

# PROYECTO DE BASE DE DATOS

SUPERCAMPEONES

BASE DE DATOS



©VT/S. 2018CTC ©2020 BNEI



SERGIO CALVO PUERTO

1º DAW

IES ALIXAR

17/05/2024

# ÍNDICE

ÍNDICE	2
1 DESCRIPCIÓN DEL PROYECTO.	3
2 MODELO ENTIDAD-RELACIÓN (DIAGRAMS)	4
3 MODELO RELACIONAL.	5
4 MODELO ENTIDAD-RELACIÓN (MySQL Workbench)	7
5 MODELO ENTIDAD-RELACIÓN (DBeaver)	8
6. CARGA MASIVA DE DATOS	9
7. CONSULTAS SQL	10
8. VISTAS.	13
9. FUNCIONES.	14
10. PROCEDIMIENTOS.	15
11. TRIGGERS.	16
12. ENLACE A GITHUB.	17
13. VALORACIÓN PERSONAL.	17

# **1 DESCRIPCIÓN DEL PROYECTO.**

Se requiere diseñar una base de datos para gestionar los diferentes torneos de la serie animada Supercampeones. En cada uno de los torneos participan diferentes equipos, cada equipo tiene un nombre, año de temporada, ciudad, país, trofeos y está dirigido por un entrenador. Cada entrenador tiene un DNI, nombre, apellido, teléfono, fecha de nacimiento, nacionalidad, el equipo al que entrena y los equipos que ha entrenado.

Cada equipo cuenta con un conjunto de jugadores, cada jugador tiene un DNI, nombre, apellido, fecha de nacimiento, teléfono, posición, dorsal, nacionalidad. También se desea almacenar de cada jugador los equipos en los que ha estado y su equipo actual. Además cada jugador cuenta con un conjunto de movimientos y de habilidades. Cada movimiento y cada habilidad tiene un nombre y una descripción.

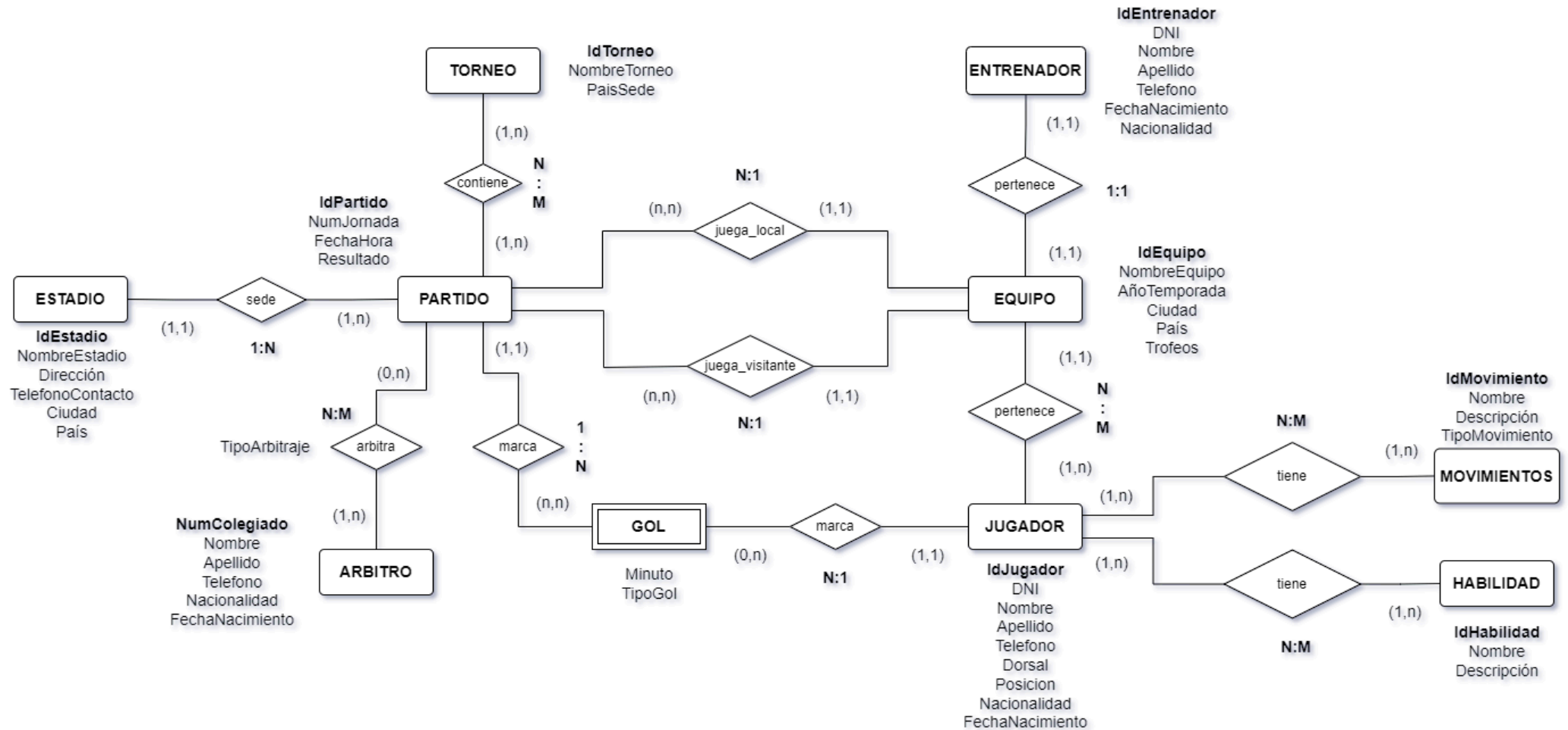
Los partidos se juegan en estadios ubicados en diversas ciudades y países. Cada partido tiene una fecha específica. También se necesita registrar los goles anotados por los jugadores en cada partido. Cada gol debe tener información sobre el jugador que lo anotó, el minuto en que se anotó y el partido en el que ocurrió. Además, en cada partido, se designan muchos árbitros que son responsables de hacer cumplir las reglas del juego. Cada árbitro tiene un número de colegiado, nombre, apellido, teléfono, nacionalidad y fecha de nacimiento.

Los equipos compiten en uno o varios partidos en cada torneo. Cada torneo tiene un id, un nombre y se lleva a cabo en un país sede. Los equipos participantes en cada torneo obtienen un resultado en función de su desempeño.

Notas:

Para los registros de la tabla torneo, los nombres de cada uno de los torneos son: Torneo Nacional de Primaria, Torneo Nacional de Secundaria, Torneo Internacional Juvenil Sub-15, Torneo Asiático Sub-19, Torneo Mundial Juvenil Sub-19, Torneo Clasificatorio Asiático Sub-23, Liga española y Torneo Olímpico Sub-23.

## 2 MODELO ENTIDAD-RELACIÓN (DIAGRAMS)



### 3 MODELO RELACIONAL.

**ESTADIO** (**IdEstadio**, NombreEstadio, Dirección, TeléfonoContacto, Ciudad, País)  
PK: **IdEstadio**

**ENTRENADOR** (**IdEntrenador**, Dni, Nombre, Apellido, Teléfono, FechaNacimiento, Nacionalidad)  
PK: **IdEntrenador**

**EQUIPO** (**IdEquipo**, NombreEquipo, AnioTemporada, Ciudad, País, Trofeos, **IdEntrenador**)  
PK: **IdEquipo**,  
FK: **IdEntrenador** REFERENCIA a **ENTRENADOR** (**IdEntrenador**)

**PARTIDO** (**IdPartido**, NumJornada, FechaHora, Resultado, **IdEstadio**, **IdEquipoLocal**, **IdEquipoVisitante**)  
PK: **IdPartido**  
FK: **IdEstadio** REFERENCIA a **ESTADIO** (**IdEstadio**)  
FK: **IdEquipoLocal** REFERENCIA a **ENTRENADOR** (**IdEquipo**)  
FK: **IdEquipoVisitante** REFERENCIA a **ENTRENADOR** (**IdEquipo**)

**TORNEO** (**IdTorneo**, NombreTorneo, FechaInicio, FechaFin, PaisSede)  
PK: **IdTorneo**

**ÁRBITRO** (**NumColegiado**, Nombre, Apellido, Nacionalidad, FechaNacimiento)  
PK: **NumColegiado**

**JUGADOR** (**IdJugador**, Dni, Nombre, Apellido, Teléfono, Dorsal, Posición, Nacionalidad, FechaNacimiento)  
PK: **IdJugador**

**MOVIMIENTO** (**IdMovimiento**, Nombre, Descripción, TipoMovimiento)  
PK: **IdMovimiento**

**HABILIDAD** (**IdHabilidad**, Nombre, Descripción)  
PK: **IdHabilidad**

**GOL** (**Minuto**, TipoGol, **IdPartido**, **IdJugador**)  
PK: **Minuto**, **IdPartido**, **IdJugador**  
FK: **IdPartido** REFERENCIA a **PARTIDO** (**IdPartido**)  
FK: **IdJugador** REFERENCIA a **JUGADOR** (**IdJugador**)

**ARBITRA** (NumColegiado, IdPartido, TipoArbitraje)

PK: NumColegiado, IdPartido

FK: NumColegiado REFERENCIA a ÁRBITRO (NumColegiado)

FK: IdPartido REFERENCIA a PARTIDO (IdPartido)

**TORNEOPARTIDO** (IdTorneo, IdPartido)

PK: IdTorneo, IdPartido

FK: IdTorneo REFERENCIA a TORNEO (IdTorneo)

FK: IdPartido REFERENCIA a PARTIDO (IdPartido)

**JUGADORMOVIMIENTO** (IdJugador, IdMovimiento)

PK: IdJugador, IdMovimiento

FK: IdJugador REFERENCIA a JUGADOR (IdJugador)

FK: IdMovimiento REFERENCIA a MOVIMIENTO (IdMovimiento)

**JUGADORHABILIDAD** (IdJugador, IdHabilidad)

PK: IdJugador, IdHabilidad

FK: IdJugador REFERENCIA a JUGADOR (IdJugador)

FK: IdMovimiento REFERENCIA a MOVIMIENTO (IdHabilidad)

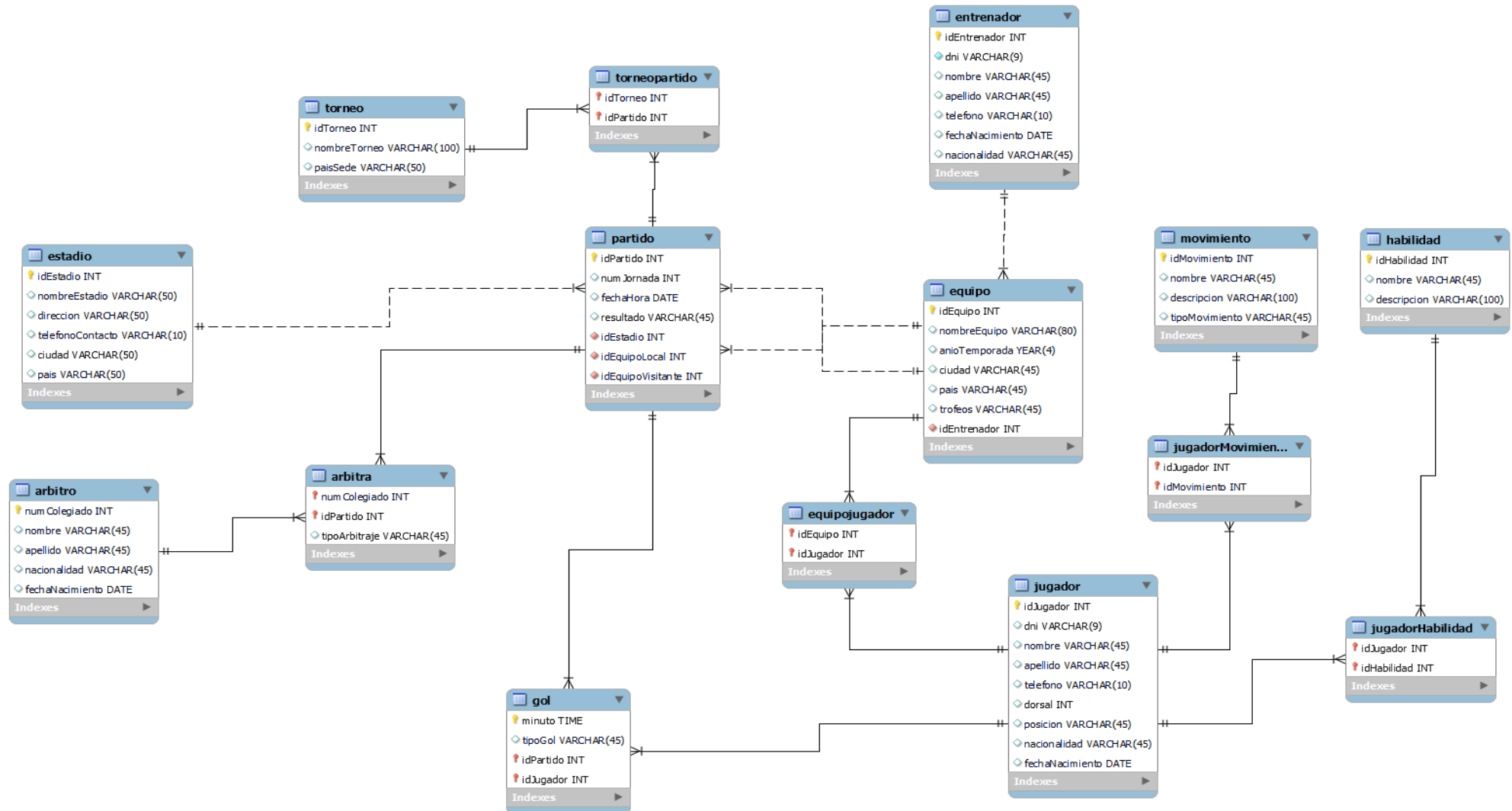
**EQUIPOJUGADOR** (IdEquipo, IdJugador)

PK: IdEquipo, IdJugador

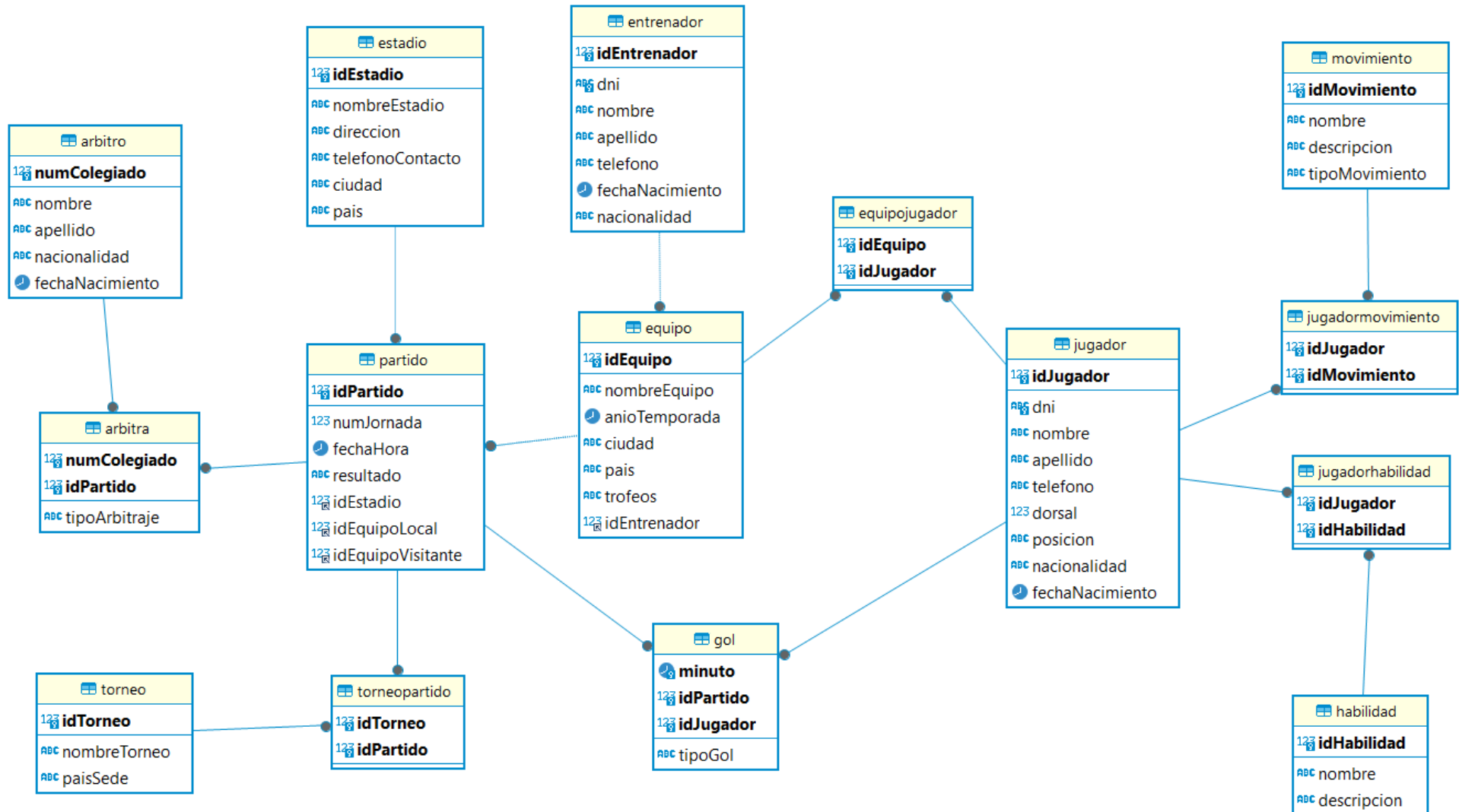
FK: IdEquipo REFERENCIA a EQUIPO (IdEquipo)

FK: IdJugador REFERENCIA a JUGADOR (IdJugador)

## 4 MODELO ENTIDAD-RELACIÓN (MySQL Workbench)



## 5 MODELO ENTIDAD-RELACIÓN (DBeaver)





## 6. CARGA MASIVA DE DATOS

La carga de datos de las tablas la he realizado con ChatGPT porque necesitaba que los datos fueran lo más parecidos a los de la serie, por ejemplo los jugadores, equipos, entrenadores, etc.



## 7. CONSULTAS SQL

Consulta 1) Devuelve un listado de cuántos jugadores que sean delanteros hay de cada país agrupando por el campo (nacionalidad), además ordenar alfabéticamente por nacionalidad.

```
-- CONSULTA 1
SELECT count(*), j.nacionalidad
FROM jugador j
WHERE j.posicion = 'Delantero'
GROUP BY j.nacionalidad
ORDER BY j.nacionalidad ASC;
```

jugador 1 ×

SELECT count(\*), j.nacionalidad FROM jugador j WHERE j.po | Enter a

	count(*)	nacionalidad
1	1	Alemán
2	5	Argentino
3	6	Brasileño
4	1	Colombiano
5	2	Español
6	1	Estadounidense
7	2	Francés
8	2	Holandés
9	2	Inglés
10	2	Italiano
11	6	Japonés
12	1	Portugués
13	1	Senegalés
14	4	Sueco
15	1	Uruguayo

Consulta 2) Devuelve un listado de los jugadores que lleven en el dorsal el número 10 y ordenar por el nombre alfabéticamente.

```
-- CONSULTA 2
SELECT j.* FROM jugador j
WHERE j.dorsal = 10 ORDER BY j.nombre ASC;
```

jugador 1 ×

SELECT j.\* FROM jugador j WHERE j.dorsal = 10 ORDER BY | Enter a SQL expression to filter results (use Ctrl+Space)

	idJugador	dni	nombre	apellido	telefono	dorsal	posicion	nacionalidad	fechaNacimiento
1	20	112233455	Carlos	Santana	555-0011	10	Delantero	Brasileño	1991-09-10
2	37	445667788	Diego	Maradona	555-4455	10	Mediocentro	Argentino	1992-03-10
3	1	123456789	Tsubasa	Ozora	555-1234	10	Mediocentro	Japonés	1990-04-10

Consulta 3) Devuelve un listado de cuántos movimientos hay de cada tipo agrupar por el campo (tipoMovimiento).

-- CONSULTA 3

```
SELECT count(*), m.tipoMovimiento
FROM movimiento m
GROUP BY m.tipoMovimiento
ORDER BY m.tipoMovimiento DESC;
```

movimiento 1 ×

SELECT count(\*), m.tipoMovimiento FROM movimiento m C | Enter a SQL

	count(*)	tipoMovimiento
1	11	Tiro
2	4	Remate
3	15	Regate
4	10	Pase
5	10	Defensa

Consulta 4) Mostrar todos los movimientos que sean de tipo Tiro o Regate.

-- CONSULTA 4

```
SELECT m.idMovimiento, m.nombre, m.descripcion, m.tipoMovimiento
FROM movimiento m
WHERE m.tipoMovimiento LIKE 'Tiro' OR m.tipoMovimiento LIKE 'Regate';
```

movimiento 1 ×

SELECT m.idMovimiento, m.nombre, m.descripcion, m.tipoMovimiento | Enter a SQL expression to filter results (use Ctrl+Space)

	idMovimiento	nombre	descripcion	tipoMovimiento
1	1	Tiro del Tigre	Potente tiro de larga distancia inspirado en el tigre.	Tiro
2	3	Regate Relámpago	Veloz regate que deja a los oponentes atrás.	Regate
3	4	Tiro con Efecto	Poderoso disparo con efecto preciso que supera las defensas y engaña a los porteros.	Tiro
4	6	Regate Sudamericano	Regate ágil que confunde a los oponentes al estilo Brasileño.	Regate
5	7	Tiro del Halcón	Disparo veloz y preciso que imita el vuelo del halcón.	Tiro
6	10	Regate del Viento	Movimiento de regate tan rápido como el viento.	Regate
7	11	Tiro del Fénix	Poderoso tiro que emula el resplandor de un fénix.	Tiro
8	13	Regate de la Pantera	Regate ágil y elusivo que imita la gracia de una pantera.	Regate
9	16	Técnica de la Niebla	Movimiento evasivo que confunde a los oponentes como una densa niebla.	Regate
10	17	Tiro del Dragón	Disparo con efecto que imita la fuerza y la majestuosidad de un dragón.	Tiro

Consulta 5) Devuelve un listado de cuántos jugadores hay en cada posición agrupando por el campo (posición).

```
-- CONSULTA 5
SELECT count(*), j.posicion
FROM jugador j
GROUP BY j.posicion
ORDER BY j.posicion DESC;
```

jugador 1 ×

SELECT count(\*), j.posicion FROM jugador j GROUP BY j.pos

	count(*)	posicion
1	11	Portero
2	40	Mediocentro
3	1	Mediocampista
4	2	Extremo
5	37	Delantero
6	26	Defensa
7	3	Centrocampista



## 8. VISTAS.

### VISTA 1: Vista del listado de todos los partidos del primer torneo.

```
-- Vista 1. Vista del listado de todos los partidos del primer torneo.
CREATE VIEW listado_Partidos_Primeria AS
SELECT p.*, t.nombreTorneo
FROM partido p
INNER JOIN torneopartido tp ON p.idPartido = tp.idPartido
INNER JOIN torneo t ON tp.idTorneo = t.idtorneo
WHERE t.nombreTorneo = 'Torneo Nacional de Primaria';

SELECT * FROM listado_Partidos_Primeria;
```

	idPartido	numJornada	fecha	duracionMinutos	resultado	idEstadio	idEquipoLocal	idEquipoVisitante	nombreTorneo
1	1	1	1982-11-30	40	7 - 6	1	2	1	Torneo Nacional de Primaria
2	2	1	1983-02-09	40	5 - 5	1	2	5	Torneo Nacional de Primaria
3	3	1	1983-02-28	40	3 - 2	1	1	5	Torneo Nacional de Primaria
4	4	1	1983-05-05	40	5 - 1	1	1	6	Torneo Nacional de Primaria
5	5	1	1983-05-13	40	3 - 1	1	4	5	Torneo Nacional de Primaria
6	6	1	1983-05-27	40	6 - 2	1	3	6	Torneo Nacional de Primaria
7	7	1	1983-08-10	40	3 - 2	1	2	4	Torneo Nacional de Primaria
8	8	1	1983-08-30	40	5 - 4	1	1	3	Torneo Nacional de Primaria
9	9	1	1984-02-02	40	4 - 2	1	1	2	Torneo Nacional de Primaria

### VISTA 2: Vista del listado de los movimientos de tipo Tiro, Pase y Remate.

```
-- Vista 2. Vista del listado de los movimientos de tipo Tiro, Pase y Remate.
CREATE VIEW listado_Movimientos_TPR AS
SELECT m.idMovimiento, m.nombre, m.descripcion, m.tipoMovimiento
FROM movimiento m
WHERE m.tipoMovimiento LIKE 'Tiro' OR m.tipoMovimiento LIKE 'Remate' OR m.tipoMovimiento LIKE 'Pase';

SELECT * FROM listado_Movimientos_TPR;
```

	idMovimiento	nombre	descripcion	tipoMovimiento
1	1	Tiro del Tigre	Potente tiro de larga distancia inspirado en el tigre.	Tiro
2	2	Pase del Águila	Preciso pase de larga distancia con efecto.	Pase
3	4	Tiro con Efecto	Poderoso disparo con efecto preciso que supera las defensas y engaña a los porteros.	Tiro
4	5	Remate del Dragón	Potente remate de cabeza con precisión milimétrica.	Remate
5	7	Tiro del Halcón	Disparo veloz y preciso que imita el vuelo del halcón.	Tiro
6	8	Pase de la Serpiente	Pase sorprendente y serpentino que elude a los defensores.	Pase
7	11	Tiro del Fénix	Poderoso tiro que emula el resplandor de un fénix.	Tiro
8	12	Pase de la Sombra	Pase rápido y preciso que se asemeja a la velocidad de una sombra.	Pase
9	15	Remate del León	Potente remate de cabeza que refleja la ferocidad de un león.	Remate
10	17	Tiro del Dragón	Disparo con efecto que imita la fuerza y la majestuosidad de un dragón.	Tiro
11	18	Pase del Viento	Pase largo y preciso que viaja con la velocidad del viento.	Pase
12	22	Pase del Mar	Preciso pase inspirado en la fluidez del mar.	Pase
13	25	Remate del Elefante	Potente remate de cabeza que refleja la fuerza de un elefante.	Remate
14	27	Tiro del Pegaso	Disparo veloz y ascendente que emula la velocidad de un pegaso.	Tiro
15	28	Pase Celestial	Pase preciso y elevado como si viniera del cielo.	Pase
16	31	Tiro del Puma	Poderoso tiro que imita la agilidad y fuerza de un puma.	Tiro
17	32	Pase del Cielo	Preciso pase que parece provenir directamente del cielo.	Pase

## 9. FUNCIONES.

**FUNCIÓN 1:** Función que recibe el código de un jugador y muestra el total de goles que ha marcado por partido. En caso de que el jugador no exista debe mostrar un mensaje en pantalla que diga “El jugador xxx no ha marcado ningún gol”.

```
DELIMITER $$
DROP FUNCTION IF EXISTS goles_jugador$$
CREATE FUNCTION goles_jugador(ideJugador int)
RETURNS VARCHAR(100)
DETERMINISTIC
BEGIN
    DECLARE nombre_jugador VARCHAR(100);
    DECLARE total_goles INT;

    SELECT CONCAT_WS(' ', j.nombre, j.apellido) AS NombreCompleto INTO nombre_jugador
    FROM jugador j
    WHERE idJugador = ideJugador;

    SELECT COUNT(g.idJugador) INTO total_goles
    FROM gol g
    WHERE g.idJugador = ideJugador;

    IF ideJugador IS NOT NULL THEN
        RETURN CONCAT('El jugador ', nombre_jugador, ' ha marcado un total de ', total_goles, ' goles');
    ELSE
        RETURN CONCAT('El jugador ', nombre_jugador, ' no ha marcado ningún gol');
    END IF;
END $$
DELIMITER ;

SELECT goles_jugador(1);
```

Resultados 1 x

SELECT goles\_jugador(1) | Enter a SQL expression to filter results (use Ctrl+Space)

	goles_jugador(1)
1	El jugador Tsubasa Ozora ha marcado un total de 5 goles

**FUNCIÓN 2:** Función para obtener el total de movimientos de un tipo específico.

```
-- FUNCION 2:Funcion para obtener el total de movimientos de un tipo especifico.
DELIMITER $$
DROP FUNCTION IF EXISTS movimientos_tpr$$
CREATE FUNCTION movimientos_tpr(typeMovements VARCHAR(100))
RETURNS VARCHAR(200)
DETERMINISTIC
BEGIN
    DECLARE total_Tipomov INT;

    SELECT COUNT(tipoMovimiento) INTO total_Tipomov
    FROM listado_Movimientos_TPR
    WHERE tipoMovimiento = typeMovements
    GROUP BY tipoMovimiento;

    IF typeMovements IS NOT NULL THEN
        RETURN CONCAT('El total de registros del tipo de movimiento ', typeMovements, ' es de ', total_Tipomov);
    END IF;
END $$
DELIMITER ;

SELECT movimientos_tpr('Tiro');
```

tados 1 x

movimientos\_tpr('Tiro') | Enter a SQL expression to filter results (use Ctrl+Space)

	movimientos_tpr('Tiro')
1	El total de registros del tipo de movimiento Tiro es de 11

## 10. PROCEDIMIENTOS.

**PROCEDIMIENTO 1:** Procedimiento que reciba los datos de un país y muestre los equipos de dicho país.

```
-- Procedimiento 1: Procedimiento que reciba los datos de un país y muestre los equipos de dicho país.
DELIMITER $$
DROP PROCEDURE IF EXISTS mostrar_equipos$$
CREATE PROCEDURE mostrar_equipos(p_equipo varchar(50))
BEGIN
    SELECT * FROM equipo e
    WHERE e.pais = p_equipo;
END $$
DELIMITER ;

CALL mostrar_equipos('Japón');
```

equipo 1 x

CALL mostrar\_equipos('Japón') | Enter a SQL expression to filter results (use Ctrl+Space)

	123 idEquipo	ABC nombreEquipo	anioTemporada	ABC ciudad	ABC pais	ABC trofeos	123 idEntrenador
1	1	Nankatsu FC	1983	Nankatsu	Japón	2	1
2	2	Meiwa ES	1983	Saitama	Japón	0	2
3	3	Musashi ES	1983	Tokio	Japón	0	5
4	4	Furano ES	1983	Hokkaido	Japón	0	14
5	5	Hanawa ES	1983	Akita	Japón	0	16
6	6	Naniwa ES	1983	Osaka	Japón	0	11

**PROCEDIMIENTO 2 (USANDO FUNCIÓN):** Procedimiento que recibe el código de jugador y usando una de las funciones anteriores, muestre el total de goles y los datos del jugador.

```
-- Procedimiento 2: Procedimiento que recibe el código de jugador y usando una de las funciones anteriores,
-- muestre el total de goles y los datos del jugador.
DELIMITER $$
DROP PROCEDURE IF EXISTS total_goles_juadores$$
CREATE PROCEDURE total_goles_juadores(j_goles int)
BEGIN
    SELECT goles_jugador(j_goles), j.*
    FROM jugador j INNER JOIN gol g
    ON j.idJugador = g.idJugador
    WHERE j.idJugador = j_goles;
END $$
DELIMITER ;

CALL total_goles_juadores(2);
```

jugador 1 x Estadísticas 1

CALL total\_goles\_juadores(2) | Enter a SQL expression to filter results (use Ctrl+Space)

	ABC goles_jugador(j_goles)	123 idJugador	ABC dni	ABC nombre	ABC apellido	ABC telefono	123 dorsal	ABC posicion	ABC nacionalidad	fechaNacimiento
1	El jugador Kojiro Hyuga ha marcado un total de 5 goles	2	987654321	Kojiro	Hyuga	555-5678	9	Delantero	Japonés	1988-07-2
2	El jugador Kojiro Hyuga ha marcado un total de 5 goles	2	987654321	Kojiro	Hyuga	555-5678	9	Delantero	Japonés	1988-07-2
3	El jugador Kojiro Hyuga ha marcado un total de 5 goles	2	987654321	Kojiro	Hyuga	555-5678	9	Delantero	Japonés	1988-07-2
4	El jugador Kojiro Hyuga ha marcado un total de 5 goles	2	987654321	Kojiro	Hyuga	555-5678	9	Delantero	Japonés	1988-07-2
5	El jugador Kojiro Hyuga ha marcado un total de 5 goles	2	987654321	Kojiro	Hyuga	555-5678	9	Delantero	Japonés	1988-07-2

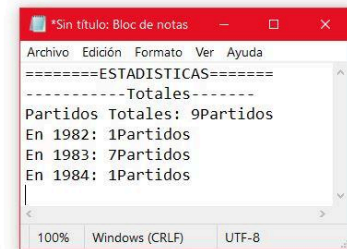
**PROCEDIMIENTO 3 (USANDO CURSOR):** Procedimiento que muestra cuántos partidos se han jugado en cada año.



```
delimiter $$
DROP PROCEDURE IF EXISTS mostrar_estadisticas$$
CREATE PROCEDURE mostrar_estadisticas()
BEGIN
    DECLARE salida varchar(10000) DEFAULT '=====ESTADISTICAS=====\\n-----Totales-----\\nPartidos Totales: ';
    DECLARE total int;
    DECLARE done bool DEFAULT FALSE;
    DECLARE anio integer;
    DECLARE n integer DEFAULT 1;
    DECLARE dia varchar(20) DEFAULT '';
    -- CURSOR DE AÑOS
    DECLARE c1 CURSOR FOR
        SELECT year(p.fechaHora), COUNT(p.numJornada)
        FROM partido p
        GROUP BY year(p.fechaHora);
    -- para salir del bucle del cursor
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    SELECT COUNT(p.numJornada) INTO total
    FROM partido p;
    SET salida = concat(salida,total,'Partidos\\n');
    -- recorremos el cursor de años
    OPEN c1;
    WHILE (NOT done) do
        FETCH c1 INTO anio, total;
        IF (NOT done) THEN
            SET salida = concat(salida,'En ',anio,': ',total,'Partidos\\n');
        END IF;
    END WHILE;
    CLOSE c1;
    SET n=1;
    SELECT salida;
END $$

delimiter ;

CALL mostrar_estadisticas();
```



## 11. TRIGGERS.

TRIGGER 1: Trigger para insertar nuevo equipo.

```
CALL mostrar_estadisticas();

-- Trigger 1: Trigger para insertar nuevo equipo.
DELIMITER $$
CREATE TRIGGER trigger_insertar_equipo
BEFORE UPDATE ON equipo
FOR EACH ROW
BEGIN
    IF NEW.idEquipo <> OLD.idEquipo THEN
        INSERT INTO equipo (idEquipo, nombreEquipo, anioTemporada, ciudad, pais, trofeos, idEntrenador)
        VALUES (NEW.idEquipo, 'Nankatsu MS', 1986, 'Nankatsu', 'Japón', 5, 5);
    END IF;
END $$
DELIMITER ;

SELECT e.* FROM equipo e ;
```

equipo 1 X

SELECT e.\* FROM equipo e

	idEquipo	nombreEquipo	anioTemporada	ciudad	pais	trofeos	idEntrenador
1	1	Nankatsu FC	1983	Nankatsu	Japón	2	1
2	2	Meiwa ES	1983	Saitama	Japón	0	2
3	3	Musashi ES	1983	Tokio	Japón	0	5
4	4	Furano ES	1983	Hokkaido	Japón	0	14
5	5	Hanawa ES	1983	Akita	Japón	0	16
6	6	Naniwa ES	1983	Osaka	Japón	0	11



TRIGGER 2: Trigger para actualizar a los entrenadores de los equipos.

```
-- TRIGGER 2: Trigger para actualizar los entrenadores de los equipos.
DELIMITER $$
CREATE TRIGGER trigger_update_entrenador
AFTER UPDATE ON equipo
FOR EACH ROW
BEGIN
    UPDATE equipo e
    SET e.idEntrenador = NEW.idEntrenador;
END$$
DELIMITER ;

UPDATE equipo SET idEntrenador = 6 WHERE idEquipo = 5;
UPDATE equipo SET idEntrenador = 7 WHERE idEquipo = 4;
UPDATE equipo SET idEntrenador = 5 WHERE idEquipo = 3;

SELECT * FROM equipo e;
```

## 12. ENLACE A GITHUB.

[GitHub Supercampeones BD](#)

## 13. VALORACIÓN PERSONAL.

Este proyecto al principio me resultó un poco complicado encontrar una temática sobre la que hacerlo, pero me decidí por la serie de Supercampeones (Oliver y Benji). Respecto a la realización del mismo, me ha ayudado bastante a mejorar diferentes aspectos del proyecto como el momento de diseñar el modelo ER, el manejo del Workbench para obtener script de creación de esta base de datos, la inserción de datos, consultas, vistas, funciones, procedimientos y triggers. Aunque de los aspectos mencionados todavía me quedan algunos de ellos por pulir, pero a pesar de la dificultad del proyecto me ha resultado muy interesante hacer este proyecto tan complejo.



**ESPERO QUE OS GUSTE!!!**