CENTRO UNIVERSITÁRIO CARIOCA – UNICARIOCA

BRUNO MEDEIROS DE SOUZA

# ENGENHARIA DE DADOS E SEUS DESAFIOS NO AMBIENTE CORPORATIVO

Rio de Janeiro

2022

BRUNO MEDEIROS DE SOUZA

# ENGENHARIA DE DADOS E SEUS DESAFIOS NO AMBIENTE CORPORATIVO

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção de Bacharel em Ciência da Computação pelo Centro Universitário Carioca – UNICARIOCA

Orientado por: Prof. Marcelo Perantoni.

Rio de Janeiro

2022.2

S719e Souza, Bruno Medeiros de
Engenharia de dados e seus desafios no ambiente corporativo / Bruno Medeiros de Souza. – Rio de Janeiro, 2022.
46 f.

Orientador: Prof. Marcelo Perantoni
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro Universitário UniCarioca - Rio de Janeiro, 2022.

1. Engenharia de dados. 2. Big Data. 3. Data Lake. 4. Ingestão de dados. 5. Cloud Computing. 6. Computação em Nuvem. 7. Tratamento de Dados. 8. Banco Central do Brasil. 9. Data Science. I. Perantoni Marcelo, Prof. Orient. II.Título.

CDD 005.10681

Bruno Medeiros de Souza

Engenharia de dados e seus desafios no ambiente corporativo

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção de Bacharel em Ciência da Computação pelo Centro Universitário Carioca – UNICARIOCA.
Rio de Janeiro, 30/11/2022.

Banca Examinadora

_____

Prof. André Luiz Avelino Sobral, D.Sc – Coordenador
Centro Universitário Carioca (Unicarioca)

_____

Prof. Marcelo Perantoni, D.Sc – Orientador
Centro Universitário Carioca (Unicarioca)

_____

Prof. Daisy Cristine Albuquerque da Silva – Convidado
Centro Universitário Carioca (Unicarioca)

*“Os campeões não são os que sempre vencem as corridas, são os campeões quem sai e tenta. Ser campeão é um estado de espírito.”*

(Simon Sinek)

## Agradecimentos

Agradeço primeiramente a Deus, pela minha vida, e por me ajudar a ultrapassar todos os obstáculos encontrados ao longo da minha jornada.

A minha mãe, Ana Gláucia, ao meu pai José de Souza e a minha esposa, Mariana Simões, agradeço por todo o incentivo nesses anos de estudos. Todo o apoio dado por eles foi fundamental para obtenção de sucesso e para que eu alcançasse todos meus objetivos. Agradeço por sempre confiarem e acreditarem em mim. Agradeço por compreenderem a minha ausência enquanto eu me dedicava à realização desse trabalho. Agradeço a minha irmã Stéphany Medeiros, por todo apoio e incentivo aos meus estudos e por sempre ser uma referência de dedicação na minha família.

Agradeço a minha terapeuta Jorgenete Silva por todo o apoio psicológico durante esse semestre, que me ajudou com diversas técnicas que me ajudaram a destravar a escrita e a me organizar para a construção desse trabalho.

Agradeço ao meu amigo Vinicius Vicente, que contribuiu com conhecimentos técnicos e incentivos para que eu percebesse na área de dados a construção da minha carreira profissional.

Agradeço aos professores e a instituição pela dedicação, empenho e conhecimento passado ao longo desse curso, e um agradecimento especial ao professor e orientador Marcelo Perantoni, por todos os conselhos, ajuda e correções que me permitiram apresentar um melhor desempenho no processo da minha formação acadêmica. Agradeço a paciência com a qual guiou o meu aprendizado.

## Resumo

A popularidade dos assuntos das áreas de dados e sua importância para o crescimento das empresas tem elevado a demanda por profissionais cada vez mais capacitados, aumentando a valorização desses profissionais no mercado de trabalho. Afinal, é fundamental para as empresas tomar decisões estratégicas com segurança, e para tal, é preciso ter um bom núcleo de dados, que construídos por diversos profissionais da área, cada um com a sua respectiva função. Um levantamento realizado pela GeekHunter, empresa de consultoria no ramo de recrutamento de profissionais de tecnologia, mostra um crescimento de 310% nas vagas abertas na área de tecnologia e que de acordo com dados do Banco Mundial, até 2024 estima-se a criação de aproximadamente 420 mil vagas na área de dados. Ainda de acordo com o levantamento, a profissão cientista de dados, será uma das grandes apostas para os próximos anos. Mas, segundo o Data Sciencie Academy a demanda por Engenheiros de Dados, pode ser ainda maior que a demanda por Cientistas de Dados, pois muitas empresas consideram ideal ter pelo menos dois Engenheiros de dados trabalhando ao lado de cada Cientista de Dados e algumas empresas estimam que essa proporção pode ser ainda maior. Devido à dificuldade que os alunos têm se encontrar com o mercado de trabalho, não compreender as proporções desses números e a dificuldade para encontrar material necessário para direcionar os estudos e sua preparação, esse trabalho foi construído. Ao longo do trabalho é possível compreender os principais conceitos que introduzem o profissional na área de dados, aprender um pouco sobre a profissão Engenheiro de Dados, alguns dos conhecimentos técnicos necessários, como, por exemplo, Python e SQL, a técnica de Web Scraping, Ingestão de Dados. Sobre as ferramentas, foi escolhida as principais ferramentas utilizadas na nuvem da Microsoft para processamento e tratamento de dados, como, por exemplo, Azure Functions, Data Factory e Databricks. E, por fim, a implementação de um caso de uso comum para o engenheiro de dados, que vai desde a extração de dados externos de uma API pública, tratamento e ingestão desses dados para o ambiente interno de uma empresa.


Palavras-chave: Engenharia de dados, Big Data, Data Lake, Ingestão de dados, Cloud Computing, Computação em Nuvem, Tratamento de Dados, Banco Central do Brasil, Data Science.

## Abstract

The popularity of subjects in the data areas and their importance for the growth of companies has increased the demand for increasingly qualified professionals, increasing the appreciation of these professionals in the job market. After all, it is essential for companies to make strategic decisions safely, and for that, it is necessary to have a good core of data, which was built by several professionals in the area, each with their respective function. A survey carried out by GeekHunter, a consulting firm in the field of recruiting technology professionals, shows a growth of 310% in open positions in the technology area and that, according to data from the World Bank, by 2024 it is estimated that approximately 420,000 vacancies in the data area. Still according to the survey, the data scientist profession will be one of the big bets for the coming years. But, according to the Data Science Academy, the demand for Data Engineers may be even greater than the demand for Data Scientists, as many companies consider ideal to have at least two Data Engineers working alongside each Data Scientist and some companies estimate that proportion may be even greater. Due to the difficulty that students have in finding the job market, not understanding the proportions of these numbers and the difficulty in finding the necessary material to direct their studies and their preparation, this work was constructed. Throughout the work it is possible to understand the main concepts that introduce the professional to the data area, learn a little about the Data Engineer profession, some of the necessary technical knowledge, such as, for example, Python and SQL, the Web Scraping technique, Data Ingestion. About the tools, the main tools used in the Microsoft cloud for data processing and treatment were chosen, such as, for example, Azure Functions, Data Factory and Databricks. And, finally, the implementation of a common use case for the data engineer, which ranges from extracting external data from a public API, processing and ingesting this data into a company's internal environment.



Keywords: Data Engineering, Big Data, Data Lake, Data Ingestion, Cloud Computing, Data Processing, Central Bank of Brazil, Data Science.

**Sumário**

## Lista de figuras

# 1. Introdução

## 1.1. Apresentação do tema e o contexto

Os assuntos relacionados às áreas de dados se tornaram cada vez mais populares e são de extrema importância para as mais modernas plataformas de serviços disponíveis hoje, como a Google, Facebook, Spotify, Uber, entre outras. É por esse motivo que essas empresas são as maiores do mundo em seus seguimentos de mercado. O desenvolvimento significativo de algumas dessas empresas e a transformação causadas por elas em setores tão consolidados e tradicionais, como o Spotify na indústria da música ou a Uber no segmento de transportes, se deve a importância que essas companhias dão as áreas de dados. (LEWIS GAVIN, 2020a).

Parte da causa desse grande desenvolvimento das empresas atribui-se ao trabalho das áreas de dados como, por exemplo, engenharia de dados e ciência de dados. As pesquisas em universidades, vagas divulgadas em plataformas de empregos, palestras e grande parte da mídia dão muito foco na divulgação sobre a área de ciência de dados e sua importância para as corporações; porém, a profissão de Engenheiro de dados é também muito importante e possui um papel fundamental para o trabalho de *Big Data*.

Engenheiro de dados é uma profissão que está em rápido crescimento e um dos grandes desafios das empresas é encontrar bons profissionais com os conhecimentos necessários para assumir essa posição. Conforme Joe Reis e Matt Housley (2022, p. 3), “A profissão Engenheiro de Dados entrou em foco junto com o surgimento da ciência de dados na década de 2010”. É uma profissão relativamente nova, com diversas ramificações e muitas possibilidades, e por não existir um currículo comum sobre o assunto, poucos treinamentos de conceitos estão disponíveis para o aprendizado exigido para entrar no mercado de trabalho, além de ser pouco citada nas universidades, que é a base de fornecimento de profissionais para o mercado de trabalho.

Todo esse desafio citado sobre encontrar profissionais qualificados acontece pelo aumento da competitividade entre as empresas para obter esses talentos ou retê-los, o que eleva muito o salário e a supervalorização desses profissionais. Segundo o G1, 2021a, uma pesquisa feita pela empresa HRTECH, do ramo de recrutamento

digital, somente no primeiro semestre de 2021, houve um crescimento de 485% na abertura de vagas para engenheiro, analista e cientista de dados; além de toda essa procura, as médias salariais das posições de engenharia de dados, apuradas pela pesquisa fica entre R$ 7.625,00 e R$ 11.125,00 e ficam entre R$ 15.166,00 e R$ 17.166,00 para cargos de especialistas ou líderes.

## 1.2. Problema a ser respondido

Devido o déficit de profissionais de engenharia de dados disponíveis no mercado de trabalho, todo o desafio da profissão, conhecimento necessário para se inserir no mercado de trabalho que foram citados no tópico 1.1 e essa alta no mercado de dados, muitos estudantes que estão na faculdade, ou pessoas que querem migrar de profissão, se sentem na dificuldade de saber por onde começar, o que devem estudar e qual o espoco dessas funções nas áreas de dados que possuem diversas funções, níveis de conhecimentos e necessidades diferentes para cada uma delas.

## 1.3. Objetivos do TCC

O Objetivo principal desse trabalho é apresentar sobre a profissão engenheiro de dados que está sendo altamente demandada pelas empresas, tendo em vista que possui uma importância estratégica na operação de empresas que querem cada vez mais aplicar a cultura de *data driven*, e ainda é uma profissão tão pouco divulgada e ensinada nos núcleos de estudos.

Considerando toda a dificuldade de se definir um currículo comum para o direcionamento das pessoas que desejam iniciar seus estudos e se inserir no mercado de trabalho de dados, esse projeto foi pensado e desenvolvido para passar pelos principais pilares e conceitos que envolvem a profissão de engenheiro de dados, citar as principais linguagens de programação necessárias, mostrar algumas das principais ferramentas utilizadas na  cloud da Microsoft e, por fim, mostrar um exemplo prático de um desenvolvimento bem comum feito por esses profissionais.

## 1.4. Estrutura do trabalho

Ao longe desse trabalho, serão abordados conceitos e fundamentos que envolvem a engenharia de dados, conhecimentos técnicos necessários, ferramentas principais de engenheiros de dados especializados em Microsoft e a implementação de um exemplo ingestão de dados.

No capítulo 2 são apresentados os conceitos principais da profissão de Engenheiro de dados e sua importância na área de dados, é apresentado o conceito de computação em nuvem que é o local onde normalmente esse profissional aplica suas implementações. Neste capítulo também é apresentado o conceito de Big Data e o crescimento da utilização de dados, a principal plataforma para o trabalho de *Big Data*, conhecido como *Data Lake* e suas zonas.

No capítulo 3, são apresentadas as duas principais linguagens de programação que são necessárias para se tornar um engenheiro de dados (SQL e Python) e suas aplicações que podem ser implementadas em qualquer plataforma e/ou nuvem. Devido à grande quantidade de ferramentas e nuvens existentes para o fluxo de trabalho do engenheiro de dados, nesse capítulo, foram escolhidas apenas as principais ferramentas da plataforma de cloud da Microsoft, com foco em 3 ferramentas, uma para cada tipo de conceito, e são elas: Azure Functions para ter uma ferramenta de codificação e baixo custo, que consiga trabalhar com uma baixa volumetria; Azure Data Factory, uma ferramenta para trabalhar com altos volumes de dados, sem a necessidade de codificação, Databricks para ter mostrar uma ferramenta de codificação que consegue trabalhar com grandes volumes de dados.

No capítulo 4 é apresentado um fluxo de ingestão de dados utilizando o conceito de Web Scraping para conexão na API do Banco Central do Brasil, extraindo os principais índices de crescimento do país e armazenando em um Data Lake. O desenvolvimento visa passar pelos principais conceitos citados no capítulo 2 e 3, e a utilização de uma das ferramentas exemplificadas. A aplicação foi desenvolvida na linguagem de programação python, e sua automatização foi feita na ferramenta Azure Funtions.

## 2. Conceitos e fundamentos da engenharia de dados

Com a grande capacidade e necessidade que as empresas têm de coletar grandes quantidades de dados, houve a necessidade de se ter pessoas especializadas nas tecnologias certas, com as habilidades corretas para garantir que esses dados sejam organizados e disponibilizados para analistas e cientistas de dados da melhor maneira possível.

Engenharia de dados é uma área muito importante na área de dados e tecnologia, e por um bom motivo, é o profissional responsável por criar a base de dados para a área de ciência de dados e análise em produção. Este capítulo explora o que é a engenharia de dados, sua função dentro da corporação, habilidades e conhecimentos que esse profissional deve ter e alguns conceitos que são exigidos para as entregas desse profissional.

### 2.1. Data Engineering

Em novembro de 2022, uma rápida pesquisa no Google por “Engenharia de dados” apresenta um total de 33.900.000 de resultados exclusivos para o termo e isso demonstra o quanto de conteúdo que é possível encontrar para aprender um pouco mais sobre o assunto. Mas antes de definirmos o que é a engenharia de dados, aqui estão alguns exemplos de especialistas da área e suas definições para essa função:

> Em relação às funções existentes anteriormente, o campo de engenharia de dados pode ser pensado como um superconjunto de inteligência de negócios e armazenamento de dados que traz mais elementos da engenharia de software. Essa disciplina também integra a especialização em torno da operação dos chamados sistemas distribuídos de “big data”, juntamente com conceitos em torno do ecossistema Hadoop estendido, processamento de fluxo e computação em escala. (Maxime Beauchemin, 2017a).

> A engenharia de dados é um conjunto de operações que visa criar interfaces e mecanismos para o fluxo e acesso de informações. São necessários especialistas dedicados — engenheiros de dados —

> para manter os dados para que permaneçam disponíveis e possam ser usados por outras pessoas. Em suma, os engenheiros de dados configuram e operam a infraestrutura de dados da organização, preparando-a para análise posterior por analistas de dados e cientistas. (ALEXSOFT, 2021a).

> A engenharia de dados tem tudo a ver com movimentação, manipulação e gerenciamento de dados. (LEWIS GAVIN, 2020a).

É possível encontrar muitas respostas em poucas pesquisas e é natural fazer confusão ou não compreender o conceito, mas, em geral, as respostas sobre engenharia de dados detalham que é a área responsável pela construção de sistemas que são capazes de extrair, transformar, armazenar e analisar dados em grande ou pequena escala, conforme pode ser observado na Figura 1, que mostra o ciclo de vida da Engenharia de Dados. O principal objetivo de um engenheiro de dados é transformar toda essa grande quantidade de dados em indicadores para a gestão utilizar na tomada de decisão.

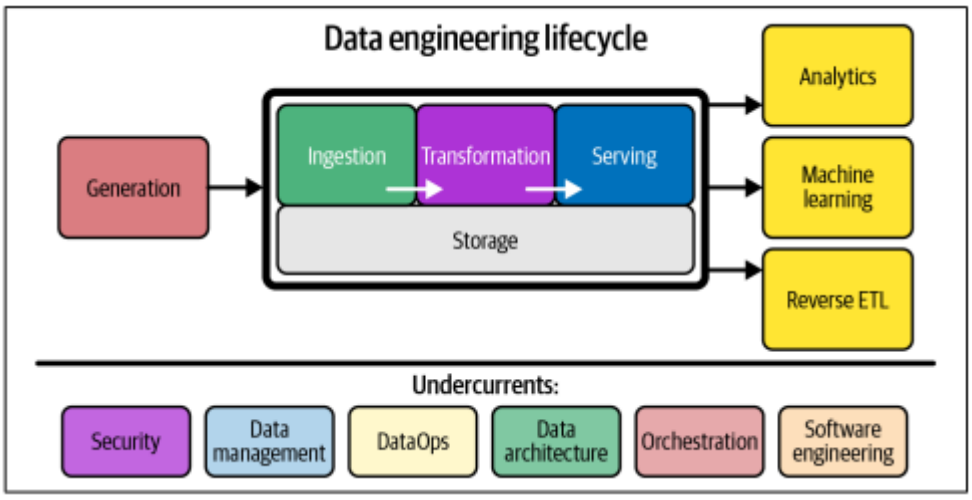



Figura 1 - O ciclo de vida da engenharia de dados.

Fonte: Fundamentals of Data Engineering, julho de 2022, p5.

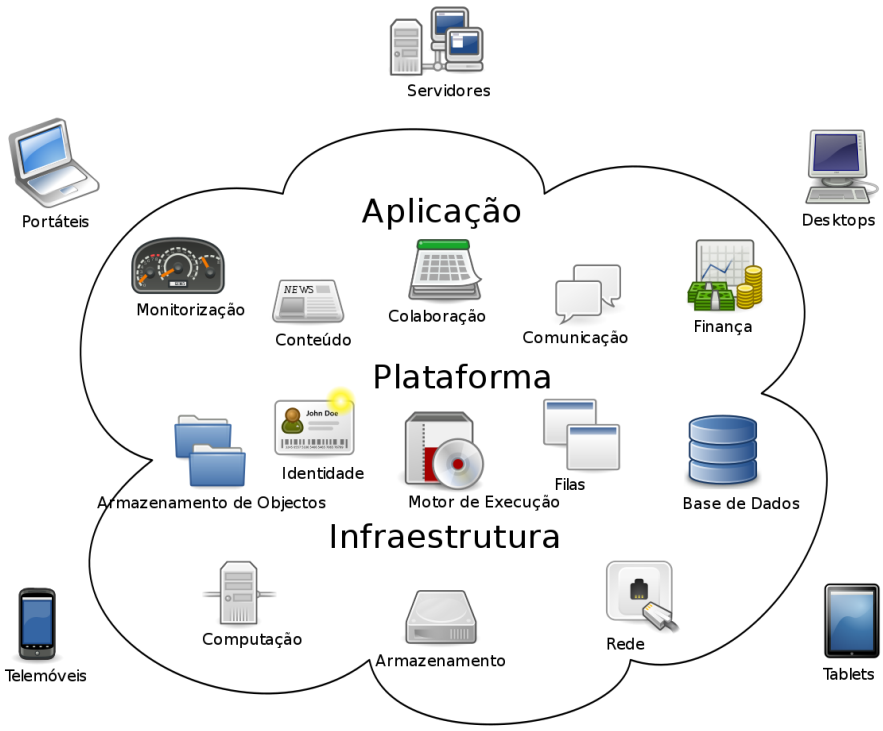
## 2.2. Computação em nuvem

Por longos anos, as pequenas, médias e grandes empresas, gastaram dinheiro em larga escala para adquirir, configurar e manter seus serviços de computação,

contratação de profissionais qualificados, sempre com a preocupação de manter seus serviços disponíveis e com segurança.

Uma das maiores dificuldades de se manter um ambiente *on premise* (ambientes locais em servidores da empresa), é que será necessário em alguns casos, vários servidores para sustentar a infraestrutura e, para redimensionar esses servidores é onde está o maior problema. Quando há a necessidade de aumentar os recursos, o gasto que envolve toda a logística é muito grande. Quando todo esse hardware já não é mais necessário, as empresas não têm muito o que fazer e perdem o investimento.

Com a chegada da computação em nuvem, todas as desvantagens de se manter um ambiente desses foram vencidos. A computação em nuvem é o fornecimento de serviços de computação como, por exemplo: servidores (os antigos *mainframes*), contas de armazenamento de dados, bancos de dados, rede de computadores, ferramentas de processamento, *softwares* como serviço etc. Alguns desses serviços podem ser analisados na Figura 2, que é um diagrama ilustrando alguns desses serviços de computação em nuvem.




Computação em nuvem

Figura 2 – Diagrama ilustrando Computação em Nuvem

Fonte: Sam Johnston, 5 de setembro de 2013.

## 2.3. Big Data

Devido à grande variedade de dados, seu crescente volume e a necessidade de trabalhar com velocidades cada vez mais altas, o conceito de *big data* foi se disseminando dentro das empresas. Segundo a Oracle, 2020a, *Big Data* é um conjunto de dados maior e mais complexo, especialmente de novas fontes de dados. Esses conjuntos de dados são tão volumosos que o software tradicional de processamento de dados simplesmente não consegue gerenciá-los.

Segundo a publicação no Statista, 2022, “A quantidade total de dados criados, capturados, copiados e consumidos globalmente está previsto para aumentar rapidamente, atingindo 64,2 *zettabytes* em 2020. Nos próximos cinco anos até 2025, a criação global de dados deverá crescer para mais de 180 zettabytes.”. Na Figura 3 é possível analisar essa previsão para os próximos cinco anos.

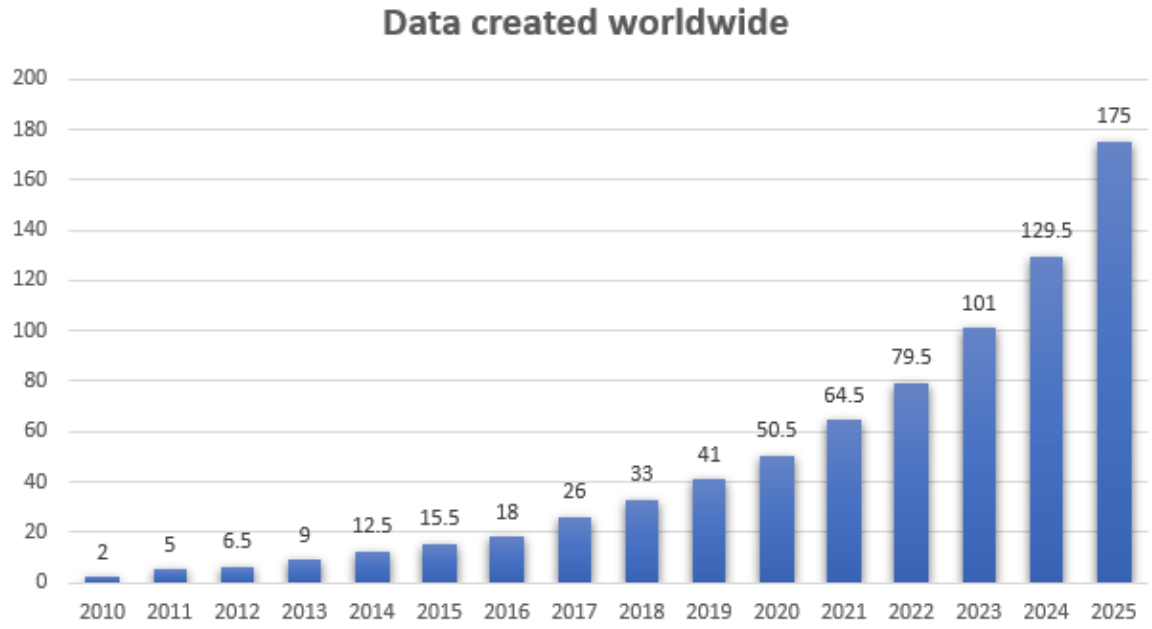



Figura 3 - Volume de dados (em zettabytes) gerados e consumidos no mundo de 2010 a 2020, com previsão de 2021 a 2025

Fonte: Statista Research Department, 08 de setembro de 2022.

Trabalhar com *big data* tem a sua complexidade e seus desafios, pois lidar com um grande volume de dados é algo que os sistemas tradicionais de processamento de dados não conseguem gerenciar e envolvem um alto custo. Mas toda essa volumetria de dados pode resolver grandes problemas de negócios e tem diversos benefícios: com muitos dados, você pode obter respostas de qualidade desde que os dados sejam adequados. Exemplo: considere analisar o perfil de consumo com dados de pessoas de uma determinada classe econômica e querer generalizar para a resposta para a população em geral. A resposta teria um bias, enfim, e não corresponde à realidade geral das pessoas, mesmo que a pesquisa tivesse levado em conta uma grande quantidade de dados.

## 2.4. Data Lake

Quando falamos de *big data,* não podemos deixar de citar uma das ferramentas mais importantes de armazenamento de dados, que gera um ganho no poder de processamento de dados e escalabilidade. Um *data lake,* bem estruturado e governado, facilita a empresa lidar com o conceito de *big data* na prática, devido suas especificidades.

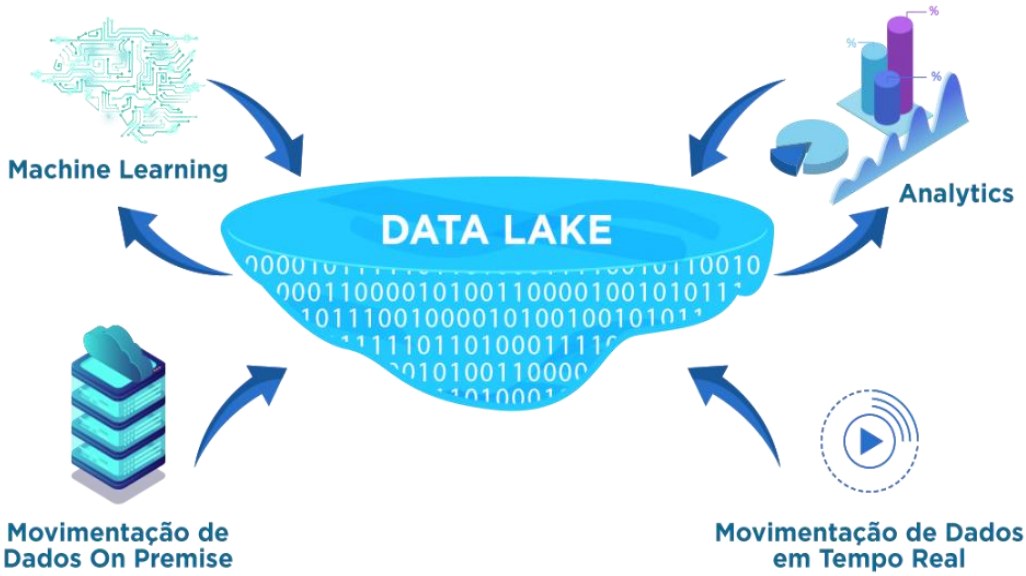



Figura 4 - Operações em um data lake.

Fonte: SOLVIMM, 20 de dezembro de 2021

Analisando a Figura 4 é possível ver diversas estruturadas de dados sendo gerados e consumidos do *data lake,* que podem ser classificadas como:

- Dados estruturados: são os dados criados e elaborados com uma estrutura específica e possuem sua finalidade bem definida, como, por exemplo, uma tabela de um banco de dados que foi pensada para ter dados de vendas por produto e cada coluna foi desenhada para receber determinados tipos de dados específicos (*string*, interger, *float etc.)*, se aceita nulo ou não, seu tamanho, entre outras definições.
- Dados não estruturados: é o oposto dos dados estruturados, não possuem estruturas definidas, não possuem padronização em seus tipos de dados e podem ser compostas por diversos elementos do cotidiano, como por exemplo: vídeos, áudios, fotos, documentos etc.
- Dados semiestruturados: são dados que possuem os dois cenários citados anteriormente, podem conter tanto dados estruturados quanto dados não estruturados em suas características. Possuem alguma estrutura organizacional, mas possuem um pouco mais de flexibilidade para seu armazenamento.

O *data lake* trouxe a possibilidade de armazenar, processar e gerar uma grande quantidade de dados estruturados, semiestruturados e não estruturados. Em um bom projeto de *data lake*, normalmente são criadas zonas para separar esses dados de maneira lógica para manter o ambiente organizado e facilitar a compreensão. Em um bom sistema de trabalho, são utilizadas quatro zonas: *transient, raw, trusted e refined*, como pode ser observada na Figura 5.

A zona *transient* é utilizada para cópias temporárias, *spools* de *streaming* ou dados que não há a necessidade de armazenamento por longos períodos. Nessa camada, os processos de expurgo são definidos de forma a manter o mínimo de tempo possível e não há um controle na estrutura de armazenamento.

Na zona *raw*, os dados são armazenados em sua forma bruta, exatamente da maneira como os dados são em suas fontes. Os dados possuem governança idealizados para possuírem a mesma estrutura e organização da fonte de dados.

Na zona *trusted*, são utilizados como base, os dados oriundos da camada bruta (*raw zone*), onde os dados são transformados em entidades centralizadas. Nessa camada, são criadas rotinas de verificação dos dados, aplicadas rotinas de qualidade

dos dados, verificação de campos, estrutura de campos, tornando o dado confiável e servindo como base para o refinamento do dado para ser disponibilizado na *refined.*

Na zona *refined*, os dados que vieram da *trusted* são enriquecidos com outras bases e, partir dela, são gerados novos indicadores, que geralmente serão utilizados em *dashboards* ou modelo de ciência de dados, facilitando a tomada de decisão e na geração de *insights* para as áreas de negócio corporativas. Além desse tipo de utilização, também temos os sistemas consumidores, catálogo de dados, ferramentas de visualização e conectores externos que fazem suas consultas nas bases disponibilizadas nessa camada.

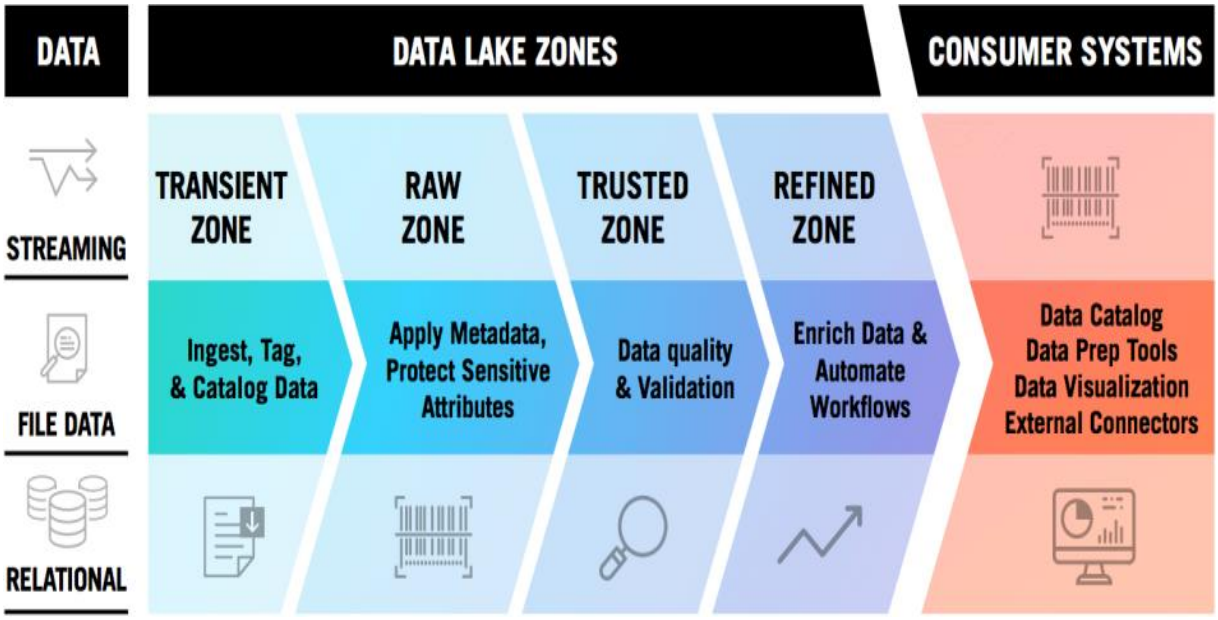



Figura 5 - exemplificação das zonas de um data lake.

Fonte: Luiz Henrique Garetti, 26 de janeiro de 2020.

## 2.5. Ingestão de dados

Com o crescente volume de dados sendo gerados, os cientistas, engenheiros ou qualquer agente do negócio de uma empresa, tem a necessidade de ter acesso a essas informações de maneira prática e organizada. A ingestão de dados é o processo de transferir essas informações, de qualquer lugar que seja, para um local adequado, que normalmente é um *data lake*, porém cada empresa pode definir sua estratégia de acordo com sua necessidade e orçamento disponibilizado.

As principais formas de ingestão de dados são em *batch* ou *streaming*. Os dados gerados em *batch* são os dados agrupados por um intervalo de tempo e sua atualização é feita por intervalos predefinidos como, por exemplo, tabelas que são atualizadas uma vez ao dia, considerando os dados que foram gerados em sistemas legados no dia anterior, e não é possível entregar em fluxos. Essa forma de ingestão também é definida para os casos em que não há a necessidade de utilização do fluxo contínuo, por ser considerada uma forma muito mais econômica dos trabalhos em *big data*. O processo em batch facilita a ingestão de grandes lotes de dados, por conta de sua característica de não ter uma necessidade de um tempo determinado para o dado estar atualizado.

O processamento em *stream* é o trabalho com dados em tempo real, que são dados sendo fornecidos continuamente e disponibilizados para geração de *insights* o tempo todo. Devido sua característica de latência, essa forma de processamento exige pequenos conjuntos de dados por tabela.

Na Figura 6 é possível observar os fluxos para processamento de dados em *Batch* e *Stream*, desde sua fase de captura de dados (ingestão dos dados), até sua fase de consumo pelas ferramentas de visualização.

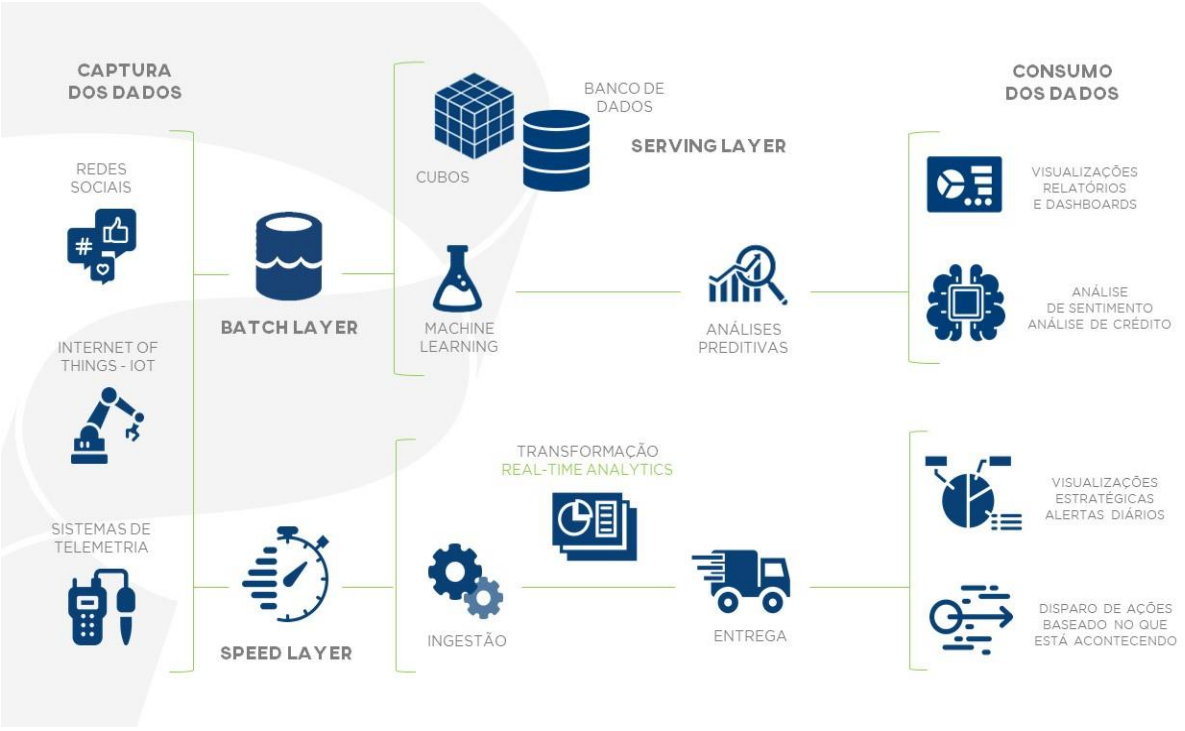



Figura 6 - Exemplo de fluxo de processamento Batch e Stream.

Fonte: Programmers, abril de 2020.

# 3. Ferramentas

A fim de realizar o trabalho do engenheiro de dados citado nos capítulos anteriores, existem tanto ferramentas *open sources* quanto ferramentas pagas e todo esse desenvolvimento pode ser feito local ou em plataformas disponibilizadas nos serviços de computação em nuvem. O que torna o processo escalável e de alta performe é justamente a escolha dessas ferramentas e todo seu processo de automatização e sua escolha sempre depende de qual serviço de nuvem contratado.

Entre os principais provedores de serviço em nuvem, destacam-se os 3 principais e mais utilizados no mercado em 2022 segundo a CRN, 2022, “Amazon, Microsoft e Google combinaram 65% da participação no mercado mundial de serviços em nuvem no segundo trimestre de 2022, acima dos 61% ano a ano”. O serviço de cloud que será abordado nesse tema é o da Microsoft e, entre as principais ferramentas para a engenharia de dados temos a linguagem SQL (*Structured Query Language*), a linguagem de programação Python, que já possui nativamente bibliotecas para manipulação de dados, a técnica de *Web Scraping* e ferramentas de processamento e automatização de processos como Azure Functions, Azure Data Factory e Databricks.

## 3.1. SQL

SQL é a linguagem padrão para manipulação em banco de dados de dados relacionais e é amplamente utilizada por engenheiros de dados para coleta, organização e consolidação de dados. Tendo em vista que as fontes mais comuns de consumo de dados são esses bancos, é muito importante que esse profissional conheça e tenha domínio dessa ferramenta, pois dessas fontes, são extraídas informações gerenciais de sistemas ERP, como financeiro, pedidos, vendas, compras etc. A linguagem SQL tem uma organização que divide seus comandos em cinco subconjuntos que possuem objetivos diferentes, como pode ser observado na Figura 7.

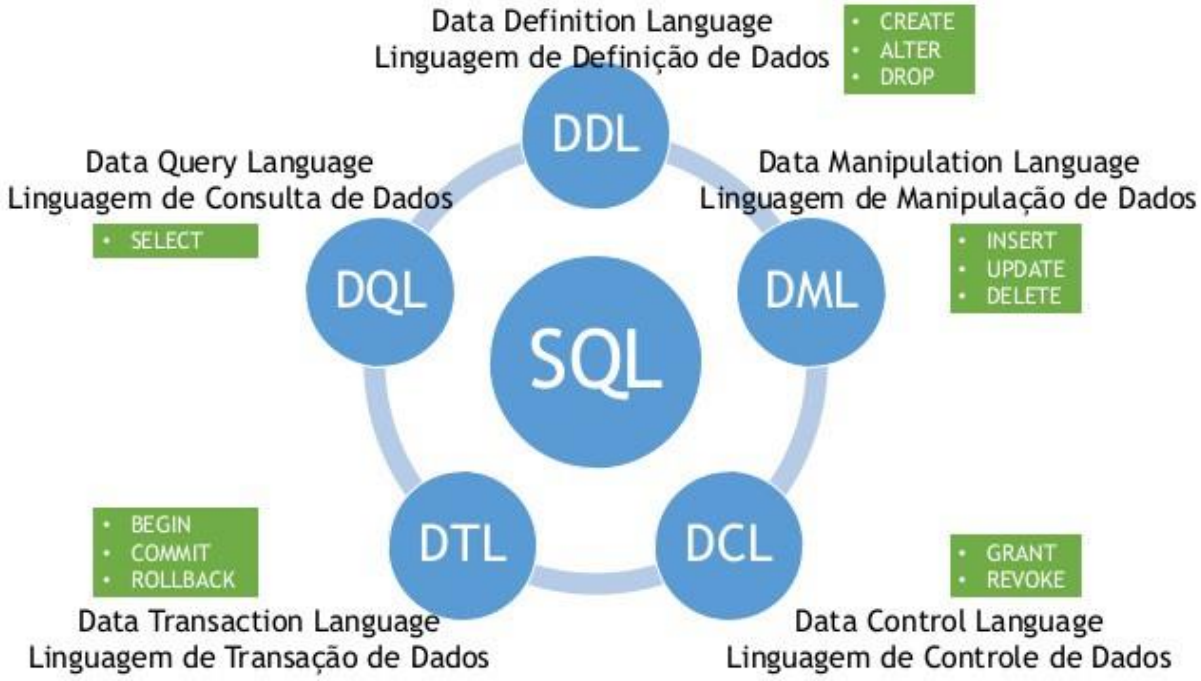



Figura 7 - Os 5 subconjuntos do SQL.

Fonte: Medium, 19 de maio de 2019.

- DML (*Data Manipulation Language*) são os comandos de manipulação de dados armazenados em bancos de dados, *data frames* e até em arquivos que possuem uma estrutura tabular. Os comandos de DML fazem operações de inserção, exclusão e alterações nos dados baseados nos parâmetros da consulta.
- DQL (*Data Query Language*) é o conjunto de SQL que serve para efetuar consultas nos dados armazenados em uma base de dados. É amplamente utilizado pela engenharia de dados, pois através do comando *select*, permite ao engenheiro de dados fazer as ingestões dos dados de uma tabela de um banco e armazenar em qualquer base de dados de sua preferência.
- DDL (*Data Definition Language*) é um subconjunto utilizado para gerenciar estruturas de bancos de dados, por meio de comandos desse subconjunto é possível criar objetos, deletar e alterar.
- DCL (*Data Control Language*) é o subconjunto utilizado para a gestão de acessos à base de dados. É possível conceder ou restringir um conjunto de acessos a objetos do banco.

- DTL (*Data Transaction Language*) é responsável por gerenciar as diferentes transações que são feitas no banco. É permitido que os comandos sejam executados, confirmados e até mesmo desfazer alterações.

## 3.2. Python

A linguagem de programação *python* teve seu desenvolvimento inicial em 1980 por Guido Von Rossum e foi projetado para que tivesse sua sintaxe facilitada em relação a outras linguagens para ler e escrever. É uma linguagem de código aberto e possui uma forte comunidade.

Pode ser utilizada para mineração de dados, ciência de dados, inteligência artificial, aprendizado de máquina, desenvolvimento *web*, estruturas *web*, sistemas integrados, aplicações de design gráfico, desenvolvimento de jogos, automação de testes, automação de *scripts*, e a lista só cresce.

A utilização do *python* por engenheiros de dados é amplamente necessária, pois seu papel abrange trabalhar com vários tipos de formato de dados. A linguagem Python é a mais adequada, pois trabalha facilmente com suas bibliotecas padrão, que suporta o manuseio de csv, excel, parquet, que são os mais utilizados nesse meio.

O engenheiro utiliza python para ingestão de dados de qualquer fonte possível, aquisição de dados de APIs, processamento de dados utilizando bibliotecas nativas e na construção de pipelines de ETL (*Extract, Transform and Load*) e ELT (*Extract, Load and Transform*).

## 3.3. Web Scraping

Com a necessidade crescente de entender seus consumidores, as empresas veem a necessidade de cruzar dados internos com dados externos para geração de *insights*. Pode ser considerado um dado externo todo dado que é gerado a partir de sites governamentais, dados públicos, dados de redes sociais ou qualquer informação disponibilizada em qualquer site, que são de suma importância a disponibilização de dados dentro da estrutura de dados da empresa.

Um desafio notório da engenharia é a criação de processos automatizados para a coleta desses dados. Segundo o Data Science Academy, “web scraping é o ato de

baixar automaticamente os dados de uma página web e extrair informações muito específicas dela. As informações extraídas podem ser armazenadas praticamente em qualquer lugar". (DSA, 2018b). Web Scraping nada mais é que, por meio de scripts em linguagem de programação, o engenheiro "raspar" o site de forma a identificar em que local da página está a informação que ele precisa, extrair essa informação, tratar os dados e disponibilizar em sistemas para utilização do cientista de dados ou para processos de engenharia de dados. Na Figura 8 é possível visualizar algumas das fontes utilizadas para extração com a técnica de *Web Scraping*.

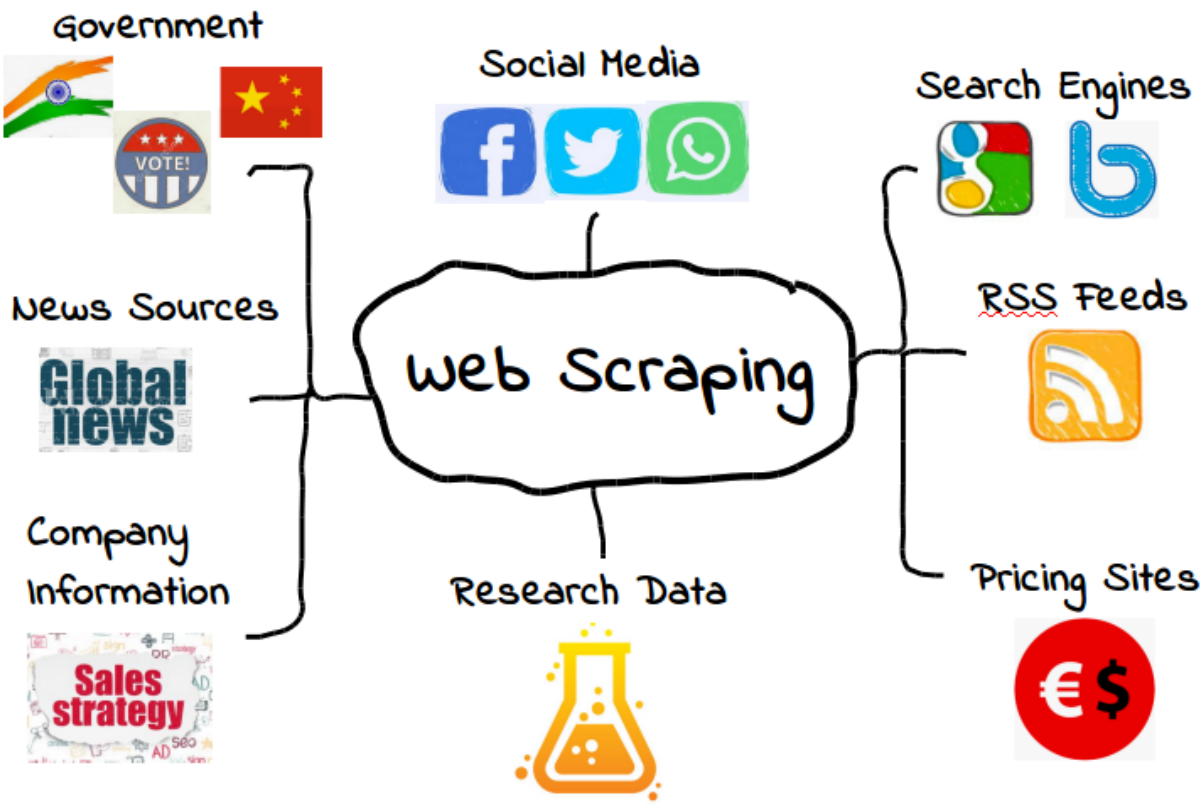



Figura 8 - Fontes de dados para a técnica de Web Scraping.

Fonte: Medium, 14 de fevereiro de 2022.

Uma das bibliotecas utilizadas pelos trabalhos de *Web Scraping* são *Beautiful Soup* e *Selenium*, ambas com o mesmo propósito, porém com utilizações diferentes. *Beautiful soup* é utilizada para extrair o código HTML da página e, por meio de *scripts* identificar onde está a informação necessária que pode ser uma tabela, texto, *links* que direcionam para outros sites, links que direcionam para arquivos, ou seja, qualquer tipo de informação disponibilizado no HTML da página. *Selenium* é uma ferramenta de testes automatizados, que simula o click na tela como se fosse um

usuário real, navegando até chegar na informação, capturando essas informações e carregando nas bases necessárias através linguagem de programação, como python por exemplo.

### 3.4. Azure Functions

Uma vez que finalizamos a criação dos processos de web scraping e tratamento de dados, é necessário a automatização desses processos e a execução deles em ambiente produtivo. O Azure Functions é uma alternativa para execução automatizada desses *scripts*, devido ao seu baixo custo e facilidade de manutenção, tem sido utilizado pelas empresas que optaram por contratar os serviços de nuvem da Microsoft.

Segundo a documentação da Microsoft, *"O Azure Functions é um serviço de nuvem disponível sob demanda que fornece toda a infraestrutura e os recursos continuamente atualizados necessários para executar os aplicativos. Você se concentra nas partes do código mais importantes e o Functions cuida do restante. O Functions fornece computação sem servidor ao Azure. Você pode usar o Functions para criar APIs Web, responder a alterações no banco de dados, processar fluxos de IoT, gerenciar filas de mensagens, entre outras ações."* (Microsoft, 2022a).

Nas Figuras 9 e 10 é exemplificado o portal de serviços das Azure Functions e algumas das funcionalidades, como, por exemplo, a área de codificação e monitoria de processos.

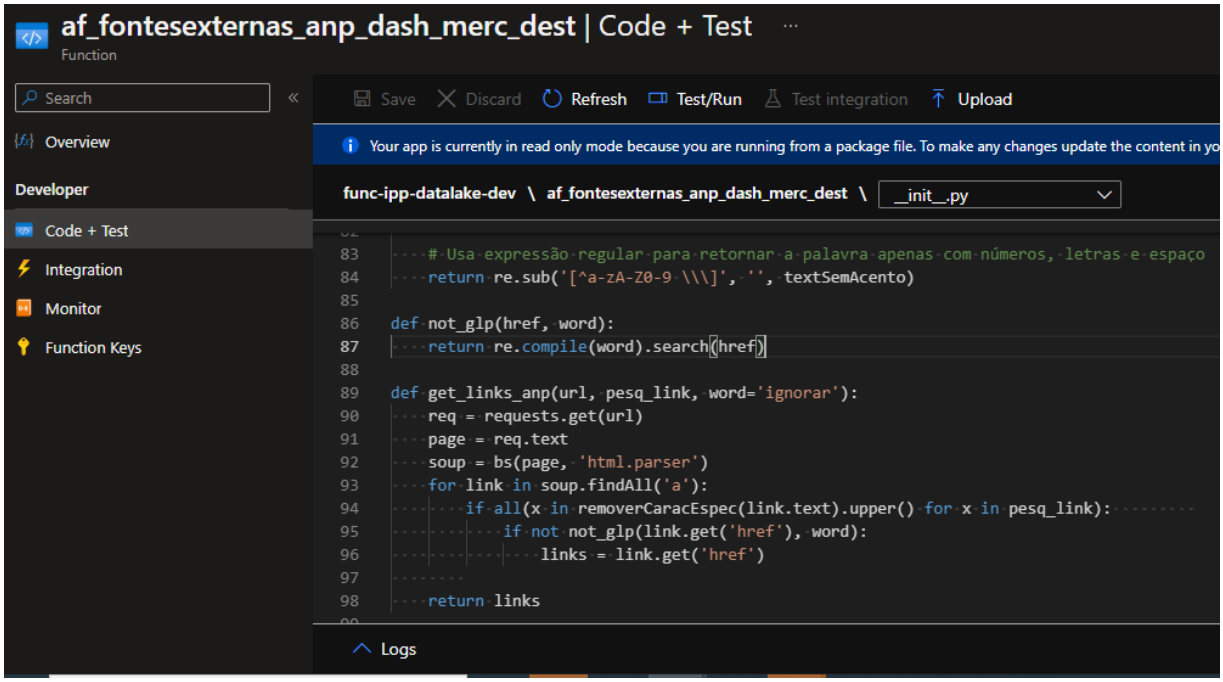



Figura 9 - Portal da Azure Functions, local de codificação e testes.

Fonte: elaborado pelo autor.

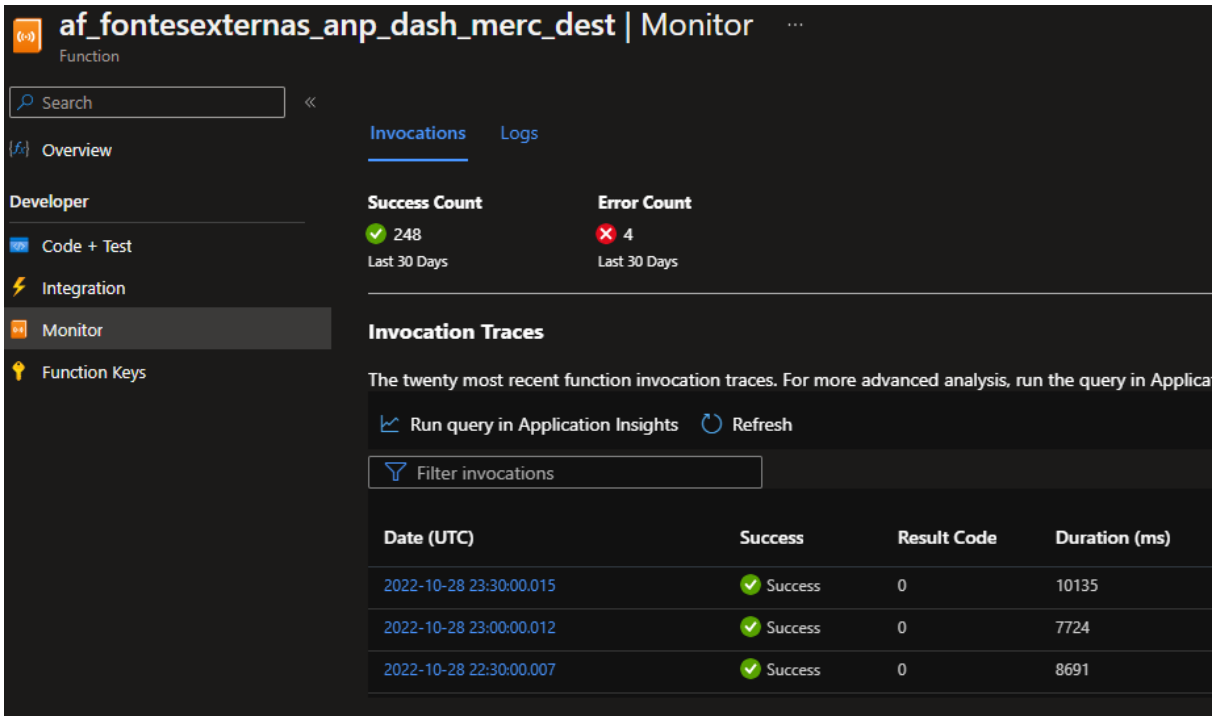



Figura 10 - Portal Azure Functions, local de monitoramento de execuções.

Fonte: Elaborado pelo autor.

A Function permite versionamento de código com as maiores plataformas com essa finalidade como, por exemplo, Devops e Git Hub. Pelos serviços serem do tipo

*serverless* há diversas vantagens, como não ter uma infraestrutura para administrar, ser cobrado somente enquanto executa seus processos, escala automática de recurso e a redução da complexidade de configurações de sistemas operacionais.

### 3.5. Azure Data Factory

Com tanto dado sendo transportado de um sistema para outro, para banco de dados e *data lakes*, em *cloud* ou *on premises*, a Microsoft entregou uma solução de ETL chamada Data Factory que facilita toda essa criação de pipeline de tratamento de dados. Essa facilidade se dá por meio do formato de desenvolvimento da plataforma, que exige pouca codificação e totalmente visual.

Essa ferramenta possui diversos sistemas interconectados que facilitam a orquestração e operacionalização de processos para o engenheiro refinar os dados brutos em *insights* de negócios. O Azure Data Factory (ADF) é uma solução de integração de dados sem necessidade de servidor, com o foco em ingerir, preparar e transformar seus dados. (Microsoft, 2022)

Um dos principais benefícios é pela ferramenta possuir uma rica variedade de conectores com fonte de dados, independentemente de onde estejam e sem custo de licenciamento adicional. É possível conectar nativamente com banco de dados (Relacionais e não relacionais), *storages* em outras nuvens, APIs, *delta lake*, FTP, HDFS, SFTP, GitHub, entre outras. Um outro ponto muito positivo é a variedade de tipos de arquivos que o data factory consegue trabalhar de forma nativa e novamente sem qualquer custo adicional, como por exemplo: parquet, csv, excel, json, avro, txt, arquivo binário, entre outros. Na Figura 11 é possível observar a plataforma do Data Factory, onde foi desenvolvido um pipeline que trabalha com essa variedade de arquivos.

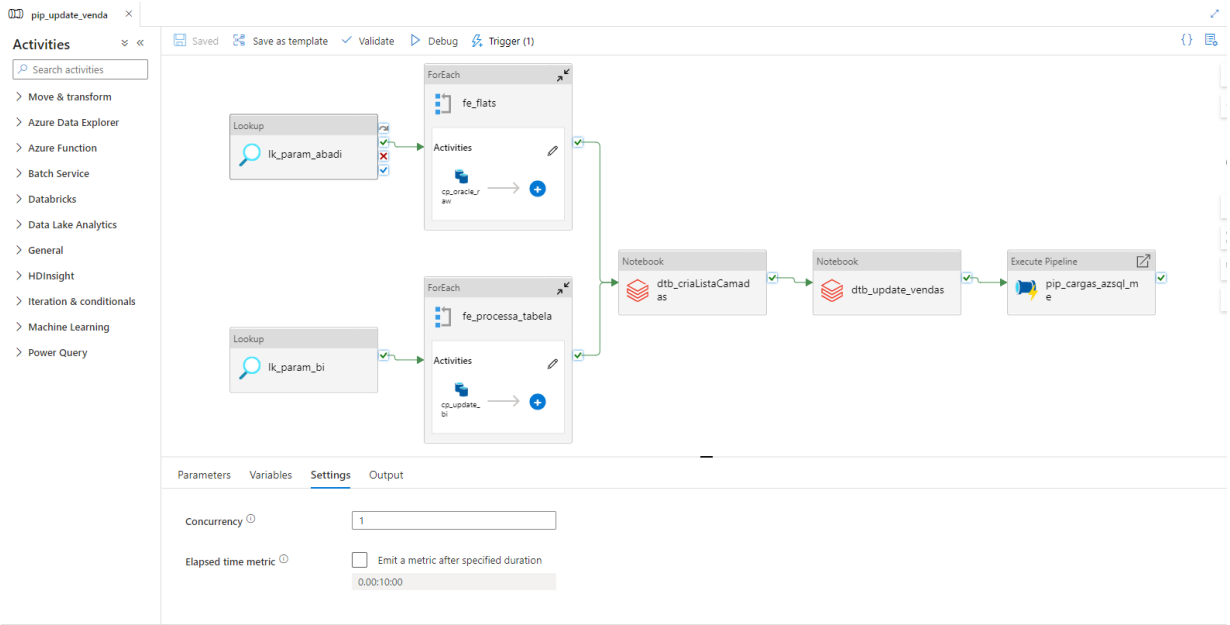


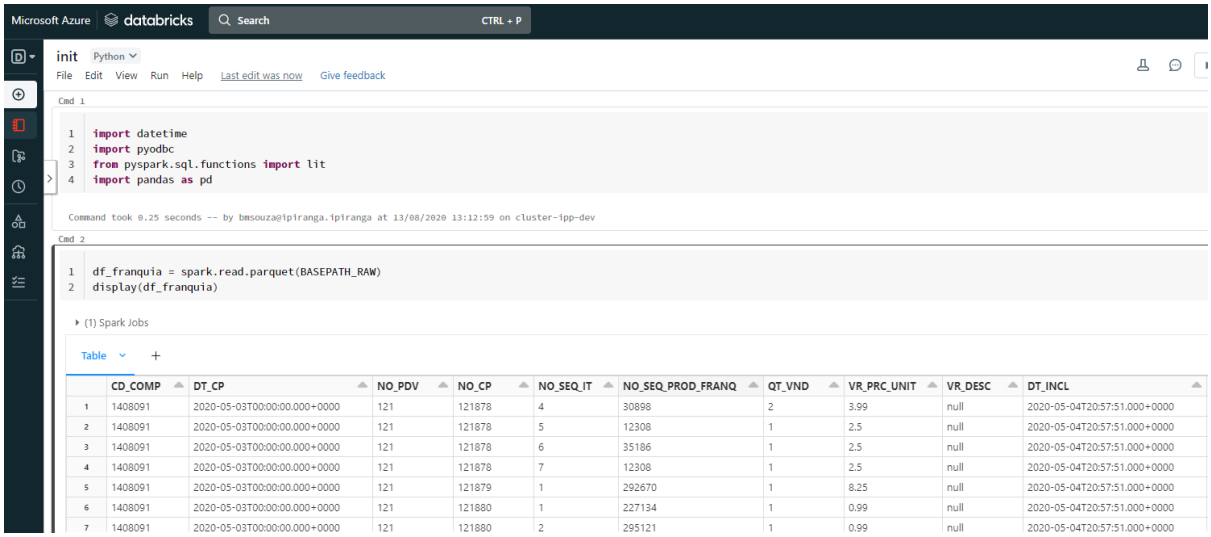

Figura 11 - Exemplo de um pipeline desenvolvido no Azure Data Factory.

Fonte: Elaborado pelo autor.

## 3.6. Databricks

Tomar decisão baseada em dados é a chave para decisão de negócio das empresas. Como foi relatado no tópico sobre *Big Data*, um volume gigantesco de dados flui de diferentes lugares e, para manusear essa grande quantidade de dados, é necessária uma ferramenta escalável, confiável e de fácil utilização. Seguindo a mesma linha de pensamento sobre a necessidade de se usar o data factory, o Databricks vem com o mesmo propósito, processar e transformar grande quantidade de dados.

Databricks é uma das ferramentas de engenharia e ciência de dados baseada em nuvem e é utilizada para explorar e gerar uma grande quantidade de dados. Existe uma grande quantidade de bibliotecas para conexão com diferentes fontes de dados, além de diversos tipos de arquivos também. O ponto principal do databricks, é a possibilidade de utilizar as mais variadas linguagens de programação dentro dele como: Apache Spark, R, Scala, Python ou SQL, como pode ser observado na Figura 12, onde foi desenvolvido um processo utilizando algumas dessas linguagens de programação.




| | CD_COMP | DT_CP | NO_PDV | NO_CP | NO_SEQ_IT | NO_SEQ_PROD_FRANQ | QT_VND | VR_PRC_UNIT | VR_DESC | DT_INCL |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1408091 | 2020-05-03T00:00:00.000+0000 | 121 | 121878 | 4 | 30898 | 2 | 3.99 | null | 2020-05-04T20:57:51.000+0000 |
| 2 | 1408091 | 2020-05-03T00:00:00.000+0000 | 121 | 121878 | 5 | 12308 | 1 | 2.5 | null | 2020-05-04T20:57:51.000+0000 |
| 3 | 1408091 | 2020-05-03T00:00:00.000+0000 | 121 | 121878 | 6 | 35186 | 1 | 2.5 | null | 2020-05-04T20:57:51.000+0000 |
| 4 | 1408091 | 2020-05-03T00:00:00.000+0000 | 121 | 121878 | 7 | 12308 | 1 | 2.5 | null | 2020-05-04T20:57:51.000+0000 |
| 5 | 1408091 | 2020-05-03T00:00:00.000+0000 | 121 | 121879 | 1 | 292670 | 1 | 8.25 | null | 2020-05-04T20:57:51.000+0000 |
| 6 | 1408091 | 2020-05-03T00:00:00.000+0000 | 121 | 121880 | 1 | 227134 | 1 | 0.99 | null | 2020-05-04T20:57:51.000+0000 |
| 7 | 1408091 | 2020-05-03T00:00:00.000+0000 | 121 | 121880 | 2 | 295121 | 1 | 0.99 | null | 2020-05-04T20:57:51.000+0000 |

Figura 12 - Plataforma de desenvolvimento Databricks.

Fonte: Elaborado pelo autor.

A utilização do Databricks SQL Analytics, permite que usuários criem *dashboards*, visualizações e alertas. Essa geração de informação dentro do código facilita as conexões de ferramentas de visualização de dados, como Tableau e PowerBI, para máximo desempenho e colaboração. O Databricks integra uma ampla variedade de ferramentas de desenvolvimento, como IntelliJ, DataGrips, Pycharm, Visual Studio Code e outras. Sua utilização pode ser feita tanto por navegação em nuvem quanto em *on premise*.

## 4. Estudo de Caso

Como foi visto nos capítulos anteriores, o trabalho da Engenharia de Dados é extrair os dados dos mais variados tipos de sistemas e disponibilizá-los no lugar adequado por meio de técnicas que automatizarão futuras ingestões. Nesse estudo de caso, foram escolhidos os dados de índice macroeconômico disponibilizados no site do Banco Central. Todo desenvolvimento, estudo e resultados estão disponibilizados no GitHub <https://github.com/mdsbruno/tcc>.

Os dados foram consumidos da API do Banco Central do Brasil, exemplificado na Figura 13. Após seu consumo, o dado bruto foi armazenado na camada *raw* de um data lake no formato definido pelo próprio engenheiro. Após o dado bruto ser disponibilizado, foi feito um tratamento dos dados, verificando os valores e tipos de cada coluna para disponibilizar o dado confiável para futuros trabalhos de ciência de dados e até mesmo para disponibilização para qualquer área financeira da empresa.


Figura 13 - Site Banco Central do Brasil (BCB).

Fonte: elaborado pelo autor.

O diagrama abaixo representado na Figura 14, apresenta o processo desenvolvido para exemplificar uma ingestão de dados do ponto de vista do engenheiro de dados.

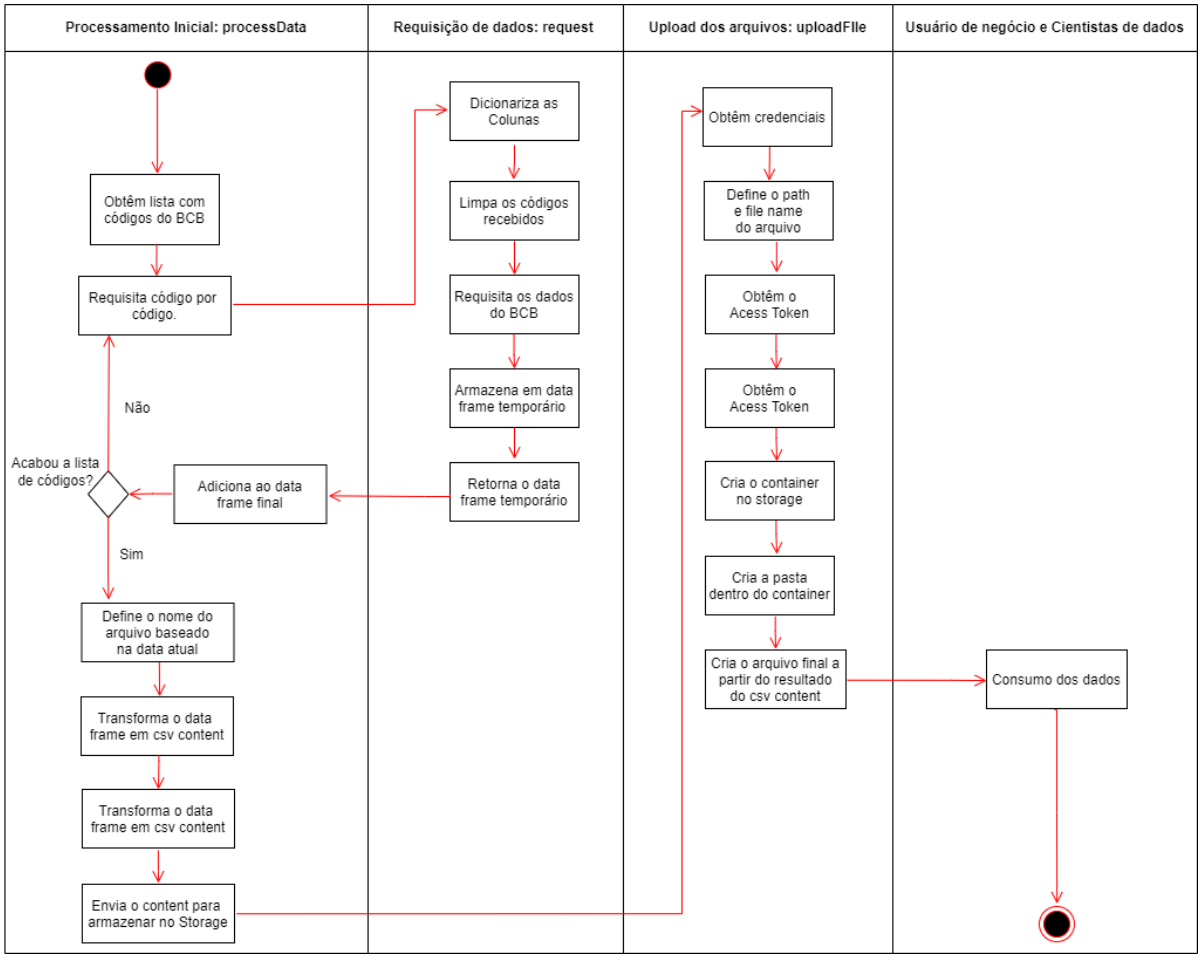



Figura 14 - Diagrama de atividades.

Fonte: elaborado pelo autor.

## 4.1. Caso de Uso

O diagrama de caso de uso da Figura 15 foi desenvolvido com o objetivo de definir a sequência de ações executadas pelo script, desde a concepção dos dados até a disponibilização e consumo em seus sistemas finais.

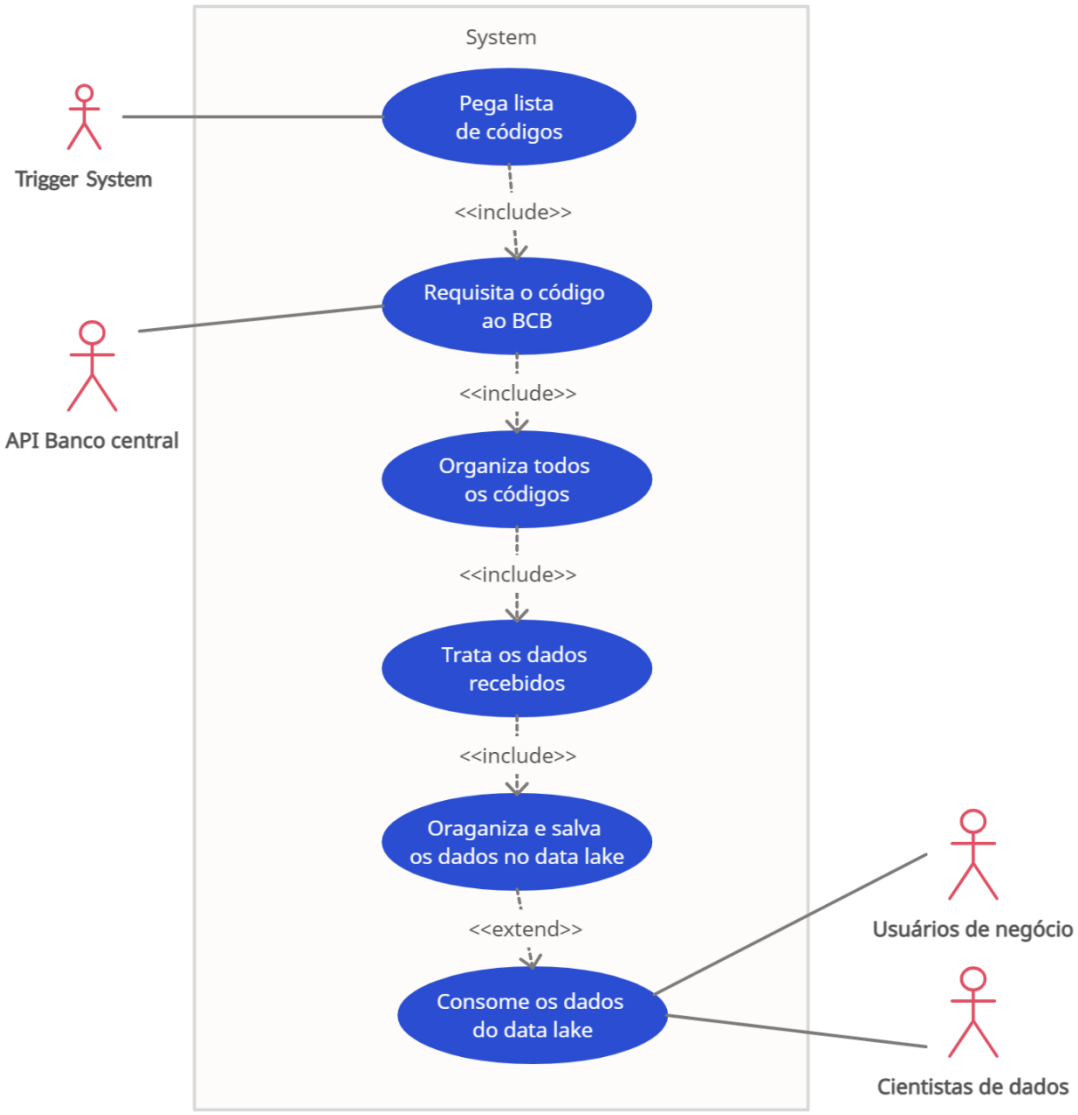



Figura 15 - Caso de uso.

Fonte: elaborado pelo autor.

A Figura 14 detalha o diagrama onde a função *"processData()"* é chamada no início da execução do *script* por meio de uma chamada de HTTP *request* que está sendo executado por um NCRONTAB. Nessa função são obtidos todos os índices macroeconômicos predefinidos pelo solicitador da demanda a partir de uma lista. Essa lista é base de códigos para consulta da API e com ela armazenada em memória, o *script* faz um *looping* em todos os códigos e requisita um por um para a API do Banco Central por meio da função *"request(param)"* e a cada requisição, os dados são armazenados em um *data frame* na memória e tudo será detalhado nos tópicos abaixo.

## 4.2. Extração de dados por API

A função *“request(param)”* ilustrado na Figura 17, recebe por parâmetro o código que será consultado na API e gera uma lista com as colunas que serão montadas a partir dos dados recebidos. O script faz uma remoção de caracteres da lista, transforma em uma variável do tipo *string* e cria outra variável do tipo *list* para efetuar as consultas. Com a lista de códigos armazenados em uma variável, é possível construir a url que será utilizada na requisição. Por meio da biblioteca nativa do python chamada pandas, criamos um *data frame* com a resposta da requisição da API do Banco Central. A Figura 16 é uma lista com os códigos que são usados como base para consulta da API, utilizados pela função *request.*

```
[28562,'Taxa de desocupação - PNADC - Norte','PC']
[28563,'Taxa de desocupação - PNADC - Centro-Oeste','PC']
[28564,'Taxa de desocupação - PNADC - Nordeste','PC']
[28565,'Taxa de desocupação - PNADC - Sudeste','PC']
[28566,'Taxa de desocupação - PNADC - Sul','PC']
[193,'IPCA Estadual - variação mensal - São Paulo','PC']
[13382,'IPCA Estadual - variação mensal - Rio de Janeiro','PC']
[27968,'IPCA Estadual - variação mensal - Rio Branco','PC']
[21900,'IPCA Estadual - variação mensal - Campo Grande','PC']
[13258,'IPCA Estadual - variação mensal - Goiania','PC']
[13250,'IPCA Estadual - variação mensal - Brasilia','PC']
[13066,'IPCA Estadual - variação mensal - Curitiba','PC']
[13077,'IPCA Estadual - variação mensal - Belém','PC']
[12662,'IPCA Estadual - variação mensal - Porto Alegre','PC']
[12853,'IPCA Estadual - variação mensal - Belo Horizonte','PC']
[13922,'Índice Nacional de Preços ao Consumidor Amplo - variação mensal - Nordeste','PC']
[13567,'Índice Nacional de Preços ao Consumidor Amplo - variação mensal - Sul','PC']
[27967,'Índice Nacional de Preços ao Consumidor Amplo - variação mensal - Norte','PC']
[12676,'IPCA Estadual - variação mensal - Recife','PC']
[189,'Índice geral de preços do mercado (IGP-M)','PC']
[190,'Índice geral de preços-disponibilidade interna (IGP-DI)','PC']
[432,'Taxa de juros - Meta Selic definida pelo Copom','PC']
[433,'Índice nacional de preços ao consumidor-amplo (IPCA)','PC']
[1211,'PIB - Deflator implícito ','PC'],[1373,'Produção total de autoveículos','QT']
```

Figura 16 - Lista com algumas series utilizadas para consulta na API.

Fonte: elaborado pelo autor.

```python
def request(param):
    columns_ndf = ['DATA', 'VALOR', 'DS_INDIC_MACRECON','CD_TIPO_DADO_INDIC_MACRECON',]
    param = str(param).replace('[','').replace(']','').replace("'","").split(',')
    url = "https://api.bcb.gov.br/dados/serie/bcdata.sgs.%s/dados/ultimos/20000?formato=json" %(str(param[0]))
    try:
        df = pd.read_json(url)
    except:
        df= pd.DataFrame()
    for i in range(5):
        if len(df.columns) == 0:
            try:
                df = pd.read_json(url)
            except:
                df= pd.DataFrame()
    if len(df.columns) == 0:
        raise ValueError('Erro ao extrair indicador ' + url)
    ndf_matrix = []
    for index, row in df.iterrows():
        for x in df.columns:
            ndf_matrix.append([row['data'], row['valor'],param[1], param[2]])
    ndf = pd.DataFrame(ndf_matrix, columns=columns_ndf)

    return ndf
```

Figura 17 - função request.

Fonte: elaborado pelo autor.

### 4.3. Tratamento de dados

A lista de colunas desse *data frame* que foi predefinida na função anterior, é utilizada para renomear as colunas vindas da API e padronizar de acordo com as regras de governança de dados de cada empresa. Após a montagem do *data frame,* o script faz iterações nos índices e nas linhas verificando os tipos de dados e por meio da função *"append()"* do pandas, cria o *data frame* temporário que será retornado para a função principal (Figura 18). Na função principal (Figura 19), o código que estava em looping nos códigos da lista, armazena cada resultado das operações retornadas pela função *"request(param)"* no *data frame* final. Os dados em sua estrutura final podem ser observados na Figura 20.

```python
for index, row in df.iterrows():
    for x in df.columns:
        ndf_matrix.append([row['data'], row['valor'],param[1], param[2]])
ndf = pd.DataFrame(ndf_matrix, columns=columns_ndf)
```

Figura 18 - Iteração nos dados retornados da API.

Fonte: elaborado pelo autor.

```
for line in lines:
    df_data = request(line)
    df_master = df_master.append(df_data)
```

Figura 19 - Armazenamento no data frame final.

Fonte: elaborado pelo autor.

Cmd 8

```
1  display(df_master)
```

(1) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [DATA: string, VALOR: string ... 2 more fields]

Table +

| | DATA | VALOR | DESCRICAO | CODIGO_TIPO_INDICADOR |
|---|---|---|---|---|
| 1 | 31/12/2021 | 9,25 | Taxa de juros - Meta Selic definida pelo Copom\t | PC |
| 2 | 31/12/2021 | 9,25 | Taxa de juros - Meta Selic definida pelo Copom\t | PC |
| 3 | 31/12/2020 | 2,0 | Taxa de juros - Meta Selic definida pelo Copom\t | PC |
| 4 | 31/12/2020 | 2,0 | Taxa de juros - Meta Selic definida pelo Copom\t | PC |
| 5 | 31/12/2019 | 4,5 | Taxa de juros - Meta Selic definida pelo Copom\t | PC |
| 6 | 31/12/2019 | 4,5 | Taxa de juros - Meta Selic definida pelo Copom\t | PC |
| 7 | 31/12/2018 | 6,5 | Taxa de juros - Meta Selic definida pelo Copom\t | PC |

Truncated results, showing first 1,000 rows. | 0.20 seconds runtime

Figura 20 – Data frame final com o resultado.

Fonte: elaborado pelo autor.

### 4.4. Estratégia de armazenamento de dados

Uma boa estratégia no armazenamento dos dados é preciso para que os dados sejam disponibilizados de maneira simples para consulta por outras áreas. Por esse motivo, foi criada a classe genérica *upload_azure_gen2*, exemplificada na Figura 21, para receber através de parâmetros, as informações necessárias para salvar os dados em formato de arquivo.

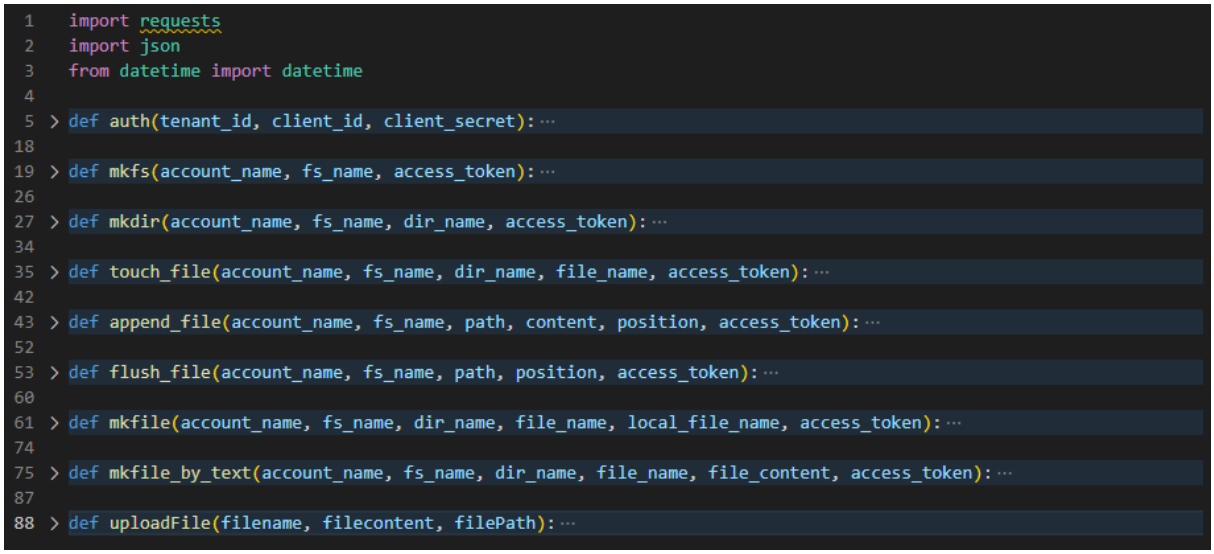



Figura 21 – Classe para interações com o data lake.

Fonte: elaborado pelo autor.

A classe *upload_azure_gen2* foi construída focada em armazenar os dados nos *Storages Accounts (data lakes)* da Microsoft e, para tal, necessita das credenciais desse *storage*, como por exemplo *tenant id*, *client id*, *client secret*, *account name*, *container name*, que são armazenados em variáveis a serem utilizados durante a execução do código. Essa classe recebe por parâmetro o nome do arquivo, o conteúdo do arquivo (dados) e caminho no data lake onde irá salvar. Possui funções para construir o link do *storage*, montar os tokens de autenticação do processo através de *service principal*, criar as pastas dentro do *container* e criar o arquivo no *storage* a partir do arquivo local. A Figura 22 demonstra como foi criada a função principal dessa classe e sua sequência de atividades para salvar os dados no *data lake*.

```python
def uploadFile(filename, filecontent, filePath):
    tenant_id = 'codigo do tenant id'
    client_id = 'codigo do client id'
    client_secret = 'codigo de client secret'
    account_name = 'nome da account name'
    fs_name = 'data'
    now = datetime.now()
    dir_name = filePath+'/'+now.strftime("%Y")+'/'+now.strftime("%m")+'/'+now.strftime("%d")
    file_name = filename
    file_content = filecontent

    #Gera o token para criar link de acesso.
    auth_status_code, auth_result = auth(tenant_id, client_id, client_secret)
    access_token = auth_status_code == 200 and auth_result['access_token'] or ''
    print(access_token)

    #Cria um container caso não exista.
    mkfs_status_code, mkfs_result = mkfs(account_name, fs_name, access_token)
    print(mkfs_status_code, mkfs_result)

    #Cria o diretório com as datas atuais.
    mkdir_status_code, mkdir_result = mkdir(account_name, fs_name, dir_name, access_token)
    print(mkdir_status_code, mkdir_result)

    #Cria o arquivo no data lake.
    mkfile_by_text(account_name, fs_name, dir_name, file_name, file_content, access_token)
```

Figura 22 - Função para salvar os dados no data lake.

Fonte: elaborado pelo autor.

A estratégia utilizada nesse projeto para armazenamento dos dados no *storage* é a criação de um arquivo com os dados históricos até o momento atual em todos as execuções. A cada execução, o processo cria uma pasta dividida por “dia/mês/ano” dentro do storage onde os dados são salvos, considerando sempre a data da execução, como por exemplo: a execução do processo foi requisitada no dia 01/10/2022, os dados serão salvos nas partições do dia 01/10/2022/nome_do_arquivo.csv. Na Figura 23 é possível observar a ilustração da estratégia de armazenamento.

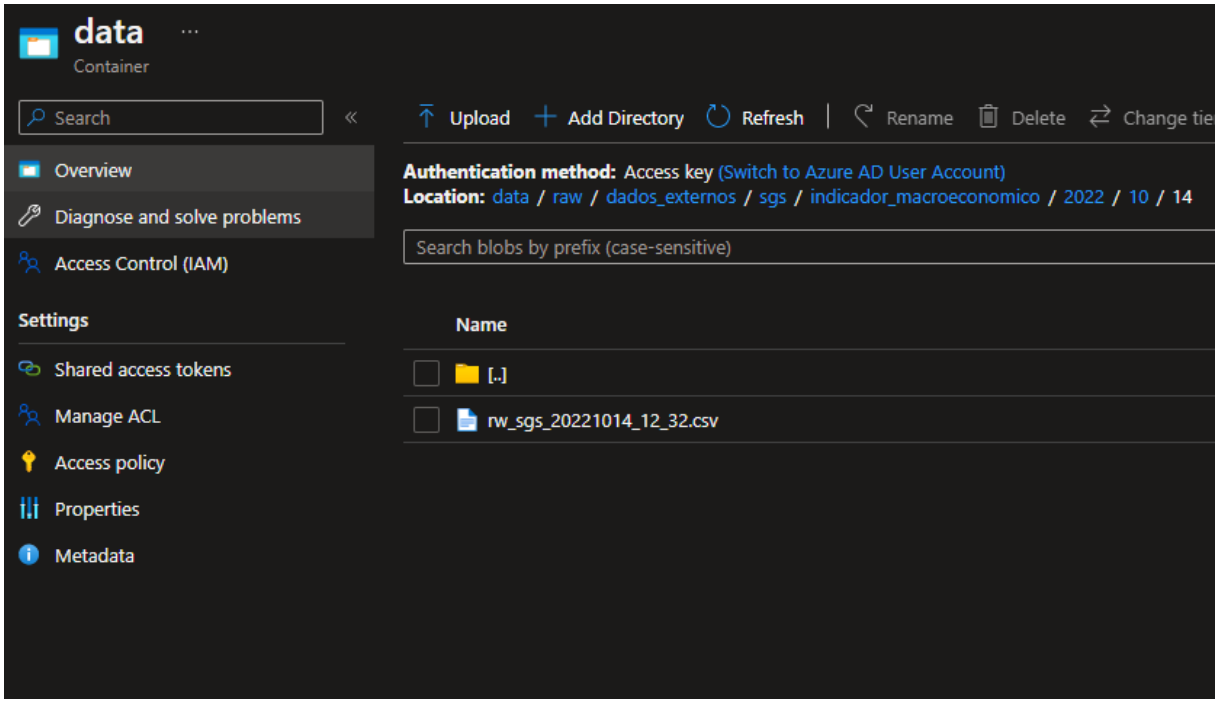



Figura 23 - arquivo salvo no data lake.

Fonte: elaborado pelo autor.

## 5. Conclusão

A profissão Engenheiro de Dados é muito recente e pouco falada nas universidades, exige um misto de conhecimentos por parte do profissional que escolhe entrar nessa área e seu estudo sobre novas ferramentas é muito constante e complexo. Muito se é falado sobre o trabalho dos cientistas de dados e sua importância no mundo corporativo, porém, muitas vezes, é esquecido o trabalho complementar e extremamente necessário que é exercido pelo engenheiro.

A fonte escolhida para implementação desse trabalho foi baseada na necessidade que as empresas possuem para ter em sua base os índices mais importantes de crescimento do país, como por exemplo: produto interno bruto, inflação, taxa de desemprego, índice de volume de vendas no varejo, taxa de juros – Selic, produção de veículos, entre outros. Esses dados ajudam o investidor e as áreas das empresas tomarem as devidas decisões sobre o que investir e qual caminho seguir em um determinado setor de uma empresa, além de ser utilizado por modelos de ciência de dados para prever ações de mercado e ajudar cada vez mais na tomada de decisão. Porém, os índices citados não são objeto de pesquisa desse trabalho, essa base foi escolhida para exemplificar o fluxo de ingestão de dados feita por um engenheiro de dados.

O objetivo geral desse trabalho foi falar um pouco sobre essa profissão, suas necessidades e desafios, apresentar alguns dos principais softwares utilizados nas grandes empresas e mostrar através de implementação todas as etapas de um desenvolvimento de engenharia de dados. Foi implementada uma extração dos principais dados de índice macroeconômicos disponibilizados no site do Banco Central do Brasil por meio de API através da linguagem de programação python. Os dados obtidos foram processados, tratados e disponibilizados em um data lake para consumo por qualquer agente da empresa, seja por software através de cientistas de dados ou usuários de negócio que trabalham com planilhas.

## 6. Trabalhos futuros

Existem diversas ferramentas de engenharia de dados além das que foram abordadas nesse trabalho, como por exemplo o Apache Spark que é também é uma linguagem de programação que atualmente é muito utilizada na parte de processamento de dados dentro da ferramenta databricks.

Atualmente o Databricks é uma ferramenta que está em diversos serviços de cloud e sua utilização vai muito além de processar os dados, ela atualmente possui acoplada em sua arquitetura, disponibilização de dados em formato de um banco sql e é possível conexão direta das plataformas de visualização de dados para a geração de dashboards.

Apesar de não estar no escopo do engenheiro de dados a criação de dashboards, o trabalho da engenharia de dados impacta diretamente e é a ponte para as entregas de visualização de dados, por isso, a apresentação de uma ferramenta de visualização, seus conceitos e implementação, podem ser abordados.

Devido a dificuldade de encontrar autores que falem sobre a profissão, não foi possível explorar um pouco os dados gerados no data lake e nem falar um pouco sobre como processar esses dados e suas utilizações pelas companhias.

A Microsoft recentemente criou uma ferramenta chamada Azure Synapse, com o objetivo de ter todas as ferramentas de dados dentro dela, como por exemplo: execução de scripts em python, processamento de dados, banco de dados dedicados e ser uma interface de link com outros recursos de outras nuvens.

Para fechamento de escopo do trabalho, foi escolhida a cloud da Microsoft para detalhar as ferramentas e conhecimentos necessários para um engenheiro de dados, é interessante também citar dados e exemplificar outras ferramentas de outras clouds, tendo em vista que são amplamente utilizadas por outras empresas também.

Trabalhar com Big Data exige que as ferramentas consigam responder e executar o processamento de dados em alta velocidade e para isso foram criadas as Delta Tables, que são tabelas físicas em sistemas de arquivos e para algumas volumetrias, é necessário trabalhar com esse conceito.

## 7. Anexos

Anexo 1 – Repositório do projeto no GitHub.
Disponível em: <https://github.com/mdsbruno/tcc/>.

## 8. Referências bibliográficas


REIS, JOE; Fundamentals of Data Engineering - Plan and Build Robusts Data Systems. 1ª ed. Estados Unidos: O'Reilly Media, 2022.

G1. Demanda por profissionais da área de dados cresce quase 500%; salários chegam a R$ 22 mil. Globo, 05 jul. 2021. Disponível em: <https://g1.globo.com/economia/concursos-e-emprego/noticia/2021/07/05/demanda-por-profissionais-da-area-de-dados-cresce-quase-500percent-salarios-chegam-a-r-22-mil.ghtml>. Acesso em: 11 de nov. 2022.

COURSERA. What Is a Data Engineer?: A Guide to This In-Demand Career. Coursera, 27 out. 2022. Disponível em: https://www.coursera.org/articles/what-does-a-data-engineer-do-and-how-do-i-become-one Acesso em: 03 de nov. 2022.

CETAX. Data Engineer ou Engenheiro de Dados – Conheça mais sobre. Cetax, 19 jan. 2022. Disponível em: <https://www.cetax.com.br/blog/data-engineer-ou-engenheiro-de-dados/>. Acesso em: 10 ago. 2022.

DIGITAL HOUSE. Data engineer: um guia completo com tudo o que você precisa saber sobre a carreira, 30 set. 2021. Disponível em: <https://www.digitalhouse.com/br/blog/data-engineer/>. Acesso em: 10 ago. 2022.

GOBLE, NICK. What is Data Engineering? Everything You Need to Know in 2022, 4 jan. 2022. Disponível em: <https://www.phdata.io/blog/what-is-data-engineering/>. Acesso em: 10 ago. 2022.

STRATIS, KYLE. What Is Data Engineering and Is It Right for You?. Disponível em: <https://realpython.com/python-data-engineer/>. Acesso em: 10 ago. 2022.

ORACLE. O que é Big Data?. Oracle, 2020. Disponível em: <https://www.oracle.com/br/big-data/what-is-big-data.html>. Acesso em: 11 ago. 2022.

CETAX. Big Data: O que é, conceito e definição, 26 jan. 2022. Disponível em: <https://www.cetax.com.br/blog/big-data/>. Acesso em: 11 ago. 2022.

BIGDATACORP. A importância dos dados para a evolução da Inteligência Artificial, 15 nov. 2021. Disponível em: https://bigdatacorp.com.br/a-evolucao-da-inteligencia-artificial-a-importancia-dos-dados/#:~:text=A%20produ%C3%A7%C3%A3o%20de%20dados%20dobra,a%2035%20trilh%C3%B5es%20de%20gigabytes. Acesso em: 12 ago. 2022.

GOOGLE CLOUD. O que é data lake?. Disponível em: https://cloud.google.com/learn/what-is-a-data-lake?hl=pt-br. Acesso em: 12 ago. 2022.

LEITE, MARCELO. 3 camadas para sucesso do meu Data Lake, Linkedin, 10 out. 2019. Disponível em: https://www.linkedin.com/pulse/3-camadas-para-sucesso-do-meu-data-lake-marcelo-leite-/?originalSubdomain=pt. Acesso em: 12 ago. 2022.

LOCK, MICHAEL. ANGLING FOR INSIGHT IN TODAY'S DATA LAKE, Aberdeen, out. 2017. Disponível em: https://s3-ap-southeast-1.amazonaws.com/mktg-apac/Big+Data+Refresh+Q4+Campaign/Aberdeen+Research+-+Angling+for+Insights+in+Today%27s+Data+Lake.pdf. Acesso em: 12 ago. 2022.

NAEEM, TEHREEM. Ingestão de dados - definição, desafios e práticas recomendadas, Astera, 16 mar. 2020. Disponível em: https://www.astera.com/pt/type/blog/data-ingestion/#:~:text=O%20que%20%C3%A9%20ingest%C3%A3o%20de,documentos%2C%20data%20mart%2C%20etc.. Acesso em: 13 ago. 2022.

LCDS. Ingestão de dados, Medium, 31 mai. 2020. Disponível em: https://medium.com/@lcorreasantos/ingest%C3%A3o-de-dados-c249366b12b2. Acesso em: 13 ago. 2022.

SANTO DIGITAL. Processamento de dados: o que é batch e stream?. Disponível em: https://santodigital.com.br/processamento-de-dados-o-que-e-batch-e-stream/. Acesso em: 13 ago. 2022.

EQUIPE INSIGHT. Entenda como funciona streaming de dados em tempo real, 31 jan. 2020. Disponível em: https://insightlab.ufc.br/entenda-como-funciona-streaming-de-dados-em-tempo-real-2/. Acesso em: 13 ago. 2022.

AWS. O que são dados em streaming?. Disponível em: https://aws.amazon.com/pt/streaming-data/. Acesso em: 13 ago. 2022.

ARAUJO, JUAREZ. Ingestão de dados: descubra as principais práticas e desafios, DBACORP BLOG, 4 ago. 2022. Disponível em: https://blog.dbacorp.com.br/2022/08/04/ingestao-de-dados/. Acesso em: 18 out. 2022.

HARANAS, MARK. Top Cloud Market Share Leaders: AWS, Microsoft, Google Lead Q2 2022, CRN, 17 ago. 2022. Disponível em: https://www.crn.com/news/cloud/top-cloud-market-share-leaders-aws-microsoft-google-lead-q2-2022?itc=refresh. Acesso em: 18 out. 2022.

BETRYBE. SQL: O que é e como usar os principais comandos básicos SQL, 07 jul. 2022. Disponível em: https://blog.betrybe.com/sql/. Acesso em: 11 nov. 2022.

QEXPERT. Indicamos 9 ferramentas para os Engenheiros de Dados construírem uma Arquitetura de Dados organizada, escalável e segura!, 19 mai. 2022. Disponível em:

https://qexpert.com.br/news/9-ferramentas-para-engenheiros-de-dados/. Acesso em: 14 ago. 2022

BETRYBE. Python: o que é, como usar, guia pra aprender a linguagem, 18 ago. 2022. Disponível em: https://blog.betrybe.com/python/. Acesso em: 11 nov. 2022.

KRIGER, DANIEL. O QUE É PYTHON, PARA QUE SERVE E POR QUE APRENDER?, Kenzie, 08 jun. 2022. Disponível em: https://kenzie.com.br/blog/o-que-e-python/. Acesso em: 2 set. 2022.

EQUIPE DEVMEDIA. Guia Completo de Python. Disponível em: https://www.devmedia.com.br/guia/python/37024. Acesso em: 2 set. 2022.

PROJECT PRO. How to learn Python for Data Engineering?, 04 out. 2022. Disponível em: https://www.projectpro.io/article/python-for-data-engineering/592. Acesso em: 5 nov. 2022.

DEMOND, KIM, What is Python used for? Top 5 Python uses, Codingnomads. Disponível em: https://codingnomads.co/blog/python/what-is-python-used-for-python-uses. Acesso em: 5 nov. 2022.

EQUIPE CODINGNOMADS. How to Learn Python: The Beginners Guide. Disponível em: https://codingnomads.co/blog/how-to-learn-python-the-beginners-guide/. Acesso em: 5 nov. 2022.

MOHANTY, SAPAN. Web Scraping & Techniques, Medium, 14 fev. 2022. Disponível em: https://thestartupcto.org/web-scraping-techniques-5030fbf1fba. Acesso em: 10 nov. 2022.

CRUMMY. Beautiful Soup Documentation. Disponível em: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Acesso em: 2 set. 2022.

BEAUTIFUL SOUP. Beautiful Soup Documentation. Disponível em: https://beautiful-soup-4.readthedocs.io/en/latest/. Acesso em: 2 set. 2022.

CERQUEIRA, NAIARA. Beautiful Soup: parseamento de htmL, Medium, 6 out. 2020. Disponível em: https://medium.com/pyladiesbh/beautiful-soup-parseamento-de-html-337197a7d4b9. Acesso em: 15 ago. 2022.

SELENIUM. The Selenium Browser Automation Project. Disponível em: https://www.selenium.dev/documentation/. Acesso em: 12 ago. 2022.

SELENIUM. Selenium With Python. Disponível em: https://selenium-python.readthedocs.io/. Acesso em: 12 ago.2022.

MICROSOFT. Azure Functions documentation. Disponível em: https://docs.microsoft.com/en-us/azure/azure-functions/. Acesso em: 01 ago. 2022.

BELORIO, MARCOS, Conhecendo o Azure Functions. DEV, 1 abr. 2021. Disponível em: https://dev.to/marcosbelorio/conhecendo-o-azure-functions-2756#:~:text=Principais%20vantagens%20ao%20utilizar%20Azure%20Functions%3A&text=Escal%C3%A1vel%20automaticamente%20%2D%20o%20servi%C3%A7o%20auto,tudo%20isso%20invis%C3%ADvel%20para%20n%C3%B3s.. Acesso em: 01 ago. 2022.

MICROSOFT. Azure Data Factory Documentation. Disponível em: https://docs.microsoft.com/en-us/azure/data-factory/. Acesso em: 01 ago. 2022.

LEARN MICROSOFT. Visão geral do conector do Azure Data Factory e do Azure Synapse Analytics, 10 nov. 2022. Disponível em: https://learn.microsoft.com/pt-br/azure/data-factory/connector-overview/. Acesso em: 12 nov. 2022.

LEARN MICROSOFT. Formatos de arquivo e codecs de compactação compatíveis no Azure Data Factory e no Synapse Analytics (herdado), 26 set. 2022. Disponível em: https://learn.microsoft.com/pt-br/azure/data-factory/supported-file-formats-and-compression-codecs-legacy. Acesso em: 12 nov. 2022.

DATABRICKS. Databricks Documentation, 03 nov. 2022. Disponível em: https://docs.databricks.com/. Acesso em: 12 nov. 2022.

EQUIPE DSA. Web Scraping e Web Crawling são Legais ou Ilegais?. Data Science Academy, 1 jul. 2018. Disponível em: https://blog.dsacademy.com.br/web-scraping-e-web-crawling-sao-legais-ou-ilegais/. Acesso em: 15 out. 2022.