoop



Conforme pôde ser visto nas Figuras 16 e 18, este framework possui diversas ferramentas, entre elas, pode-se destacar:

Sqoop: é uma ferramenta de extração de dados para bancos de dados relacionais, conforme citados na seção 2.2.1. É responsável por fazer a ingestão do Hadoop e inserir os dados no Hive.

Pig: é uma linguagem de script executada sobre o Yarn e o Tez, que permite o processamento de grandes volumes de dados.

HBase: banco de dados NoSQL orientado à colunas, como mencionado na seção 2.2.2, de baixa latência e processamento rápido. Permite o processamento de tabelas com bilhões de registros e milhões de colunas.

Kyle: é uma ferramenta de cubo de dados, tem como função capturar os dados do HBase ou Hive e processá-los, fazendo desta maneira, o que chama-se de pré-processamento de dados.

Zookeeper: tem como objetivo assegurar a consistência do *cluster*, verificando a replicação de dados, a indisponibilidade de máquinas e fazendo a nomeação de serviços. Foi desenvolvido para rodar em máquinas problemáticas, uma de suas características é fazer a reserva do dado e posterior processamento, além disso, é instalado em todos os nós do *cluster*.

Apache Range: pode ser considerado como um tipo de banco de dados, uma vez que gerencia perfis para usuários e grupos, tratando de acessos e assegurando a segurança de dados, por esse motivo está intimamente atrelado às questões relacionadas à LGPD. Funciona através de estratégias de plugins, interceptando e validando dados. Além disso em apoio ao Hive consegue mascarar os dados.

DremiW: possibilita fazer consulta exploratórias, provendo uma estrutura de self-service BI com experimentação dos dados, interage entre a camada de visualização e as de baixo. Possui um *engine* de processamento próprio, e trabalha juntamente com o Apache Arrow e Arrow Flight, na parte de transformação e disponibilização dos dados.

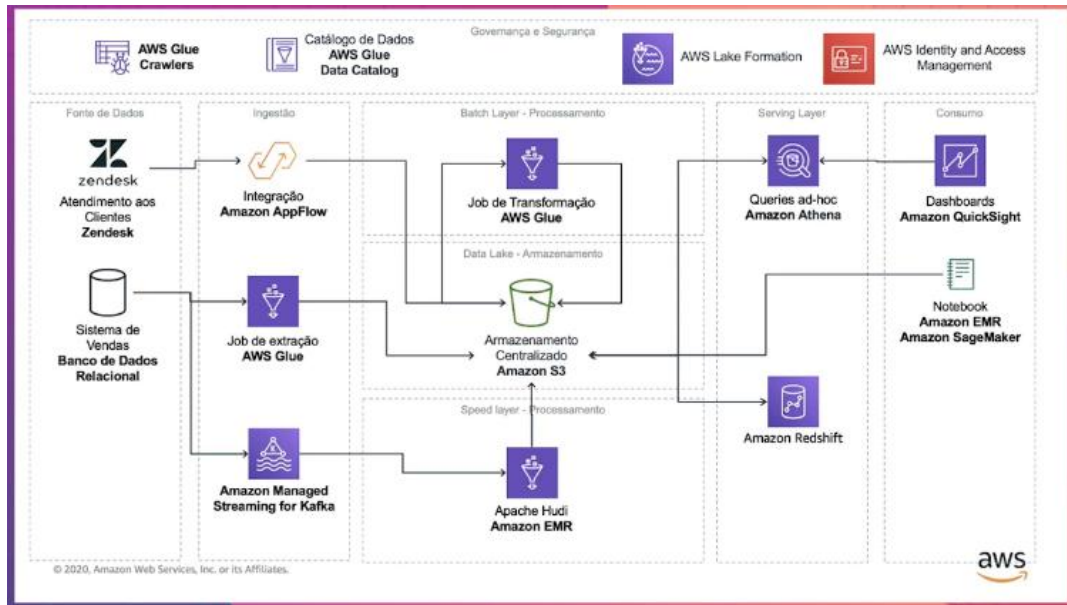
Spark: é uma ferramenta usada para processamento de grandes volumes de dados, com velocidade até cem vezes maior se comparado ao MapReduce.

Conforme visto nesta seção o Apache Hadoop apresenta diversos benefícios, como a possibilidade de rodar em máquinas problemáticas e mesmo assim assegurar a integridade dos dados.

4.2 Amazon AWS

Como *software* proprietário, será apresentada a AWS, Amazon Web Services Inc., que é uma plataforma de serviços de computação em nuvem da Amazon.

Figura 19 – Ecossistema AWS



AWS, 2021.

A AWS possui diversas ferramentas e também viabiliza o uso de suas próprias com outras ferramentas *open source*. Como o foco deste trabalho é mostrar como montar uma estrutura de *Data Lake*, serão apresentadas nesta seção as ferramentas que serão utilizadas para tal. Serão elas: Amazon S3, Glue, Athena e QuickSight. Todo conteúdo aqui descrito será baseado na documentação disponibilizada pela Amazon (AMAZON WEB SERVICES, 2021).

4.2.1 Amazon S3

Conhecido popularmente como S3, o Amazon Simple Storage Service, é a ferramenta de armazenamento para internet. Pode-se pensar nele como se fosse similar ao Drive da Google só que mais complexo. Com o S3 é possível armazenar dados de forma altamente escalonável, econômica, rápida e confiável. Através dessa ferramenta é possível armazenar, acessar e recuperar qualquer quantidade de dados de forma *online* e de maneira simples, uma vez que tudo é feito via interface gráfica. Como a própria AMAZON WEB SERVICES (2021) cita, o S3 foi intencionalmente planejado para funcionar com um conjunto mínimo de recursos com alta robustez e simplicidade.

Principais conceitos do S3 (AMAZON WEB SERVICES, 2021):

- **Buckets:** nada mais são do que containers para objetos armazenados no S3. Possuem diversos propósitos, dentre eles, pode-se destacar: a organização do namespace do S3 num nível mais alto, a identificação da conta responsável pelo armazenamento e pelas transferências de dados, e sua parte na gestão de acesso.

- **Objetos:** são as entidades fundamentais armazenadas no S3, compreendem os dados e metadados do objeto. Metadados são um conjunto de pares nome-valor que detalham o objeto, sendo o mesmo identificado por uma chave (nome) e um ID de versão.
- **Chaves:** a chave é o identificador exclusivo do objeto no bucket. A combinação do bucket, da chave e do ID de versão identificam de forma única cada objeto. Dito isto, pode-se concluir que cada objeto inserido no S3 por ser endereçado pela combinação do nome do bucket, da chave e ID de versão.
- **Regiões:** no ato da criação do bucket é possível escolher em qual região geográfica o S3 irá armazená-lo, de modo que só saíram de lá quando forem explicitamente transferidos para uma outra região.
- **Modelo de consistência de dados:** o S3 fornece forte consistência de leitura após a gravação dos objetos no bucket; Atualizações atômicas de uma única chave, sendo assim ao fazer uma alteração (PUT) numa chave existente em uma thread e executar um GET nessa mesma chave em outra thread de forma simultânea, o resultado será os dados antigos ou os dados novos, de modo que nunca será retornado dados parciais ou corrompidos. Além disso possui alta disponibilidade de replicação de dados, e sendo esta bem sucedida, a segurança no armazenamento é garantida.

4.2.2 AWS Glue

É um serviço de preparação de dados sem servidor para operações de extração, transformação e carregamento (ETL) (AMAZON WEB SERVICES, 2021). O Glue foi projetado para lidar com dados semiestruturados e pode ser usado para organizar, formatar, limpar e validar dados para armazenamento em um DW ou DL, nesse sentido ele auxilia simplificando diversas tarefas, tais como:

- Descoberta e catalogação de metadados sobre seus armazenamentos de dados em um catálogo central.
- Preenchimento do Glue Data Catalog com definições de tabelas de programas de crawler agendados.
- Escalonamento de recursos para execução de trabalhos.
- Suporte a erros e novas tentativas de forma automática.
- Viabiliza a criação de *jobs* para mover automaticamente seus dados para um DW ou DL.
- Agrega métricas de tempo de execução para monitoramento de atividades, tanto no DW quanto no DL.

- Auxilia na criação de pipelines ETL orientados a eventos, de modo que sempre que houver novos dados disponíveis no S3, ele irá executar um *job* conforme programado.

Outra contribuição é o entendimento sobre os ativos de dados, no sentido de que você armazená-los através de diversos serviços disponibilizados pela AWS e ainda assim manter uma visão unificadas dos seus dados através do Glue Data Catalog. A contribuição mais importante do Glue para este trabalho será a catalogação dos dados armazenados no S3 para que possamos consultá-los através do Athena.

4.2.3 Amazon Athena

O Amazon Athena é um serviço de consulta de dados que permite a utilização da linguagem SQL para tal, viabilizando a análise dos dados inseridos no S3. Além disso, ele auxilia na análise de dados variados, podendo estes ser estruturados, semiestruturados e não estruturados. Outro ponto relevante é que é um serviço que não utiliza servidor, de modo que não necessita de infraestrutura, configuração e gerenciamento da mesma, a Amazon ressalta que dessa maneira é possível manter o foco nos dados e não na infraestrutura. Esse serviço viabiliza consultas instantâneas, de maneira ágil, simples, com alta performance, de forma segura e com alta disponibilidade, bastando configurá-lo para apontar para os dados armazenados no S3, que é exatamente o será feito no próximo capítulo. Ainda é possível integrá-lo com o QuickSight a fim gerar relatórios ou explorar dados utilizando ferramentas de BI.

4.3 Equivalência entre as ferramentas

Visando facilitar a compreensão das diferenças e semelhanças entre esses dois conjuntos de aplicações foi montada a seguinte Tabela:

Tabela 4 – Equivalências das ferramentas do Hadoop e AWS

Funcionalidade	Software do Hadoop	Software da AWS
ingestão de dados	HDFS	S3
gestão e agendamento de jobs de execução	Askaban	Glue
estrutura de modelagem de dados (interface relacional - DW)	Hive	Glue
motor de processamento	Spark (+ veloz que o Tez)	Spark/Athena
ferramenta de extração de dados p/ DB relacionais	Scop	Glue
linguagem de script, executa sob o Yarn e o Tez	Pig	SQL/Python
ferramenta de cubo de dados, faz o pré-processamento dos dados (pega os dados do HBase ou Hive)	Kyle	QuickSight
faz a segurança dos dados	Apache Range	S3
possibilita consulta exploratórias	DremiW/Apache Atrial	Athena
ferramentas para visualização dos dados	Power BI/Zeplin/Jupyter	QuickSight

Autoria própria, 2021.

Após a conclusão do estudo acerca das ferramentas apresentadas, optou-se por seguir com a demonstração da estruturação de um DL em nuvem da Amazon, posto o acesso limitado a *hardware*, o que inviabiliza a criação de um cluster e posterior implementação com as ferramentas do Hadoop.

5 ESTRUTURAÇÃO DE UM AMBIENTE DE DATA LAKE NA NUVEM DA AMAZON

Posto que o objetivo deste trabalho é a criação e estruturação de um ambiente de *Data Lake*, sendo esta a maior contribuição em virtude do desafio tecnológico. Nesta seção será apresentado o cenário de criação desta estrutura para fins de demonstração. Por ser um serviço em nuvem, todas as ferramentas aqui apresentadas não necessitam de instalação. Conforme explicitado no capítulo anterior, os serviços da Amazon AWS que serão utilizados nesse experimento serão: Amazon S3, AWS Glue e Amazon Athena.

Figura 20 – Arquitetura da solução proposta



Autoria própria, 2021.

Na Figura acima é possível observar a arquitetura que será seguida no experimento do presente capítulo.

5.1 Cenário

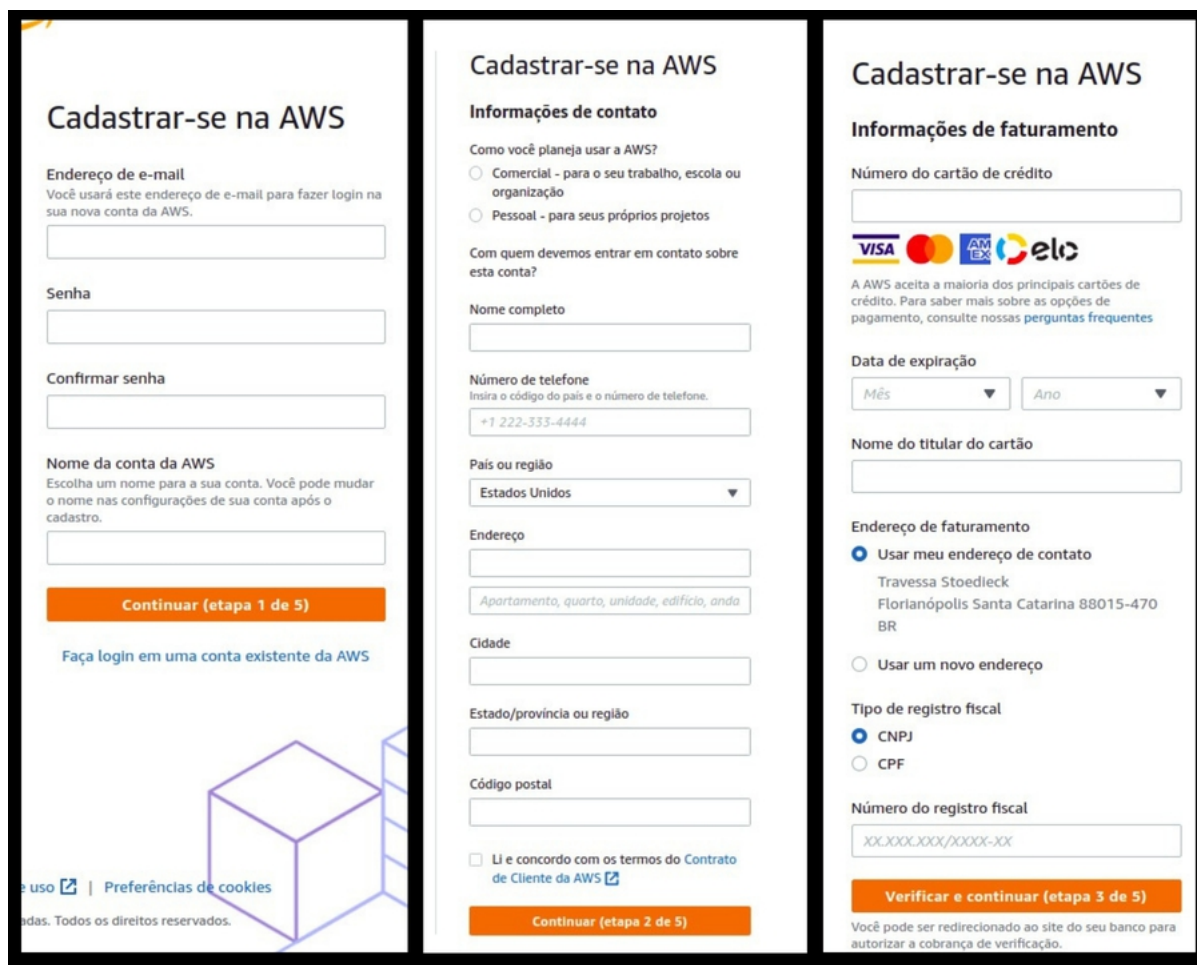
Foram consideradas duas bases de dados abertas para a demonstração, a primeira considerava dados financeiros abertos do governo e a segunda, dados sobre a pandemia do covid-19. Tendo em vista questões de segurança da informação no decorrer do experimento que não permitiram o acesso completo à base optou-se por fazer a demonstração utilizando os dados da pandemia da covid-19. Os dados foram coletados do site Coronavírus Brasil, montado e mantido pelo governo Federal. Uma consideração a ser feita é em relação aos dados coletados para este experimento, por ser uma demonstração de estruturação de um ambiente de data lake, não haveria motivos de se exceder o limite de memória disponibilizado pela Amazon na conta free, desse modo para o experimento foram utilizados dados referentes à região sul do país que, engloba os estados do Rio Grande do Sul, Santa Catarina e Paraná.

Assim, como a base de dados disponível foram iniciados os trabalhos. Para a estruturação deverá ser acessada a URL da provedora de ferramentas e serviços Amazon, na qual se criará uma conta, a fim de se obter acesso à todas as ferramentas desejadas. A partir disso se iniciará a configuração e estruturação do ambiente de *Data Lake*.

5.1.1 Criação de conta na Amazon

Para a criação da conta, deve-se acessar a URL da Amazon, <https://www.aws.amazon.com>, e clicar na opção “Crie uma conta na AWS”, feito isso, basta seguir o passo a passo para tal. Esse passo a passo é composto por 6 etapas, conforme é possível ver nas Figuras 21, 22 e 23.

Figura 21 – Parte 1 da criação de conta na AWS



The figure displays three sequential panels of the AWS account creation process:

- Panel 1: Cadastrar-se na AWS (Step 1 of 5)**
 - Endereço de e-mail:** Field for email address.
 - Senha:** Password field.
 - Confirmar senha:** Confirm password field.
 - Nome da conta da AWS:** Field for account name.
 - Continuar (etapa 1 de 5):** Orange button to proceed.
 - Faça login em uma conta existente da AWS:** Link for existing accounts.
- Panel 2: Cadastrar-se na AWS (Step 2 of 5)**
 - Informações de contato:**
 - Como você planeja usar a AWS?:** Radio buttons for "Comercial" or "Pessoal".
 - Com quem devemos entrar em contato sobre esta conta?:** Field for contact name.
 - Nome completo:** Full name field.
 - Número de telefone:** Phone number field.
 - País ou região:** Dropdown menu (currently "Estados Unidos").
 - Endereço:** Address field.
 - Cidade:** City field.
 - Estado/província ou região:** State/region field.
 - Código postal:** Zip code field.
 - Li e concordo com os termos do Contrato de Cliente da AWS:** Checkbox.
 - Continuar (etapa 2 de 5):** Orange button to proceed.
- Panel 3: Cadastrar-se na AWS (Step 3 of 5)**
 - Informações de faturamento:**
 - Número do cartão de crédito:** Field for credit card number.
 - Data de expiração:** Month and year dropdowns.
 - Nome do titular do cartão:** Field for cardholder name.
 - Endereço de faturamento:**
 - Usar meu endereço de contato:** Selected radio button.
 - Usar um novo endereço:** Radio button.
 - Tipo de registro fiscal:** Radio buttons for "CNPJ" (selected) or "CPF".
 - Número do registro fiscal:** Field for tax registration number.
 - Verificar e continuar (etapa 3 de 5):** Orange button to proceed.

Autoria própria, 2021, adaptado de AWS, 2021.

A Figura 21 mostra a parte inicial do cadastro, onde é preciso cadastrar e-mail, senha e nome da conta para posterior acesso; Além disso, é necessário o preenchimento das informações de contato. E o cadastramento de um cartão de crédito para eventuais despesas de armazenamento.

Figura 22 – Parte 2 da criação de conta na AWS

Cadastrar-se na AWS

Confirmar sua identidade

Antes que você possa usar a conta da AWS, é necessário confirmar seu número de telefone. Ao continuar, o sistema automatizado da AWS entrará em contato com você com um código de verificação.

Como devemos enviar para você o código de verificação?

☒ Mensagem de texto (SMS)

☐ Chamada de voz

Código do país ou região

Brasil (+55)

Número de telefone celular

Verificação de segurança

8yxc6x

Digite os caracteres como mostrado acima

Enviar SMS (etapa 4 de 5)

Cadastrar-se na AWS

Confirmar sua identidade

Verificar código

Continuar (etapa 4 de 5)

Autoria própria, 2021, adaptado de AWS, 2021.

Figura 23 – Parte 3 da criação de conta na AWS

A imagem mostra a interface de criação de conta na AWS, dividida em duas partes principais. A parte da esquerda, intitulada 'Cadastrar-se na AWS', apresenta a seção 'Selecionar um plano de suporte'. Abaixo do título, há um texto explicando que o usuário deve escolher um plano para sua conta comercial ou pessoal, com um link para 'Compare planos e exemplos de definição de preço'. Três opções de plano são listadas: 'Suporte Basic - gratuito' (selecionado), 'Suporte Developer - a partir de 29 USD/mês' e 'Suporte Business - a partir de 100 USD/mês'. Cada opção tem uma lista de benefícios e um ícone representativo. Abaixo das opções, há uma pergunta 'Precisa de suporte de nível Enterprise?' com uma explicação e um link 'Saiba mais'. Um botão laranja 'Concluir cadastramento' está no rodapé. A parte da direita, intitulada 'Parabéns!', mostra uma mensagem de agradecimento, uma barra amarela com o texto 'Acesse o Console de Gerenciamento da AWS' e um link para 'Cadastre-se para outra conta ou entre em contato com a equipe de vendas'.

Cadastrar-se na AWS

Selecionar um plano de suporte

Escolha um plano de suporte para sua conta comercial ou pessoal. [Compare planos e exemplos de definição de preço](#). Você pode alterar seu plano a qualquer momento no Console de Gerenciamento da AWS.

- ☒ **Suporte Basic - gratuito**
 - Recomendado para novos usuários que estão começando a usar a AWS
 - Acesso por autoatendimento 24 horas por dia/7 dias por semana a todos os recursos da AWS
 - Apenas para questões de conta e faturamento
 - Acesso ao Personal Health Dashboard e ao Trusted Advisor
- ☐ **Suporte Developer - a partir de 29 USD/mês**
 - Recomendado para desenvolvedores que estão avaliando a AWS
 - Acesso por e-mail ao AWS Support durante o horário comercial
 - Tempo de resposta de 12 horas (horário comercial)
- ☐ **Suporte Business - a partir de 100 USD/mês**
 - Recomendado para executar cargas de trabalho de produção na AWS
 - Suporte técnico 24 horas por dia/7 dias por semana por e-mail, telefone e chat
 - Tempo de resposta de 1 hora
 - Conjunto completo de recomendações de melhores práticas do Trusted Advisor

Parabéns!

Agradecemos por se cadastrar na AWS.

Estamos ativando a sua conta, o que levará alguns minutos. Você receberá um e-mail quando o processo for concluído.

[Acesse o Console de Gerenciamento da AWS](#)

[Cadastre-se para outra conta ou entre em contato com a equipe de vendas](#)

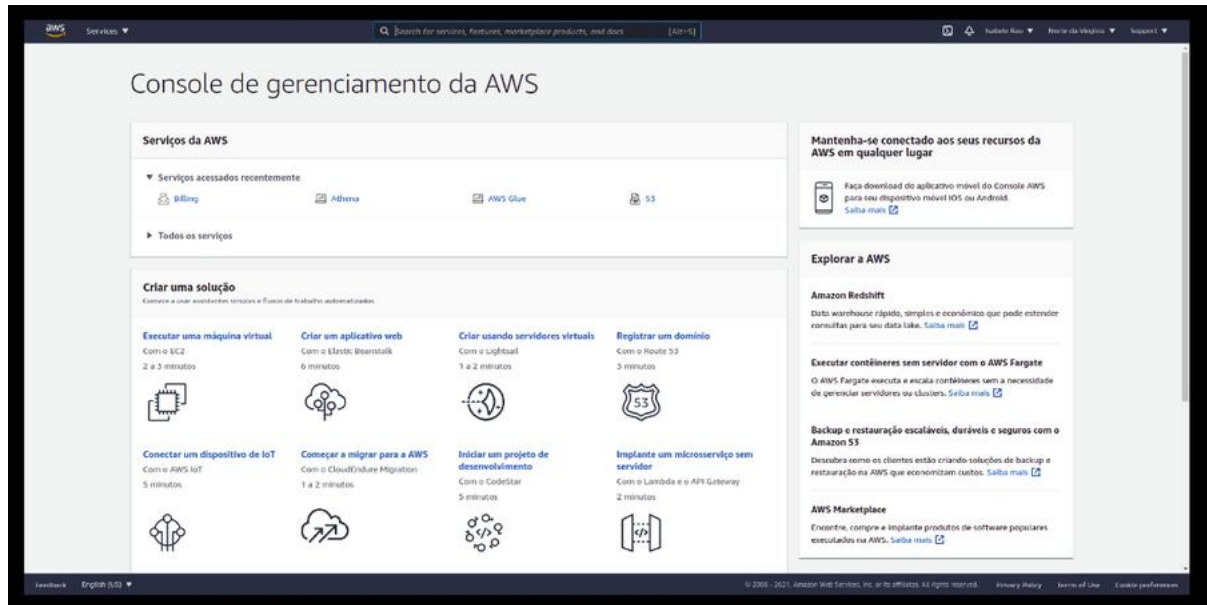
Autoria própria, 2021, adaptado de AWS, 2021.

Nas Figuras 22 e 23 é possível observar as partes 2 e 3 da criação da conta, elas contemplam a confirmação de identidade, sendo esta feita através do número de celular, com envio de código verificador por mensagem de texto ou chamada de voz. Além disso, o passo final é a escolha do tipo de conta, há três opções, a escolha dependerá da necessidade de cada um, para o experimento foi feita uma conta gratuita.

5.1.2 Configuração do Amazon S3

Para que seja possível iniciar, é necessário acessar a conta recém criada, para isso basta logar-se. A Figura 24 mostra o painel geral que se visualiza logo após o *login*.

Figura 24 – Painel de gerenciamento de serviços da AWS



Autoria própria, 2021, adaptado de AWS, 2021.

Agora é possível iniciar a estruturação do ambiente de DL, para isso usaremos a ferramenta S3, detalhada na seção 4.2.1. Ao acessá-la através do painel ou através do botão “Services”, criaremos um *bucket*, é ele que irá receber os dados que desejamos armazenar.

Figura 25 – Painel e criação de buckets do S3 da AWS

The image shows the AWS S3 console interface. The top section displays a list of existing buckets with columns for Name, Region, Access, and Creation Date. Below this, the 'Criar bucket' (Create bucket) wizard is shown, divided into several steps: 'Configuração geral' (General configuration), 'Configurações de bloqueio do acesso público deste bucket' (Public access blocking), 'Versionamento de bucket' (Bucket versioning), 'Tags' (Tags), 'Criptografia padrão' (Default encryption), and 'Configurações avançadas' (Advanced configurations). The 'Criar bucket' button is highlighted in orange at the bottom right.

Nome	Região da AWS	Acesso	Data de criação
aws-athena-query-results-us-east-1-334456897040	Leste dos EUA (Norte da Virgínia) us-east-1	Os objetos podem ser públicos	16 May 2021 06:29:50 PM -03
aws-glue-scripts-334456897040-us-east-1	Leste dos EUA (Norte da Virgínia) us-east-1	Os objetos podem ser públicos	16 May 2021 03:58:31 PM -03
aws-glue-temporary-334456897040-us-east-1	Leste dos EUA (Norte da Virgínia) us-east-1	Os objetos podem ser públicos	16 May 2021 03:58:31 PM -03
data-lake-tcc	Leste dos EUA (Norte da Virgínia) us-east-1	Bucket e objetos não públicos	16 May 2021 03:36:45 PM -03

Criar bucket

Os buckets são contêineres para dados armazenados no S3. Saiba mais

Configuração geral

Nome do bucket
mynewbucket

O nome do bucket deve ser exclusivo e não deve conter espaços ou letras maiúsculas. Consulte as regras de nomenclatura de bucket

Região da AWS
Leste dos EUA (Norte da Virgínia) us-east-1

Copiar configurações do bucket existente - opcional
Somente as configurações de bucket na configuração a seguir são copiadas.
Escolher bucket

Configurações de bloqueio do acesso público deste bucket

O acesso público é concedido a buckets e objetos por meio de listas de controle de acesso (ACLs), políticas de bucket, políticas de ponto de acesso ou todas elas. Para garantir que o acesso público a este bucket e todos os seus objetos seja bloqueado, ative a opção de Bloquear todo o acesso público. Essas configurações serão aplicadas apenas a este bucket e aos respectivos pontos de acesso. A AWS recomenda ativar a opção Bloquear todo o acesso público. Porém, antes de aplicar qualquer uma dessas configurações, verifique se as aplicações funcionarão corretamente sem acesso público. Caso precise de algum nível de acesso público a este bucket ou aos objetos que ele contém, é possível personalizar as configurações individuais abaixo para que atendam aos seus casos de uso de armazenamento específicos. Saiba mais

☒ **Bloquear todo o acesso público**
Ativar essa configuração é o mesmo que ativar todas as quatro configurações abaixo. Cada uma das configurações a seguir são independentes uma da outra.

☐ Bloquear acesso público a buckets e objetos concedidos por meio de novas listas de controle de acesso (ACLs)
Ativar essa configuração impedirá as permissões de acesso público aplicadas a buckets ou objetos novos, atualizados e existentes a partir do momento em que for criada.

Versionamento de bucket

O versionamento é um meio de manter múltiplas variantes de um objeto no mesmo bucket. Você pode usar o versionamento para preservar, recuperar e restaurar todas as versões de cada objeto armazenado no bucket do Amazon S3. Com o versionamento, você pode recuperar facilmente ações não intencionais do usuário e falhas de aplicação. Saiba mais

Versionamento de bucket
☒ Desativar
☐ Ativar

Tags (0) - opcional

Acompanhe o custo de armazenamento ou outros critérios marcando seu bucket. Saiba mais

Nenhuma tag associada a este bucket.
Adicionar tag

Criptografia padrão

Criptografar automaticamente novos objetos armazenados neste bucket. Saiba mais

Criptografia no lado do servidor
☒ Desativar
☐ Ativar

Configurações avançadas

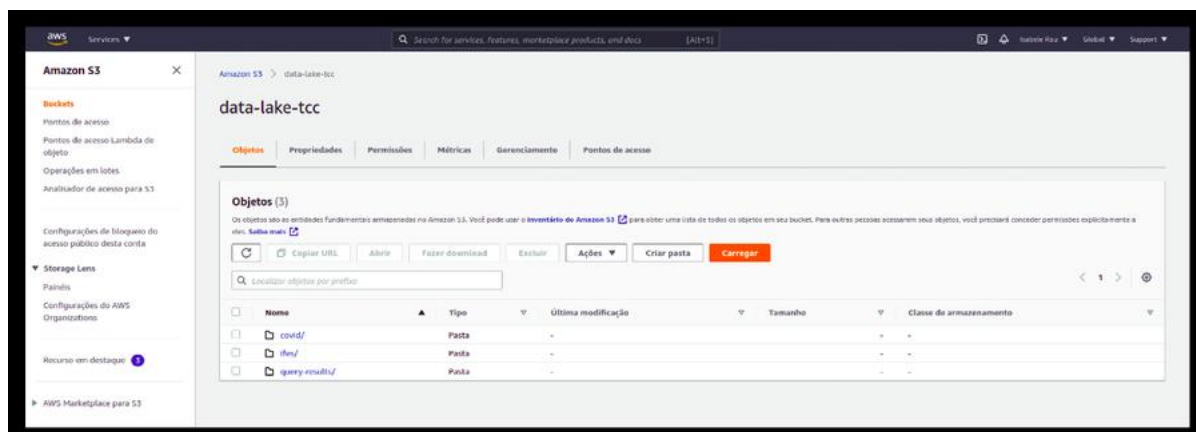
Depois de criar o bucket, você pode fazer upload de arquivos e pastas para o bucket e definir configurações adicionais do bucket.

Cancelar **Criar bucket**

Autoria própria, 2021, adaptado de AWS, 2021.

A Figura 25 mostra o painel para configuração do S3 e a tela para criação do *bucket*, é um processo simples e intuitivo, basta escolher um nome que faça sentido e que esteja disponível para o *bucket* e as demais configurações podem continuar com as opções *default*. Quando o *bucket* for criado, irá aparecer na listagem que aparece na parte superior da Figura 24. Ao clicar em cima do nome do *bucket* você será direcionado para a tela que permite a visualização e gerenciamento da estrutura de diretórios do *bucket*, conforme mostra a Figura 26.

Figura 26 – Visualização da estrutura de diretórios no bucket do S3 da AWS



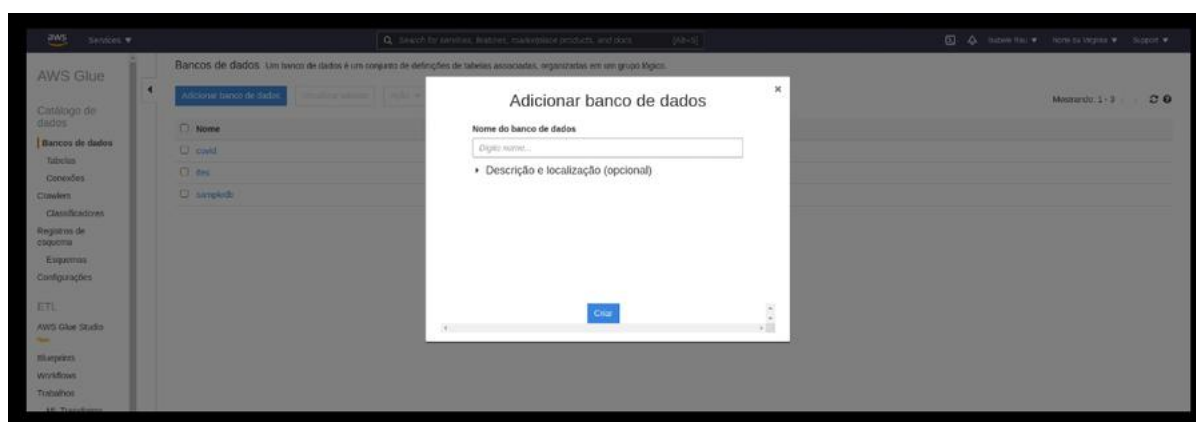
Autoria própria, 2021, adaptado de AWS, 2021.

Na aba de Objetos é possível criar diretórios para uma maior organização do *bucket*. Na imagem acima observam-se três diretórios: 'covid', 'ifex' e 'query-results'. Conforme citado no Capítulo 2.4, a inserção de dados pode ser feita de duas maneiras, manualmente ou via *script*. Observe na Figura 26 o botão “Carregar”, é através dele que é possível fazer o carregamento manual de dados, para o experimento faremos de forma manual e, posteriormente de forma automatizada. Para fazer manualmente basta clicar em “Carregar”, que você será direcionado para uma página onde é possível fazer o *upload* de arquivos.

5.1.3 Configuração do AWS Glue

Uma vez que os dados estiverem no S3, precisaremos mapeá-los, para isso devemos adicionar um banco de dados, conforme mostrado na Figura abaixo:

Figura 27 – Adição de banco de dados no Glue da AWS

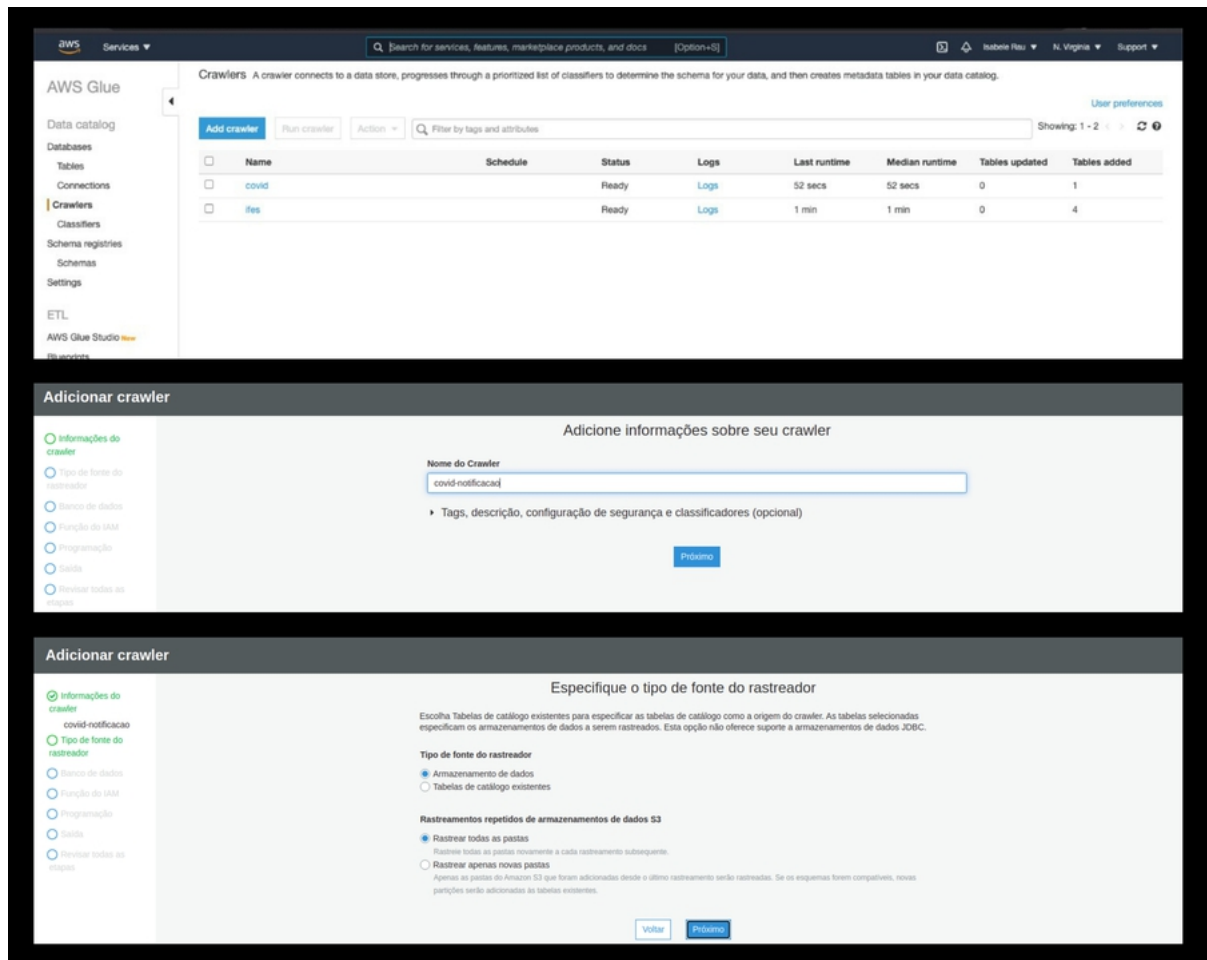


Autoria própria, 2021, adaptado de AWS, 2021.

Agora é preciso configurar uma estrutura chamada *crawler*. Segundo a definição da própria AMAZON WEB SERVICES (2021), o *crawler* se conecta a um *datastore*, passa por

uma lista prioritária de classificadores para determinar o esquema dos seus dados e, em seguida, cria tabelas de metadados em seu catálogo de dados, em outras palavras o Glue irá mapear os dados por meio do seu catálogo de dados. Observe como é feita a criação de um *crawler* a seguir:

Figura 28 – Parte 1 de criação de crawler no Glue da AWS



Autoria própria, 2021, adaptado de AWS, 2021.

Na Figura 28 visualiza-se o painel principal onde é possível ver todos os *crawlers* já criados e acima deste painel, encontra-se o botão que possibilita adicionarmos um novo *crawler*. Ainda na Figura 28 podemos ver os dois primeiros passos, inicialmente damos um nome para o *crawler*, o tipo de código do *crawler* e a repetição deles para os *datastores* do S3.

Figura 29 – Parte 2 de criação de crawler no Glue da AWS

The figure consists of three screenshots of the AWS Glue console, showing the steps to create a crawler. Each screenshot has a sidebar on the left with the following menu items: 'Informações do crawler' (selected), 'covid-notificacao', 'Tipo de fonte do rastreador', 'Armazenamento de dados', 'Banco de dados', 'Função do IAM', 'Programação', 'Valida', and 'Revisar todos os passos'.

Screenshot 1: Adicionar crawler - Adicione um datastore

Escolha um datastore

S3

Conexão

Selecione uma conexão

[Adicionar conexão](#)

Rastrear dados no

☒ Caminho especificado na minha conta
☐ Caminho especificado em outra conta

Incluir caminho

s3://data-lake-tcc/covidnotificacao

Todos os pastas e arquivos contidos no caminho de inclusão são rastreados. Por exemplo, digite s3://MyBucket/MyFolder/ para rastrear todos os objetos no MyFolder em MyBucket.

Tamanho da amostra (opcional)

Insira um número inteiro entre 1 e 249.

Este campo define o número de arquivos em cada pasta folha a ser rastreada. Se não for definido, todos os arquivos serão rastreados.

[Excluir padrões \(opcional\)](#)

[Voltar](#) [Próximo](#)

Screenshot 2: Adicionar crawler - Escolha uma função do IAM

Escolha uma função do IAM

A função do IAM permite que o crawler para executar e acessar os datastores do Amazon S3.

☐ Atualize uma política em uma função do IAM
☒ Escolha uma função do IAM existente
☐ Crie uma função do IAM

Função do IAM

AWSGlueServiceRole-iam

Essa função deve fornecer permissões semelhantes a política gerenciada pela AWS, **AWSGlueServiceRole**, mais acesso aos seus datastores.

• s3://data-lake-tcc/covid/notificacao

Você também pode criar uma função do IAM no [console do IAM](#).

[Voltar](#) [Próximo](#)

Screenshot 3: Adicionar crawler - Criar uma programação para este crawler

Criar uma programação para este crawler

Frequência

Executar sob demanda

[Voltar](#) [Próximo](#)

Autoria própria, 2021, adaptado de AWS, 2021.

Já na Figura 29 vemos mais três passos para criação dele, adicionamos um *datastore*, uma conexão, o IAM role e a frequência em que esse *crawler* deverá ser rodado.

Figura 30 – Parte 3 de criação de crawler no Glue da AWS

Adicionar crawler

Configure o crawler de saída

Banco de dados **covid**

Adicionar banco de dados

Prefixo adicionados a tabelas (opcional)

Digite um prefixo adicionado aos nomes da tabela

► Agrupar comportamento de dados do S3 (opcional)

► Opções de configuração (opcional)

Voltar Próximo

Adicionar crawler

Informações do crawler

Nome covid-notificacao

Tags -

Datastores

Banco de dados S3

Incluir caminho s3: // data-lake-tcc / covid / notificacao

Conexão

Excluir padrões

Função do IAM

Função do IAM am: aws: iam :: 334456897040: role / service-role / AWSGlueServiceRole-iam

Programação

Programação Executar sob demanda

Saída

Banco de dados covid

Prefixo adicionados a tabelas (opcional)

Crie um único esquema para cada caminho do S3 falso

► Opções de configuração

Voltar Concluir/Próximo

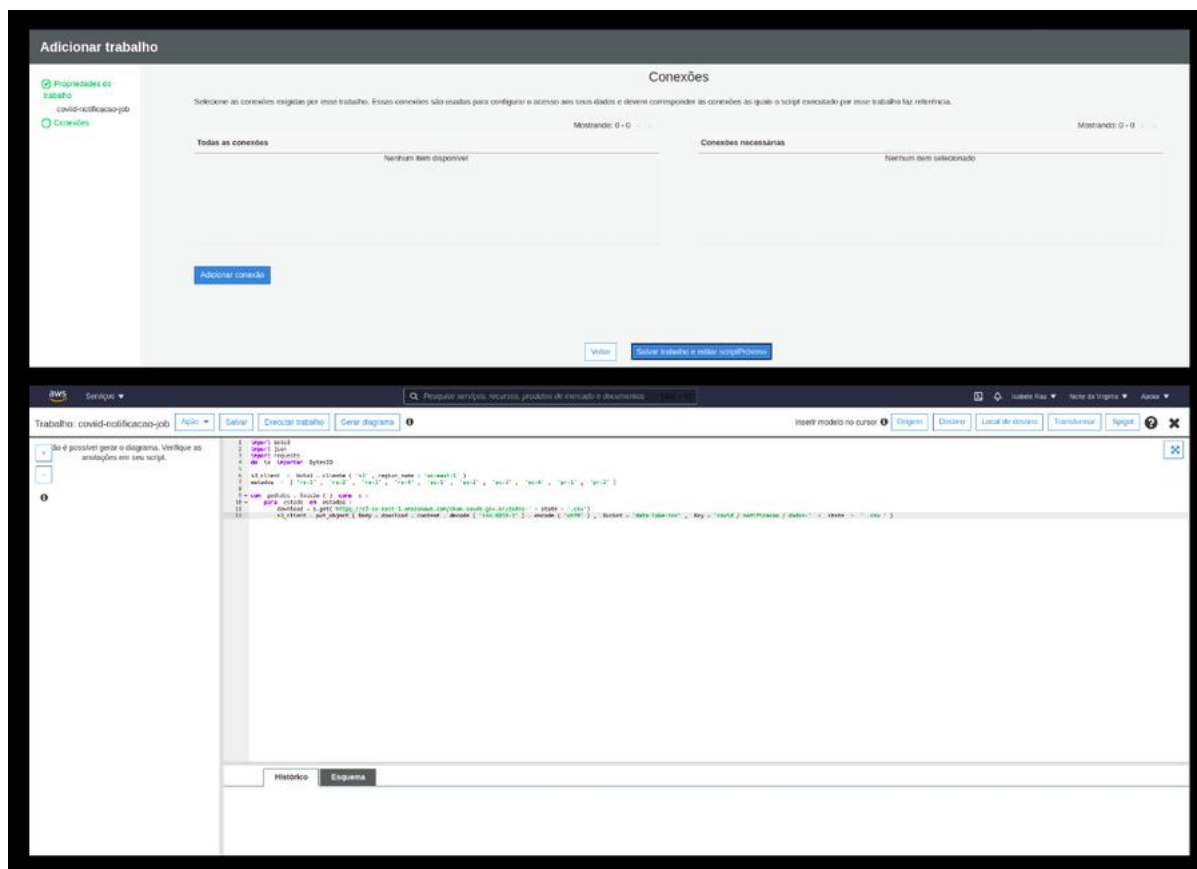
Autoria própria, 2021, adaptado de AWS, 2021.

Na Figura 30 visualizamos os dois passos finais para a criação do *crawler*, configuramos a base de dados à qual o *crawler* deverá atualizar e revisamos parte das configurações anteriores.

Essa estrutura do *crawler* vai até o diretório onde estão os arquivos, em nosso cenário no *bucket* “data-lake-tcc,” lê o conteúdo dos arquivos, entende a estrutura deles e define um mapeamento para essa estrutura de dados num formato de tabela. Para o cenário do trabalho, o *crawler* foi configurado para criar o banco de dados chamado “covid” e uma tabela chamada “notificacao”. A estrutura da tabela possui a mesma estrutura dos arquivos inseridos nesse banco e, se desejarmos é possível editar essa estrutura.

Para automatizar o armazenamento de dados no S3 criado anteriormente, podemos utilizar uma função conhecida como trabalho (*job*) no Glue, ela é a lógica responsável pelo ETL (*Extract, transformation and loading*). Para a criação do trabalho (*job*) basta clicar no botão “Adicionar trabalho” no painel central de “Trabalhos” e preencher as propriedades necessárias, como mostram as Figuras 31 e 32.

Figura 32 – Parte 2 da criação de um job no Glue na AWS



Autoria própria, 2021, adaptado de AWS, 2021.

Para o cenário compreendido neste trabalho, foi criado um trabalho (*job*) em cima da ferramenta Spark, citada na seção 4.1 em ferramentas do Apache Hadoop, em Python. Na parte inferior da Figura 32 já é possível criar o roteiro (*script*) que faça a ingestão e atualização dos dados no S3 de forma automatizada.

Logo após a criação deste trabalho (*job*), ele passará a aparecer no painel principal, conforme mostrado na parte superior da Figura 31. Neste mesmo painel, ao clicar em cima do nome do trabalho (*job*), abre-se um painel na parte inferior da tela, onde é possível visualizar o histórico de execução do mesmo e, entre outras coisas, é possível editar o roteiro (*script*) criado anteriormente.

Para o cenário deste trabalho foi criado um roteiro (*script*) em *Python* que executa o *download* dos arquivos da base de dados do governo e faz o *upload* no *bucket* criado anteriormente no S3. Uma observação a ser feita neste ponto, é que foi feita uma modificação na codificação dos dados, pois ao visualizá-los, percebeu-se que havia um problema na codificação de alguns caracteres, mas este passo pode não se faz obrigatório.

Figura 33 – Painel de trabalho e roteiro no Glue na AWS

The figure consists of two screenshots of the AWS Glue console interface. The top screenshot shows the 'Trabalhos' (Jobs) page, which lists jobs with columns for Name, Modelo, Linguagem do ETL, Local do script, Última modificação, and Marcador de trabalho. The job 'covid-notificacao-job' is selected. The bottom screenshot shows the 'Roteiro' (Script) tab for the same job, displaying a PySpark script. The script includes comments in Portuguese and code for reading data from a CSV file and writing it to a database.

Trabalhos

Nome	Modelo	Linguagem do ETL	Local do script	Última modificação	Marcador de trabalho
<input checked="" type="checkbox"/> covid-notificacao-job	Fagulha	Pido	s3://aws-glue-scripts-33...	16 maio 2021 5:29 PM UTC-3	Desabilitar
<input type="checkbox"/> ites-load-csv	Fagulha	Pido	s3://aws-glue-scripts-33...	16 maio 2021 4:00 PM UTC-3	Desabilitar

Roteiro

```

1: # Script para carregar dados do CSV para o banco de dados
2: # Nome do job: covid-notificacao-job
3: # Descrição: Carregar dados do CSV para o banco de dados
4: # Autor: [Nome do Autor]
5: # Data: [Data de Criação]
6: # Versão: 1.0
7: #
8: # Configurações de conexão
9: # Nome da conexão: covid-notificacao-conn
10: # Tipo de conexão: JDBC
11: # Driver: org.apache.hadoop.hive.jdbc.HiveDriver
12: #
13: # Configurações de leitura
14: # Nome do arquivo: covid-notificacao.csv
15: # Formato: CSV
16: #
17: # Configurações de escrita
18: # Nome da tabela: covid-notificacao
19: # Formato: CSV
20: #
21: # Execução
22: #
23: #
24: #
25: #
26: #
27: #
28: #
29: #
30: #
31: #
32: #
33: #
34: #
35: #
36: #
37: #
38: #
39: #
40: #
41: #
42: #
43: #
44: #
45: #
46: #
47: #
48: #
49: #
50: #
51: #
52: #
53: #
54: #
55: #
56: #
57: #
58: #
59: #
60: #
61: #
62: #
63: #
64: #
65: #
66: #
67: #
68: #
69: #
70: #
71: #
72: #
73: #
74: #
75: #
76: #
77: #
78: #
79: #
80: #
81: #
82: #
83: #
84: #
85: #
86: #
87: #
88: #
89: #
90: #
91: #
92: #
93: #
94: #
95: #
96: #
97: #
98: #
99: #
100: #

```

Histórico

ID da execução	Novas tentativas	Status de execução	Erro	Resultado	Histórico	Logs de erro	Versão cola	Capacidade máxima	Acionado por	Horário de início	Horário de término	Tempo de inicialização	Tempo de execução	Tempo limite	Atraso	Entrada de execução de trabalho
<input type="radio"/> p_3a5f5a72394940e...	-	Bem sucedido	-	-	Histórico	Logs de erro	2.0	10	-	16 maio ...	16 maio ...	6s	2 minutos	2880 min.	-	s3://aws-glue-electric...
<input type="radio"/> p_7f8b6d67e1e2a0962...	-	Bem sucedido	-	-	Histórico	Logs de erro	2.0	10	-	16 maio ...	16 maio ...	7s	2 minutos	2880 min.	-	s3://aws-glue-electric...
<input type="radio"/> p_5e0e6920525503c...	-	Bem sucedido	-	-	Histórico	Logs de erro	2.0	10	-	16 maio ...	16 maio ...	7s	22s	2880 min.	-	s3://aws-glue-electric...

Autoria própria, 2021, adaptado de AWS, 2021.

A Figura 33 mostra o painel central de trabalhos (*jobs*) e o trabalho (*job*) 'covid-notificacao-job' selecionado. Logo abaixo é possível ver as abas referentes a esse trabalho (*job*) e o roteiro (*script*) criado. Este trabalho (*job*) pode ser rodado manualmente ou por gatilhos (*trigger*), através de agendamento. Para rodá-lo manualmente basta selecioná-lo no painel, clicar em Ação > Executar trabalho. Já, se quiser que ele rode de tempos em tempos, é preciso criar um gatilho (*trigger*). Veja como nas Figuras 34 e 35:

Figura 34 – Parte 1 da criação de um gatilho (trigger) no Glue na AWS

The figure consists of two screenshots from the AWS Glue console, showing the initial steps of creating a trigger.

Screenshot 1: Adicionar gatilho (Add trigger)

The left sidebar shows the navigation menu with options: **Propriedades do gatilho** (selected), **Trabalhos a serem iniciados**, and **Revisar todas as etapas**.

The main area is titled **Configurar as propriedades do gatilho** (Configure trigger properties). It contains the following fields:

- Nome**: A text input field with the placeholder "Digite um nome para o gatilho..." (Enter a name for the trigger...).
- Tags (opcional)**: A section for adding optional tags.
- Tipo do gatilho**: Radio buttons for **Programado** (selected), **Eventos de trabalho**, and **Sob demanda**. Below this, a note states: "Escolha Programado para ativar o gatilho em um tempo específico, Eventos de trabalho para ativar o gatilho quando eventos de trabalho correspondem à sua lista monitorada ou Sob demanda para ativar o gatilho imediatamente quando iniciado."
- Frequência**: A dropdown menu currently set to **Dia a dia** (Daily).
- Hora de início (UTC)**: A dropdown menu set to **21**.
- Minuto inicial**: A dropdown menu set to **52**.
- Próximo**: A blue button to proceed to the next step.

Screenshot 2: Adicionar gatilho (Add trigger)

The left sidebar shows the navigation menu with options: **Propriedades do gatilho**, **Trabalhos a serem iniciados** (selected), and **Revisar todas as etapas**.

The main area is titled **Selecionar trabalhos para o gatilho** (Select jobs for the trigger). It contains the following elements:

- Escolha trabalhos para iniciar quando o gatilho é acionado.** (Choose jobs to start when the trigger is triggered.)
- Todos os trabalhos** (All jobs): A table listing available jobs.

Trabalho	Ações
covid-notificacao-job	Adicionar
aws-lead-csv	Adicionar
- Trabalhos para iniciar** (Jobs to start): A table listing selected jobs.

Trabalho	Ações
covid-notificacao-job	X
- Os parâmetros passados para o trabalho covid-notificacao-job quando iniciado** (Parameters passed to the covid-notificacao-job job when triggered). A note below states: "(Opcional) adicione parâmetros para substituir os parâmetros padrão fornecidos para este trabalho quando iniciado por este gatilho." (Optional) add parameters to replace the default parameters provided for this job when triggered by this trigger.

Configuração de segurança	
Perfil	None
Marcação de trabalho	
Tempo limite do trabalho (minutos)	2000
Limite de notificação de atraso (minutos)	
Chave	Valor
- Botões**: **Voltar** (Back) and **Próximo** (Next) buttons.

Autoria própria, 2021, adaptado de AWS, 2021.

Figura 35 – Parte 2 da criação de um gatilho (trigger) no Glue na AWS

The screenshot shows the final step of the AWS Glue console, titled **Revisar** (Review).

The left sidebar shows the navigation menu with options: **Propriedades do gatilho** (selected), **Trabalhos a serem iniciados**, and **Revisar todas as etapas**.

The main area displays a summary of the trigger configuration:

- Propriedades do gatilho** (Trigger properties):

Nome	Tags	Tipo do gatilho	Programação
covid-notificacao-trigger	-	Programado	Az 21:52
- Trabalhos a serem iniciados** (Jobs to be started):

Trabalhos
covid-notificacao-job
- Botões**: **Voltar** (Back) and **Concluir/Próximo** (Finish/Next) buttons.

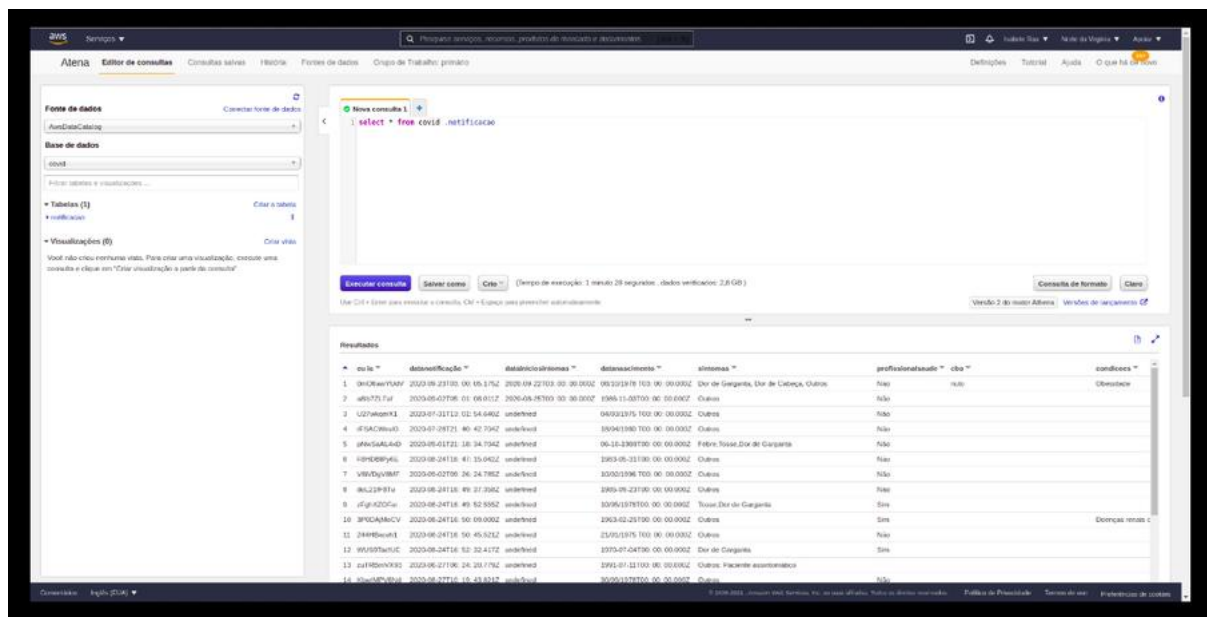
Autoria própria, 2021, adaptado de AWS, 2021.

Para criar um gatilho (*trigger*) basta nomeá-la, escolher a frequência de execução e escolher qual trabalho (*job*) ela deverá executar. Dessa maneira ela irá executar o trabalho (*job*) de tempos em tempos, conforme o agendamento.

5.1.4 Consultas através do Athena

Após essa sequência de passos documentada nas seções anteriores deste Capítulo, é possível utilizar um outro serviço disponibilizado pela AWS, chamado Amazon Athena. Através dele é possível fazer consultas, baseadas em linguagem SQL, nas tabelas criadas pelo Glue. Ele mapeia automaticamente os bancos de dados da conta e ao acessar já é possível fazer consultas.

Figura 36 – Exibição de consulta no Athena na AWS



Autoria própria, 2021, adaptado de AWS, 2021.

A Figura 36 mostra a consulta e o resultado da mesma, à tabela notificação, criada através dos serviços do Glue ao acessar a base de dados do governo em relação aos dados de notificação da covid-19.

Concluindo-se estas etapas, já se tem uma estrutura de DL pronta para uso. Posto isso, o fluxo para a criação de um DL seria:

- 1) Injetar os dados no S3, e;
- 2) Mapear esses dados como metadados, utilizando o Glue, e;
- 3) Indexar a consulta desses metadados através do Athena, utilizando os metadados configurados pelo Glue.

Essa demonstração contempla o objetivo de criação e estruturação de um ambiente de *Data Lake*, conforme exposto no Capítulo 1, seção 1.2.2.

A partir deste momento é possível utilizar outra ferramenta, ainda não explorada neste trabalho, chamada QuickSight. Na Figura abaixo é possível observar três painéis gerados através dela.

6 CONCLUSÃO

Inicialmente, são apresentados os conceitos básicos dos assuntos tratados nessa monografia, tais conhecimentos puderam ser resgatados de disciplinas, como Fundamentos de Banco de Dados e Sistemas de Apoio a Decisão, ambas ministradas neste curso.

Com todo conteúdo disponibilizado no presente trabalho, foi possível desmistificar as principais diferenças entre *Data Warehouses* e *Data Lakes*, trazendo às claras que essas duas tecnologias abordam diferentes propostas e suas diferenças se destacam em relação à tamanho, flexibilidade e colaboração.

Conclui-se que o *Data Lake* viabiliza o armazenamento dos dados na íntegra e possibilita processá-los sob demanda de forma escalável. De modo que facilita a automação de processos e inovação baseando-se em dados e assim impulsionando a transformação digital nas empresas. É válido ressaltar que para que seja atribuído sentido aos dados inseridos no DL faz-se necessário a análise por um profissional especializado como um engenheiro ou cientista de dados. Posto isso, DL se mostra como uma solução dinâmica e econômica e que alinha a empresa com as tendências de mercado.

Já *Data Warehouse*, por sua vez, tem como princípio a disponibilização de visões organizadas que possibilitem o direcionamento das tomadas de decisões com base nos dados. Pode-se dizer que no DW os dados relevantes para o negócio são centralizados e sistematizados para que se tornem ponto de apoio na criação de estratégias de mercado. Foi possível concluir que é uma abordagem que necessita altos investimentos, em decorrência da necessidade de planejamento arquitetural e modelagem antes mesmo da ingestão de dados.

Posto isso, ficou evidente que os contextos em que devem ser utilizadas uma ou outra arquitetura irá depender de alguns fatores, como: quais fontes de informações estão disponíveis, como está disposta a organização dos dados, qual a necessidade da empresa no uso destes e, qual a capacidade de investimento da mesma. E nota-se que uma abordagem muito comum é combinar a utilização de ambas tecnologias, garantindo assim aumento de produtividade, maior precisão nas análises e otimização dos custos.

No decorrer do trabalho, também foi possível conhecer dois *frameworks* disponíveis para a implementação de um *Data Lake*, são eles: Apache Hadoop e Amazon AWS. Em virtude da falta de infraestrutura necessária para a estruturação de um *Data Lake* utilizando as ferramentas do Apache Hadoop, foi escolhido o *framework* da Amazon para tal finalidade, visto que sua implementação é feita utilizando a *cloud* da própria Amazon, necessitando pagamento apenas quando extrapolado o limite máximo oferecido.

Durante a estruturação do ambiente de *Data Lake* na nuvem da Amazon, houveram alguns imprevistos, como a restrição de acesso à base de dados financeiros do IFES - Instituto Federal do Espírito Santo, posto que na criação do *bucket* a configuração foi feita na Região AWS da Virgínia, o que provavelmente causou esta restrição de acesso, desta forma, foi escolhida uma outra base para a demonstração. A base escolhida então, foi a de dados referente a pandemia da covid-19 no

Brasil. Uma outra limitação encontrada no decorrer do experimento foi a limitação do volume de armazenamento na nuvem da Amazon, uma vez que a memória ofertada gratuitamente possui um limite de 5GB, desse modo, o job criado não pôde ser rodado através de agendamentos, para que não excedesse o volume e gerasse custos de armazenamento. O maior desafio encontrado nessa estruturação foi a criação do *script* em *Python* que tinha por objetivo fazer a busca automatizada das informações contidas na base de dados e atualizar o repositório de dados no S3, o que foi concluído e rodado algumas vezes.

Por fim, conclui-se que é de suma importância entender a real necessidade da empresa em relação à análise de dados, bem como sua capacidade de investimentos para assim direcionar qual tecnologia utilizar. Para trabalhos futuros fica como sugestão seguir o estudo sobre ferramentas de análise de dados.

REFERÊNCIAS

- ABERDEEN. **Angling for insight in today's data lake**. 2017. Disponível em: <https://s3-ap-southeast-1.amazonaws.com/mktg-apac/Big+Data+Refresh+Q4+Campaign/Aberdeen+Research+-+Angling+for+Insights+in+Today's+Data+Lake.pdf>. Acesso em: 05 de jul. de 2020.
- ABES. **Setor de TI cresce no Brasil 10,5% em 2019**. 2020. Disponível em: <https://abessoftware.com.br/wp-content/uploads/2020/10/ABES-EstudoMercadoBrasileirodeSoftware2020.pdf>. Acesso em: 21 de jul. de 2020.
- AMAZON WEB SERVICES. **AWS Documentation**. 2021. Disponível em: <https://docs.aws.amazon.com/index.html>. Acesso em: 17 de maio de 2021.
- ASAAD, R. R.; AHMAD, H. B.; ALI, R. I. A Review: Big Data Technologies with Hadoop Distributed Filesystem and Implementing M/R. **Academic Journal of Nawroz University (AJNU)**, p. 25 – 33, 2021. Disponível em: <https://bit.ly/3sVYnFP>. Acesso em: 17 de fev. de 2020.
- BREWER, E. **Towards Robust Distributed Systems**. 2000. Disponível em: https://sites.cs.ucsb.edu/~rich/class/cs293-cloud/papers/Brewer_podc_keynote_2000.pdf. Acesso em: 29 de jul. de 2020.
- BRITO, J. J. **Data Warehouses in the era of Big Data**: efficient processing of Star Joins in Hadoop. 2018. 161 p. Tese (Doutorado em Ciência da Computação e Matemática Computacional) — Universidade de São Paulo.
- CÔRREA, T. da S.; ALMEIDA, D. E. C. de; GRAÇA NETO, A. F. Comparação entre banco de dados relacional e não relacional em arquitetura distribuída. In: INSTITUTO NACIONAL DE TELECOMUNICAÇÕES, 2017. **III Seminário de Desenvolvimento Mobile e Cloud Computing e Conectividade**. 2017. p. 1 – 7. ISSN 2447-2352. Disponível em: <file:///home/isabele/Downloads/Compara%C3%A7%C3%A3o%20entre%20banco%20de%20dados%20relacional%20e%20n%C3%A3o%20relacional%20em%20arquitetura%20distribu%C3%ADa.pdf>. Acesso em: 30 de jul. de 2020.
- DATE, C. **Introdução a sistemas de bancos de dados**. 8. ed. Rio de Janeiro: Elsevier Brasil, 2004. 865 p.
- DAVENPORT, T. H.; PRUSAK, L. **Conhecimento empresarial**. 14. ed. Rio de Janeiro: Elsevier, 2014.
- DIXON, J. **Pentaho, Hadoop, and Data Lakes**. 2010. Disponível em: <https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/>. Acesso em: 10 de jul. de 2020.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Education do Brasil, 2011.
- ENDEAVOR. **Metadados: conheça os principais agentes da mais recente revolução da web**. 2015. Disponível em: <https://endeavor.org.br/tecnologia/metadados/>. Acesso em: 17 de maio de 2021.

FERNANDES, A. de S. **Introdução aos conceitos de NoSQL, BASE vs ACID e o Teorema CAP**. 2013. Ppt.

GRAY, P.; ISRAEL, C. The Data Warehouse Industry. **UC Irvine: Center for Research on Information Technology and Organizations.**, 1999. Disponível em: <https://escholarship.org/uc/item/1hp1k5m7>. Acesso em: 05 de jul. de 2020.

HAYES, F. **The Story So Far**. 2002. Disponível em: <https://www.computerworld.com/article/2588199/the-story-so-far.html>. Acesso em: 21 de jul. de 2020.

HOJI, E. Y. **Melhoria de um sistema de data warehouse em uma empresa de telefonia móvel**. 2012. 95 p. Monografia (Engenharia de Produção) — Universidade de São Paulo. Disponível em: <http://pro.poli.usp.br/trabalho-de-formatura/melhoria-de-um-sistema-de-data-warehouse-em-uma-empresa-de-telefonia-movel/>. Acesso em: 12 de ago. de 2020.

INMON, W. H. **Building the Data Warehouse**. 3. ed. Nova York: Wiley, 2002. 428 p. Disponível em: <http://fit.hcmute.edu.vn/Resources/Docs/SubDomain/fit/ThayTuan/DataWH/Bulding%20the%20Data%20Warehouse%204%20Edition.pdf>. Acesso em: 30 de jul. de 2020.

INTERNATIONAL DATA CORPORATION. **Worldwide Enterprise Performance Management Software Forecast Update, 2020–2024**. 2020. Disponível em: <https://www.idc.com/getdoc.jsp?containerId=prUS46794720>. Acesso em: 08 de jul. de 2020.

IPSENSE. **Tudo o que você precisa saber sobre o ecossistema Hadoop!** 2019. Artigo online. Disponível em: <https://www.ipsense.com.br/blog/tudo-o-que-voce-precisa-saber-sobre-o-ecossistema-hadoop/>. Acesso em: 17 de maio de 2021.

KHINE, P. P.; WANG, Z. S. Data lake: a new ideology in big data era. In: **4th Annual International Conference on Wireless Communication and Sensor Network**. [s.n.], 2018. v. 17, n. 03025. Disponível em: <https://doi.org/10.1051/itmconf/20181703025>.

KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling**. 3. ed. Indianapolis: Wiley, 2013. 608 p.

LANEY, D. **3D Data Management: Controlling Data Volume, Velocity, and Variety**. 2001. Online. Disponível em: <https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>. Acesso em: 30 de março de 2020.

LAUDON, K. C.; LAUDON, J. P. **Sistemas de informação com internet**. 4. ed. [S.l.]: LCT, 1999. 389 p.

MILOSLAVSKAYA, N.; TOLSTOY, A. Big Data, Fast Data and Data Lake Concepts. In: **7th Annual International Conference on Biologically Inspired Cognitive Architectures**. ELSEVIER, 2016a. v. 88, p. 300 – 305. Disponível em: <https://www.researchgate.net/publication/306941043>. Acesso em: 08 de jul. de 2020.

MILOSLAVSKAYA, N.; TOLSTOY, Alexander. Big Data, Fast Data and Data Lake Concepts. **Procedia Computer Science**, ELSEVIER, v. 88, p. 300 – 305, 2016b. ISSN 1877-0509. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050916316957>. Acesso em: 06 de abril de 2020.

- MINAYO, M. C. de S. **O desafio do conhecimento: Pesquisa qualitativa em saúde**. 10. ed. São Paulo: HUCITEC, 2014. v. 46. 406 p.
- MORESI, E. A. D. Delineando o valor do sistema de informação de uma organização. In: **Ci. Inf.** [online]. [s.n.], 2000. v. 29, p. 14 – 24. ISSN 1518-8353. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652000000100002&lng=pt&nrm=iso.
- MORGADO, A. C. **Data Warehouses Espaciais: Projeto e implementação**. 2013. 82 p. Dissertação (Engenharia Informática) — Universidade do Minho. Disponível em: <https://core.ac.uk/download/pdf/55628291.pdf>. Acesso em: 08 de jul. de 2020.
- MOTA, A. D. M. *et al.* **Universidade e Ciência**. Palhoça: Unisul Virtual, 2016. Disponível em: https://www.uaberta.unisul.br/repositorio/download/web/OA_100_anos_manifesto_estudantil_cordoba/universidade_ciencia.pdf. Acesso em: 29 de jul. de 2020.
- NOGUEIRA, I. D.; ROMDHANE, M. Modeling Data Lake Metadata with a Data Vault. **22nd International Database Engineering & Applications Symposium**, Villa San Giovanni, p. 253 – 261, Jun 2018. Disponível em: <https://arxiv.org/abs/1807.04035>. Acesso em: 17 de maio de 2021.
- O'BRIEN, J. A. **Sistemas de informação: e as decisões gerenciais na era da internet**. [S.l.]: Saraiva, 2004. 431 p.
- OLIVEIRA, D. de Pinho Rebouças de. **Sistemas de informações gerenciais: estratégicas, táticas, operacionais**. 17. ed. São Paulo: Atlas, 2018. 314 p.
- ROSINI, A. M.; PALMISANO, A. **Administração de sistemas de informação e gestão do conhecimento**. São Paulo: Pioneira Thomson Learning, 2003. 219 p.
- SÁ, J. V. de Oliveira e. **Metodologia de Sistemas de Data Warehouse**. 2009. 400 p. Dissertação (Tecnologias e Sistemas de Informação) — Universidade do Minho. Disponível em: <http://repositorium.sdum.uminho.pt/handle/1822/10663>. Acesso em: 29 de jul. de 2020.
- SALESFORCE BRASIL. **Data Warehouse e Data Lake: O que são?** 2020. Artigo online. Disponível em: [DataWarehouseeDataLake:Oques~ao?](https://www.salesforce.com/pt-br/data/data-warehouse/data-lake/) Acesso em: 07 de abril de 2020.
- SAS. **Big Data**. 2019. Disponível em: https://www.sas.com/pt_br/insights/big-data/what-is-big-data.html. Acesso em: 02 de julho de 2020.
- SILVA, E. L. da; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4. ed. [s.n.], 2005. 139 p. Disponível em: https://projetos.inf.ufsc.br/arquivos/Metodologia_de_pesquisa_e_elaboracao_de_teses_e_dissertacoes_4ed.pdf. Acesso em: 30 de jul. de 2020.
- SILVA, H. M. da. **Sociedade da informação**. 2007. Disponível em: http://www.profcordella.com.br/unisanta/textos/tgs21_dados_info_conhec.htm. Acesso em: 08 de ago. de 2020.
- SOMASUNDARAM, G.; SHRIVASTAVA, A. **Armazenamento e Gerenciamento de Informações: Como armazenar, gerenciar e proteger informações digitais**. Porto Alegre: Bookman, 2011. 461 p.

STATISTA. **Volume of data/information created worldwide from 2010 to 2024**. 2020. Disponível em: <https://www.statista.com/statistics/871513/worldwide-data-created/#:~:text=The%20total%20amount%20of%20data,reaching%2059%20zettabytes%20in%202020>. Acesso em: 04 de jul. de 2020.

TAURION, C. **Big Data**. 1ª. ed. Rio de Janeiro: Brasport, 2015. 184 p. ISBN 9788574527277.

THE APACHE SOFTWARE FOUNDATION. **Apache Hadoop 3.2.2**. 2021. Disponível em: <https://hadoop.apache.org/docs/stable/index.html>. Acesso em: 17 de maio de 2021.

Apêndices

APÊNDICE A

Dada a importância do assunto para o presente trabalho, é aqui abordada a conceituação de *Big Data*. Segundo Miloslavskaya e Tolstoy (2016b), Big Data pode ser definido como um conjunto de dados extremamente amplos e que excedem os recursos das ferramentas de programação tradicionais.

Este conceito ganhou força no início dos anos 2000, quando o analista Laney (2001) articulou a atual definição de Big Data como os 3V's publicando no Gartner: Big Data são ativos de informações de alto volume, alta velocidade e/ou alta variedade que exigem formas inovadoras e econômicas de processamento de informações que permitem uma visão aprimorada na tomada de decisão e automação de processos.

Com o passar do tempo essa conceituação passou de 3 para 5 V's, como é possível observar na Figura:

Figura 38 – Resumo dos 5 V's do Big Data



Autoria própria (2021), adaptado de SAS Institute Inc. Cary, NC, USA, 2019.

Conforme publicado pelo SAS Institute Inc. Cary, a definição dos 5 V's, (SAS, 2019):

- **Volume:** as organizações coletam dados de várias fontes, incluindo transações comerciais, dispositivos inteligentes (IoT), equipamentos industriais, vídeos, mídias sociais, entre outros. No passado, armazená-los teria sido um problema - mas o armazenamento mais barato em plataformas como *Data Lake* e *Hadoop* diminuiu a carga.
- **Velocidade:** Com o crescimento da Internet das Coisas, os dados passam para as empresas a uma velocidade sem precedentes e devem ser tratados em tempo hábil. Exemplo: Tags RFID, sensores e medidores inteligentes estão aumentando a necessidade de lidar com esses torrents de dados em, quase, tempo real.
- **Variedade:** os dados são fornecidos em todos os tipos de formatos - de dados numéricos estruturados em bancos de dados tradicionais a documentos de texto não estruturados, como e-mails, vídeos, áudios, dados de cotações de ações e transações financeiras.
- **Variabilidade:** além das crescentes velocidades e variedades de dados, os fluxos de dados são imprevisíveis - mudando frequentemente e variando bastante. É um desafio, mas as empresas precisam saber quando algo está em alta nas mídias sociais e como gerenciar o pico diário, sazonal e de pico de carga de dados acionada por eventos.
- **Veracidade:** refere-se à qualidade dos dados. Como os dados vêm de muitas fontes diferentes, é difícil vincular, corresponder, limpar e transformar dados nos sistemas. As

empresas precisam conectar e correlacionar relacionamentos, hierarquias e múltiplas ligações de dados. Caso contrário, seus dados podem rapidamente sair do controle.

Para Taurion (2015), esse conjunto de tecnologias, processos e práticas permitem às empresas analisarem dados a que antes não tinham acesso e tomar decisões ou mesmo gerenciar atividades de forma muito mais eficiente.

Conforme proposto pelo SAS (2019), a importância do *Big Data* não gira em torno da quantidade de dados que você possui, mas do que você faz com eles. Nesse sentido algumas das possíveis vantagens a serem alcançadas com o uso de *Big Data* em empresas, são:

- diagnosticar causas de falhas em processos interno em tempo real
- redução de custos
- otimização de processos internos
- aumento de produtividade
- redução de churn (cancelamentos de contratos, perda de clientes)
- elaboração de ações de marketing assertivas
- melhoria e personalização de serviços e produtos

Big Data integra várias técnicas de várias disciplinas como bancos de dados e *Data Warehouses*, estatísticas, aprendizado de máquina, alto desempenho reconhecimento de padrões, redes neurais, visualização de dados, recuperação de informações, imagem e processamento de sinais e análise de dados espaciais ou temporais (MILOSLAVSKAYA; TOLSTOY, 2016b).

Por fim, vale destacar um apontamento feito por Miloslavskaya e Tolstoy (2016b) em relação a TI tradicional e a TI de *Big Data*, no que tange suas diferenças. Na TI tradicional os mecanismos de processamento estão no centro dos processos, enquanto na TI de *Big Data* os mecanismos de processamento devem ser construídos no decorrer dos fluxos, de modo constante e de maneira contínua.