



## **CICLO FORMATIVO DE GRADO SUPERIOR DESARROLLO DE APLICACIONES MULTIPLATAFORMA**

<b>FCB FEMENINO</b>
---------------------

Antonio De Llamas Oliva

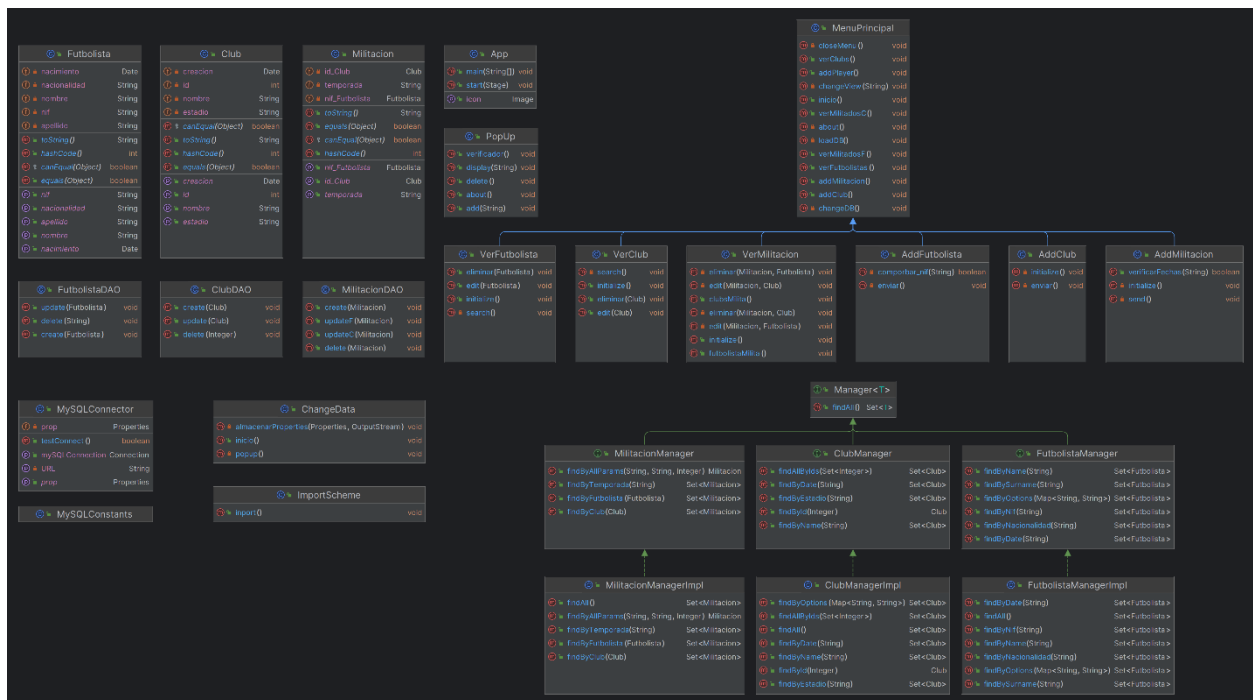
SEVILLA, 2023

<b>1. ESTUDIO DEL PROBLEMA Y ANÁLISIS DEL SISTEMA</b>	<b>3</b>
a. INTRODUCCIÓN	3
b. FUNCIONES Y RENDIMIENTOS	3
c. OBJETIVOS	4
d. MODELADO DE LA SOLUCIÓN	4
i. RECURSOS HUMANOS	4
ii. RECURSOS HARDWARE	4
iii. RECURSOS SOFTWARE	4
<b>2. EJECUCION DE LA PRACTICA</b>	<b>5</b>
a. ELABORACIÓN	5
b. IMPLEMENTACIÓN	6
<b>3. DOCUMENTACION DEL SISTEMA</b>	<b>7</b>
a. MANUAL DE INSTALACION Y CONFIGURACION	7
b. MANUAL DE USUARIO	7
<b>4. CONCLUSIONES</b>	<b>9</b>
a. GRADO DE CUMPLIMIENTO DE OBJETIVOS	9
b. PROPUESTAS DE MODIFICACIONES O AMPLIACIONES	9
<b>5. BIBLIOGRAFIA</b>	<b>10</b>

## 1. ESTUDIO DEL PROBLEMA Y ANÁLISIS DEL SISTEMA

### a. INTRODUCCIÓN

El equipo FC Barcelona Femenino necesita una aplicación para gestionar sus jugadoras que han militado en el club a lo largo de la historia. Esta idea fue innovadora y a la Liga de Fútbol Profesional decidieron de hacerla extensiva al resto de clubs de primera y segunda división.



### b. FUNCIONES Y RENDIMIENTOS

- Crear una futbolista.
- Crear un club.
- Asignar una militación.
- Listar futbolistas.
- Listar los clubs.
- Listar las futbolistas que han militado en un club.
- Listar los clubs en los que ha militado una futbolista.
- Eliminar una futbolista.
- Eliminar un club.
- Eliminar una militación.

**c. OBJETIVOS**

El objetivo principal de la aplicación es generar un sistema en el cual podamos manejar las futbolistas de la liga, los clubs y las militaciones que estas realizan a lo largo de las temporadas.

**d. MODELADO DE LA SOLUCIÓN**

**i. RECURSOS HUMANOS**

- Diseñador SQL
- Desarrollador Java
- Diseñador de visualización FXML
- Code tester

**ii. RECURSOS HARDWARE**

Requisitos mínimos:

- CPU: Intel Core I7 3537U o superior.
- RAM: 2GB o más.
- Gráfica: Intel HD 530 o superior.
- HDD: 50MB (Aplicación) + almacenamiento SQL

Requisitos recomendados:

- CPU: Intel Core I5 6600 o superior.
- RAM: 4GB o más.
- Gráfica: Intel HD 530 o superior.
- HDD: 50MB (Aplicación) + almacenamiento SQL

**iii. RECURSOS SOFTWARE**

- Windows / Linux
- Java JDK 17
- JavaFX 17
- MySQL Server 8.0 o MariaDB 10.11.3

## 2. EJECUCION DE LA PRACTICA

### a. ELABORACIÓN

Al comenzar el proyecto, el primer paso fue tomar notas sobre los requisitos y elementos necesarios para la aplicación. Luego, se procedió a estructurar las tablas necesarias en la base de datos.

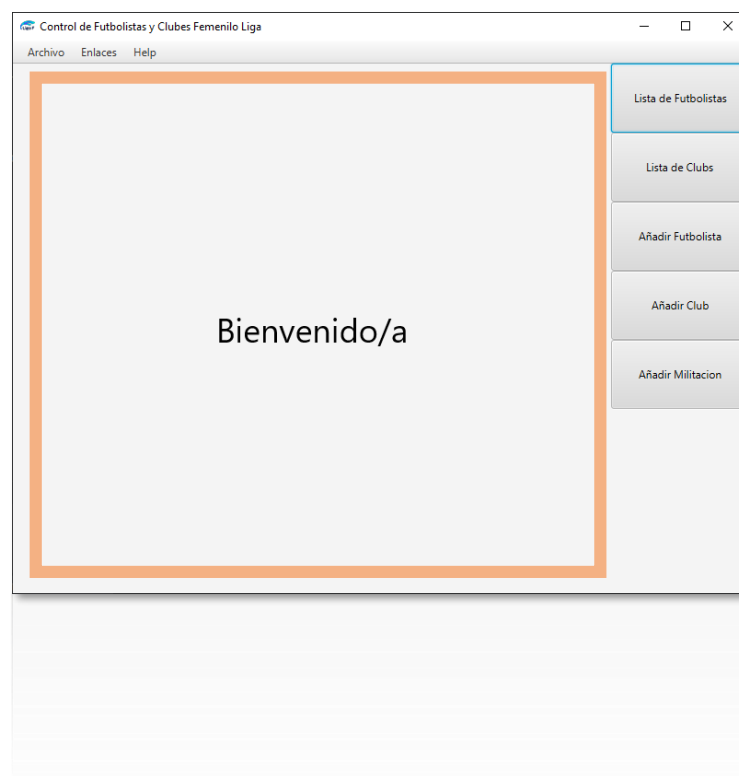
Una vez decidido el diseño de la base de datos, se creó la base de la aplicación desde cero utilizando Maven para gestionar las dependencias, como Lombok, JavaFX y el conector Java de MySQL.

La inicialización del proyecto incluyó la creación de paquetes para diferentes aspectos del programa, como DTO (Objetos de Transferencia de Datos), DAO (Objetos de Acceso a Datos), Persistence y Controladores para las visualizaciones FXML.

Después de definir los paquetes, se comenzó a crear las clases para la definición de objetos (DTO) y las clases relacionadas con la manipulación de objetos y la base de datos. Los controladores se definieron a medida que se crearon las visualizaciones, para manejar las acciones de los botones y la carga de información.

Una vez que se definieron las clases principales, se procedió a investigar y crear la visualización principal, que serviría como base para el desarrollo completo de la aplicación. Esta visualización se creó utilizando SceneBuilder, que permite diseñar archivos FXML para generar las visualizaciones de la aplicación. Luego, se generó una clase para controlar la visualización, donde se definieron las acciones de los botones y la carga de datos en una tabla.

Después de decidir el diseño final de la pantalla principal, se intentó mantenerlo para ir cambiando las visualizaciones en la zona central de la aplicación (sección naranja).



La investigación exhaustiva para encontrar un método que permitiera cambiar la visualización del área común no resultó ser tan difícil. Sin embargo, el verdadero problema surgió cuando fue necesario cambiar la visualización nuevamente.

Una vez encontrado el método para realizar esta acción, resultó interesante la disposición de carga de los métodos en los diferentes controladores cuando era necesario acceder a un objeto que perteneciera a su clase superior. Después de cambiar la visualización, parecía que la clase padre no permitía el acceso (a pesar de ser público) a los métodos directamente disponibles en el controlador. En este caso, era necesario acceder desde la propia visualización después de su carga completa y llamar dos veces a su padre para poder acceder a los objetos de forma relativa. Una vez en este punto, se pudo combinar acciones externas a la visualización general y activarlas.

```
Button b = (Button) windowsGeneric.getParent().getParent().lookup( s: "#addPlayer");  
b.fire();
```

Recientemente hemos visto un ejemplo de lo que mencioné, ya que estas acciones ocurren principalmente al agregar un nuevo elemento a la base de datos (como Añadir Futbolista y Añadir Club). Sin embargo, estas acciones solo se llevan a cabo después de aceptar o rechazar el PopUp diseñado para preguntar si se desea crear otra entrada o no.

Además, me di cuenta de la necesidad de generar PopUps que muestren mensajes de advertencia al ingresar datos incorrectos o informen sobre algún error cuando ocurra un problema en el código, como la carga de una visualización o la conexión a la base de datos.

Después de completar todas las acciones principales del proyecto y verificar que funcionaran correctamente, se realizó el ajuste en el manejo de excepciones para mostrar un PopUp informativo que indique qué tipo de error ha ocurrido, sin mostrar el código del error.

También se agregó un PopUp que contiene información básica del desarrollador del programa (About) y funciones para establecer la conexión a la base de datos e importar las tablas utilizadas por el programa. Estas dos últimas acciones se agregaron de forma opcional para demostrar habilidades más allá de las requeridas en la investigación del proyecto en su totalidad. Sin embargo, a dos semanas de la entrega, se notificó a los estudiantes (que dejaron todo para el último día) que el programa ya no requería una interfaz gráfica, sino que debía ejecutarse desde la consola.

## **b. IMPLEMENTACIÓN**

La implementación de esta aplicación sigue un proceso en el cual se debe definir primero la base de datos a utilizar, ya sea MySQL o MariaDB. Se creará la base de datos que contendrá las tablas, así como el usuario y la contraseña para el control de los datos.

Una vez que la base de datos ha sido definida, ejecutaremos el programa y procederemos a configurar la conexión con la base de datos y la importación de las tablas.

A partir de ahí, podremos comenzar a utilizar la aplicación de manera fluida y sin problemas.

### **3. DOCUMENTACION DEL SISTEMA**

#### **a. MANUAL DE INSTALACION Y CONFIGURACION**

Descargar el archivo .zip y descomprimirlo, o clonar el repositorio desde GitHub utilizando el comando "git clone https://github.com/alnazo/ProyectoIntermodular.git". Colocar el proyecto en el directorio deseado.

Abrir el IDE (IntelliJ IDEA/NetBeans/Eclipse) y cargar la carpeta "Proyect" que se encuentra dentro del repositorio. Esto permitirá cargar el entorno de trabajo y detectar los paquetes Maven.

Una vez cargadas todas las dependencias del proyecto, acceder a la configuración de arranque de perfil. Configurar la clase "App" ubicada en el paquete "com.proyectointermodular" como clase de inicio.

Agregar los siguientes argumentos de carga de la máquina virtual de Java en la configuración de arranque:

```
--module-path ${PATH_TO_FX} --add-modules javafx.fxml,javafx.controls,javafx.graphics
```

En \${PATH\_TO\_FX}, indicar la ruta donde se encuentra la librería de JavaFX 17 o una versión superior.

Aplicar los cambios y guardar la configuración.

Ahora se puede ejecutar la aplicación desde el IDE.

#### **b. MANUAL DE USUARIO**

Al iniciar la aplicación, el primer paso es establecer la conexión con la base de datos MySQL o MariaDB. Esto se realiza seleccionando la opción "Help > Cambiar conexión BD" o presionando F2. Un mensaje emergente indicará si la conexión se estableció correctamente.

Una vez configurada la conexión a la base de datos, se debe importar el esquema de tablas que se utilizarán en el programa. Esto se hace seleccionando la opción "Help > Cargar Esquema BD" o presionando F3. Un mensaje emergente informará si la carga del esquema se realizó correctamente.

Una vez completados estos pasos, se puede acceder a las secciones de las listas de futbolistas y clubes, así como a la inserción de datos en cada sección.

Para añadir una futbolista, se debe seleccionar el botón "Añadir Futbolista" o presionar Ctrl+4. Aparecerá un formulario donde se debe ingresar la fecha en formato internacional (yyyy-MM-dd) y el DNI/NIF, que será verificado para evitar errores de identificación. Después de ingresar los datos, al hacer clic en el botón "Guardar", se mostrará un mensaje emergente que preguntará si se desea almacenar más información. Si no se desea agregar más información, se redirigirá a la visualización de la lista de futbolistas.

Para añadir un club, se debe seleccionar el botón "Añadir Club" o presionar Ctrl+5. Aparecerá un formulario con tres campos, donde la fecha de creación del club también se verifica en formato internacional (yyyy-MM-dd). Después de ingresar los datos y hacer clic en el botón "Guardar", se mostrará un mensaje emergente para confirmar si se desea agregar más información. Si no se desea agregar más información, se redirigirá a la visualización de la lista de clubes.

Para asociar futbolistas y clubes mediante la militación, se debe seleccionar el botón "Añadir Militacion". Aparecerá un campo donde se debe indicar la temporada de la militación, con una verificación que garantiza que los dos años ingresados sean consecutivos y que el primero sea menor que el segundo. Además, en la visualización se encontrarán dos tablas con información de las futbolistas y los clubes almacenados en la base de datos.

En las visualizaciones de listado de futbolistas y clubes, se disponen de métodos internos. Al hacer clic derecho en un registro, se abrirá un menú contextual que permite editar o eliminar el registro seleccionado.

Cabe destacar que también se encuentra disponible un botón "Clubs militados" o "Futbolistas que militan", dependiendo de la lista en la que se encuentre. Este botón muestra la lista de clubes en los que una futbolista ha militado (si se está en la lista de futbolistas) o la lista de futbolistas que han militado en un club (si se está en la lista de clubes). Para que esta visualización se cargue correctamente, es necesario seleccionar un registro de la lista antes de hacer clic en el botón, para obtener la información correspondiente a la jugadora o el club y su relación en la militación.



#### **4. CONCLUSIONES**

##### **a. GRADO DE CUMPLIMIENTO DE OBJETIVOS**

El proyecto cumple completamente su objetivo principal, permitiendo la introducci3n y modificaci3n de datos de futbolistas y clubes, as3 como la asociaci3n entre ellos. Tambi3n es capaz de generar listas de futbolistas asociados a un club y de mostrar los clubes y futbolistas disponibles.

Adem3s, se han incorporado opciones para cambiar la conexi3n a la base de datos, tanto si es MySQL como MariaDB, y se ha agregado la funcionalidad de cargar tablas vac3as para un reinicio del sistema.

##### **b. PROPUESTAS DE MODIFICACIONES O AMPLIACIONES**

Las posibles mejoras futuras de esta aplicaci3n incluyen:

- Mejorar la visualizaci3n de los elementos y permitir descargar las listas mostradas en diferentes visualizaciones en formato PDF.
- Explorar la posibilidad de cambiar el sistema de bases de datos de MySQL o MariaDB a otros sistemas NoSQL como MongoDB.

## 5. BIBLIOGRAFIA

- Realización de búsqueda de información desde:

[www.you.com](http://www.you.com) (Búsqueda de información entre las web de desarrolladores)

[stackoverflow.com](http://stackoverflow.com) (Foro de desarrollo)

[www.google.es](http://www.google.es) (Realización de búsquedas en inglés)

[chat.openai.com](http://chat.openai.com) (IA para comprobar acciones específicas y extrapolar la respuesta)

[www.youtube.es](http://www.youtube.es) (Videos explicativos sobre JavaFX)

[openjfx.io/javadoc/17](http://openjfx.io/javadoc/17) (Documentación oficial de Java FX 17)