Miguel Calvo Gómez

Correo: mcalvog03@educantabria.es

DNI: 72204281J

Curso: 2024/2025

Befootball

Proyecto de DAM

Índice

[1. Resumen 3](#_Toc199266131)

[2. Palabras clave 3](#_Toc199266132)

[3. Introducción 3](#_Toc199266133)

[4. Objetivos 4](#_Toc199266134)

[5. Análisis de contexto 4](#_Toc199266135)

[5.1 Análisis de contexto 4](#_Toc199266136)

[5.2 Innovación 5](#_Toc199266137)

[6. Diseño 6](#_Toc199266138)

[6.1 Diseño de la interfaz 7](#_Toc199266139)

[6.2 Casos de uso de los usuarios 14](#_Toc199266140)

[6.3 Modelo de datos 15](#_Toc199266141)

[6.4 Diagrama de clases 19](#_Toc199266142)

[6.5 Despliegue 22](#_Toc199266143)

[7. Planificación 23](#_Toc199266144)

[7.1 Diagrama de gant 23](#_Toc199266145)

[7.2 Definición de recursos y logística necesarios para cada actividad 24](#_Toc199266146)

[8. Implementación 24](#_Toc199266147)

[8.1 Implementación del servidor socket 24](#_Toc199266148)

[8.2 Implementación en PC 25](#_Toc199266149)

[8.3 Implementación de la API 41](#_Toc199266150)

[8.4 Implementación en Android 46](#_Toc199266151)

[8.5 Implementación web Odoo 51](#_Toc199266152)

[9. Puesta en marcha, explotación 52](#_Toc199266153)

[9.1 Informe de cambios de configuración, seguridad y legalidad 52](#_Toc199266154)

[9.2 Pasos necesarios para llevar el producto a producción 53](#_Toc199266155)

[10. Prueba y control de calidad 54](#_Toc199266156)

[10.1 Pruebas API 54](#_Toc199266157)

[10.2 Pruebas aplicación de escritorio 55](#_Toc199266158)

[10.3 Pruebas aplicación para smartphone 57](#_Toc199266159)

[11. Plan de empresa 58](#_Toc199266160)

[A) Sector productivo 58](#_Toc199266161)

[B) La empresa 58](#_Toc199266162)

[C) Tipo de producto 60](#_Toc199266163)

[12. Conclusión y valoración personal 61](#_Toc199266164)

[13. Bibliografía 61](#_Toc199266165)

# 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

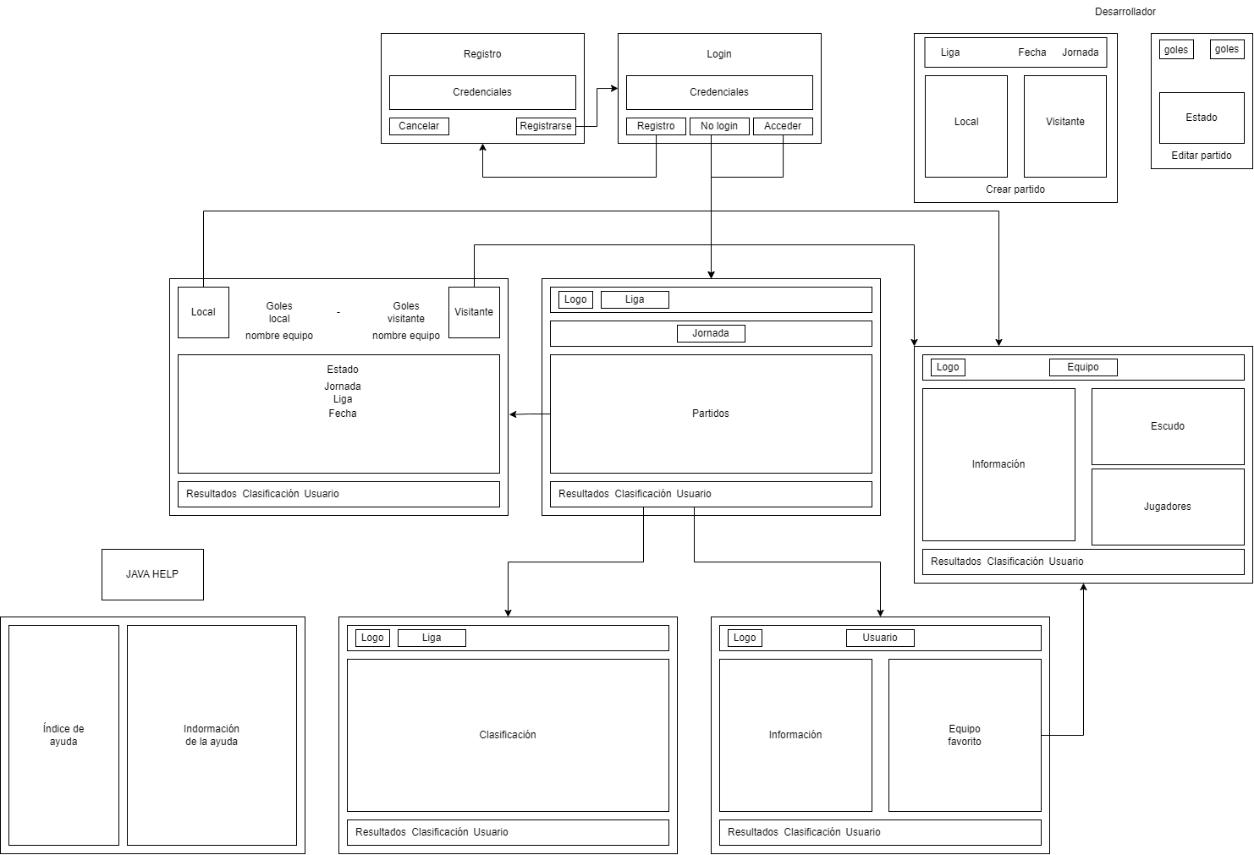
La innovación que tendrá este proyecto será principalmente su intuitiva interfaz debido a la simplicidad en cuanto a mostrar los datos y poseer un control mediante números como un teletexto o por controles habituales para navegar por los menús. También al existir un botón que permite cambiar el estilo 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.

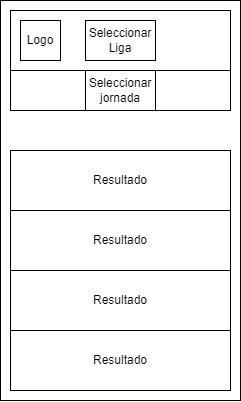
# Diseño

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 y los próximos partidos, por último, en la zona superior derecha hay un botón solo visible para usuario y desarrolladores para agregar el equipo como favorito.

Imagen que contiene Escala de tiempo

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 dispondrán de un diálogo para crear los partidos, pudiendo establecer los equipos, la fecha y la jornada.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

En cuanto a la accesibilidad, 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

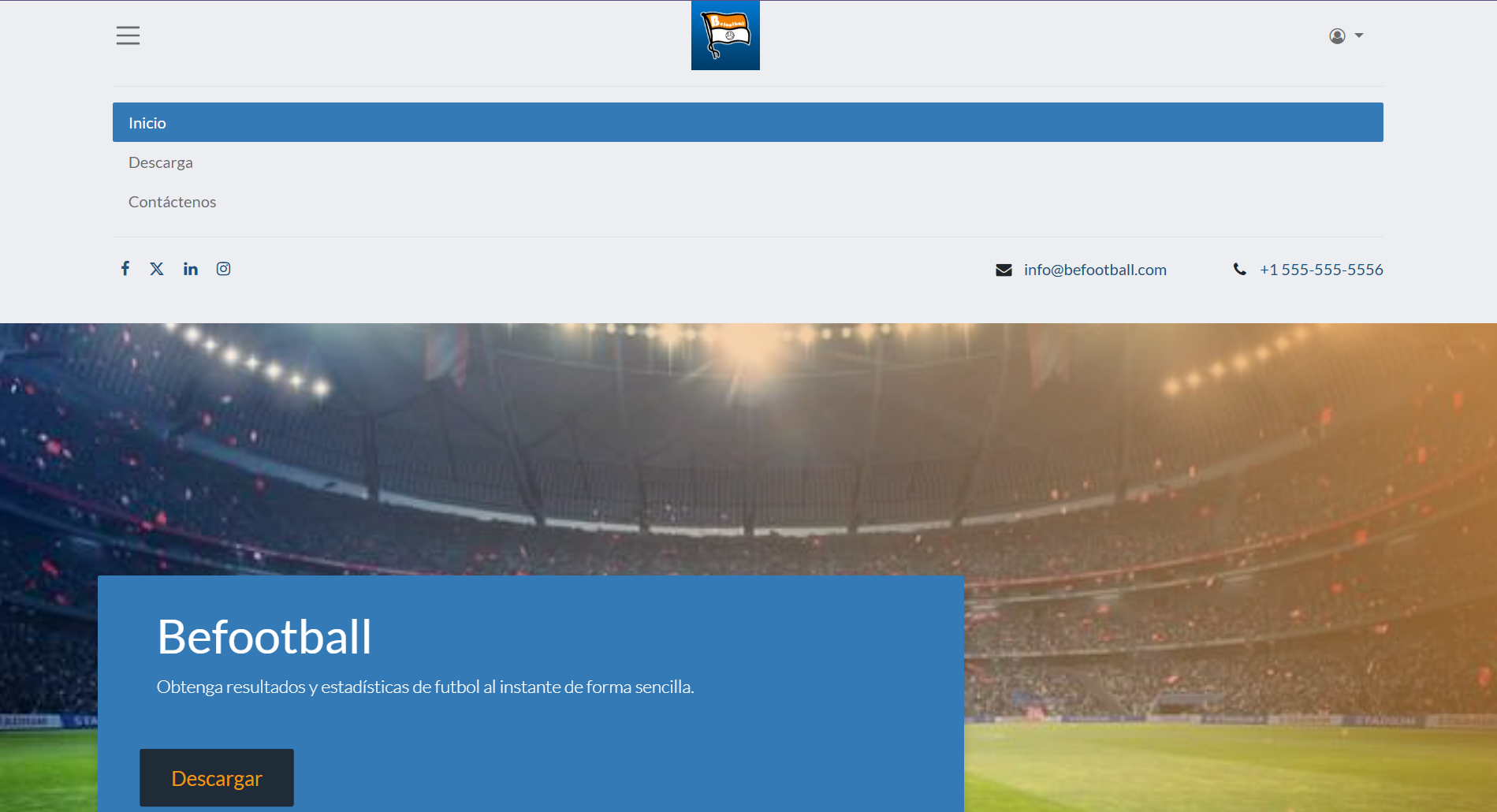
El contenido generado por IA puede ser incorrecto.

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.

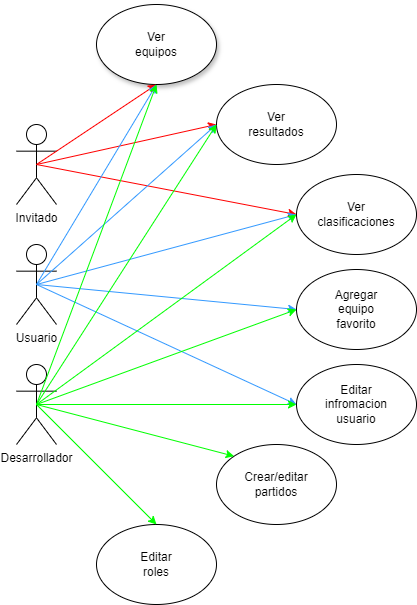
También se ha desarrollado una página web para promocionar el producto, descargarlo, hacer anuncios y un servicio para que los usuarios contacten con la empresa. Para llevar a cabo el desarrollo, se ha utilizado el módulo de sitio web de Odoo 16. Este sitio web también será accesible desde el menú de ayuda de la aplicación de escritorio.



## Casos de uso de los usuarios

En la aplicación de escritorio constarán tres tipos de usuarios.

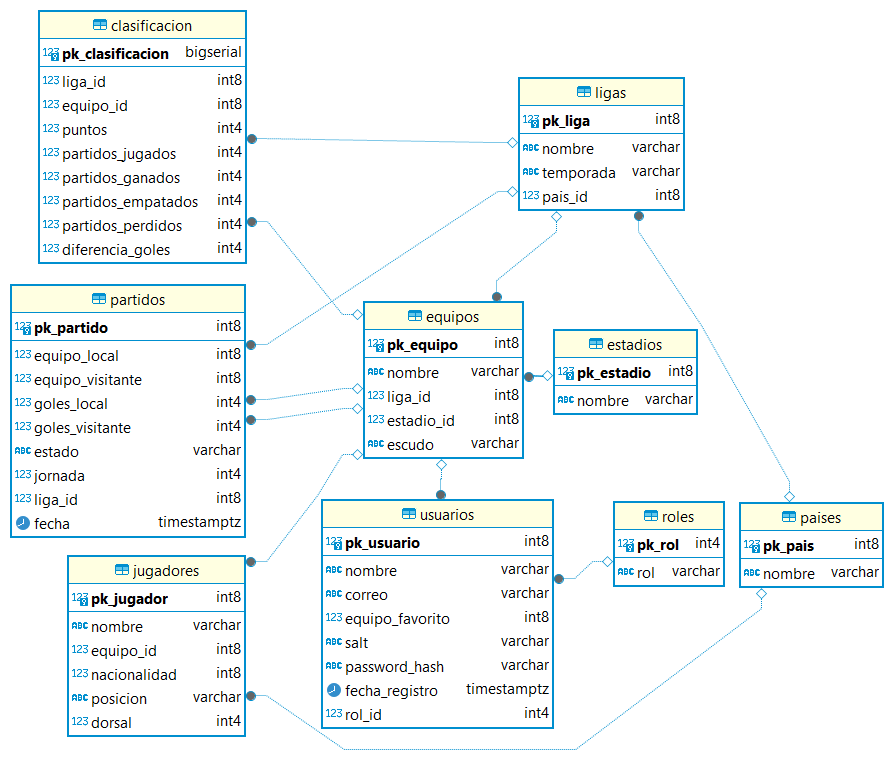
* Invitados: este tipo de usuario tendrá acceso solamente a un limitado contenido de la aplicación como la pestaña de usuario, además no poder agregar equipos como favorito.
* Usuarios (usuarios con una cuenta): los usuarios que dispongan de una cuenta podrán acceder a casi todas las funcionalidades 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

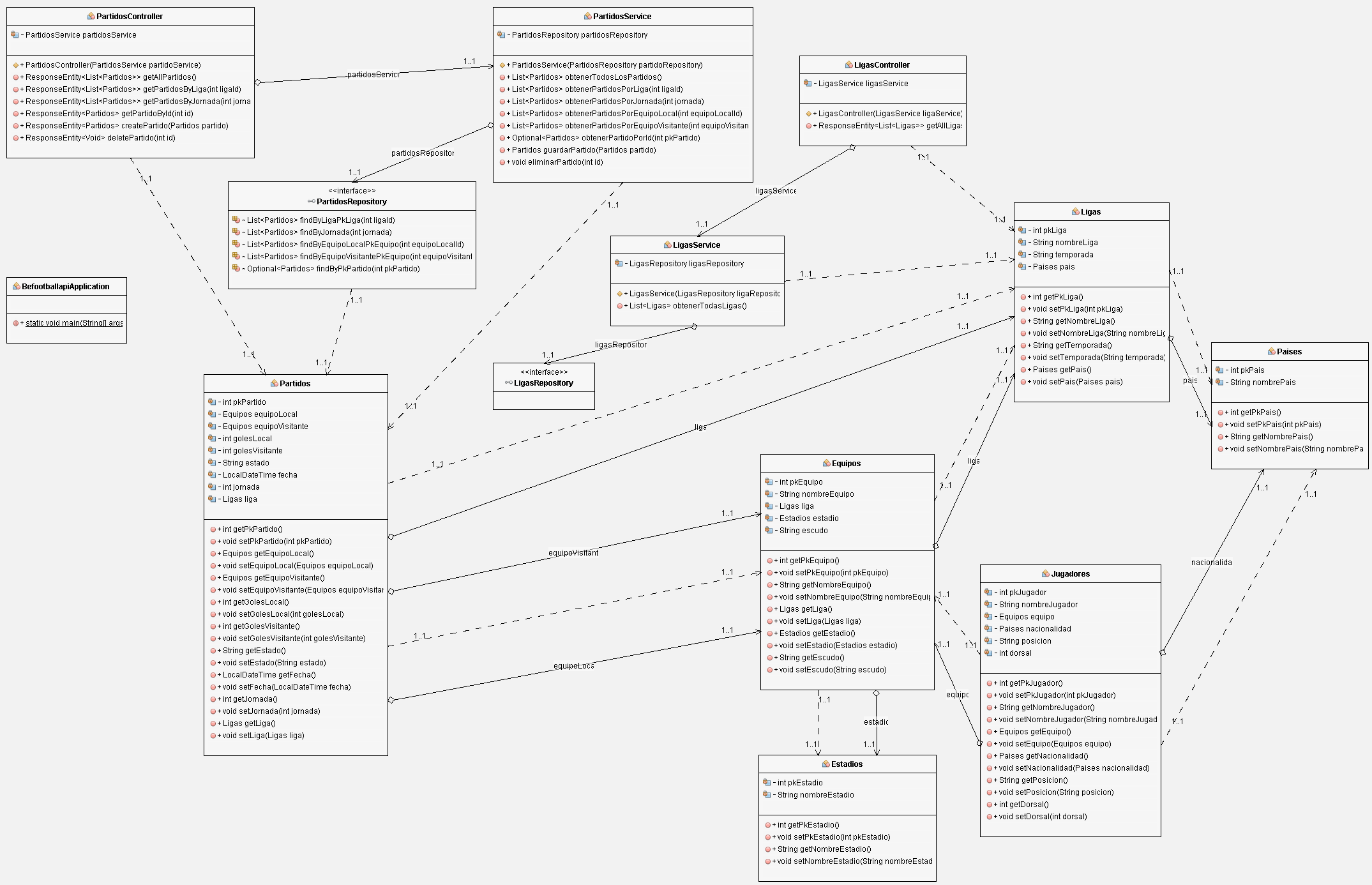
El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

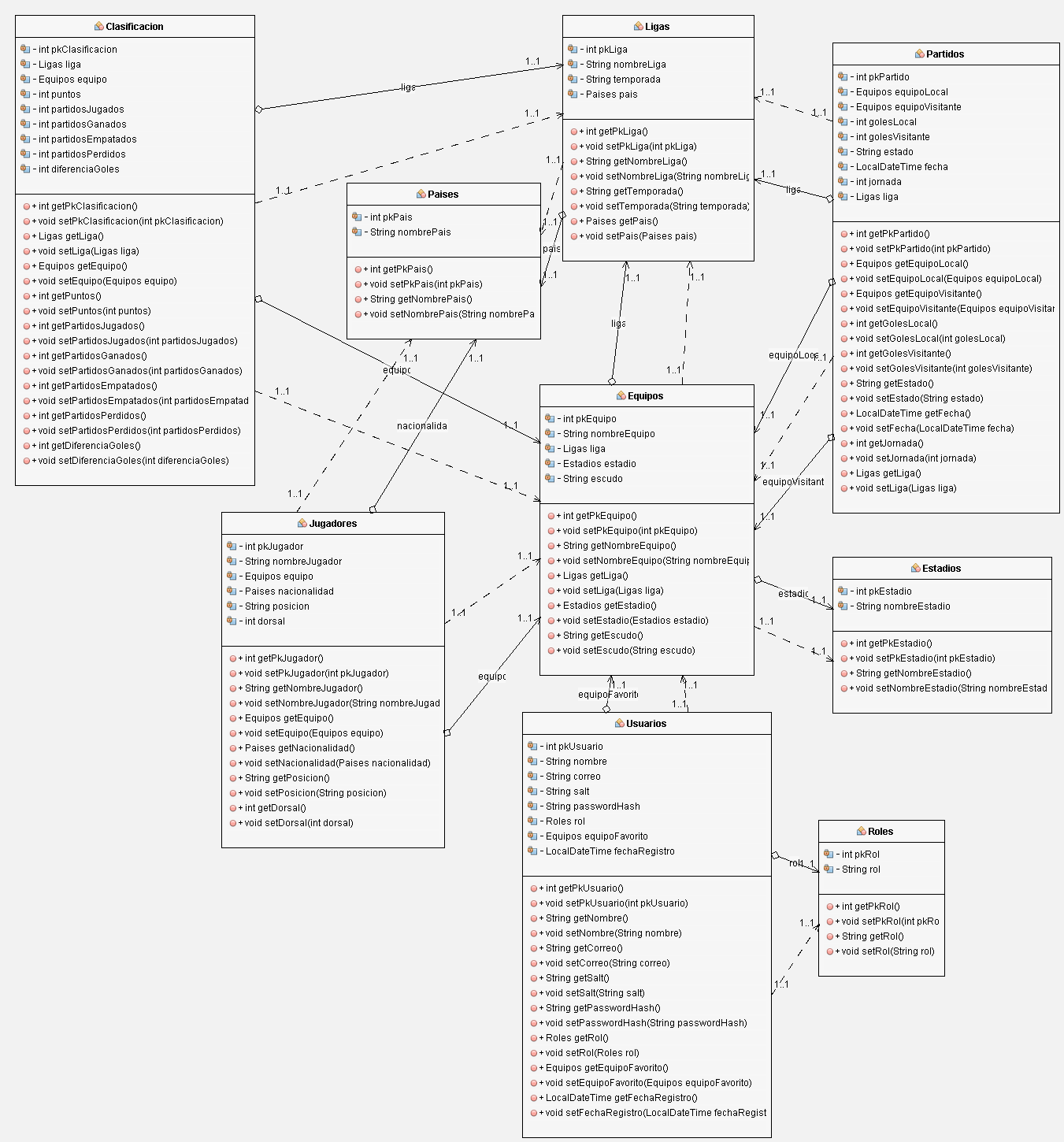
## Diagrama de clases

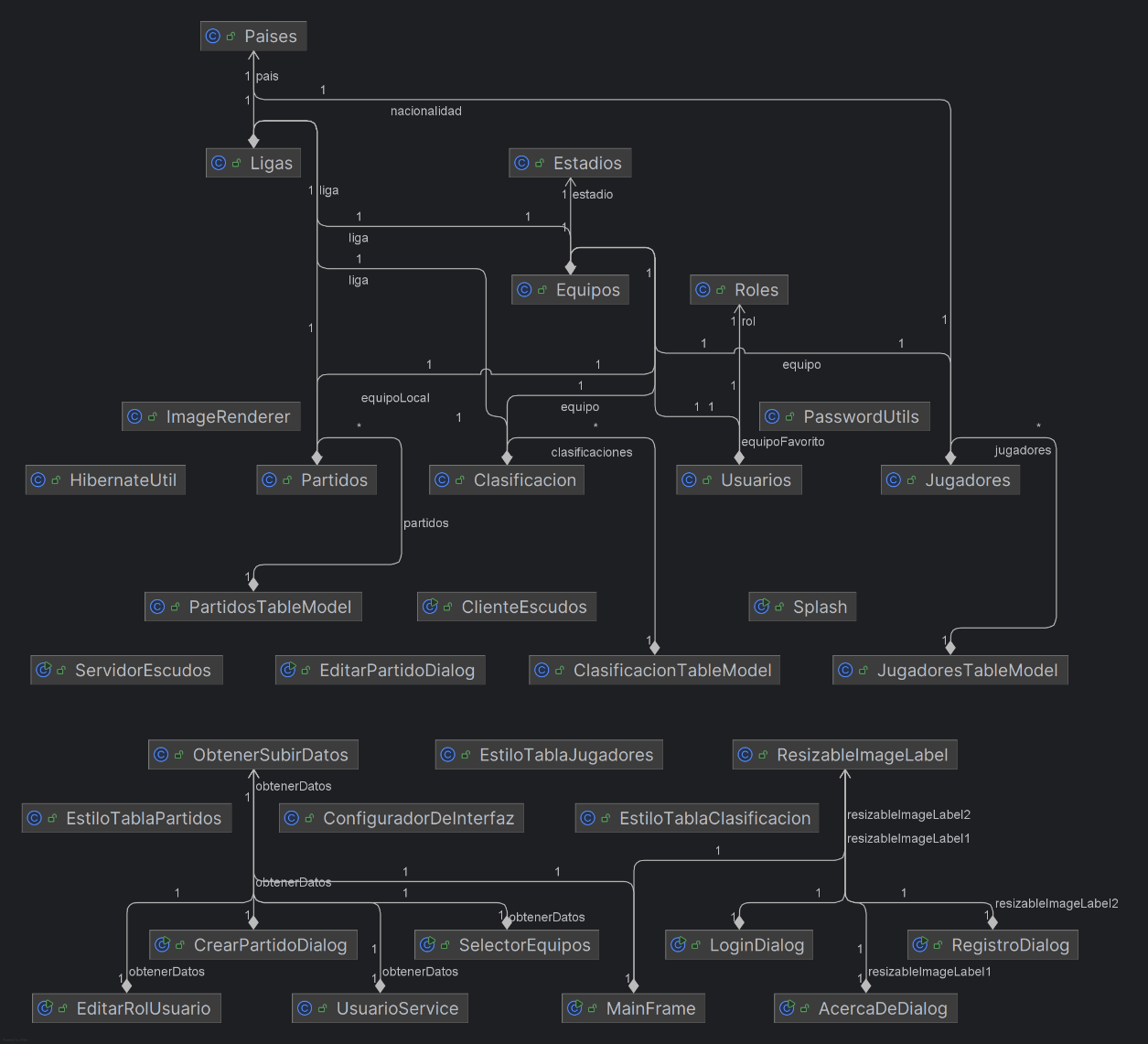
A continuación, se muestran los diagramas de clases de la API, la aplicación de smartphone y la aplicación de PC.





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.

Tabla

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 comprobarán 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.

Una vez se ha accedido a la aplicación se ejecutarán e inicializarán métodos como, por ejemplo, para crear el directorio para almacenar los escudos descargados en caso de que no exista, se inicializara el sessionFactory y la clase para obtener datos para realizar consultas de hibernate, también se cargará el panel de usuario mediante el id del usuario que ha iniciado sesión y combobox con las ligas disponibles en la base de datos y comprobaciones para detectar el rol de usuario que ha accedido y configurar la interfaz en base al tipo de usuario.

Tabla

El contenido generado por IA puede ser incorrecto.

Para poder cargar los partidos en la tabla del panel principal, primero se obtendrá una lista de los nombres de la tabla de ligas mediante el método obtenerLigas de la clase ObtenerSubirDatos y se cargará en un comboBox.

Texto

El contenido generado por IA puede ser incorrecto.

Después, dependiendo de la liga seleccionada se rellenará otro ComboBox con las jornadas disponibles de esa liga, para ello se obtendrá una lista con los números de las jornadas existentes de la liga seleccionada, para generar esta lista se deberán obtener todos los partidos pertenecientes a una liga especifica en la tabla partidos de la base de datos y mediante un HashSet eliminar los números de jornadas repetidos para que no aparezca la misma jornada más de una vez en el ComboBox ya que cada partido almacena el numero de la jornada.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

El comboBox de las jornadas deberá ser limpiado antes de cargar nuevas jornadas de otra liga.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Cuando se hayan seleccionado la liga y la jornada deseadas se creará una lista con los partidos y se cargarán en una tabla junto sus escudos.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Para poder mostrar los escudos en la tabla de partidos y clasificación se deben crear modelos de tabla, ambos modelos funcionarán igual para mostrar los escudos de los equipos, en el caso del modelo de la tabla para los partidos, se declarará una lista con los partidos, un array con el nombre de las columnas y los datos del servidor socket.

En el siguiente método se establecen los datos obtenidos en su respectiva fila y columna.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Para poder obtener el escudo, en caso de que no exista en el directorio, se descargará mediante el servidor socket, para ello se enviará un comando con el nombre del archivo solicitado, después se redimensionará a 60x60 y se aplicacrá en la celda correspondiente.

Texto

El contenido generado por IA puede ser incorrecto.

Una vez se ha establecido el escudo en su lugar, deberá renderizarse la celda eliminando el fondo y el texto de la ruta de memoria de la imagen.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

De forma similar a como se cargan los partidos, en el caso del panel de clasificaciones, dependiendo de la liga seleccionada en el mismo comboBox se obtendrá una lista de equipos de la tabla de clasificación pertenecientes a la liga seleccionada.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Para que los equipos se ordenen en la tabla por sus puntos, se ejecuta el siguiente método. Utiliza la clase Collections junto con la interfaz Comparator mediante la cual se ordena la posición de cada equipo dependiendo de sus puntos.

Interfaz de usuario gráfica, Texto, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Para poder mostrar el panel con los detalles de un partido seleccionado en la tabla del panel de resultados, mediante un método se obtendrá el id del partido y se mostrará el panel.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Para poder cargar los detalles, básicamente se obtendrán mediante una concatenación de gets. En el caso de renderizar los escudos de los equipos, se ejecutará un método similar al de las tablas de resultados y clasificación, pero en este caso se mostrarán en dos paneles, para ello se creará un jLabel en la que se agregará la imagen y después se añadirá a su respectivo panel.

Texto

El contenido generado por IA puede ser incorrecto.

Para poder cargar el panel de los detalles de un equipo en todos los casos en el que se accede a dicho panel, mediante un evento haciendo clic con el ratón en la tabla de clasificación, en el panel de partidos o en el del equipo favorito en el panel de usuario se obtendrá la id del equipo y cargará la información básica con una concatenación de gets, en el caso de obtener su posición en liga, se establecerá en el comboBox de ligas del panel de clasificación su competición para posteriormente obtener su posición en la tabla.

Texto

El contenido generado por IA puede ser incorrecto.

Si el equipo cuenta con una serie de jugadores en la base de datos se creará una lista con estos y se renderizarán en una tabla con sus datos.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

También se cargarán en otra tabla los próximos enfrentamientos de ese equipo, para ello se filtraran partidos con una fecha mayor a la actual del equipo seleccionado.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

En el panel de equipo también existe un botón solo visible para usuarios que hayan iniciado sesión para que puedan agregar el equipo mostrado como favorito. Para poder agregar el equipo, se utilizará el id del usuario que tiene la sesión iniciada y el id del equipo que se ha cargado en el panel.

Texto

El contenido generado por IA puede ser incorrecto.

Por último, el panel de usuario tiene un funcionamiento similar a los anteriores, se mostrará información básica del usuario que ha iniciado sesión mediante concatenación de gets y se mostrará su equipo favorito junto el escudo, además habrá un botón para editar el equipo el cual mostrará un panel con un comboBox con las ligas, un jList con los equipos de la liga seleccionada y un botón para establecer el equipo deseado.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

A continuación, se mostrará el funcionamiento de las funcionalidades para desarrolladores.

Para poder crear partidos, un combobox cargará las ligas y dependiendo de la competición seleccionada se cargarán dos jList con los equipos correspondiéndose con el equipo local y el visitante. Además, existirán dos campos para introducir la jornada y la fecha.

El método que se encargará de registrar el partido obtendrá los valores introducidos y realizará validaciones para posteriormente establecer la información en el objeto de partido y enviarlo a la base de datos.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

En caso de que se vaya a actualizar el resultado y el estado de un partido, desde el panel de partido del jFrame principal se podrá acceder a este menú que recibirá el id del partido, sus goles y el estado que se esté mostrando. Estos datos se cargarán en sus respectivos campos para ser editados. Para poder actualizar la información se creará un objeto de Partido con sus datos y después se enviará a la base de datos con los datos que se hayan actualizado.

Texto

El contenido generado por IA puede ser incorrecto.

La última herramienta para desarrollador permitirá editar el rol de cualquier usuario tan solo seleccionando el rol en un comboBox y estableciendo el id del usuario a editar. Para subir los cambios a la base de datos se generará un objeto de Usuario mediante el id introducido y después se le asignará el rol.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Por último, hay que destacar que se implementó un sistema de logs para que en los usuarios puedan enviar las trazas de error al equipo de desarrollo en caso de errores. Para ello se diseño la clase LogManagerApp que generará un archivo .log para guardar las trazas, además de configurar el logger.

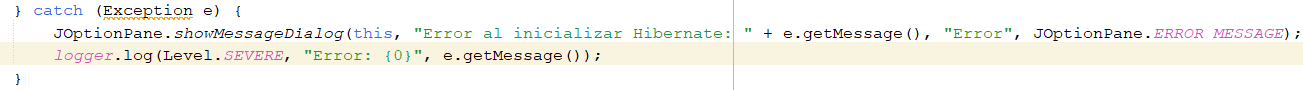
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Para que las excepciones se guarden en el fichero, en cada clase del proyecto se declarará el logger y en los catch se imprimrá la excepción en el fichero.

Imagen que contiene Texto

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

Para el desarrollo de la aplicación de Android con kotlin, primero deberemos instalar las dependencias de retrofit para poder trabajar con la API desarrollada en spring.

Texto

El contenido generado por IA puede ser incorrecto.

Se declarará en una clase de constantes la dirección del servidor y las url para acceder a los datos de ligas y partidos.

Texto

El contenido generado por IA puede ser incorrecto.

En un paquete nuevo se establecen los modelos de datos de las distintas tablas de la base de datos necesarias para el funcionamiento deseado de la aplicación, por lo que solo se crean los modelos de Equipos, Estadios, Ligas, Países y Partidos.

Este es el modelo de Partidos:

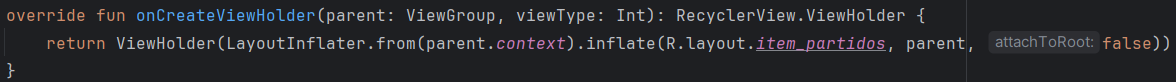
Texto

El contenido generado por IA puede ser incorrecto.

Dos interfaces service establecerán que se generará una lista de ligas y partidos utilizando la url de las constantes.

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

La clase PartidosListAdapter funciona como adaptador de RecyclerView para mostrar la lista de partidos, para ello utilizará el método onCreateViewHolder el cual enviara al layout del recycleView item\_partidos.xml la información de cada partido.



El método onBindViewHolder se encargará de vincular cada valor del objeto Partidos en la posición correspondiente de la vista.

Texto

El contenido generado por IA puede ser incorrecto.

En la clase principal se declaran los atributos principales. Entre estos atributos se destacan adapter para mostrar los partidos en el recycleView y binding para acceder a las vistas.

Texto

El contenido generado por IA puede ser incorrecto.

El método setupRetrofit creará una instancia de retrofit con la url de servidor y un convertidor de json a objetos kotlin para poder utilizar los modelos, además se construirán los servicios de ligas y partidos.

Texto

El contenido generado por IA puede ser incorrecto.

El método getLigas permitirá mapear las ligas en un spinner asociando un listener para detectar cuando el usuario selecciona una de ellas, todo esto lo realizará en un hilo de fondo gracias a Dispatchers.IO.

Texto

El contenido generado por IA puede ser incorrecto.

setupLigaSpinnerListener permitirá guardar el id de la liga seleccionada por el usuario en el spinner y rellenará el spinner de las jornadas con las disponibles de la liga actual eliminando la selección anterior previamente para evitar errores.

Texto

El contenido generado por IA puede ser incorrecto.

getJornadas obtendrá datos de todos los partidos de la API, despues los filtrará por la liga seleccionada, extraerá las jornadas disponibles y las enviará al spinner de jornadas.

Texto

El contenido generado por IA puede ser incorrecto.

Por último, el método setupJornadasSpinnerListener configura el spinner de jornadas para que, cuando el usuario seleccione una jornada, se guarde la jornada elegida y se llamen a los partidos correspondientes de la liga seleccionada actualizando automáticamente los partidos que se muestran según la jornada elegida.

Texto

El contenido generado por IA puede ser incorrecto.

## 8.5 Implementación web Odoo

Para generar la página web se ha optado por el módulo de sitio web de Odoo 16.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Esta web está compuesta por cuatro páginas:

* Home: contiene información sobre lo que ofrece el producto.
* Contact us: permite al usuario comunicarse con la empresa para consular algún problema.
* Thanks (Contact us): pantalla que muestra que se envió el mensaje correctamente.
* Descarga: permite descargar la aplicación de escritorio y de smartphone.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

# Puesta en marcha, explotación

## 9.1 Informe de cambios de configuración, seguridad y legalidad

**Configuración**

Por el momento la puesta en marcha de la aplicación solo se hará en España, para ello se tendrá en cuenta esta configuración:

* Base de datos: Se alojará en una instancia de PostgreSQL gestionada en la nube, dentro de un entorno seguro con acceso restringido. Se utilizarán roles personalizados y cifrado de contraseñas.
* Servidor de API y sockets: Se desplegará en una máquina virtual configurada con un sistema Linux (Ubuntu Server) donde correrán tanto los servicios REST como el servidor de comunicación en tiempo real. El tráfico se enrutará a través de HTTPS con certificados SSL válidos para proteger los datos transmitidos.
* Odoo: Se instalará en una instancia separada dentro del mismo proveedor cloud, con acceso seguro y dominio personalizado para la página web y en un futuro la gestión de la empresa.
* Costes asociados: La puesta en marcha implicará costes mensuales vinculados al hosting en la nube (máquinas virtuales, almacenamiento, copias de seguridad automáticas) y la compra de un dominio con certificado SSL para garantizar la seguridad de las conexiones.

**Seguridad**

Todas las contraseñas de los usuarios se encriptarán antes de ser enviadas a la base de datos de esta forma nadie podrá acceder a la cuenta de los usuarios. Además en caso de ataque de bots a la hora de registrarse, podremos verificar la fecha y hora del registro y no permitir acceso de ese cliente.

**Legalidad**

En el entorno de producción se cumplirán los principios de legalidad vigentes, especialmente los relacionados con la protección de datos personales y la normativa aplicable a servicios digitales en España y la Unión Europea:

* Protección de datos: El sistema sigue el Reglamento General de Protección de Datos (RGPD o GDPR), limitando la recogida de datos personales al mínimo necesario y cifrando toda la información sensible.
* Registro de accesos: Se implementa un sistema de trazabilidad para registrar accesos y acciones relevantes, útil tanto para fines de seguridad como para cumplir con auditorías legales, en este caso se utilizará la fecha de registro.
* Subcontratación: se verificarán las condiciones legales de tratamiento de datos por parte de los proveedores servicios externos cloud, asegurando que cumplen con la normativa europea.

## 9.2 Pasos necesarios para llevar el producto a producción

El proceso de transición de la aplicación desde el entorno de desarrollo hasta el entorno de producción requiere una serie de pasos técnicos organizados para garantizar la estabilidad, accesibilidad y seguridad del sistema una vez lanzado al público. A continuación, se describen los pasos que se seguirán para poner en marcha tanto la aplicación de escritorio como la móvil, además de los servicios de backend y la página web.

**Despliegue del código en el servidor de producción**

El primer paso consistirá en desplegar los servicios centrales (API REST, servidor socket y página web en Odoo) en un entorno cloud, inicialmente previsto en AWS o Google Cloud, como se especifica en el apartado 6.5.

* Se transferirá el código fuente desde los repositorios locales a las máquinas virtuales configuradas en el entorno cloud.
* Spring Boot se compilará y desplegará en una instancia con sistema operativo Ubuntu Server.
* El servidor socket para la gestión de escudos se instalará y configurará en la misma máquina virtual.
* El módulo de sitio web de Odoo será desplegado en una instancia separada, también en la nube.

**Configuración del entorno de producción**

* Una instancia PostgreSQL para la base de datos, con acceso restringido por usuarios y cifrado activado.
* Una máquina virtual para alojar la API REST de Spring Boot y el servidor socket de Java.
* Otra instancia para la página web creada con Odoo, utilizando un dominio personalizado y HTTPS con un certificado SSL creado con Let’s encrypt.

**Pruebas en producción**

Una vez desplegado el sistema, se realizarán pruebas integrales en el entorno de producción:

* Verificación del funcionamiento de los endpoints de la API desde la aplicación Android.
* Confirmación del correcto funcionamiento del servidor socket en la descarga y visualización de escudos con un programa de pruebas.

**Lanzamiento de las aplicaciones de PC y móvil**

* La aplicación de escritorio se empaquetará como archivo ejecutable y se distribuirá a través de la página web en Odoo en la sección de descargas.
* La aplicación Android será compilada como archivo APK y también se pondrá a disposición del público en la página web.
* Posteriormente se evaluará la publicación oficial en la Google Play Store.

# 10. Prueba y control de calidad

A continuación, se explicarán las pruebas realizadas en el proyecto para comprobar el funcionamiento.

## 10.1 Pruebas API

Para realizar pruebas en para la API, se utilizará postman. Una aplicación que permite enviar solicitudes HTTP a servidores y mostrar sus respuestas lo cual nos permitirá hacer la llamada GET utilizada en la aplicación de smartphone.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

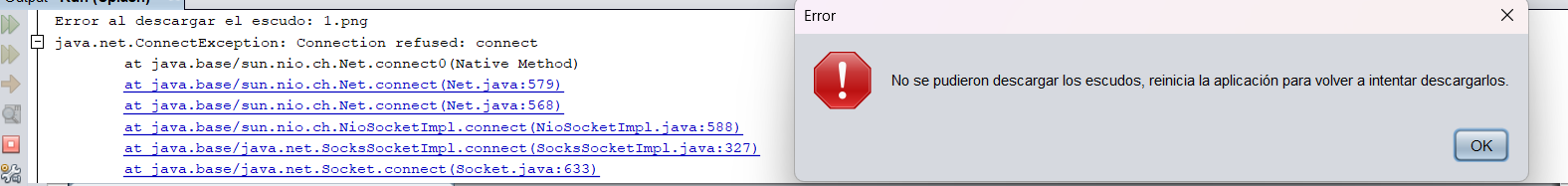
## 10.2 Pruebas aplicación de escritorio

En la versión de escritorio, principalmente se ha utilizado la consola como una herramienta para detectar errores y poder saber de dónde provienen, en algún caso, la consola también se ha utilizado para mostrar información durante el desarrollo, por ejemplo, para para configurar la interfaz dependiendo del usuario que ha iniciado sesión.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

En caso de que el servidor socket no este en funcionamiento se le mostrará un dialogo al usuario y se mostrará un mensaje por consola el mensaje junto la excepción.



A pesar de no poder descargarse los escudos, la aplicación se podrá seguir utilizando con normalidad, pero mostrando un placeholder para los escudos.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

A la hora de registrarse, se comprobará si se han rellenado los campos y se ha utilizado el formato adecuado para el correo electrónico.

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

También al iniciar sesión se comprobará si el usuario y la contraseña introducidos existen en la base de datos.

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

Para comprobar el funcionamiento del servidor socket en la descarga y visualización de escudos se desarrolló un programa de prueba para obtener las imágenes (este pequeño programa fue desarrollado al inicio del proyecto para realizar las pruebas).

Imagen que contiene Diagrama

El contenido generado por IA puede ser incorrecto.

## 10.3 Pruebas aplicación para smartphone

En caso de que la aplicación no pueda conectar con la API mostrará un mensaje en la parte inferior informando cual es problema. En el caso de que sí se conecte, se mostrarán los resultados, las ligas y las jornadas disponibles.

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

# 11. Plan de empresa

## A) Sector productivo

El proyecto se enmarca en el sector productivo de las Nuevas Tecnologías de la Información y la Comunicación (TIC) en España, un sector caracterizado por su dinamismo, constante crecimiento y capacidad de transformación digital.

**Características fundamentales del sector TIC en España:**

* Número de empresas y tamaño: El sector TIC en España contaba en 2023 con más de 25.000 empresas, según datos del INE. La mayoría son pequeñas y medianas empresas (pymes), con una plantilla media de entre 3 y 50 empleados. No obstante, existen también grandes actores con cientos de trabajadores especializados.
* Cifra de negocio: Este sector generó más de 120.000 millones de euros, suponiendo más del 4% del PIB nacional, y con una tendencia de crecimiento sostenido en los últimos años.
* Forma jurídica frecuente: Predominan las sociedades de responsabilidad limitada, debido a su simplicidad de constitución y bajo capital inicial exigido.
* Dimensión geográfica: Aunque Madrid y Barcelona concentran el grueso de la actividad, la digitalización y el teletrabajo han facilitado la expansión de empresas TIC por todo el territorio español.
* Estructura organizativa: Las empresas TIC suelen adoptar estructuras organizativas horizontales, con equipos multidisciplinares y metodologías ágiles (Scrum, Kanban, DevOps) que permiten una alta adaptabilidad al cambio.

**Variables adicionales:**

* Cultural/social: Este sector tiene una clara orientación a la innovación, el emprendimiento joven y el fomento de la igualdad de género y diversidad.
* Medioambiental: Se promueve el uso eficiente de recursos mediante soluciones cloud y virtualización, reduciendo el impacto ecológico asociado al hardware.

## B) La empresa

La forma jurídica elegida para este proyecto es la Sociedad de Responsabilidad Limitada (S.L.), a continuación, se detallará la elección junto otros aspectos.

**Justificación de la elección:**

* Ventajas:
  + Limitación de la responsabilidad al capital aportado.
  + Mayor profesionalización y confianza ante clientes e inversores.
  + Régimen fiscal más flexible que el del empresario individual.
  + Facilidad para incorporar nuevos socios o empleados en el futuro.
* Inconvenientes:
  + Costes de constitución y trámites legales más elevados que el empresario individual.
  + Obligación de llevar contabilidad formal y depositar cuentas anuales en el Registro Mercantil.

**Trámites para la constitución y puesta en marcha:**

* Certificación negativa de denominación social – Registro Mercantil Central.
* Solicitud del NIF provisional – Agencia Estatal de Administración Tributaria.
* Redacción de estatutos sociales y firma ante notario.
* Inscripción de la sociedad – Registro Mercantil Provincial.
* Alta en el censo de empresarios – Agencia Estatal de Administración Tributaria.
* Alta en la Seguridad Social – Tesorería General de la Seguridad Social.
* Alta de centro de trabajo – Inspección Provincial de Trabajo.
* Licencia de actividad – Ayuntamiento local.
* Alta como demandante de empleo y contratación – Servicio Público de Empleo Estatal.
* Registro de marca y dominio – Oficina Española de Patentes y Marcas.

**Organigrama:**

* Fundador, desarrollador y diseñador.
* Operador de resultados en tiempo real (1 persona por cada 3 ligas activas)
* Soporte al cliente

**Convenio colectivo aplicable:**

Convenio Estatal de Empresas de Consultoría, Estudios de Mercado y Opinión Pública (sector TIC).

**Plan de prevención de riesgos laborales:**

* Se podría elaborar un plan básico con la colaboración de una empresa de prevención externa.
* Riesgos más relevantes: fatiga visual, posturas prolongadas, estrés por carga de trabajo.
* Medidas preventivas: descansos activos, mobiliario ergonómico, formación en salud digital, evaluaciones periódicas.

## C) Tipo de producto

**Descripción técnica:**

Se trata de una plataforma tecnológica compuesta por tres elementos:

* Aplicación de escritorio en Java: permite consultar resultados, clasificaciones y estadísticas de fútbol.
* Aplicación Android en Kotlin: visualización rápida de resultados por jornada y liga.
* Página web (Odoo 16): acceso a descargas, contacto y promoción del producto.

**Tecnologías empleadas:**

* Backend: Spring Boot (Java), Hibernate, PostgreSQL.
* Frontend (PC): Java Swing.
* Frontend (Android): Kotlin + Retrofit.
* Infraestructura: Servidor cloud, servidor socket, cifrado de contraseñas con SHA-256 y pgcrypto.
* Seguridad: HTTPS, cifrado de contraseñas, roles de usuario.

**Necesidades que satisface:**

* Acceso rápido y simplificado a resultados de fútbol.
* Interfaz inspirada en el teletexto para personas mayores o con bajo nivel de alfabetización digital.
* Eliminación de publicidad intrusiva y diseño amigable.
* Seguimiento personalizado de un equipo favorito.

**Factores de innovación:**

* Diseño retro e intuitivo para personas mayores.
* Resultados en tiempo real actualizados por operadores humanos.
* Multiplataforma (PC, móvil, web).
* Rol de desarrollador para actualizaciones internas sin necesidad de relanzar la app o instalar actualizaciones frecuentemente.

# 12. Conclusión y valoración personal

# 13. Bibliografía

**App para Android**

Consumo Api desde kotlin:

<https://youtu.be/xm0GhiLD_gw?si=eFmpeBbanWvqo_pV>

List Adapter y ViewBinding:

<https://www.youtube.com/watch?v=-uxKse1g4_8>

Crear modelos:

<https://www.youtube.com/watch?v=zkGBa-0QGeQ>

Vista:

<https://www.youtube.com/watch?v=eAgNdiTlLFc>

Diseño RecycleView:

<https://youtu.be/ODFZHjt9k-Y?si=8kkM_IMzMkk5uTpS>

Configurar RecycleView:

<https://www.youtube.com/watch?v=GmrKCVMG7qQ>

**API:**

- Spring JPA: <https://spring.io/guides/gs/accessing-data-jpa>

- Postgre SQL Driver: <https://www.postgresql.org/download/products/2-drivers-and-interfaces/#:~:text=The%20PostgreSQL%20JDBC%20driver.,standard%2C%20database%20independent%20Java%20code>.

- Spring web: <https://medium.com/@AlexanderObregon/exploring-the-spring-web-dependency-a-beginners-overview-f19e4620ef5e#:~:text=The%20Spring%20Web%20dependency%20is%20a%20module%20within%20the%20larger,and%20building%20dynamic%20web%20content>.

**Proteccion de datos:**

<https://www.aepd.es/guias/guia-proteccion-datos-por-defecto.pdf>