

## Índice

1- Introdução	3
2- Modelo Entidade-Associação	4
3- Modelo Relacional	7
4- SQL como LDD	8
5- SQL como LMD	14
6- Conclusões	16

## Introdução

Para este trabalho foi escolhida como tema, uma API (Application programming interface) que fornece resultados de jogos em tempo real assim como informações de jogadores de diversos desportos.

Os objetivos desta base de dados serão armazenar e organizar corretamente os dados, como os tipos de desportos que a API possui, os países e as ligas que as constituem, assim como as equipas que lhes pertencem.

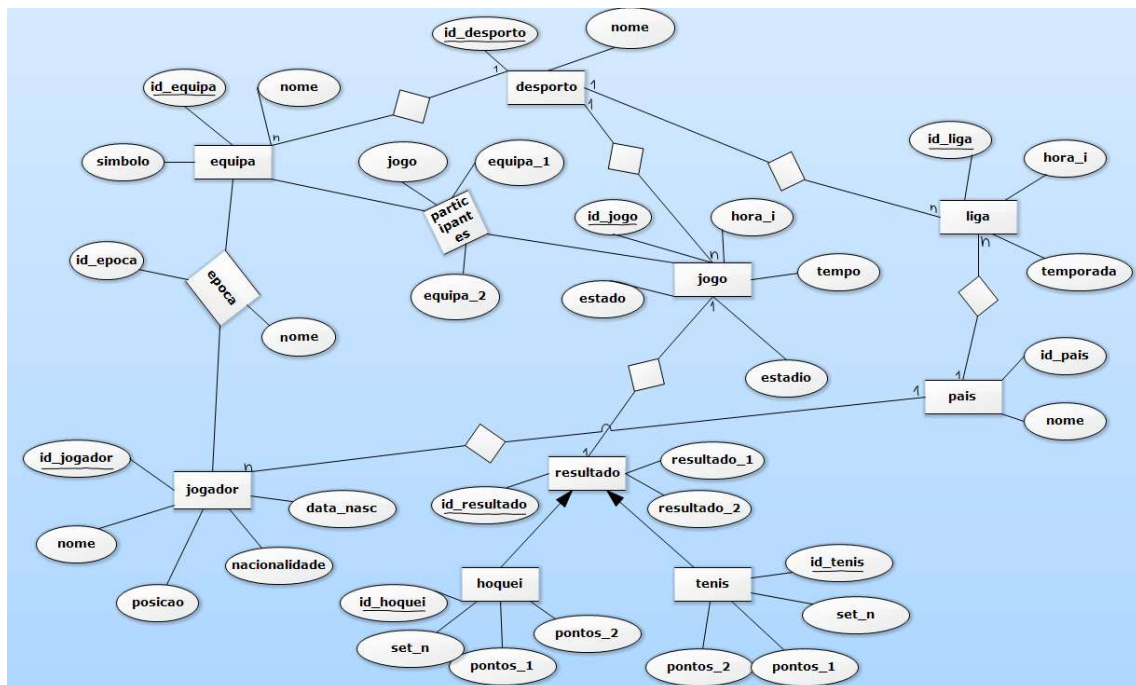
Apesar de não representar totalmente o nosso projeto conjunto que é, um site de resultados online que possui outras funcionalidades para além do uso desta API para fornecer os resultados, esta está ligada diretamente a ele, representando uma parte essencial do site, sem o qual este não seria possível.

Para a implementação correta da base de dados, de modo a ela ser 100% funcional e livre de erros, foram desenhados e projetados vários tipos de modelos com o propósito de verificar e testar o funcionamento da base de dados.

## Modelo Entidade-Associação

O primeiro modelo concebido foi o de entidade-associação para desenhar a estrutura da base de dados e organizar as ligações necessárias entre as entidades criadas.

Abaixo, uma imagem do modelo entidade-associação:



O esquema apresentado acima, não possui referências a chaves estrangeiras, estando estas representadas no modelo que se encontra abaixo, o modelo relacional.

Voltando agora ao modelo entidade-associação, este é composto por 9 entidades e, 2 associações com atributos próprios.

É também utilizado o símbolo , para representar uma associação ISA.

Todas as entidades têm atribuída a multiplicidade correta, exceto as associações com atributos próprios (mais concretamente, “Participantes” e “Epoca”), estas apenas têm a ligação às entidades que ela interliga.

Para a construção deste modelo, consideramos 9 entidades como já referi, sendo estas:

- Equipa
- Desporto
- Jogo
- Resultado
- Hoquei
- Tenis
- Pais
- Jogador
- Liga

Consideramos também, 2 associações com atributos próprios:

- Epoca
- Participantes

Como a API apresenta resultados online como já referimos de diversos desportos, é essencial termos uma entidade (“Desporto”) que represente os vários tipos de desportos que a API suporta.

Como estamos a falar de desportos, é obrigatório termos entidades que representem, as ligas que esse desporto possui, o país a que essas ligas pertencem, e, as equipas que cada liga tem.

Para isto, foram criadas as entidades “Pais”, “Liga” e “Equipa”.

Após tudo isto, foi necessário criar uma entidade que representasse os jogos que irão ocorrer, mostrando informações sobre esse jogo como por exemplo, a hora a que ele se inicia e em que estádio irá decorrer a partida, e também, uma associação denominada “Participantes”, de modo a ser possível mostrar quais as equipas a defrontar-se nesses jogos.

Mas, tudo isto não fará sentido senão tivermos um método para representar o resultado de cada jogo, que é o aspeto mais importante.

Para resolver este problema, criou-se então uma entidade chamada “Resultado” que irá mostrar os golos que cada equipa marcou.

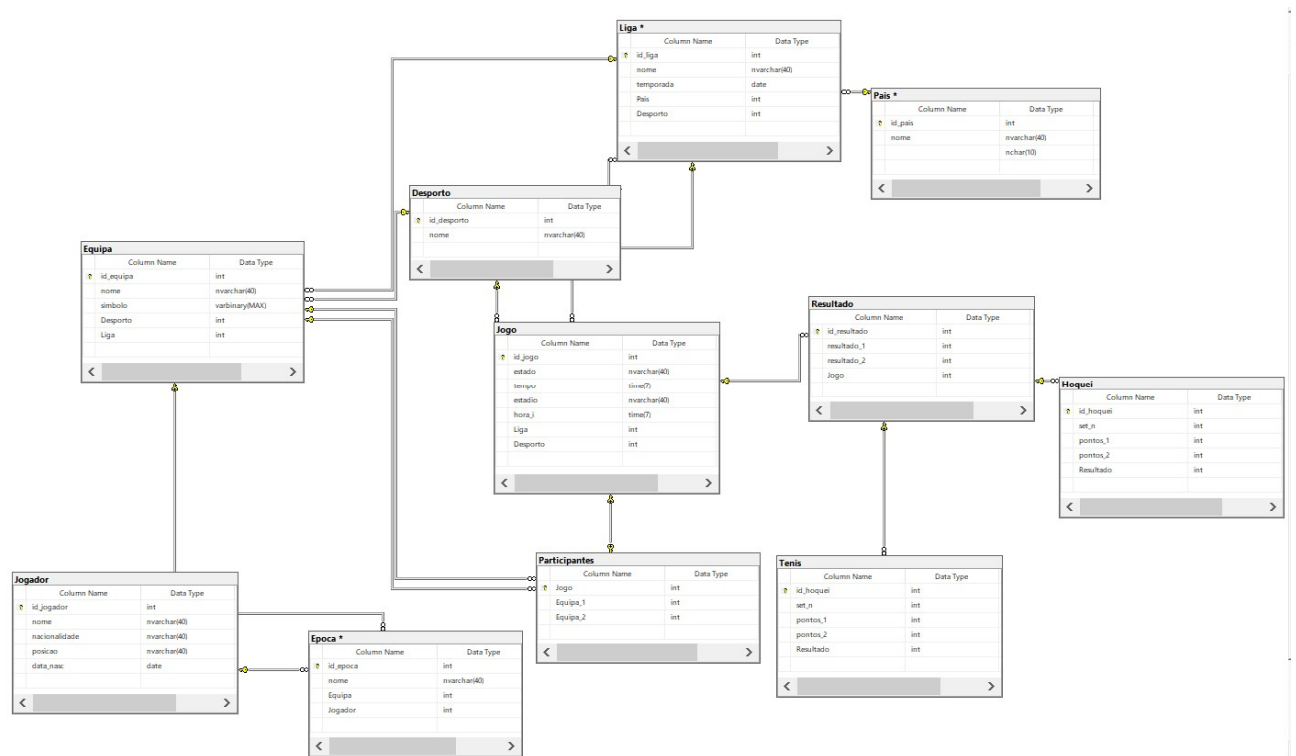
Como não existe só futebol, apesar de em Portugal esta ser a realidade, tivemos de criar mais 2 entidades que tornassem possível representar desportos, em que os resultados têm um formato mais complexo, como por exemplo, Ténis e Hóquei em que existem vários períodos durante um jogo, tendo cada período um número de pontos marcado por cada equipa.

Para os outros desportos, o modelo usado no futebol é compatível e pode ser usado para representar os resultados, não sendo necessário criar mais nenhuma entidade.

Como referimos antes, a API mostra também, para além de resultados online, informações de jogadores, como por exemplo em que equipa é que eles estão na época atual, assim como a sua data de nascimento e o seu país de origem. Foram então criadas, a entidade “Jogador” que possui as características pessoais do mesmo, e uma associação chamada “Epoca” onde serão guardadas a data da época, e o nome da equipa onde o jogador esteve nessa época.

# Modelo Relacional

Tendo representado o modelo entidade-associação, passamos agora ao modelo relacional:



No modelo relacional são agora referidas todas as ligações entre as entidades, mostrando as chaves estrangeiras e a qual atributo de uma certa entidade ela faz referência.

## SQL como LDD

Após a criação e verificação dos modelos entidade-associação e modelo relacional, procedeu-se à implementação em SQL da base de dados.

Criação da base de dados e respetivas tabelas, assim como a introdução de dados:



```
CREATE DATABASE bd_trab1;  
USE bd_trab1;
```

```
CREATE TABLE Equipa(  
    id_equipa INT PRIMARY KEY,  
    nome NVARCHAR(40),  
    simbolo VARBINARY(MAX)  
);
```

```
CREATE TABLE Jogador(  
    id_jogador INT PRIMARY KEY,  
    nome NVARCHAR(40),  
    nacionalidade NVARCHAR(40),  
    posicao NVARCHAR(40),  
    data_nasc date  
);
```

```
CREATE TABLE Desporto(  
    id_desporto INT PRIMARY KEY,  
    nome NVARCHAR(40)  
);
```

```
CREATE TABLE Pais(  
    id_pais INT PRIMARY KEY,  
    nome NVARCHAR(40)  
);
```

```
CREATE TABLE Liga(  
    id_liga INT PRIMARY KEY,  
    nome NVARCHAR(40),  
    temporada date  
);
```

```

]CREATE TABLE Jogo(
    id_jogo INT PRIMARY KEY,
    estado NVARCHAR(40),
    tempo TIME,
    estadio NVARCHAR(40),
    hora_i TIME
) ;

]CREATE TABLE Epoca(
    id_epoca INTEGER PRIMARY KEY,
    nome NVARCHAR(40)
) ;

]CREATE TABLE Participantes(
    Jogo INT PRIMARY KEY
) ;

]CREATE TABLE Resultado(
    id_resultado INT PRIMARY KEY,
    resultado_1 INTEGER,
    resultado_2 INTEGER
) ;

]CREATE TABLE Hoquei(

    id_hoquei INTEGER PRIMARY KEY,
    set_n INTEGER,
    pontos_1 INTEGER,
    pontos_2 INTEGER
) ;

]CREATE TABLE Tenis(
    id_hoquei INTEGER PRIMARY KEY,
    set_n INTEGER,
    pontos_1 INTEGER,
    pontos_2 INTEGER
) ;

```

```
ALTER TABLE Equipa ADD Desporto INTEGER;
ALTER TABLE Equipa ADD FOREIGN KEY (Desporto) REFERENCES Desporto(id_desporto);
ALTER TABLE Equipa ADD Liga INTEGER;
ALTER TABLE Equipa ADD FOREIGN KEY (Liga) REFERENCES Liga(id_liga);

ALTER TABLE Epoca ADD Equipa INTEGER;
ALTER TABLE Epoca ADD FOREIGN KEY (Equipa) REFERENCES Equipa(id_equipa);
ALTER TABLE Epoca ADD Jogador INTEGER;
ALTER TABLE Epoca ADD FOREIGN KEY (Jogador) REFERENCES Jogador(id_jogador);

ALTER TABLE Jogo ADD Liga INTEGER;
ALTER TABLE Jogo ADD FOREIGN KEY (Liga) REFERENCES Liga(id_liga);
ALTER TABLE Jogo ADD Desporto INTEGER;
ALTER TABLE Jogo ADD FOREIGN KEY (Desporto) REFERENCES Desporto(id_desporto);

ALTER TABLE Participantes ADD Equipa_1 INTEGER;
ALTER TABLE Participantes ADD FOREIGN KEY (Equipa_1) REFERENCES Equipa(id_equipa);
ALTER TABLE Participantes ADD Equipa_2 INTEGER;
ALTER TABLE Participantes ADD FOREIGN KEY (Equipa_2) REFERENCES Equipa(id_equipa);
ALTER TABLE Participantes ADD FOREIGN KEY (Jogo) REFERENCES Jogo(id_jogo);

ALTER TABLE Liga ADD Pais INTEGER;
ALTER TABLE Liga ADD FOREIGN KEY (Pais) REFERENCES Pais(id_pais);
ALTER TABLE Liga ADD Desporto INTEGER;
ALTER TABLE Liga ADD FOREIGN KEY (Desporto) REFERENCES Desporto(id_desporto);

ALTER TABLE Resultado ADD Jogo INTEGER;
ALTER TABLE Resultado ADD FOREIGN KEY (Jogo) REFERENCES Jogo(id_jogo);

ALTER TABLE Hoquei ADD Resultado INTEGER;
ALTER TABLE Hoquei ADD FOREIGN KEY (Resultado) REFERENCES Resultado(id_resultado);

ALTER TABLE Tennis ADD Resultado INTEGER;
ALTER TABLE Tennis ADD FOREIGN KEY (Resultado) REFERENCES Resultado(id_resultado);
```

```
insert into Desporto values
```

```
(1, 'Soccer'),  
(2, 'Basketball'),  
(3, 'Volleyball'),  
(4, 'Hockey'),  
(5, 'Tennis'),  
(6, 'A.M Football')
```

```
insert into Pais values
```

```
(1, 'Portugal'),  
(2, 'Brazil'),  
(3, 'England'),  
(4, 'Spain'),  
(5, 'Argentina'),  
(6, 'Egypt'),  
(7, 'Colombia'),  
(8, 'Germany'),  
(9, 'Belgium'),  
(10, 'Italy')
```

```
insert into Jogador values
```

```
(1, 'Cristiano Ronaldo', 1, 'Forward', '02-05-1985'),  
(2, 'Mohamed Salah', 6, 'Forward', '06-15-1992'),  
(3, 'Manuel Neuer', 8, 'Goalkeeper', '03-27-1986'),  
(4, 'Kevin De Bruyne', 9, 'Right Midfielder', '06-28-1991'),  
(5, 'Alex Sandro', 2, 'Left Back', '01-26-1991')
```

```
]insert into Liga values
```

```
(1,'Liga NOS','01-18-2019',1,1),  
(2,'Bundesliga','01-18-2019',8,1),  
(3,'Premier League','01-18-2019',3,1),  
(4,'Liga BBVA','01-18-2019',4,1) ,  
(5,'Serie A','01-18-2019',10,1),  
(6,'DEL','01-18-2019',8,4)
```

```
]insert into Equipa values
```

```
(1,'Juventus FC',NULL,1,5),  
(2,'Liverpool FC',NULL,1,3),  
(3,'FC Bayern Munich',NULL,1,2),  
(4,'Real Madrid FC',NULL,1,4),  
(5,'Vitoria FC',NULL,1,1),  
(6,'Manchester United FC',NULL,1,3),  
(7,'AS Roma',NULL,1,5),  
(8,'Borussia Dortmund',NULL,1,2),  
(9,'Augsburg',NULL,4,6),  
(10,'Düsseldorf',NULL,4,6)
```

```
]insert into Epoca values
```

```
(1,'2018-2019',1,1),  
(2,'2018-2019',2,2),  
(3,'2018-2019',3,3),  
(4,'2018-2019',1,5)
```

```
]insert into Jogo values
```

```
(1,'In progress','01:26:23','Allianz Stadium','08:00:00PM',5,1),  
(2,'Finished','00:00:00','Anfield','08:00:00PM',3,1),  
(3,'In progress','00:39:23','Santiago Bernabéu Stadium','09:30:00PM',4,1),  
(4,'In progress','00:23:10','Allianz Arena','08:00:00PM',2,1),  
(5,'Finished','00:00:00','Curt Frenzel Stadium','01:00:00PM',6,4)
```

```
]insert into Participantes values
```

```
(2,2,6),  
(4,3,8),  
(1,1,7),  
(5,9,10)
```

```
]insert into Resultado values
```

```
(1,1,3,2),  
(2,1,1,3),  
(3,1,1,4),  
(4,2,0,1),  
(5,3,6,5)
```

```
]insert into Hoquei values
```

```
(3,1,0,0,5),  
(1,2,1,3,5),  
(2,3,2,3,5)
```



## SQL como LMD

Em relação ao SQL como LMD, não nos foi possível usar os “selects” que fizemos na base de dados no nosso projeto, uma vez que como já referimos, apesar de o nosso tema (API) ser uma parte essencial do nosso Website, não representa a totalidade do nosso projeto.

Abaixo, as queries que foram efetuadas à base de dados (incluindo as vistas e os índices):

```
/* Mostrar quais os jogos que já terminaram, indicando a hora que começaram e o estádio em que decorreram */  
SELECT Jogo.hora_i, Jogo.estadio FROM Jogo WHERE Jogo.estado='Finished'
```

	hora_i	estadio
1	20:00:00.0000000	Anfield
2	13:00:00.0000000	Curt Frenzel Stadium

```
/* Indicar quais as equipas que ainda não jogaram */  
SELECT Equipa.nome FROM Equipa  
INNER JOIN Jogo ON Equipa.Liga NOT IN(  
    SELECT Jogo.Liga FROM Jogo  
) GROUP BY Equipa.nome
```

	nome
1	Vitoria FC

```
/* Indicar o nome das equipas que pertencem à Premier League, criando primeiro uma vista que obtenha os id's das equipas  
pertencentes a essa liga, e utilizar essa vista para obter os nomes */
```

```
CREATE VIEW equipas_liga_inglesa AS  
SELECT Equipa.id_equipa FROM Equipa WHERE Equipa.Liga=3  
GROUP BY Equipa.id_equipa  
  
SELECT Equipa.nome FROM equipas_liga_inglesa  
INNER JOIN Equipa ON equipas_liga_inglesa.id_equipa=Equipa.id_equipa
```

	nome
1	Liverpool FC
2	Manchester United FC

```
/* Alterar a equipa a que um jogador pertence caso ocorra uma transferência */  
UPDATE Epoca SET Equipa=4 WHERE Jogador=1
```

```
/* Criar um índice para o nome das equipas */  
CREATE INDEX nomes_equipas ON Equipa(nome)
```

## Conclusões

Este trabalho foi realizado com o intuito de compreender melhor um segmento do nosso projeto, que nos foi disponibilizado por um outro website, sob certas condições.

Com a construção desta base de dados e dos vários modelos para a representar assim como as respetivas queries para testar o seu funcionamento, conseguimos obter um conhecimento mais profundo acerca deste tema com o qual não estávamos familiarizados, preparando-nos assim para projetos a realizar no futuro.