

Materia: Introducción a Bases de Datos(COM112)

Profesora: Sofia Ortiz Valenzuela

Fecha de entrega: 16 de noviembre 2022

Ciclo: 1228

Nombre del proyecto: TrackEmUP

Miembros del Equipo		
ID	Nombre	Carrera
0241823	Enrique Ulises Báez Gómez Tagle	LIDCI
0239907	Guillermo Alejandro Hernández Sosa	LIDCI
0212508	Carlos Isunza Frank	LIDCI
0212570	Santiago Valdés Uriarte	LIDCI

Rúbricas							
ID	1-identify		4-impact	5-teams			
	IP	ASA	II	ER	TC	ME	CTD

Contexto y Planteamiento problema

Hoy en día, sabemos que el fútbol es el deporte más famoso de todo el planeta. Es el deporte que genera más dinero que cualquier otro. Tan solo para que nos demos una idea de que tanta gente lo ve, en la final de la UEFA Champions League, donde se enfrentaron el Bayern y el PSG, hubo una audiencia de más de 350 millones de personas.

Ahora, imaginemos que para poder tener todo bajo control y bien ordenado, tienes que crear una base de datos con muchísima información, nada más pensemos que en un torneo como la Champions League, existe una gran cantidad de jugadores, equipos, entrenadores, árbitros, goles, tarjetas, penales, etc. Manejar tanta información es muy complicado y muchas veces abrumador, sobre todo si no sabes cómo, y en muchas ocasiones, en estos torneos se pueden llegar a cometer errores.

Incluso, por experiencia propia dentro del torneo interno de la UP, nos hemos dado cuenta de que es necesario llevar un control de este tipo, para que tanto como los organizadores, como los participantes (jugadores, equipos o entrenadores) tengan acceso a datos y estadísticas relevantes, de una manera ordenada y limpia.

Otra de las propuestas sería poder compartir la información y estadísticas recopiladas con algunas de las diferentes casas de apuestas que existen actualmente para generar momios lo más acertado posibles.

Propuesta de solución

Nuestro proyecto presenta una base de datos con 10 tablas con diferentes campos en donde podemos generar diversas consultas para obtener información específica, ya sea de un jugador, equipo, liga, evento o una tabla en general.

Como propuesta para la comercialización en un futuro, nos gustaría tener cuatro tipos de acceso a nuestra aplicación:

- a) Super Usuario: tiene acceso a edición y consulta de toda la información de la base de datos completa.
- b) Organizador: tiene acceso a toda la información y estadísticas de una liga en específico.
- c) Entrenador: tiene acceso a la información y estadísticas de su equipo.
- d) Casa De Apuestas: acceso a consulta y estadísticas de la información de toda la base de datos.

Nosotros usamos las siguientes tablas de información:

- Tabla de País
- Tabla de Árbitro
- Tabla de Liga
- Tabla de Estadio
- Tabla de Entrenador
- Tabla de Equipo
- Tabla de Partido (con marcadores)
- Tabla de Jugador (datos personales)
- Tabla de Eventos (tipos de eventos importantes y su código)
- Tabla de Eventos de partido (relación de los eventos con un partido y jugador en específico)

Contando con esto, podríamos tener una buena base de datos para poder ver la información y desplegarla al público, sin temor a cometer algún error, pues todo estaría bien organizado y listo para servir a los intereses del particular. De igual manera, tendríamos acceso a una gran cantidad de datos estadísticos y diversas maneras de filtrarlos.

Valor de la solución

Nosotros creemos que este es un gran proyecto que en realidad puede llegar a ser usado en diferentes tipos de torneos, desde nivel universitario (adaptación), nacional e incluso hasta internacional, para así poder expresar al máximo toda la información relacionada con este deporte tan importante a nivel mundial.

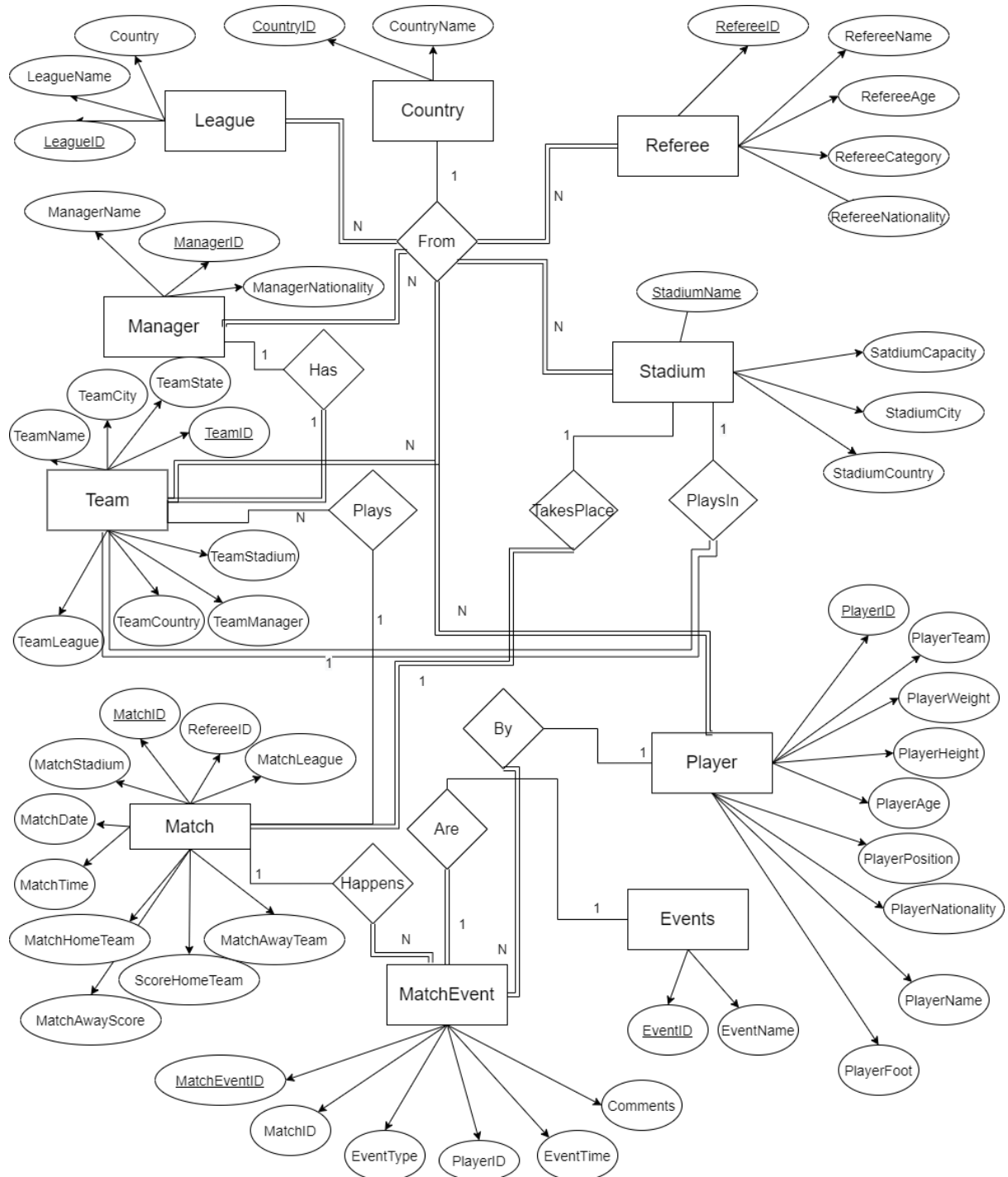
Algunas de las ventajas que nuestra solución aporta a esta problemática son:

- Información completa sobre los partidos.
- Control de los jugadores registrados.
- Estadísticas detalladas para cada jugador (goles, faltas, tarjetas...)
- Marcadores y datos generales de los partidos que ha tenido cada equipo.
- Registro sobre los partidos que ha presidido un árbitro.
- Acceso a información detallada del *roster* de un equipo.

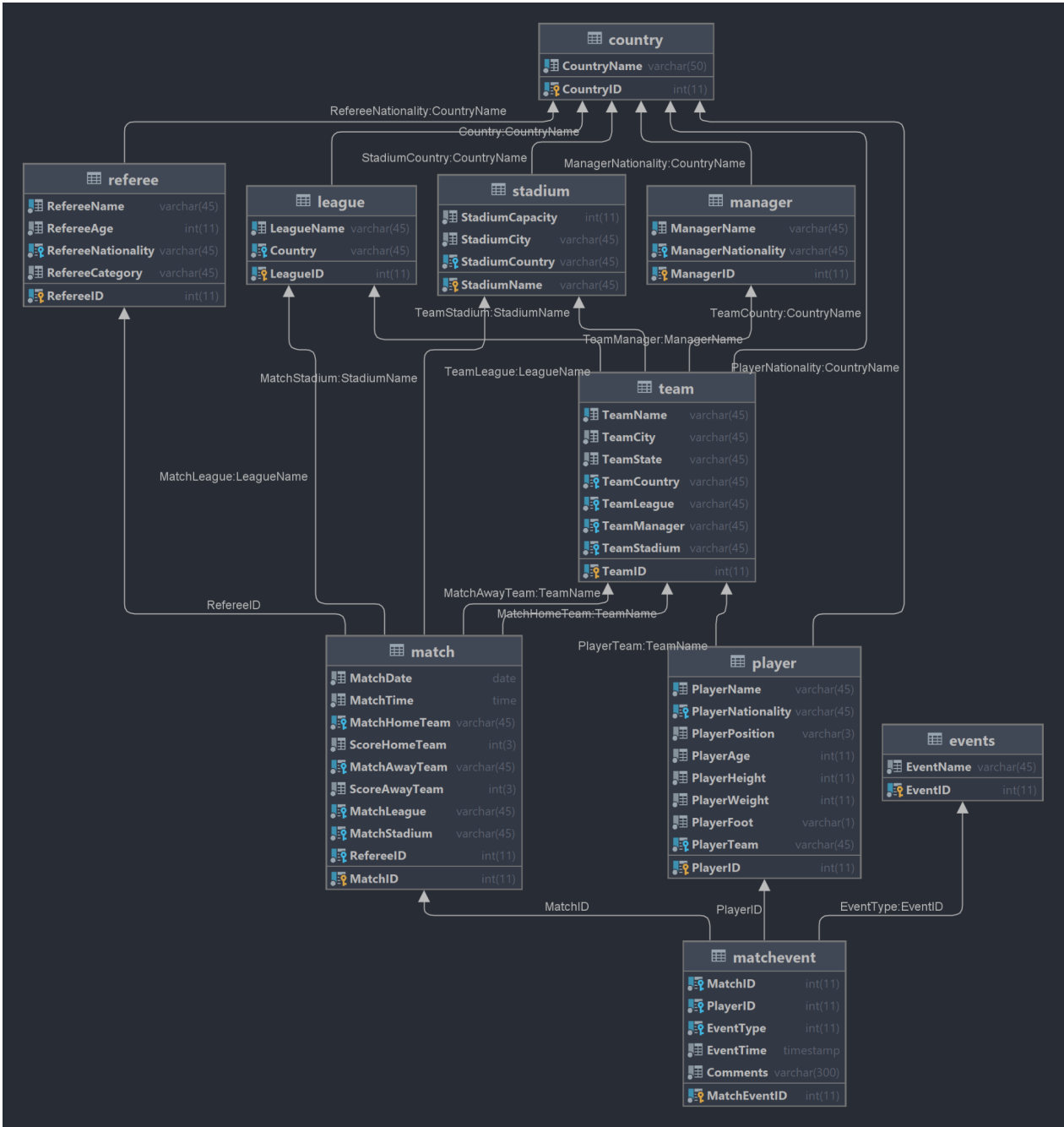
A grandes rasgos, nuestro sistema ayuda tanto a los organizadores como a los participantes de los torneos a tener un orden y una estadística confiable. De esta manera, el deporte que tanto amamos, podrá ser cada vez más limpio y más transparente, ya que todo quedará registrado en la base de datos y se podrá utilizar con diferentes objetivos finales, como los expuestos anteriormente.

Implementación

Diagrama E-R



Esquema relacional (Esquema físico)



Normalización

1FN => Cada campo tiene un valor único y no existen grupos repetidos

2FN => Los campos no clave dependen por completo de la llave primaria, sólo aplica para tablas con llaves compuestas.

3FN => Los campos no clave son accesibles únicamente por los campos clave.

Tabla	Dependencias Funcionales	1FN	2FN	3FN
matchevent	MatchEventID => PlayerID, MatchID, EventType, Comments, EventTime.	Ok	Ok	Ok
match	MatchID => RefereeID, MatchStadium, MatchLeague, MatchAwayTeam, MatchHomeTeamMatchDate, MatchTime, ScoreHomeTeam, ScoreAwayTeam.	Ok	Ok	Ok
player	PlayerID => PlayerTeam, PlayerNationality, PlayerPosition, PlayerFoot, PlayerAge, PlayerHeight, PlayerWeight, PlayerName.	Ok	Ok	Ok
events	EventID => EventName.	Ok	Ok	Ok
team	TeamID => TeamManager, TeamStadium, TeamLeague, TeamCountry, TeamCity, TeamState, TeamName.	Ok	Ok	Ok
referee	RefereeID => RefereeNationality, RefereeAge, RefereeCategory, RefereeName.	Ok	Ok	Ok
league	LueagueID => LeagueName, Country.	Ok	Ok	Ok
stadium	StadiumName => StadiumCountry, StadiumCity, StadiumCapacity.	Ok	Ok	Ok
manager	ManagerID => ManagerName, ManagerNationality.	Ok	Ok	Ok
country	CountryID => CountryName.	Ok	Ok	Ok

Creación DB

Scripts de creación de tablas

```
DROP DATABASE IF EXISTS TrackEmUP;

# CREATE DATABASE
CREATE DATABASE IF NOT EXISTS TrackEmUP;

# Select DATABASE
USE TrackEmUP;

# CREATE TABLES
CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Country`
(
    `CountryID` INT NOT NULL AUTO_INCREMENT,
    `CountryName` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`CountryID`),
    UNIQUE INDEX `CountryName_UNIQUE` (`CountryName` ASC)
) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`League`
(
    `LeagueID` INT NOT NULL AUTO_INCREMENT,
    `LeagueName` VARCHAR(45) NOT NULL,
    `Country` VARCHAR(45) NOT NULL,
    PRIMARY KEY (`LeagueID`),
    FOREIGN KEY (`Country`) REFERENCES `TrackEmUP`.`Country`
    (`CountryName`),
    UNIQUE KEY `LeagueName_UNIQUE` (`LeagueName`)
) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Manager`
(
    `ManagerID` INT NOT NULL AUTO_INCREMENT,
    `ManagerName` VARCHAR(45) NOT NULL,
    `ManagerNationality` VARCHAR(45) NOT NULL,
    PRIMARY KEY (`ManagerID`),
    FOREIGN KEY (`ManagerNationality`) REFERENCES `TrackEmUP`.`Country`
    (`CountryName`),
    UNIQUE KEY `ManagerName_UNIQUE` (`ManagerName`)
) ENGINE = InnoDB;
```



```

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Stadium`
(
    `StadiumName`      VARCHAR(45) NOT NULL,
    `StadiumCapacity`  INT          NOT NULL,
    `StadiumCity`      VARCHAR(45) NOT NULL,
    `StadiumCountry`   VARCHAR(45) NOT NULL,
    PRIMARY KEY (`StadiumName`),
    FOREIGN KEY (`StadiumCountry`) REFERENCES `TrackEmUP`.`Country`
(`CountryName`),
    UNIQUE KEY `StadiumName_UNIQUE` (`StadiumName`)
) ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Team`
(
    `TeamID`           INT          NOT NULL AUTO_INCREMENT,
    `TeamName`         VARCHAR(45) NOT NULL,
    `TeamCity`         VARCHAR(45) NOT NULL,
    `TeamState`        VARCHAR(45) NOT NULL,
    `TeamCountry`      VARCHAR(45) NOT NULL,
    `TeamLeague`       VARCHAR(45) NOT NULL,
    `TeamManager`      VARCHAR(45) NOT NULL,
    `TeamStadium`      VARCHAR(45) NOT NULL,
    PRIMARY KEY (`TeamID`),
    FOREIGN KEY (`TeamLeague`) REFERENCES `TrackEmUP`.`League`
(`LeagueName`),
    FOREIGN KEY (`TeamManager`) REFERENCES `TrackEmUP`.`Manager`
(`ManagerName`),
    FOREIGN KEY (`TeamCountry`) REFERENCES `TrackEmUP`.`Country`
(`CountryName`),

    FOREIGN KEY (`TeamStadium`) REFERENCES `TrackEmUP`.`Stadium`
(`StadiumName`),
    UNIQUE KEY `TeamName_UNIQUE` (`TeamName`)
) ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Player`
(
    `PlayerID`         INT          NOT NULL AUTO_INCREMENT,
    `PlayerName`       VARCHAR(45) NOT NULL,
    `PlayerNationality` VARCHAR(45) NOT NULL,
    `PlayerPosition`   VARCHAR(3)  NOT NULL,
    `PlayerAge`        INT          NOT NULL,
    `PlayerHeight`     INT          NOT NULL,
    `PlayerWeight`     INT          NOT NULL,
    `PlayerFoot`       VARCHAR(1)  NOT NULL,
    `PlayerTeam`       VARCHAR(45) NOT NULL,
    PRIMARY KEY (`PlayerID`),
    FOREIGN KEY (`PlayerNationality`) REFERENCES `TrackEmUP`.`Country`
(`CountryName`),

```

```

        FOREIGN KEY (`PlayerTeam`) REFERENCES `TrackEmUP`.`Team`
        (`TeamName`),
        UNIQUE KEY `PlayerName_UNIQUE` (`PlayerName`)
    ) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Referee`
(
    `RefereeID`          INT          NOT NULL AUTO_INCREMENT,
    `RefereeName`        VARCHAR(45) NOT NULL,
    `RefereeAge`         INT          NOT NULL,
    `RefereeNationality` VARCHAR(45) NOT NULL,
    `RefereeCategory`    VARCHAR(45) NOT NULL,
    PRIMARY KEY (`RefereeID`),
    FOREIGN KEY (`RefereeNationality`) REFERENCES `TrackEmUP`.`Country`
    (`CountryName`),
    UNIQUE KEY `RefereeName_UNIQUE` (`RefereeName`)
) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Match`
(
    `MatchID`          INT          NOT NULL AUTO_INCREMENT,
    `MatchDate`        DATE         NOT NULL,
    `MatchTime`        TIME         NOT NULL,
    `MatchHomeTeam`    VARCHAR(45) NOT NULL,
    `ScoreHomeTeam`    INT(3)       NOT NULL,
    `MatchAwayTeam`    VARCHAR(45) NOT NULL,
    `ScoreAwayTeam`    INT(3)       NOT NULL,
    `MatchLeague`      VARCHAR(45) NOT NULL,
    `MatchStadium`     VARCHAR(45) NOT NULL,
    `RefereeID`        INT          NOT NULL,
    PRIMARY KEY (`MatchID`),
    FOREIGN KEY (`MatchHomeTeam`) REFERENCES `TrackEmUP`.`Team`
    (`TeamName`),
    FOREIGN KEY (`MatchAwayTeam`) REFERENCES `TrackEmUP`.`Team`
    (`TeamName`),
    FOREIGN KEY (`MatchLeague`) REFERENCES `TrackEmUP`.`League`
    (`LeagueName`),
    FOREIGN KEY (`MatchStadium`) REFERENCES `TrackEmUP`.`Stadium`
    (`StadiumName`),
    FOREIGN KEY (`RefereeID`) REFERENCES `TrackEmUP`.`Referee`
    (`RefereeID`),
    UNIQUE KEY `MatchID_UNIQUE` (`MatchID`)
) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`Events`
(
    `EventID`          INT          NOT NULL AUTO_INCREMENT,
    `EventName`        VARCHAR(45) NOT NULL,
    PRIMARY KEY (`EventID`),
    UNIQUE KEY `EventID_UNIQUE` (`EventID`)
) ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `TrackEmUP`.`MatchEvent`
(
    `MatchEventID` INT NOT NULL AUTO_INCREMENT,
    `MatchID` INT NOT NULL,
    `PlayerID` INT NOT NULL,
    `EventType` INT NOT NULL,
    `EventTime` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    `Comments` VARCHAR(300) NOT NULL DEFAULT 'No comments',
    PRIMARY KEY (`MatchEventID`),
    FOREIGN KEY (`MatchID`) REFERENCES `TrackEmUP`.`Match` (`MatchID`),
    FOREIGN KEY (`PlayerID`) REFERENCES `TrackEmUP`.`Player`
(`PlayerID`),
    FOREIGN KEY (`EventType`) REFERENCES `TrackEmUP`.`Events`
(`EventID`),
    UNIQUE KEY `MatchEventID_UNIQUE` (`MatchEventID`)
) ENGINE = InnoDB;

```

Scripts para insertar registros en DB

```

# Select Database
USE TrackEmUP;

# INSERT DATA INTO TABLES

# SET AUTO_INCREMENT
ALTER TABLE `TrackEmUP`.`Country`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`Country` (CountryName)
VALUES ('Spain'),
    ('France'),
    ('Germany'),
    ('Italy'),
    ('United Kingdom');

# CREATE A FUNCTION TO REGISTER MORE COUNTRIES
DROP FUNCTION IF EXISTS TrackEmUP.`RegisterCountry`;

CREATE FUNCTION TrackEmUP.`RegisterCountry` (vCountry VARCHAR(250))
    RETURNS VARCHAR(250)
BEGIN
    INSERT INTO TrackEmUP.Country (CountryName) VALUES (vCountry);
    RETURN CONCAT('Successfully Registered: ', vCountry);
END;

```

```
SELECT TrackEmUP.RegisterCountry('Argentina') as Registered_Country;
SELECT TrackEmUP.RegisterCountry('Brazil') as Registered_Country;
SELECT TrackEmUP.RegisterCountry('Portugal') as Registered_Country;
SELECT TrackEmUP.RegisterCountry('Croatia') as Registered_Country;
```

```
ALTER TABLE `TrackEmUP`.`League`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`League` (LeagueName, Country)
VALUES ('La Liga', 'Spain'),
       ('Ligue 1', 'France'),
       ('Bundesliga', 'Germany'),
       ('Serie A', 'Italy'),
       ('Premier League', 'United Kingdom');
```

```
ALTER TABLE `TrackEmUP`.`Manager`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`Manager` (ManagerName, ManagerNationality)
VALUES ('Pep Guardiola', 'Spain'),
       ('Zinedine Zidane', 'France'),
       ('Jurgen Klopp', 'Germany'),
       ('Carlo Ancelotti', 'Italy'),
       ('Alex Ferguson', 'United Kingdom'),
       ('José Mourinho', 'Portugal'),
       ('Lionel Scaloni', 'Argentina'),
       ('Tite', 'Brazil');
```

```
ALTER TABLE `TrackEmUP`.`Stadium`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`Stadium` (StadiumName, StadiumCapacity,
StadiumCity, StadiumCountry)
VALUES ('Camp Nou', 99354, 'Barcelona', 'Spain'),
       ('Parc des Princes', 47938, 'Paris', 'France'),
       ('Allianz Arena', 75000, 'Munich', 'Germany'),
       ('San Siro', 80000, 'Milan', 'Italy'),
       ('Old Trafford', 76212, 'Manchester', 'United Kingdom'),
       ('Etihad Stadium', 53400, 'Manchester', 'United Kingdom');
```

```
ALTER TABLE `TrackEmUP`.`Team`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`Team` (TeamName, TeamCity, TeamState,
TeamCountry, TeamLeague, TeamManager, TeamStadium)
VALUES ('Real Madrid', 'Madrid', 'Madrid', 'Spain', 'La Liga', 'Pep
Guardiola', 'Camp Nou'),
       ('Paris Saint-Germain', 'Paris', 'Ile-de-France', 'France',
'Ligue 1', 'Zinedine Zidane', 'Parc des Princes'),
       ('Juventus', 'Turin', 'Piemonte', 'Italy', 'Serie A', 'Carlo
Ancelotti', 'Allianz Arena');
```

```

        ('Bayern Munich', 'Munich', 'Bavaria', 'Germany', 'Bundesliga',
        'Jurgen Klopp', 'San Siro'),
        ('Tottenham Hotspur', 'London', 'England', 'United Kingdom',
        'Premier League', 'Alex Ferguson', 'Old Trafford'));

INSERT INTO `TrackEmUP`.`Team` (TeamName, TeamCity, TeamState,
TeamCountry, TeamLeague, TeamManager, TeamStadium)
VALUES ('Barcelona', 'Barcelona', 'Catalonia', 'Spain', 'La Liga',
'Tite', 'Camp Nou'),
        ('Manchester United', 'Manchester', 'England', 'United Kingdom',
        'Premier League', 'Lionel Scaloni',
        'Old Trafford'),
        ('Manchester City', 'Manchester', 'England', 'United Kingdom',
        'Premier League', 'José Mourinho',
        'Etihad Stadium');

ALTER TABLE `TrackEmUP`.`Player`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`Player` (PlayerName, PlayerNationality,
PlayerPosition, PlayerAge, PlayerHeight, PlayerWeight,
PlayerFoot, PlayerTeam)
VALUES ('Sergio Ramos', 'Spain', 'CB', 35, 184, 82, 'R', 'Real Madrid'),
        ('Kylian Mbappe', 'France', 'ST', 23, 178, 73, 'R', 'Paris
Saint-Germain'),
        ('Leonardo Bonucci', 'Italy', 'CB', 34, 187, 85, 'R',
        'Juventus'),
        ('Manuel Neuer ', 'Germany', 'GK', 35, 193, 92, 'R', 'Bayern
Munich'),

        ('Harry Kane', 'United Kingdom', 'ST', 27, 188, 89, 'R',
        'Tottenham Hotspur'),
        ('Lionel Messi', 'Argentina', 'ST', 34, 170, 72, 'R',
        'Barcelona'),
        ('Cristiano Ronaldo', 'Portugal', 'ST', 36, 187, 83, 'R',
        'Juventus'),
        ('Paul Pogba', 'France', 'CM', 28, 191, 84, 'R', 'Manchester
United'),
        ('Neymar', 'Brazil', 'ST', 29, 175, 68, 'R', 'Paris
Saint-Germain'),
        ('Luka Modric', 'Croatia', 'CM', 35, 172, 66, 'R', 'Real
Madrid'),
        ('Sergio Aguero', 'Argentina', 'ST', 33, 173, 70, 'R',
        'Manchester City'),
        ('Antoine Griezmann', 'France', 'ST', 30, 176, 73, 'R',
        'Barcelona'),
        ('Raheem Sterling', 'United Kingdom', 'RW', 27, 170, 70, 'R',
        'Manchester City'),
        ('Toni Kroos', 'Germany', 'CM', 31, 183, 76, 'R', 'Real Madrid'),
        ('Karim Benzema', 'France', 'ST', 33, 185, 81, 'R', 'Real
Madrid'),
        ('Gareth Bale', 'United Kingdom ', 'RW', 31, 183, 79, 'R', 'Real

```

```

Madrid'),
    ('Thiago Silva', 'Brazil', 'CB', 36, 183, 79, 'R', 'Paris
Saint-Germain'),
    ('Marcelo', 'Brazil', 'LB', 34, 174, 66, 'R', 'Real Madrid'),
    ('Sergio Busquets', 'Spain', 'CM', 33, 189, 76, 'R',
'Barcelona'),
    ('Gerard Pique', 'Spain', 'CB', 34, 194, 85, 'R', 'Barcelona'),
    ('Luis Nani', 'Portugal', 'RW', 35, 175, 69, 'R', 'Manchester
United'),
    ('David De Gea', 'Spain', 'GK', 30, 192, 82, 'R', 'Manchester
United'),
    ('Paulo Dybala', 'Argentina', 'ST', 27, 177, 75, 'R',
'Juventus'),
    ('Andrea Pirlo', 'Italy', 'CM', 41, 180, 75, 'R', 'Juventus'),
    ('Gianluigi Buffon', 'Italy', 'GK', 43, 191, 92, 'R',
'Juventus'),
    ('Gonzalo Higuain', 'Argentina', 'ST', 33, 185, 81, 'R',
'Juventus'),
    ('Mario Mandzukic', 'Croatia', 'ST', 34, 191, 88, 'R',
'Juventus'),
    ('Leon Goretzka', 'Germany', 'CM', 26, 188, 84, 'R', 'Bayern
Munich'),
    ('Thomas Muller', 'Germany', 'ST', 32, 183, 76, 'R', 'Bayern
Munich'),
    ('Joshua Kimmich', 'Germany', 'CM', 26, 183, 76, 'R', 'Bayern
Munich'),
    ('Mats Hummels', 'Germany', 'CB', 32, 191, 92, 'R', 'Bayern
Munich'),
    ('Serge Gnabry', 'Germany', 'RW', 25, 180, 75, 'R', 'Bayern
Munich'),
    ('Dele Alli', 'United Kingdom', 'CM', 25, 178, 74, 'R',
'Tottenham Hotspur'),
    ('Eric Dier', 'United Kingdom', 'CM', 27, 183, 76, 'R',
'Tottenham Hotspur'),
    ('Hugo Lloris', 'France', 'GK', 34, 188, 84, 'R', 'Tottenham
Hotspur'),
    ('Kyle Walker', 'United Kingdom', 'RB', 30, 183, 76, 'R',
'Tottenham Hotspur');

```

```

INSERT INTO `TrackEmUP`.`Referee` (RefereeName, RefereeAge,
RefereeNationality, RefereeCategory)
VALUES ('Pierluigi Collina', 60, 'Italy', 'Legend'),
    ('Howard Webb', 50, 'United Kingdom', 'Elite'),
    ('Mark Clattenburg', 50, 'United Kingdom', 'Elite'),
    ('Massimiliano Irrati', 50, 'Italy', 'Legend'),
    ('Michael Oliver', 50, 'United Kingdom', 'Elite'),
    ('Nikola Rizzoli', 50, 'Italy', 'Legend'),
    ('Carlos Velasco Carballo', 50, 'Spain', 'Medium');

```

```

ALTER TABLE `TrackEmUP`.`Match`
    AUTO_INCREMENT = 1;

```

```

INSERT INTO `TrackEmUP`.`Match` (MatchDate, MatchTime, MatchHomeTeam,
ScoreHomeTeam, MatchAwayTeam, ScoreAwayTeam,
MatchLeague, MatchStadium, RefereeID)
VALUES ('2022-01-01', '20:00:00', 'Real Madrid', 2, 'Barcelona', 1, 'La
Liga', 'Allianz Arena', 1),
('2022-12-02', '19:00:00', 'Manchester City', 1, 'Tottenham
Hotspur', 2, 'Premier League', 'Etihad Stadium', 2),
('2022-10-03', '21:00:00', 'Manchester United', 1, 'Manchester
City', 2, 'Serie A', 'Old Trafford', 3),
('2022-07-06', '20:00:00', 'Manchester United', 2, 'Tottenham
Hotspur', 1, 'Premier League', 'Old Trafford', 6),
('2022-06-07', '20:00:00', 'Barcelona', 7, 'Real Madrid', 4, 'La
Liga', 'Camp Nou', 7);

```

```

ALTER TABLE `TrackEmUP`.`Events`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`Events` (EventName)
VALUES ('Goal'),
('Yellow Card'),
('Red Card'),
('Foul'),
('Offside'),
('Penalty'),
('Injury'),
('Own Goal');

```

```

ALTER TABLE `TrackEmUP`.`MatchEvent`
    AUTO_INCREMENT = 1;
INSERT INTO `TrackEmUP`.`MatchEvent` (MatchID, PlayerID, EventType)
VALUES (1, 12, 1),
(1, 14, 1),
(1, 10, 1),
(2, 11, 2),
(2, 13, 2),
(2, 21, 1),
(4, 22, 8);

```

Consultas SQL

```
USE TrackEmUP;
```

1) Proyección y selección

```
# SELECTS ALL COUNTRIES (s)
SELECT *
FROM TrackEmUP.country
ORDER BY CountryID;
```

```
# SELECTS PLAYER NAME, NATIONALITY, POSITION, FOOT & TEAM (p)
SELECT PlayerName, PlayerNationality, PlayerPosition, PlayerFoot,
PlayerTeam
FROM TrackEmUP.player
GROUP BY PlayerName, PlayerPosition
ORDER BY PlayerPosition;
```

2) Join

```
# SELECTS COUNTRIES WITH A LEAGUE
SELECT CountryID, Country, LeagueName, LeagueID
FROM TrackEmUP.league
      JOIN TrackEmUP.country
      ON TrackEmUP.league.Country =
TrackEmUP.country.CountryName;
```

3) Agregados

```
# COUNT HOW MANY TEAMS ARE IN EACH LEAGUE
SELECT LeagueName, COUNT(TeamName) AS TeamCount
FROM TrackEmUP.team
      JOIN TrackEmUP.league
      ON TrackEmUP.team.TeamLeague =
TrackEmUP.league.LeagueName
GROUP BY LeagueName
ORDER BY TeamCount DESC;
```

4) EXTRAS

```
# UPDATE AGUERO'S YELLOW CARD
UPDATE TrackEmUP.matchevent
SET Comments = 'Kicking corner flag'
WHERE MatchEventID = 4;
```

```
# CHECK UPDATED
SELECT MatchEventID, MatchID, PlayerName, EventName, Comments
FROM TrackEmUP.matchevent
JOIN TrackEmUP.player
      ON TrackEmUP.matchevent.PlayerID = TrackEmUP.player.PlayerID
JOIN TrackEmUP.events ON TrackEmUP.matchevent.EventType =
TrackEmUP.events.EventID
WHERE MatchEventID = 4;
```



```

# SEE ALL MATCH EVENTS
SELECT MatchEventID, MatchID, PlayerName, EventName, Comments
FROM TrackEmUP.matchevent
JOIN TrackEmUP.player
    ON TrackEmUP.matchevent.PlayerID = TrackEmUP.player.PlayerID
JOIN TrackEmUP.events ON TrackEmUP.matchevent.EventType =
TrackEmUP.events.EventID;

```

Funciones

```

USE TrackEmUP;

# CHECK IF FUNCTION ALREADY EXISTS & DROP IT (FUNCTION CODE MOVED
TO DML)
# DROP FUNCTION IF EXISTS TrackEmUP.`RegisterCountry`;

# CREATE FUNCTION (BEFORE INSERTING PLAYERS) RegisterCountry
# CREATE FUNCTION TrackEmUP.`RegisterCountry`(vCountry
VARCHAR(250))
#     RETURNS VARCHAR(250)
# BEGIN
#     INSERT INTO TrackEmUP.Country (CountryName) VALUES
(vCountry);
#     RETURN CONCAT('Successfully Registered: ', vCountry);
# END;

# # CALL FUNCTION (ALREADY CALLED IN DML)
# SELECT TrackEmUP.RegisterCountry('Argentina') as
Registered_Country;
# SELECT TrackEmUP.RegisterCountry('Brazil') as
Registered_Country;
# SELECT TrackEmUP.RegisterCountry('Portugal') as
Registered_Country;
# SELECT TrackEmUP.RegisterCountry('Croatia') as
Registered_Country;

# CHECK IF FUNCTION ALREADY EXISTS & DROP IT
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchTeamsByCountry`;

# CREATE PROCEDURE
CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchTeamsByCountry`(vCountry VARCHAR(250))
BEGIN
    SELECT * FROM TrackEmUP.Team WHERE TeamCountry = vCountry;
END;

```

```

# CALL PROCEDURE
CALL TrackEmUP.`TrackEmUP.SearchTeamsByCountry`('Argentina'); #
RETURNS 0 ROWS
CALL TrackEmUP.`TrackEmUP.SearchTeamsByCountry`('Spain');
# RETURNS 2 ROWS

# CREATE A FUNCTION THAT SHOWS ALL MATCHES BETWEEN A DATE
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchMatchesByDate`;

CREATE PROCEDURE TrackEmUP.`TrackEmUP.SearchMatchesByDate`(sDate
DATE, eDate DATE)
BEGIN
    SELECT * FROM TrackEmUP.Match WHERE MatchDate BETWEEN sDate AND
eDate;
END;

CALL TrackEmUP.`TrackEmUP.SearchMatchesByDate`('2022-01-01',
'2022-06-30');
# RETURNS 2 ROWS

# CREATE A FUNCTION THAT SHOWS ALL PLAYERS WITH RED CARDS
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchPlayersWithRedCards`;

CREATE PROCEDURE TrackEmUP.`TrackEmUP.SearchPlayersWithRedCards`()
BEGIN
    SELECT MatchEventID, MatchID, matchEvent.PlayerID, PlayerName,
EventTime, Comments
    FROM TrackEmUP.MatchEvent
        JOIN TrackEmUP.Player ON matchEvent.PlayerID =
Player.PlayerID
    WHERE EventType = 3;
END;

CALL TrackEmUP.`TrackEmUP.SearchPlayersWithRedCards`();
# RETURNS 0 ROWS

# CREATE A FUNCTION THAT SHOWS ALL PLAYERS WITH YELLOW CARDS
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchPlayersWithYellowCards`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchPlayersWithYellowCards`()
BEGIN
    SELECT MatchEventID, MatchID, matchEvent.PlayerID, PlayerName,
EventTime, Comments
    FROM TrackEmUP.MatchEvent
        JOIN TrackEmUP.Player ON matchEvent.PlayerID =
Player.PlayerID

```

```

WHERE EventType = 2;
END;

CALL TrackEmUP.`TrackEmUP.SearchPlayersWithYellowCards`();
# RETURNS 2 ROWS

# CREATE A FUNCTION THAT SHOWS ALL PLAYERS WITH GOALS
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchPlayersWithGoals`;

CREATE PROCEDURE TrackEmUP.`TrackEmUP.SearchPlayersWithGoals`()
BEGIN
    SELECT MatchEventID, MatchID, matchEvent.PlayerID, PlayerName,
    EventTime, Comments
    FROM TrackEmUP.MatchEvent
    JOIN TrackEmUP.Player ON matchEvent.PlayerID =
    Player.PlayerID
    WHERE EventType = 1;
END;

CALL TrackEmUP.`TrackEmUP.SearchPlayersWithGoals`();
# RETURNS 4 ROWS

# CREATE A FUNCTION THAT SHOWS ALL THE GAMES OF A TEAM
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchMatchesByTeam`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchMatchesByTeam`(teamName VARCHAR(250))
BEGIN
    SELECT * FROM TrackEmUP.Match WHERE MatchHomeTeam = teamName OR
    MatchAwayTeam = teamName;
END;

CALL TrackEmUP.`TrackEmUP.SearchMatchesByTeam`('Manchester
United');
# RETURNS 2 ROWS

# CREATE A FUNCTION THAT SHOWS ALL THE GAMES WITH A SPECIFIC
REFEREE
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchMatchesByReferee`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchMatchesByReferee`(refID INT)
BEGIN
    SELECT MatchID,
           MatchDate,
           MatchHomeTeam,
           ScoreHomeTeam,
           MatchAwayTeam,
           ScoreAwayTeam,
           MatchLeague,

```

```

        MatchStadium
    FROM TrackEmUP.Match
    WHERE RefereeID = refID;
END;

CALL TrackEmUP.`TrackEmUP.SearchMatchesByReferee`(2);
# RETURNS 1 ROW

# CREATE A FUNCTION THAT SHOWS ALL THE GAMES IN A STADIUM
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchMatchesByStadium`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchMatchesByStadium`(stadiumName
VARCHAR(250))
BEGIN
    SELECT MatchID,
           MatchDate,
           MatchHomeTeam,
           ScoreHomeTeam,
           MatchAwayTeam,
           ScoreAwayTeam,
           MatchLeague,
           MatchStadium
    FROM TrackEmUP.Match
    WHERE MatchStadium = stadiumName;
END;

CALL TrackEmUP.`TrackEmUP.SearchMatchesByStadium`('Old Trafford');
# RETURNS 2 ROWS

# SHOW ALL MATCH EVENTS WHERE A PLAYER WAS INVOLVED
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchMatchEventsByPlayer`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchMatchEventsByPlayer`(ID INT)
BEGIN
    SELECT matchevent.MatchEventID, matchevent.MatchID,
           Events.EventName, matchevent.EventTime, matchevent.Comments
    FROM TrackEmUP.MatchEvent
           JOIN TrackEmUP.Events ON matchevent.EventType =
           Events.EventID
    WHERE matchevent.PlayerID = ID;
END;

CALL TrackEmUP.`TrackEmUP.SearchMatchEventsByPlayer`(11);
# RETURNS 1 ROW

```

```

# SHOW ALL PLAYERS OF A TEAM
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchPlayersByTeam`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchPlayersByTeam`(teamName VARCHAR(250))
BEGIN
    SELECT player.playerid,
           playername,
           playernationality,
           playerposition,
           playerage,
           playerheight,
           playerweight,
           playerfoot
    FROM TrackEmUP.Player
    WHERE Player.PlayerTeam = teamName
    ORDER BY playerposition;
END;

CALL TrackEmUP.`TrackEmUP.SearchPlayersByTeam`('Juventus');
# RETURNS 7 ROWS

# SHOW ALL PLAYERS OF A TEAM WITH A SPECIFIC POSITION
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchPlayersByTeamAndPosition`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchPlayersByTeamAndPosition`(teamName
VARCHAR(250), position VARCHAR(3))
BEGIN
    SELECT player.playerid,
           playername,
           playernationality,
           playerposition,
           playerage,
           playerheight,
           playerweight,
           playerfoot
    FROM TrackEmUP.Player
    WHERE Player.PlayerTeam = teamName
        AND Player.PlayerPosition = position;
END;

CALL
TrackEmUP.`TrackEmUP.SearchPlayersByTeamAndPosition`('Barcelona',
'ST');
# RETURNS 2 ROWS

# SHOW ALL PLAYERS OF A TEAM WITH A SPECIFIC POSITION

```

```
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchPlayersByPosition`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchPlayersByPosition`(position VARCHAR(3))
BEGIN
    SELECT * FROM TrackEmUP.Player WHERE Player.PlayerPosition =
position;
END;

CALL TrackEmUP.`TrackEmUP.SearchPlayersByPosition`('GK');
# RETURNS 4 ROWS

# SHOW ALL PLAYERS OF A NATIONALITY
DROP PROCEDURE IF EXISTS
TrackEmUP.`TrackEmUP.SearchPlayersByNationality`;

CREATE PROCEDURE
TrackEmUP.`TrackEmUP.SearchPlayersByCountry`(country VARCHAR(250))
BEGIN
    SELECT * FROM TrackEmUP.Player WHERE Player.PlayerNationality =
country;
END;

CALL TrackEmUP.`TrackEmUP.SearchPlayersByCountry`('Spain'); #
RETURNS 4 ROWS
```

Resultados y Conclusiones

Nuestra tarea fue crear una base de datos con lo esencial para poder implementarla en la vida cotidiana. Los resultados del proyecto fueron los esperados. Nuestra base de datos logró correr con toda la información de jugadores, entrenadores, equipos, y más, a la perfección, e incluso se le pudo dar un enfoque estadístico.

Quedamos satisfechos con la realización de este proyecto porque pudimos unir dos de nuestras áreas de interés, como son el fútbol y las bases de datos, con el objetivo de proponer una mejora a los sistemas que se emplean actualmente.

En un futuro nos gustaría generar la opción de tener diferentes tipos de usuarios y poder implementarlo primero a nivel torneo interno UP, posteriormente inter colegial, luego nacional y finalmente a nivel mundial.