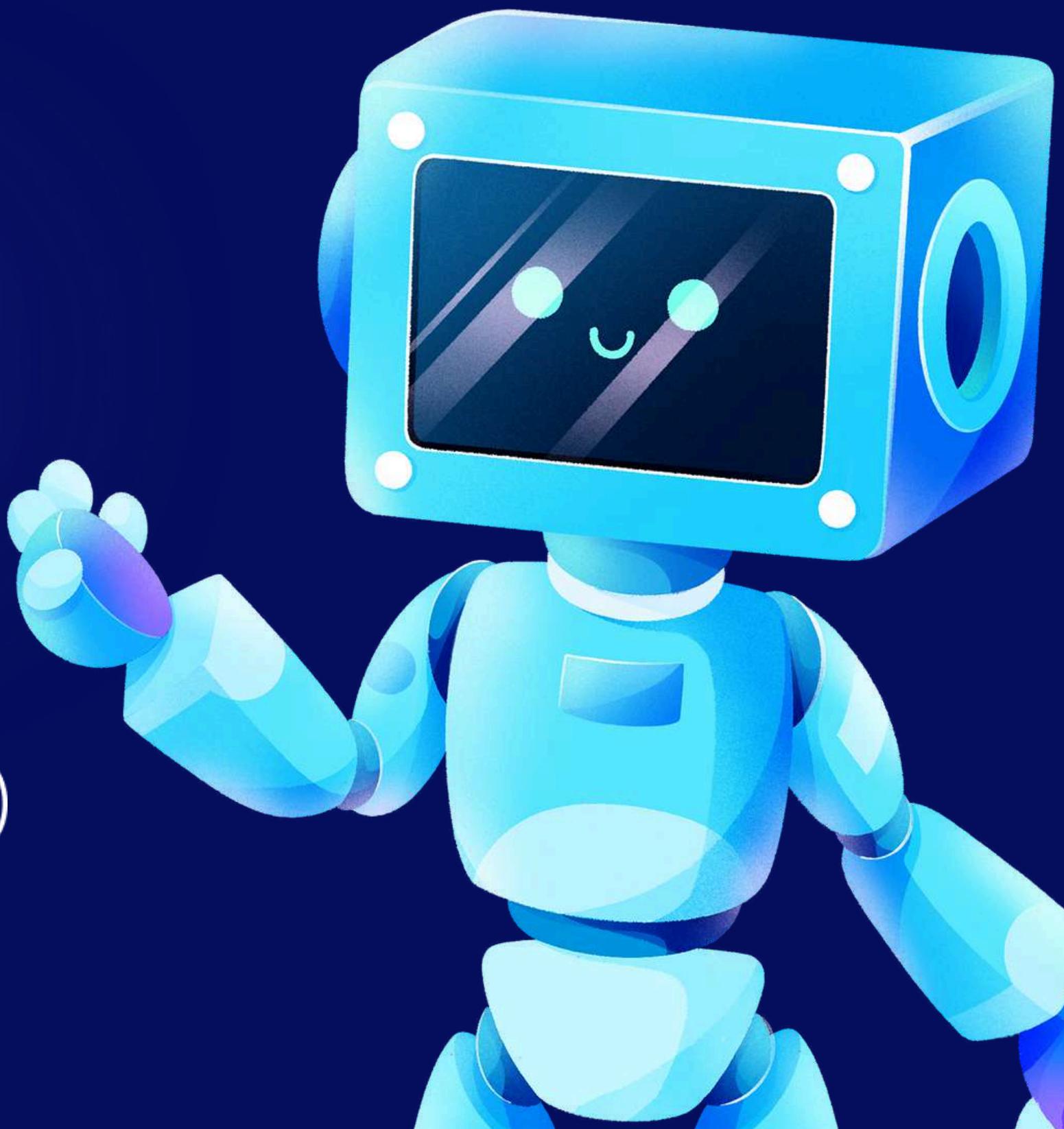




# DATA LOGIC

MANUAL

GITHUB.COM/TULIOMENDESDEV/DATALOGIC





# SUMÁRIO

• Project Overview.....	03
• Metodologia.....	04
• Panorama .....	05
• Introdução ao MongoDB.....	06
• Estrutura.....	07
• Passo 01 (Coleta de dados).....	10
• Passo 02 (Pré-tratamento dos dados).....	11
• Passo 03 (Docker compose).....	12
• Passo 04 (Normalização).....	13
• Passo 05 (Modelagem de dados).....	14
• Passo 06 (Estrutura, consulta e manipulação de dados).....	15
• Passo 07 (Dicionário de dados).....	16
• Passo 08 (Importação e Desnormalização de Dados para o MongoDB Com Python).....	17
• Passo 09 (Automatizando a Migração de Dados: MySQL para MongoDB).....	17
• Passo 10 (Finalização da Migração de Dados para o MongoDB).....	17

# PROJECT OVERVIEW

Este projeto tem como objetivo coletar e manipular o dataset NYC Airbnb Open Data, disponível na plataforma Kaggle. Fazendo gerenciamento em contêiner pelo docker, utilizando bancos SQL e NoSQL. Nossa equipe selecionou a tecnologia MongoDB orientado a documentos.

New York City Airbnb Open Data

Airbnb listings and metrics in NYC, NY, USA (2019)

 <https://www.kaggle.com/datasets/dgomonov/new-york-city-airbnb-open-data>





DATALOGIC



# METODOLOGIA



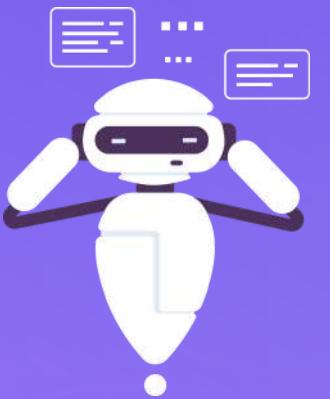
## DATA COLLECTION

Definição do Dataset e  
Reunir dados  
relevantes



## PREPROCESSING

Limpar, normalizar e  
estruturar os dados

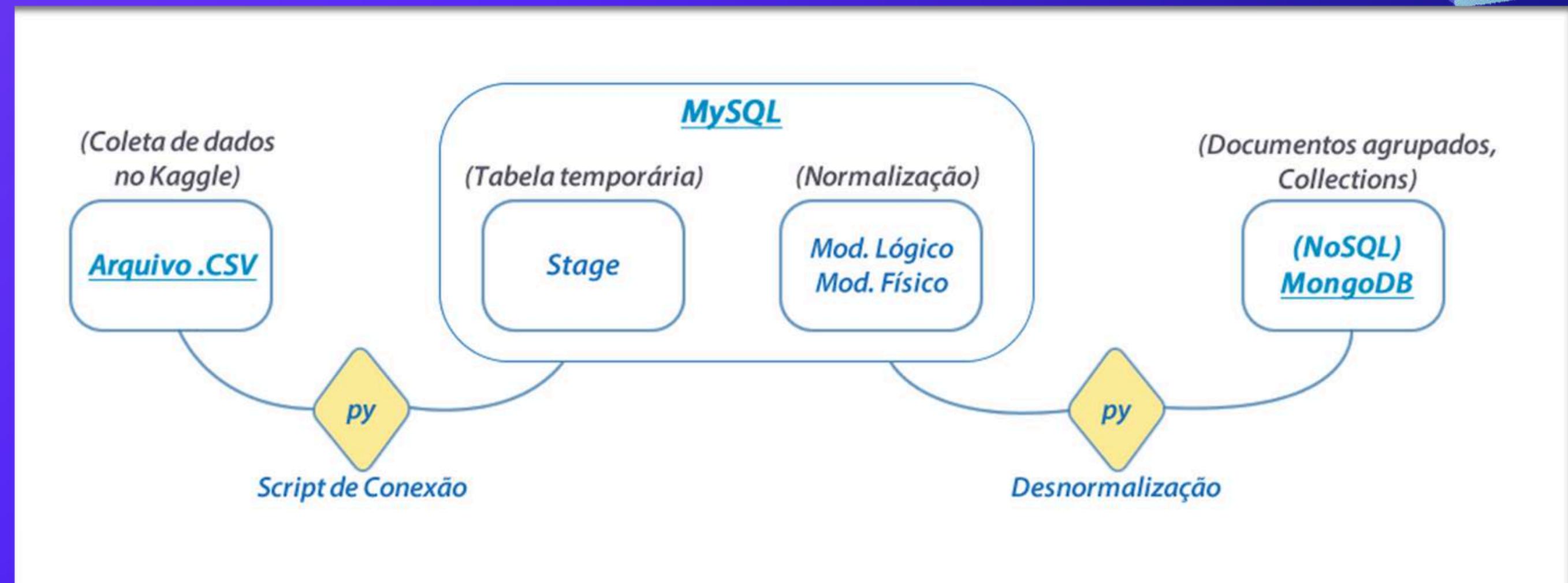


## MODEL SELECTION

Fazer passagem de  
dados  
(SQL para NoSQL)  
MongoDB



# PANORAMA DO PROJETO



# INTRODUÇÃO AO MONGODB

"O **MongoDB** é um banco de dados NoSQL, orientado a documentos, que oferece flexibilidade e escalabilidade. Ele armazena dados em documentos JSON, permitindo esquemas dinâmicos e consultas poderosas. O MongoDB é altamente escalável e distribuído, adequado para aplicativos que precisam lidar com grandes volumes de dados não estruturados e que exigem agilidade no desenvolvimento."

---

Filipe Nascimento

## Amplamente utilizado em diversas áreas

Esses documentos podem ser facilmente escalonados e distribuídos em clusters, tornando o MongoDB uma escolha popular para aplicativos modernos que precisam lidar com grandes volumes de dados e demandas de escalabilidade.



# ESTRUTURA

01

## COLEÇÕES

Os dados no MongoDB são organizados por coleções, que são equivalentes a tabelas em banco de dados relacionais, e cada coleção contém vários documentos.

02

## DOCUMENTOS

Um documento no MongoDB é um objeto JSON (JavaScript Object Notation) que consiste em pares de chave-valor. Os documentos são a unidade básica de armazenamento de dados.

03

## ESQUEMA FLEXÍVEL

O MongoDB possui um esquema flexível, o que significa que os documentos em uma coleção não precisa ter a mesma estrutura ou campos. Isso difere dos bancos de dados relacionais, onde as tabelas têm um esquema rígido e pré-definido.

04

## CAMPOS E VALORES

Cada documento pode ter diferentes campos com tipos de dados variados, como strings, números, arrays, ou até mesmo outros documentos aninhados (subdocumentos).

# ESTRUTURA

05

**IDENTIFICADORES ÚNICOS**  
Cada documento em uma coleção possui um identificador único chamado `_id`. Esse campo é automaticamente gerado se não fornecido explicitamente e garante a unicidade do documento dentro da coleção..

06

**ÍNDICES EFICIENTES**  
O MongoDB suporta índices para permitir consultas rápidas e eficientes. Os índices podem ser criados em um ou mais campos de um documento, o que acelera consultas e operações de busca.

07

**ESCALABILIDADE: REPLICAÇÃO E SHARDING**  
O MongoDB oferece recursos embutidos de replicação e sharding para alta disponibilidade e escalabilidade. A replicação é usada para manter cópias dos dados em vários servidores, enquanto o sharding divide os dados em várias máquinas para distribuir a carga.

08

**CONSULTA USANDO BSON**  
As consultas no MongoDB são feitas usando uma linguagem de consulta baseada em documentos, geralmente usando o formato BSON (Binary JSON), que é uma extensão do JSON que suporta tipos de dados adicionais usados internamente pelo MongoDB.

# ESTRUTURA

09

**CONSULTAS COMPLEXAS**  
O MongoDB oferece suporte a consultas complexas usando sua linguagem de consultas flexível. Além disso, ele possui recursos como agregação para executar operações de análise e transformação de dados no próprio banco de dados.

10

**ARMAZENAMENTO EFICIENTE**  
O MongoDB utiliza um formato de armazenamento otimizado chamado WiredTiger, que oferece compressão de dados e outras otimizações para melhorar o desempenho e a utilização de espaço em disco.

11

**CACHE EM MEMÓRIA**  
O MongoDB pode aproveitar o cache em memória para consultas frequentes e operações de leitura, o que melhora significativamente

# PASSO (01)

## COLETA DE DADOS

A coleta de dados é o processo de reunir informações de diversas fontes para análise, interpretação e tomada de decisões.

Isso pode envolver a obtenção de dados de bancos de dados, registros físicos, dispositivos eletrônicos, formulários online, ou até mesmo a criação de novos conjuntos de dados por meio de pesquisas ou experimentos.

Neste projeto, coletamos dados do dataset New York City Airbnb Open Data disponível na plataforma Kaggle.



# PASSO (02)

## PRÉ-TRATAMENTO DOS DADOS

Após coletar o conjunto de dados (.csv) na plataforma kaggle, foi realizada uma conversão de texto (",") para colunas no Excel, para entender a estrutura e obter uma noção para a normalização.

M15	A	B	C	D	E	F	G	H	I	J	K
1	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights
2	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	4.064.749	-7.397.237	Private room	149	1
3	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	4.075.362	-7.398.377	Entire home	225	1
4	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	4.080.902	-739.419	Private room	150	3
5	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	4.068.514	-7.395.976	Entire home	89	1
6	5022	Entire Apt: Spacious Studio/Loft by centr	7192	Laura	Manhattan	East Harlem	4.079.851	-7.394.399	Entire home	80	10
7	5099	Large Cozy 1 BR Apartment In Midtown E	7322	Chris	Manhattan	Murray Hill	4.074.767	-73.975	Entire home	200	3
8	5121	BlissArtsSpace!	7356	Garon	Brooklyn	Bedford-Stuyvesant	4.068.688	-7.395.596	Private room	60	45
9	5178	Large Furnished Room Near B'way	8967	Shunichi	Manhattan	Hell's Kitchen	4.076.489	-7.398.493	Private room	79	2
10	5203	Cozy Clean Guest Room - Family Apt	7490	MaryEllen	Manhattan	Upper West Side	4.080.178	-7.396.723	Private room	79	2
11	5238	Cute & Cozy Lower East Side 1 bdrm	7549	Ben	Manhattan	Chinatown	4.071.344	-7.399.037	Entire home	150	1
12	5295	Beautiful 1br on Upper West Side	7702	Lena	Manhattan	Upper West Side	4.080.316	-7.396.545	Entire home	135	5
13	5441	Central Manhattan/near Broadway	7989	Kate	Manhattan	Hell's Kitchen	4.076.076	-7.398.867	Private room	85	2
14	5803	Lovely Room 1, Garden, Best Area, Legal	9744	Laurie	Brooklyn	South Slope	4.066.829	-7.398.779	Private room	89	4
15	6021	Wonderful Guest Bedroom in Manhattan	11528	Claudio	Manhattan	Upper West Side	4.079.826	-7.396.113	Private room	85	2



# PASSO (03)

## DOCKER COMPOSE

Docker Compose é uma ferramenta para definição e execução de aplicações multi-container.

### CRIAÇÃO DO DOCKER COMPOSE

Criar o Docker Compose no VS Code, para Conectar Dataset com o MySQL.

### IMPORTAÇÃO DE BIBLIOTECAS

Importar ferramentas de conexão, como o mysql-connector-python; Importar a biblioteca Pandas; Import do SQLAlchemy para a criação da Engine.

### ATRIBUIÇÃO E LEITURA DOS DADOS

Atribuir leitura do arquivo do dataset csv em um Dataframe, através do Pandas, para uma variável.

### ENGINE

Atribuir a string de conexão a uma variável. Criar a Engine com a string de conexão como parâmetro.

### CONEXÃO COM MYSQL

Por fim, utilizamos o comando ".to\_sql" com parâmetros do nome da DB a ser criada.



# PASSO (04)

## NORMALIZAÇÃO

(01)

### PRIMEIRA FORMA NORMAL (1NF)

Cada coluna em uma tabela deve conter apenas valores atômicos (indivisíveis). Cada célula da tabela deve conter um único valor, não listas de valores separados por vírgula ou outro delimitador

---

(02)

### SEGUNDA FORMA NORMAL (2NF)

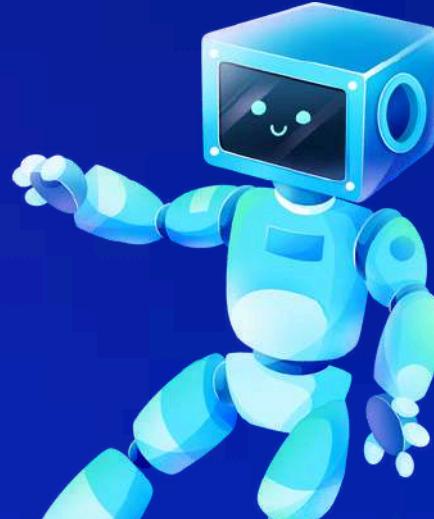
A tabela deve estar na 1NF. Todos os atributos não chave devem depender completamente da chave primária. Em outras palavras, não deve haver dependência parcial da chave primária.

---

(03)

### TERCEIRA FORMA NORMAL (3NF)

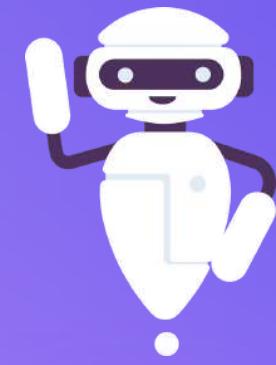
A tabela deve estar na 2NF. Todos os atributos não chave devem depender apenas da chave primária, não de outros atributos não chave. Evite dependências transitivas, ou seja, se A depende de B e B depende de C, então A não deve depender diretamente de C.





## PASSO (05)

## MODELAGEM DE DADOS



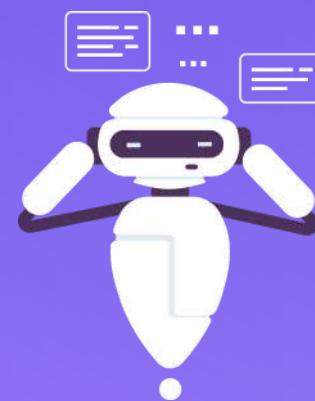
## MODELO CONCEITUAL

Representa o domínio do negócio em um nível de abstração mais alto e objetivo é descrever as principais entidades, relacionamentos e regras de negócio que compõem o domínio do problema.



## MODELO LÓGICO

Representação de dados mais detalhadas do que na modelagem conceitual, com suas entidades, relacionamentos e atributos definidos e como estes dados serão em um banco de dados armazenados



## MODELO FÍSICO

Representa um modelo definido para um banco de dados específico e na construção dos códigos na sua linguagem específica para criação das entidades, atributos, dependências e restrições.

## PASSO (06)

ESTRUTURA, CONSULTA E  
MANIPULAÇÃO DE DADOS

(01)

**DDL's (Data Definition Language)**

- Responsáveis pela definição da estrutura e organização dos objetos no banco de dados, como tabelas, índices, visões e procedimentos armazenados. Exemplos: CREATE, ALTER e DROP.
- 

(02)

**DML's (Data Manipulation Language)**

- Usadas para manipular os dados dentro das tabelas, realizando operações como inserção, atualização, exclusão e recuperação de dados. Exemplos: INSERT, UPDATE, DELETE e SELECT INTO.
- 

(03)

**DQL's (Data Query Language)**

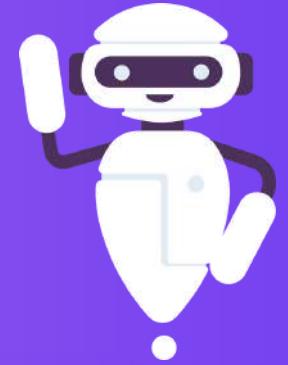
- Responsáveis por recuperar dados específicos do banco de dados. O comando principal da DQL é o SELECT, que permite realizar consultas para recuperar informações de uma ou mais tabelas.



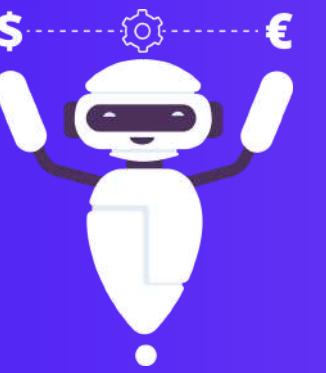
DATALOGIC

PASSO (07)

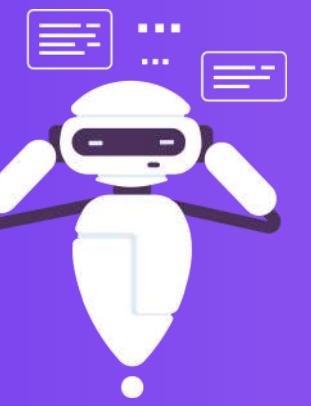
## DICIONÁRIO DE DADOS



INÍCIO DA  
MONTAGEM



CONTAGEM E  
VALIDAÇÃO DE  
DADOS



CONCLUSÃO  
DO  
DICIONÁRIO

## PASSO (08)

IMPORTAÇÃO E DESNORMALIZAÇÃO DE DADOS PARA O MONGODB COM PYTHON

## PASSO (09)

AUTOMATIZANDO A MIGRAÇÃO DE DADOS: MYSQL PARA MONGODB

## PASSO (10)

FINALIZAÇÃO DA MIGRAÇÃO DE DADOS PARA O MONGODB





DATALOGIC

# OUR TEAM

ÂNGELO SANTOS  
01589358



JOSÉ MIGUEL  
01665230



LIVYA CARVALHO  
01645272



THAIS MELO  
01068175



TULIO MENDES  
01633581

# THANK YOU!



DATALOGIC

