

## RELATÓRIO T3 - INFRAESTRUTURA PARA GESTÃO DE DADOS

**Nomes: Brenda Pereira Camara, Carolina Michel Ferreira, João Pedro Salles da Silva, Leonardo Bertoletti, Mateus Campos Caçabuena**

a) Link da fonte de dados original relacional:

<https://www.kaggle.com/datasets/adamgbor/club-football-match-data-2000-2025/data>

### **Descrição da construção da estrutura relacional:**

A estrutura relacional foi construída com base na tabela matches, que fornece um conjunto simplificado de informações sobre partidas de futebol. Cada registro representa uma partida específica, tendo:

- Dados da competição (divisão e data)
- Informações dos clubes envolvidos (mandante e visitante)
- Ratings Elo antes da partida
- Desempenho no placar (gols marcados)
- Resultado final da partida (vitória do mandante, empate ou vitória do visitante).

Para garantir uma normalização mínima e clareza, foi definida uma chave primária (id) do tipo SERIAL, assegurando unicidade para cada partida registrada.

O banco de dados relacional foi modelado diretamente no PostgreSQL, utilizando tipos de dados adequados, como DATE para datas, VARCHAR para textos (nomes de clubes e divisões), NUMERIC para valores decimais (como os ratings Elo) e CHAR para os resultados categóricos (H, D, A).

Embora fosse possível realizar uma normalização adicional com tabelas auxiliares para clubes e ratings, neste projeto decidimos por manter uma estrutura única, utilizando apenas a tabela matches, com foco na simplicidade de manipulação em consultas SQL. Essa abordagem facilita a análise direta sobre os dados das partidas e permite cruzamentos rápidos entre atributos como desempenho dos times e efeito do mando de campo.

```
CREATE TABLE matches (
```

```

id SERIAL PRIMARY KEY,
match_date DATE NOT NULL,
home_team VARCHAR(100) NOT NULL,
away_team VARCHAR(100) NOT NULL,
home_goals SMALLINT,
away_goals SMALLINT,
division VARCHAR(5),
home_elo NUMERIC(6,2),
away_elo NUMERIC(6,2),
ft_result CHAR(1) CHECK (ft_result IN ('H', 'D', 'A'))
);

```

Em anexo, a imagem da modelagem:

matches	
id 	SERIAL
match_date	DATE NN
home_team	VARCHAR(100) NN
away_team	VARCHAR(100) NN
home_goals	SMALLINT
away_goals	SMALLINT
division	VARCHAR(5)
home_elo	NUMERIC(6,2)
away_elo	NUMERIC(6,2)
ft_result	CHAR (1)

b) Link da fonte de dados original não relacional

<https://www.kaggle.com/datasets/antimoni/football-stadiums>

### **Descrição da construção da estrutura não relacional:**

A fonte de dados não relacional utilizada foi baseada no arquivo *stadiums.csv*, que contém informações sobre estádios de futebol, como nome, cidade, país, capacidade e equipe mandante. Essa base foi inicialmente obtida em formato CSV e posteriormente convertida para um modelo compatível com o MongoDB.

Para isso, realizamos a análise e limpeza dos dados, padronizando os nomes dos campos e selecionando apenas os atributos mais relevantes para a estrutura da coleção *Stadiums*, como *Stadium*, *City*, *HomeTeams*, *Capacity* e *Country*. Após a formatação, os dados foram inseridos no banco de dados MongoDB hospedado no Azure Cosmos DB com a API MongoDB.

A estrutura da coleção *Stadiums* foi criada no banco *T3-Database* com os campos relevantes do arquivo original *stadiums.csv*. A inserção dos dados foi realizada via importação de um arquivo *.csv*, contendo os documentos estruturados. A criação de índice no campo *HomeTeams* visa otimizar futuras consultas cruzadas com o campo *home\_team* do banco relacional.

```
use futebol_norelacional;  
  
db.createCollection("Stadiums");  
  
db.Stadiums.createIndex({ HomeTeams: 1 });
```

Em anexo, a imagem da modelagem:

stadiums_hackolade_schema			
stadiums_hackolade_sche...			
title	str	*	
type	str	*	
properties	obj	*	
_id	obj	*	
bsonType	str	*	
name	obj	*	
bsonType	str	*	
team	obj	*	
bsonType	str	*	
country	obj	*	
bsonType	str	*	
city	obj	*	
bsonType	str	*	
capacity	obj	*	
bsonType	int	*	
latitude	obj	*	
bsonType	num	*	
longitude	obj	*	
bsonType	num	*	
opened	obj	*	
bsonType	int	*	
required	arr	*	
[0]	str		

c) código de pelo menos 3 consultas

## Consulta 1

```

SELECT *

FROM OPENROWSET(

    BULK
    'https://infrabdstoraget3.dfs.core.windows.net/dados-t3/matches.csv',

    FORMAT = 'CSV',

    FIELDTERMINATOR = ',',

    FIELDQUOTE = '"',

    ROWTERMINATOR = '\n',

    HEADER_ROW = TRUE,

```

```

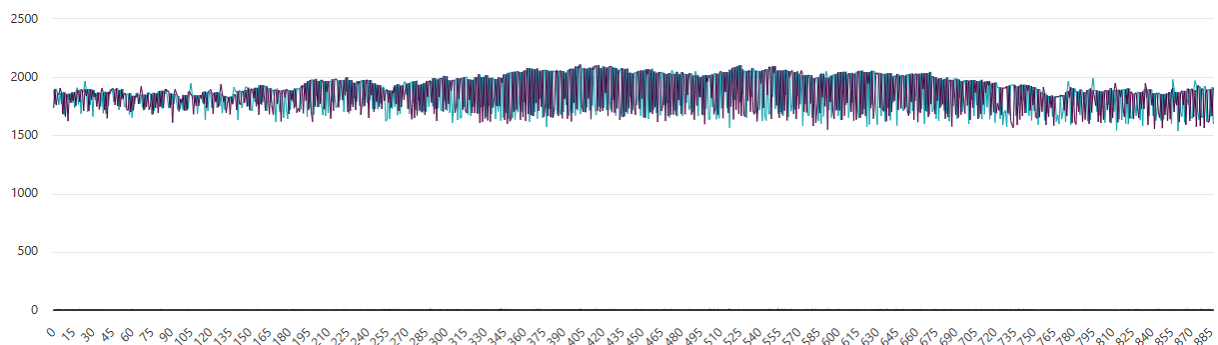
    PARSER_VERSION = '2.0'

) AS MATCHES

WHERE home_team = 'Barcelona' OR away_team = 'Barcelona';

```

Essa consulta tem como objetivo recuperar todos os registros de partidas em que o clube “Barcelona” atuou como mandante (*home\_team*) ou visitante (*away\_team*). Ela permite análises específicas relacionadas ao desempenho ou frequência de jogos desse time no período coberto pelos dados. Segue abaixo a imagem do gráfico do retorno da consulta:



## Consulta 2

```

SELECT Stadium, City, Country, Capacity

FROM OPENROWSET (

    BULK
    'https://infrabdstoraget3.dfs.core.windows.net/dados-t3/stadiums.csv',

    FORMAT = 'CSV',

    FIELDTERMINATOR = ',',

    FIELDQUOTE = '"',

    ROWTERMINATOR = '\n',

    HEADER_ROW = TRUE,

    PARSER_VERSION = '2.0'

```

```

)

WITH (

    [Confederation] VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

    [Stadium]          VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

    [City]              VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

    [HomeTeams]         VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

    [Capacity]          INT,

    [Country]           VARCHAR(50) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

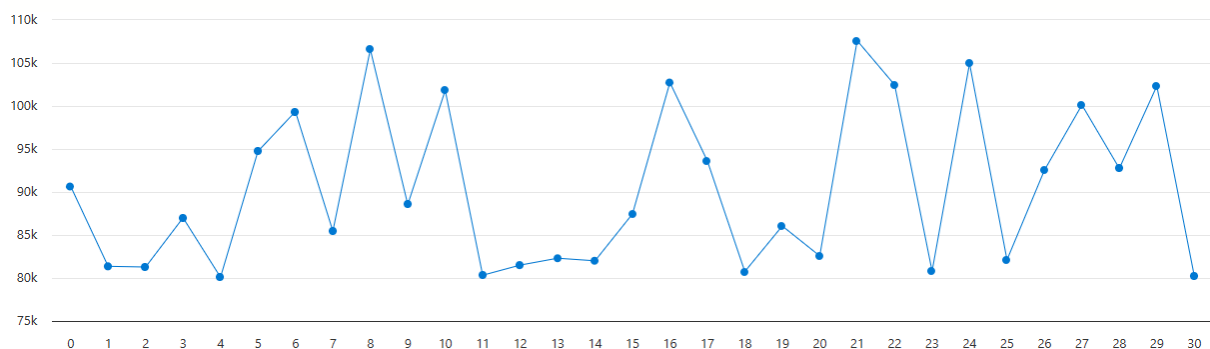
    [IOC]                VARCHAR(10) COLLATE
Latin1_General_100_CI_AS_SC_UTF8

) AS STADIUMS

WHERE Capacity > 80000;

```

Essa consulta filtra os estádios com capacidade superior a 80.000 lugares, permitindo identificar as maiores arenas do conjunto de dados. Isso pode ser útil para análises relacionadas à infraestrutura esportiva e impacto do público nos jogos. Segue abaixo a imagem do gráfico do retorno da consulta:



### Consulta 3

```
SELECT

    MATCHES.match_date,

    MATCHES.home_team,

    STADIUMS.Stadium AS stadium_name,

    STADIUMS.City,

    STADIUMS.Capacity

FROM OPENROWSET(

    BULK

    'https://infrabdstoraget3.dfs.core.windows.net/dados-t3/matches.csv',

    FORMAT = 'CSV',

    FIELDTERMINATOR = ',',

    FIELDQUOTE = '"',

    ROWTERMINATOR = '\n',

    HEADER_ROW = TRUE,

    PARSER_VERSION = '2.0'

)

WITH (

    match_date DATE,

    home_team VARCHAR(100) COLLATE

Latin1_General_100_CI_AS_SC_UTF8,
```

```

        away_team VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

        home_goals INT,

        away_goals INT,

        division VARCHAR(10) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

        home_elo FLOAT,

        away_elo FLOAT,

        ft_result VARCHAR(10) COLLATE
Latin1_General_100_CI_AS_SC_UTF8
    )

    AS MATCHES

INNER JOIN OPENROWSET(

    BULK
'https://infrabdstoraget3.dfs.core.windows.net/dados-t3/stadiums.csv',

    FORMAT = 'CSV',

    FIELDTERMINATOR = ',',

    FIELDQUOTE = '"',

    ROWTERMINATOR = '\n',

    HEADER_ROW = TRUE,

    PARSER_VERSION = '2.0'

)

WITH (

```



```

[S.No]          INT,

[Confederation] VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

[Stadium]       VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

[City]          VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

[HomeTeams]     VARCHAR(100) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

[Capacity]      INT,

[Country]       VARCHAR(50) COLLATE
Latin1_General_100_CI_AS_SC_UTF8,

[IOC]           VARCHAR(10) COLLATE
Latin1_General_100_CI_AS_SC_UTF8

) AS STADIUMS

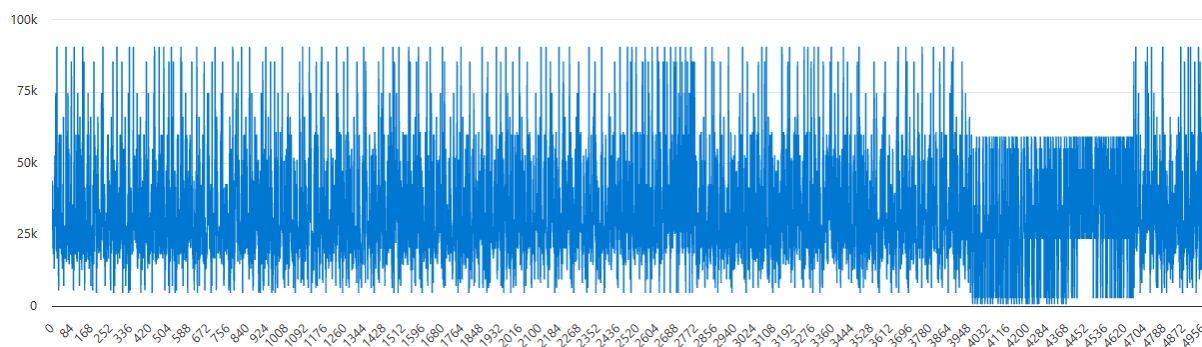
ON MATCHES.home_team = STADIUMS.HomeTeams

WHERE MATCHES.home_team <> STADIUMS.City;

```

Essa consulta cruza as informações de partidas com dados dos estádios, retornando os casos em que o time mandante (*home\_team*) joga em um estádio localizado em uma cidade diferente de seu nome. Isso ajuda a identificar clubes que atuam em cidades-sede distintas, como clubes que utilizam estádios emprestados, compartilham arenas ou representam regiões mais amplas.

O uso da cláusula COLLATE Latin1\_General\_100\_CI\_AS\_SC\_UTF8 nas colunas de texto foi necessário para evitar erros de leitura ao lidar com arquivos em codificação UTF-8 contendo acentos e caracteres especiais. Essa prática garante a correta leitura dos dados pelo Synapse Analytics. Segue abaixo a imagem do gráfico do retorno da consulta:



#### d) Citação dos recursos utilizados






##### Citação de uso de IA

Durante o desenvolvimento, utilizamos plataformas de inteligência artificial, como o ChatGPT, para auxiliar na redução e simplificação das colunas do dataset *stadiums.csv*, a fim de adequá-lo à estrutura esperada pelo banco de dados não relacional e facilitar a integração com a outra fonte de dados.

##### e) Observações adicionais

A coleção Stadiums foi estruturada de forma a permitir o cruzamento com a base relacional de partidas de futebol, utilizando o campo *HomeTeams* como chave de ligação com a coluna *home\_team* da tabela *matches*. Essa relação possibilita análises integradas, como verificar a capacidade do estádio associado a uma partida ou a cidade onde o time mandante costuma jogar.

Após criado, não foi possível alterar os nomes dos recursos utilizados para o nome da equipe conforme pedido. Portanto, aqui estão os nomes para facilitar a verificação da existência destes mesmos recursos:

<input type="checkbox"/>	Nome		Tipo	Localização
<input type="checkbox"/>	 <a href="#">infrabd-statsbomb-mongo</a>	...	Azure Cosmos DB for MongoDB (vCore)	Brazil South
<input type="checkbox"/>	 <a href="#">infrabd-datalake-t3</a>	...	Synapse workspace	Brazil South
<input type="checkbox"/>	 <a href="#">infrabdstorage-t3</a>	...	Conta de armazenamento	
<input type="checkbox"/>	 <a href="#">infrabd-postgre-resources</a>	...	Grupo de recursos	
<input type="checkbox"/>	 <a href="#">pg-clubmatches-kaggle2</a>	...	Banco de Dados do Azure para servidor...	Brazil South