



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación web para la
gestión de clubs deportivos y red social para
sus miembros (Back-End)

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: José María Viu Domínguez

Tutor: Félix Buendía García

[2015-2016]

Resumen

La elaboración de Yulava nace con el objetivo de satisfacer las necesidades administrativas de clubs deportivos, facilitando la gestión deportiva y poniendo a disposición de todos los usuarios información relevante, así como una red social que fomenta la participación de todos los miembros.

Se ha creado una aplicación web usando Codeigniter, un framework que utiliza el patrón modelo-vista-controlador, haciendo especial hincapié en el desarrollo Back-End de la aplicación.

Tras la puesta en marcha de la aplicación, se ha comprobado que el uso de la misma ha crecido exponencialmente, ya que es una herramienta útil y de fácil manejo, gracias a su diseño adaptable a todas las resoluciones de pantalla.

Palabras clave: Aplicación web, Gestión deportiva, Deportes, Codeigniter, Framework, Red social, Responsive.

Abstract

The creation of Yulava born with the objective of satisfying the administrative needs of the sports clubs, making the sporting management easy and making available the relevant information to all users, as well as providing a social network which promotes the participation of all team members.

A web application has been created using Codeigniter, a framework that uses the Model-view-controller model, emphasizing the Back-End development of the application.

The exponential growth of the application's use has been observed after its launch because, thanks to its adaptable design to all screen resolutions, it is a really useful and easy tool.

Keywords : Web Application, Sporting management, Sports, Codeigniter, Framework, Social network, Responsive.



Resum

L'elaboració de Yulava naix amb l'objectiu de satisfer les necessitats administratives de clubs esportius, facilitant la gestió esportiva i posant a disposició de tots els usuaris informació rellevant, així com una xarxa social que fomenta la participació de tots els membres.

S'ha creat una aplicació web usant Codeigniter, un framework que utilitza el patró model-vista-controlador, fent especial insistència en el desenvolupament back-end de l'aplicació.

Després de la posada en marxa de l'aplicació, s'ha comprovat que l'ús de la mateixa ha crescut exponencialment, ja que és una ferramenta útil i de fàcil maneig, gràcies al seu disseny adaptable a totes les resolucions de pantalla.

Paraules clau: Aplicació web, Gestió esportiva, Esports, Codeigniter, Framework, Xarxa social, Responsive.

Tabla de contenidos

1. Introducción	9
1.1. Objetivos	9
1.2. Contexto.....	9
1.3. Ámbito de la aplicación	9
1.4. Trabajo realizado	10
2. Especificación de requisitos	11
2.1. Introducción	11
2.1.1. Propósito	11
2.1.2. Alcance	11
2.1.3. Definiciones, siglas y abreviaciones.....	11
2.1.4. Visión global	13
2.2. Descripción general.....	13
2.2.1. Perspectiva del producto	13
2.2.2. Funciones del producto	14
2.2.3. Características de los usuarios	14
2.2.4. Restricciones	15
2.2.5. Supuestos y dependencias.....	16
2.3. Requisitos específicos	16
2.3.1. Requisitos de las interfaces externas.....	16
2.3.2. Requisitos funcionales	17
2.3.3. Requisitos de rendimiento	38
2.3.4. Restricciones del diseño.....	38
2.3.5. Atributos del sistema software	39
3. Análisis.....	41
3.1. Introducción	41
3.2. Diagramas de casos de uso.....	41
3.2.1. Actores de la aplicación	41
3.2.2. Administrador	42
3.2.3. Usuario entrenador	47
3.2.3. Usuario jugador.....	49



3.2.4. Usuario padre/madre.....	51
3.2.5. Usuario invitado.....	53
3.2.6. Usuario no registrado.....	55
3.3. Diagrama de clases.....	57
3.4. Diagramas de flujo para la gestión de equipos	58
3.5. Diagramas de flujo para la gestión de la web.....	59
4. Diseño	60
4.1. Introducción	60
4.2. Arquitectura del sistema	60
4.3. Modelo.....	61
4.4. Vista	61
4.5. Controlador	62
5. Implementación.....	64
5.1 Introducción	64
5.2 Tecnologías	64
5.2.1. Codeigniter	64
5.2.1.1. Language	64
5.2.1.2. Form Validation.....	65
5.2.1.3. Active record.....	66
5.2.1.4. IonAuth.....	66
5.2.1.5. Sesión.....	67
5.2.1.6. Routes	67
5.2.2. PHP5.....	68
5.2.3. MySQL.....	68
5.2.4. JSON	68
5.2.5. Conexión Modelo - Vista - Controlador	70
5.3. Herramientas y entorno	71
5.3.1. Entorno de desarrollo	71
5.3.1.1. Eclipse	71
5.3.1.2. Sybase Powerdesigner	71
5.3.1.3. PhpMyAdmin.....	72
5.3.2. Control de versiones.....	73
5.3.2.1. GIT.....	73
5.3.2.2. Bitbucket.....	73
5.3.3. Servidor web.....	73
6. Evaluación	74

6.1. Introducción	74
6.2. Pruebas funcionales.....	74
6.3. Seguridad del sistema.....	77
6.4. Rendimiento del segmento de usuarios.	77
6.4.1. Visitas	77
6.4.2. Dispositivos.....	78
6.4.3 Flujo de usuarios.....	79
7. Conclusiones	80
7.1 Valoraciones.....	80
7.2. Trabajo futuro.....	81
8. Bibliografía	82



1. Introducción

1.1. Objetivos

Los objetivos que se pretenden alcanzar tras la realización de este proyecto son:

- Crear una aplicación web (Yulava) destinada a la gestión de clubes deportivos.
- Desarrollar una red social dentro de la misma aplicación para que pueda ser utilizada por los usuarios del club.
 - Fomentar la interacción entre los miembros del club a través de cualquier terminal con conexión a internet, ya sea un dispositivo móvil o un ordenador.
- Elaborar un repositorio de clubes deportivos.

1.2. Contexto

El desarrollo de aplicaciones web, tanto para ordenador como para dispositivos móviles, se está viendo incrementado de forma sustancial en estos últimos años. El acceso de la informática a la población ha supuesto una gran evolución en la creación de aplicaciones web, como consecuencia de la demanda de éstas por parte de los usuarios que, cada vez con más frecuencia, buscan recursos para solventar sus problemas diarios.

Actualmente existen infinitas aplicaciones que nos permiten desde hacer la compra o leer las noticias hasta establecer relaciones sociales de forma online. En este contexto nace Yulava, una aplicación web destinada a la gestión de clubes deportivos que además incorpora una red social para todos sus miembros.

1.3. Ámbito de la aplicación

El ámbito de esta aplicación se centra en la gestión de clubes deportivos. Entre sus características destaca el control de la contabilidad del club, creación y seguimiento de estadísticas tanto a nivel individual (jugador) como colectivo (equipo), así como almacenar todas las localizaciones de las instalaciones deportivas, incluyendo sus respectivas coordenadas de geolocalización, para posteriormente ser asignadas a un partido.

Todo usuario registrado en Yulava tiene por defecto asignado el rol de invitado. Además de este rol, también se pueden asignar otros permisos al usuario dependiendo del estatus que éste posea dentro del club. Estos roles son:

- Administrador: Tiene control total sobre el club.
- Entrenador: Tiene control total sobre el equipo que entrena.
- Jugador: Puede ser asignado a uno o varios equipos. El jugador posee un perfil con sus datos en los que aparecen estadísticas personales.
- Padre/Madre: Puede ser asignado a uno o varios equipos con la finalidad de estar informado de las últimas novedades del equipo al que ha sido asignado.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Los roles anteriores solo tienen valor dentro del club al que pertenece el usuario, por lo que si un miembro intenta acceder a otro club sólo podrá ver la parte pública del mismo.

Para asegurar que personas ajenas al club se registren, se ha creado un código de seguridad, que será necesario introducirlo a la hora de darse de alta. Además, para disminuir la posibilidad de fallo, el administrador del club debe activar al usuario después de que éste se haya registrado. En caso de detectar registros no válidos, el administrador puede eliminar dicha suscripción.

Una vez el usuario ha sido activado por un administrador del club, puede empezar a interactuar con Yulava. Ver partidos, leer noticias o participar en eventos son algunas de las características que han sido desarrolladas en Yulava.

Además de todas las características anteriormente mencionadas, el administrador del club, tiene la opción de crear noticias que pueden ser públicas (toda persona tiene acceso a ellas y pueden ser compartidas en distintas redes sociales) o privadas (solo las personas con acceso en el club pueden visualizarlas), crear eventos en los que se puede participar siempre que el rol del usuario esté permitido por el evento o crear simples páginas de información que serán incluidas en el menú principal.

1.4. Trabajo realizado

El proyecto consta de dos partes; el Front-End que es la parte visual de la web, con la que interactúa el usuario final y el Back-End que es la parte encargada del tratamiento de datos y funcionalidad de la web.

Este proyecto se ha realizado en pareja, dividiendo el trabajo en los bloques anteriormente mencionados y siguiendo una estructura MVC. Para mantener el código sincronizado y tener ambos la última versión del código, se ha utilizado un repositorio web (Bitbucket), adoptando la tecnología GIT para controlar las últimas versiones añadidas en el repositorio. Asimismo el código ha sido desarrollado con el programa Eclipse.

Por otro lado, la descripción de esta memoria se centra en el desarrollo Back-End de la aplicación. Este desarrollo incluye la creación de la base de datos, la programación del controlador y del modelo, así como todo el control de permisos que se realiza en la aplicación web para que la lógica sea la correcta.

Se menciona también, la comunicación con mi compañero, ya que es de vital importancia que la vista este sincronizada con el controlador.

2. Especificación de requisitos

2.1. Introducción

2.1.1. Propósito

En la presente memoria se formaliza la especificación de requisitos, indicando todas las funcionalidades que van a ser implementadas en la aplicación, diferenciando los distintos casos de uso dependiendo del tipo de usuario.

Concretamente se han definido seis actores diferentes: administrador, entrenador, jugador, padre/madre, invitado y el usuario no registrado. Los usuarios registrados tendrán la posibilidad de ejercer más de un permiso simultáneamente.

2.1.2. Alcance

Originalmente, el proyecto fue pensado para satisfacer las necesidades de un club de fútbol sala. La idea principal fue crear un gestor deportivo, facilitando así el funcionamiento interno del club. Como valor añadido se propuso crear una red social que estuviera incluida en la misma aplicación, para que, de esta forma todos los miembros del club pudieran participar activamente en la plataforma y fueran conocedores de la gestión del club.

Posteriormente, tras asentar las bases del proyecto, se decidió que la aplicación tuviera soporte para más de un club y de este modo que pudiera ser utilizada en cualquier tipo de deporte colectivo.

Con este proyecto se pretende conseguir que cualquier club pueda tener acceso a todas las funcionalidades disponibles, así como conseguir una interfaz intuitiva y fácil de usar. La finalidad es que un perfil de usuario no experto sea capaz de sacar todo el potencial de la aplicación, teniendo en cuenta que pueden existir usuarios de todas las edades.

Por último, observando cómo ha cambiado el consumo de internet en los últimos años, un aspecto fundamental es la adaptabilidad del diseño de la aplicación a cualquier dispositivo con conexión a internet.

2.1.3. Definiciones, siglas y abreviaciones

A continuación se definirán todos los términos, siglas y abreviaturas utilizadas en la especificación de requisitos:

- Yulava: Nombre que recibe la aplicación web realizada en este proyecto.
- Facebook: Nombre de una popular red social.
- Twitter: Nombre de una popular red social
- API: Interfaz de programación de aplicaciones ^[1], se define como un conjunto de subrutinas, funciones y procedimientos.
- Google Maps^[2]: Aplicación web desarrollada por la compañía estadounidense Google, y que ofrece la posibilidad de desplazarte por los mapas de todo el mundo.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

- HTML^[3]: (Lenguaje de marcas de hipertexto), es un lenguaje de marcado para la elaboración de páginas web.
- CSS^[4]: Es un lenguaje usado para definir y crear la presentación de un documento escrito en el lenguaje HTML
- JavaScript^[5]: Es un lenguaje de programación orientado a objetos que principalmente es usado en el lado del cliente
- Front-End: Es la parte visible de la aplicación y la que interactúa con el usuario final.
- Back-End: Es la parte interna de la aplicación que se encarga del acceso y manipulación de los datos.
- PHP: Es un lenguaje de programación orientado a objetos que principalmente es usado en el lado del cliente.
- BBDD: Siglas del término base de datos.
- SQL: Es un lenguaje de programación que se encarga de operar con la base de datos.
- Responsive: Es una forma de diseñar y desarrollar las vistas de una aplicación web y cuyo objetivo es adaptar su apariencia al tamaño de pantalla que se está utilizando.
- GB: Siglas del término inglés gigabyte. Es una unidad de almacenamiento.
- RAM: Siglas del término memoria de acceso aleatorio. Se encarga de cargar las instrucciones que ejecuta el procesador.
- PDF: Siglas del término formato de documento portátil. Es un tipo de formato que es utilizado para el almacenamiento de documentos digitales.
- Framework^[6]: Su principal objetivo es ofrecer una serie de librerías y funciones que son habitualmente utilizadas en el desarrollo de una aplicación. Se construyen utilizando un patrón de diseño.
- Codeigniter^[7]: Es un Framework utilizado para la creación de aplicaciones web.
- SEO^[8]: es el proceso técnico mediante el cual se realizan cambios en la estructura e información de una página web, con el objetivo de mejorar la visibilidad de un sitio web en los resultados orgánicos de los diferentes buscadores.
- CRON: Es una aplicación que ejecuta una acción durante un intervalo de tiempo y que anteriormente ha sido programada.

2.1.4. Visión global

Este apartado se va a dividir en dos subapartados:

- Descripción general: Donde se indicará la perspectiva y las funciones del producto, las características de usuario, las restricciones detectadas, las dependencias y los requisitos futuros.
- Requisitos específicos: Se describe con detalle los requisitos de interfaz externo, los funcionales, los de rendimiento, así como las restricciones de diseño y los atributos de calidad que debe tener la aplicación.

2.2. Descripción general

2.2.1. Perspectiva del producto

El proceso previo para la realización de Yulava fue la búsqueda de información, acerca de productos con características similares a las definidas en el proyecto, a nivel nacional.

Una de las primeras webs es futbolvista^[9]. Su objetivo principal es ofrecer a los entrenadores de un club deportivo la posibilidad de gestionar sus entrenamientos, partidos y estadísticas. Asimismo incluye un acceso específico para que los padres puedan consultar calendarios, informes o información del equipo. Todo esto en forma de intranet, lo que hace que no sea un diseño muy intuitivo para una persona que no está acostumbrada a este tipo de aplicación.

Otra web del sector sería miclub-manager^[10], encargada de dar soporte a los entrenadores del club con un diseño más atractivo ya que dispone de aplicación móvil. En este caso, los padres no tendrían acceso a la web teniendo de este modo menos impacto social en el club.

En ambos casos se trata de aplicaciones de pago y para la creación del club debes contactar con la empresa, limitando de este modo la expansión del producto.

Por otro lado existe myequipo^[11]. Esta web no es un gestor deportivo, sino que ofrece una herramienta para la creación de páginas web personalizadas. Dispone de un servicio gratuito donde el dominio de la página web sería “www.myequipo.com/nombre_del_club” y otra versión de pago donde se utilizaría un dominio propio.

La principal diferencia entre los productos ya existentes y Yulava es la apuesta por la interacción de todos los miembros con la aplicación en forma de red social, fomentando así la transparencia y la participación de todos los usuarios.

Un aspecto destacable que ofrece Yulava es la disponibilidad de una serie de herramientas cuyo objetivo es gestionar el club de una manera sencilla y rápida; se ofrece un gestor de cuentas, organización de equipos, creación de partidos y almacenamiento de estadísticas tanto por jugador como por equipo.

Asimismo, es imprescindible nombrar el diseño amigable de la web. Su aspecto, que recuerda a las redes sociales más utilizadas en la actualidad (Facebook, Twitter), ayuda a su fácil manejo.

Cabe mencionar que su uso no está limitado únicamente al fútbol, sino que sus funciones son ampliables a otros deportes como fútbol sala, baloncesto, hockey y/o balonmano.

2.2.2. Funciones del producto

Nuestra aplicación web dispondrá de las siguientes funciones:

- **Gestión de miembros:** Donde se podrán gestionar todos los miembros del club, ya sean administradores, jugadores, entrenadores o padres/madres.
- **Registro de usuarios:** Los propios usuarios podrán crearse una cuenta del club, que el administrador deberá validar antes de completar el registro.
- **Gestión de equipos:** Se podrán crear, modificar y eliminar equipos, incluyendo los miembros, competiciones y rivales de cada equipo
- **Gestión de partidos:** Se podrán gestionar los partidos, indicando el lugar del partido, la lista de convocados, la fecha y la hora.
- **Gestión de instalaciones:** Permitirá crear, editar y eliminar las instalaciones donde se jugarán los partidos. Todo ello geolocalizado utilizando la API Google Maps.
- **Gestión de noticias:** En la aplicación, los miembros con permisos suficientes podrán crear y editar noticias.
- **Gestión de eventos:** En la aplicación, los miembros con permisos suficientes podrán crear, editar y participar en eventos del club.
- **Añadir y visualizar estadísticas:** Los entrenadores o administradores podrán añadir estadísticas de los jugadores en cada partido para luego ser visualizadas.
- **Gestión de cuentas:** En la aplicación existirá un apartado donde poder llevar la contabilidad del club, indicando los gastos e ingresos.

2.2.3. Características de los usuarios

Cada usuario de la aplicación dentro de un club puede disponer de más de un rol a la vez. Esto tiene sentido ya que un jugador de un equipo de la máxima categoría puede ser a la vez un entrenador de un equipo de categorías inferiores.

El sistema dispone de siete tipos de usuarios:

- **Usuario registrado:** Es el usuario que está registrado en la aplicación y por tanto tiene alguno de los roles asociados. Este tipo de usuario es una generalización de todos los tipos de usuarios que necesiten registro: administrador, entrenador, jugador, padre/madre e invitado.
- **Usuario administrador:** Es el usuario encargado de la gestión interna del club. Puede administrar los usuarios, equipos, partidos, noticias y eventos sin limitaciones. El usuario administrador es una persona adulta y forma parte de la

directiva o el cuerpo técnico del club deportivo. Tiene conocimientos básicos de navegar por internet, con un nivel de educación medio y no necesita tener conocimientos técnicos.

- **Usuario entrenador:** Es el usuario que forma los equipos. Un entrenador puede gestionar más de un equipo y tiene permisos para crear los partidos de sus equipos. Es una persona adulta, tiene un conocimiento medio de navegar por internet, un nivel de educación medio y no necesita tener conocimientos técnicos.
- **Usuario jugador:** Es el usuario que pertenece a uno o más equipos como jugador se le podrán insertar estadísticas individuales de los partidos. El usuario es una persona joven, con altos conocimientos en navegar por internet y sin necesidad de tener conocimientos técnicos.
- **Usuario padre/madre:** Es el usuario vinculado a un jugador, este usuario podrá visualizar datos de la aplicación. Este rol está pensado para que el padre o madre realice un seguimiento de los hijos. Este usuario es una persona adulta con conocimientos básicos de navegar por internet.
- **Usuario invitado:** Es el usuario creado por defecto al registrarse en un club. Este usuario podrá visualizar datos de la aplicación pero no podrá ser añadido a ningún equipo. Está pensado para usuarios que no tienen un rol determinado en el club. Este usuario es una persona adulta o joven con conocimientos básicos de navegar por internet.
- **Usuario no registrado:** Es el usuario que no necesita crearse una cuenta. Dicho usuario podrá visualizar solo los datos públicos del club. Este usuario es una persona adulta o joven con conocimientos básicos de navegar por internet.

2.2.4. Restricciones

Las restricciones para un buen uso de la aplicación son escasas. Únicamente es necesario un navegador web para hacer uso de la aplicación. Dicho navegador podrá ser ejecutado en cualquier tipo de dispositivo ya sea ordenador, Tablet o móvil.

La aplicación tiene que ser escalable, ya que tiene que ser capaz de crecer si el número de usuarios aumenta significativamente.

Los lenguajes utilizados para la aplicación se restringirán a HTML, CSS y JavaScript para la parte del Front-End. Para la parte de Back-End se utilizará PHP y para el acceso a los datos de la BBDD se utilizará SQL.

La aplicación debe cumplir la Ley Orgánica 15/1999, de 13 de diciembre ^[12], de Protección de Datos de Carácter Personal. Asimismo, para cumplir la ley anteriormente descrita, se restringirá el registro de usuarios menores de 14 años, salvo que aporten una autorización del padre madre o tutor legal.

Como restricción de servicios en red utilizados por la aplicación, se encuentra el límite de uso de los mapas de Google Maps, que limita el número de consultas de su API a 2500 solicitudes por día y 10 por segundo.



2.2.5. Supuestos y dependencias

En caso de cambios en la ley de protección de datos, deberemos realizar modificaciones para cumplir con todos los términos de dicha ley.

Como restricción de otros servicios en red, tenemos a Google Maps. Al ser un servicio externo a nosotros tenemos el riesgo de cambios en las políticas de uso o en la sintaxis de utilización del API, de esta manera debemos estar al corriente de cualquier cambio. También existe la posibilidad de cambios en el límite de consultas a su API, en este caso nos veremos obligados a contratar el servicio de Google Maps API for Work^[13].

En el supuesto de aumentar significativamente el número de usuarios y el tráfico en la web, tendremos que utilizar un servidor superior que pueda dar servicio a todos los clientes sin perjudicar el rendimiento.

2.3. Requisitos específicos

2.3.1. Requisitos de las interfaces externas

2.3.1.1. Requisitos hardware

El requisito hardware para esta aplicación es tener todos los componentes necesarios para poder ejecutar un navegador web. Hoy en día dichos componentes son comunes en todos los dispositivos por lo cual se estaría afirmando que no existe un requisito hardware, ya que cualquier dispositivo actual podrá ejecutar un navegador web.

2.3.1.2. Requisitos de software

Como requisito de software la aplicación web deberá ser compatible con las últimas versiones de navegadores, para ellos cumpliremos con todos los estándares establecidos por W3C.

2.3.1.3. Requisitos de usuario

La interfaz de usuario deberá ser fácil e intuitiva, para ello la aplicación tendrá un diseño web centrado en el usuario, donde toda la información esté bien estructurada. El uso de migas de pan ayudará a tener una mejor accesibilidad.

Además la aplicación deberá ser rápida para que el usuario tenga una buena experiencia de uso.

Otro apartado importante es la usabilidad en dispositivos móviles y Tablet. Para suplir este requisito nuestra web será responsive, de este modo se adaptará a cualquier tamaño de pantalla.

2.3.2. Requisitos funcionales

En este apartado definiremos los diferentes requisitos funcionales que tendrá el sistema, identificando todas sus funcionalidades.

Para definir dichos requisitos utilizaremos un esquema donde indicaremos el nombre del requisito, las entradas y el proceso que desencadena la acción de entrada y la salida, que es el resultado del proceso.

No todos los siguientes casos de uso pueden ser realizados por todos los tipos de usuario. En el apartado de análisis, se utilizarán diagramas de casos de uso para definir que quien puede realizar cada acción en la aplicación.

2.3.2.1. Gestión de noticias

Nombre	Crear noticias
Introducción	El usuario puede crear noticias en el club.
Entradas	Contenido de la noticia.
Proceso	El usuario se dirige al apartado de crear noticia, rellena el título de la noticia, selecciona si es pública o privada y redacta la noticia. Cuando tenga la noticia preparada pulsa el botón de guardar y el sistema, entonces, realiza una comprobación de los datos y guarda la noticia en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar noticia
Introducción	El usuario puede editar noticias ya creadas en el club.
Entradas	Identificador de la noticia.
Proceso	El usuario se dirige al apartado de administración de noticias y selecciona la noticia que desea editar. El usuario puede modificar el título, la privacidad o el contenido de la noticia. Una vez realizadas las modificaciones el usuario pulsa el botón de guardar. El sistema realiza una comprobación de los datos y sobrescribe la noticia en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Eliminar noticias
Introducción	El usuario puede eliminar noticias del club.
Entradas	Identificador de la noticia.
Proceso	<p>El usuario se dirige al apartado de administración de noticias y pulsa el botón de eliminar en la noticia deseada.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema comprueba si el usuario tiene permisos para realizar el borrado y si es afirmativo elimina la noticia de la base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver noticia
Introducción	El usuario puede ver noticias del club.
Entradas	Identificador de la noticia.
Proceso	<p>El usuario se dirige al apartado de administración de noticias y pulsa el botón de ver en la noticia deseada. También podría ver las noticias desde la página inicial del club.</p> <p>El sistema redirige al usuario a la noticia seleccionada. En el caso de ser una noticia pública el usuario podría compartir la noticia en redes sociales.</p>
Salidas	Redirección a la noticia seleccionada por el usuario.

2.3.2.2. Gestión de miembros

Nombre	Crear usuarios
Introducción	El usuario puede crear nuevos usuarios.
Entradas	Datos del nuevo usuario.
Proceso	<p>El usuario se dirige al apartado de crear usuarios, rellena los datos del usuario, selecciona los permisos que tendrá el nuevo usuario (administrador, entrenador, jugador, padre/madre) y por último pulsa el botón de crear.</p> <p>El sistema realiza una comprobación de datos, para detectar posibles errores. Si todo es correcto crea al nuevo usuario y devuelve un mensaje satisfactorio. En caso contrario devuelve un mensaje de error.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar usuarios
Introducción	El usuario puede editar usuarios del club.
Entradas	Identificador del usuario.
Proceso	<p>El usuario se dirige al apartado de administración de usuarios y selecciona el usuario que desea editar.</p> <p>Una vez realizadas las modificaciones el usuario pulsa el botón de guardar.</p> <p>El sistema realiza una comprobación de los datos y sobrescribe los datos del usuario en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Eliminar usuarios
Introducción	El usuario puede eliminar usuarios.
Entradas	Identificador del usuario.
Proceso	<p>El usuario se dirige al apartado de administración de usuarios y pulsa el botón de eliminar en el usuario deseado.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema comprueba si el usuario tiene permisos para realizar la eliminación del usuario y si es afirmativo elimina al usuario de la base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver usuarios
Introducción	El usuario puede ver los perfiles de los usuarios del club.
Entradas	Identificador del usuario.
Proceso	<p>El usuario se dirige al apartado de administración de usuarios y pulsa el botón de ver en el usuario indicado. También podría ver el listado de usuarios en el apartado de miembros del club.</p> <p>El sistema redirige al usuario al perfil de usuario.</p>
Salidas	Redirección al perfil de usuario seleccionado.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Activar/desactivar usuarios
Introducción	El usuario puede activar y desactivar.
Entradas	Identificador del usuario.
Proceso	El usuario se dirige al apartado de administración de usuarios y selecciona el usuario que desea activar o desactivar. Dentro del usuario seleccionado pulsa el botón de activar o desactivar el usuario. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Eliminar roles
Introducción	El usuario puede eliminar roles a los usuarios.
Entradas	Identificador del usuario.
Proceso	El usuario se dirige al apartado de administración de usuarios y selecciona al usuario al que desea eliminar un rol. En el formulario indica qué roles desea eliminar y pulsa el botón de guardar. El sistema realiza una comprobación de los datos y sobrescribe los datos del usuario en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Añadir roles
Introducción	El usuario puede añadir roles a los usuarios.
Entradas	Identificador del usuario.
Proceso	El usuario se dirige al apartado de administración de usuarios y selecciona al usuario al que desea añadir un rol, En el formulario indica qué roles le desea y pulsa el botón de guardar. El sistema realiza una comprobación de los datos y sobrescribe los datos del usuario en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

2.3.2.3. Gestión de eventos

Nombre	Crear eventos
Introducción	El usuario es capaz de crear eventos en el club.
Entradas	Contenido del evento.
Proceso	<p>El usuario se dirige al apartado de crear evento, rellena el título del evento, selecciona los tipos de miembros que pueden participar y redacta el cuerpo del evento.</p> <p>Cuando tenga el evento preparado pulsa el botón de guardar y el sistema, realiza una comprobación de los datos y guarda el evento en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar eventos
Introducción	El usuario puede editar eventos del club.
Entradas	Identificador del evento.
Proceso	<p>El usuario se dirige al apartado de administración de eventos y selecciona el evento que desea editar.</p> <p>El usuario puede modificar el título, los usuarios que pueden participar o el contenido del evento.</p> <p>Una vez realizadas las modificaciones el usuario pulsa el botón de guardar.</p> <p>El sistema realiza una comprobación de los datos y sobrescribe el evento en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver eventos
Introducción	El usuario puede ver los eventos del club.
Entradas	Identificador del evento.
Proceso	<p>El usuario se dirige al apartado de administración de eventos y pulsa el botón de ver en el evento deseado. También podría ver los eventos desde el apartado eventos del club.</p> <p>El sistema redirige al usuario al evento seleccionado.</p>
Salidas	Redirección al evento seleccionado por el usuario.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Participar en eventos
Introducción	El usuario puede participar en los eventos en los que su rol está permitido
Entradas	Identificador del evento.
Proceso	<p>El usuario se dirige al apartado de administración de eventos y pulsa el botón de ver en el evento deseado. También podría ver los eventos desde el apartado eventos del club.</p> <p>Una vez dentro del evento, pulsa el botón de participar.</p> <p>El sistema añade al usuario a la lista de participantes del evento.</p>
Salidas	El usuario aparece en la lista de participantes.

Nombre	Eliminar eventos
Introducción	El usuario puede eliminar eventos del club.
Entradas	Identificador del evento.
Proceso	<p>El usuario se dirige al apartado de administración de eventos y pulsa el botón de eliminar en el evento deseado.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema comprueba si el usuario tiene permisos para realizar el borrado y si es afirmativo elimina el evento de la base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

2.3.2.4. Gestión del club

Nombre	Editar club
Introducción	El usuario puede editar los datos del club.
Entradas	Identificador del club.
Proceso	<p>El usuario se dirige al apartado de preferencias del club.</p> <p>El usuario puede modificar todos los datos del club.</p> <p>Una vez realizadas las modificaciones el usuario pulsa el botón de guardar.</p> <p>El sistema realiza una comprobación de los datos y sobrescribe la información del club en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar redes sociales
Introducción	El usuario puede editar las redes sociales del club
Entradas	Identificador del club.
Proceso	<p>El usuario se dirige al apartado de preferencias del club.</p> <p>El usuario puede ver los links de las diferentes redes sociales.</p> <p>Una vez realizadas las modificaciones el usuario pulsa el botón de guardar.</p> <p>El sistema realiza una comprobación de los datos y sobrescribe la información de las redes sociales del club en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar logo
Introducción	El usuario puede cambiar el logo o escudo del club
Entradas	Identificador del club.
Proceso	<p>El usuario pulsa el botón de cambiar imagen del logo en la cabecera del club. Se abre un diálogo que permite subir y recortar una imagen.</p> <p>Una vez seleccionada pulsa el botón subir.</p> <p>El sistema realiza una comprobación de la imagen (tamaño, pixeles). El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	<p>Mensaje satisfactorio o mensaje de error en caso de fallo.</p> <p>Cambia la imagen del logo.</p>

Nombre	Editar cabecera
Introducción	El usuario puede cambiar la imagen de cabecera del club
Entradas	Identificador del club.
Proceso	<p>El usuario pulsa el botón de cambiar imagen en la cabecera del club. Se abre un diálogo que le permite subir y recortar una imagen.</p> <p>Una vez seleccionada pulsa el botón subir.</p> <p>El sistema realiza una comprobación de la imagen (tamaño, pixeles). El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	<p>Mensaje satisfactorio o mensaje de error en caso de fallo.</p> <p>Cambia la imagen de cabecera.</p>

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Crear páginas del club
Introducción	El usuario puede crear páginas del club
Entradas	Identificador del club y contenido de la página.
Proceso	<p>El usuario se dirige al apartado de administración de páginas y pulsa el botón de crear.</p> <p>Pone un título a la página y escribe el contenido que desee.</p> <p>Pulsa el botón de guardar, el sistema realiza una comprobación de los datos y guarda la información de la página en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar páginas del club
Introducción	El usuario puede editar las páginas del club
Entradas	Identificador del club y contenido de la página.
Proceso	<p>El usuario se dirige al apartado de administración de páginas y pulsa el botón de editar sobre la página deseada.</p> <p>Modifica el título o el contenido de la página.</p> <p>Pulsa el botón de guardar, el sistema realiza una comprobación de los datos y sobrescribe la información de la página en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Eliminar páginas del club
Introducción	El usuario puede eliminar las páginas del club
Entradas	Identificador del club e identificador de la página.
Proceso	<p>El usuario se dirige al apartado de administración de páginas y pulsa el botón de eliminar sobre la página deseada.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema comprueba si el usuario tiene permisos para realizar el borrado y si es afirmativo elimina la página del sistema.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver páginas del club
Introducción	El usuario puede ver las páginas del club
Entradas	Identificador del club e identificador de la página.
Proceso	El usuario se dirige al apartado de administración de páginas y pulsa el botón de ver sobre la página deseada. También podría acceder desde el apartado de páginas del club. El sistema redirige al usuario a la página seleccionada.
Salidas	Redirección a la página seleccionada por el usuario.

2.3.2.5. Gestión de instalaciones

Nombre	Añadir instalaciones
Introducción	El usuario puede crear instalaciones.
Entradas	Nombre y localización.
Proceso	El usuario se dirige al apartado de crear instalación, rellena el nombre de la instalación, selecciona si es una instalación propia o del rival y selecciona la localización en un mapa. Pulsa el botón de guardar y el sistema, realiza una comprobación de los datos y guarda la instalación en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar instalaciones
Introducción	El usuario puede editar instalaciones.
Entradas	Identificador de la instalación.
Proceso	El usuario se dirige al apartado de administración de instalaciones y selecciona la instalación que desea editar. El usuario puede modificar el nombre, la propiedad de la instalación o la localización. Una vez realizadas las modificaciones el usuario pulsa el botón de guardar. El sistema realiza una comprobación de los datos y sobrescribe la instalación en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Eliminar instalaciones
Introducción	El usuario puede eliminar instalaciones.
Entradas	Identificador de la instalación.
Proceso	<p>El usuario se dirige al apartado de administración de instalaciones y pulsa el botón de eliminar en la instalación deseada.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema comprueba si el usuario tiene permisos para realizar el borrado y si es afirmativo elimina la instalación de la base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver instalaciones
Introducción	El usuario puede ver las instalaciones creadas.
Entradas	Identificador de la instalación.
Proceso	<p>El usuario se dirige al apartado de administración de instalaciones y pulsa el botón de ver en la instalación deseada. También podría ver las instalaciones propias del club desde la página de instalaciones.</p> <p>El sistema redirige al usuario a la página de la instalación seleccionada.</p>
Salidas	Redirección a la instalación seleccionada por el usuario.

2.3.2.6. Gestión de cuentas

Nombre	Añadir movimientos bancarios
Introducción	El usuario puede añadir movimientos bancarios.
Entradas	Tipo de movimiento, miembro y cantidad.
Proceso	<p>El usuario se dirige al apartado de administración de cuentas. Selecciona el tipo de movimiento (gasto o ingreso), selecciona un miembro del club o indica un concepto, especifica la cantidad de dinero y pulsa el botón de añadir.</p> <p>El sistema realiza una comprobación de los datos y sobrescribe la instalación en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver movimientos bancarios
Introducción	El usuario puede ver los movimientos bancarios.
Entradas	Filtro de búsqueda.
Proceso	El usuario se dirige al apartado de administración de cuentas. En dicho apartado se listan todos los gastos o ingresos del club. También se le proporciona un sistema de filtros para seleccionar lo que quiere ver. Puede filtrar por temporada, por miembros y por tipo de movimiento. El sistema devuelve al usuario una lista con los movimientos
Salidas	Listado de movimientos bancarios.

2.3.2.7. Gestión de equipos

Nombre	Crear equipos
Introducción	El usuario puede crear equipos en el club.
Entradas	Datos del equipo.
Proceso	El usuario se dirige al apartado de crear equipos, rellena el nombre del equipo, selecciona la temporada y la categoría del equipo. También puede seleccionar un color identificativo del equipo. A continuación pulsa el botón de guardar y el sistema, realiza una comprobación de los datos y guarda el nuevo equipo en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar equipos
Introducción	El usuario puede editar equipos.
Entradas	Identificador del equipo.
Proceso	El usuario se dirige al apartado de administración de equipos y selecciona el equipo que desea editar. El usuario puede modificar el nombre, la categoría y la temporada del equipo Una vez realizadas las modificaciones el usuario pulsa el botón de guardar. El sistema realiza una comprobación de los datos y sobrescribe el equipo en base de datos. El sistema devuelve un mensaje indicando el resultado del proceso.
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Eliminar equipos
Introducción	El usuario puede eliminar equipos.
Entradas	Identificador del equipo.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de eliminar en el equipo deseado.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema comprueba si no tiene ningún partido asociado y si es afirmativo elimina el equipo de la base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver equipos
Introducción	El usuario puede ver los equipos del club.
Entradas	Identificador del equipo.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de ver en el equipo deseado. También podría ver los diferentes equipos del club desde la página de equipos.</p> <p>El sistema redirige al usuario a la página del equipo seleccionado.</p>
Salidas	Redirección al equipo seleccionado por el usuario.

Nombre	Añadir miembros
Introducción	El usuario puede añadir miembros a un equipo.
Entradas	Identificador del equipo y miembros.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de miembros en el equipo deseado. El sistema lo redirecciona a una página donde se muestran los miembros del club y un formulario para añadir más miembros.</p> <p>El usuario rellena el formulario y pulsa el botón añadir.</p> <p>El sistema realiza una comprobación de los datos y añade al nuevo miembro al equipo.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Eliminar miembros
Introducción	El usuario puede eliminar miembros de un equipo.
Entradas	Identificador del equipo.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de miembros en el equipo deseado. El sistema lo redirecciona a una página donde se muestran los miembros del club y un formulario para añadir más miembros. También se muestra un botón de eliminar sobre cada miembro del equipo.</p> <p>El usuario pulsa el botón eliminar sobre el miembro deseado.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema elimina al miembro del equipo y devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Añadir rivales
Introducción	El usuario puede añadir rivales a un equipo.
Entradas	Identificador del equipo y datos del rival.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de rivales en el equipo deseado. El sistema lo redirecciona a una página donde se muestran los rivales que ya tiene el equipo y un formulario para añadir más rivales.</p> <p>El usuario rellena el formulario y pulsa el botón añadir.</p> <p>El sistema realiza una comprobación de los datos y añade el nuevo rival al equipo.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Editar rivales
Introducción	El usuario puede editar rivales de un equipo.
Entradas	Identificador del equipo.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de rivales en el equipo deseado. El sistema lo redirecciona a una página donde se muestran los rivales del equipo y un formulario para añadir más rivales. También se muestra un botón de eliminar y otro de editar sobre cada rival del equipo.</p> <p>El usuario pulsa el botón editar sobre el rival deseado. Puede modificar el nombre, la descripción y el escudo del rival.</p> <p>El usuario pulsa el botón guardar.</p> <p>El sistema realiza una comprobación de los datos y si todo es correcto sobrescribe los datos del rival y devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Eliminar rivales
Introducción	El usuario puede eliminar rivales de un equipo.
Entradas	Identificador del equipo.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de rivales en el equipo deseado. El sistema lo redirecciona a una página donde se muestran los rivales del equipo y un formulario para añadir más rivales. También se muestra un botón de eliminar sobre cada rival del equipo.</p> <p>El usuario pulsa el botón eliminar sobre el rival deseado.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema elimina el rival del equipo y devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Añadir competiciones
Introducción	El usuario puede añadir competiciones en un equipo.
Entradas	Identificador del equipo y datos de la competición.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de competiciones en el equipo deseado. El sistema lo redirecciona a una página donde se muestran las diferentes competiciones en las que participa un equipo y un formulario para añadir más.</p> <p>El usuario rellena el formulario y pulsa el botón añadir.</p> <p>El sistema realiza una comprobación de los datos y añade la nueva competición al equipo.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar competiciones
Introducción	El usuario puede editar las competiciones en las que participa un equipo.
Entradas	Identificador del equipo e identificador de la competición.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de competiciones en el equipo deseado. El sistema lo redirecciona a una página donde se muestran las competiciones en las que participa el equipo y un formulario para añadir más competiciones. También se muestra un botón de eliminar y otro de editar sobre cada competición.</p> <p>El usuario pulsa el botón editar sobre la competición deseada. Puede modificar el nombre y la descripción.</p> <p>El usuario pulsa el botón guardar.</p> <p>El sistema realiza una comprobación de los datos y si todo es correcto sobrescribe los datos de la competición y devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Eliminar competiciones
Introducción	El usuario puede eliminar las competiciones de un equipo.
Entradas	Identificador del equipo.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de competiciones en el equipo deseado. El sistema lo redirecciona a una página donde se muestran las competiciones en las que participa el equipo y un formulario para añadir más. También se muestra un botón de eliminar y otro de editar sobre cada competición.</p> <p>El usuario pulsa el botón eliminar sobre la competición deseada.</p> <p>El sistema comprueba que esa competición no tenga ningún partido asociado y si es así, muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema elimina la competición del equipo y devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

2.3.2.8. Gestión de partidos

Nombre	Crear partidos
Introducción	El usuario puede crear los partidos de un equipo.
Entradas	Identificador del equipo y datos del partido.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón de añadir partido y rellena el formulario con nombre del rival, localización, hora del partido e indica si juegan como local o como visitante.</p> <p>A continuación pulsa el botón de guardar, el sistema realiza una comprobación de los datos y guarda el nuevo partido en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar partidos
Introducción	El usuario puede editar los partidos de un equipo.
Entradas	Identificador del equipo y datos del partido.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón de editar partido en el partido deseado. El usuario puede modificar el nombre del rival, localización, la hora del partido, la citación y si juega como local o visitante.</p> <p>A continuación pulsa el botón de guardar, el sistema realiza una comprobación de los datos y sobrescribe el partido en base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Eliminar partidos
Introducción	El usuario puede eliminar los partidos de un equipo.
Entradas	Identificador del equipo y del partido.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón de eliminar en el partido deseado.</p> <p>El sistema muestra un aviso donde se indica al usuario la acción que va a realizar. El usuario acepta el diálogo.</p> <p>El sistema comprueba si no tiene ningún dato asociado (convocatoria o estadísticas) y si es afirmativo elimina el partido de la base de datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Ver partidos
Introducción	El usuario puede ver los partidos de un equipo.
Entradas	Identificador del equipo y del partido.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón ver en el partido deseado. También podría ver los partidos desde el apartado de partidos.</p> <p>El sistema redirige al usuario a la página del partido, donde están todos los datos del partido (marcador, lista de convocados, estadísticas, localización, fecha y hora).</p>
Salidas	Redirección la página del partido.

Nombre	Añadir estadísticas
Introducción	El usuario puede añadir estadísticas en un partido.
Entradas	Identificador del partido y datos estadísticos.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón estadísticas en el partido deseado. El sistema redirige al usuario a una página que contiene a todos los jugadores del equipo y unos formularios donde incluir la estadística de cada jugador.</p> <p>Una vez los datos estadísticos han sido insertados el usuario pulsa el botón guardar.</p> <p>El sistema realiza una comprobación de los datos y guarda los datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar estadísticas
Introducción	El usuario puede editar las estadísticas de un partido.
Entradas	Identificador del partido y datos estadísticos.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón estadísticas en el partido deseado. El sistema redirige al usuario a una página que contiene a todos los jugadores del equipo y unos formularios rellenos con las estadísticas de cada jugador.</p> <p>El usuario modifica los datos de los formularios y pulsa el botón guardar.</p> <p>El sistema realiza una comprobación de los datos y sobrescribe los datos.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver estadísticas
Introducción	El usuario puede ver las estadísticas de un partido.
Entradas	Identificador del partido y datos estadísticos.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón ver en el partido deseado. También podría dirigirse directamente a la página del partido desde el apartado partidos.</p> <p>El sistema redirige al usuario a la página del partido, donde están todos los datos del partido (marcador, lista de convocados, estadísticas, localización, fecha y hora).</p> <p>El usuario selecciona la pestaña de estadísticas para ver las estadísticas del partido.</p>
Salidas	Redirección la página del partido y visualización de las estadísticas.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Nombre	Crear convocatoria
Introducción	El usuario puede crear la convocatoria en un partido.
Entradas	Identificador del partido y lista de convocados.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón convocatoria en el partido deseado.</p> <p>El sistema redirige al usuario a una página que contiene un listado de todos los jugadores del equipo, selecciona a los jugadores convocados y pulsa el botón de crear convocatoria.</p> <p>El sistema realiza una comprobación de los datos y crea la convocatoria.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Editar convocatoria
Introducción	El usuario puede editar la convocatoria en un partido.
Entradas	Identificador del partido y lista de convocados.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón convocatoria en el partido deseado. También podría editar la convocatoria desde la página del partido.</p> <p>El sistema redirige al usuario a una página que contiene un listado de todos los jugadores del equipo, modifica los jugadores seleccionados y pulsa el botón de modificar convocatoria.</p> <p>El sistema realiza una comprobación de los datos y modifica la convocatoria.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Ver convocatoria
Introducción	El usuario puede ver la convocatoria de un partido.
Entradas	Identificador del partido.
Proceso	<p>El usuario se dirige al apartado de administración de equipos y pulsa el botón de partidos en el equipo deseado.</p> <p>El sistema le muestra una lista con todos los partidos del equipo.</p> <p>El usuario pulsa el botón ver en el partido deseado. También podría dirigirse directamente a la página del partido desde el apartado partidos.</p> <p>El sistema redirige al usuario a la página del partido, donde están todos los datos del partido (marcador, lista de convocados, estadísticas, localización, fecha y hora).</p> <p>El usuario selecciona la pestaña de convocatoria para ver la convocatoria del partido.</p>
Salidas	Redirección la página del partido y visualización de la convocatoria.

Nombre	Editar perfil
Introducción	El usuario puede editar su propio perfil.
Entradas	Identificador de usuario.
Proceso	<p>El usuario se dirige al apartado de configuración</p> <p>El sistema le mostrará un formulario con todos sus datos.</p> <p>El usuario modifica los datos que desee y pulsa el botón guardar. También puede modificar su foto de perfil</p> <p>El sistema realiza una comprobación de los datos y sobrescribe los nuevos datos del perfil.</p> <p>El sistema devuelve un mensaje indicando el resultado del proceso.</p>
Salidas	Mensaje satisfactorio o mensaje de error en caso de fallo.

Nombre	Iniciar sesión
Introducción	El usuario puede iniciar sesión.
Entradas	Usuario y contraseña.
Proceso	<p>El usuario se dirige al apartado de iniciar sesión, escribe su usuario y contraseña y pulsa el botón de iniciar sesión.</p> <p>El sistema comprueba los datos y si coinciden con un usuario del sistema inicia la sesión del usuario.</p>
Salidas	Usuario con la sesión iniciada.

Nombre	Crear club
Introducción	El usuario puede crear un nuevo club.
Entradas	Datos del club y del usuario administrador.
Proceso	<p>El usuario se dirige al apartado de crear club.</p> <p>EL sistema le muestra un formulario para rellenar los datos del club. El usuario rellena los datos y pulsa siguiente. El sistema muestra otro formulario, esta vez para añadir los datos del usuario administrador. El usuario los rellena y pulsa el botón de crear.</p> <p>El sistema comprueba todos los datos y crea el nuevo club.</p>
Salidas	Creación de un nuevo club.

Nombre	Registro
Introducción	El usuario puede registrarse como miembro de un club.
Entradas	Identificador del club y datos del usuario.
Proceso	<p>El usuario se dirige a la página de registro del club donde desee registrarse.</p> <p>Rellena los datos del registro y pulsa el botón de registrar.</p> <p>El sistema realiza una comprobación de los datos y registra al usuario en el club.</p> <p>Aunque el usuario esté registrado, este no tendrá acceso al club hasta que un usuario administrador lo active.</p>
Salidas	Usuario registrado en el club.

2.3.3. Requisitos de rendimiento

Inicialmente utilizaremos un Servidor en la nube con un procesador Intel Xeon E312xx, contará con 4 GB de memoria RAM y 20 GB de disco duro local. Además se ha contratado un disco duro externo al servidor para guardar todos los archivos pesados (imágenes, videos o archivos PDF)

Inicialmente esta configuración es suficiente para mantener un buen rendimiento de la aplicación con el volumen de usuarios inicial.

Aunque inicialmente sea suficiente, la aplicación debe ser escalable para poder hacer frente a posibles subidas en el número de clientes y conexiones simultáneas.

Al ser escalable se puede aumentar el número de servidores, crear copias redundantes de la base de datos o ampliar el espacio en disco. De esta forma será posible mantener un buen rendimiento.

2.3.4. Restricciones del diseño

Para realizar el desarrollo de la aplicación se han seguido los estándares establecidos por W3C en los cuales se ofrecen una serie de recomendaciones de HTML, CSS y Javascript. De esta manera y tal como se ha comentado anteriormente, se asegura que los navegadores

traducen los HTML, CSS y Javascript con el objetivo de mostrar la web tal y como ha sido diseñada.

En caso de haber errores o si no se hubieran seguido las recomendaciones de W3C, cada navegador realizaría una interpretación diferente de la web desfavoreciendo la experiencia final del usuario.

2.3.5. Atributos del sistema software

Seguridad

La seguridad de la aplicación es un aspecto fundamental, ya que contiene información privada de clubs que debe mantenerse oculta a usuarios ajenos al club. Por este motivo, la primera medida de seguridad es la autenticación del usuario, el usuario necesitará introducir un nombre de usuario y una contraseña. Una vez comprobado que los datos de inicio de sesión son los correctos, el usuario ya tendrá acceso a las zonas privadas del club.

Por otro lado, el club puede decidir qué usuarios tienen permisos para acceder a según qué recursos; un buen ejemplo es que un usuario con rol de jugador, no puede tener acceso a crear una convocatoria de partido. El sistema tiene que ser capaz de identificar qué usuarios tienen acceso a los diferentes recursos del sistema. Por este motivo, antes de dar de alta al usuario, o una vez ya dado de alta, es importante asignar qué roles va a tener dicho miembro.

Fiabilidad

Para garantizar la fiabilidad de la aplicación se garantiza una arquitectura de servidores redundante, para que en caso de fallar un servidor por algún tipo de problema se pueda seguir utilizando la aplicación en el otro servidor.

Además, se tendrá que crear una tarea CRON, para realizar copias de seguridad del servidor, tanto de la BBDD como de las imágenes y archivos de los usuarios. Dichas copias de seguridad se llevarán a cabo en periodos de tiempo con poca afluencia de tráfico para perjudicar al mínimo el rendimiento del sistema.

Disponibilidad

Una característica fundamental de las aplicaciones web es que siempre están accesibles a todos los usuarios. Por este motivo es importante tener una buena configuración de los servidores para que se produzcan el menor número de errores en el sistema. De esta forma poder ofrecer a los usuarios una alta disponibilidad. También se ha de tener en cuenta el porcentaje de disponibilidad que nos ofrece nuestro proveedor.

Mantenibilidad

Se usará un framework como Codeigniter no solo es por el uso de las librerías y facilidades que incorpora, sino por la utilización del patrón Modelo - Vista - Controlador. Este patrón ayuda a mantener el código ordenado y facilita su reutilización. Todo esto unido a una buena escritura del código hace que la aplicación web sea perfectamente mantenible por ambos desarrolladores o en un futuro por nuevos desarrolladores que se incorporen al proyecto.



Portabilidad

La aplicación debe ser portable, para poder ser cambiada a cualquier otro servidor, ya sea por querer contratar un servidor más potente o por querer ampliar la redundancia. Dichos servidores deben tener las tecnologías usadas por la aplicación.

Eficiencia

También es importante la eficiencia de la aplicación. En este caso la aplicación debe ser lo más eficiente posible, ya que se trata de una aplicación que puede ampliar el número de usuarios y si no se realiza una implementación correcta pueden surgir problemas de rendimiento.

También la eficiencia repercute en los tiempos de respuesta del servidor y hace que el usuario final tenga una mejor experiencia de uso. Además la velocidad de la aplicación también repercute en el SEO de la misma.

3. Análisis

3.1. Introducción

En este apartado se realiza un análisis de la aplicación a desarrollar, describiendo la estructura y la funcionalidad de la aplicación a base de diagramas UML^[14].

Dicho lenguaje es utilizado para especificar y describir métodos o procesos del sistema. En este análisis se han utilizado diagramas de flujo, de casos de uso y de secuencia para definir las diferentes interacciones.

3.2. Diagramas de casos de uso

Los diagramas de casos de uso se utilizan para describir de forma sencilla y esquemática cada uno de los casos de uso descritos en el apartado de requisitos funcionales [2.3.2.](#)

Éstos son diagramas de comportamiento y son utilizados para describir paso a paso las diferentes acciones para realizar un proceso concreto. De esta forma el diagrama expresa las interacciones que deben realizar los actores de la aplicación para obtener una respuesta u operación por parte del sistema.

3.2.1. Actores de la aplicación

En la figura 1, se muestran los diferentes actores de la aplicación. Cada actor hace referencia a los distintos tipos de usuario que pueden hacer uso de la aplicación. En este caso, se dispone de siete actores diferentes, cinco de ellos son especializaciones de un actor más general, usuario registrado. El otro actor sería el usuario no registrado.

Como se especifica en el apartado de características de usuarios [2.2.3;](#) un usuario puede disponer de más de un rol a la vez.

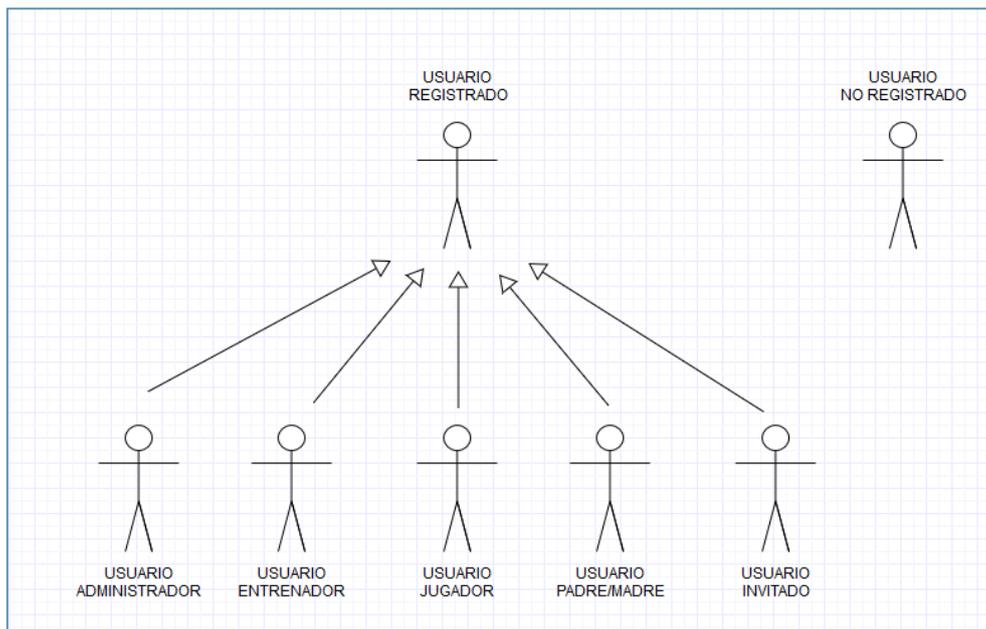


Figura 1. Actores de la aplicación

3.2.2. Administrador

La función del actor “administrador”, se divide en pequeños grupos que hacen referencia al mismo apartado de la aplicación facilitando de este modo la comprensión de los casos de uso.

3.2.2.1. Gestión de noticias

Como se observa en la figura 2, el usuario administrador puede gestionar todas las noticias del club, esto incluye crearlas, editarlas y eliminarlas. También, como cualquier otro usuario del club podrá visualizar las noticias.

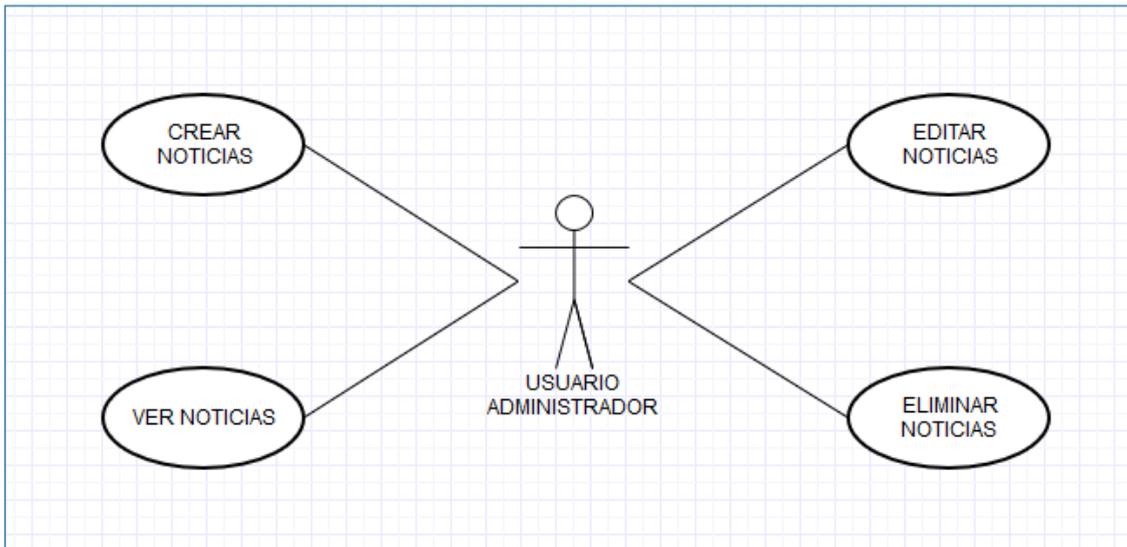


Figura 2. Gestión de noticias del usuario administrador.

3.2.2.2. Gestión de eventos

El usuario administrador puede gestionar los eventos del club; la creación, edición y/o eliminación de éstos. También puede ver los eventos y participar en ellos siempre que el evento le permita a este tipo de usuarios participar, tal y como se muestra en la figura 3.

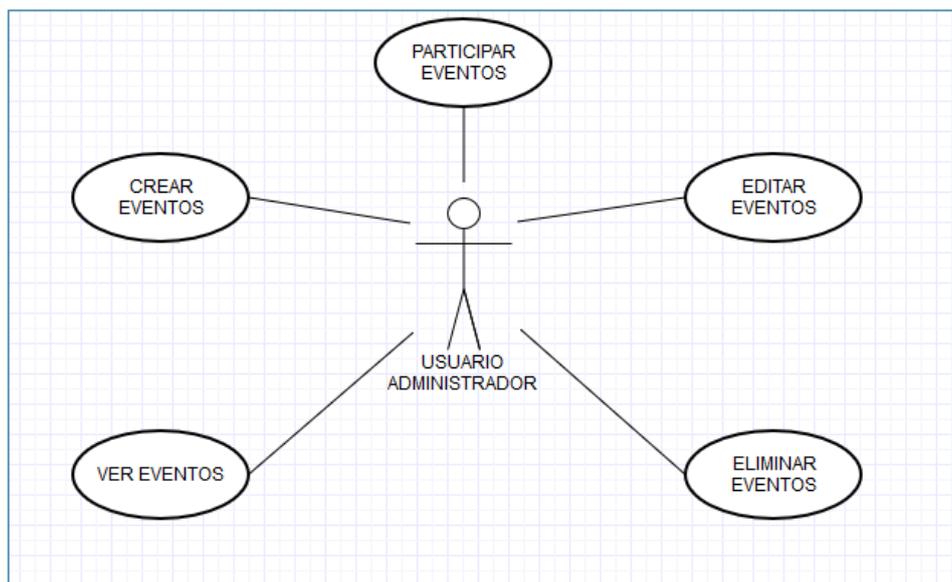


Figura 3. Gestión de eventos del usuario administrador.

3.2.2.3. Gestión de usuarios

Como usuario administrador, tiene los permisos necesarios para poder crear, editar y eliminar usuarios del club. También puede activarlos y desactivarlos, para permitir o no el inicio de sesión a cada usuario. Otra acción que puede realizar el usuario administrador es la de añadir o quitar roles a los usuarios. Por último, también puede editar su propio perfil de usuario. Ver detalle en la figura 4.

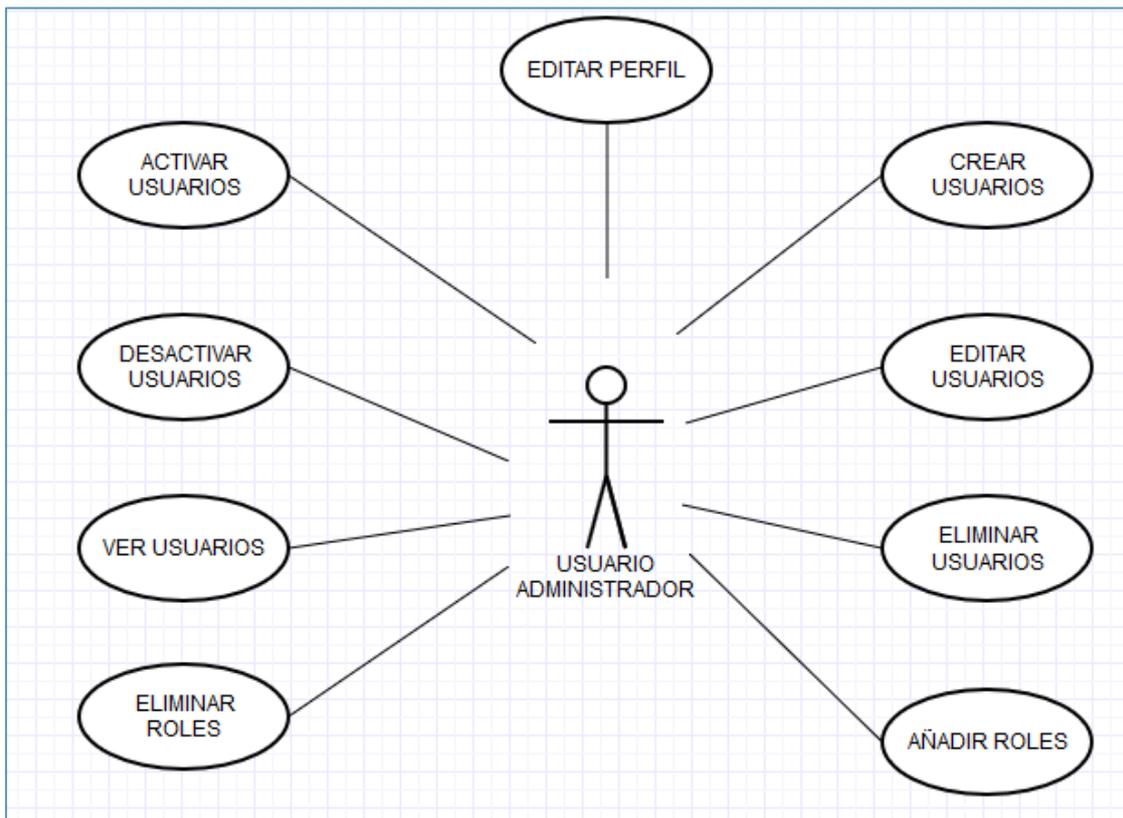


Figura 4. Gestión de usuarios del usuario administrador.

3.2.2.4. Editar club

El usuario administrador puede cambiar tanto el escudo del club como la imagen de cabecera, también tiene acceso a la página de preferencias del club que le permite añadir o quitar enlaces de redes sociales y editar la información del club. Además, puede crear, editar, eliminar o ver páginas del club, que son páginas con información pública. Ver detalle en la figura 5

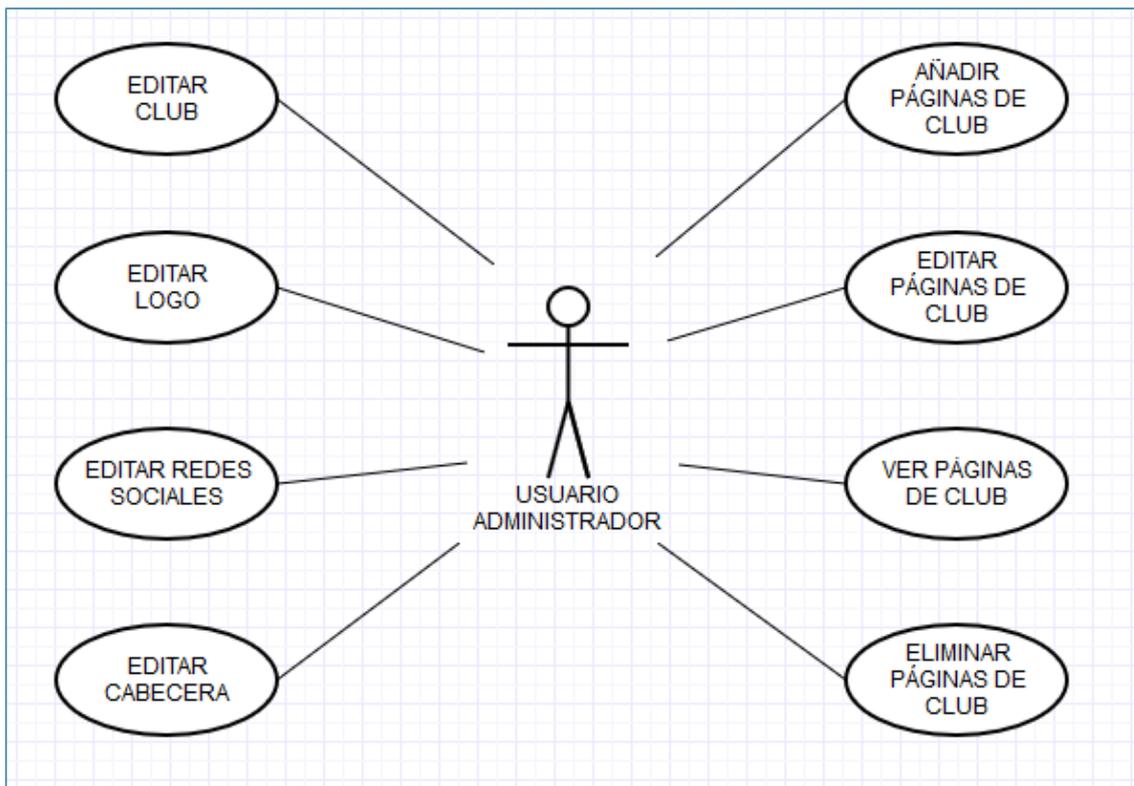


Figura 5. Editar club.

3.2.2.5. Gestión de instalaciones

Como se observa en la figura 6 , el usuario administrador puede visualizar, crear, editar y eliminar las instalaciones. Dichas instalaciones pueden ser propias del club, o de los rivales.

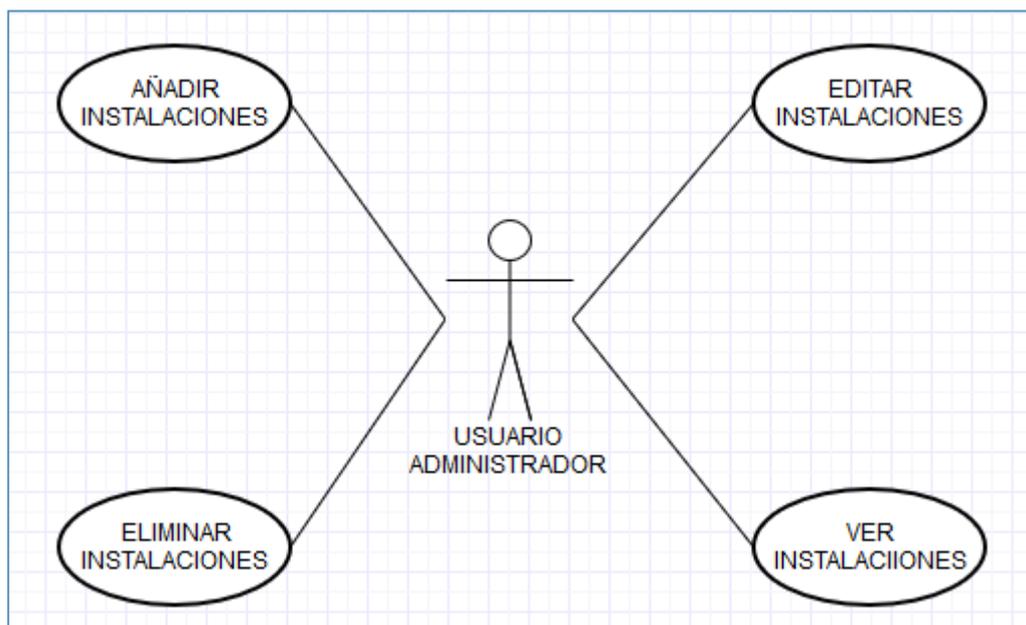


Figura 6. Gestión de instalaciones del usuario administrador.

3.2.2.6. Gestión de equipos

El usuario administrador puede crear, editar y eliminar los equipos de un club. Dentro de las opciones de los equipos, también es capaz de añadir o eliminar miembros, rivales o competiciones al equipo. Los rivales y las competiciones pueden ser modificados por el usuario administrador. Ver detalle en la figura 7.

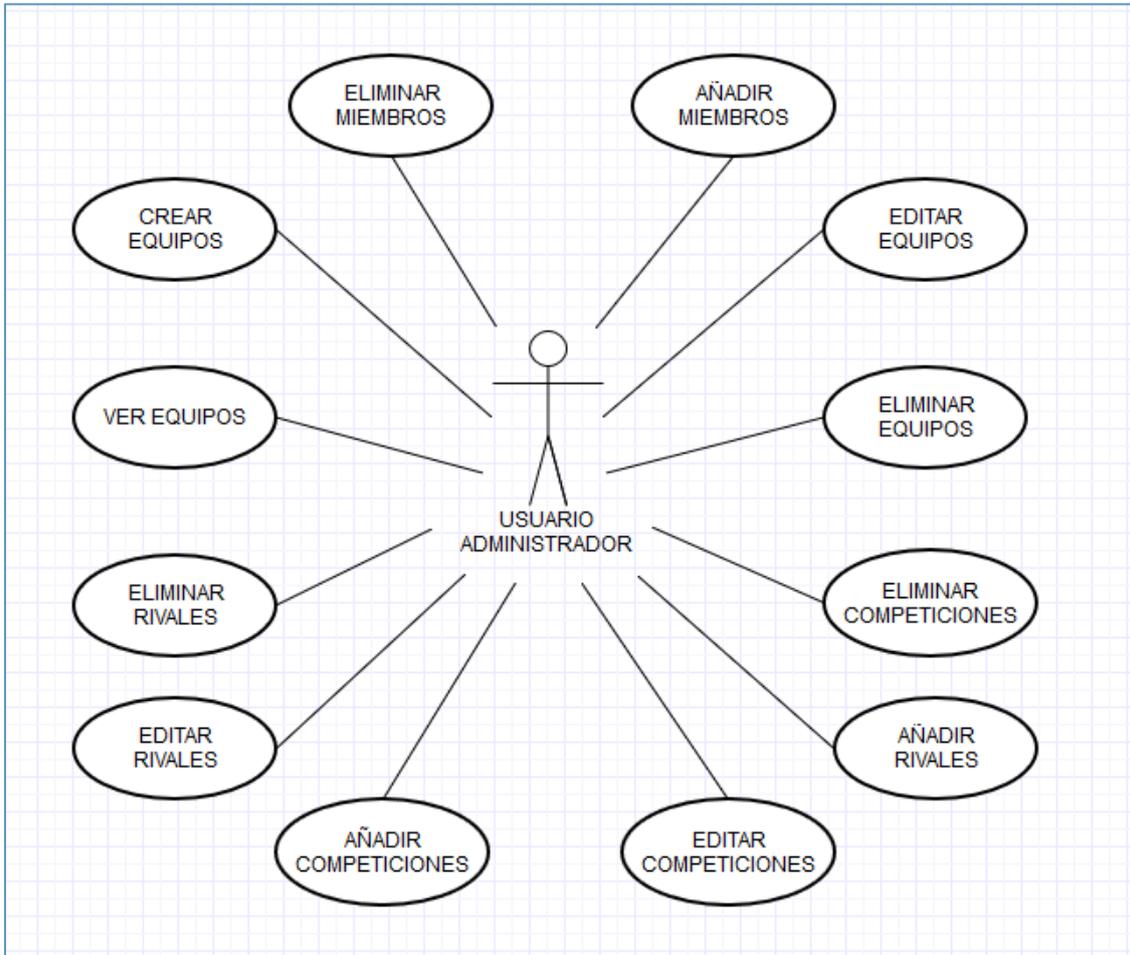


Figura 7. Gestión de equipos del usuario administrador.

3.2.2.7. Gestión bancaria

El usuario administrador tiene acceso al apartado bancario donde puede añadir movimientos, tanto gastos como ingresos, asimismo puede visualizar un listado de todos los movimientos que se han producido. Ver detalle en la figura 8.

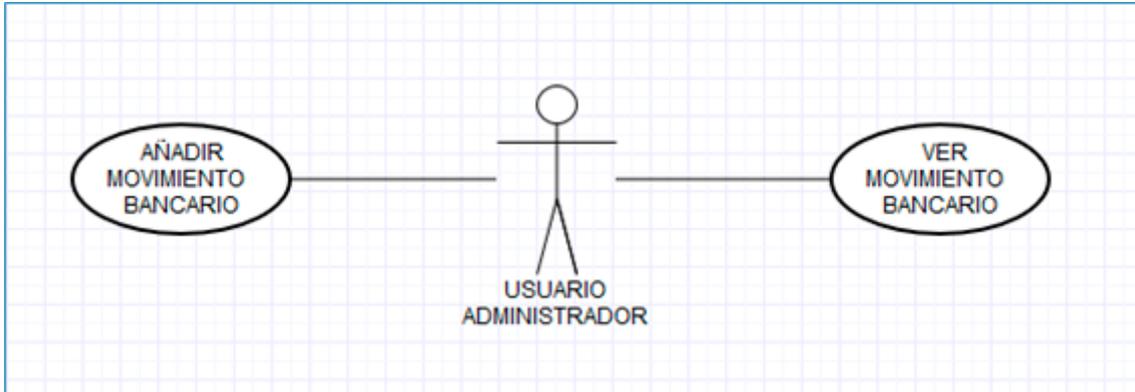


Figura 8. Gestión bancaria del usuario administrador.

3.2.2.8. Gestión de partidos

El usuario administrador puede gestionar los partidos de los equipos, por tanto, puede crear, editar y eliminar partidos. Además puede realizar acciones dentro de dichos partidos, ya sea crear o editar la convocatoria del partido o incluso añadir o editar las estadísticas. Ver detalle en la figura 9.

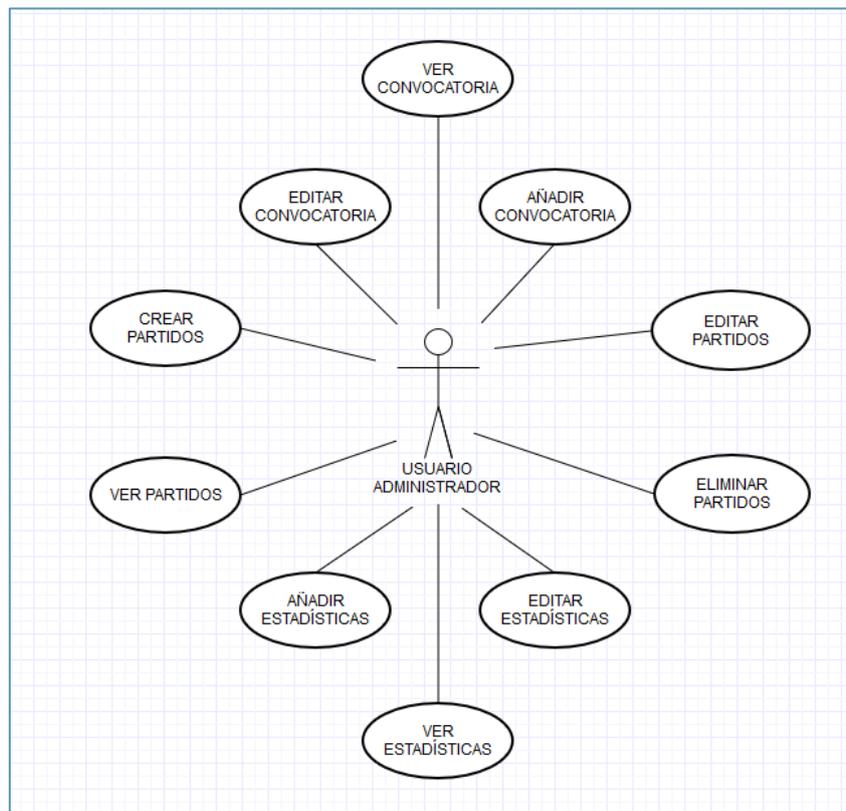


Figura 9. Gestión de partidos del usuario administrador.

3.2.3. Usuario entrenador

La función del actor “entrenador”, se divide en pequeños grupos que hacen referencia al mismo apartado de la aplicación facilitando de este modo la comprensión de los casos de uso.

3.2.3.1. Visualizaciones

El usuario entrenador puede visualizar las noticias, las instalaciones que son propias del club, sus movimientos bancarios y los eventos. En caso de que el evento permita la participación de los entrenadores, también podrá participar en los eventos. Ver detalle en la figura 10.

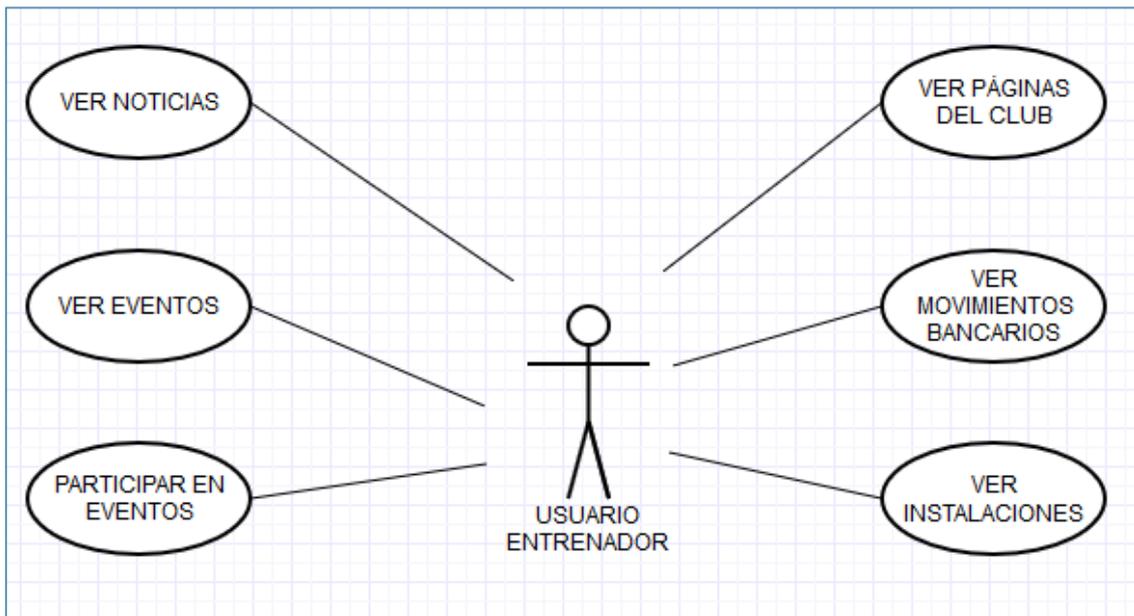


Figura 10. Visualizaciones disponibles para el usuario entrenador.

3.2.3.2. Gestión de perfil

El usuario entrenador puede visualizar los usuarios que tiene el club y editar su propio perfil de usuario. Ver detalle en la Figura 11.

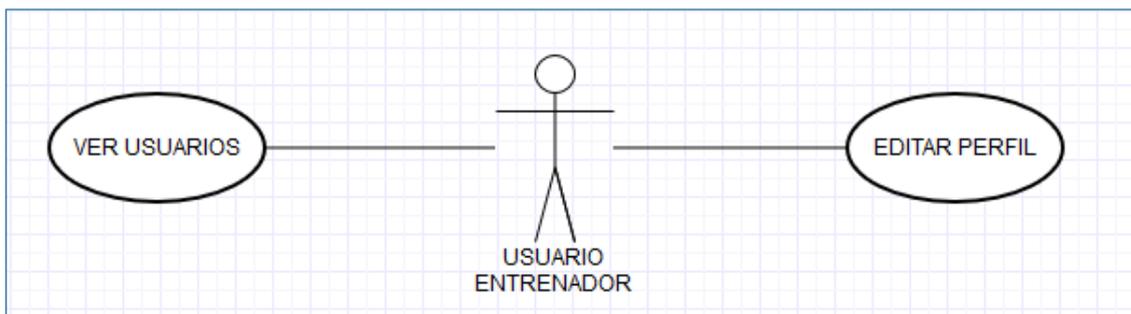


Figura 11. Gestión de perfil para el usuario entrenador.

3.2.3.3. Gestión de partido

El usuario entrenador puede visualizar todos los partidos del club. En aquellos en los que actúa como entrenador, también puede crear o editar un partido, añadir o editar una convocatoria o incluso añadir o editar estadísticas a los jugadores del equipo. Ver detalle en la figura 12.

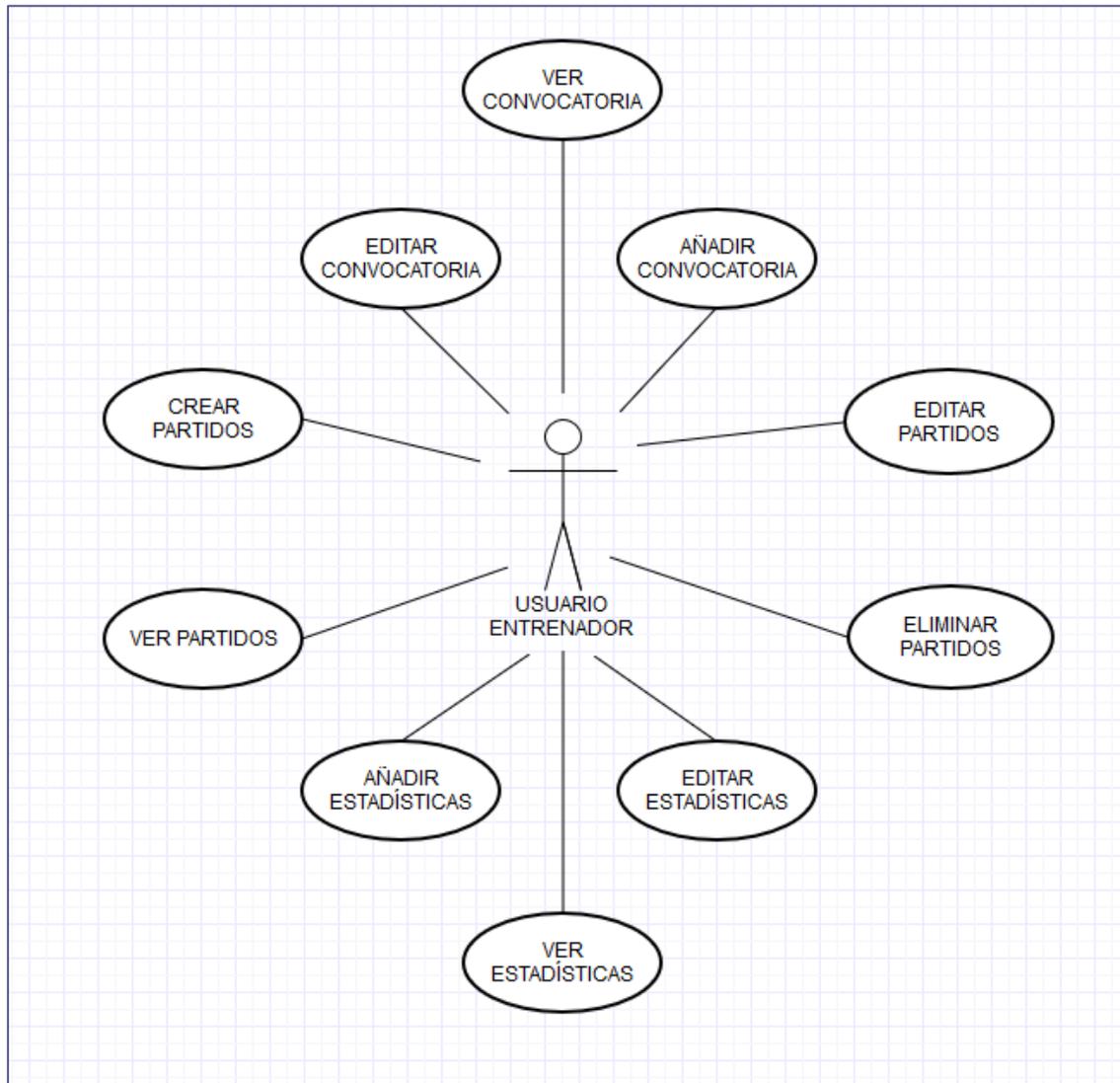


Figura 12. Gestión de partidos para el usuario entrenador.

3.2.3.4. Gestión de equipo

El usuario entrenador puede visualizar todos los equipos del club, En aquellos en los que actúa como entrenador, puede crear o editar una competición, añadir o eliminar miembros o editar el equipo. Ver detalle en la figura 13.

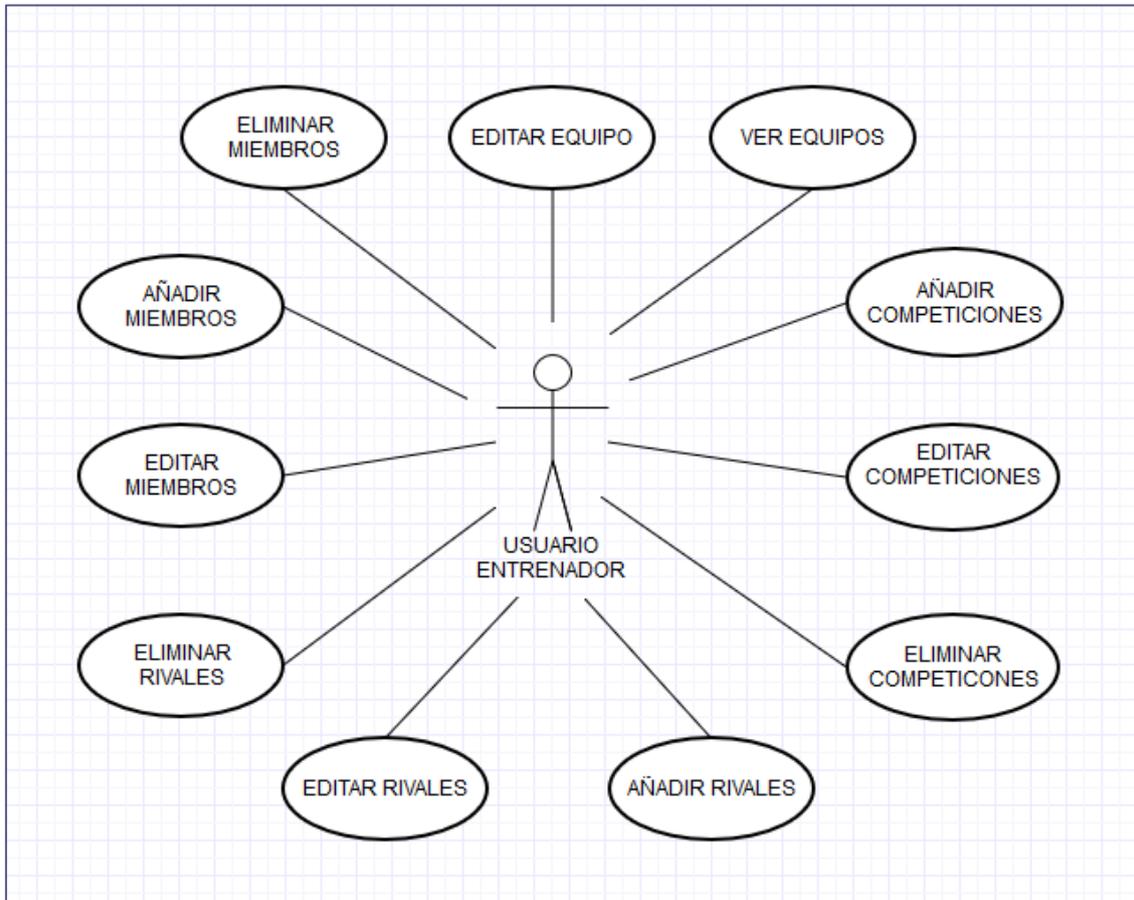


Figura 13. Gestión de equipos para el usuario entrenador.

3.2.3. Usuario jugador

La función del actor “jugador”, se divide en pequeños grupos que hacen referencia al mismo apartado de la aplicación facilitando de este modo la comprensión de los casos de uso.

3.2.3.1. Visualizaciones

El usuario jugador puede visualizar las noticias, las instalaciones que son propias del club, sus movimientos bancarios, los equipos y los eventos. En caso de que el evento permita la participación de los jugadores, también podrá participar en los eventos. Ver detalle en la figura 14.

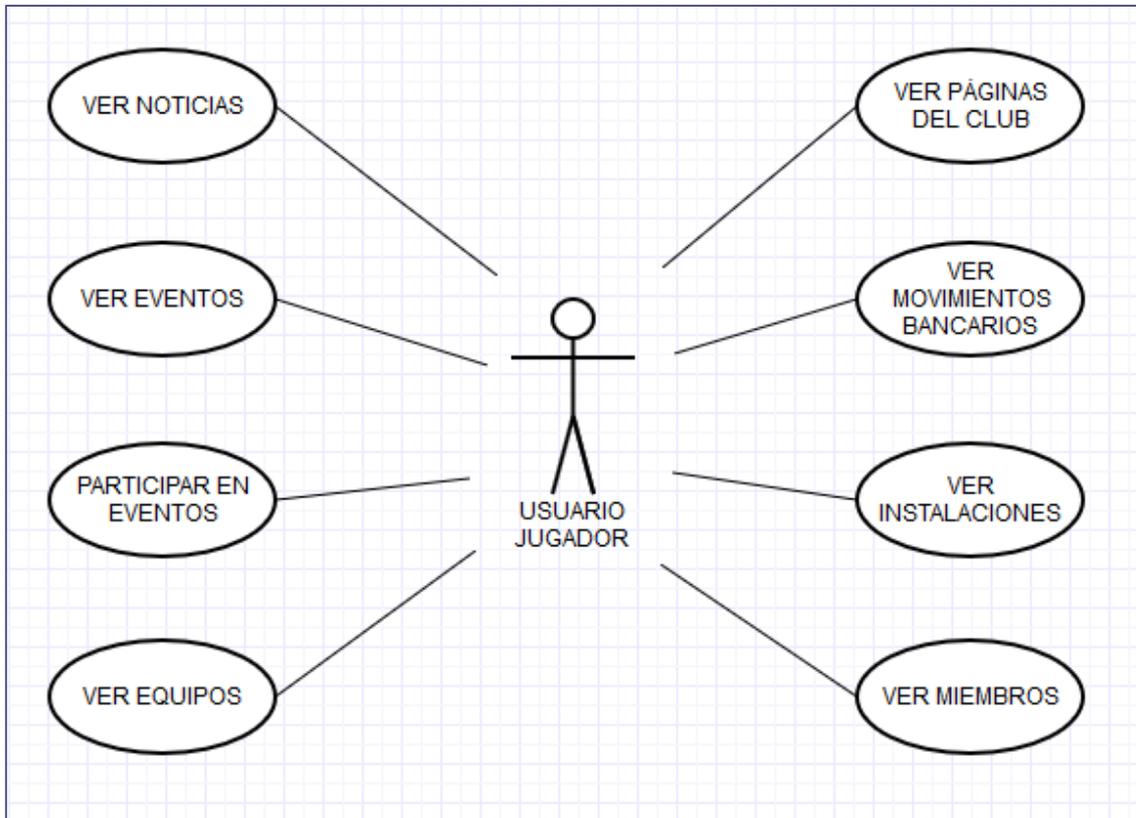


Figura 14. Visualizaciones del usuario jugador.

3.2.3.2. Gestión de perfil

El usuario jugador puede visualizar los usuarios que tiene el club y editar su propio perfil de usuario. Ver detalle en la figura 15.

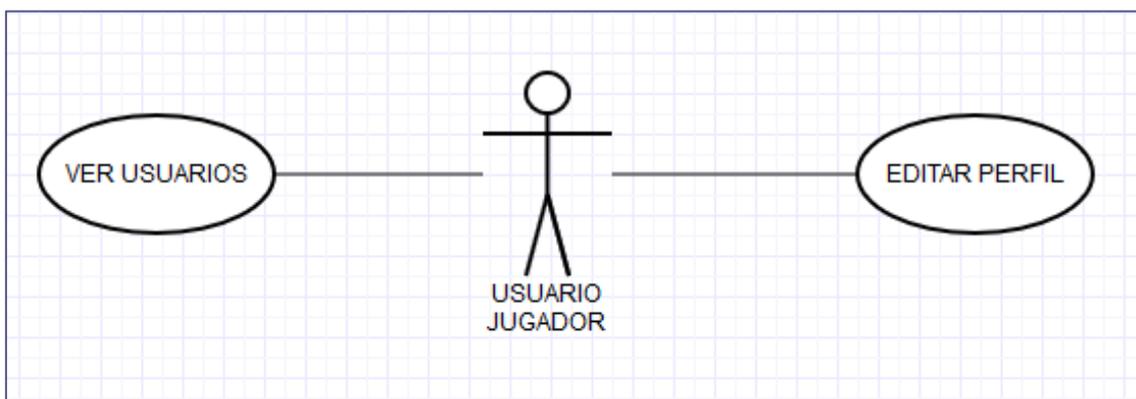


Figura 15. Gestión de perfil del usuario jugador.

3.2.3.3. Gestión de partido

El usuario jugador puede visualizar todos los partidos del club. El privilegio de ser jugador es que el entrenador o administrador puede añadirlo a una convocatoria o también puede tener estadísticas del partido. Ver detalle en la figura 16.

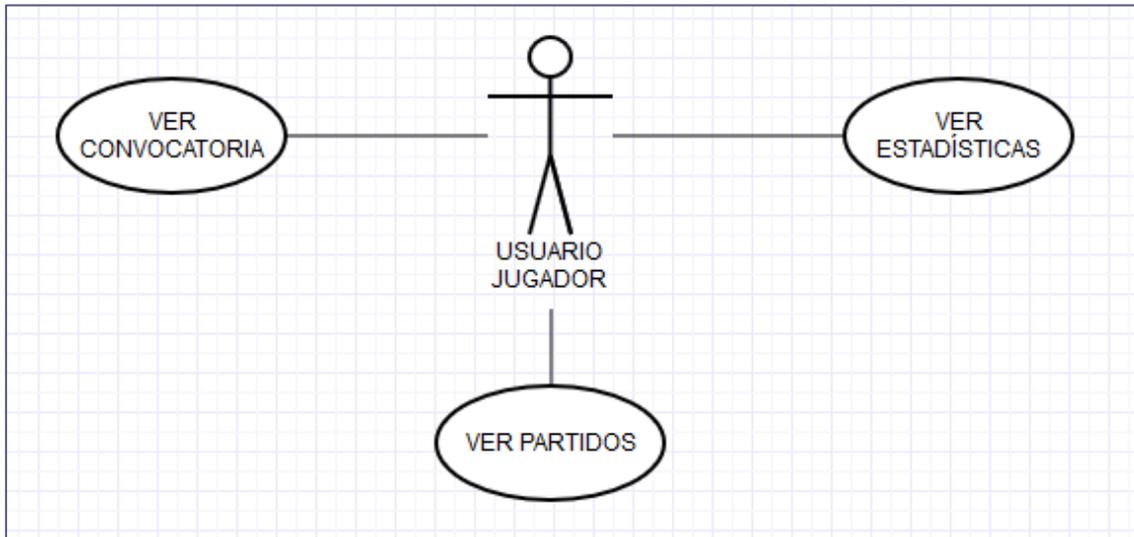


Figura 16. Gestión de partidos para el usuario jugador.

3.2.4. Usuario padre/madre

La función del actor “padre/madre”, se divide en pequeños grupos que hacen referencia al mismo apartado de la aplicación facilitando de este modo la comprensión de los casos de uso.

3.2.4.1. Visualizaciones

El usuario padre/madre puede visualizar las noticias, las instalaciones que son propias del club, sus movimientos bancarios, los equipos y los eventos. En caso de que el evento permita la participación de los padres/madres, también podrá participar en los eventos. Ver detalle en la figura 17.

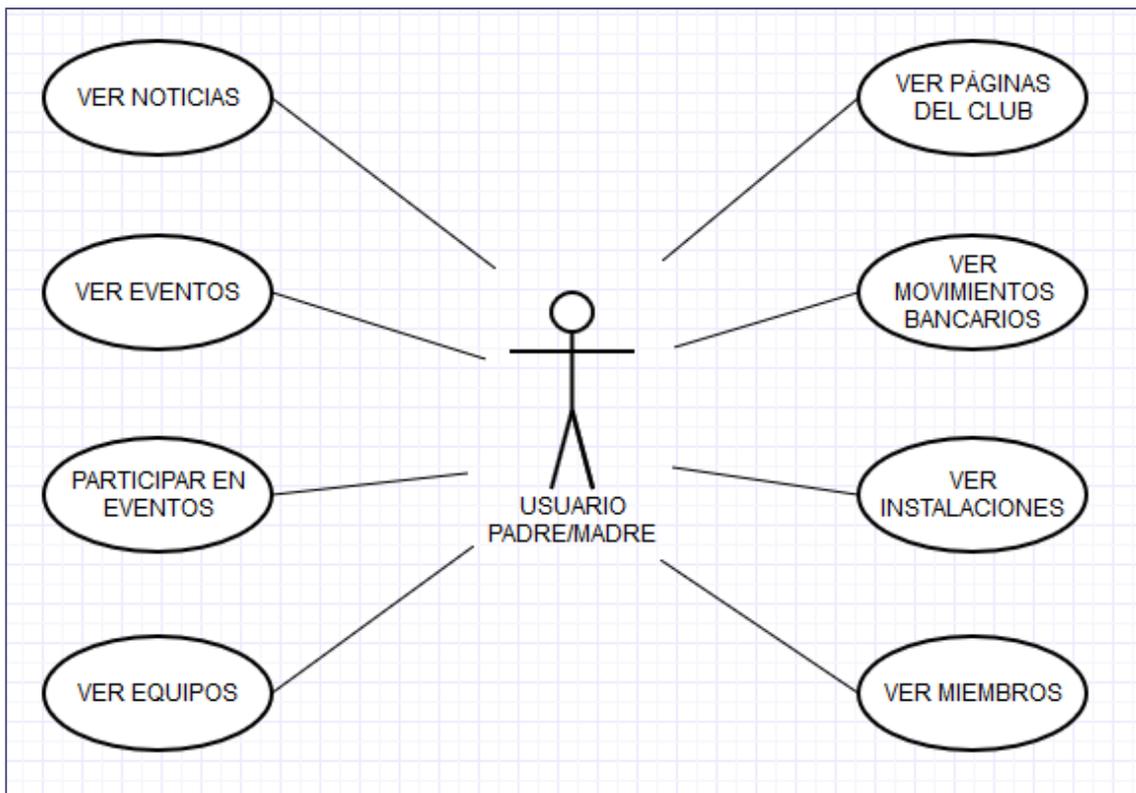


Figura 17. Visualizaciones del usuario padre/madre.

3.2.4.2. Gestión de perfil

El usuario padre/madre puede visualizar los usuarios que tiene el club y editar su propio perfil de usuario. Ver detalle en la figura 18.

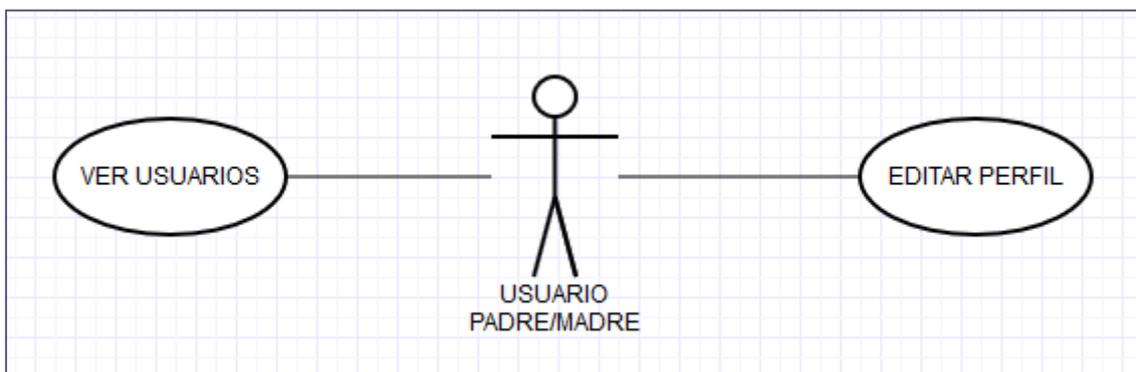


Figura 18. Gestión de perfil del usuario padre/madre.

3.2.4.3. Gestión de partido

El usuario padre/madre puede visualizar todos los partidos del club. Ver detalle en la figura 19.

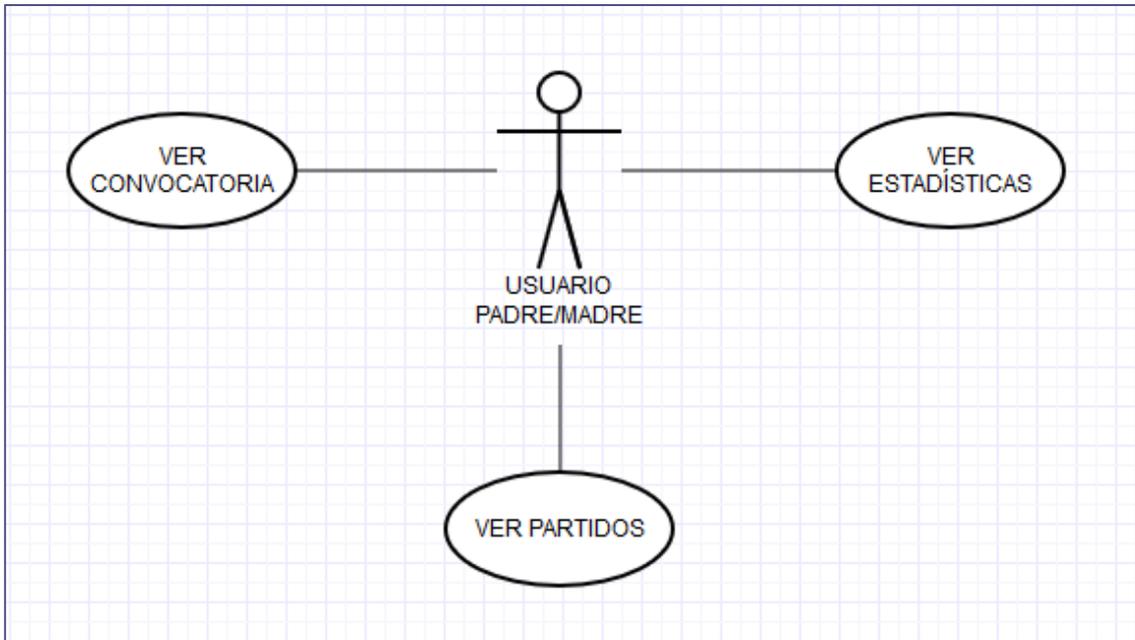


Figura 19. Gestión de partidos del usuario padre/madre.

3.2.5. Usuario invitado

La función del actor “invitado”, se divide en pequeños grupos que hacen referencia al mismo apartado de la aplicación facilitando de este modo la comprensión de los casos de uso.

3.2.5.1. Visualizaciones

El usuario invitado puede visualizar las noticias, las instalaciones que son propias del club, sus movimientos bancarios, los equipos y los eventos. En el caso del usuario invitado, este no podrá participar en ningún evento. Ver detalle en la figura 20.

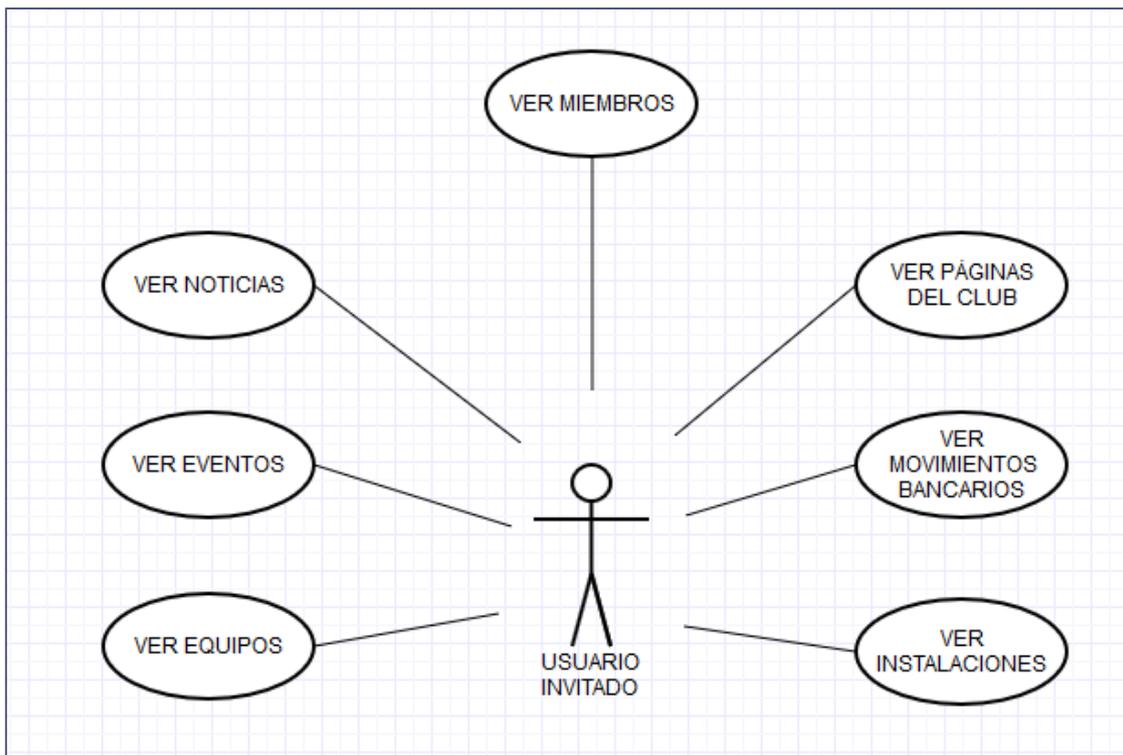


Figura 20. Visualizaciones para el usuario invitado.

3.2.5.2. Gestión de perfil

El usuario invitado puede visualizar los usuarios que tiene el club y editar su propio perfil de usuario. Ver detalle en la figura 21

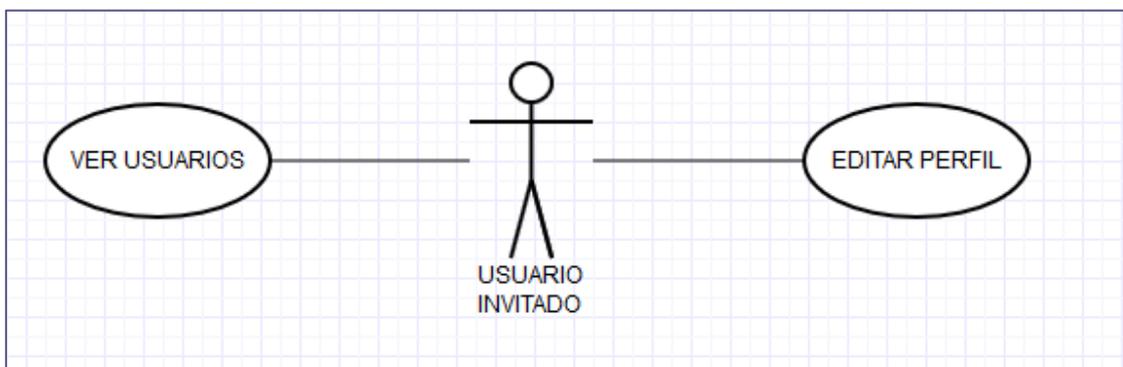


Figura 21. Gestión de perfil del usuario invitado.

3.2.5.3. Gestión de partido

El usuario invitado puede visualizar todos los partidos del club. Véase en la figura 22.

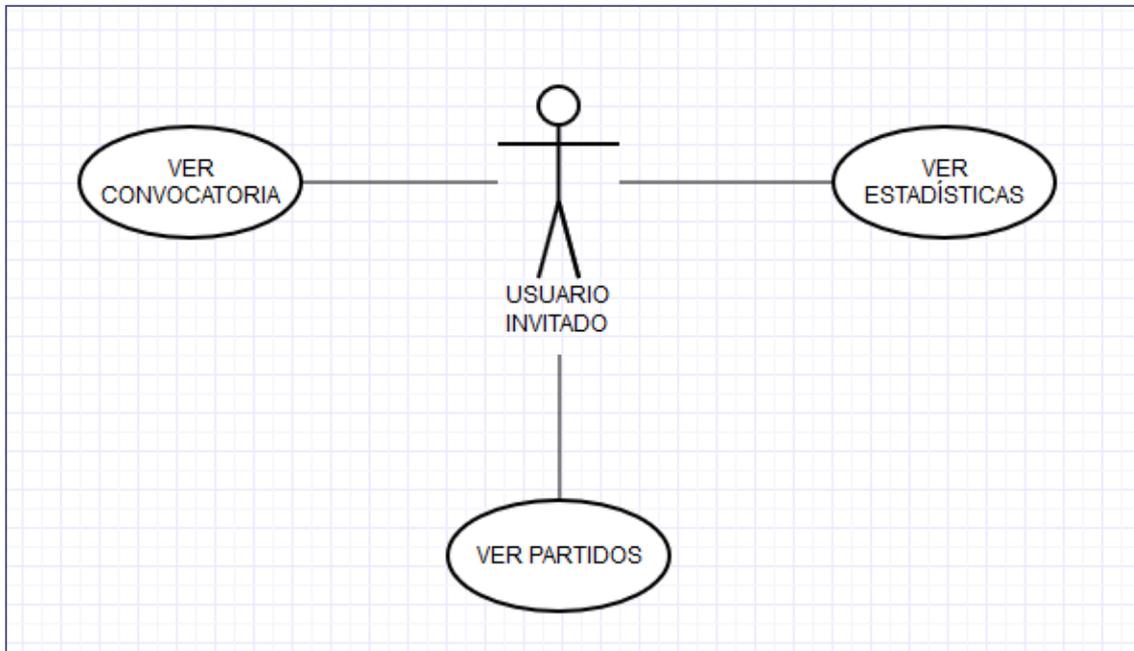


Figura 22. Gestión de partidos del usuario invitado.

3.2.6. Usuario no registrado

La función del actor “no registrado”, se divide en pequeños grupos que hacen referencia al mismo apartado de la aplicación facilitando de este modo la comprensión de los casos de uso.

3.2.6.1. Visualizaciones

El usuario no registrado puede visualizar las noticias que son públicas, las instalaciones que son propias del club, los equipos y las páginas del club. Ver detalle en la figura 23.

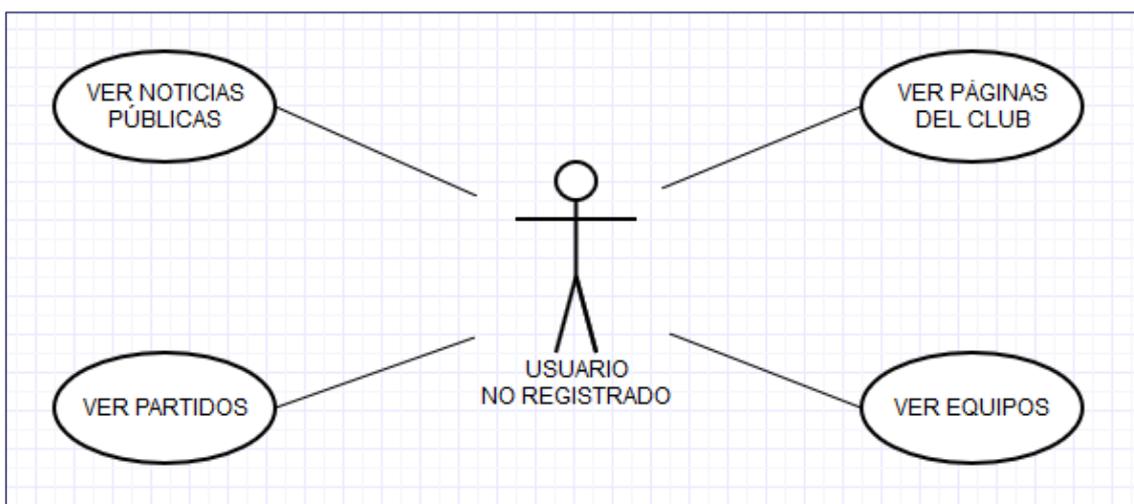


Figura 23. Visualizaciones del usuario no registrado.

3.2.6.2. Acciones

El usuario no registrado puede iniciar sesión, crear un club o registrarse en un club ya creado. Ver detalle en la figura 24.

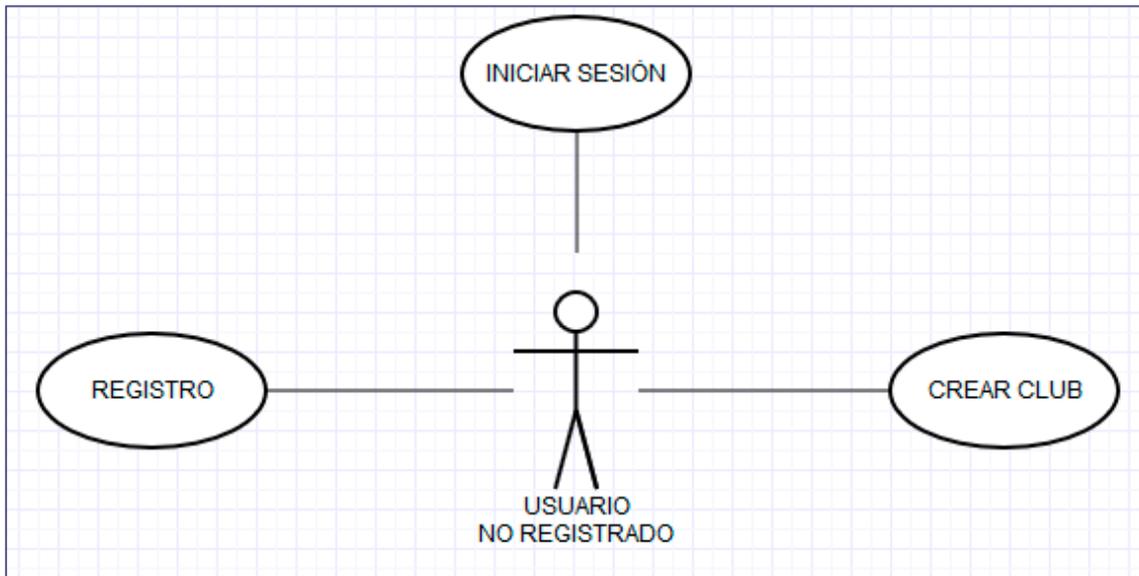


Figura 24. Acciones que puede realizar el usuario no registrado.

3.4. Diagramas de flujo para la gestión de equipos

El siguiente diagrama muestra el flujo de acciones que hay que realizar para gestionar un equipo, desde la creación del mismo hasta las acciones que se realizan en un partido del equipo. Ver detalle en la figura 26.

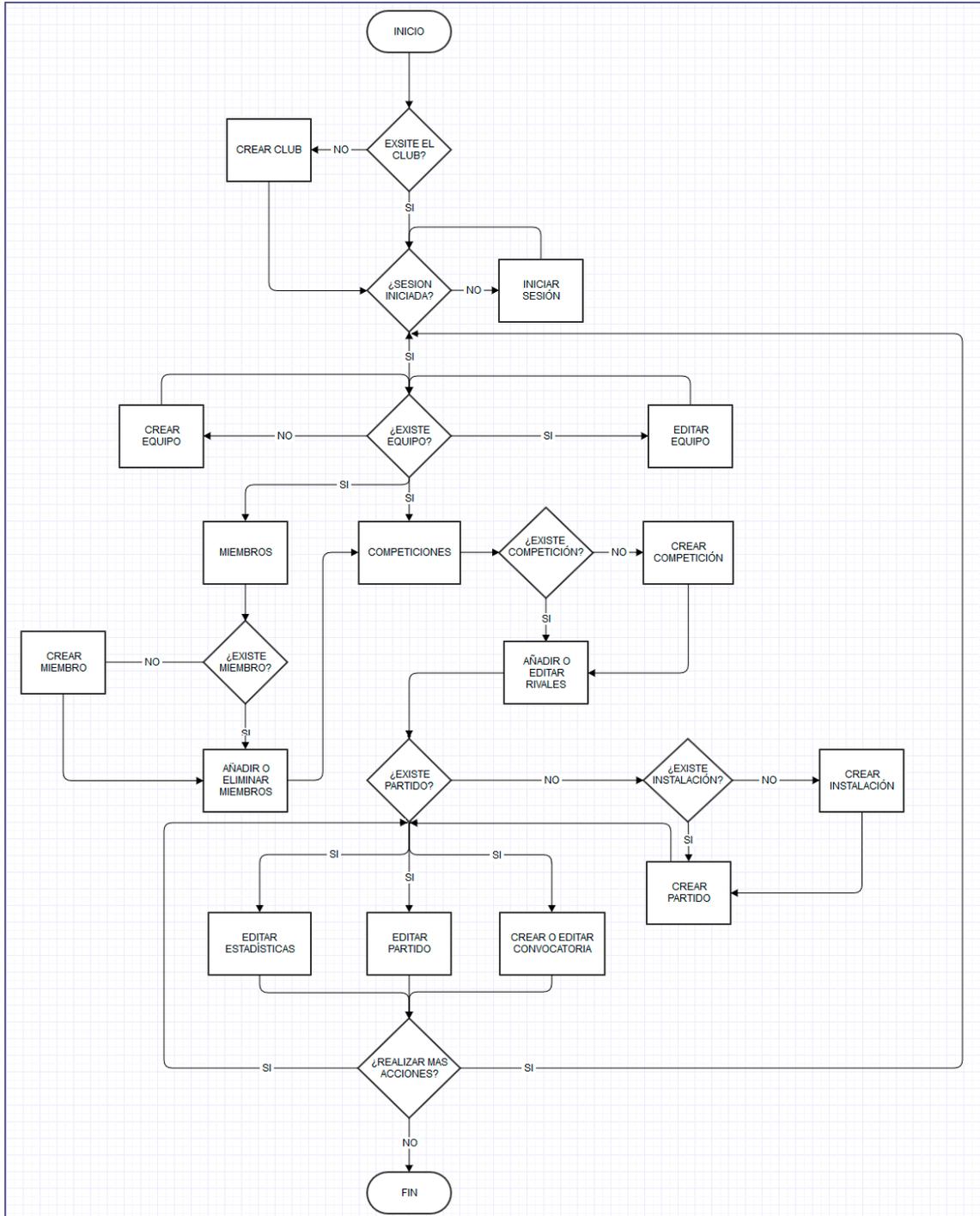


Figura 26. Diagrama de flujo para la gestión de equipos.

3.5. Diagramas de flujo para la gestión de la web

En este diagrama se describe el flujo de la aplicación para realizar la gestión del club. Esto incluye la gestión de noticias, eventos, páginas del club y movimientos bancarios. Ver detalle en la figura 27.

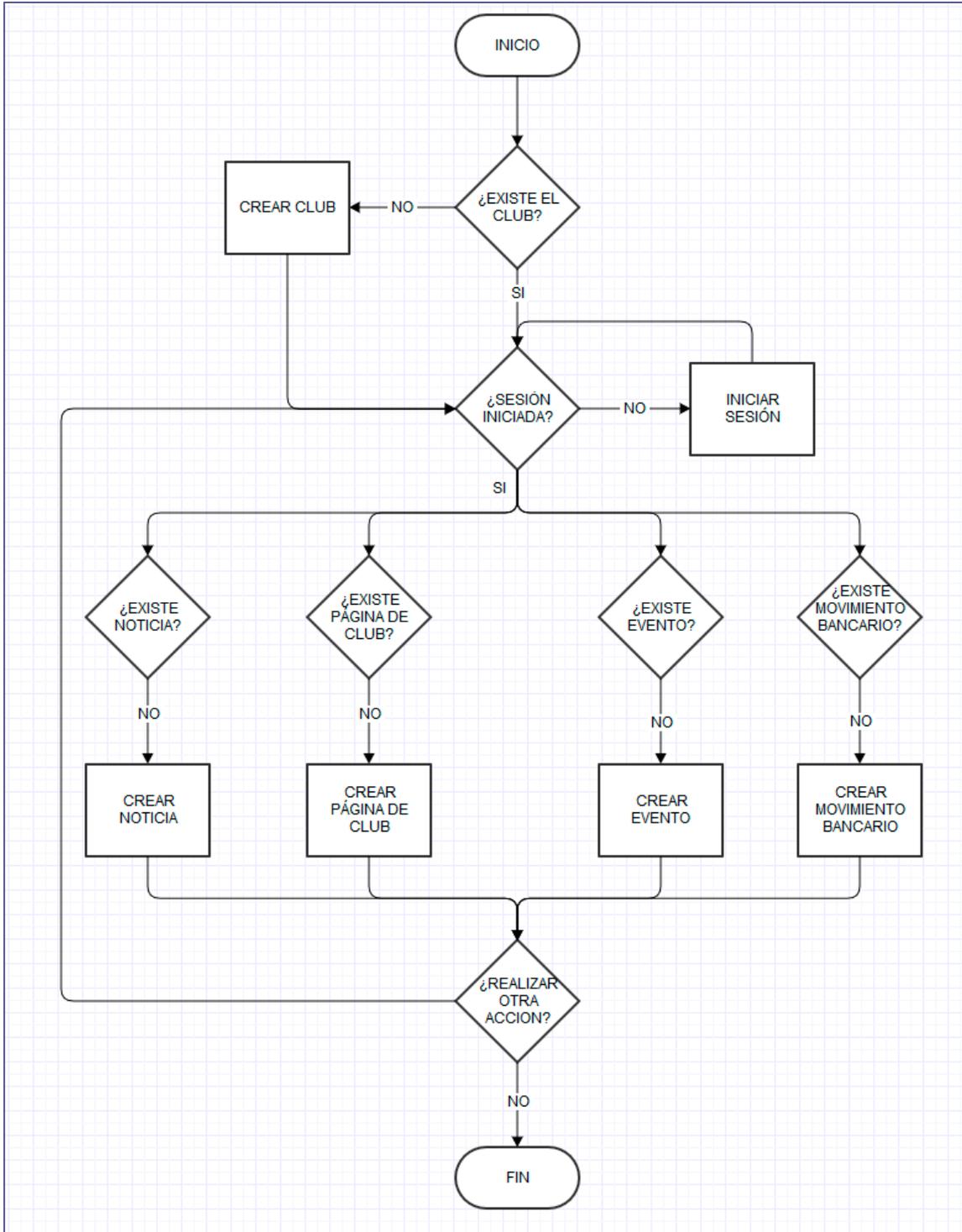


Figura 27. Diagrama de flujo para la gestión de la web.



4. Diseño

4.1. Introducción

Este proyecto ha sido realizado con un patrón de diseño software a tres capas (MVC). El objetivo de utilizar este patrón es dividir nuestra aplicación en paquetes modulares fáciles de mantener. Hacer uso de un diseño modular ayuda a los desarrolladores y diseñadores a trabajar simultáneamente y como el proyecto web ha sido realizado en pareja es de vital importancia utilizar un diseño de estas características. En nuestro caso, se han separado los datos de la aplicación (Modelo), el sistema de negocio (Controlador) y la interfaz (Vista).

4.2. Arquitectura del sistema

Actualmente existen muchos frameworks que hacen uso de este tipo de patrón software.

Tras examinar los frameworks más importantes, el que se ajusta a nuestras necesidades es CodeIgniter. Entre sus características encontramos que es un producto de código libre que implementa el proceso de desarrollo Modelo Vista Controlador.

Además es compatible con las últimas versiones de PHP, su documentación está escrita a modo de tutorial, lo que facilita su fácil entendimiento, y también incluye un amplio abanico de librerías que son muy útiles para el desarrollo de una aplicación web.

En la figura 28 se muestra un ejemplo de la arquitectura MVC utilizada por CodeIgniter.

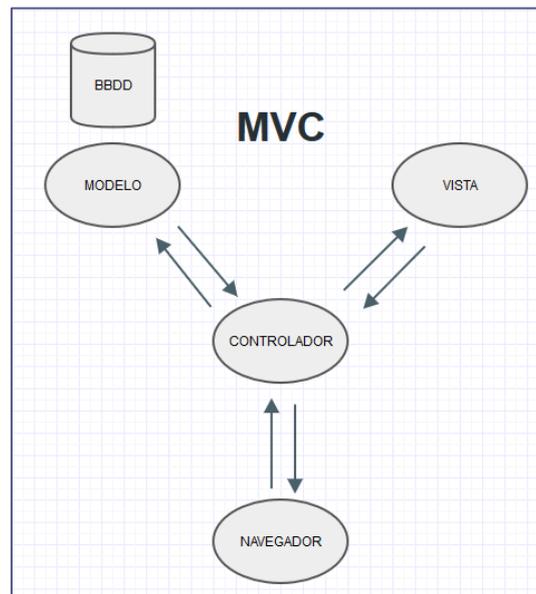


Figura 28. Arquitectura MVC.

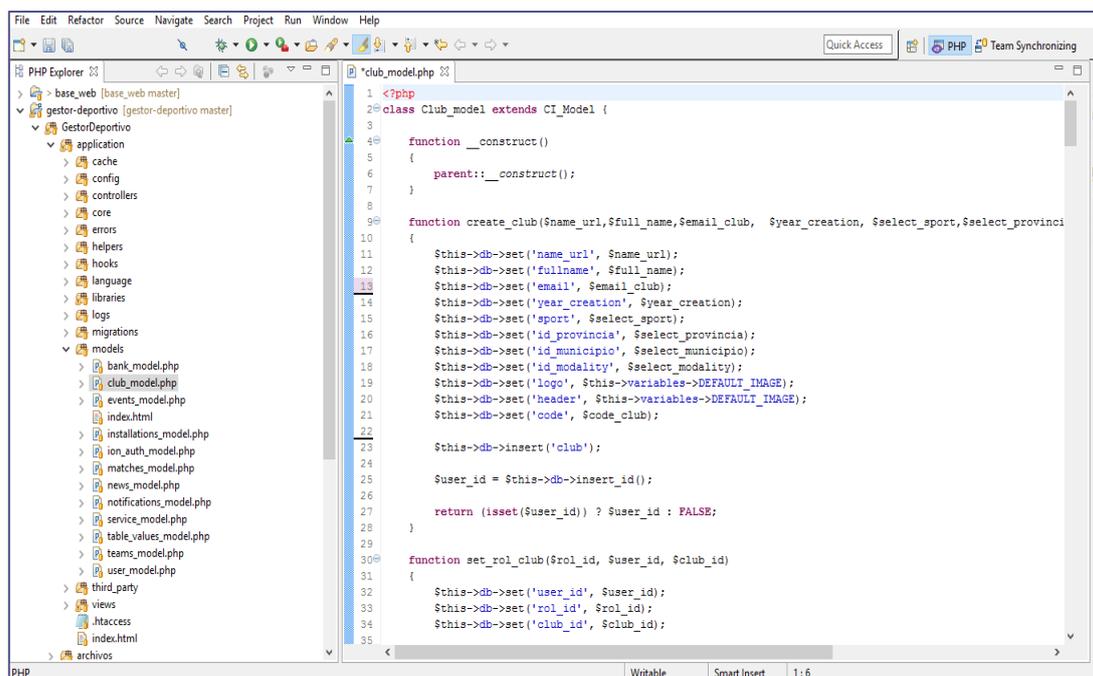
4.3. Modelo

La capa de modelo es la encargada de gestionar todos los accesos a base de datos. Por lo tanto todas las consultas, actualizaciones o inserciones se sitúan en esta capa. Por otro lado, CodeIgniter nos proporciona una librería llamada Active Record.

Esta librería se encarga de hacer que las consultas que realicemos sean independientes del lenguaje de base de datos, esto significa que si en un futuro se decide cambiar el motor de la base de datos, no sería necesario actualizar las todas las consultas.

Además dicha librería hace que las conexiones sean mucho más seguras ya que automáticamente todos los parámetros son escapados, evitando de este modo un ataque por inyección de SQL.

En la siguiente imagen (Figura 29) se muestra un ejemplo del código de la capa de modelo.



```
1 <?php
2 class Club_model extends CI_Model {
3
4     function __construct()
5     {
6         parent::__construct();
7     }
8
9     function create_club($name_url,$full_name,$email_club, $year_creation, $select_sport,$select_provinci
10
11     {
12         $this->db->set('name_url', $name_url);
13         $this->db->set('full_name', $full_name);
14         $this->db->set('email', $email_club);
15         $this->db->set('year_creation', $year_creation);
16         $this->db->set('sport', $select_sport);
17         $this->db->set('id_provincia', $select_provincia);
18         $this->db->set('id_municipio', $select_municipio);
19         $this->db->set('id_modalidad', $select_modalidad);
20         $this->db->set('logo', $this->variables->DEFAULT_IMAGE);
21         $this->db->set('header', $this->variables->DEFAULT_IMAGE);
22         $this->db->set('code', $code_club);
23
24         $this->db->insert('club');
25
26         $user_id = $this->db->insert_id();
27
28         return (isset($user_id)) ? $user_id : FALSE;
29
30     }
31
32     function set_rol_club($rol_id, $user_id, $club_id)
33     {
34         $this->db->set('user_id', $user_id);
35         $this->db->set('rol_id', $rol_id);
36         $this->db->set('club_id', $club_id);
```

Figura 29. Ejemplo de modelo.

4.4. Vista

La vista es la encargada de mostrar la información obtenida del modelo. Esta es la capa en la que interactúa el usuario final. Además las primeras validaciones se realizan en esta capa utilizando un lenguaje del lado del cliente como puede ser JavaScript.

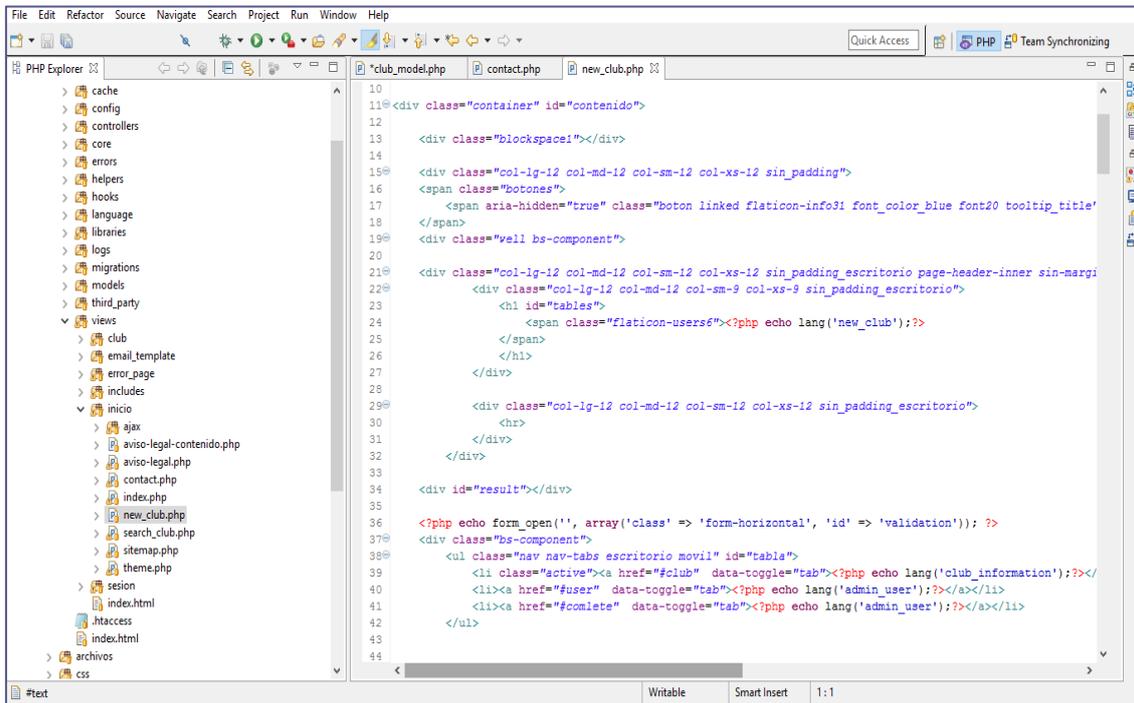
Este hecho hace que la experiencia final del usuario mejore ya que si los datos introducidos en el formulario no son válidos, este no será enviado al servidor, evitando de este modo una sobrecarga innecesaria. Aunque lo habitual es que la vista proporcione un fichero HTML, la respuesta también puede ser un XML, un vector JSON, una imagen o un video.



Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

Además se utilizan hojas de estilo CSS externas para mejorar la reutilización de código y la mantenibilidad de la interfaz.

La figura 30 muestra un ejemplo del código de la vista.



```
10
11 <div class="container" id="contenido">
12
13     <div class="blockspace"></div>
14
15     <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 sin_padding">
16         <span class="botones">
17             <span aria-hidden="true" class="boton linked flaticon-info31 font_color_blue font20 tooltip_title">
18         </span>
19     </div class="well bs-component">
20
21     <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 sin_padding escritorio page-header-inner sin-marg">
22         <div class="col-lg-12 col-md-12 col-sm-9 col-xs-9 sin_padding escritorio">
23             <h1 id="tables">
24                 <span class="flaticon-users6"><?php echo lang('new_club');?>
25             </span>
26         </h1>
27     </div>
28
29     <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 sin_padding escritorio">
30         <hr>
31     </div>
32 </div>
33
34 <div id="result"></div>
35
36 <?php echo form_open('', array('class' => 'form-horizontal', 'id' => 'validation')); ?>
37 <div class="bs-component">
38     <ul class="nav nav-tabs escritorio movil" id="table">
39         <li class="active"><a href="#club" data-toggle="tab"><?php echo lang('club_information');?></a></li>
40         <li><a href="#user" data-toggle="tab"><?php echo lang('admin_user');?></a></li>
41         <li><a href="#comiete" data-toggle="tab"><?php echo lang('admin_user');?></a></li>
42     </ul>
43
44
```

Figura 30. Ejemplo de vista.

4.5. Controlador

La principal función del controlador es la de actuar como intermediario entre la vista y el modelo. Es la capa responsable de interpretar y dar sentido a las peticiones que realiza el usuario.

También en esta capa se comprueba que el usuario tenga los permisos necesarios para poder visualizar el contenido solicitado o incluso realiza las peticiones de datos al modelo para después transmitirlos a la vista.

Además en el controlador también se validan los formularios o las diferentes peticiones que realiza el usuario, dando así un extra de seguridad, ya que las validaciones hechas en la vista, al producirse en el lado del cliente, es fácil que un usuario logre saltárselas.

A continuación, en la figura 31, se incluye una imagen de ejemplo de la capa de controlador.

```
18 }
19
20 public function index($name_url)
21 {
22     delete_cookie("cookie_url");
23     set_cookie('cookie_url', current_url(), '120');
24     $array_data['club'] = $this->Club_model->isClub($name_url);
25
26     //TITULOS Y PALABRAS CLAVES
27     $array_data['titulo_pagina'] = $array_data['club']->fullname." | ".$this->config->item('site_title');
28     $array_data['descripcion_pagina'] = lang('subtitle_club')." ".$array_data['club']->fullname;
29     $array_data['keywords_pagina'] = $this->config->item('site_title', 'ion_auth')." ".$array_data['club']->fullname;
30
31     if(!$array_data['club'])
32     {
33         redirect($name_url."/error_page/no_club");
34     }
35     else
36     {
37         $array_data['logout'] = $name_url;
38         $array_data += $this->menu_club->create_menu_club($this->ion_auth, $array_data['club']);
39
40         $this->load->view('includes/head', $array_data);
41         $this->load->view('includes/top_menu/menu');
42         $this->load->view('club/includes/header');
43         $this->load->view('club/includes/menu_club');
44         $this->load->view('club/main/news');
45         $this->load->view('includes/footer');
46     }
47 }
48
49 public function news($name_url, $offset=0)
50 {
51     $array_data['club'] = $this->Club_model->isClub($name_url);
52 }
```

Figura 31. Ejemplo de controlador.



5. Implementación

5.1 Introducción

A continuación se detalla la implementación de la aplicación web. En primer lugar se describen todas las tecnologías utilizadas. Después se detallarán los entornos que se han usado para la realización de la aplicación web.

5.2 Tecnologías

5.2.1. Codeigniter

Codeigniter es un marco de desarrollo de aplicaciones para personas que realizan sitios web usando PHP. Su principal objetivo es permitir el desarrollo de proyectos web de una forma más rápida que si se escribiera la aplicación desde cero. Codeigniter funciona con un conjunto de librerías que resuelve las tareas más comunes que se pueden encontrar en el desarrollo de una aplicación. Además, incluye una interfaz y estructura muy simple que facilita el acceso a todas las librerías.

Para el desarrollo de Yulava, se han utilizado las librerías y funcionalidades que vienen incluidas en Codeigniter y que a continuación se van a comentar.

5.2.1.1. Language

Language es un directorio donde se almacenan todos los archivos de idiomas. Esta funcionalidad es útil y da al proyecto un alto nivel de escalabilidad. Actualmente, la aplicación solo está disponible en español, pero en el supuesto de querer introducir otros idiomas, solo sería necesario añadir un nuevo directorio con el idioma deseado.

En la figura 32 se muestra un ejemplo del contenido del fichero idioma en español. Asimismo, en la figura 33 se puede observar como la capa de vista hace uso de las variables que se encuentran en el fichero anteriormente mencionado.

```

1 <?php if (! defined('BASEPATH')) exit('No direct script access allowed');
2
3
4 //INDEX
5 $lang['search_your_club'] = 'Busca un club';
6 $lang['go_club'] = 'Ir al club';
7 $lang['your_club?'] = '¿Tu club aún no está en YULAVA?';
8 $lang['init_session_club'] = 'Inicia sesión en tu club';
9 $lang['made_club'] = 'Crear tu Club';
10 $lang['legal'] = 'Aviso Legal';
11 $lang['new_user'] = 'Crear usuario';
12 $lang['rol_admin_user'] = 'Roles aparte de Administrador';
13 $lang['new_club'] = 'Crear club';
14 $lang['select_image'] = 'Seleccionar';
15 $lang['init_session'] = 'Iniciar sesión';
16 $lang['title_info'] = 'Información de la página';
17 $lang['options'] = 'Opciones';
18 $lang['welcome'] = 'Bienvenido';
19 //CLUB
20
21 $lang['domain_name'] = 'Nombre de dominio';
22 $lang['domain_example'] = 'http://www.yulava.es/nombreDominio';
23 $lang['full_name_club'] = 'Nombre completo del club';
24 $lang['sport'] = 'Deporte';
25 $lang['modality'] = 'Modalidad';
26 $lang['select_sport'] = 'Elige un deporte';
27 $lang['select_modality'] = 'Elige una modalidad';
28 $lang['provincia'] = 'Provincia';
29 $lang['municipio'] = 'Municipio';
30 $lang['club_information'] = 'Información del club';
31 $lang['create_year'] = 'Año de creación';
32 $lang['email_club'] = 'Email del club';
33 $lang['page_club'] = 'Crear página';
34 $lang['title_club'] = 'Titulo de la página';
35 $lang['visible'] = 'Página visible';
36 $lang['edit_page'] = 'Editar página';
37 $lang['show_page'] = 'Ver página';
38 $lang['edit_page'] = 'Editar página';
39 $lang['delete_page'] = 'Borrar página';
40 $lang['delete'] = 'Eliminar';
41 $lang['edit'] = 'Editar';
42 $lang['complet'] = 'Completado';
43 $lang['preferences'] = 'Preferencias';
44 $lang['social_network'] = 'Redes sociales';
45 $lang['facebook'] = 'Facebook';
46 $lang['twitter'] = 'Twitter';
47 $lang['google_plus'] = 'Google+';
48 $lang['web_club'] = 'Web del club';
49 $lang['correct_edit_club'] = 'Información editada correctamente';
50 $lang['club_data'] = 'Datos del club';
51

```

Figura 32. Ejemplo del fichero club_lang.

```

<div class="form-group">
  <label for="email" class="col-lg-4 control-label"><?php echo lang('registro_email_label');?>
  </label>
  <div class="col-lg-8">
    <input type="text" class="form-control validate[custom[email]]"
      id="email" name="email">
  </div>
</div>

```

Figura 33. Ejemplo de la utilización de la función lang.

5.2.1.2. Form Validation

Form Validation^[15] Proporciona una serie de utilidades que ayudan a validar los formularios que se han enviado desde la vista, y de este modo, cumplan todos los requisitos previstos. Entre sus utilidades se encuentra la validación del tamaño de la cadena, si el campo es requerido o también se pueden utilizar expresiones regulares ya definidas por Codeigniter, por ejemplo para comprobar si un correo electrónico ha sido bien introducido. En la figura 34 se muestra un ejemplo de la librería Form validation.



```
//validate form input
$this->form_validation->set_rules('first_name', $this->lang->line('edit_user_fname_label'), 'required');
$this->form_validation->set_rules('last_name', $this->lang->line('edit_user_lname_label'), 'required');
$this->form_validation->set_rules('last2_name', $this->lang->line('edit_user_l2name_label'), 'required');

$this->form_validation->set_rules('mask_date', $this->lang->line('mask_date'), 'required');

$this->form_validation->set_rules('dni', $this->lang->line('registro_DNI_label'));
$this->form_validation->set_rules('email', $this->lang->line('edit_user_email_label'), 'valid_email');
$this->form_validation->set_rules('mask_phone', $this->lang->line('edit_user_phone_label'));

if ($this->form_validation->run() == true)
{
```

Figura 34. Ejemplo del uso de la librería Form Validation.

5.2.1.3. Active record

Codeigniter utiliza una versión modificada del modelo de base de datos Active Record^[6]. Esta librería se encarga de hacer que las consultas que se realizaren sean independientes del lenguaje de base de datos. En la figura 35 se muestra un ejemplo del uso de esta librería.

```
public function list_matches($id_club, $id_team)
{

    $this->db->select('id_match, name_team, name_rival, logo, place.description as place,
        status.description as status, date, Championship.name as championship ');
    $this->db->join('rival', 'rival.id_rival = matches.rival_id');
    $this->db->join('team', 'team.id_team = matches.team_id');
    $this->db->join('place', 'place.id_place = matches.place_id');
    $this->db->join('status', 'status.id_status = matches.status_id');
    $this->db->join('championship', 'championship.id_championship = matches.id_championship');
    $this->db->where('team.club_id', $id_club);
    $this->db->where('team.id_team', $id_team);
    $this->db->order_by("id_match", "desc");
    $this->db->from('matches');

    $consulta = $this->db->get();

    return $consulta->result();
}
```

Figura 35. Ejemplo del uso de la librería Form Validation.

5.2.1.4. IonAuth

A pesar de que esta librería no es propia de Codeigniter, su implementación sí que es exclusiva de este Framework. IonAuth es utilizado para la autenticación de los usuarios. Además, incluye funciones que ayudan a recuperar la contraseña del usuario, mediante el envío de un email, se incluye una clave de recuperación que sólo podrá ser usada una vez y durante un tiempo limitado. También se incluyen funciones para elegir la forma en la que el usuario iniciará sesión; bien mediante el nombre de usuario o por medio del email. En la figura 36 se puede ver un ejemplo del controlador con la función que realiza el login.

```

//log the user in
public function login()
{
    //validate form input
    $this->form_validation->set_rules('identity', $this->lang->line('login_identity_label'), 'required');
    $this->form_validation->set_rules('password', $this->lang->line('login_password_label'), 'required');

    if ($this->form_validation->run() == true)
    {

        //check to see if the user is logging in
        //check for "remember me"
        $remember = (bool) $this->input->post('remember');
        if (!$this->ion_auth->login($this->input->post('identity'), $this->input->post('password'), $remember))
        {
            header('HTTP/1.1 400 Bad Request');
            $array_message['message'] = $this->lang->line('error_login');
            $page_html = $this->load->view('includes/messages/error_page', $array_message, true);
            exit($page_html);
        }else{

            if(get_cookie('cookie_url')){
                echo get_cookie('cookie_url');
            }else{
                $this->load->model('Club_model');
                $name_url = $this->Club_model->get_club_name($this->ion_auth->user()->row()->id);
                echo base_url().$name_url;
            }
            exit();
        }
    }
    else
    {
        header('HTTP/1.1 400 Bad Request');
        $array_message['message'] = validation_errors();
        $page_html = $this->load->view('includes/messages/error_page', $array_message, true);
        exit($page_html);
    }
}
}

```

Figura 36. Ejemplo del uso de la librería IonAuth.

5.2.1.5. Sesión

La librería de sesión ^[17] permite al usuario mantener un estado y seguir su actividad mientras navega por el sitio web. Esta librería es usada por IonAuth. Haciendo uso de las variables de sesión se conoce en todo momento que usuario es el que está utilizando la web.

5.2.1.6. Routes

El fichero routes se encuentra en el directorio config. Dicho fichero se encarga de conseguir que las URL que se introducen en la aplicación web sean amigables. Internamente traduce la URL introducida por el usuario y redirecciona al controlador deseado. En la figura 37 se puede ver un ejemplo de estas redirecciones.

```
$route['(:any)/temporada/(:any)'] = "club/season_team/$1/$2";

$route['(:any)/matches/create/(:any)'] = "admin/matches/view_create/$1/$2";
$route['(:any)/matches/edit/(:any)'] = "admin/matches/view_edit/$1/$2";
$route['(:any)/partido/(:any)'] = "admin/matches/view_match/$1/$2";
$route['(:any)/matches/create_match/(:any)'] = "admin/matches/create_match/$1/$2";
$route['(:any)/matches/edit_match/(:any)'] = "admin/matches/edit_match/$1/$2";
$route['(:any)/matches/convocation/(:any)'] = "admin/matches/view_create_convocation/$1/$2";
$route['(:any)/matches/create_convocation/(:any)'] = "admin/matches/create_convocation/$1/$2";
$route['(:any)/matches/statistics/(:any)'] = "admin/matches/view_create_statistics/$1/$2";
$route['(:any)/matches/create_statistics/(:any)'] = "admin/matches/create_statistics/$1/$2";
$route['(:any)/matches/edit_live/(:any)'] = "admin/matches/edit_live/$1/$2";
$route['(:any)/matches/create_message/(:any)'] = "admin/matches/create_message/$1/$2";
$route['(:any)/matches/list_photos/(:any)'] = "admin/matches/list_photos/$1/$2";
$route['(:any)/matches/delete_match'] = "admin/matches/delete_match/$1";
$route['(:any)/matches/list_message/(:any)'] = "admin/matches/list_message/$1/$2";

$route['(:any)/partidos'] = "admin/matches/view_calendar/$1";
$route['matches/list_matches/(:any)'] = "admin/matches/list_matches/$1";
$route['matches/list_matches/(:any)/(:any)'] = "admin/matches/list_matches/$1/$2";
$route['(:any)/matches/(:any)'] = "admin/matches/index/$1/$2";

$route['notifications/list_notifications'] = "admin/notifications/list_notifications/$1";
$route['notifications/list_notifications/(:any)'] = "admin/notifications/list_notifications/$1/$2";
$route['notifications/set_read'] = "admin/notifications/set_read";
$route['notifications/set_read_all'] = "admin/notifications/set_read_all";
$route['(:any)/notifications'] = "admin/notifications/index/$1";

$route['(:any)/(:any)'] = "error_page/no_found/$1";
$route['(:any)'] = "club/index/$1";
```

Figura 37. Ejemplo del uso de la librería IonAuth.

5.2.2. PHP5

PHP^[18] es un lenguaje de programación de uso general de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.

Se ha utilizado PHP para desarrollar toda la parte Backend, donde se incluyen todos los controladores y los modelos, así como algunas librerías.

5.2.3. MySQL

MySQL^[19] es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

5.2.4. JSON

JSON^[20] es un formato que se utiliza principalmente para el intercambio de datos. Una de las ventajas que tiene JSON es que puede ser leído por cualquier lenguaje de programación, además, su estructura y su fácil creación (recuerda a un array), hace que en los últimos años sea uno de los formatos más usados, dejando de lado el uso del XML.

En la figura 38 se puede ver un ejemplo de varias peticiones hechas por el cliente y como el servidor devuelve un JSON con la información deseada. Después, la vista se encarga de leer los datos e insertarlos en el lugar deseado.

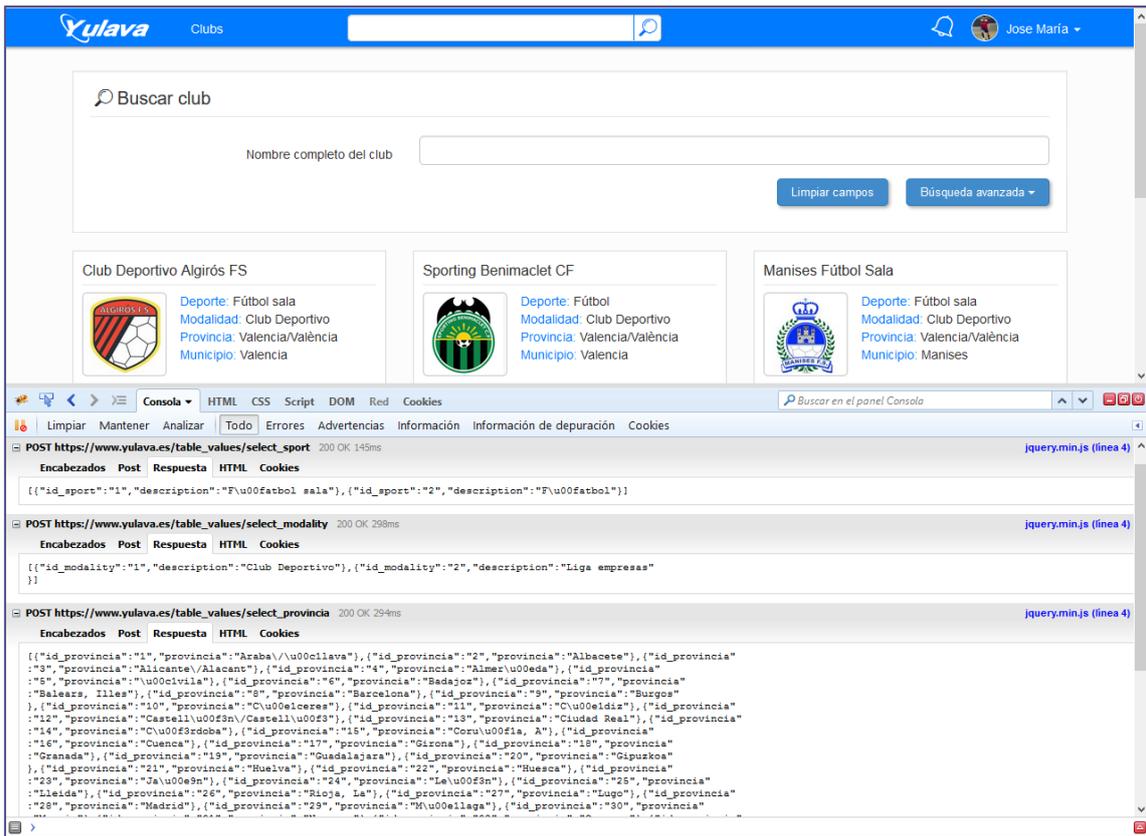


Figura 38. Ejemplo del uso del formato JSON.

En la figura 39 se puede ver otro ejemplo acerca del uso de JSON. En esta ocasión, el enfoque que se ha dado es totalmente diferente. Nos situamos en el hecho de que las estadísticas de un jugador pueden variar dependiendo del deporte que practica. Por este motivo, en lugar de crear un campo en la base de datos con cada estadística, la cual no sería nada escalable, se genera un JSON con todos los tipos de estadísticas que podemos encontrar en cada deporte, y una vez el JSON ha sido generado se almacena en un único campo de la base de datos, aumentando su escalabilidad y su rendimiento. Después, solo hay que leer dicho campo y dibujar las estadísticas que dispone cada jugador.



1. Creación del JSON

```
if(isset($_POST['penalti_parado'])){\n    $statistics['penalti_parado'] = $_POST['penalti_parado'];\n}\nif(isset($_POST['paradas'])){\n    $statistics['paradas'] = $_POST['paradas'];\n}\nif(isset($_POST['penalti_encajado'])){\n    $statistics['penalti_encajado'] = $_POST['penalti_encajado'];\n}\nif(isset($_POST['doblepenalti_marcado'])){\n    $statistics['doblepenalti_marcado'] = $_POST['doblepenalti_marcado'];\n}\nif(isset($_POST['doblepenalti_fallado'])){\n    $statistics['doblepenalti_fallado'] = $_POST['doblepenalti_fallado'];\n}\n\n$jsondata = json_encode($statistics);
```

2. JSON en base de datos

match_id	user_id	data
44	6	{\"titular\":1,\"goles\":\"0\",\"goles_encajados\":\"0\",\"minutos\":\"0\",\"amarillas\":\"0\",\"rojas\":\"0\",\"asistencias\":\"0\",\"tiros_puerta\":\"5\",\"tiros_fuera\":\"1\",\"faltas_cometidas\":\"0\",\"faltas_recibidas\":\"1\",\"balones_recuperados\":\"0\",\"balones_perdidos\":\"1\",\"penalti_marcado\":\"0\",\"penalti_fallado\":\"0\",\"penalti_parado\":\"0\",\"penalti_encajado\":\"0\",\"doblepenalti_marcado\":\"0\",\"doblepenalti_fallado\":\"0\"}
44	4	
44	7	
44	32	{\"titular\":1,\"goles\":\"0\",\"goles_encajados\":\"2\",\"mi...

3. Lectura del JSON

```
foreach ($data as $json){\n\n    $stats = json_decode($json->data);\n\n    isset($stats->goles) ? $goles += $stats->goles : $goles = null;\n    isset($stats->amarillas) ? $amarillas += $stats->amarillas : $amarillas = null;\n    isset($stats->rojas) ? $rojas += $stats->rojas : $rojas = null;
```

Figura 39. Ejemplo de uso del formato JSON.

5.2.5. Conexión Modelo - Vista - Controlador

A lo largo del proyecto, se ha mencionado que se ha utilizado para el desarrollo de esta aplicación web el patrón Modelo-Vista-Controlador. Dicha aplicación depende de los controladores, ya que son los encargados de gestionar las relaciones entre el modelo y la vista. En la figura 40 se puede ver un ejemplo de cómo el controlador obtiene los datos del modelo y los envía a la vista para que sean mostrados al usuario final.

1. En el constructor se cargan los modelos necesarios

```
class Club extends CI_Controller {  
  
    public function __construct()  
    {  
        parent::__construct();  
        $this->load->model('Club_model');  
        $this->load->library('menu_club');  
        $this->load->library('breadcrumb');  
    }  
}
```

2. Se piden los datos al modelo y se envían a la vista

```
//SOLO ACCEDE SI ERES ADMINISTRADOR  
if($array_data['permission']['administrador'])  
{  
    $array_data['pages'] = $this->Club_model->pages_club($array_data['club']->id_club);  
    $this->load->view('club/admin/club/index', $array_data);  
}  
else  
{  
    redirect($name_url."/error_page/permission_error");  
}
```

Figura 40. Ejemplo de cómo se conecta el modelo con la vista a través del controlador

5.3. Herramientas y entorno

5.3.1. Entorno de desarrollo

5.3.1.1. Eclipse

El proyecto ha sido producido con el entorno de desarrollo Eclipse^[21], uno de los motivos de su elección es que está creado con Java. Esto significa que el programa puede ser usado en cualquier plataforma, ya sea Windows, Mac o Linux. Además otra característica es que es de código abierto. Eclipse cuenta con un analizador sintáctico que facilita la lectura del código, incluso, integra un control de versiones GIT, ayudando a mantener la última versión del código. La versión de Eclipse que se ha utilizado para el desarrollo de la aplicación es Eclipse for PHP Developers -Version: Mars Release (4.5.0).

5.3.1.2. Sybase Powerdesigner

Sybase Powerdesigner^[22] es un programa para la gestión y modelado de bases de datos. Tiene la capacidad de generar modelos físicos, conceptuales y orientados a objetos. Una vez ha sido creado todo el modelo de base de datos, existe una función para generar un script sql que puede ser importado en la mayoría de gestores, facilitando de este modo la creación de las bases de datos. Ver detalle en la figura 41

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

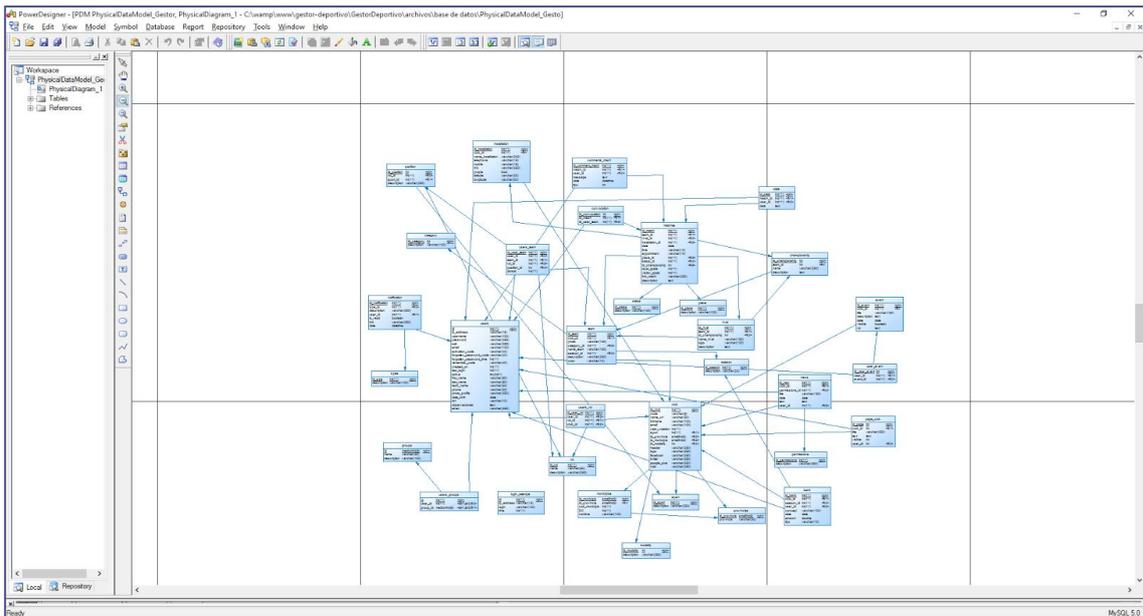


Figura 41. Ejemplo del uso del formato JSON.

5.3.1.3. PhpMyAdmin

PhpMyAdmin^[23] es una herramienta web, creada con PHP. Es utilizada para la administración de bases de datos MySQL a través de la web. Dicha herramienta facilita las actividades de crear, eliminar o actualizar tablas en la base de datos. Ver detalle en la figura 42.

The image shows a screenshot of the phpMyAdmin web interface. The interface is divided into several sections. On the left, there is a sidebar with a tree view showing the database structure, including a 'New' button and a search box. The main area displays a list of tables for a database named 'yslvra'. The table list has columns for 'Table', 'Action', 'Rows', 'Type', 'Collation', 'Size', and 'Overhead'. Each table entry includes a star icon, a 'Browse' button, and several icons for 'Structure', 'Search', 'Insert', 'Empty', and 'Drop'. The tables listed include 'bank', 'category', 'championship', 'club', 'comments_mach', 'convocation', 'event', 'groups', 'installation', 'login_attempts', 'matches', 'modality', 'municipios', 'news', 'notification', 'page_club', 'permissions', 'place', 'provincias', 'rival', 'rol', 'season', 'sport', and 'stats'.

Table	Action	Rows	Type	Collation	Size	Overhead
bank	Browse Structure Search Insert Empty Drop	84	InnoDB	latin1_swedish_ci	84 K	-
category	Browse Structure Search Insert Empty Drop	28	InnoDB	latin1_swedish_ci	32 K	-
championship	Browse Structure Search Insert Empty Drop	34	InnoDB	latin1_swedish_ci	32 K	-
club	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	88 K	-
comments_mach	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	48 K	-
convocation	Browse Structure Search Insert Empty Drop	493	InnoDB	latin1_swedish_ci	48 K	-
event	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	32 K	-
groups	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	32 K	-
installation	Browse Structure Search Insert Empty Drop	43	InnoDB	latin1_swedish_ci	32 K	-
login_attempts	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	32 K	-
matches	Browse Structure Search Insert Empty Drop	184	InnoDB	latin1_swedish_ci	112 K	-
modality	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	32 K	-
municipios	Browse Structure Search Insert Empty Drop	8,134	InnoDB	latin1_swedish_ci	844 K	-
news	Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	112 K	-
notification	Browse Structure Search Insert Empty Drop	2,492	InnoDB	latin1_swedish_ci	432 K	-
page_club	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	48 K	-
permissions	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	32 K	-
place	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	32 K	-
provincias	Browse Structure Search Insert Empty Drop	28	InnoDB	latin1_swedish_ci	48 K	-
rol	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	32 K	-
season	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	32 K	-
sport	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	32 K	-
stats	Browse Structure Search Insert Empty Drop	187	InnoDB	latin1_swedish_ci	144 K	-

Figura 42. Ejemplo de la interfaz de phpMyAdmin.

5.3.2. Control de versiones

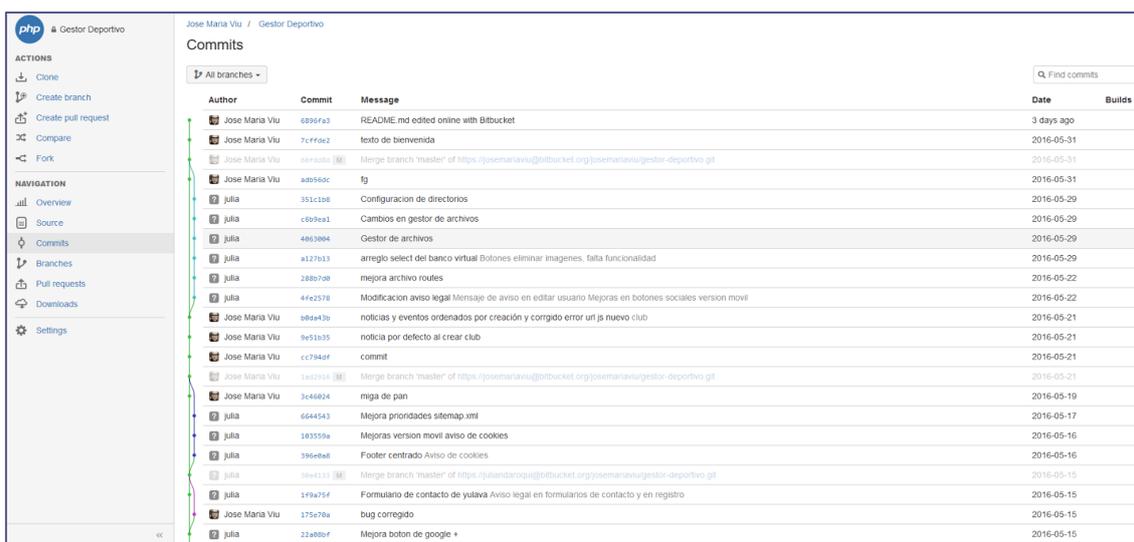
Como se ha mencionado anteriormente, Eclipse incluye GIT como control de versión. Se llama control de versiones ^[24] a la gestión de los cambios que se realizan sobre los elementos del proyecto. Una versión de un producto es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

5.3.1.1. GIT

GIT^[25] es la tecnología que se ha usado para controlar todas las versiones que se han generado en el transcurso del desarrollo del proyecto web. GIT, además, es de código abierto y ha sido diseñado para proyectos pequeños o de gran envergadura.

5.3.1.2. Bitbucket

Bitbucket^[26] es una plataforma web que es utilizada como repositorio para proyectos web que utilizan un sistema de control de versiones como puede ser mercurial o GIT. Se puede encontrar Bitbucket de manera gratuita. Ver detalle en la figura 43.



The screenshot shows the Bitbucket interface for a repository named 'Gestor Deportivo'. The main content is a list of commits, with a search bar and a dropdown menu for branches. The commits are listed in a table with columns for Author, Commit ID, Message, Date, and Builds. The commits are ordered by date, with the most recent at the top. The interface also includes a sidebar with navigation options like 'Clone', 'Create branch', 'Compare', 'Fork', 'Overview', 'Source', 'Commits', 'Branches', 'Pull requests', 'Downloads', and 'Settings'.

Author	Commit	Message	Date	Builds
Jose Maria Vlu	699f7a3	README.md edited online with Bitbucket	3 days ago	
Jose Maria Vlu	7c4f9a2	texto de bienvenida	2016-05-31	
Jose Maria Vlu	6a4a2a4	Merge branch "master" of https://josemaria.vlu@bitbucket.org/josemaria.vlu/gestor-deportivo.git	2016-05-31	
Jose Maria Vlu	ad956dc	fg	2016-05-31	
julia	351c1b8	Configuracion de directorios	2016-05-29	
julia	c199ca1	Cambios en gestor de archivos	2016-05-29	
julia	4063804	Gestor de archivos	2016-05-29	
julia	a127613	arreglo select del banco virtual Botones eliminar imagenes, talla funcionalidad	2016-05-29	
julia	288708	mejora archivo routes	2016-05-22	
julia	4fe2578	Modificacion aviso legal Mensaje de aviso en editar usuario Mejoras en botones sociales version movi	2016-05-22	
Jose Maria Vlu	9b8443b	noticias y eventos ordenados por creación y corregido error url js nuevo club	2016-05-21	
Jose Maria Vlu	9e51b35	noticia por defecto al crear club	2016-05-21	
Jose Maria Vlu	cc794df	commit	2016-05-21	
Jose Maria Vlu	1a45916	Merge branch "master" of https://josemaria.vlu@bitbucket.org/josemaria.vlu/gestor-deportivo.git	2016-05-21	
Jose Maria Vlu	3c46804	miga de pan	2016-05-19	
julia	6644543	Mejora prioridades sitemap.xml	2016-05-17	
julia	183559a	Mejoras version movi aviso de cookies	2016-05-16	
julia	396e0a8	Footer centrado Aviso de cookies	2016-05-16	
julia	3944113	Merge branch "master" of https://julianderoga@bitbucket.org/josemaria.vlu/gestor-deportivo.git	2016-05-15	
julia	149a75f	Formulario de contacto de ystava Aviso legal en formularios de contacto y en registro	2016-05-15	
Jose Maria Vlu	175e70a	bug corregido	2016-05-15	
julia	22a880f	Mejora boton de google +	2016-05-15	

Figura 43. Ejemplo de la interfaz de Bitbucket

5.3.3. Servidor web

Como servidor HTTP se ha usado Apache^[27]. El servidor Apache es un servidor web HTTP de código abierto, disponible para todas las plataformas. Este tipo de servidor es modular y ofrece gran cantidad de módulos. Dicho servidor es utilizado para recibir peticiones de los clientes y devolver los ficheros necesarios para formar la interfaz de usuario.

6. Evaluación

6.1. Introducción

Se han realizado una serie de pruebas funcionales para comprobar el correcto funcionamiento de la aplicación. Además, se ha intentado llevar al límite la aplicación haciendo un ataque por SQL INJECTION para comprobar que, efectivamente, el uso de la librería Active record explicada anteriormente funciona de la manera deseada.

6.2. Pruebas funcionales

Una prueba funcional [28] es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software.

Una de las pruebas funcionales más importantes es comprobar que los usuarios solo se les permite acceder a los recursos que disponen autorización.

A continuación se va a realizar un ejemplo completo, para comprobar el correcto funcionamiento de la aplicación.

1. Accedemos como usuario administrador del club Deportivo Algorós FS. Como se puede apreciar en la figura 44 se ha iniciado sesión con un usuario que además de tener el rol de administrador, también dispone del rol de entrenador y jugador. El menú lateral de la izquierda muestra todas las opciones que este usuario como administrador tiene en el club.

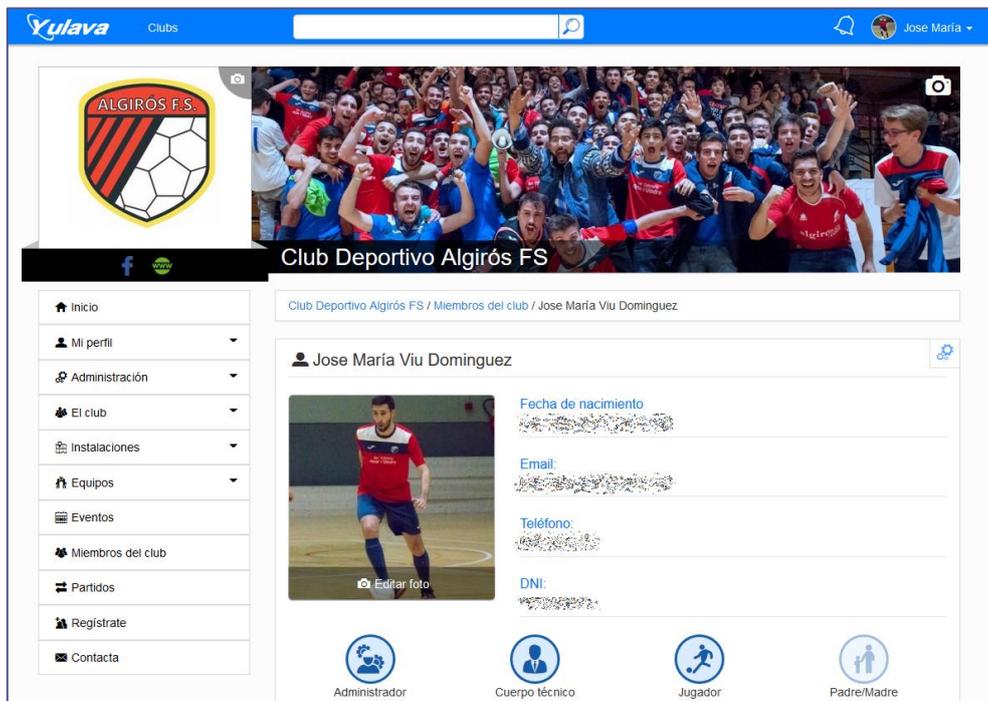


Figura 44. Vista de perfil del usuario administrador

2. La primera validación que se ha realizado es acceder a otro club y comprobar que el menú lateral izquierdo, tan solo muestra las opciones públicas del club. En la figura 45 se puede comprobar que las opciones del menú han disminuido tal y como se esperaba.

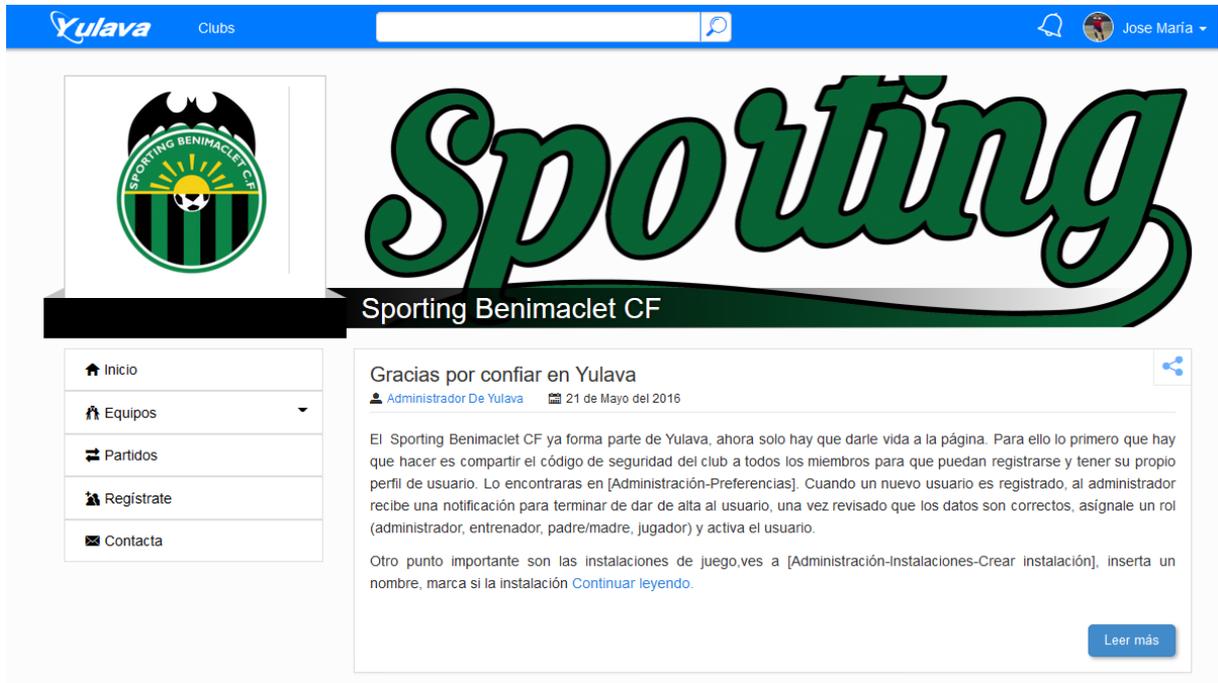


Figura 45. Vista principal de un club.

3. La siguiente acción que se ha realizado, es intentar forzar la entrada al panel de administración de, por ejemplo, la creación de un equipo en un club que no es el del usuario administrador. Para poder realizar este proceso, se cambia la URL manualmente ya que las opciones no se muestran al no pertenecer en este club.
 - a. En primer lugar se accede a crear equipo en el club del usuario administrador. URL del ejemplo: <https://www.yulava.es/algirosfs/teams/create> Figura 46.

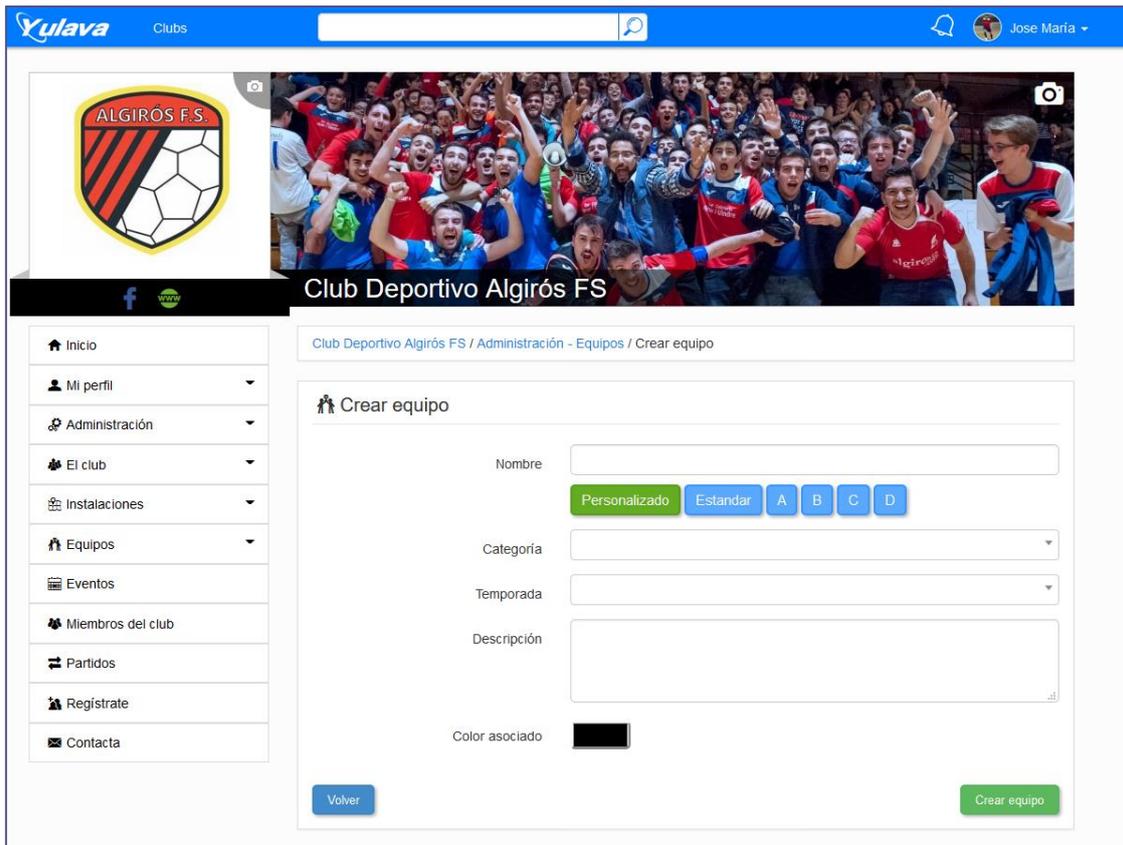


Figura 46. Vista de creación de un equipo

- b. Ahora se procede a cambiar en la URL el nombre del equipo, quedando de esta forma <https://www.yulava.es/spbenimaclet/teams/create> . En la figura 47 se puede visualizar que el resultado final es el esperado.

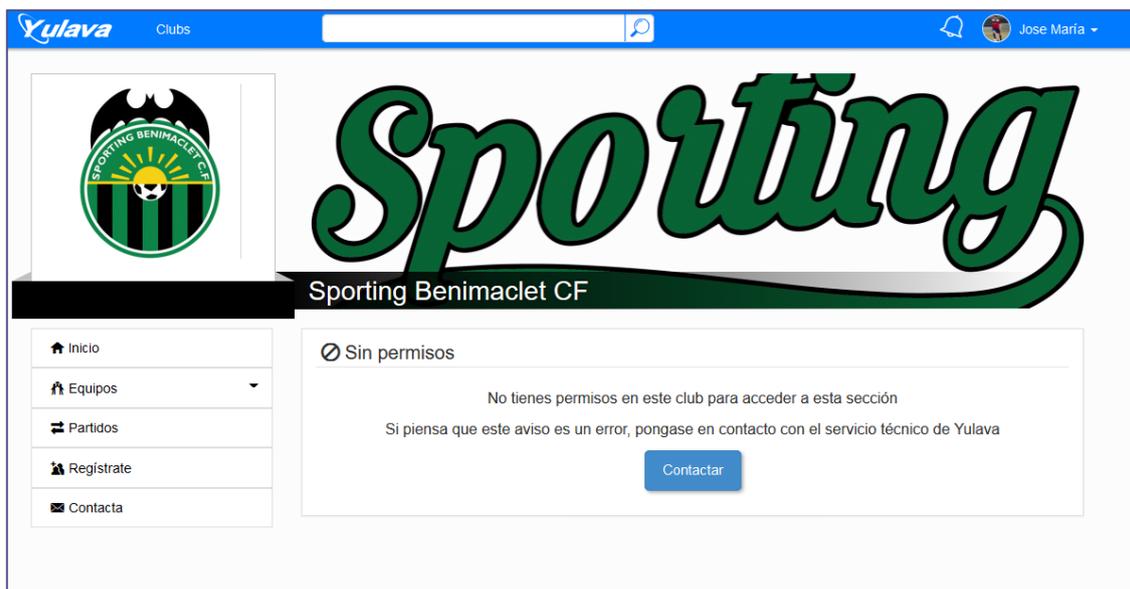


Figura 47. Vista de negación de permisos

Antes de publicar la aplicación en un entorno de producción final, todas las pruebas funcionales han tenido que ser satisfactorias. Comprobando los permisos de todos los tipos de usuario, entre distintos clubes como en el ejemplo anterior, o en el mismo club cambiando los roles del usuario. Independientemente de la comprobación de los roles, también se ha comprobado que todas las funciones que ofrece la aplicación funcionan correctamente.

6.3. Seguridad del sistema.

Inyección SQL^[29] es un método que infiltra código intruso en una aplicación en el nivel de validación de las entradas para realizar operaciones en la base de datos.

Por ejemplo, viendo la URL del perfil de usuario, <https://www.yulava.es/algiosfs/user/josemariaviu> se puede deducir que internamente se hace una consulta SQL para obtener la información del usuario.

```
SELECT * FROM user WHERE username = 'josemariaviu'
```

Entonces, para realizar un ataque, lo que habría que hacer es cambiar el nombre del usuario por un nombre que incluya una sentencia SQL. Siendo un poco maliciosos, la consulta podrá eliminar la tabla usuarios del sistema de base de datos. Quedando el nombre del usuario de esta forma:

```
https://www.yulava.es/algiosfs/user/josemariaviu" DROP TABLE user
```

Al insertar esta URL comprobamos que no se ha eliminado ninguna tabla gracias a que la librería ACTIVE RECORD utiliza consultas parametrizadas, y además escapa los caracteres especiales, evitando de este modo ataques indeseados.

6.4. Rendimiento del segmento de usuarios.

Gracias a la tecnología que ofrece Google Analytics se puede tener un control de las estadísticas que produce la aplicación web.

6.4.1. Visitas

En la figura 48 podemos apreciar el número de sesiones, usuarios o número de visitas a páginas durante todo el mes de mayo. Analizando las estadísticas, se afirma que un usuario permanece en la página una media de 7 minutos y que suele volver a visitar la aplicación ya que tan solo hay un 20 % de nuevos visitantes.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Back-End)

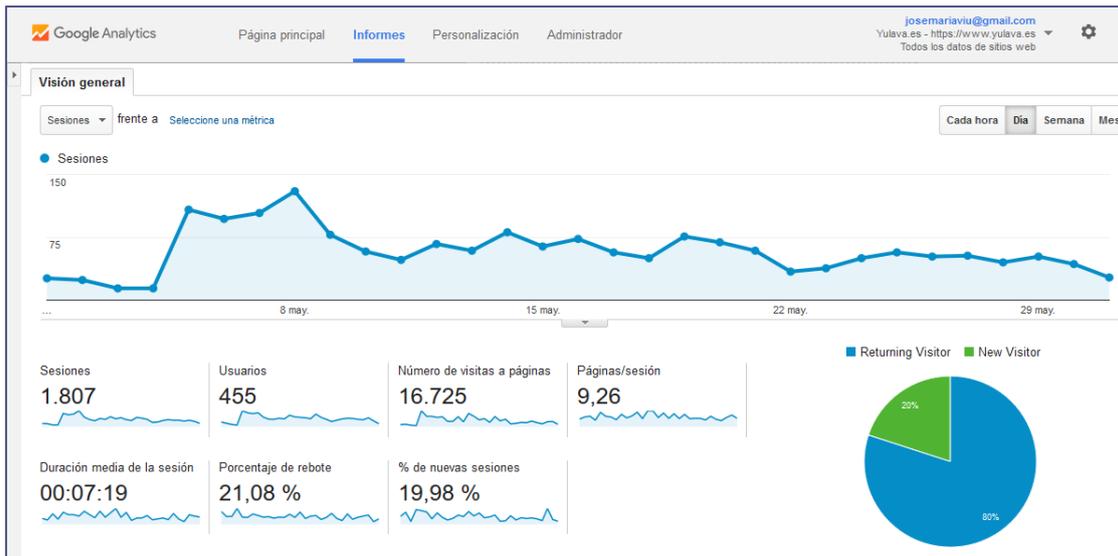


Figura 48. Vista global de las vistas durante el mes de mayo.

6.4.2. Dispositivos

Observando la figura 49 se corrobora que se ha cumplido uno de los principales objetivos; la aplicación web puede ser utilizada en cualquier tipo de dispositivo ya sea móvil, Tablet o un ordenador. En la figura 49 se observa que durante todo el mes de mayo, un 70 % de los usuarios que han accedido a Yulava, lo han hecho a través de un dispositivo móvil.

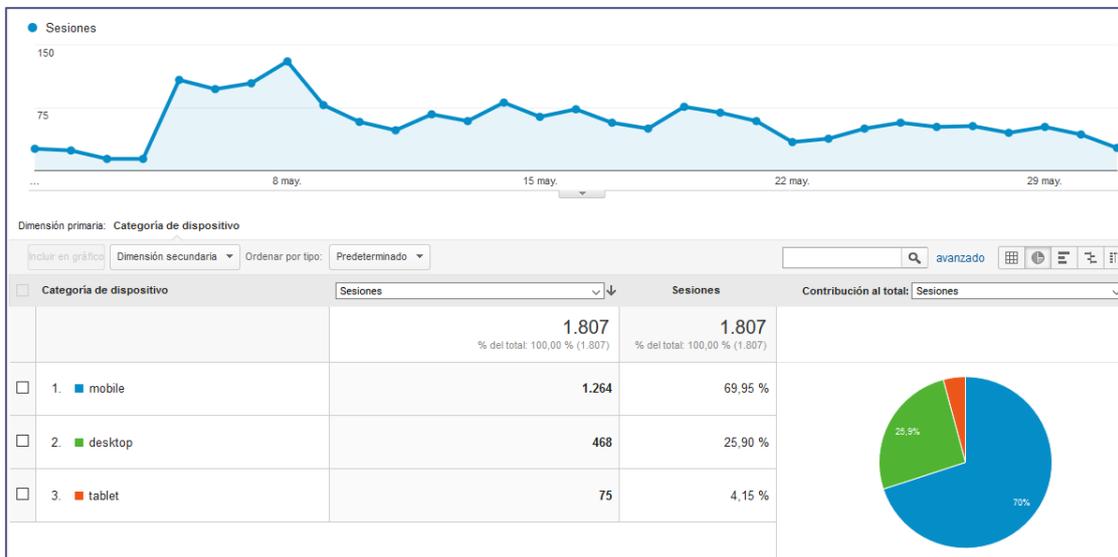


Figura 49. Vista global de los dispositivos usados durante el mes de mayo.

6.4.3 Flujo de usuarios.

A continuación en la figura 50 se puede apreciar todas las interacciones que un usuario realiza en la aplicación web.

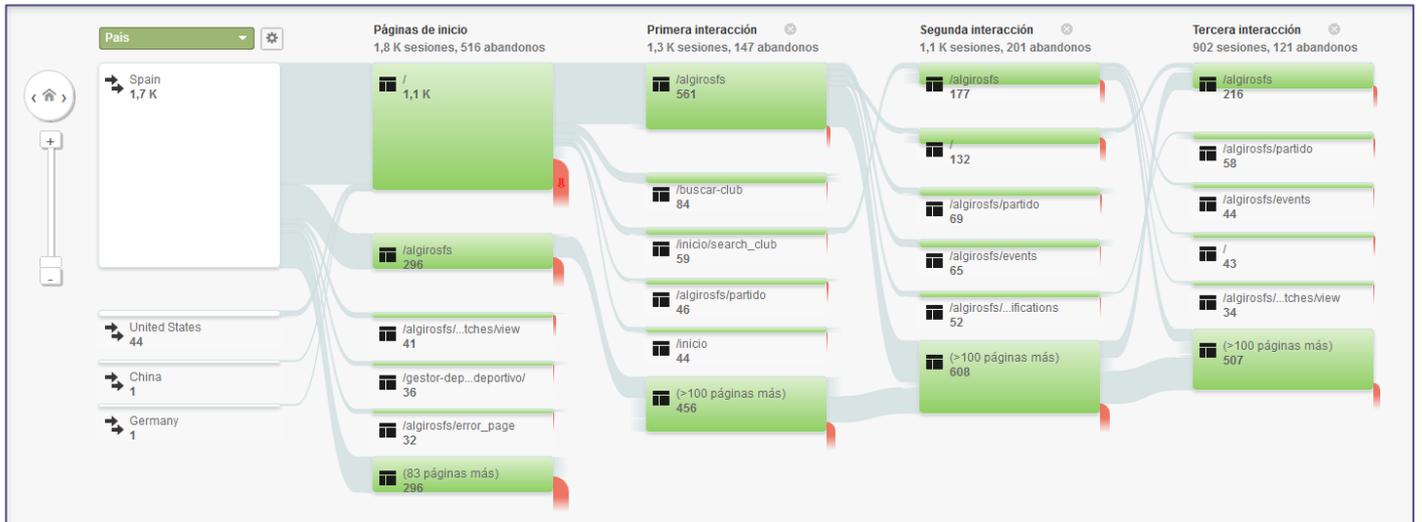


Figura 50. Vista global de las interacciones realizadas durante el mes de mayo.

7. Conclusiones

La creación de Yulava surge como motivación personal y profesional de poner a disposición del público una herramienta capaz de gestionar clubs deportivos, facilitando así su administración y fomentando la participación de todos sus miembros, poniendo a su disposición información relevante del club.

Un aspecto identificativo es la utilidad de la aplicación, ya que ésta es una herramienta de la cual pueden beneficiarse todo tipo de clubs deportivos; independientemente de la categoría. Es importante mencionar que cada club dispone de una parte pública y una parte privada, en la cual solo tienen acceso los usuarios registrados en dicho club.

Para que un club forme parte de Yulava tan solo es necesario rellenar un formulario, evitando de este modo la instalación en un servidor diferente y ahorrando en infraestructuras y coste.

En cuanto al diseño, la prioridad ha sido realizar un diseño dinámico y adaptado a todo tipo de usuarios, teniendo en cuenta los diferentes tipos de miembros que se pueden encontrar. Cada usuario se le otorga un rol en función del papel que desarrolla dentro del club. Este hecho, favorece la comunicación entre todos los miembros.

Tras su puesta en marcha, se ha observado el aumento progresivo en la utilización de la aplicación por parte de los miembros pertenecientes a un club de Fútbol Sala. Este hecho radica en la sencillez de la aplicación y la comodidad en su uso y acceso, ya que desde cualquier dispositivo con conexión a internet se puede acceder fácilmente a la aplicación y consultar toda la información.

Se puede afirmar, por tanto, que la utilización de Yulava facilita las tareas de gestión del día a día de un club y fomenta la participación de los miembros, favoreciendo la comunicación, no sólo con su equipo sino con la totalidad del club.

7.1 Valoraciones

Este ambicioso proyecto, no solo ha servido para aplicar gran parte de los conocimientos adquiridos durante el grado, sino que también, con los problemas que han ido surgiendo durante su realización, ha servido para el autoaprendizaje. Ha sido necesaria una buena documentación y asesoramiento, así como muchas horas de trabajo. Otro aspecto positivo que ha traído la realización de este proyecto, ha sido aprender a trabajar en equipo junto con mi compañero; la organización y comunicación han sido fundamentales.

Ver el resultado final es muy satisfactorio, y saber que es una aplicación útil que cada día cuenta con más miembros, hace que aumenten las ganas de seguir trabajando en ella, ampliando sus servicios y facilitando la gestión deportiva.

7.2. Trabajo futuro

La prioridad de una futura ampliación de Yulava, y después de analizar las estadísticas, en las que se observa que el uso de los teléfonos móviles predomina sobre otro tipo de dispositivos. Por este motivo la realización de aplicaciones móviles nativas para las plataformas más comunes, sería el principal objetivo de cara al futuro.

Otra nueva funcionalidad que ampliará Yulava y que le daría un enfoque más comercial, sería ofrecer a los clubs una serie de servicios web que conectara su página principal con Yulava. De esta forma, cada club podría tener su propia página web totalmente personalizada pero además, incorporaría los datos que estuvieran alojados en Yulava.

Por último, ampliar los idiomas que ofrece la aplicación, haría la zona geográfica de uso se extendiera notablemente, aumentado de esta forma, el número de clubs registrados.

8. Bibliografía

- [1] API de Google for Work. Junio 2016. <https://developers.google.com/maps/premium/>
- [2] Google Maps. Junio 2016. <https://developers.google.com/maps/?hl=es>
- [3] HTML5. Junio 2016. <https://www.w3.org/html/>
- [4] CSS3. Junio 2016. <https://www.w3.org/Style/CSS/>
- [5] JavaScript. Junio 2016. <http://www.w3schools.com/js/>
- [6] Framework BootStrap. Junio 2016. <http://getbootstrap.com/>
- [7] CodeIgniter Web Framework. Junio 2016. <https://www.codeigniter.com/>
- [8] SEO. Junio 2016. https://es.wikipedia.org/wiki/Posicionamiento_en_buscadores
- [9] Futbolvista. Junio 2016. <http://www.futbolvista.com>
- [10] Miclub-manager. Junio 2016. <http://www.miclub-manager.es>
- [11] Myequipo. Junio 2016. <http://www.myequipo.com>
- [12] Ley Orgánica 15/1999, de 13 de Diciembre.
<https://www.boe.es/buscar/doc.php?id=BOE-A-1999-2375013> Google Maps API for Work
- [14]- UML (Unified Modeling Language) Junio 2016. <http://www.uml.org/>
- [15] Form Validation Junio 2016.
http://www.codeigniter.com/user_guide/libraries/form_validation.html?highlight=form%20validation
- [16] Active Record. Junio 2016. https://ellislab.com/codeigniter/user-guide/database/active_record.html
- [17] Sesión. Junio 2016.
http://www.codeigniter.com/user_guide/libraries/sessions.html?highlight=session
- [18] PHP. Junio 2016. <https://secure.php.net/manual/es/index.php>
- [19] MySQL language. Junio 2016. <http://dev.mysql.com/doc/refman/5.7/en/language-structure.html>
- [20] JSON. Junio 2016. <http://www.json.org/>
- [21] Eclipse for PHP Developers -Version: Mars Release (4.5.0). Junio 2016
<http://www.eclipse.org/downloads/packages/release/mars/r>
- [22] Sybase PowerDesigner. Junio 2016 . <http://powerdesigner.de/en/>
- [23] phpMyAdmin. Junio 2016. <https://www.phpmyadmin.net/>

- [24] Control de versiones. Junio 2016. https://es.wikipedia.org/wiki/Control_de_versiones
- [25] GIT. Junio 2016 <https://git-scm.com/>
- [26] BitBucket. Junio 2016 <https://bitbucket.org/>
- [27] Apache. Junio 2016. <http://www.apache.org/>
- [28] Pruebas funcionales. Junio 2016. https://es.wikipedia.org/wiki/Pruebas_funcionales
- [29] Inyección SQL. Junio 2016. https://es.wikipedia.org/wiki/Inyecci%C3%B3n_SQL

