

# Proyecto Bases de Datos

Grupo 39 - Furbo

Integrantes: Alejandro Tapia  
Ricardo García  
Joaquin Uribe  
Profesor: Claudio Gutierrez  
Matias Toro

Fecha de entrega: 29 de octubre de 2021  
Santiago de Chile

# Índice de Contenidos

1. Hito 0	1
2. Hito 1	2
2.1. Modelo Entidad-Relación . . . . .	2
2.2. Modelo relacional . . . . .	4
3. Hito 2	6
4. Hito 3	9

# Índice de Figuras

1. Modelo entidad relación de la base de datos: <i>Imagen original</i> . . . . .	3
2. Datos de tabla games . . . . .	9
3. Datos de tabla leagues . . . . .	9
4. Datos de tabla player_stats . . . . .	10
5. Datos de la tabla players . . . . .	10
6. Datos de tabla shot . . . . .	10
7. Datos de tabla team_stats . . . . .	10
8. Datos de tabla teams . . . . .	11

# Índice de Códigos

1. Configuración del servidor . . . . .	6
2. Creación de tablas . . . . .	6
3. Carga de tablas . . . . .	9

# 1. Hito 0

Se utilizará la base de datos "*Football and Betting Statistics of the European Top5 Leagues*" que se encuentra en el enlace: [Kaggle](#)

Fuente de los datos:

[Understat](#) tiene la información referida a estadísticas del partido.

[Football-data](#) datos de apuestas.

La base de datos consta de 7 tablas que contiene información respecto a las mejores 5 ligas europeas, las cuales se describen a continuación:

- **Appearances:** Muestra las apariciones de los jugadores en los partidos de su liga.
- **Games:** Muestra los partidos jugados en las top 5 ligas europeas.
- **Leagues:** Contiene las top 5 ligas europeas
- **Players:** Contiene a los jugadores que pertenecen a las ligas.
- **Shots:** Contiene todos los tiros realizados en los partidos de las ligas.
- **Teams:** Contiene todos los equipos pertenecientes a las ligas.
- **Team stats:** Contiene las estadísticas de juego de los equipos.

El problema que se busca resolver con esta base de datos es poder visualizar de qué manera varía el estilo de juego en los partidos de diferentes ligas. Todo eso, tomando en cuenta factores como cantidad de tarjetas, pases, faltas, goles, etc.

Se escogió esta base de datos, porque presenta relaciones interesantes entre las tablas. Además, se presentan estadísticas relevantes para lo que se busca, por ejemplo, se detalla la posición del arco en donde hizo un gol, tarjetas por partido, faltas, entre otras, que permiten comparar el estilo de juego entre las diferentes ligas del dataset.

## 2. Hito 1

### 2.1. Modelo Entidad-Relación

En la figura 1, se muestra el modelo entidad relación desarrollado para la base de datos.

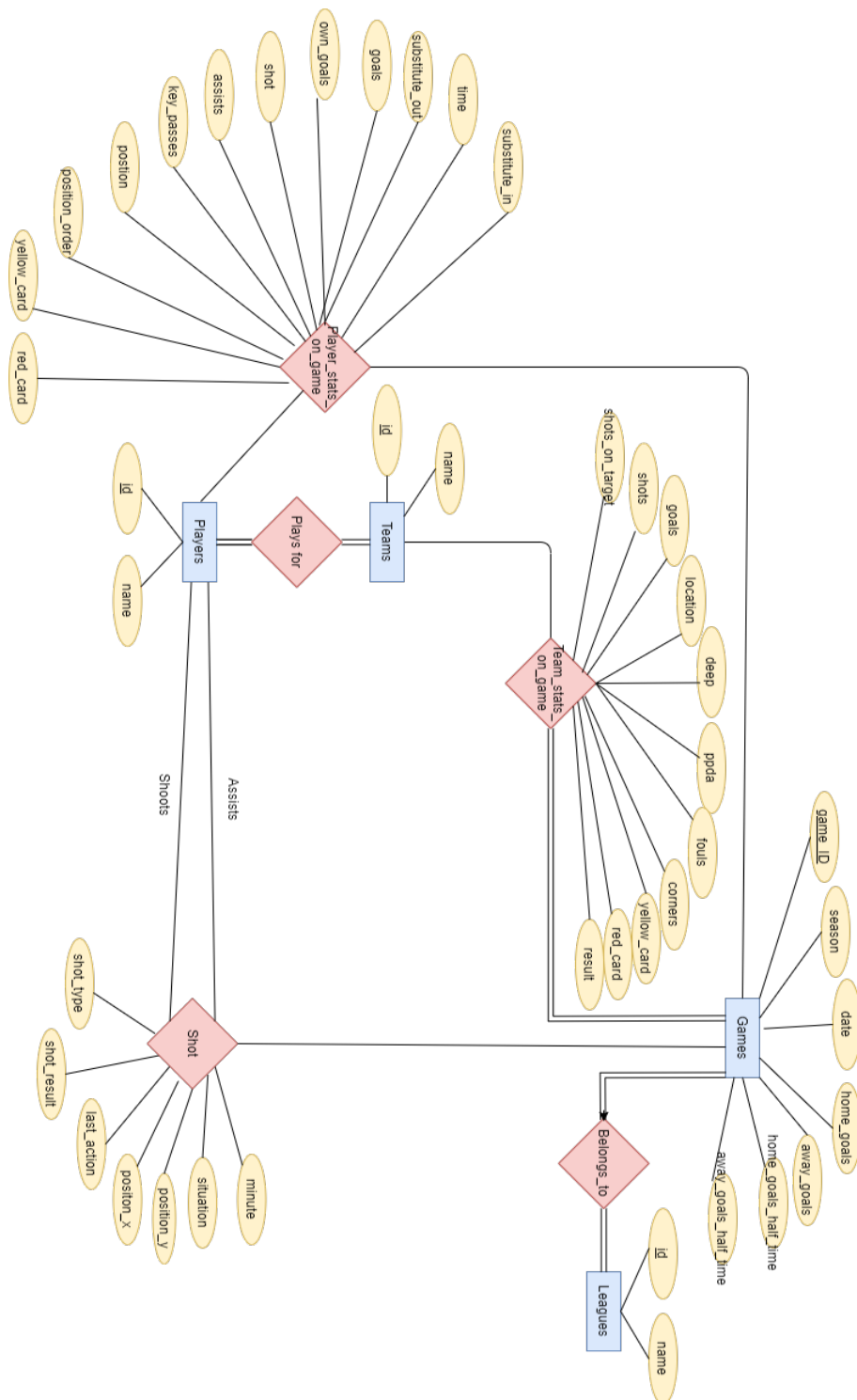


Figura 1: Modelo entidad relación de la base de datos: [Imagen original](#)

## 2.2. Modelo relacional

A continuación, se muestra el modelo relacional de la base de datos:

- Players(  
    id: INT,  
    name: STR)
- Leagues(  
    id: INT,  
    name: STR)
- Teams(  
    id: INT,  
    name: STR)
- Games(  
    gamesID: INT,  
    season: INT,  
    date: DATE,  
    home\_goals: INT,  
    away\_goals: INT,  
    home\_goals\_half\_time: INT,  
    away\_goals\_half\_time: INT,  
    Leagues.id: INT)
- Players\_stats\_on\_game(  
    P.Id: INT,  
    G.gameID: INT,  
    substitute\_in: INT,  
    time: INT,  
    substitute\_out: INT,  
    goals: INT,  
    own\_goals: INT,  
    shot: INT,  
    assists: INT,  
    key\_passes: INT,  
    position: STR,  
    position\_order: INT,  
    yellow\_card: INT,  
    red\_card: INT)

- Teams\_stats\_on\_game(  
    T.id: INT,  
    G.gameID: INT,  
    shots\_on\_target: INT,  
    shots: INT,  
    goals: INT,  
    location: STR,  
    deep: INT,  
    ppda: FLOAT,  
    fouls: INT,  
    corners: INT,  
    yellow\_card: INT,  
    red\_card: INT,  
    result: STR)
- Shot(  
    shotId: INT,  
    P.id: INT,  
    P.assistsPlayerId: INT,  
    G.gameId: INT,  
    minute: INT,  
    positionX: FLOAT,  
    positionY: FLOAT,  
    shot\_type: STR,  
    shot\_result: STR,  
    last\_action: STR,  
    situation: STR)

### 3. Hito 2

En el código 1, se configura el servidor, el usuario, se crea el esquema “furbol” y se define como principal.

Código 1: Configuración del servidor

```
1 sudo su postgres
2 createuser cc3201 -s
3 createdb -O cc3201 cc3201
4 exit
5 psql
6 CREATE SCHEMA <furbol>;
7 ALTER USER cc3201 SET search_path TO <furbol>, public;
```

Posteriormente, se crean las tablas en el esquema “furbol”. El código utilizado se muestra en el código 2.

Código 2: Creación de tablas

```
1 CREATE SCHEMA furbol;
2 ALTER USER cc3201 SET search_path TO furbol, public;
3
4
5 CREATE TABLE leagues(
6     id SMALLINT,
7     name VARCHAR(255),
8     PRIMARY KEY(id)
9 );
10
11 CREATE TABLE teams(
12     id INT,
13     name VARCHAR(255),
14     PRIMARY KEY(id)
15 );
16
17
18 CREATE TABLE players(
19     id INT,
20     name VARCHAR(255),
21     PRIMARY KEY (id)
22 );
23
24
25 CREATE TABLE games(
26     id INT,
27     leagueId INT,
28     season SMALLINT,
29     date TIMESTAMP,
30     homeGoals INT,
31     awayGoals INT,
32     homeGoalsHalfTime INT,
```



```
33     awayGoalsHalfTime INT,  
34     PRIMARY KEY (id),  
35     FOREIGN KEY (leagueId) REFERENCES leagues(id)  
36 );  
37  
38 CREATE TABLE players_stats_on_game(  
39     playerId INT,  
40     gameId INT,  
41     substituteIn INT,  
42     time INT,  
43     substituteOut INT,  
44     goals INT,  
45     ownGoals INT,  
46     shots INT,  
47     assists INT,  
48     keyPasses INT,  
49     position VARCHAR(5),  
50     positionOrder INT,  
51     yellowCard INT,  
52     redCard INT,  
53     PRIMARY KEY (gameId, playerId),  
54     FOREIGN KEY (playerId) REFERENCES players(id),  
55     FOREIGN KEY (gameId) REFERENCES games(id)  
56 );  
57  
58 CREATE TABLE team_stats_on_game(  
59     teamId INT,  
60     gameId INT,  
61     shotsOnTarget INT,  
62     shots INT,  
63     goals INT,  
64     LOCATION VARCHAR(1),  
65     deep INT,  
66     ppda REAL,  
67     fouls INT,  
68     corners INT,  
69     yellowCard INT,  
70     redCard INT,  
71     result VARCHAR(1),  
72     PRIMARY KEY (gameId, teamId),  
73     FOREIGN KEY (teamId) REFERENCES teams(id),  
74     FOREIGN KEY (gameId) REFERENCES games(id)  
75 );  
76  
77 CREATE TABLE shot(  
78     shotId INT,  
79     playerId INT,  
80     assistsId INT,  
81     gameId INT,  
82     minute INT,  
83     positionX REAL,
```

```
84     positionY REAL,  
85     shotType VARCHAR(15),  
86     shotResult VARCHAR(15),  
87     lastAction VARCHAR(15),  
88     sistuation VARCHAR(20),  
89     PRIMARY KEY (shotId, gameId, playerId, assistsId),  
90     FOREIGN KEY (gameId) REFERENCES games(id),  
91     FOREIGN KEY (playerId) REFERENCES players(id),  
92     FOREIGN KEY (assistsId) REFERENCES players(id)  
93 );  
94  
95 CREATE TABLE plays_for(  
96     playerId INT,  
97     teamId INT,  
98     PRIMARY KEY (playerId, teamId),  
99     FOREIGN KEY (playerId) REFERENCES players(id),  
100    FOREIGN KEY (teamId) REFERENCES teams(id)  
101 );
```

## 4. Hito 3

Para agregar los datos a la tabla, primero fue necesario eliminar las columnas que decidimos no utilizar de la base de datos original y rellenar los datos faltantes con place holders para evitar errores. Para esto, escribimos un pequeño script en Python con ayuda de la librería Pandas.

También para la relación Shot, fue necesario añadir una nueva columna "shotId" para que funcionen como llave primaria. Esto debido a que, al intentar usar los otros atributos como llave primaria, habían demasiadas repeticiones de tuplas.

Además, se notó que la relación Plays\_For debía ser eliminada porque con la información obtenida de la base de datos, no era posible determinar con certeza a que equipo pertenecía cada uno de los jugadores.

Luego, con la ayuda de WinSCP, movimos los nuevos archivos a la maquina virtual para así poder subirlos a la base de datos. Utilizando los siguientes comandos:

Código 3: Carga de tablas

```
1 psql -c "\copy leagues FROM 'leagues.csv' CSV HEADER"
2 psql -c "\copy teams FROM 'teams.csv' CSV HEADER"
3 psql -c "\copy players FROM 'players.csv' CSV HEADER"
4 psql -c "\copy games FROM 'games.csv' CSV HEADER"
5 psql -c "\copy players_stats_on_game FROM 'appearances.csv' CSV HEADER"
6 psql -c "\copy team_stats_on_game FROM 'teamStats.csv' CSV HEADER"
7 psql -c "\copy shot FROM 'shots.csv' CSV HEADER"
```

```
cc3201=# SELECT *
cc3201=# FROM games
cc3201=# LIMIT 5;
 id | leagueid | season | date           | homegoals | awaygoals | homegoalshalftime | awaygoalshalftime
-----+-----+-----+-----+-----+-----+-----+-----
 81 |      1 | 2015 | 2015-08-08 15:45:00 |          1 |          0 |          1 |          0
 82 |      1 | 2015 | 2015-08-08 18:00:00 |          0 |          1 |          0 |          0
 83 |      1 | 2015 | 2015-08-08 18:00:00 |          2 |          2 |          0 |          1
 84 |      1 | 2015 | 2015-08-08 18:00:00 |          4 |          2 |          3 |          0
 85 |      1 | 2015 | 2015-08-08 18:00:00 |          1 |          3 |          0 |          1
(5 rows)
```

Figura 2: Datos de tabla games

```
cc3201=# select*
cc3201=# from leagues;
 id | name
-----+-----
  1 | Premier League
  2 | Serie A
  3 | Bundesliga
  4 | La Liga
  5 | Ligue 1
(5 rows)
```

Figura 3: Datos de tabla leagues

```
cc3201=# SELECT *
cc3201=# FROM players_stats_on_game
cc3201=# LIMIT 5;
```

playerid	gameid	substitutein	time	substituteout	goals	owngoals	shots	assists	keypasses	position	positionorder	yellowcard	redcard
560	81	0	90	0	0	0	0	0	0	GK	1	0	0
557	81	222605	82	0	0	0	0	0	1	DR	2	0	0
548	81	0	90	0	0	0	0	0	0	DC	3	0	0
628	81	0	90	0	0	0	0	0	0	DC	3	0	0
1006	81	0	90	0	0	0	0	0	0	DL	4	0	0

(5 rows)

Figura 4: Datos de tabla *players\_stats*

```
cc3201=# SELECT *
cc3201=# FROM players
cc3201=# LIMIT 5;
```

id	name
560	Sergio Romero
557	Matteo Darmian
548	Daley Blind
628	Chris Smalling
1006	Luke Shaw

(5 rows)

Figura 5: Datos de la tabla *players*

```
cc3201=# SELECT *
cc3201=# FROM shot
cc3201=# LIMIT 5;
```

shotid	playerid	assistsid	gameid	minute	positionx	positiony	shottype	shotresult	lastaction	situation
0	554	-1	81	27	0.794	0.421	LeftFoot	BlockedShot	Standard	DirectFreekick
1	555	631	81	27	0.86	0.627	RightFoot	BlockedShot	Pass	SetPiece
2	554	629	81	35	0.843	0.333	LeftFoot	BlockedShot	Pass	OpenPlay
3	554	-1	81	35	0.848	0.533	LeftFoot	MissedShots	Tackle	OpenPlay
4	555	654	81	40	0.812	0.707	RightFoot	BlockedShot	BallRecovery	OpenPlay

(5 rows)

Figura 6: Datos de tabla *shot*

```
cc3201=# SELECT *
cc3201=# FROM team_stats_on_game
cc3201=# LIMIT 5;
```

teamid	gameid	shotsontarget	shots	goals	location	deep	ppda	fouls	corners	yellowcard	redcard	result
89	81	1	9	1	h	4	13.8261	12	1	2	0	W
82	81	4	9	0	a	10	8.2188	12	2	3	0	L
73	82	2	11	0	h	11	6.9	13	6	3	0	L
71	82	3	7	1	a	2	11.8462	13	3	4	0	W
72	83	5	10	2	h	5	6.65	7	8	1	0	D

(5 rows)

Figura 7: Datos de tabla *team\_stats*

```
cc3201=# SELECT *
cc3201=# FROM teams
cc3201=# LIMIT 5;
 id |          name
----+-----
 71 | Aston Villa
 72 | Everton
 74 | Southampton
 75 | Leicester
 76 | West Bromwich Albion
(5 rows)
```

Figura 8: Datos de tabla teams

En las figuras anteriores, se puede apreciar que las tablas cumplen con las restricciones del modelo relacional, por lo que se puede concluir que la subida de datos fue exitosa. Cabe destacar que, sólo se presentan 5 filas con el comando *LIMIT 5*, de forma de que se pueda visualizar de mejor forma el resultado.