

# GUIÓN DE LA MEMORIA



***SOCER & FOOTBALL***

Hecho por: Alejandro Fernandez  
Gomez-Caminero

# Índice

Introducción

Plan de Empresa

Descripción detallada del sistema y listado de historias de usuarios

Modelado y diseño

Diseño

Implementación

## Introducción

Soccer & Football es una red social en la que puedes acceder a información de futbolistas españoles, en nuestra red social podemos hacer publicaciones que estén relacionadas con el mundo del fútbol, es una aplicación en la que podrás expresar tu gusto sobre el fútbol en general o cuanto quieres a tu equipo(por ejemplo: el Real Betis Balompié) o a un jugador de ese equipo(por ejemplo: Sergio Canales). Aparte también podrás ver como esos jugadores van durante la temporada actual. También podrás seleccionar tus equipos favoritos, tus jugadores favoritos

# Plan de Empresa

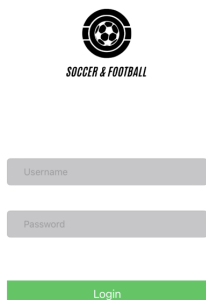
## JUSTIFICACIÓN:

Si tenemos en cuenta que las redes sociales en España son muy populares y si le añadimos un deporte como el fútbol que es un deporte muy popular en España, es cierto que en España, la gente usa cualquier red social según su edad, puede que ninguna red social haya incluido algo relacionado con el mundo del deporte.

## NOMBRE y LOGO:



## PRODUCTO:



MODELO DE NEGOCIO: El modo en el que se va a tener ganancias son en una sección premium para dichos usuarios que hayan pagado una cuota de 3,0 € mensual y una de 20,0€ anual ya depende de cómo quiere invertir el dinero el cliente.

CONSUMIDORES Y POSIBLES CLIENTES: Las personas que pueden consumir la aplicación pueden desde chicos/as de 16 años hasta hombres y mujeres de 50 años. En un principio la aplicación será a nivel nacional, en un futuro será internacional.

COMPETENCIA: Instagram, Twitter, Facebook...

## DAFO:

- Debilidades: que no sea tan atractiva a la hora de subir fotos.
- Amenazas: que roben todos los datos de los servidores e intenten pedir una recompensa para recuperar esos datos.
- Fortalezas: dar a los clientes algo que no tienen todas las aplicaciones que es la pantalla de

fútbol.

- Oportunidades:

PUBLICIDAD Y PROMOCIÓN: pagar a un youtuber (por ejemplo: DjMariio) para que haga publicidad de la aplicación en sus vídeos.

## Descripción detallada del sistema y listado de historias de usuario

En la aplicación Soccer & Football queremos gestionar toda la información que pasan los usuarios cuando hacen una publicación y toda la información de todas las ligas, de sus equipos y sus jugadores.

En primer lugar, lo que tiene que hacer el usuario es loguearse si tiene cuenta pero si este no tiene cuenta registrada en el sistema, tendrá que registrarse dando sus datos. Entonces cuando el usuario se haya registrado correctamente y sin ningún problema volverá a la pantalla de login en el que le pedirá su correo electrónico y su contraseña.

Cuando el usuario se haya logueado correctamente, la aplicación le enviará por defecto a la pantalla donde se muestra todas las publicaciones de los usuarios que están registrados en el sistema, a partir de aquí puede el usuario mirar su propio perfil en el que se encontrará con sus datos personales, que serán algunos de los datos que introdujo cuando se registró (por ejemplo: su número de teléfono, correo electrónico...). Si el usuario quiere hacer una publicación en el menú de navegación encontrará un menú específico que será el símbolo de suma, cuando haga acceda a ese menú le permitirá añadir una descripción a la foto, subir una foto desde la galería desde el mismo dispositivo.

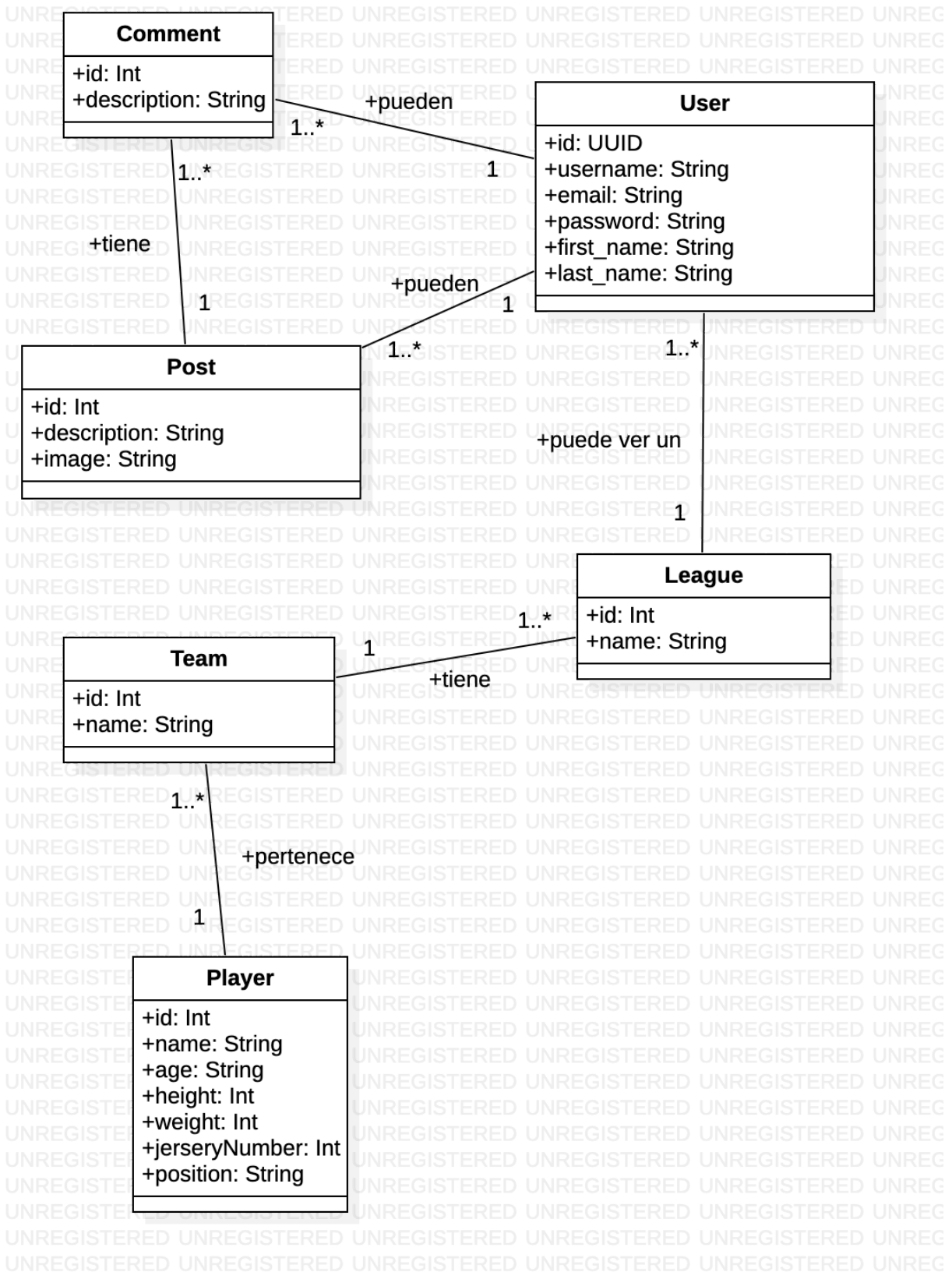
También en la barra de navegación nos encontraremos un logo de una lupa, cuando accedemos nos saldrá una pantalla en la que podemos buscar otros usuarios y mostrar si quieres seguir a este usuario o si ya lo sigues. Por último si quieres saber cómo va tu jugador favorito, pues nos iremos a la barra de navegación en la que abra el logo de un balón de fútbol, en el que cuando acceda el usuario le mostrará las ligas que están registradas en la aplicación, para ello tendrá que acceder a liga en el que se encuentra el jugador, pero claro también tenemos buscar el equipo en el que juega.

Cuando hayamos encontrado dicho jugador nos dejará marcar/desmarcar como favorito que esta función se verá con una estrella(marcada la estrella estara de color amarillo y desmarcada se encontrara vacia). Aparte de marcar/desmarcar favorito un jugador también podemos observar sus estadísticas actuales en la liga que se encuentra(por ejemplo: el número de goles que ha marcado en la temporada, el número de asistencias que lleva).

Listado de usuarios

<https://trello.com/invite/b/8GJeeJ7k/ATTI2f4fd032d5a91764843e1f4581dac03bFF4CE6CA/pdam>

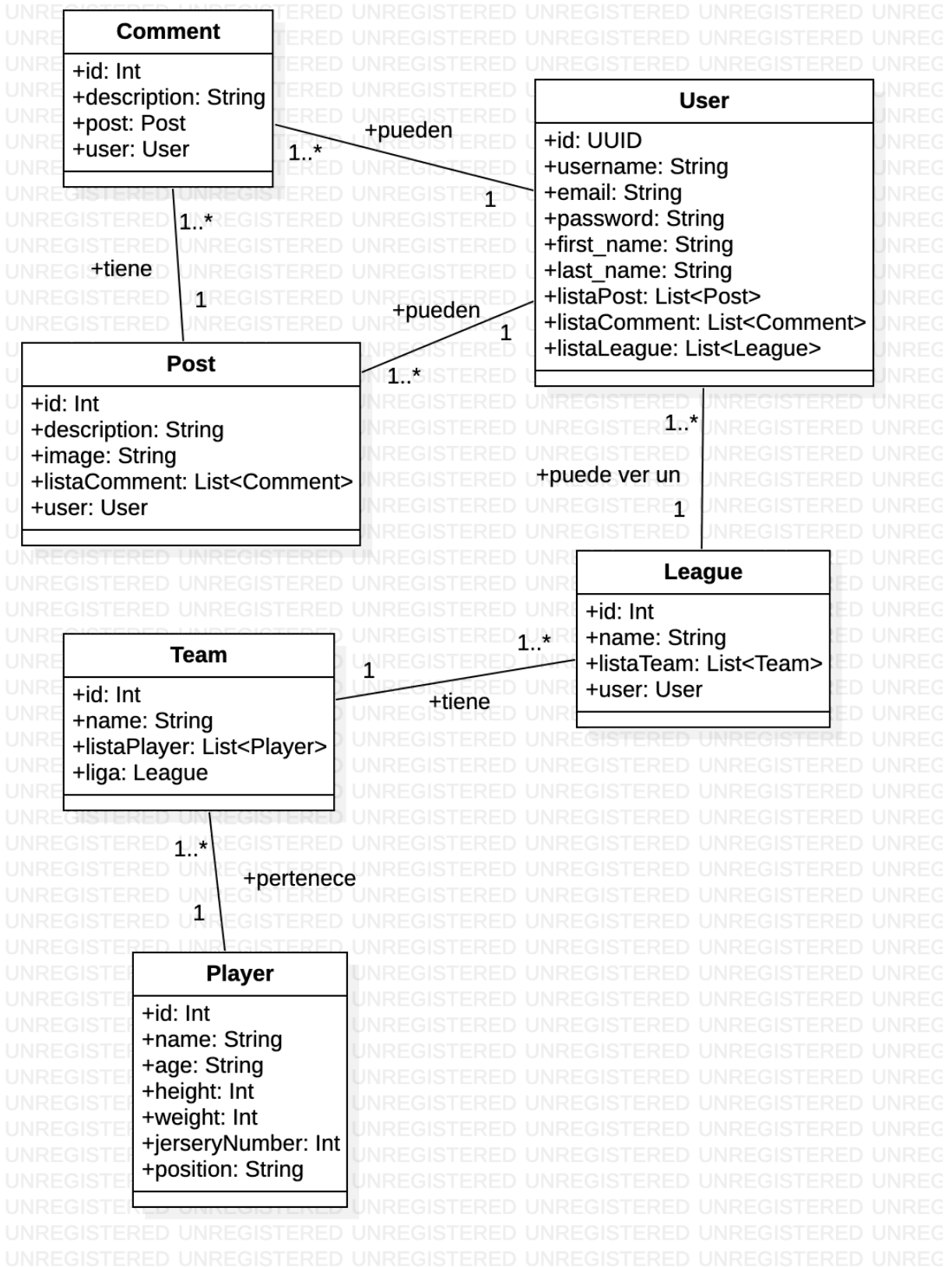
## Modelado y diseño







# Diseño



HTTP	URL	Descripción
POST	/auth/register/user	Añade un nuevo usuario. Si se añade correctamente, devuelve <i>201 Created</i> Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i>
POST	/auth/register/admin	Añade un nuevo administrador. Si se añade correctamente, devuelve <i>201 Created</i> Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i>
POST	/auth/login	Loguea a un usuario/administrador. Si se añade correctamente, devuelve <i>201 Created</i> Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i>
PUT	/user/changePassword	Modifica la contraseña de un usuario. Si se realiza correctamente, la respuesta será <i>200 OK</i> Si el artista a modificar no existe, la respuesta será <i>404 Not Found</i> .
DELETE	/user/{userId}	Borra un usuario/administrador. Si se realiza correctamente, debe devolver <i>204 No Content</i> .
GET	/user/me	Obtiene los datos del usuario logueado. Si se realiza correctamente devuelve 200 sino 404.

HTTP	URL	Descripción
POST	/team/	Añade un nuevo equipo. Si se añade satisfactoriamente, devuelve <i>201 Created</i> Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i>
GET	/team/	Obtener una lista de todos los equipos. Código de respuesta 200 (o 404 si no hay ninguna)
GET	/team/{id}	Obtener un equipo por su ID. Código de respuesta 200 (o 404 si no se encuentra)
PUT	/team/{id}	Modificar un equipo. Solamente se puede modificar su nombre de equipo. Si se realiza correctamente, la respuesta será <i>200 OK</i> Si el artista a modificar no existe, la respuesta será <i>404 Not Found</i> .
DELETE	/team/{id}	Borrar un equipo. Si se realiza correctamente, debe devolver <i>204 No Content</i> .

HTTP	URL	Descripción
POST	/post/	Añade un nuevo artista. Si se añade satisfactoriamente, devuelve <i>201 Created</i> Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i>
GET	/post/	Obtener una lista de todos los post. Código de respuesta 200 (o 404 si no hay ninguna)
GET	/post/{id}	Obtener un post por su ID. Código de respuesta 200 (o 404 si no se encuentra)
POST	/post/{id}/comment/{id2}	Añade un comentario a un post existente. Si se realiza correctamente, la respuesta será <i>200 OK</i> Si el artista a modificar no existe, la respuesta será <i>404 Not Found</i> .
DELETE	/post/{id}	Borrar un post. Si se realiza correctamente, debe devolver <i>204 No Content</i> .

HTTP	URL	Descripción
POST	/league/	Añade una nueva liga Si se añade satisfactoriamente, devuelve <i>201 Created</i> Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i>
POST	/league/{id}/team/{id2}	Añade un equipo a un equipo Si se añade satisfactoriamente, devuelve <i>201 Created</i> Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i>
GET	/league/	Obtener una lista de todas las ligas. Código de respuesta 200 (o 404 si no hay ninguna)
GET	/league/{id}	Obtener una liga por su ID Código de respuesta 200 (o 404 si no se encuentra)
PUT	/league/{id}	Modificar una liga. Solamente se puede modificar el nombre de la liga. Si se realiza correctamente, la respuesta será <i>200 OK</i> Si el artista a modificar no existe, la respuesta será <i>404 Not Found</i> .

DELETE	/league/{id}	<p>Borrar una liga.</p> <p>Las canciones de este artista no se deben borrar si se borra un artista.</p> <p>Si se realiza correctamente, debe devolver <i>204 No Content</i>.</p>
--------	--------------	--

HTTP	URL	Descripción
POST	/comment/	<p>Añade un nuevo comentario</p> <p>Si se añade satisfactoriamente, devuelve <i>201 Created</i></p> <p>Si hay algún error en los datos que nos envían, debe devolver <i>400 Bad Request</i></p>
GET	/comment/	<p>Obtener una lista de todos los comentarios.</p> <p>Código de respuesta 200 (o 404 si no hay ninguna)</p>
GET	/comment/{id}	<p>Obtener un comentario por su ID</p> <p>Código de respuesta 200 (o 404 si no se encuentra)</p>

# Implementación

En la carpeta uploads se guardarán los ficheros que se suban a la aplicación.

En la carpeta main se encontrarán dos carpetas una main y otra testing, en la de testing se harán las operaciones relacionadas con el testing mientras que en la carpeta main encontraremos todas las carpetas relacionadas la aplicación dentro de esta carpeta encontraremos una carpeta llamada resources en el que se encontraran las propiedades de la aplicación, el archivo sql con los datos que se van a cargar al arrancar el programa, dentro de esta encontraremos una carpeta llamada messages en el que podremos tener un archivo con los mensajes de error que nos encontraremos. Si nos vamos antes de entrar en la carpeta resources hay una carpeta java con el nombre de nuestro proyecto en el que dentro de este encontraremos:

- comment: en esta carpeta tendremos todo lo que esté relacionado con los comentarios.
- config: en esta carpeta tendremos archivos de configuración relacionados con la seguridad del proyecto
- error: en esta carpeta habrá archivos que estén relacionados con los errores que tengamos en el proyecto.
- exception: en esta carpeta contendremos archivos con mensajes en el que si sucede una excepción en el programa pues nos notificará con un mensaje predefinido.
- files: en esta carpeta se encontrará todo lo que esté relacionado con la subida de ficheros a nuestra aplicación.
- league: en esta carpeta tendremos todo lo que esté relacionado con las ligas.
- post: en esta carpeta tendremos todo lo que esté relacionado con los posts.
- search: en esta carpeta se recogerán las búsquedas para hacer consultas.
- security: en esta carpeta se guardarán todos los archivos relacionados con la seguridad.
- team: en esta carpeta tendremos todo lo que esté relacionado con los equipos.
- user: en esta carpeta tendremos todo lo que esté relacionado con los usuarios.
- utils: en esta carpeta tendremos un archivo para separar los roles.
- validation: en esta carpeta tendremos todo lo que este con la validation en nuestro proyecto.