

ESCUELAS SALESIANAS MARÍA AUXILIADORA

CICLO FORMATIVO DE GRADO SUPERIOR
DESARROLLO DE APLICACIONES MULTIPLATAFORMA

PROYECTO 3: Base datos LFP

Ruso Álvarez, José Francisco
SEVILLA, 2020

INDICE

1. ESTUDIO DEL PROBLEMA Y ANÁLISIS DEL SISTEMA.	3
1.1. INTRODUCCIÓN.....	3
1.2. FUNCIONES Y RENDIMIENTOS DESEADOS	5
1.3. OBJETIVOS.	17
1.4. MODELADO DE LA SOLUCIÓN.....	17
1.4.1. RECURSOS HUMANOS.....	17
1.4.2. RECURSOS HARDWARE.....	17
1.4.3. RECURSOS SOFTWARE.	17
2. EJECUCIÓN DE LA PRÁCTICA	17
2.1. ELABORACIÓN DE LA DOCUMENTACIÓN TÉCNICA E IMPLEMENTACIÓN DE LA APLICACIÓN	17
3. DOCUMENTACIÓN DEL SISTEMA	26
3.1. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DE LA APLICACIÓN	26
3.2. MANUAL DE USUARIO	26
4. CONCLUSIONES FINALES	30
4.1. GRADO DE CUMPLIMIENTO DE LOS OBJETIVOS FIJADOS	30
4.2. PROPUESTA DE MODIFICACIONES O AMPLIACIONES FUTURAS DEL SISTEMA IMPLEMENTADO.....	31
5. BIBLIOGRAFÍA	31

1. ESTUDIO DEL PROBLEMA Y ANÁLISIS DEL SISTEMA.

1.1. INTRODUCCIÓN

El presente proyecto está englobado dentro de un entorno formativo. Se trata de una aplicación de escritorio que ofrece una gestión de clubes y futbolistas dentro de una base de datos por parte de un administrador. En el enunciado se detalla las características del proyecto:

El Granada C.F. pretende desarrollar una aplicación que permita almacenar y gestionar en una base de datos con información acerca de los futbolistas que han militado en el club a lo largo de su historia. La idea gusta mucho en la Liga de Fútbol Profesional, y deciden hacerla extensiva al resto de clubes de primera y segunda división. De esta forma tendremos que para cada futbolista se almacenaría la siguiente información: su nombre y apellido, su año de nacimiento, su nacionalidad y su NIF. De cada club de fútbol se guardaría su nombre oficial, el año de creación y el nombre de su estadio. Además, se almacenaría información de qué futbolistas militaron en que clubes en las diferentes campañas ligueras.

NOTA: Para simplificar, supondremos que un futbolista dado sólo puede militar en un club a lo largo de una temporada.

Funcionalidades requeridas

Se debe diseñar una aplicación Java con una interfaz gráfica de usuario que permita:

1. Introducir datos de futbolistas.
2. Introducir datos de clubes de fútbol.
3. Modificar datos de futbolistas y clubes.
4. Asociar futbolistas a clubes (un futbolista puede estar asociado a varios clubes si ha militado en varios).
5. Obtener la lista de futbolistas asociados a un club.
6. Obtener la lista de clubes en los que ha militado un futbolista.
7. Listar todos los clubes.
8. Listar todos los futbolistas.

En cada caso, se consideraría la forma adecuada de interactuar con el usuario; por ejemplo, evitando que haya que escribir el nombre completo de un club para poder ver los futbolistas que han militado en él, evitando que haya tablas editables cuya actualización no se refleje en la base de datos, etc.

Diagrama de clases

Tras analizar el problema se identifican 3 clases con una relación de composición entre ellas. La clase Histórico forma parte de ambas. La vida de esta clase aparece de la relación entre club y futbolista. La cardinalidad es de 1...* entre ambas, ya que un club puede tener muchos futbolistas, y un mismo futbolista puede pertenecer a muchos clubes (aunque esto vendrá limitado por el atributo temporada, el cual limitará su participación a un equipo por temporada).

Todas ellas están relacionadas con una clase funcionalidades, que será la encargada de llevar a cabo todos los procesos que requiera este proyecto.

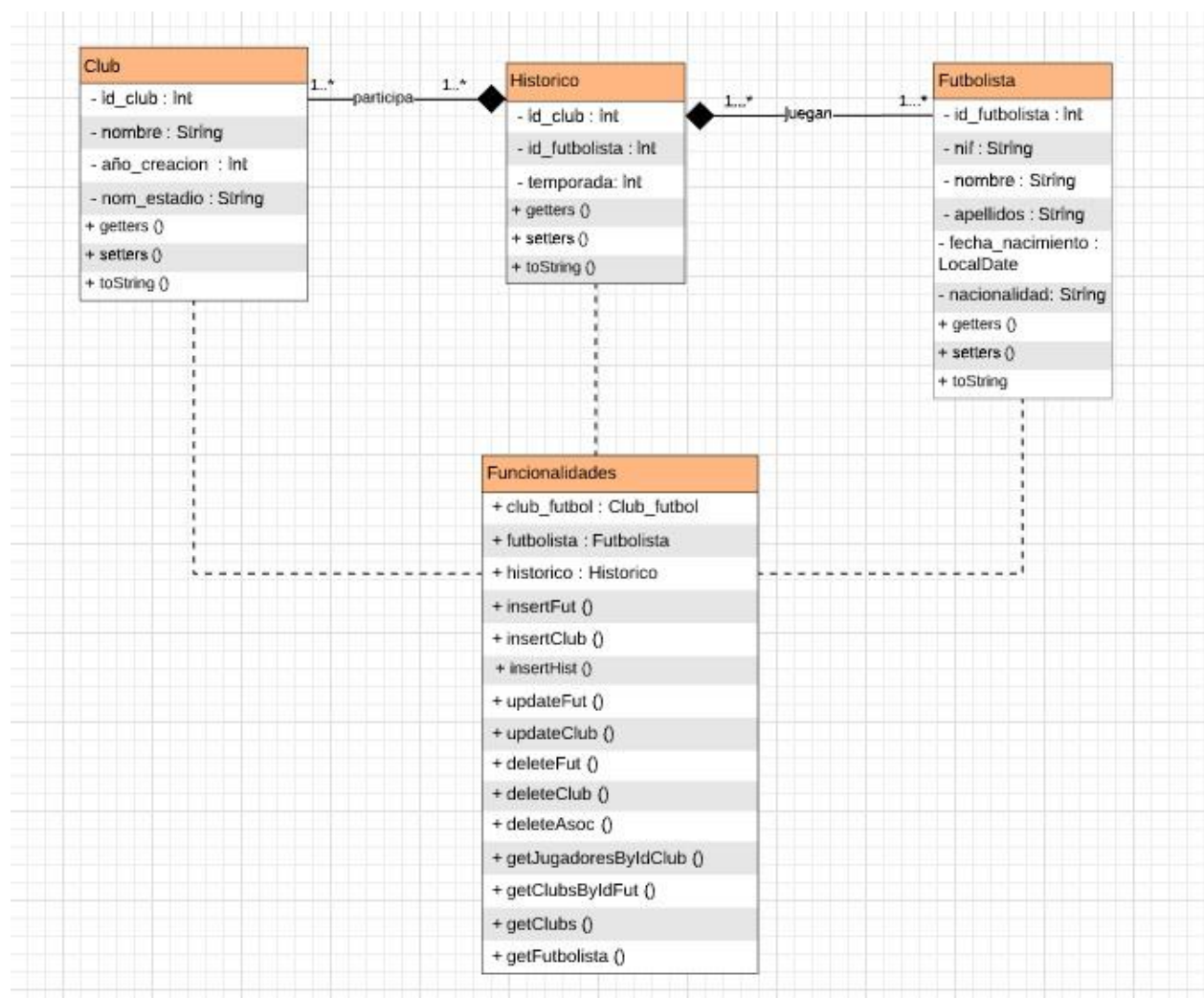
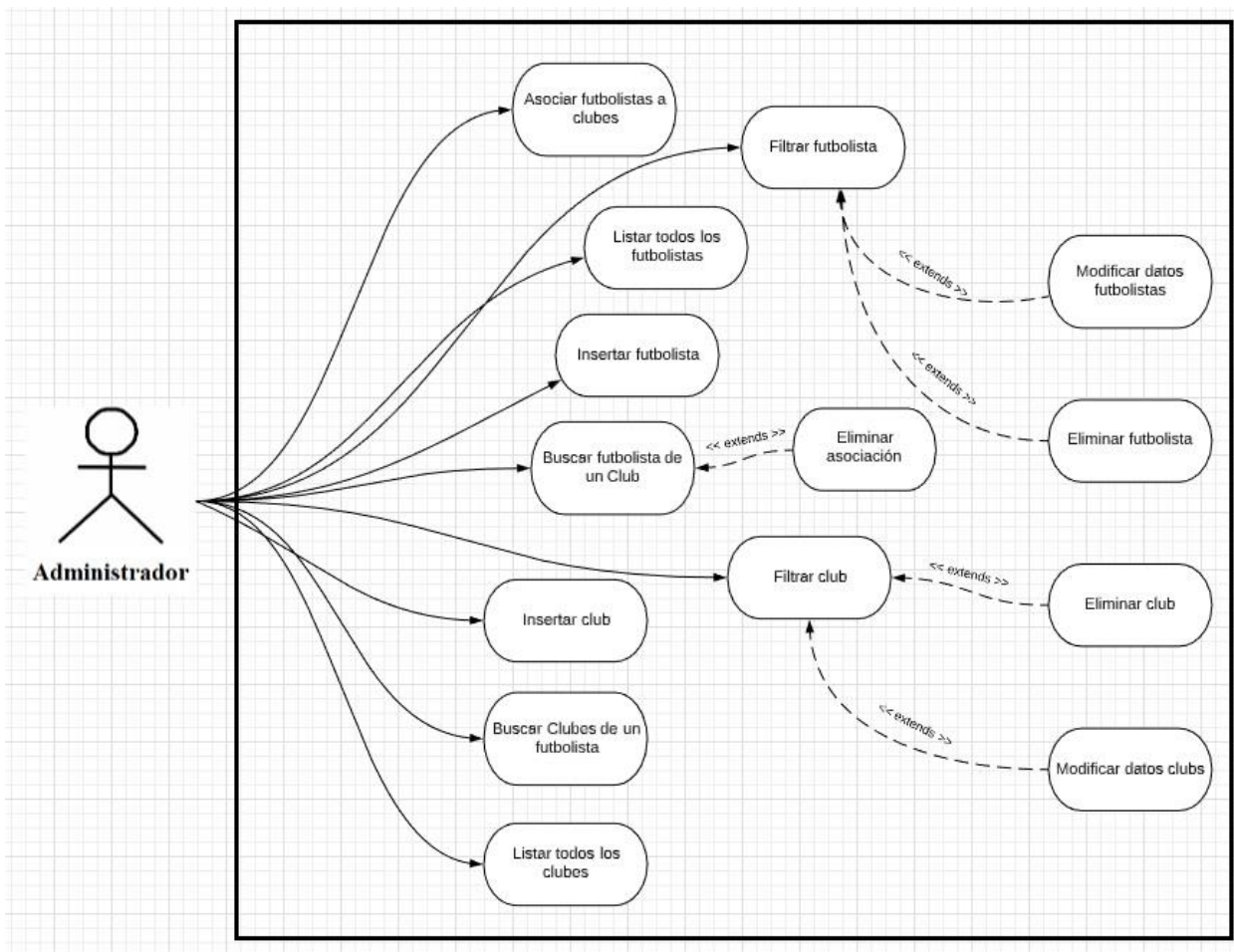


Diagrama de clases

1.2. FUNCIONES Y RENDIMIENTOS DESEADOS

Se necesita que nuestra aplicación sea capaz de manipular datos sobre futbolistas y clubes además de la relación entre estos. Para poder consultar y/o modificar una información determinada. Para ello, proponemos este diagrama de casos de uso (UML) para el proyecto, que muestra todas las interacciones del actor con nuestro sistema. Se observa que existe una serie de casos de uso que son «extend», ya que requieren de ese caso previo para poder llegar a ellos.



A partir de este diagrama se realiza una tabla por cada caso de uso, que detallará lo que ocurre en cada uno.

Nombre: Insertar un futbolista

ID: CU-01

Descripción:

Proceso por el cual se ingresa un nuevo futbolista a la base de datos.

Actores:

Administrador

Precondiciones:

Ninguno

Curso normal del caso de uso:

1. El usuario rellena los campos de datos.
2. El sistema valida los datos introducidos.
3. El sistema pide confirmación para realizar la inserción.
4. El usuario acepta introducir esos datos.
5. Se muestra mensaje de inserción exitosa.

Postcondiciones:

Se inserta un futbolista en la base de datos.

Alternativas:

2. El sistema lanza un mensaje de error informando de que los datos son incorrectos y finaliza el proceso.
 4. El usuario elige cancelar, no produciéndose la inserción.
-

Nombre: Insertar un club

ID: CU-02

Descripción:

Proceso por el cual se ingresa un nuevo club a la base de datos.

Actores:

Administrador

Precondiciones:

Ninguna

Curso normal del caso de uso:

1. El usuario rellena los campos de datos.
 2. El sistema valida los datos introducidos.
 3. El sistema pide confirmación para realizar inserción.
-

-
4. El usuario acepta introducir esos datos.
 5. Se muestra mensaje de inserción exitosa.

Postcondiciones:

Se inserta un futbolista en la base de datos.

Alternativas:

2. El sistema lanza un mensaje de error informando de que los datos son incorrectos y finaliza el proceso.
 4. El usuario elige cancelar, no produciéndose la inserción.
-

Nombre: Listar todos los futbolistas

ID: CU-03

Descripción:

El usuario consulta todos futbolistas que están dentro de la base de datos. Sin ningún criterio de búsqueda.

Actores:

Administrador

Precondiciones:

Que existas futbolistas inscritos.

Curso normal del caso de uso:

1. El usuario solicita al sistema que le muestre todos los futbolistas.
2. El sistema muestra sus datos.

Postcondiciones:

El sistema genera una tabla mostrando todos los futbolistas inscritos en la base de datos.

Alternativas:

2. El sistema no muestra ningún contenido al no haber futbolistas.
-

Nombre: Listar todos los clubes

ID: CU-04

Descripción:

El usuario consulta los clubes que están dentro de la base de datos. Sin ningún criterio de búsqueda.

Actores:

Administrador

Precondiciones:

Que existan clubes inscritos.

Curso normal del caso de uso:

1. El usuario solicita al sistema que le muestre todos los futbolistas.
2. El sistema muestra sus datos.

Postcondiciones:

El sistema genera una tabla mostrando todos los clubes inscritos en la base de datos

Alternativas:

2. El sistema no muestra ningún contenido al no haber ningún club.
-

Nombre: Filtra futbolista

ID: CU-05

Descripción:

El usuario consulta un club específico con el criterio del nombre del futbolista.

Actores:

Administrador

Precondiciones:

Que existan futbolistas inscritos.

Curso normal del caso de uso:

1. El usuario escribe en el campo de filtrado cualquier dato del futbolista que desea buscar.
2. El sistema dinámicamente va filtrando los resultados conforme va escribiendo.
3. El sistema muestra la tabla con el filtro realizado.

Postcondiciones:

El sistema genera una tabla mostrando el futbolista deseado.

Alternativas:

3. El sistema no muestra ningún contenido al no haber ningún futbolista.
-

Nombre: Filtrar club

ID: CU-06

Descripción:

El usuario consulta un club específico con el criterio del nombre del club.

Actores:

Administrador

Precondiciones:

Que existan clubes inscritos.

Curso normal del caso de uso:

1. El usuario escribe en el campo de filtrado cualquier dato del equipo que desea buscar.
2. El sistema dinámicamente va filtrando los resultados conforme va escribiendo.
3. El sistema muestra la tabla con el filtro realizado.

Postcondiciones:

El sistema genera una tabla mostrando el equipo deseado.

Alternativas:

3. El sistema no muestra ningún contenido al no haber ningún club.
-

Nombre: Buscar clubes de un futbolista

ID: CU-07

Descripción:

El usuario consulta los clubes en los que ha jugado un futbolista determinado.

Actores:

Administrador

Precondiciones:

Que existan jugadores y clubes sobre los que buscar.

Curso normal del caso de uso:

1. El usuario selecciona de un listado el futbolista sobre el que consultar.
2. El sistema le muestra los clubes donde ha militado ese jugador.

Postcondiciones:

Se muestra una tabla personalizada con la búsqueda que ha realizado el usuario, visionando los clubes donde ha jugado un futbolista concreto.

Alternativas:

2. El sistema no muestra ningún contenido, ya que ese futbolista no pertenece a ningún club.
-

Nombre: Buscar futbolistas de un club

ID: CU-08

Descripción:

El usuario consulta los jugadores que ha tenido un club determinado.

Actores:

Administrador

Precondiciones:

Que existan jugadores y clubes sobre los que buscar.

Curso normal del caso de uso:

1. El usuario selecciona de un listado el club sobre el que consultar.
2. El sistema le muestra los jugadores que han jugado en dicho club.

Postcondiciones:

Se muestra una tabla personalizada con la búsqueda que ha realizado el usuario, visionando los jugadores que tiene ese club en concreto.

Alternativas:

2. El sistema no muestra ningún contenido, ya que ese club no tiene futbolistas.

Nombre: Asociar futbolistas a un club

ID: CU-09

Descripción: Proceso por el cual el usuario asocia un jugador a un club concreto, en una temporada en particular.

Actores:

Administrador

Precondiciones:

Que que existan jugadores y clubes sobre los que hacer la asociación

Curso normal del caso de uso:

1. El usuario le pide al sistema que le muestre los jugadores sobre los que hacer la asociación.
2. El sistema muestra los jugadores disponibles.
3. El usuario selecciona un jugador de la tabla, elige el club sobre el que realizar la asociación, e indica la temporada.
4. El sistema valida los datos de la asociación.
5. El sistema pide confirmación.
6. El usuario acepta a realizar la asociación.
7. Se muestra mensaje de asociación exitosa.

Postcondiciones:

Se crea una asociación entre un futbolista y un club en una temporada determinada.

Alternativas:

2. El sistema muestra una lista vacía ya que no hay jugadores ni clubes en la base de datos.
3. El usuario indica una temporada en la que ese jugador ha pertenecido a otro equipo.
 - 3.1 El sistema informa del error y finaliza el proceso.
6. El usuario cancela realizar la asociación, no produciéndose.

Nombre: Eliminar asociación

ID: CU-10

Descripción:

Proceso por el cual el usuario desvincula un jugador de club concreto, en una temporada en particular.

Actores:

Administrador

Precondiciones:

Que exista una asociación entre un futbolista y un club.

Curso normal del caso de uso:

1. El usuario selecciona de un listado el club sobre el que consultar los jugadores que posee.
2. El sistema le muestra los jugadores que han jugado en dicho club.
3. El usuario selecciona la relación que desea eliminar.
4. El sistema pide confirmación para proceder a la eliminación.
5. El usuario confirma la eliminación
6. Se muestra mensaje de asociación exitosa.

Postcondiciones:

Se produce la eliminación de la asociación entre un jugador y un club, en una temporada seleccionada.

Alternativas:

2. Se muestra una tabla vacía ya que dicho equipo no tiene jugadores asociados.
5. El usuario cancela el proceso de eliminación.

Nombre: Modificar datos de futbolista

ID: CU-11

Descripción:

Proceso por el cual se cambian los datos de un futbolista concreto.

Actores:

Administrador

Precondiciones:

Que exista un jugador sobre el que modificar los datos.

Curso normal del caso de uso:

1. El usuario pulsa en carga la tabla de todos los futbolistas.
2. El sistema le muestra la tabla.
3. El usuario filtra o busca en la tabla el jugador sobre el que desea realizar los cambios, y pincha sobre él.
4. El sistema le completa los campos de texto con los datos del futbolista seleccionado.
5. El usuario modifica los datos que necesite.
6. El sistema valida los datos introducidos.
7. El sistema pide confirmación para realizar la modificación.
8. El usuario acepta realizar la modificación.
9. Se muestra mensaje de modificación exitosa.

Postcondiciones:

Se realiza una modificación de uno o varios campos del futbolista.

Alternativas:

2. Se muestra una tabla vacía al no haber jugadores registrados.
4. El filtro realizado por el usuario le lleva a una tabla vacía.
6. Existen errores en los campos modificados y el sistema finaliza el proceso.
8. El usuario rechaza realizar la modificación.

Nombre: Eliminar futbolista

ID: CU-12

Descripción:

Proceso por el cual se elimina a un futbolista

Actores:

Administrador

Precondiciones:

Que exista un futbolista que eliminar.

Curso normal del caso de uso:

1. El usuario pulsa en carga la tabla de todos los futbolistas.

2. El sistema le muestra la tabla.
3. El usuario filtra o busca en la tabla el jugador sobre el que desea eliminar, y pincha sobre él.
4. El sistema le completa los campos de texto con los datos del futbolista seleccionado.
5. El usuario pulsa en eliminar.
6. El sistema pide confirmación para realizar la eliminación.
7. El usuario acepta realizar la eliminación.
8. Se muestra mensaje de eliminación exitosa.

Postcondiciones:

Se elimina un futbolista determinado.

Alternativas:

3. Se muestra una tabla vacía al no haber jugadores registrados.
5. El filtro realizado por el usuario le lleva a una tabla vacía.
7. El usuario rechaza realizar la eliminación.

Nombre: Modificar datos de club

ID: CU-13

Descripción:

Proceso por el cual se cambian los datos de un club concreto.

Actores:

Administrador

Precondiciones:

Que exista un club sobre el que modificar los datos.

Curso normal del caso de uso:

1. El usuario pulsa en carga la tabla de todos los clubes.
2. El sistema le muestra la tabla.
3. El usuario filtra o busca en la tabla el club sobre el que desea realizar los cambios, y pincha sobre él.
4. El sistema le completa los campos de texto con los datos del club seleccionado.
5. El usuario modifica los datos que necesite.
6. El sistema valida los datos introducidos.
7. El sistema pide confirmación para realizar la modificación.
8. El usuario acepta realizar la modificación.
9. Se realiza la modificación

Postcondiciones:

Se realiza una modificación de uno o varios campos del club.

Alternativas:

2. Se muestra una tabla vacía al no haber clubes registrados.

4. El filtro realizado por el usuario le lleva a una tabla vacía.
 6. Existen errores en los campos modificados y el sistema finaliza el proceso.
 8. El usuario rechaza realizar la modificación.
-

Nombre: Eliminar club

ID: CU-14

Descripción:

Proceso por el cual se elimina a un club

Actores:

Administrador

Precondiciones:

Que exista un club que eliminar.

Curso normal del caso de uso:

1. El usuario pulsa en carga la tabla de todos los clubes.
2. El sistema le muestra la tabla.
3. El usuario filtra o busca en la tabla el club sobre el que desea eliminar, y pincha sobre él.
4. El sistema le completa los campos de texto con los datos del club seleccionado.
5. El usuario pulsa en eliminar.
6. El sistema pide confirmación para realizar la eliminación.
7. El usuario acepta realizar la eliminación.
8. Se muestra mensaje de eliminación exitosa.

Postcondiciones:

Se elimina un club determinado.

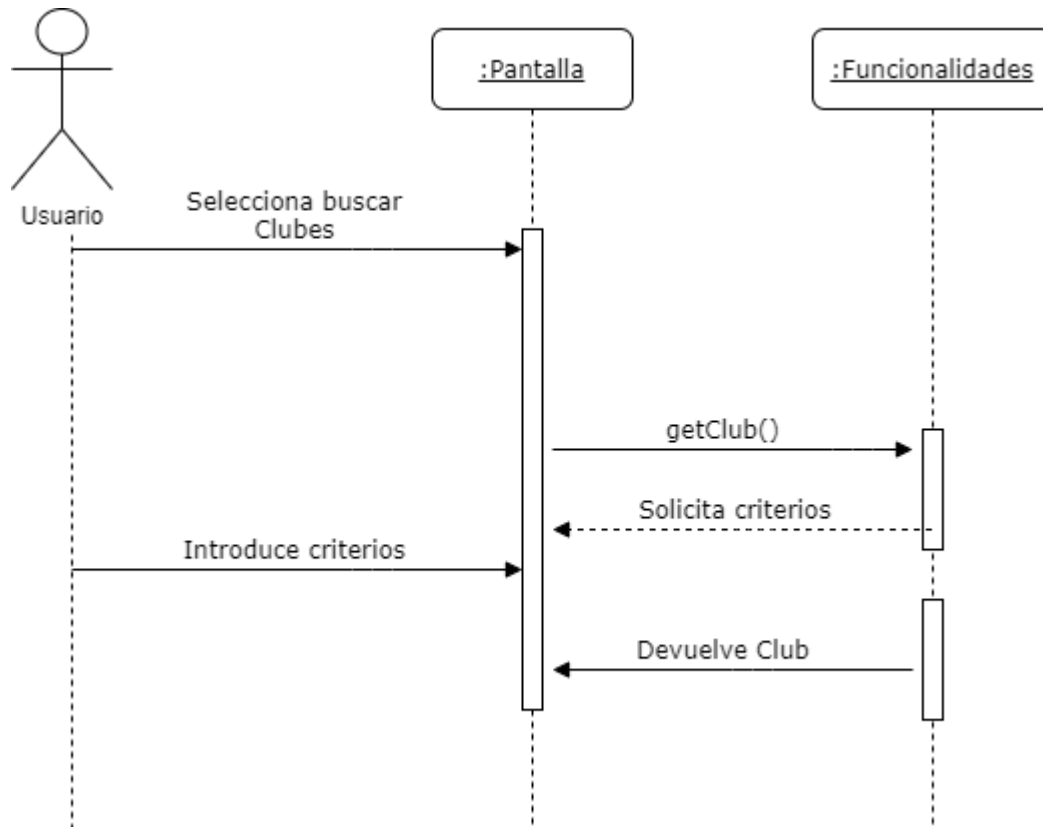
Alternativas:

2. Se muestra una tabla vacía al no haber jugadores registrados.
 5. El filtro realizado por el usuario le lleva a una tabla vacía.
 7. El usuario rechaza realizar la eliminación.
-

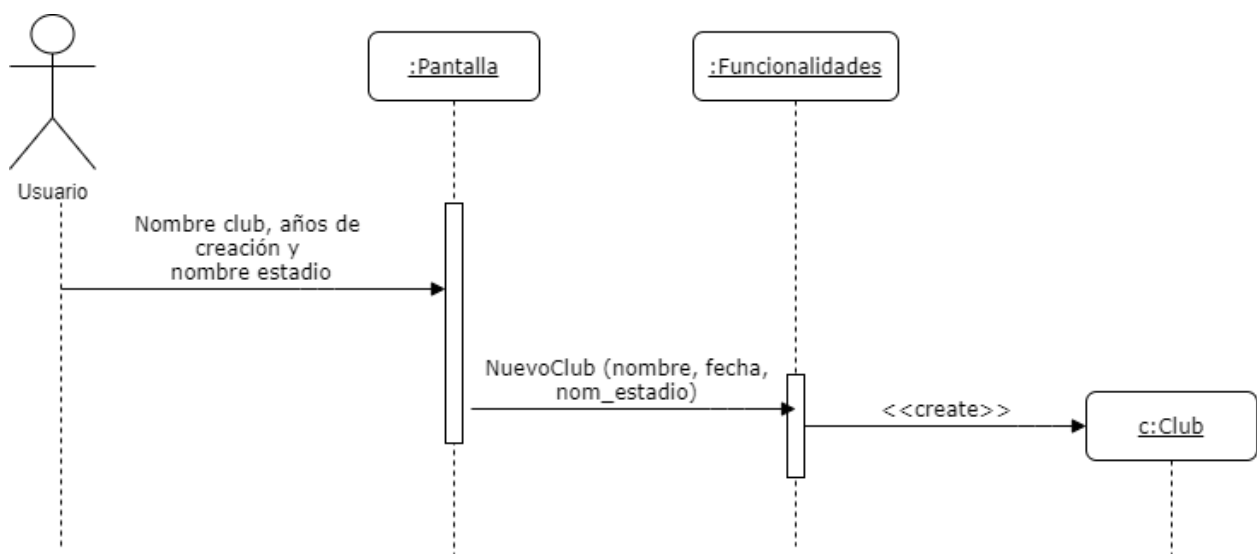
Diagramas de secuencia

Por último, tenemos una serie de diagramas de secuencia que indicarán de una manera visual la interacción que se lleva a cabo entre el usuario y el sistema. Para no sobrecargar la memoria hemos elegido mostrar el diagrama del CRUD de Clubes.

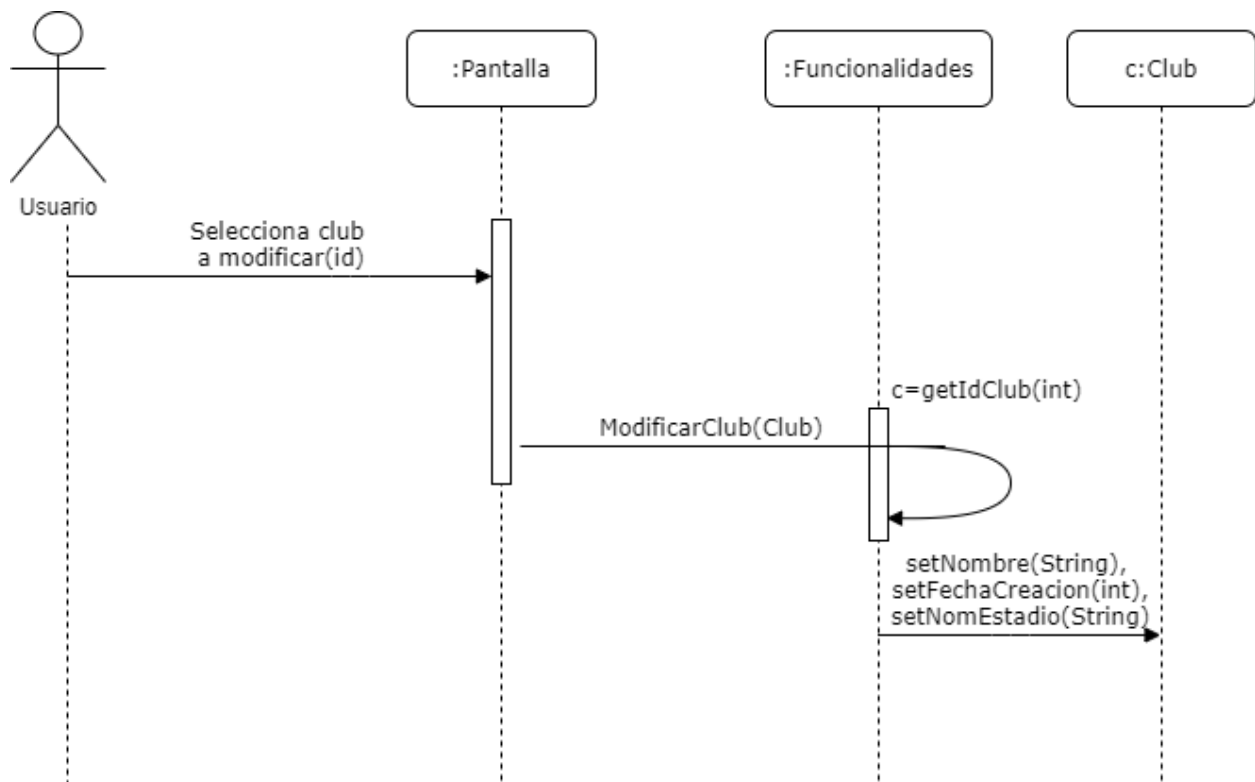
Buscar clubes



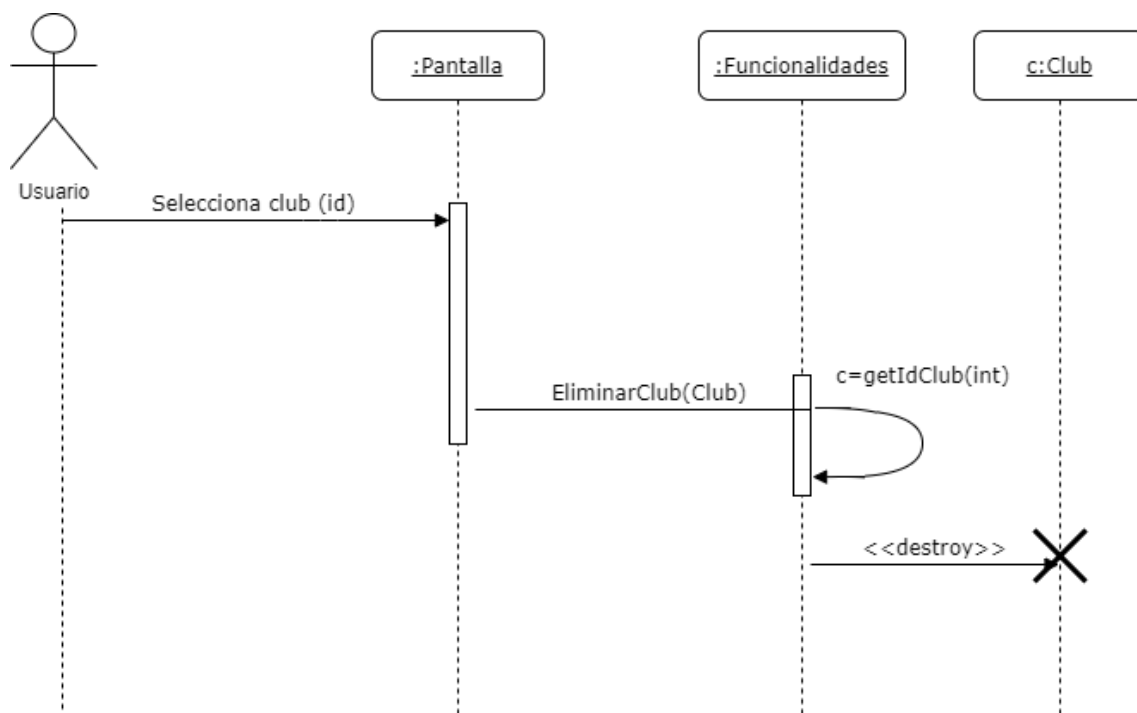
Insertar club



Modificar club



Borrar club



1.3. OBJETIVOS.

Implementar una plataforma para poder gestionar los datos de clubes y futbolistas de la Liga de Fútbol Profesional, cubriendo las funcionalidades básicas de la capa de persistencia del software CRUD (crear, leer, actualizar y borrar).

1.4. MODELADO DE LA SOLUCIÓN

1.4.1.RECURSOS HUMANOS.

La aplicación requerirá de un administrador que será el encargado de la propia base de datos.

1.4.2. RECURSOS HARDWARE.

Solo se requiere de un ordenador personal para poder ejecutar nuestra aplicación de escritorio además de una base de datos (local o en un servidor) en la que podamos almacenar dichos datos.

1.4.3. RECURSOS SOFTWARE.

Para la ejecución de nuestra aplicación tenemos que tener instalado en el ordenador un JRE ya que se trata de una aplicación Java. Para la base de datos se necesita un gestor basado en SQL, preferiblemente MYSQL.

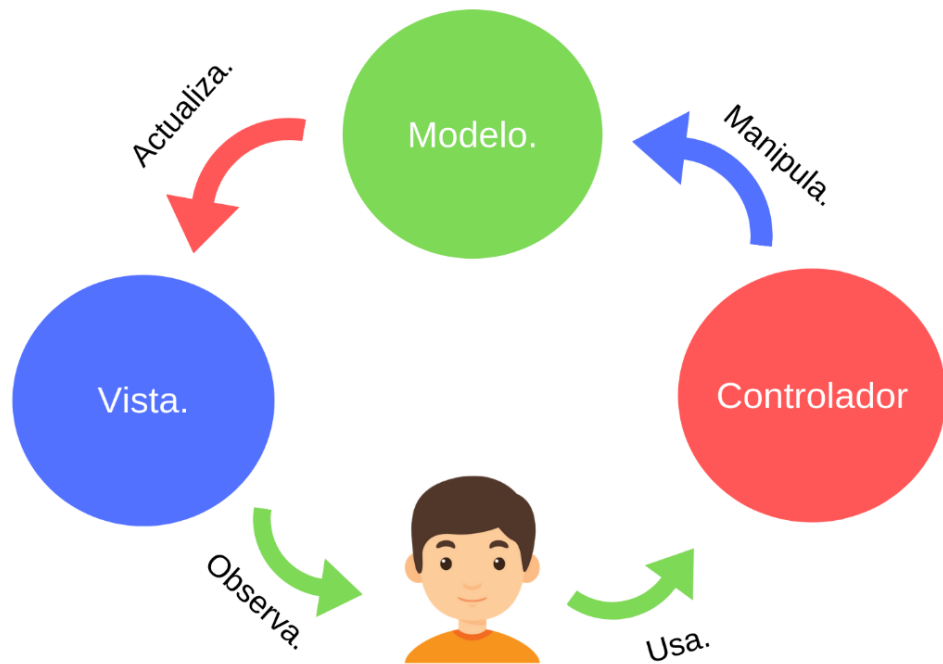
2. EJECUCIÓN DE LA PRÁCTICA

2.1. ELABORACIÓN DE LA DOCUMENTACIÓN TÉCNICA E IMPLEMENTACIÓN DE LA APLICACIÓN

Dentro de nuestra aplicación hay que “dividirla” en dos vertientes. La gestión de la base de datos, y el diseño del propio proyecto Java (el código de la aplicación).

APLICACIÓN

Nuestra aplicación se divide en tres grandes bloques:



Modelo-Vista-Controlador

Vista:

La vista tendrá todos los componentes de diseño (botones, combobox, jtable, etc) que serán declarados como públicos para luego asignarles un evento en el controlador. De esta manera, podemos trabajar con todos los ítems del JFRAME a través del controlador.

Controlador:

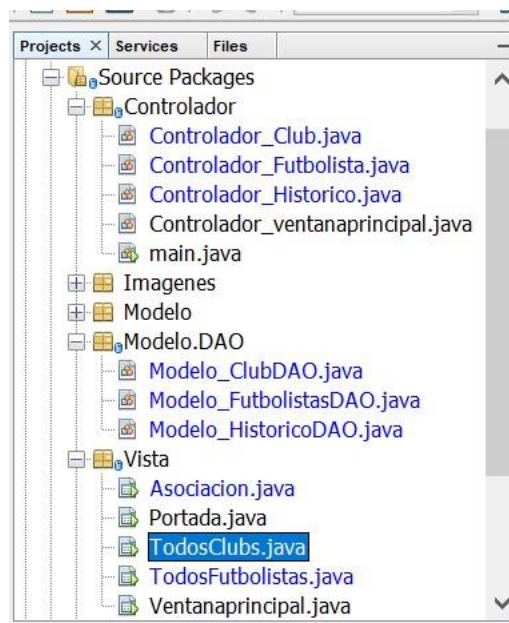
El controlador será el encargado de la comunicación entre la vista y el modelo. Aquí es donde se declaran las vistas y los modelos. El controlador heredará los métodos de los ActionListener, MouseListener, etc, que requiera. A cada botón, tabla o lista que queramos que tenga interacción con el usuario le tendremos que asignar estos ActionListener, KeyListener, etc, para realizar una funcionalidad específica.

Modelo:

El modelo es el encargado de realizar todas las operaciones para obtener la información de la base de datos. Para llevar esto a cabo, necesitamos crear una conexión con una serie de métodos declarados en la clase Conexión, que será padre de los distintos modelos (Modelo_Clubes, Modelo_Futbolistas, etc).

Además, en el modelo están las clases POJO (Plain Old Java Object) que serán clases básicas con los respectivos constructores y métodos getter y setter, sirviendo para asignar los datos en el modelo.

La estructura del proyecto fue la siguiente:



Estructura del proyecto

BASE DE DATOS

Nuestra base de datos está gestionada en MYSQL. Ya que se trata de una base de datos relacional. Además, ha sido implementada en un servidor externo (el facilitado por la escuela).

Aquí se muestra el diagrama de Entidad-Relación

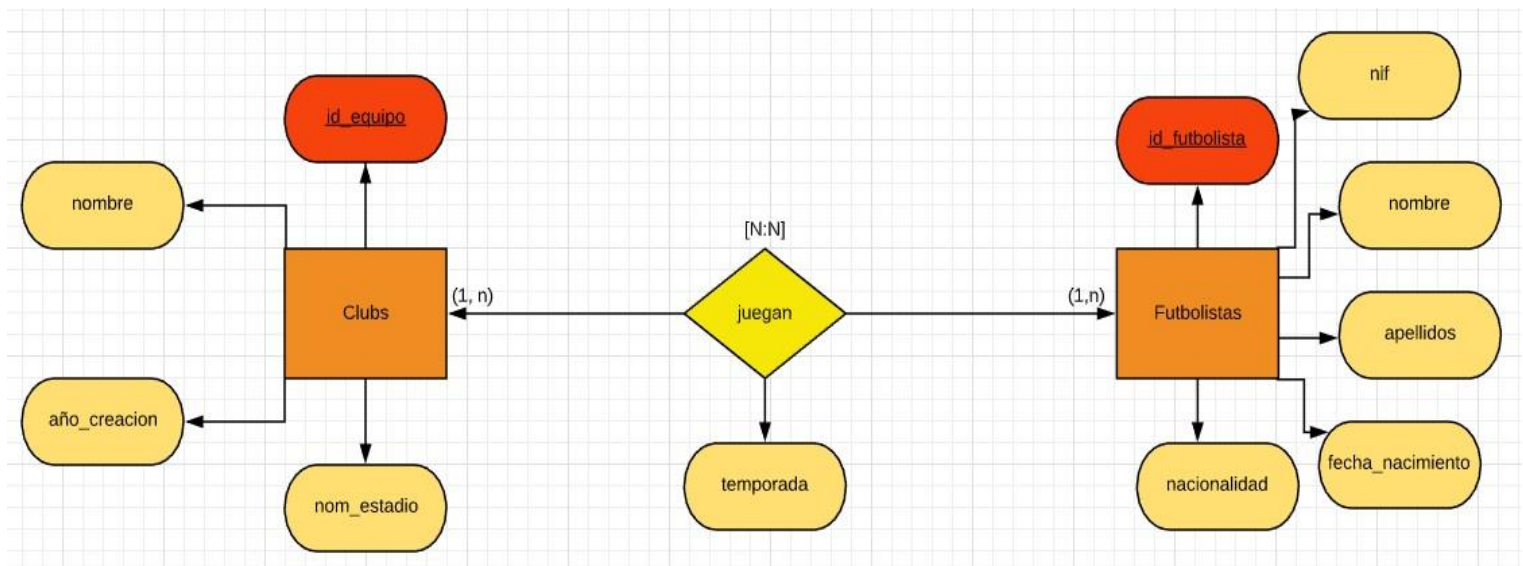


Diagrama Entidad-Rrelación

Una vez realizado el estudio y análisis de nuestro problema, se concluye en el esquema presentado. Se trata de un modelo bastante sencillo, compuesto por dos entidades CLUBS y FUTBOLISTAS, con una relación N:N entre ellas, que vendrá determinada por una temporada en concreto.

Debido a este tipo de relación, se crea una nueva entidad que en nuestro caso hemos llamado HISTÓRICO, y restringirá las relaciones entre clubes y futbolistas. A continuación, mediante el modelo relacional podemos verlo más claramente.



Como vemos en este modelo, la relación N:N ha generado la tabla Histórico.

Se observa como la tabla Clubs tiene como Primary Key una ID que se autogenera y es incremental, además de añadirle el nombre del club como UNIQUE para evitar dos equipos con el mismo nombre. Dentro de la tabla Futbolistas, al igual que en Clubs, tiene una ID autogenerada, con el NIF siendo tratado como UNIQUE para evitar repeticiones del atributo.

Por último, la tabla creada en la relación se compone de una triple Primary Key (id_club – id_futbolista – temporada). Hemos decidido tener en cuenta el atributo temporada como Primary ya que así podemos asignar un jugador a un equipo en distintas temporadas. Además tiene como Foreign Key las id de las otras dos tablas.

Script SQL:

Definición de las tablas:

```
CREATE TABLE Clubs (  
id_club INT (11) NOT NULL AUTO_INCREMENT,  
nombre VARCHAR (100) NOT NULL UNIQUE,  
fecha_creacion INT (11),  
nom_estadio VARCHAR (100),  
PRIMARY KEY (id_club)  
)
```

```
CREATE TABLE Futbolistas(  
id_futbolista INT (11) NOT NULL AUTO_INCREMENT,  
nif VARCHAR (9) NOT NULL UNIQUE,  
nombre VARCHAR (60),  
apellidos VARCHAR (120),  
fecha_nacimiento DATE,  
nacionalidad VARCHAR (60),  
PRIMARY KEY (id_futbolista)  
)
```

```
CREATE TABLE Historico(  
id_club INT (11) NOT NULL,  
id_futbolista INT (11) NOT NULL,
```

```
temporada INT (11) NOT NULL,  
PRIMARY KEY (id_club, id_futbolista, temporada),  
FOREIGN KEY (id_club) REFERENCES Clubs(id_club),  
ON DELETE CASCADE  
ON UPDATE CASCADE  
FOREIGN KEY (id_futbolista) REFERENCES Futbolistas(id_futbolista)  
ON DELETE CASCADE  
ON UPDATE CASCADE  
)
```

Control de la base de datos:

Para proteger la integridad y coherencia de los datos de nuestra base de datos se han incluido una serie de rutinas, principalmente procedimientos y disparadores, que se encargan de que no existan problemas a la hora de realizar los registros. Los procedimientos están centrados en el CRUD. Además de los procedimientos indicados, tenemos dos más para realizar las búsquedas concretas, tanto de los equipos en los que juega un jugador determinado y viceversa. Son los siguientes:

DELIMITER \$\$

```
CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `insertClub`(IN `vnombre` VARCHAR(60), IN  
`vfecha_creacion` INT, IN `vnom_estadio` VARCHAR(60))
```

NO SQL

BEGIN

INSERT INTO Clubs

(nombre, fecha_creacion, nom_estadio)

VALUES

(vnombre, vfecha_creacion, vnom_estadio);

END\$\$

DELIMITER ;

DELIMITER \$\$

```
CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `updateClub`(IN `vid_club` INT, IN `vnombre`  
VARCHAR(60), IN `vfecha_creacion` INT, IN `vnom_estadio` VARCHAR(60))
```

NO SQL

BEGIN

```
UPDATE Clubs SET nombre = vnombre, fecha_creacion = vfecha_creacion,  
nom_estadio = vnom_estadio WHERE id_club = vid_club;  
END$$  
DELIMITER ;
```

DELIMITER \$\$

```
CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `deleteClub`(IN `vclub` INT, IN `vnombre`  
VARCHAR(60))
```

NO SQL

BEGIN

```
DELETE FROM Clubs WHERE id_club = vclub and nombre = vnombre;
```

END\$\$

DELIMITER ;

DELIMITER \$\$

```
CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `insertFutbolista`(IN `vnif` VARCHAR(60), IN  
`vnombre` VARCHAR(60), IN `vapellidos` VARCHAR(60), IN `vfecha_nacimiento` DATE, IN  
`vnacionalidad` VARCHAR(60))
```

NO SQL

BEGIN

```
INSERT INTO Futbolistas
```

```
(nif, nombre, apellidos, fecha_nacimiento, nacionalidad)
```

VALUES

```
(vnif, vnombre, vapellidos, vfecha_nacimiento, vnacionalidad);
```

END\$\$

DELIMITER ;

DELIMITER \$\$

```
CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `updateFutbolista`(IN `vid_futbolista` INT, IN  
`vnif` VARCHAR(60), IN `vnombre` VARCHAR(60), IN `vapellidos` VARCHAR(60), IN  
`vfecha_nacimiento` DATE, IN `vnacionalidad` VARCHAR(60))
```

NO SQL

BEGIN

```
UPDATE Futbolistas SET nif = vnif, nombre = vnombre,
```

```
apellidos = vapellidos, fecha_nacimiento = vfecha_nacimiento, nacionalidad = vnacionalidad WHERE  
id_futbolista = vid_futbolista;
```

END\$\$

DELIMITER ;

DELIMITER \$\$

CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `deleteFutbolista`(IN `vfutbolista` INT)

NO SQL

BEGIN

DELETE FROM Futbolistas WHERE id_futbolista = vfutbolista;

END\$\$

DELIMITER ;

DELIMITER \$\$

CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `insertHistorico`(IN `vid_club` INT, IN
`vid_futbolista` INT, IN `vtemporada` INT)

NO SQL

BEGIN

INSERT INTO Historico

(id_club, id_futbolista, temporada)

VALUES

(vid_club, vid_futbolista, vtemporada);

END\$\$

DELIMITER ;

DELIMITER \$\$

CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `deleteHistorico`(IN `vid_futbolista` INT, IN
`vtemporada` INT)

NO SQL

BEGIN

DELETE FROM Historico WHERE id_futbolista = vid_futbolista AND temporada = vtemporada;

END\$\$

DELIMITER ;

DELIMITER \$\$

CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `buscarEquipos`(IN `vnombre` VARCHAR(60))

NO SQL

BEGIN

SELECT Clubs.id_club, Clubs.nombre, Clubs.fecha_creacion, Clubs.nom_estadio, Historico.temporada


```
FROM Historico JOIN Futbolistas ON Historico.id_futbolista=Futbolistas.id_futbolista
JOIN Clubs ON Historico.id_club=Clubs.id_club
WHERE Futbolistas.nombre=vnombre;
END$$
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE DEFINER=`JoseLFP`@`%` PROCEDURE `buscarJugadores`(IN `vnomclub`
VARCHAR(60))
```

```
NO SQL
```

```
BEGIN
```

```
SELECT Futbolistas.id_futbolista, Futbolistas.nif, Futbolistas.nombre, Futbolistas.apellidos,
Futbolistas.fecha_nacimiento, Futbolistas.fecha_nacimiento, Futbolistas.nacionalidad,
Historico.temporada
```

```
FROM Historico JOIN Futbolistas ON Historico.id_futbolista=Futbolistas.id_futbolista
```

```
JOIN Clubs ON Historico.id_club=Clubs.id_club
```

```
WHERE Clubs.nombre=vnomclub;
```

```
END$$
```

```
DELIMITER ;
```

Disparadores:

Control sobre la tabla Histórico:

No permite registrar futbolistas en dos equipos distintos en la misma temporada.

```
CREATE TRIGGER `tempRepetida` BEFORE INSERT ON `Historico`
```

```
FOR EACH ROW IF EXISTS
```

```
(SELECT temporada, id_futbolista FROM Historico WHERE new.temporada =temporada AND
new.id_futbolista=id_futbolista)
```

```
THEN
```

```
SIGNAL SQLSTATE VALUE '45000'
```

```
SET MESSAGE_TEXT = '[table:Historico] - temporada repetida';
```

```
END IF
```

Registro de todas las relaciones que se llevan a cabo entre clubes y futbolistas. Realizando un Insert “duplicado” para llevar el registro histórico.

```
CREATE TRIGGER `registro` AFTER INSERT ON `Historico`  
FOR EACH ROW INSERT INTO RegistroHistorico  
(id_club, id_futbolista, temporada)  
VALUES  
(new.id_club, new.id_futbolista, new.temporada)
```

3. DOCUMENTACIÓN DEL SISTEMA

3.1. MANUAL DE INSTALACIÓN Y CONFIGURACIÓN DE LA APLICACIÓN

Para poder ejecutar la aplicación necesitamos tener instalados en nuestra computadora un JRE. Éste puede ser descargado a través de la página oficial de Oracle de forma gratuita.

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

En lo referente a la base de datos, tenemos dos opciones, guardar los datos en local o en un servidor externo.

3.2. MANUAL DE USUARIO

Al iniciar la aplicación y tras visualizar la Portada, el usuario tendrá la opción de realizar gestiones sobre clubes o futbolistas.



Cuando nos dirigimos a Gestión de clubes la interfaz se divide entre las Acciones de búsqueda y Detalles del registro.

Nombre	Fecha Creacion	Nombre Estadio
Real Betis Balompié	1907	Benito Villamarín
Sevilla F.C.	1905	Sánchez Pizjuan
F.C. Barcelona	1899	Camp Nou
Real Madrid C.F.	1902	Santiago Bernabeu
Valencia C.F.	1915	Mestalla
RCD Espanyol	1909	RCDE Stadium

Acciones de búsqueda

Buscar:

Consultar los Clubes de un futbolista:


Nombre:  **Buscar**


Detalles del registro

Nombre del Club:

Año de creación:


Nombre del estadio:

 **Nuevo**  **Modificar**  **Eliminar**

 **Limpiar**  **Asociar**

 **Mostrar tabla**

 **PDF**



Dentro de los Detalles del Registro, si el usuario quiere insertar un “Nuevo” club tan solo deberá rellenar los 3 campos correspondientes.

En el caso de “Eliminar/Modificar”, se requiere que previamente seleccione de la tabla el club concreto y posteriormente se cambian los campos que desee, previa confirmación del sistema, siempre que los campos sean válidos. El botón de “Limpiar” vaciará todos los cajones de texto.


Listado de todos los Cubles participantes en la LFP

Nombre	Fecha Creacion	Nombre Estadio
Real Betis Balompié	1907	Benito Villamarín
Sevilla F.C.	1905	Sánchez Pizjuan
F.C. Barcelona	1899	Camp Nou
Real Madrid C.F.	1902	Santiago Bernabeu
Valencia C.F.	1915	Mestalla
RCD Espanyol	1909	RCDE Stadium

Acciones de búsqueda

Buscar:

Consultar los Clubes de un futbolista:


Nombre:  **Buscar**



Detalles del registro


Nombre del Club:


Año de creación:


Nombre del estadio:

 **Modificar**  **Eliminar**

 **Limpiar**  **Asociar**

 **Mostrar tabla**

 **PDF**



Selección de una Opción

¿Desea modificar los datos del club?

Dentro de las Acciones de búsqueda, tan solo tiene que ir escribiendo en el cajón cualquier dato de los clubes para que dinámicamente vaya filtrándose hasta llegar al resultado deseado.

Además, existe la posibilidad de *Consultar los clubes de un futbolista* mediante un desplegable donde están todos los jugadores registrados en la BD. Si se le da a “Buscar” se generará una nueva tabla con los equipos en los que ha jugado dicho futbolista.

Nombre	Fecha Creacion	Nombre Estadio
Real Betis Balompié	1907	Benito Villamarin
Sevilla F.C.	1905	Sánchez Pizjuan
F.C. Barcelona	1899	Camp Nou
Real Madrid C.F.	1902	Santiago Bernabeu
Valencia C.F.	1915	Mestalla
RCD Espanyol	1909	RCDE Stadium

Acciones de búsqueda

Buscar:

Consultar los Clubes de un futbolista:

Nombre: Seleccione un futbolista

- Seleccione un futbolista
- Joaquín
- Diego
- Karim
- Lionel
- Ferran
- Jose Ángel
- Raul

Detalles del registro

Nombre del Club:

Año de creación:

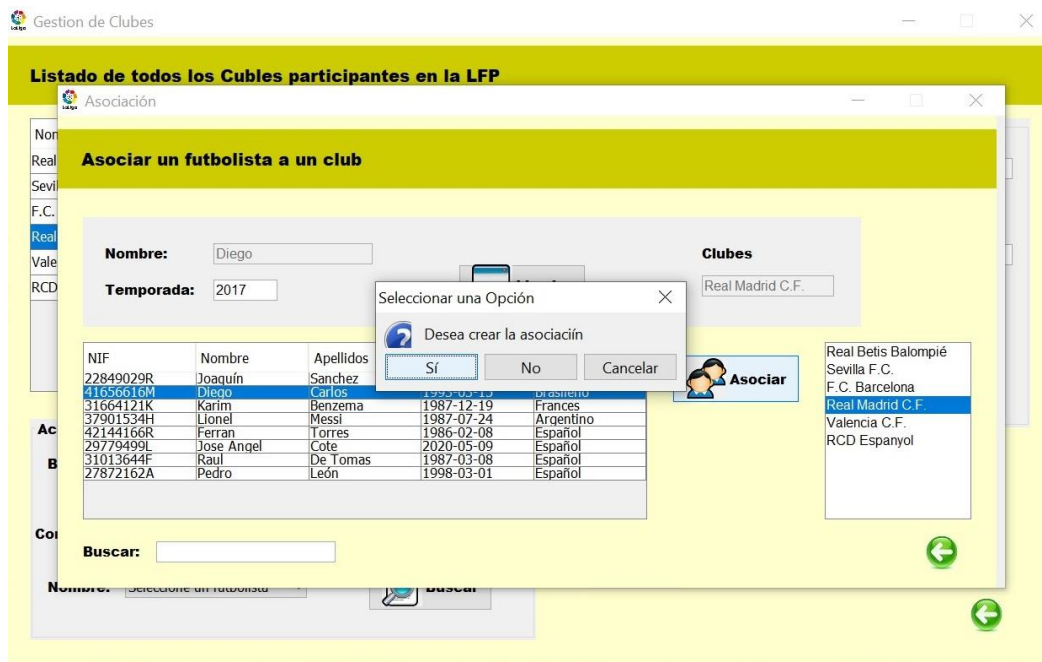
Nombre del estadio:

Acciones: Nuevo, Modificar, Eliminar, Limpiar, Asociar, Mostrar tabla, PDF

El botón “Mostrar tabla” realiza una actualización para que se muestren los clubes. Y el botón “PDF” da la opción de guardar la tabla en dicho formato. Por último, el botón de “Asociar” nos abre una nueva interfaz en la que se asocia un futbolista a un club.

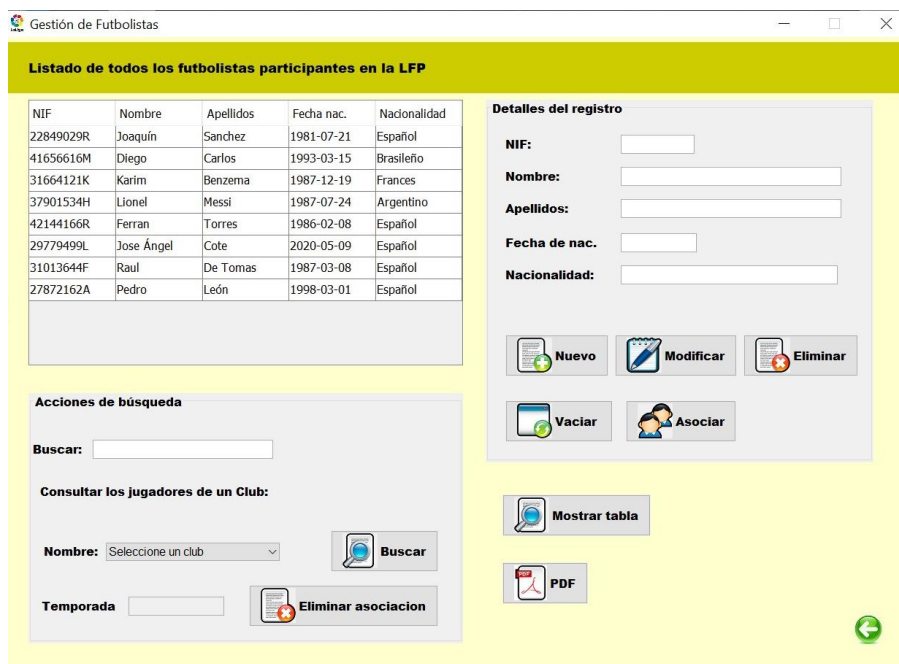
Desarrollo de Aplicaciones Multiplataforma. Proyecto 3

Dentro de la pantalla Asociación, el usuario tiene un buscador para poder filtrar al jugador sobre el que quiera realizar la asociación. Y a su derecha una lista con los clubes disponibles.



Para asociar un futbolista a un club, tendrá que seleccionar en la tabla el jugador deseado y realizar lo mismo en la lista de equipos y por último indicar la temporada. Finalizará cuando pulsa “Asociar”.

Cuando nos dirigimos a futbolistas, la interfaz se divide al igual que clubes, en Acciones de búsqueda y los Detalles del registro.



Dentro de la gestión de futbolistas, tenemos los mismos botones que en la interfaz clubes, con excepción de uno nuevo que es “Eliminar asociación”. Para realizar esto, el usuario tendrá que buscar previamente en el desplegable *Consultar futbolistas de un club*, generándose una nueva tabla que mostrará los datos personalizados. Posteriormente, tendrá que seleccionar al jugador a través del buscador o en la propia tabla, pinchando sobre él para que se completen los datos en todos los campos de texto, y por último pulsar el botón de “Eliminar asociación”.

Se observa en esta imagen el funcionamiento del buscador, indicando solo los jugadores que son españoles.

NIF	Nombre	Apellidos	Fecha nac.	Nacionalidad
22849029R	Joaquín	Sanchez	1981-07-21	Español
42144166R	Ferran	Torres	1986-02-08	Español
29779499L	Jose Ángel	Cote	2020-05-09	Español
31013644F	Raul	De Tomas	1987-03-08	Español
27872162A	Pedro	León	1998-03-01	Español

4. CONCLUSIONES FINALES

4.1. GRADO DE CUMPLIMIENTO DE LOS OBJETIVOS FIJADOS

El presente proyecto es una aplicación Java que reúne todos los requisitos mínimos presentes en la problemática propuesta. Es una aplicación funcional, realizando los requerimientos de un CRUD básico.

El usuario es capaz de leer los datos desde diferentes interfaces, tablas o listas, permitiendo insertar, modificar o eliminar dichos datos. Esta aplicación se conecta a una base de datos MYSQL y tiene además una funcionalidad extra que es la de generar un fichero pdf con la información mostrada en la tabla que desee.

4.2. PROPUESTA DE MODIFICACIONES O AMPLIACIONES FUTURAS DEL SISTEMA IMPLEMENTADO

Dentro de este proyecto se pueden observar varias mejoras. Por una parte, se puede añadir la gestión de distintos usuarios con roles definidos (administrador o usuario sin privilegios). También es posible la mejora del uso de las listas o combobox, que si bien son un componente muy potente, tienen una profundidad enorme. Además, se puede optimizar la forma en la que se estructura el código para que la aplicación tenga mayor fluidez.

Sería interesante poder ampliar y permitir más personalización a la hora de realizar el informe o trabajar con distintos archivos, pudiendo incluso importar datos desde un archivo externo. Por último, se puede tratar de perfeccionar la interfaz gráfica, que aun siendo amigable para el usuario, tenemos mucho margen de mejora para que el flujo de trabajo sea más intuitivo.

5. BIBLIOGRAFÍA

Apuntes y documentación de la asignatura de Programación

Apuntes y documentación de la asignatura de Bases de Datos.

Apuntes y documentación de la asignatura de Entornos de Desarrollo.

Sitio web <https://stackoverflow.com/>

Sitio web: <https://community.oracle.com/>

Sitio web: <https://github.com/>