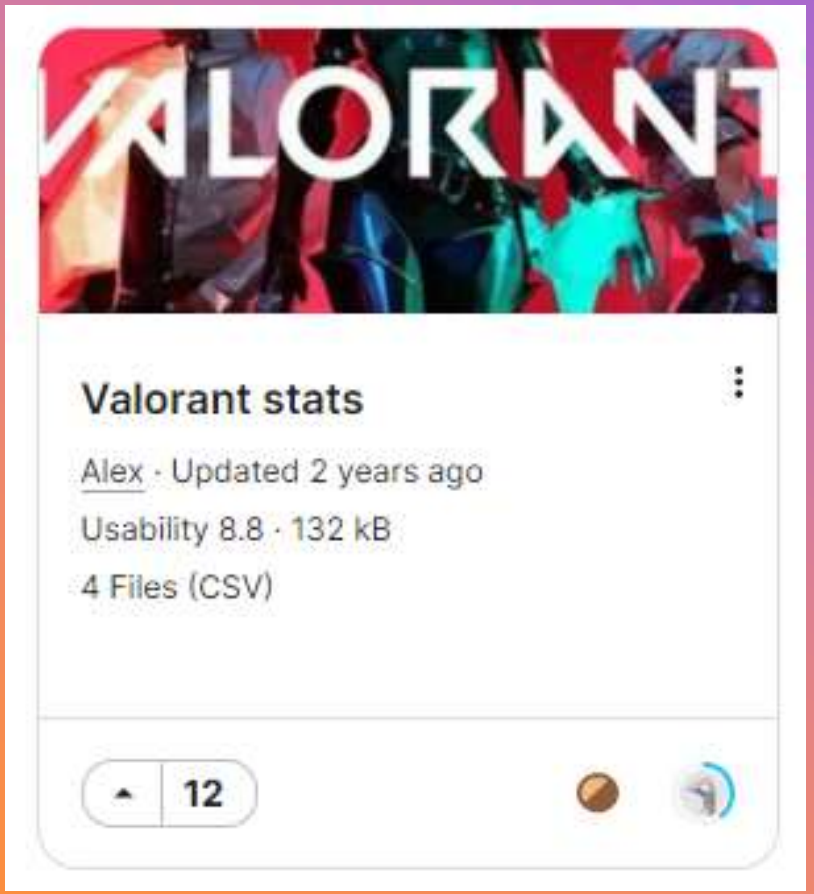
TUTORIAL DO PROJETO

MPTJY TEAM

DATASET



O dataset utilizado foi o Valorant Stats, disponível no site Kaggle.com. A pasta disponível possui 4 arquivos CSV, com informações sobre jogadores, agentes do jogo, mapas do jogo, estatísticas de jogadores e mais. Todos foram coletados em 14/04/2022. Essas são estatísticas coletadas dos últimos 3 meses a partir da data acima.

MPTJY TEAM

SCRIPT MYSQL

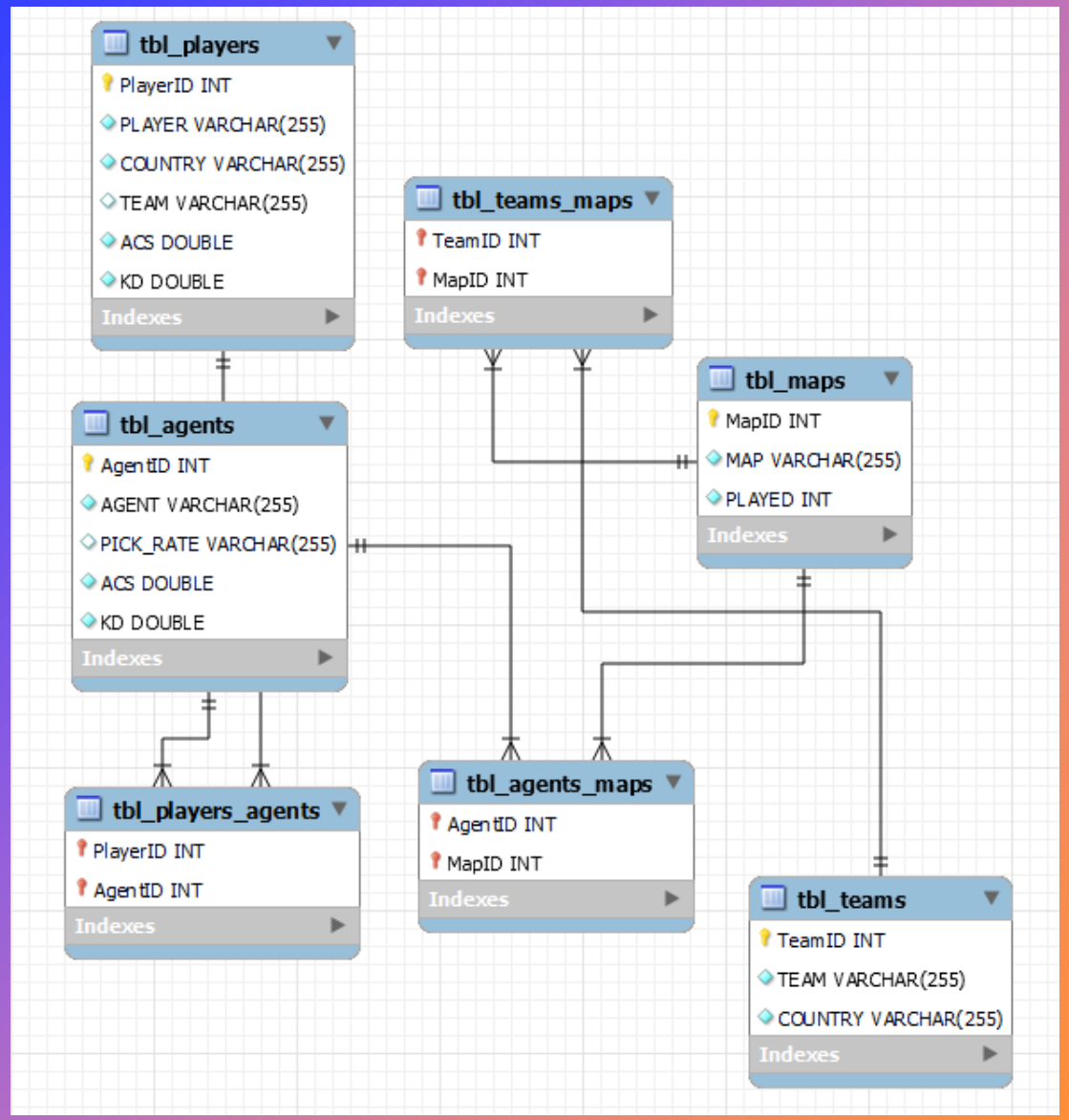
PARTE DO SCRIPT FEITO PARA A CRIAÇÃO DO BANDO DE DADOS.

```sql
CREATE DATABASE IF NOT EXISTS `VAVASTATS` DEFAULT CHARACTER SET utf8mb3 ;
USE `VAVASTATS` ;

CREATE TABLE TBL_AGENTS (
    AgentID INT AUTO_INCREMENT PRIMARY KEY,
    AGENT VARCHAR(255) NOT NULL,
    PICK_RATE DOUBLE NOT NULL,
    ACS DOUBLE NOT NULL,
    KD DOUBLE NOT NULL
);

CREATE TABLE TBL_PLAYERS (
    PlayerID INT AUTO_INCREMENT PRIMARY KEY,
    PLAYER VARCHAR(255) NOT NULL,
    COUNTRY VARCHAR(255) NOT NULL,
    TEAM VARCHAR(255),
    ACS DOUBLE NOT NULL,
    KD DOUBLE NOT NULL
);

CREATE TABLE TBL_MAPS (
    MapID INT AUTO_INCREMENT PRIMARY KEY,
    MAP VARCHAR(255) NOT NULL,
    PLAYED INT NOT NULL
);

CREATE TABLE TBL_TEAMS (
    TeamID INT AUTO_INCREMENT PRIMARY KEY,
    TEAM VARCHAR(255) NOT NULL,
    COUNTRY VARCHAR(255) NOT NULL
);
```

MPTJY TEAM

MODELO LÓGICO

APÓS A NORMALIZAÇÃO DOS DADOS, O MODELO LÓGICO DEFINITIVO FOI CRIADO.



MPTJY TEAM

```python
def extrair_tabela(table_name, csv_filename):
    sql = f"SELECT * FROM {table_name}"

    # Executa a query e salva os resultados em um dataframe
    df = pd.read_sql(sql, con)

    # Caminho arquivo CSV
    csv_path = os.path.join(diretorio, csv_filename)

    # Salva o dataframe
    df.to_csv(csv_path, index=False)


# Cria o diretório se ele não existir
os.makedirs(diretorio, exist_ok=True)

# Chama a função para extrair os dados e salvar em um arquivo CSV
extrair_tabela("show_extract", "show_extract.csv")

# Desliga conexão com o banco
con.close()
```

ARQUIVO PANDAS

EXTRAÇÃO DE DADOS E CRIAÇÃO DOS ARQUIVOS CSV

MPTJY TEAM

```python
def create_indexes(self):
    with self.driver.session() as session:
        # Cria um índice para a propriedade MAP do nó Map
        session.run("CREATE INDEX IF NOT EXISTS FOR (m:Map) ON (m.MAP)")

def import_maps(self, csv_file):
    with self.driver.session() as session:
        with open(csv_file, 'r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            for row in reader:
                # Cria o nó Map com as propriedades do arquivo CSV
                session.run(
                    "CREATE (m:Map {MapID: $MapID, MAP: $MAP, PLAYED: $PLAYED})",
                    parameters={'MapID': int(
                        row['MapID']), 'MAP': row['MAP'], 'PLAYED': int(row['PLAYED'])}
                )
```

```python
def create_indexes(self):
    with self.driver.session() as session:
        # Cria índices para as propriedades dos nós Agent
        session.run(
            "CREATE INDEX IF NOT EXISTS FOR (a:Agent) ON (a.AGENT)")
        session.run(
            "CREATE INDEX IF NOT EXISTS FOR (p:Agent) ON (p.PICK_RATE)")
        session.run("CREATE INDEX IF NOT EXISTS FOR (c:Agent) ON (c.ACS)")

def import_agents(self, csv_file):
    with self.driver.session() as session:
        with open(csv_file, 'r', encoding='utf-8') as file:
            reader = csv.DictReader(file)
            for row in reader:
                # Importa os dados dos agentes do arquivo CSV para o Neo4j
                session.run(
                    "CREATE (a:Agent {AGENT: $AGENT, PICK_RATE: $PICK_RATE, ACS: $ACS})",
                    parameters={
                        'AGENT': row['AGENT'], 'PICK_RATE': row['PICK_RATE'], 'ACS': float(row['ACS'])}
                )
```

EXTRAÇÃO DE DADOS PARA O NEO4J

MPTJY TEAM

COMANDOS CYPHER

COMANDOS QUE FORAM UTILIZADOS PARA A CONSTRUÇÃO DO GRAFO NO NEO4J.

```
- Mostrar a tabela :

MATCH (a:Agent)
RETURN a

- Relacionar as tabelas:

MATCH (a:Agent)-[:JOGA_COM]->(p:Player)
RETURN a.AGENT AS Agente, p.PLAYER AS Jogador

- Mostrar as ligações do nos:

MATCH (a:Agent)-[r]->(p:Player)
RETURN a, r, p

- Criar mais relacionamentos:

MATCH (a:Agent), (m:Map)
CREATE (a)-[:JOGA_EM]->(m)

- Mostrar as ligação :

MATCH (m:Map)-[:JOGA_EM]->(a:Agent)
RETURN m, a
```

MPTJY TEAM

OBRIGADO(A)!!

Equipe:

- **Yuri Gabriel**
- **João Victor Borges de Lira**
- **Márcio da Costa Ferreira Junior**
- **Pedro Cavalcanti**
- **Thaynã Queiroz Mota**

MPTJY TEAM