Miguel Calvo Gómez

Correo: mcalvog03@educantabria.es

DNI: 72204281J

Curso: 2024/2025

Befootball

Proyecto de DAM

Índice

[1. Resumen 2](#_Toc195282100)

[2. Palabras clave 2](#_Toc195282101)

[3. Introducción 2](#_Toc195282102)

[4. Objetivos 3](#_Toc195282103)

[5. Análisis de contexto 3](#_Toc195282104)

[5.1 Análisis de contexto 3](#_Toc195282105)

[5.2 Innovación 4](#_Toc195282106)

[6. Diseño 5](#_Toc195282107)

[6.1 Diseño de la interfaz 6](#_Toc195282108)

[6.2 Casos de uso de los usuarios 12](#_Toc195282109)

[6.3 Modelo de datos 13](#_Toc195282110)

[6.4 Diagrama de clases 17](#_Toc195282111)

[6.5 Despliegue 20](#_Toc195282112)

[7. Planificación 21](#_Toc195282113)

[7.1 Diagrama de gant 21](#_Toc195282114)

[7.2 Definición de recursos y logística necesarios para cada actividad 22](#_Toc195282115)

[8. Implementación 22](#_Toc195282116)

[8.1 Implementación del servidor socket 22](#_Toc195282117)

[8.2 Implementación en PC 23](#_Toc195282118)

[8.3 Implementación de la API 28](#_Toc195282119)

[8.4 Implementación en Android 32](#_Toc195282120)

# Resumen

Este proyecto consta de un programa de ordenador y una aplicación móvil. En la versión de PC se podrán consultar todo tipo de datos de partidos, equipos, clasificaciones y ligas de futbol, además dependiendo del tipo de usuario que seas, en caso de ser desarrollador se podrá acceder a una versión que permitirá actualizar los datos como por ejemplo cuando se jueguen partidos, se funden nuevos equipos, inscripciones de nuevos jugadores, etc. En cuanto a la aplicación de móvil, esta solamente ofrecerá la opción de ver los resultados de los partidos por el momento.

La innovación que propone esta aplicación es la simplicidad y accesibilidad ya que con controles muy sencillos el usuario podrá acceder a la información que quiera de forma rápida y eficiente, también será posible cambiar a un estilo de interfaz más clásica en caso de que resulte incomodo el diseño de teletexto. Por último, la generación de resultados de partidos a tiempo real es algo que también caracterizará a esta aplicación.

# Palabras clave

Fútbol, resultados en vivo, partidos en directo, aplicación para móvil, aplicación para ordenador…

# Introducción

En la actualidad, la tecnología avanza rápidamente. Sin embargo, esto no siempre es beneficioso para todos los sectores de la población. En particular, muchas personas mayores encuentran dificultades al utilizar internet para consultar información sobre sus intereses, como los resultados de los partidos de fútbol. Las páginas web suelen estar saturadas de publicidad, requieren múltiples clics para encontrar la información deseada y, en muchos casos, presentan interfaces complejas que no están diseñadas pensando en usuarios con menos experiencia digital.

Este problema se ha hecho evidente al observar cómo muchas personas de edad avanzada prefieren preguntar a familiares o esperar a ver los resultados en la televisión antes que intentar buscarlos en internet.

Recordando la simplicidad y efectividad de sistemas como el teletexto, surge la idea de crear una solución moderna que mantenga esa facilidad de uso y accesibilidad.

Para resolver este problema, se desarrollará una aplicación de escritorio que permita visualizar resultados y estadísticas básicas de los partidos mediante una interfaz y controles intuitivos, estos controles podrán funcionar de forma habitual o mediante números como si de un teletexto se tratase.

También se desarrollará una aplicación para Android que mostrará solamente los resultados de los partidos de manera clara.

El objetivo de este proyecto es ofrecer una alternativa accesible y eficiente para que las personas mayores puedan seguir el fútbol sin complicaciones.

# Objetivos

El objetivo principal de esta aplicación es simplificar y mostrar la información básica acerca de competiciones de futbol, de esta forma el producto podrá alcanzar un público mayor.

Un objetivo secundario que se tiene en cuenta es que mediante de la opción de seguir a un equipo, el usuario quiera acceder más veces a la aplicación para conocer en qué situación esta su club en todo momento.

Otro objetivo es garantizar una experiencia sin publicidad invasiva ni contenido innecesario, diferenciándose de otras plataformas actuales que pueden resultar confusas o poco amigables para ciertos usuarios. Esto contribuirá a fidelizar a los usuarios y hacer que la aplicación sea percibida como una herramienta práctica y cómoda.

# Análisis de contexto

## 5.1 Análisis de contexto

En la actualidad la tecnología ha evolucionado rápidamente en comparación con décadas anteriores por lo que existen personas que se han resistido al cambio y ahora les puede costar acostumbrarse a las nuevas tecnologías. Por ejemplo, en el sector del deporte, concretamente en el futbol han surgido nuevas formas de conocer datos sobre distintas competiciones, pero puede resultar compleja la forma de obtener esta información, por esto es por lo que se ha decidido desarrollar este proyecto, el cual solucionará el exceso de información, publicidad y complejidad de otras aplicaciones.

Estas son algunas aplicaciones que ofrecen servicios similares mediante un análisis de competencia:

* OneFootball: es una web y aplicación para smartphones que permite ver resultados y datos en vivo de hasta 200 competiciones.
* Besoccer: es un sitio web que también cuenta con aplicación para smartphones que proporciona principalmente información de ligas europeas, aunque también algunas del resto del mundo.
* LiveScore: es una página web como las anteriores pero esta cuenta con menos competiciones ya que se centra en más deportes como hockey o baloncesto, por ejemplo.

A continuación, se muestra un análisis DAFO frente a la competencia:

Fortalezas:

* Accesibilidad al servicio desde ordenador y teléfono móvil.
* Resultados a tiempo real.
* Diseño intuitivo y retro, por lo que el público mayor le resultará familiar el uso de la aplicación.
* Poder agregar nuevos datos como ligas y equipos, etc. Sin necesidad de actualizar la aplicación.

Debilidades:

* Recursos limitados, puesto que todavía no se cuenta con una gran infraestructura de calidad.
* Falta de personal. Será necesario contratar personal para actualizar los resultados de los partidos.

Oportunidades:

* Crecimiento de la población de la tercera edad, lo que supone que aumenta el público al cual se dirige el producto.
* Crecimiento del consumo de aplicaciones deportivas, cada vez más gente quiere utilizar aplicaciones de deportes.

Amenazas:

* Competencia emergente.
* Baja retención de usuarios.
* Costes elevados de infraestructura.

## 5.2 Innovación

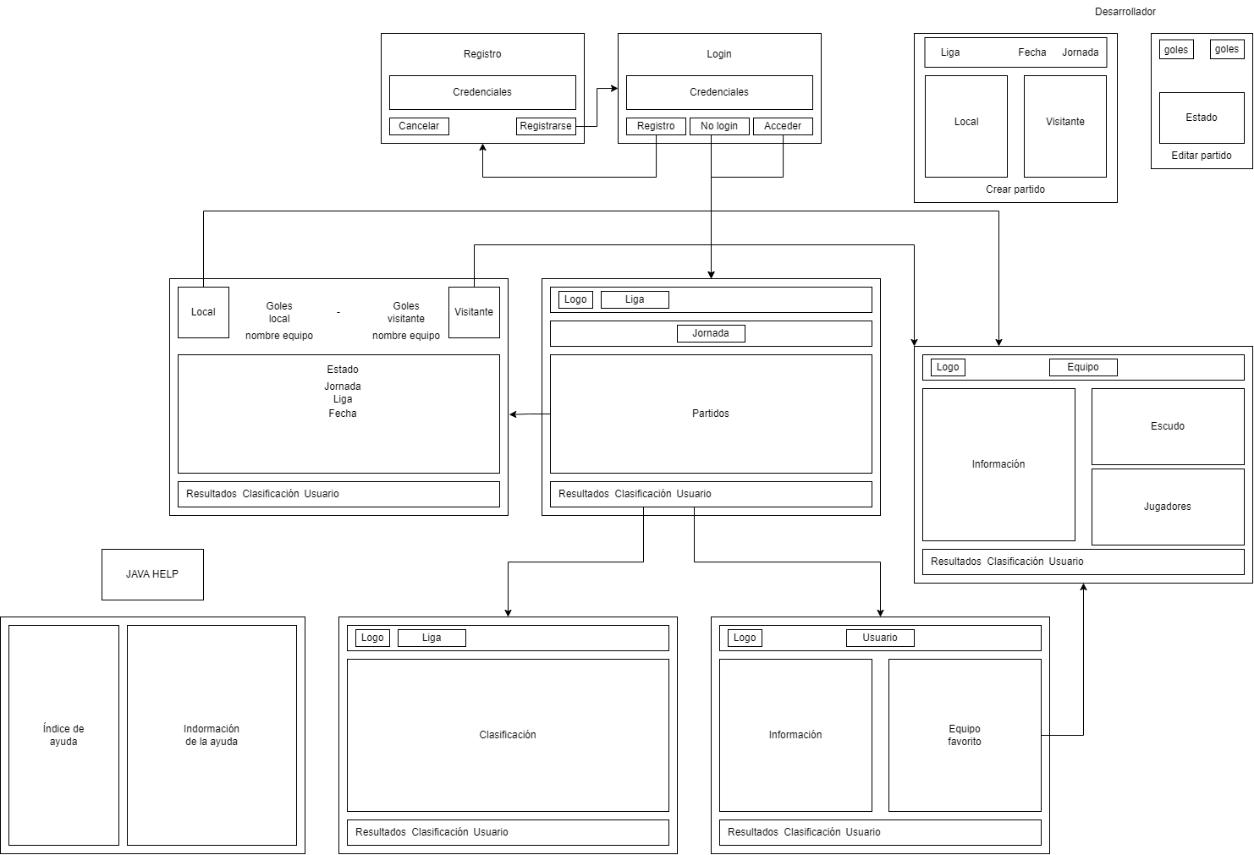
La innovación que tendrá este proyecto será principalmente su intuitiva interfaz debido a la simplicidad en cuanto a mostrar los datos y poseer un control mediante números como un teletexto o por controles habituales para navegar por los menús. También al existir un botón que permite cambiar el estilo de la interfaz, la accesibilidad se verá mejorada el cual es un aspecto que no se suele tener muy en cuenta.

Además, para que la aplicación sea capaz de obtener los resultados con mayor rapidez a tiempo real, habrá un grupo de personas encargadas de actualizar los resultados de forma manual al instante mediante la versión del programa para desarrolladores, está versión también permitirá establecer que usuarios pueden obtener el rol y otros ajustes especiales.

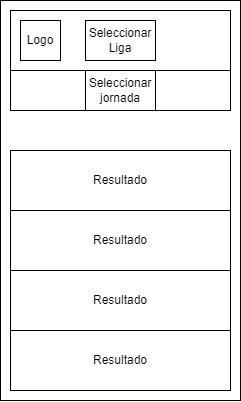
# Diseño

A continuación, se muestra un boceto de la interfaz que se plantea para ambas aplicaciones, las cuales presentan una navegación muy sencilla dirigida al público objetivo.

Aplicación de escritorio:



Aplicación para smartphone:



## Diseño de la interfaz

En la versión de la aplicación de escritorio, la primera ventana que aparecerá será un diálogo que mostrará la opción de acceder sin cuenta, registrarse y acceder con las credenciales introducidas por el usuario.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

En caso de registrarse aparecerá este diálogo.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Una vez iniciada sesión se mostrará el panel principal, pero dependiendo del tipo de usuario se omitirán algunos componentes como por ejemplo en caso de ser invitado no se mostrará la sección de usuario.

Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

Panel de clasificación que mostrará la tabla de la competición seleccionada en el comboBox superior.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Cuando se seleccione un partido del panel de resultados se cargará el siguiente panel con información de este. En las dos esquinas superiores aparecerán los escudos de los equipos que estén jugando. En la zona inferior hay un botón para editar el estado y el resultado que estará disponible solo para los desarrolladores.

|  |  |
| --- | --- |
| Texto  El contenido generado por IA puede ser incorrecto. | Interfaz de usuario gráfica, Texto, Aplicación  El contenido generado por IA puede ser incorrecto. |

El panel de usuario mostrará datos del usuario además de su equipo favorito el cual podrá editar desde un botón, en este menú deberás seleccionar la liga para cargar la lista de los equipos y posteriormente seleccionarlo.

|  |  |
| --- | --- |
|  |  |

El panel de equipo se mostrará cuando se pulse sobre un escudo en cualquier parte de la aplicación, en este panel aparecerá información sobre el equipo, como el estadio, su liga, etc, además se mostrará el escudo y una tabla con los jugadores, por último, en la zona superior derecha hay un botón solo visible para usuario y desarrolladores para agregar el equipo como favorito.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Los desarrolladores tendrán acceso a una ventana para establecer que rol puede tener cada usuario.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

También existe una opción para cambiar el estilo de la interfaz la cual servirá para mejorar la accesibilidad.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Por último, se ha implementado java help para mostrar una guía de cada sección de la aplicación.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

La aplicación de Android estará compuesta por dos items cargando el segundo mediante un reclycle view. El primer item (activity\_main.xml) tendrá dos toolbar, una para seleccionar la liga y otra para seleccionar la jornada dependiendo de la liga que se seleccionó anteriormente. El segundo item (item\_partidos.xml) cargará una lista de partidos de la liga y jornada seleccionada.

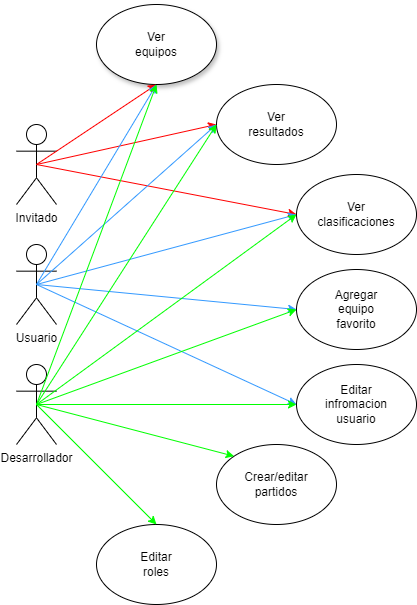
Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## Casos de uso de los usuarios

En la aplicación de escritorio constarán tres tipos de usuarios.

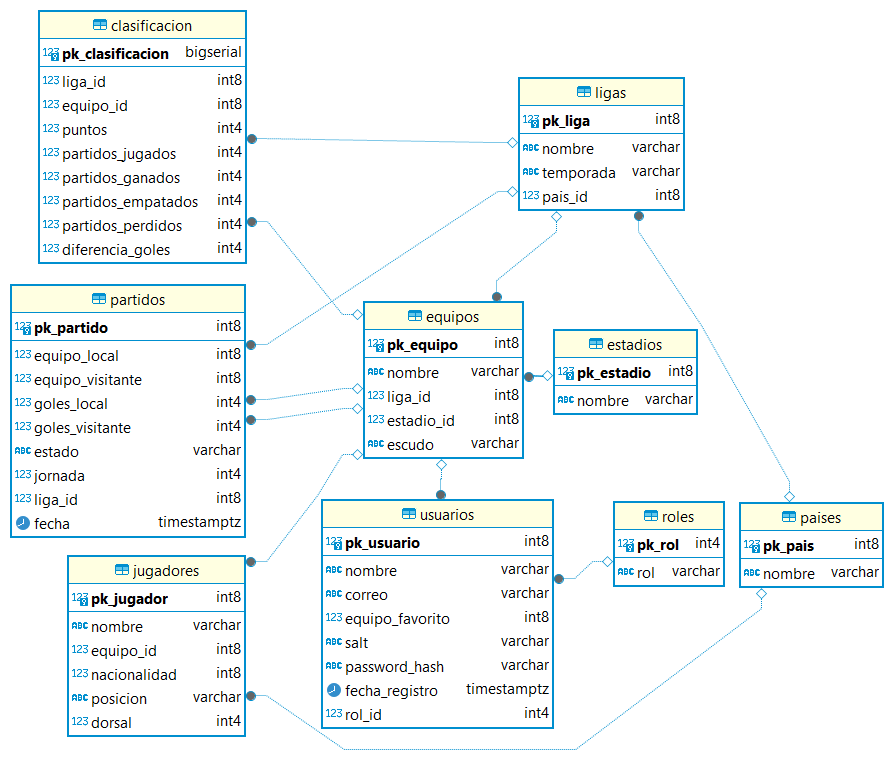
* Invitados: este tipo de usuario tendrá acceso solamente a un limitado contenido de la aplicación como la pestaña de usuario, además no poder agregar equipos como favorito.
* Usuarios (usuarios con una cuenta): los usuarios que dispongan de una cuenta podrán acceder a casi todas las funcionalidades de la aplicación además de poder agregar un equipo como favorito.
* Desarrollador: estos usuarios tendrán acceso a todas las funcionalidades de la aplicación, además podrán crear partidos nuevos, editar los resultados y decidir que nuevas cuentas pueden recibir el rol de desarrollador. Este tipo de cuenta estará solo disponible para los trabajadores de la empresa.



En la versión de smartphone no existirán usuarios, cualquiera podrá ver los resultados sin necesidad de una cuenta.

## Modelo de datos

Este proyecto utiliza una base de datos de PostgreSQL la cual será desplegada en una máquina virtual como servidor. La base de datos presenta el siguiente diagrama:



La tabla “paises” contiene varios nombres de países con un id que utilizarán las tablas “jugadores” para establecer la nacionalidad de cada jugador y la tabla “ligas” para decidir a qué país pertenece una liga.

La tabla “jugadores” almacena información de cada jugador, como su nombre, id del equipo al que pertenece, id de la nacionalidad, posición y dorsal.

La tabla “ligas” tiene la información del nombre de la liga, la temporada en la que se compite el torneo y por último el id del país al que pertenece.

La tabla “estadios” tan solo contiene el nombre de los estadios y será utilizada por la tabla “equipos” para establecer que equipos juegan en él.

La tabla “equipos” se utiliza para almacenar el nombre del equipo, el id de la liga a la que pertenece, el id de su estadio y el escudo el cual es el nombre del archivo de la imagen de este.

La tabla “usuarios” contine el nombre, el correo, un rol\_id para establecer los permisos que tiene en la aplicación, también se almacena el equipo favorito del usuario mediante la id de este, además esta tabla cuenta con una columna salt para guardar el salt de la contraseña y otra columna password\_hash para almacenar la contraseña codificada para tener mayor seguridad, por último, se guardará automáticamente la fecha y hora de cuando el usuario se da de alta.

La tabla “roles” almacena los nombres de los roles junto con un identificador que se le asociará al usuario, al tener los roles en una tabla separada permite añadir nuevos en un futuro.

La tabla “partidos” almacenará todos los partidos de todas las competiciones, para ello esta tabla cuenta con las siguientes columnas, equipo\_local utilizando el id del equipo que jugará como local, equipo\_visitante utilizando el id del equipo que jugará como visitante, goles\_local, es decir, los goles del equipo local, goles\_visitante, es decir, los goles del equipo visitante, estado significa si el partido ha comenzado o en qué momento está partido, jornada es el numero de la jornada del partido, liga\_id es el id de la liga al que pertenece el partido y fecha es la fecha y hora a la que se jugará el partido.

La tabla “clasificación” se generará automáticamente mediante un trigger en la tabla “partidos” el cual establecerá en id\_liga la liga al que pertenece el equipo, equipo\_id el id del equipo, los puntos obtenidos, los partidos jugados, ganados, empatados y perdidos, en la última columna se establecerá la diferencia de goles.

El trigger de la tabla partidos comprobará si se está creando un partido o se están actualizando los resultados, en estos casos, se hará una validación para que los goles no sean negativos. Después se comprobará el equipo con más goles y se le otrogarán 3 puntos al equipo que esté ganando, en caso de mismo número de goles se otorgará un solo punto a ambos equipos. A continuación, se calculará la diferencia de goles, para ello al valor de la columna de diferencia de goles se le añadirá la diferencia de los goles del partido en juego.

Texto

El contenido generado por IA puede ser incorrecto.

Por último, dependiendo de si se trata de un update (se revierten los datos anteriores) o un insert (se actualiza la información una vez para el equipo local y otra para el visitante), se establecerán los datos en la tabla para los dos equipos dependiendo del caso que suceda, además para las columnas de partidos ganados, empatados o perdidos se comprobarán los goles y en caso de que si se cumpla la condición se sumará 1 a la cantidad existente.

Texto

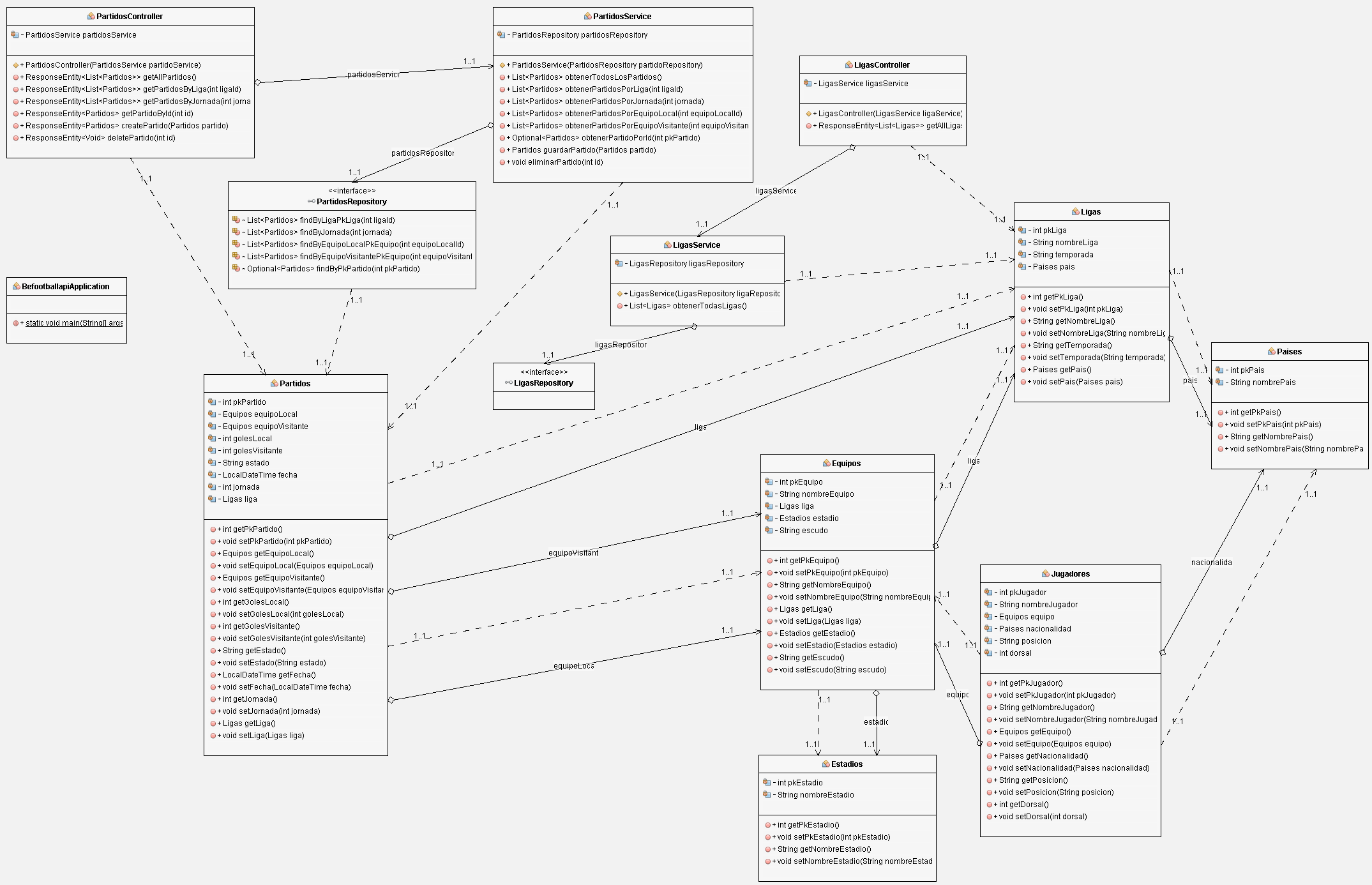
El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

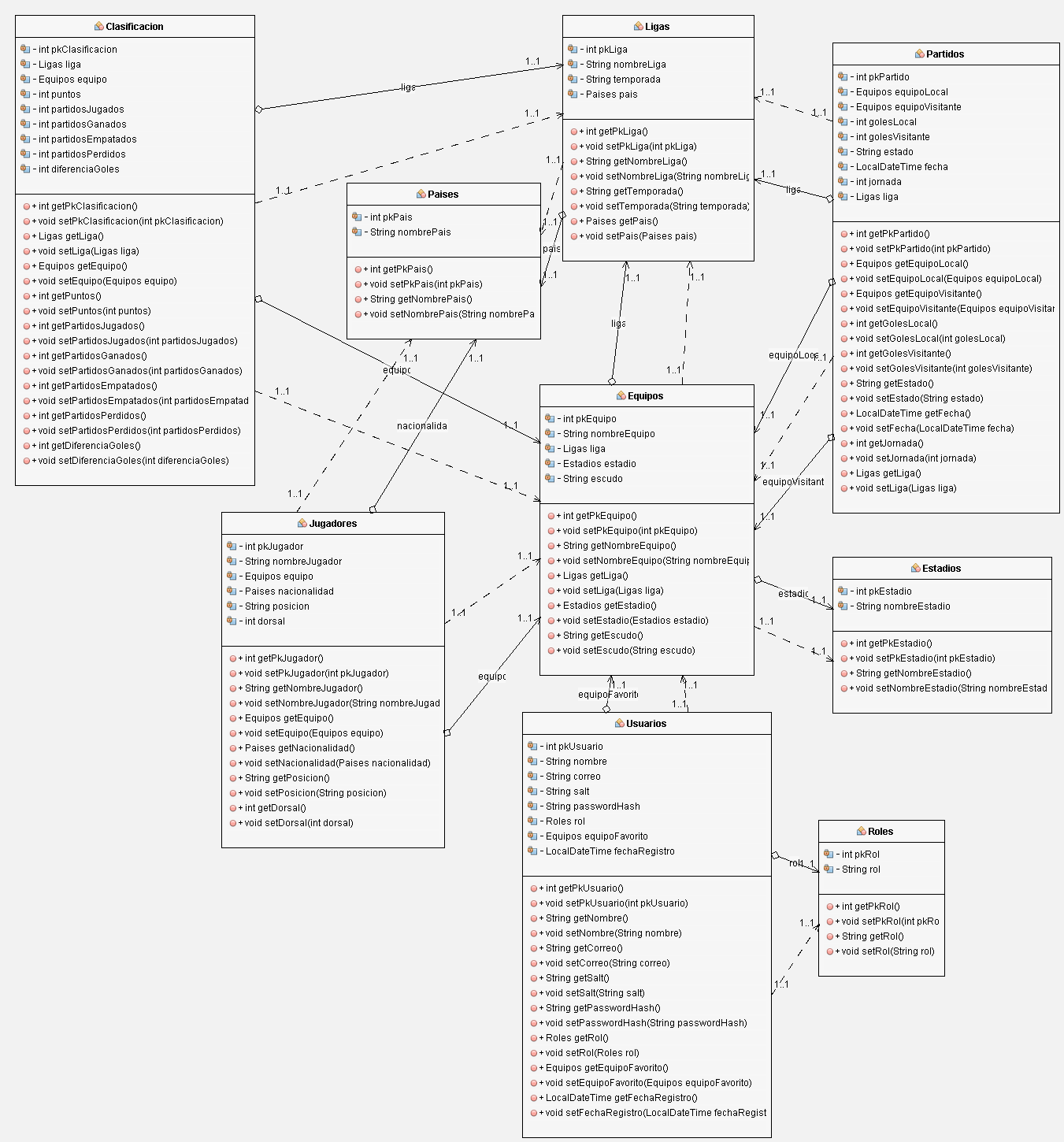
## Diagrama de clases

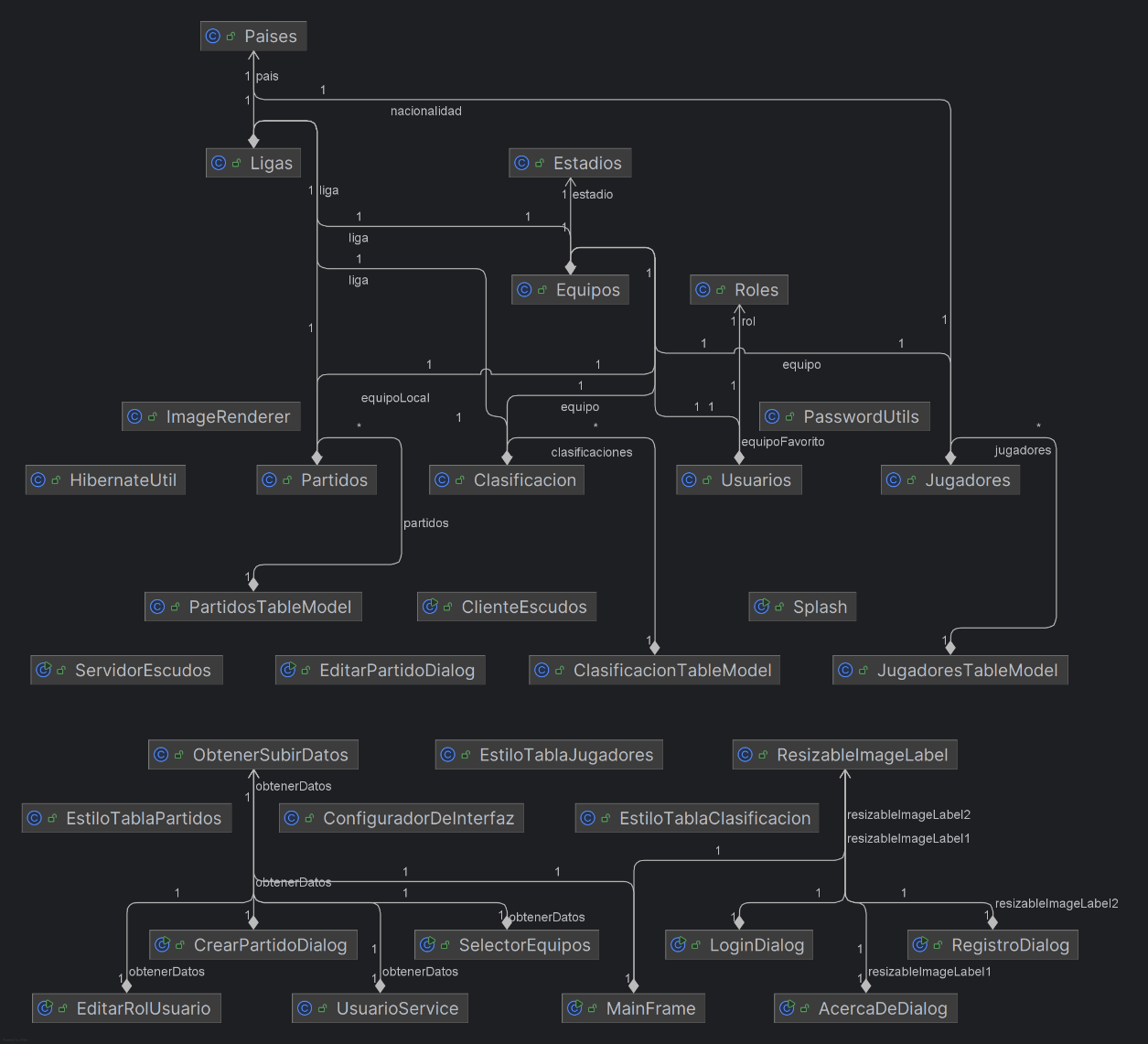
A continuación, se muestran los diagramas de clases de la API, la aplicación de smartphone y la aplicación de PC.





En el caso de la aplicación de pc por un lado tenemos el diagrama de los pojos y por otro el funcionamiento de la aplicación.





## Despliegue

Por el momento ambas aplicaciones se lanzarán en España, pero se prevé traducirlas para realizar un lanzamiento global.

En cuanto al alojamiento de la base de datos, la API, el servidor socket y el odoo, se ha optado por un servidor en la nube como AWS o Google Cloud, debido a la sencillez en la transición desde el entorno de desarrollo ya que los servicios de la aplicación ya se encontraba operativos en una máquina virtual, además este sistema permite tener un mayor control a la hora de personalización, escalabilidad y mantenimiento.

A la hora de analizar los costos, principalmente se tendrá en cuenta el gasto mensual en el mantenimiento del servidor y sus copias de seguridad. Además, se invertirá en un dominio personalizado para la página web de Odoo y la API, lo cual no solo aporta una imagen más profesional, sino que también permite establecer conexiones seguras mediante el protocolo HTTPS en lugar de HTTP. Esta capa adicional de seguridad es fundamental para proteger la información que se transmite entre el cliente y el servidor, especialmente al manejar datos sensibles de usuarios.

En cuanto a la seguridad, las contraseñas de los usuarios se encriptarán y posteriormente se guardarán de forma que nadie podrá acceder a la cuenta en caso de ataque al servidor.

Por último, el mantenimiento será sencillo, semanalmente se comprobarán actualizaciones de sistema, el estado de los servicios y se realizarán copias de seguridad diarias.

# Planificación

La planificación de este proyecto pretende establecer un orden en la realización y tiempo dedicado en cada parte del desarrollo. Para ello se ha creado un diagrama de gant al inicio del proyecto marcando además las entregas de las tareas antes de la presentación.

## Diagrama de gant

Gráfico, Gráfico en cascada

El contenido generado por IA puede ser incorrecto.

Las tareas que se han llevado a cabo para realizar este proyecto son las siguientes:

1. Establecer las bases del proyecto:
   * Planificación
   * Resumen
   * Introducción
   * Objetivos
   * Análisis del contexto
2. Desarrollo de las aplicaciones:
   * Modelo de datos
   * Diseño de las interfaces
   * Diagrama de clases
   * Implementación de las pantallas con funcionalidad parcial
3. Implementación de funcionalidad total:
   * Prueba de control y calidad
   * Puesta en marcha
4. Documentación:
   * Documentar el desarrollo de la aplicación.
   * Gestión económica/plan de empresa.

## Definición de recursos y logística necesarios para cada actividad

Para el desarrollo de la aplicación, ha sido necesario planificar cuidadosamente los recursos y la logística asociados a cada actividad. En cuanto a los recursos materiales, se ha utilizado equipamiento informático adecuado para el desarrollo de software, incluyendo ordenadores con características técnicas medias-altas y dispositivos Android para realizar pruebas de la versión móvil. Además, se cuenta con un servidor destinado al alojamiento de la base de datos, la API y los archivos multimedia, como los escudos de los equipos. Las herramientas de desarrollo empleadas han sido NetBeans para la creación de la API y la aplicación de escritorio, y Android Studio para la versión móvil, además de un sistema de control de versiones Git para gestionar el código fuente.

En cuanto a los recursos humanos, todo el desarrollo del sistema ha sido llevado a cabo por el autor del proyecto, incluyendo el diseño de la base de datos, la programación de la API, la aplicación de escritorio y la versión móvil. Sin embargo, para el correcto funcionamiento del sistema en producción y garantizar que los resultados de los partidos estén siempre actualizados, se tiene previsto contratar personal encargado de la gestión de resultados. Concretamente, se contempla la contratación de una persona por cada tres ligas, cuya tarea será introducir y actualizar los resultados en tiempo real a través de la versión de desarrollador de la aplicación de PC.

Desde el punto de vista logístico, se contemplan aspectos relacionados con la gestión de las cuentas de usuario. En este sentido, los datos proporcionados por los usuarios, como el correo electrónico, se utilizarán exclusivamente con fines informativos o administrativos. Por ejemplo, se podrán enviar notificaciones sobre nuevas funciones disponibles en la aplicación, actualizaciones importantes del sistema o información relevante sobre el servicio. La gestión de esta información cumplirá con las normativas vigentes en materia de protección de datos, garantizando la seguridad y privacidad de los usuarios registrados.

# Implementación

## Implementación del servidor socket

Para poder almacenar los escudos en el servidor y mandarlos al cliente se creará un servidor socket. Este sistema consta de una clase en la cual se le establece el puerto y la carpeta contenedora de los escudos, además varios métodos como run el cual interactuara con el cliente, recibiendo comandos y enviando archivos a través de flujos (inputStream, outputStream), bufferedReader junto InputStreamReader para leer las líneas de texto y PrintWriter.

El método handleDownload para enviar el archivo al cliente recibirá el nombre del archivo a enviar y el flujo OutputStream, dentro del método se comprobará si el archivo solicitado existe y posteriormente se leeran los bytes y se enviaran al cliente.

Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Implementación en PC

La base de datos creada será manipulada con Hibernate y JPA. Mediante Hibernate será posible mapear las tablas en un objeto, lo que facilitará la interacción entre java y la base de datos.

Para utilizar hibernate y postgre es necesario agregar las siguientes dependencias al proyecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Tras agregar las dependencias al pom.xml y realizar un clean & build, se generará el archivo hibernate.cfg.xml en el cual se establecerán los parametros de la conexión, como el driver (en este caso el de postgre), la url, el usuario y la contraseña de la base de datos, algunas otras configuraciones como el dialecto y otras para hacer debug, por último se establecerán las clases a mapear.

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Las modelos siguientes contienen anotaciones propias de JPA que representan las tablas y sus relaciones correspondientes en la base de datos, como @Entity, @Id y @ManyToOne por ejemplo.

En total se han creado nueve modelos, uno por cada tabla.

Este es el modelo de Clasificación, en él se puede observar el valor de cada columna y que relaciones tiene con otras tablas como con la columna liga\_id en la tabla ligas.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Para poder interactuar con la base de datos, será necesaria la clase HibernateUtil, la cual permite inicializar SessionFactory para crear sesiones de Hibernate e interactuar con la base de datos, además utiliza el archivo de configuración mencionado anteriormente para aplicar los parámetros establecidos en la sesión.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

La clase UsuarioService se encargará de registrar a los nuevos usuarios, para ello recibe el nombre, el correo y una contraseña. Tras verificar que se han introducido los datos, se generará un salt aleatorio para la contraseña y posteriormente se hasheará junto con el salt, después se rellenará el modelo de usuario y mediante una sesión de hibernate se realizará una transacción para crear el usuario.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Para cifrar la contraseña se utiliza la clase PasswordUtils, la cual genera el salt aleatorio y codifica la contraseña con SHA-256.

Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Tras codificar la contraseña se establecen los datos introducidos y el rol de usuario por defecto en el modelo para posteriormente hacer un commit con hibernate y subirlo a la base de datos.

A la hora de inicar sesión, para poder acceder a una cuenta se comprobaran si los datos introducidos son iguales a los almacenados en la base de datos mediante una sesión de hiberante, pero como la contraseña almacenada se encuentra encriptada se deberá hasehar la contraseña introducida y verificarla mediante un método que la comparará con el hash de la base de datos junto con el salt.

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Implementación de la API

Para poder interactuar con la base de datos desde una aplicación Android será necesario crear una API. En este caso, se ha optado por desarrollar una API con el framework de spring utilizando las dependencias de jpa, spring-boot y PostgreSQL.

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

* Spring Data JPA: se centra en el uso de JPA para almacenar información en una base de datos relacional. Su característica más atractiva es la capacidad de crear implementaciones de repositorios automáticamente, en tiempo de ejecución, desde una interfaz de repositorio.
* PostgreSQL Driver: permite que los programas Java se conecten a una base de datos PostgreSQL mediante código Java estándar e independiente de la base de datos.
* Spring web: proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones web y servicios web RESTful. Simplifica el proceso de gestión de solicitudes web, la administración de sesiones y la creación de contenido web dinámico.

Antes de desarrollar la API hay que establecer los parámetros de configuración de la API, en este caso se han establecido estos valores en el archivo de properties.

Texto

El contenido generado por IA puede ser incorrecto.

La arquitectura de la API está compuesta por varias capas:

* Capa model: contiene las clases de entidad que representan las estructuras de datos del sistema. Estas clases están mapeadas directamente a las tablas de la base de datos mediante anotaciones de JPA, como @Entity y @Table.

Aquí se muestra parte de la entidad de equipos.

Texto

El contenido generado por IA puede ser incorrecto.

* Capa repository: es responsable de la interacción directa con la base de datos. En esta capa se definen interfaces que extienden de JpaRepository para una implementación concreta que permite realizar operaciones CRUD u otras consultas personalizadas.

Aquí se muestran algunos métodos de la interfaz de repositorio para obtener partidos dependiendo de distintos parámetros, como partidos de una liga específica.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

* Capa service: su función principal es procesar, validar o transformar la información que proviene de los controladores que reciben peticiones de los clientes, antes de interactuar con la base de datos, y viceversa.

Esta es la clase de servicio de partidos para declarar lo que retornará cada método de la interfaz de repositorio.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

* Capa controller: es la puerta de entrada del cliente al sistema, en ella se definen los endpoints y se encarga de recibir las solicitudes HTTP, procesarlas y retornar una respuesta adecuada en formato JSON. Cada clase perteneciente a esta capa está anotada con @RestController y utiliza métodos mapeados a rutas especificas utilizando anotaciones como @GetMapping (para obtener datos) o @DeleteMapping (para eliminar datos).

En la clase controlador para partidos se establecen los endpoints y que tipo de operación se va a realizar mediante la anotación establecida, después, en el caso del método getAllPartidos se crean listas para almacenar los resultados obtenidos y se llama al método de la capa service para obtener los datos.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Implementación en Android