



Universidade do Minho-Escola de Engenharia  
2017/2018  
3ºAno

# Relatório do Projeto

Tecnologias de Bases de Dados

MIETI

Elaborado por: David Alves A79625

## Conteúdo

1.Introdução.....	2
2.Desenvolvimento do projeto.....	3
2.1. Enunciado do projeto.....	3
2.2. Modelo Conceptual de Dados/DER .....	4
2.3. Esquema Relacional.....	5
2.4. Explicação das <i>scripts</i> de criação da base de dados em mySQL .....	6
3.Instruções de utilização .....	26
4.Dicionário de dados .....	35
5.Conclusões.....	38

## 1.Introdução

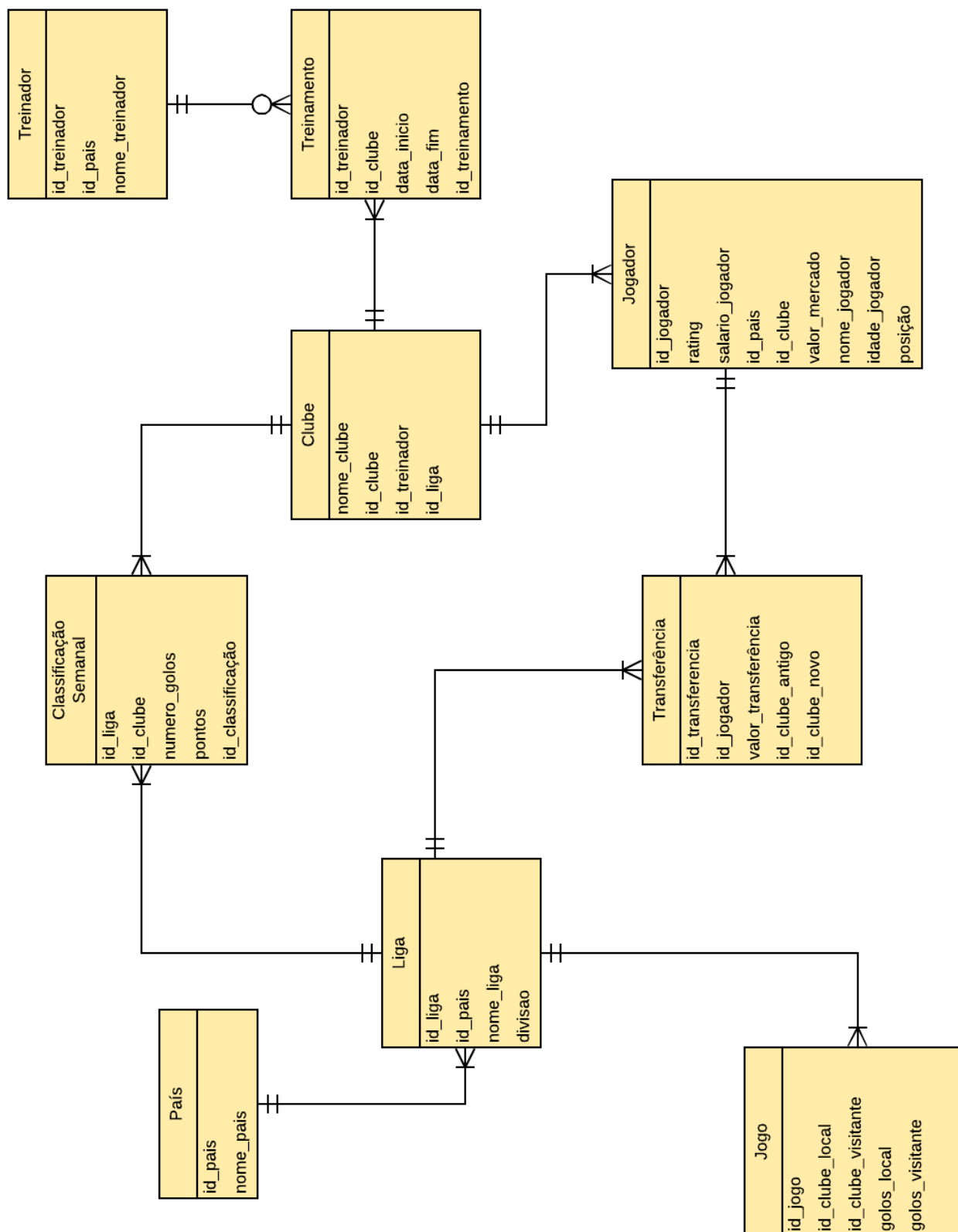
No âmbito da unidade curricular de Tecnologias de Bases de Dados, foi proposto aos alunos a elaboração de um projeto no qual fosse exemplificado os conteúdos apreendidos ao longo do semestre. Para isso o docente pediu a cada grupo que desenvolvesse a sua própria proposta de projeto. No meu caso o meu projeto é sobre uma base de dados para um videojogo de futebol, onde constará todas as informações relativas ao mundo de futebol, tais como dados dos jogadores, de treinadores, de clubes, e como estes se relacionam, havendo também a possibilidade de alterações e inserção de dados reativos a cada um destes elementos

## 2.Desenvolvimento do projeto

### 2.1. Enunciado do projeto

O *Football Engine*, um videojogo de futebol pretende desenvolver a sua base de dados, a fim de proporcionar a melhor experiência de jogo aos seus utilizadores. O jogo fundamentalmente permite cada utilizador ser **treinador** de um clube de futebol, sendo que os treinadores têm as seguintes características: são identificados por um id de treinador criado automaticamente pela base de dados, além disso são conhecidos os seus nomes e nacionalidades do treinador. Cada treinador treina um **clube**, cada clube pertence a uma determinada **liga** de um dado **país**. Quando um treinador entra para este jogo não tem nenhum clube associado, entra para um contrato de emprego chamado **Treinamento** no qual fica à espera de poder ser contratado por um clube, sendo que ao longo da sua carreira pode treinar vários clubes, sendo os mesmos registados no seu historial. Cada clube tem um conjunto de **jogadores**. Cada jogador é identificado por um id criado automaticamente pela base de dados, além disso são conhecidos o clube a que pertence, a sua nacionalidade, nome, rating, valor de mercado, salário, posição e idade. Cada treinador após negociar com outro treinador pode concordar em comprar ou vender determinado jogador, sendo que a **transferência** só pode ser feita entre clubes da mesma liga e o valor de mercado do jogador passa a ser o valor da última transferência. Cada equipa faz **jogos** apenas com equipas da sua liga, ficando registado na base de dados em cada jogo os golos de cada equipa (local e visitante), sendo após cada jogo alterada a pontuação de cada equipa numa **classificação semanal**.

## 2.2. Modelo Conceptual de Dados/DER



### 2.3. Esquema Relacional

País (id\_pais, nome\_pais)

Liga (id\_liga, id\_pais, divisao, nome\_liga)

Clube (id\_clube, id\_liga, id\_treinador, nome\_clube)

Treinamento (id\_treinamento, id\_treinador, id\_clube,  
data\_inicio, data\_fim)

Treinador (id\_treinador, id\_pais, nome\_treinador)

Jogador (id\_jogador, id\_clube, id\_pais, nome\_jogador,  
rating, valor\_de\_mercado, salario\_jogador, idade\_jogador,  
posicao)

Transferencia (id\_transferencia, id\_jogador,  
valor\_transferencia, id\_clube\_antigo, id\_clube\_novo)

Jogo (id\_jogo, id\_clube\_local, id\_clube\_visitante,  
golos\_local, golos\_visitante)

Classificação Semanal (id\_classificacao, id\_liga, id\_clube,  
numero\_golos, pontos)

## 2.4. Explicação das *scripts* de criação da base de dados em mySQL

### Tabelas

#### -Pais

```
create table pais(  
id_pais int not null auto_increment,  
nome_pais varchar(100),  
  
primary key(id_pais)  
);
```

Como se pode observar esta tabela é constituída por uma chave primária que identifica o país chamada id\_pais, esta chave é incrementada automaticamente cada vez que se insere um novo país e não pode assumir valores nulos. Além disso tem mais um atributo que representa o nome do país que se está a adicionar.

#### -Liga

```
create table liga(  
id_liga int not null auto_increment,  
id_pais int,  
divisao int,  
nome_liga varchar(100),  
  
primary key (id_liga),  
foreign key (id_pais) references pais(id_pais)  
);  
ALTER TABLE liga ADD pais varchar(100) AFTER id_pais;
```

Nesta tabela, como se pode observar existe um id identificativo para cada liga inserida, funcionando este como chave primária, não podendo esta tomar valores nulos. Além disso, esta tabela é caracterizada também por id\_pais (que é uma chave estrangeira referente ao país a que pertence a liga criada), a divisão da liga e o nome da liga. Por fim, acrescenta-se que adicionou-se um campo extra à tabela chamado pais, apenas para transformar, através de um procedimento, o id do país para o nome do país a que se refere esse id e apresenta-lo nesta tabela.

#### -Treinador

```
create table treinador(  
id_treinador int not null auto_increment,  
id_pais int,  
nome_treinador varchar(100),  
  
primary key (id_treinador),  
foreign key (id_pais) references pais(id_pais)  
);  
ALTER TABLE treinador ADD nacionalidade varchar(100) AFTER id_pais;  
ALTER TABLE treinador ADD clube varchar(100) AFTER nome_treinador;
```

Nesta tabela estarão presentes todas as informações dos treinadores, cada treinador é identificado por um id que é incrementado automaticamente à medida que se insere um novo treinador na tabela, este id não toma valores nulos, além disto existe um id\_pais, que neste caso servirá para identificar a nacionalidade de cada treinador, e por fim ainda existe o campo que define o nome do treinador. A chave primária é o id\_treinador e a chave estrangeira é id\_pais. Existem 2 campos adicionais (nacionalidade e clube) para transformar id\_pais e clube respectivamente em nacionalidade e clube através de procedimentos presentes na base de dados.

#### -Clube

```
create table clube(  
id_clube int not null auto_increment,  
id_liga int,  
nome_clube varchar(100),  
id_treinador int,  
  
primary key (id_clube),  
foreign key (id_liga) references liga(id_liga),  
foreign key (id_treinador) references treinador(id_treinador)  
);  
ALTER TABLE clube ADD nome_treinador varchar(100) AFTER id_treinador;  
ALTER TABLE clube ADD liga varchar(100) AFTER id_liga;  
ALTER TABLE clube ADD pais varchar(100) AFTER liga;
```

Nesta tabela são armazenadas as informações relativas aos clubes presentes na base de dados, cada clube é identificado por um id não nulo, incrementado automaticamente, além disso é sabido o id da liga a que cada clube pertence, o nome do clube, e o id do seu treinador, além disso acrescentaram-se campos (nome\_treinador, liga, pais) que servem para transformar id\_treinador no seu nome, o id\_liga no seu respectivo nome, e a liga no respectivo nome de pais a que pertence, através de procedimentos presentes na base de dados.



### -Jogador

```
create table jogador(  
id_jogador int not null auto_increment,  
id_clube int,  
id_pais int,  
idade_jogador int,  
nome_jogador varchar(100),  
rating int,  
posicao varchar(4),  
valor_de_mercado decimal (8,3),  
  
salario_jogador decimal (8,3),  
  
primary key (id_jogador),  
foreign key(id_clube) references clube(id_clube),  
foreign key(id_pais) references pais(id_pais)  
);
```

```
ALTER TABLE jogador ADD nome_clube varchar(100) AFTER id_clube;  
ALTER TABLE jogador ADD nacionalidade varchar(100) AFTER id_pais;
```

Nesta tabela são armazenadas as informações relativas aos jogadores presentes na base de dados, cada clube é identificado por um id não nulo, sendo esta a chave primária, incrementado automaticamente, além disso é sabido o id do clube a que cada jogador pertence, o id do pais, a idade do jogador, nome\_jogador, rating, posição, e valor de mercado e salário de cada jogador (em milhões de euros), além disso acrescentaram-se campos (nome\_clube, nacionalidade) que servem para transformar id\_treinador no seu nome, o id\_liga no seu respetivo nome, e a liga no respetivo nome de pais a que pertence, através de procedimentos presentes na base de dados.

### -Treinamento

```
create table treinamento(  
id_treinamento int not null auto_increment,  
id_treinador int,  
id_clube int,  
data_inicio varchar(100),  
data_fim varchar(100),  
  
primary key (id_treinamento),  
foreign key(id_treinador) references treinador(id_treinador),  
foreign key(id_clube) references clube(id_clube)  
);  
ALTER TABLE treinamento ADD clube varchar(100) AFTER id_clube;  
ALTER TABLE treinamento ADD nome_treinador varchar(100) AFTER id_treinador;
```

Nesta tabela são armazenadas as informações relativas aos empregos que treinadores tiveram ao longo do tempo, presentes na base de dados, cada linha do treinamento correspondente a um determinado passado num dado clube por parte de um treinador, é identificada por um id\_treinamento que não é nulo e é incrementado automaticamente, sendo este a chave primária, incrementada automaticamente, além disso é sabido o id do treinador, o id do clube que estava ou está a treinar, a data de quando começou a treinar o clube e a data de quando saiu, se ainda não saiu ou entrou essas datas encontram-se a NULL. Depois adicionaram-se campos extra (clube, nome\_treinador) que servem para transformar o id do clube no respetivo nome, e o id\_treinador no respetivo nome.

#### -Transferencia

```
create table transferencia(
id_transferencia int not null auto_increment,
id_jogador int,
id_clube_antigo int,
id_clube_novo int,
valor_transferencia decimal (8,3),

primary key (id_transferencia),
foreign key(id_jogador) references jogador(id_jogador),
foreign key(id_clube_antigo) references clube(id_clube),
foreign key(id_clube_novo) references clube(id_clube)
);
ALTER TABLE transferencia ADD nome_jogador varchar(100) AFTER id_jogador;
ALTER TABLE transferencia ADD clube_antigo varchar(100) AFTER id_clube_antigo;
ALTER TABLE transferencia ADD clube_novo varchar(100) AFTER id_clube_novo;
```

Nesta tabela são armazenadas as informações relativas às transferências que jogadores tiveram ao longo do tempo, presentes na base de dados, cada linha da transferências correspondente a uma determinada transferência, é identificada por um id\_transferencia, que não é nulo e é incrementado automaticamente é identificado por um id não nulo, sendo este a chave primária, incrementada automaticamente, além disso é sabido o id do jogador, o id do clube antigo, o id do novo clube e o valor da transferência. Depois adicionaram-se campos extra (nome\_jogador,clube\_antigo, clube\_novo) que servem para transformar o id\_jogador no respetivo nome, o id\_clube antigo no respetivo nome e o id\_clube\_novo no respetivo nome.

#### -Jogo

```
create table jogo(  
id_jogo int not null auto_increment,  
id_clube_local int,  
id_clube_visitante int,  
golos_local int,  
golos_visitante int,  
  
primary key (id_jogo),  
foreign key(id_clube_local) references clube(id_clube),  
foreign key(id_clube_visitante) references clube(id_clube)  
);  
ALTER TABLE jogo ADD clube_local varchar(100) AFTER id_clube_local;  
ALTER TABLE jogo ADD clube_visitante varchar(100) AFTER id_clube_visitante;
```

Nesta tabela são armazenadas as informações relativas aos jogos disputados entre as várias equipas pertencentes à mesma liga, presentes na base de dados. Cada jogo é identificado por um id\_jogo, não nulo que é incrementado automaticamente, esta é a chave primária, além disto é sabido o id\_clube\_local, o id\_clube\_visitante, os golos da equipa local e da equipa visitante, sendo as chaves estrangeiras o id\_clube\_local e id\_clube\_visitante. Depois são ainda acrescentados campos extra clube\_local para transformar o id\_clube\_local no seu nome, e o id\_clube\_visitante no seu nome.

#### -Classificação Semanal

```
create table classificacao_semanal(  
id_classificacao int not null auto_increment,  
id_liga int ,  
id_clube int,  
numero_golos int,  
pontos int,  
  
primary key (id_classificacao),  
foreign key(id_liga) references liga(id_liga),  
foreign key(id_clube) references clube(id_clube)  
);  
ALTER TABLE classificacao_semanal ADD nome_clube varchar(100) AFTER id_clube;  
ALTER TABLE classificacao_semanal ADD jogos int AFTER pontos;
```

Nesta tabela são armazenadas as informações relativas às classificações de cada clube numa dada liga, depois de alterados os seus dados na mesma, através de procedimentos e *triggers*, presentes na base de dados, após terem jogos disputados com outros clubes, cada classificação referente ao um clube é única daí ser identificada por uma chave primária id\_classificacao, além disso são sabidos o id da liga, o id do clube, o id\_clube, o número de golos, e os pontos acumulados. Depois são adicionados campos extra para transformar o id\_clube no se nome, e ainda número de jogos disputados por cada equipa.

### **Procedures, Triggers e Functions**

#### **-Nova Transferência**

```
delimiter //
create procedure nova_transferencia (in id int, novo_clube int, valt decimal (8,3) )
begin
declare id__clube int;
declare nomej varchar(100);
declare nomecla varchar(100);
declare nomecln varchar(100);
declare nc varchar(100);

select id_clube
into id__clube
from jogador
where id_jogador=id;

select nome_clube
into nc
from clube
where id_clube=novo_clube;

update jogador
set id_clube = novo_clube,
    nome_clube=nc
where id_jogador = id;

select nome_jogador
into nomej
from jogador
where id_jogador=id;

select nome_clube
into nomecla
from clube
where id_clube=id__clube;

select nome_clube
into nomecln
from clube
where id_clube=novo_clube;

insert into transferencia(id_jogador,nome_jogador,id_clube_antigo,clube_antigo,id_clube_novo,clube_novo,valor_transferencia)
values (id,nomej,id__clube,nomecla,novo_clube,nomecln,valt);

update jogador
set valor_de_mercado = valt
where id_jogador=id;

end//
delimiter ;
```

Este procedimento serve para transferir um jogador de um clube para outro, recebe como entrada, o id do jogador, o id do novo clube e o valor da transferência (em milhões de euros), e é alterado o clube de cada jogador na tabela dos jogadores, o valor de mercado que passa a ser o valor da transferencia, e ainda é atualizada a tabela de transferências.

#### -Inserir novo jogador

```
delimiter //
create procedure inserir_jogador (in idc int, idp int, idad int, nom varchar(100), rat int, pos varchar(4), vm decimal (8,3), sal decimal (8,3))
begin
declare clube varchar(100);
declare nac varchar(100);

select nome_clube
into clube
from clube
where id_clube=idc;

select nome_pais
into nac
from pais
where id_pais=idp;

insert into jogador(id_clube,nome_clube,id_pais,nacionalidade, idade_jogador,nome_jogador,rating,posicao,valor_de_mercado,salario_jogador)
values (idc,clube,idp,nac,idad,nom,rat,pos,vm,sal);

end//
delimiter ;
```

Este procedimento serve para adicionar um novo jogador à base de dados, recebe como parâmetro de entrada o id do clube, o id do país da sua nacionalidade, a idade, o nome, o rating, a posição, o valor de mercado e o salário.

#### -Inserir novo treinador

```
delimiter //
create procedure inserir_treinador (in idp int, nom varchar(100))
begin
declare nomep varchar(100);

select nome_pais
into nomep
from pais
where id_pais=idp;

insert into treinador(id_pais,nacionalidade,nome_treinador)
values (idp,nomep,nom);

end//
delimiter ;

delimiter //
```

Este procedimento serve para adicionar um novo treinador à base de dados, recebe como parâmetros de entrada o id do país da sua nacionalidade e o seu nome.

-Trigger para inserir treinador no treinamento

```
create trigger inserir_treinamento
after insert on treinador
for each row
begin
declare nt varchar(100);

select nome_treinador
into nt
from treinador
where id_treinador=NEW.id_treinador;

insert into treinamento(id_treinador,nome_treinador,id_clube,clube,data_inicio,data_fim)
values(NEW.id_treinador,nt,NULL,NULL,NULL,NULL);

end//
delimiter ;
```

Este *trigger* serve para que cada vez que se insere um novo treinador, este também é adicionado ao treinamento, para assim poder ser contratado.

-Inserir nova liga

```
delimiter //
create procedure inserir_liga (in idp int,dv int, nom varchar(100))
begin
declare np varchar(100);
select nome_pais
into np
from pais
where id_pais=idp;

insert into liga(id_pais,pais,divisao,nome_liga)
values (idp,np,dv,nom);

end//
delimiter ;
```

Este procedimento serve para adicionar uma nova liga à base de dados, recebe como parâmetros de entrada, o id do país, divisão,, e o nome da liga.

#### -Inserir novo clube

```
delimiter //
create procedure inserir_clube (in idl int, nom varchar(100), idt int)
begin
declare nt varchar(100);
declare nl varchar(100);
declare idp int;
declare np varchar(100);

select nome_liga
into nl
from liga
where id_liga=idl;

select nome_treinador
into nt
from treinador
where id_treinador=idt;

select id_pais
into idp
from liga
where id_liga=idl;

select nome_pais
into np
from pais
where id_pais=idp;

insert into clube(id_liga,liga,pais,nome_clube, id_treinador,nome_treinador)
values (idl,nl,np,nom,idt,nt);

end//
delimiter ;
```

Este procedimento serve para inserir um novo clube na base de dados, recebe como parâmetros de entrada o id da liga, o nome do clube, e o nome de treinador que em principio será NULL.

#### -Apagar Clube

```
delimiter //  
create procedure apagar_clube (in idc int)  
begin  
  
delete from clube where id_clube=idc;  
  
end//  
delimiter ;
```

Este procedimento serve para apagar um clube da base de dados, recebe como parâmetros de entrada, o id do clube.

#### -Apagar treinador

```
delimiter //  
create procedure apagar_treinador (in idt int)  
begin  
  
delete from treinador where id_treinador=idt;  
  
end//  
delimiter ;
```

Este procedimento serve para apagar um treinador da base de dados, recebe como parâmetros de entrada, o id do treinador.

#### -Apagar liga

```
delimiter //  
create procedure apagar_liga (in idl int)  
begin  
  
delete from liga where id_liga=idl;  
  
end//  
delimiter ;
```

Este procedimento serve para apagar uma liga da base de dados, recebe como parâmetros de entrada, o id da liga.



#### -Apagar País

```
delimiter //  
create procedure apagar_pais (in idp int)  
begin  
  
delete from pais where id_pais=idp;  
  
end//  
delimiter ;
```

Este procedimento serve para apagar um país da base de dados, recebe como parâmetros de entrada, o id do país.

#### -Apagar Jogador

```
delimiter //  
create procedure apagar_jogador (in idj int)  
begin  
  
delete from jogador where id_jogador=idj;  
  
end//  
delimiter ;
```

Este procedimento serve para apagar um jogador da base de dados, recebe como parâmetros de entrada, o id do jogador.

#### -Editar país

```
delimiter //  
create procedure editar_pais (in idp int, nom varchar(100))  
begin  
update pais  
set nome_pais = nom where id_pais=idp;  
  
end//  
delimiter ;
```

Este procedimento serve para editar um país da base de dados, recebe como parâmetros de entrada, o id do país e o novo nome do país .

#### -Editar Liga

```
delimiter //  
create procedure editar_liga (in idl int,idp int,dv int, nom varchar(100))  
begin  
declare p varchar(100);  
select nome_pais  
into p  
from pais  
where id_pais=idp;  
  
update liga  
set nome_liga = nom,  
    id_pais=idp,  
    pais=p,  
    divisao=dv  
where id_liga=idl;  
  
end//  
delimiter ;
```

Este procedimento serve para editar uma liga da base de dados, recebe como parâmetros de entrada, o id da liga, o novo id do pais, a nova divisao e o novo nome.

#### -Editar Treinador

```
delimiter //  
create procedure editar_treinador (in idt int,idp int, nom varchar(100))  
begin  
declare np varchar(100);  
select nome_pais  
into np  
from pais  
where id_pais=idp;  
  
update treinador  
set nome_treinador = nom,  
    id_pais=idp,  
    nacionalidade=np  
where id_treinador=idt;  
  
end//  
delimiter ;
```

Este procedimento serve para editar um treinador da base de dados, recebe como parâmetros de entrada, o id do treinador, o id do pais da sua nova nacionalidade, e o novo nome.

#### -Editar clube

```
delimiter //
create procedure editar_clube (in idc int, idl int, nom varchar(100), idt int)
begin
declare l varchar(100);
declare t varchar(100);
declare idp int;
declare np varchar(100);

select nome_liga
into l
from liga
where id_liga=idl;

select nome_treinador
into t
from treinador
where id_treinador=idt;

select id_pais
into idp
from liga
where id_liga=idl;

select nome_pais
into np
from pais
where id_pais=idp;

update clube
set nome_clube = nom,
    id_liga=idl,
    id_treinador=idt,
    liga=l,
    pais=np,
    nome_treinador=t
where id_clube=idc;

end//
delimiter ;
```

Este procedimento serve para editar um clube da base de dados, recebe como parâmetros de entrada, o id do clube, o novo id da liga, o novo nome, e o novo id de treinador, que em principio estará a NULL.

### - Editar jogador

```
delimiter //
create procedure editar_jogador (in idj int, idc int, nom varchar(100), idp int, idade int, rat int, pos varchar(4), vm decimal (8,3), sal decimal (8,3))
begin
update jogador
set nome_jogador = nom,
    id_pais=idp,
    id_clube=idc,
    idade_jogador=idade,
    rating=rat,
    posicao=pos,
    valor_de_mercado=vm,
    salario_jogador=sal
where id_jogador=idj;

end//
delimiter ;
```

Este procedimento serve para editar um jogador da base de dados, recebe como parâmetros de entrada, o id do jogador, o novo id do id do clube, o novo nome, o id do pais da nova nacionalidade, a idade, o novo rating, a nova posição, o novo valor de mercado e o novo salário.

### -Despedir Treinador

```
delimiter //
create procedure despedir_treinador (in idc int)
begin
declare id int;
declare nc varchar(100);
declare nt varchar(100);

select id_treinador
into id
from clube
where id_clube=idc;

select nome_clube
into nc
from clube
where id_clube=idc;

select nome_treinador
into nt
from treinador
where id_treinador=id;

update treinamento
set data_fim= NOW()
where id_treinador=id and id_clube=idc;

insert into treinamento(id_treinador,nome_treinador,id_clube,clube,data_inicio,data_fim)
values(id,nt,NULL,NULL,NULL,NULL);

update clube
set id_treinador=NULL,
    nome_treinador=NULL
where id_clube=idc;

end//
delimiter ;
```

Este procedimento serve para despedir um treinador de um clube, recebendo para isso como parâmetro de entrada o id do clube do qual se quer despedir. Depois disto é adicionado ao centro de emprego “treinamento” uma nova linha sobre o treinador.

#### -Contratar Treinador

```
delimiter //
create procedure contratar_treinador (in idc int, idt int)
begin

declare nc varchar(100);
declare nt varchar(100);
declare idtr int;

select nome_clube
into nc
from clube
where id_clube=idc;

select nome_treinador
into nt
from treinador
where id_treinador=idt;

update clube
set id_treinador=idt,
    nome_treinador=nt
where id_clube=idc;

update treinamento
set data_inicio= NOW(),
    nome_treinador=nt,
    clube=nc,
    id_clube=idc
where id_treinador=idt and data_fim is NULL;

end//
delimiter ;
```

Este procedimento serve para contratar um treinador que se encontre no centro de emprego “treinamento”, recebendo como parâmetros de entrada o id do treinador e o clube para o qual vai.

### -Novo jogo

```
create procedure novo_jogo (in idcl int, idcv int, gol int, gov int)
begin
  declare idliga_l int;
  declare idliga_v int;
  declare clubelocal varchar(100);
  declare clubevisi varchar(100);

  select id_liga
  into idliga_l
  from clube
  where id_clube=idcl;

  select id_liga
  into idliga_v
  from clube
  where id_clube=idcv;

  select nome_clube
  into clubelocal
  from clube
  where id_clube=idcl;

  select nome_clube
  into clubevisi
  from clube
  where id_clube=idcv;

  insert into jogo(id_clube_local, clube_local, id_clube_visitante, clube_visitante, golos_local, golos_visitante)
  values(idcl, clubelocal, idcv, clubevisi, gol, gov);

  if exists (select id_clube from classificacao_semanal where id_clube=idcl) then

    if(gol > gov) then
      update classificacao_semanal
      set pontos=pontos+3,
          jogos=jogos+1,
          numero_golos=numero_golos+gol
      where id_clube=idcl;

    if(gov > gol) then
      update classificacao_semanal
      set pontos=pontos+0,
          jogos=jogos+1,
          numero_golos=numero_golos+gol
      where id_clube=idcl;

    if(gov=gol) then
      update classificacao_semanal
      set pontos=pontos+1,
          jogos=jogos+1,
          numero_golos= numero_golos+gol
      where id_clube=idcl;
```

```

end if;
end if;
end if;

else
if(gol> gov) then
insert into classificacao_semanal(id_liga,id_clube,nome_clube,numero_golos, pontos,jogos)
values(idliga_l,idcl,clubelocal,gol,3,1);
if(gov>gol) then
insert into classificacao_semanal(id_liga,id_clube,nome_clube,numero_golos, pontos,jogos)
values(idliga_l,idcl,clubelocal,gol,0,1);
if(gov=gol) then
insert into classificacao_semanal(id_liga,id_clube,nome_clube,numero_golos, pontos,jogos)
values(idliga_l,idcl,clubelocal,gol,1,1);
end if;
end if;
end if;

end if;

if exists (select id_clube from classificacao_semanal where id_clube=idcv) then

if(gol> gov) then
update classificacao_semanal
set pontos=pontos+0,
    jogos=jogos+1,
    numero_golos=numero_golos+gov
where id_clube=idcv;

if(gov> gol) then
update classificacao_semanal
set pontos=pontos+3,
    jogos=jogos+1,
    numero_golos=numero_golos+gov
where id_clube=idcv;

if(gov=gol) then
update classificacao_semanal
set numero_golos=numero_golos+gov,
    jogos=jogos+1,
    pontos=pontos+1
where id_clube=idcv;

end if;
end if;
end if;

else
if(gol> gov) then
insert into classificacao_semanal(id_liga,id_clube,nome_clube,numero_golos, pontos,jogos)
values(idliga_v,idcv,clubevisi,gov,0,1);

if(gov> gol) then
insert into classificacao_semanal(id_liga,id_clube,nome_clube,numero_golos, pontos,jogos)
values(idliga_v,idcv,clubevisi,gov,3,1);

if(gol= gov) then
insert into classificacao_semanal(id_liga,id_clube,nome_clube,numero_golos, pontos,jogos)
values(idliga_v,idcv,clubevisi,gov,1,1);

end if;
end if;
end if;

end if;

end//
delimiter ;

```

Este procedimento serve para registrar os jogos feitos entre 2 clubes, recebendo como parâmetros de entrada o id do clube local e visitante e os golos do clube local e visitante. Além disso faz automaticamente o incremento de pontos e golos na classificação semanal de cada clube.

#### -Verificar transferencia

```
delimiter //
create trigger verifica_transferencia
after insert on transferencia
for each row
begin
declare idliga_a int;
declare idliga_n int;
declare idclube int;
declare nvalor decimal (8,3);
|
select id_liga
into idliga_a
from clube
where id_clube=NEW.id_clube_antigo;

select id_liga
into idliga_n
from clube
where id_clube=NEW.id_clube_novo;

if(idliga_a=idliga_n) then
update jogador
set id_clube = NEW.id_clube_novo
where id_jogador = NEW.id_jogador;

else
select id_clube_antigo
into idclube
from transferencia
where id_jogador=NEW.id_jogador;

update jogador
set id_clube = idclube
where id_jogador = NEW.id_jogador;

delete from transferencia where id_transferencia=NEW.id_transferencia;
end if;
```

Este trigger serve para verificar se uma transferencia foi feita entre clubes da mesma liga, senão ele apaga a transferencia e o jogador fica no mesmo clube.



### -Plantel

```
delimiter //  
create procedure plantel (in id int)  
begin  
select nome_jogador,nacionalidade,rating,posicao,valor_de_mercado,salario_jogador from jogador where id_clube=id;  
end//  
delimiter ;
```

Este procedimento serve para visualizar o plantel de cada clube, recebendo como parâmetro de entrada o id do clube.

### -Procurar jogador pelo nome

```
delimiter //  
create procedure procurar_jogador_nome (in nom varchar(100))  
begin  
select nome_jogador,rating,posicao,valor_de_mercado,salario_jogador from jogador where nome_jogador=nom;  
end//  
delimiter ;
```

Este procedimento serve para procurar um jogador pelo seu nome, recebendo como parâmetros o nome a procurar.

### -Classificação da liga

```
delimiter //  
create procedure classificacao_liga (in id int)  
begin  
select nome_clube,numero_golos,pontos,jogos from classificacao_semanal where id_liga=id order by pontos desc ;  
end//  
delimiter ;
```

Este procedimento serve para mostrar a classificação de uma liga, recebendo como parâmetros de entrada o id da liga.

### -Mostrar ligas de um pais

```
delimiter //  
create procedure ligas_pais (in id int)  
begin  
select nome_liga,divisao from liga where id_pais=id ;  
end//  
delimiter ;
```

Este procedimento serve para mostrar as ligas que um pais tem, recebendo como entrada o id do pais.

#### -Historial do treinador

```
delimiter //  
create procedure historial_treinador (in id int)  
begin  
select clube,data_inicio,data_fim from treinamento where id_treinador=id ;  
end//  
delimiter ;
```

Este procedimento serve para mostrar os clubes que um treinador já treinou, recebendo como entrada o id do treinador.

#### -Nome do clube por id

```
delimiter @  
create function nome_clube (id int)  
returns varchar(100)  
begin  
declare nome varchar(100);  
  
select nome_clube  
into nome  
from clube  
where id_clube=id;  
if ( id>0 ) then  
return (nome);  
else  
return(false);  
end if;  
  
end @  
delimiter ;
```

Esta função serve para saber qual o clube com um dado id, para isso recebe como entrada o id do clube.

### 3.Instruções de utilização

Nesta secção pode-se ver como se pode trabalhar com a base de dados utilizando o MySQL,

Para começar deve-se carregar a base de dados, com o nome base\_de\_dados.txt

```
mysql> source C:\BD\base_de_dados.txt
```

De seguida deve-se carregar a script para popular as tabelas.

```
mysql> source C:\BD\dados.txt
```

Depois basta recorrer ao comando call para chamar os procedimentos.

-Nova transferência

```
mysql> call nova_transferencia(1,2,0.400)
```

-Inserir Treinador

```
mysql> call inserir_treinador(1,"Novo treinador")
```

-Inserir Jogador

```
mysql> call inserir_jogador(1,1,19,"Novo jogador",83,"MC",0.400,0.500);
```

-Inserir País

```
mysql> call inserir_pais("NOVO PAIS");
```

-Inserir Liga

```
mysql> call inserir_liga(1,1,"NOVA LIGA");
```

-Inserir Clube

```
mysql> call inserir_clube(1,"NOVO CLUBE",1);
```

-Apagar Clube

```
mysql> call apagar_clube(1);
```

-Apagar treinador

```
mysql> call apagar_treinador(1);
```

-Apagar Liga

```
mysql> call apagar_liga(1);
```

-Apagar País

```
mysql> call apagar_pais(1);
```

-Apagar jogador

```
mysql> call apagar_jogador(1);
```

-Editar pais

```
mysql> call editar_pais(1,"PAIS EDITADO");
```

-Editar liga

```
mysql> call editar_liga(1,1,2,"liga editada");
```

-Editar treinador

```
mysql> call editar_treinador(1,1,"treinador editado");
```

-Editar Clube

```
mysql> call editar_clube(1,1,"clube editado",NULL);
```

-Editar jogador

```
call editar_jogador(1,1,"jogador editado",1,20,80,"MC",0.400,0.500);
```

-Despedir treinador

```
mysql> call despedir_treinador(1);
```

-Contratar treinador

```
mysql> call contratar_treinador(1,1);
```

-Novo jogo

```
call novo_jogo(1,2,1,0);
```

-Plantel

```
mysql> call plantel(1);
```

nome_jogador	nacionalidade	rating	posicao	valor_de_mercado	salario_jogador
jogador editado	Portugal	80	MC	0.400	0.500
Cristiano Gomes	Portugal	73	DD	0.210	0.003
Filipe Oliveira	Portugal	78	MC	1.200	0.008

3 rows in set (0.01 sec)

-Procurar jogador pelo nome

```
mysql> call procurar_jogador_nome("jogador editado");
```

nome_jogador	rating	posicao	valor_de_mercado	salario_jogador
jogador editado	80	MC	0.400	0.500

1 row in set (0.07 sec)

Query OK, 0 rows affected (0.18 sec)

-Ligas de um país

```
mysql> call ligas_pais(1);
```

nome_liga	divisao
Liga Portuguesa	1
II Liga	2

2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.11 sec)

### -Historial de um treinador

```
mysql> call historial_treinador(1);
+-----+-----+-----+
| clube      | data_inicio      | data_fim      |
+-----+-----+-----+
| clube editado | 2018-06-01 15:30:01 | 2018-06-01 15:30:32 |
| Rio Ave      | 2018-06-01 15:30:39 | NULL           |
+-----+-----+-----+
2 rows in set (0.00 sec)

Query OK, 0 rows affected (0.15 sec)
```

### -Nome de clube pelo id

```
mysql> select nome_clube(1);
+-----+
| nome_clube(1) |
+-----+
| clube editado |
+-----+
1 row in set (0.02 sec)
```

Para facilitar a forma como se chamavam algumas tabelas e também como eram apresentadas, criaram-se as seguintes *views*.

### -Ver treinadores no centro de emprego

```
mysql> source C:\BD\centro_emprego.txt
Query OK, 0 rows affected (0.09 sec)

+-----+-----+-----+-----+
| nome_treinador | clube      | data_inicio      | data_fim      |
+-----+-----+-----+-----+
| Jose Mourinho  | clube editado | 2018-06-01 15:30:01 | 2018-06-01 15:30:32 |
| Pep Guardiola  | NULL         | NULL              | NULL           |
| Massimiliano Allegri | NULL         | NULL              | NULL           |
| Jose Mourinho  | Rio Ave      | 2018-06-01 15:30:39 | NULL           |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

-Ver todas as classificações

```
mysql> source C:\BD\classificacoes.txt
Query OK, 0 rows affected (0.10 sec)

+-----+-----+-----+-----+
| id_liga | id_clube | numero_golos | pontos |
+-----+-----+-----+-----+
|      1 |      1 |          39 |      21 |
|      1 |      2 |           5 |       3 |
+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

-Ver os todos os clubes

```
mysql> source C:\BD\clubes.txt
Query OK, 0 rows affected (0.12 sec)

+-----+-----+-----+-----+
| liga      | pais      | nome_clube      | nome_treinador |
+-----+-----+-----+-----+
| Liga Portuguesa | Portugal | clube editado   | NULL            |
| Liga Portuguesa | Portugal | Rio Ave         | Jose Mourinho  |
| Liga Portuguesa | Portugal | Belenenses      | NULL            |
| Liga Portuguesa | Portugal | V.Setubal       | NULL            |
| II Liga        | Portugal | Santa Clara     | NULL            |
| II Liga        | Portugal | U.Madeira       | NULL            |
| II Liga        | Portugal | A.Viseu         | NULL            |
| La Liga        | Espanha  | Barcelona       | NULL            |
| La Liga        | Espanha  | A.Madrid        | NULL            |
| La Liga        | Espanha  | R.Madrid        | NULL            |
| La Liga        | Espanha  | Valencia        | NULL            |
| Ligue 1        | Franca   | PSG             | NULL            |
| Ligue 1        | Franca   | Monaco          | NULL            |
| Ligue 1        | Franca   | Lyon            | NULL            |
| Serie A        | Italia   | Juventus        | NULL            |
| Serie A        | Italia   | AC Milan        | NULL            |
| Serie A        | Italia   | Inter Milao     | NULL            |
| Premier League | Inglaterra | Bayern         | NULL            |
| Premier League | Inglaterra | Dortmund       | NULL            |
| Liga Alema     | Alemanha | Standard Liege  | NULL            |
| Liga Alema     | Alemanha | Anderlecht      | NULL            |
| Liga Alema     | Alemanha | Charleroi       | NULL            |
+-----+-----+-----+-----+
22 rows in set (0.00 sec)
```

mysql>

-Ver todos os jogadores

```
mysql> source C:\BD\jogadores.txt
Query OK, 0 rows affected (0.14 sec)
```

nome_jogador	nacionalidade	idade_jogador	rating	valor_de_mercado	salario_jogador	nome_clube
Cristiano Gomes	Portugal	23	73	0.210	0.003	Maritimo
jogador editado	Portugal	20	80	0.400	0.500	Maritimo
Filipe Oliveira	Portugal	24	78	1.200	0.008	Maritimo
Pedro Moreira	Portugal	29	82	1.300	0.012	Rio Ave
Nelson Monte	Portugal	22	83	2.200	0.012	Rio Ave
Sergi Roberto	Espanha	26	91	20.000	0.084	Barcelona
Sergio Busquets	Espanha	29	95	44.000	0.189	Barcelona

7 rows in set (0.01 sec)

-Ver todos os jogos

```
mysql> source C:\BD\jogos.txt
Query OK, 0 rows affected (0.12 sec)
```

clube_local	golos_local	clube_visitante	golos_visitante
clube editado	0	Rio Ave	0
clube editado	0	Rio Ave	1
clube editado	1	Rio Ave	0
clube editado	1	Rio Ave	1
clube editado	2	Rio Ave	2
clube editado	1	Rio Ave	2
clube editado	2	Belenenses	2
Rio Ave	1	clube editado	2
clube editado	1	Rio Ave	2
clube editado	1	Rio Ave	2
clube editado	1	Rio Ave	0
clube editado	11	Rio Ave	0
clube editado	11	Rio Ave	1
clube editado	1	Rio Ave	1
clube editado	11	Rio Ave	1
clube editado	1	Rio Ave	0
clube editado	1	Rio Ave	2
Rio Ave	2	clube editado	1
clube editado	2	Rio Ave	1
Rio Ave	1	clube editado	2

20 rows in set (0.01 sec)



-Ver todas as ligas

```
mysql> source C:\BD\ligas.txt
Query OK, 0 rows affected (0.11 sec)
```

pais	nome_liga	divisao
Portugal	Liga Portuguesa	1
Portugal	II Liga	2
Espanha	La Liga	1
Franca	Ligue 1	1
Italia	Serie A	1
Inglaterra	Premier League	1
Alemanha	Liga Alema	1
Belgica	Liga Belga	1

```
8 rows in set (0.01 sec)
```

-Ver todos os treinadores

```
mysql> source C:\BD\treinadores.txt
Query OK, 0 rows affected (0.08 sec)
```

nome_treinador	nacionalidade
Jose Mourinho	Portugal
Pep Guardiola	Espanha
Massimiliano Allegri	Italia

```
3 rows in set (0.00 sec)
```

-Ver todas as transferências

```
mysql> source C:\BD\transferencias.txt
ERROR 1050 (42S01): Table 'transferencias' already exists
```

nome_jogador	clube_antigo	clube_novo	valor_transferencia
Andre Teles	Maritimo	Rio Ave	3.000
Cristiano Gomes	Maritimo	Rio Ave	3.000

```
2 rows in set (0.00 sec)
```

Para uma melhor utilização da base de dados, sem ser necessária a introdução manual de *queries* SQL, criei um programa em linguagem C:

```

-----|
-----BEM-VINDO-----|
----- AO FOOTBALL ENGINE -----|
-----|
Desenvolvido por David Alves

A CONECTAR COM A BASE DE DADOS...
...
...
Conectado com Sucesso!

ESCOLHA UMA OPÇÃO DO MENU:(0 para sair)
[1]-LISTAR PAISES
[2]-LISTAR LIGAS
[3]-LISTAR TREINADORES
[4]-LISTAR CLUBES
[5]-LISTAR JOGADORES
[6]-CENTRO DE EMPREGO
[7]-LISTAR TRANSFERENCIAS
[8]-LISTAR TODOS OS JOGOS
[9]-LISTAR TODAS AS CLASSIFICACOES
[10]-NOVA TRANSFERENCIA
[11]-NOVO JOGADOR
[12]-NOVO TREINADR
[13]-NOVO PAIS
[14]-NOVO CLUBE
[15]-APAGAR CLUBE
[16]-APAGAR PAIS
[17]-APAGAR TREINADOR
[18]-APAGAR LIGA
[19]-APAGAR JOGADOR
[20]-DESPEDIR TREINADOR
[21]-CONTRATAR TREINADOR
[22]-NOVO JOGO
[23]-VER PLANTEL DE CLUBE
[24]-PROCURAR JOGADOR PELO NOME
[25]-CLASSIFICACAO DE UMA LIGA
[26]-HISTORIAL DE UM TREINADOR
```

Este programa chama na mesma os procedimentos como no mySQL. A título de exemplo segue-se o procedimento para uma nova transferência:

```

[24]-PROCURAR JOGADOR PELO NOME
[25]-CLASSIFICACAO DE UMA LIGA
[26]-HISTORIAL DE UM TREINADOR
10
id do jogador:
1
id do novo clube:
2
valor da transferencia:
3
```

```

7
| nome_jogador          | clube_antigo          | clube_novo          | valor_transferencia
-----|-----|-----|-----|
Andre Teles             Maritimo               Rio Ave              3.000
```

Caso o docente queira experimentar a utilização do programa, deve:

1.Utilizar o sistema operativo Linux(este pode ser instalado recorrendo a uma máquina virtual como por exemplo *VMware*);

2.Instalar o *mysql* como comando “`sudo apt-get install mysql-server-5.7`” e definir “ubuntu” como a password de root;

3.instalar o conector c para o mysql, com o comando “`sudo apt-get install libmysqlclient-dev`”

4.abrir um terminal e executar o mysql com comando” `mysql -u root -p`” e inserir a password “ubuntu”

5.carregar a base de dados e os dados no mysql

6.abrir outro terminal no local onde esta armazenado o ficheiro e compilar o ficheiro .c com o comando

```
gcc -o projeto projeto.c -I/usr/include/mysql/ -lmysqlclient
```

É normal aparecerem alguns *warnings* no código, apesar de estar funcional.

7.executar o programa com o comando “`./projeto`”

#### 4.Dicionário de dados

##### -País

Id\_pais→ identificador do país

Nome\_pais→ nome do país

##### -Liga

Id\_liga→identificador da liga

Id\_pais→identificador do país

Divisão→divisão da liga

pais→nome do país da liga

nome\_liga→nome da liga

##### -Treinador

Id\_treinador→identificador do treinador

Id\_pais→identificador da nacionalidade

Nome\_treinador→nome do treinador

##### -Clube

Id\_clube→identificador do clube

Id\_liga→identificador da liga

Nome\_clube→nome da liga

Id\_treinador→identificador do treinador

#### -Jogador

Id\_jogador→identificativo do jogador

Id\_clube→identificativo do clube

Id\_pais→identificativo do pais

Idade\_jogador→idade do jogador

Nome\_jogador→

Rating→rating do jogador

Posicao→posição ocupada em campo pelo  
jogador:"GR","DE","DD","DC","MC","ME","MD","PL"

Valor\_de\_mercado→valor de mercado do jogador, em milhões de euros

Salario\_jogador→salário do jogador, em milhões de euros.

#### -Treinamento

Id\_treinamento→identificativo da linha do treinamento

Id\_treinador→identificativo do treinador

Id\_clube→ identificativo do clube

Data\_inicio→data de inicio

Data\_fim→data de fim

#### -Transferencia

Id\_transferencia→identificativo da transferência

Id\_jogador→ identificativo do jogador

Id\_clube\_antigo→identificativo do clube antigo

Id\_clube\_novo→identificativo do clube novo

Valor\_de\_transferencia→valor da transferência (em milhões de euros)

#### -Jogo

Id\_jogo→identificativo pelo jogo

Id\_clube\_local→ identificativo do clube local

Id\_clube\_visitante→identificativo do clube visitante

Golos\_local→golos da equipa local

Golos\_visitante →golos da equipa visitante

-Classificação semanal

Id\_classificacao→identificativo da classificação de e um clube

Id\_liga→ identificativo da liga

Id\_clube→identificativo do clube

Numero\_golos→numero de golos

Pontos→pontos

## 5.Conclusões

Finalizada a elaboração deste projeto, posso concluir que este possivelmente será o trabalho mais completo que fiz ao longo da minha carreira académica, apesar de ter ocupado uma grande parte dos meus dias nas últimas semanas, mas posso dizer que me sinto orgulhoso do resultado final, além de que me deu um imenso gosto trabalhar neste projeto, esperando por isso ter uma boa nota neste projeto prático.