



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 (Front-End)

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Daroqui Chafé, Julián Angel

Tutor: Buendía García, Félix

2015/2016

Resumen

En el presente documento se plasma la memoria del proyecto final de grado. Dicho proyecto tiene como objetivo principal ofrecer una herramienta útil para la gestión de clubes deportivos.

Para alcanzar los objetivos propuestos por el proyecto se han seguido las cinco etapas expuestas en la ingeniería del software. De este modo, hemos realizado la especificación de requisitos, el análisis, el diseño de la aplicación, la implementación y por último, una evaluación de la aplicación.

Según el resultado de la evaluación deberemos volver a una etapa anterior para intentar resolver el problema.

Abstract

This document reflects the final project of degree. The first objective of this project is to offer an useful tool for the management of sports clubs.

To get the proposed objectives, the five stages of software engineering have been used. In this way, we have made the specification of requirements, analysis, design of the application, implementation and finally, an evaluation of the application.

According to the results of the evaluation, if there is an error, we will have to come back to a previous stage to try to solve the problem.

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
2.3.1.1.	Requisitos hardware.....	16
2.3.1.2.	Requisitos de software	16
2.3.1.3.	Requisitos de usuario	16
2.3.2.	Requisitos funcionales.....	16
2.3.2.1.	Gestión de noticias.....	17
2.3.2.2.	Gestión de miembros	18
2.3.2.3.	Gestión de eventos.....	21
2.3.2.4.	Gestión del club.....	22
2.3.2.5.	Gestión de instalaciones	25
2.3.2.6.	Gestión de cuentas	26
2.3.2.7.	Gestión de equipos.....	27

2.3.2.8. Gestión de partidos	32
2.3.2.9. Registro e inicio de sesión.....	37
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
3.2.2.1. Gestión de noticias.....	42
3.2.2.2. Gestión de eventos.....	42
3.2.2.3. Gestión de usuarios.....	43
3.2.2.4. Editar club.....	43
3.2.2.5. Gestión de instalaciones	44
3.2.2.6. Gestión de equipos.....	44
3.2.2.7. Gestión bancaria	45
3.2.2.8. Gestión de partidos	46
3.2.3. Usuario entrenador	46
3.2.3.1. Visualizaciones.....	46
3.2.3.2. Gestión de perfil	47
3.2.3.3. Gestión de partido.....	47
3.2.3.4. Gestión de equipo	48
3.2.3. Usuario jugador	49
3.2.3.1. Visualizaciones.....	49
3.2.3.2. Gestión de perfil	50
3.2.3.3. Gestión de partido.....	50
3.2.4. Usuario padre/madre	51
3.2.4.1. Visualizaciones.....	51
3.2.4.2. Gestión de perfil	52
3.2.4.3. Gestión de partido.....	52
3.2.5. Usuario invitado	52
3.2.5.1. Visualizaciones.....	53
3.2.5.2. Gestión de perfil	53
3.2.5.3. Gestión de partido.....	54
3.2.6. Usuario no registrado	54



3.2.6.1. Visualizaciones.....	54
3.2.6.2. Acciones	55
3.3. Diagrama de clases	56
3.4. Diagramas de flujo para la gestión de equipos.....	57
3.5. Diagramas de flujo para la gestión de la web.....	58
4. Diseño	59
4.1. Introducción	59
4.2. Arquitectura del sistema.....	59
4.3. Modelo.....	60
4.4. Vista	60
4.5. Controlador	61
5. Implementación.....	63
5.1 Introducción	63
5.2 Tecnologías	63
5.2.1 CodeIgniter.....	63
5.2.1.1 Language	63
5.2.2 PHP5	64
5.2.3. MYSQL.....	65
5.2.4. HTML.....	65
5.2.5. CSS.....	65
5.2.6. BOOTSTRAP.....	65
5.2.7. AJAX.....	66
5.2.8. JAVASCRIPT	66
5.2.9. PLUGINS Y LIBRERÍAS	66
5.3. Herramientas y entorno	67
5.3.1. Entorno de desarrollo	67
5.3.1.1 Eclipse	67
5.3.2. Control de versiones	68
5.3.2.1 GIT	68
5.3.2.2 Bitbucket.....	68
5.3.3. Servidor web	68
5.3.4 Base de datos.....	68
5.4. Descripción de la implementación	68
5.4.1. Estructura de directorios	68
5.5. Interfaces de usuario	69
6. Evaluación	76

6.1	Introducción	76
6.2	Validación de estándares	76
6.2.1	Validación HTML.....	76
6.2.2	Validación CSS.....	79
6.2.3	Validación de enlaces rotos	79
6.3	Análisis SEO	80
6.4	Análisis de rendimiento.....	81
7.	Conclusiones	83
7.1	Resumen	83
7.2	Valoración.....	83
7.3	Trabajo futuro.....	83
8.	Referencias bibliográficas	85



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. También existen aplicaciones web enfocadas a la gestión de equipos, como puede ser el caso de myequipo. 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.

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

- 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.

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. Dependiendo de los permisos que le otorgue el administrador, podrá realizar unas acciones u otras, tal y como especificamos en el apartado [2.2.3](#)

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.

Este trabajo se centra en la parte del Front-End. Al ser la parte visible de la web, he tenido muy en cuenta el apartado de usabilidad del usuario, diseñando una interfaz fácil e intuitiva.

Además he prestado atención a todos los estándares web para obtener una implementación que se ajuste a ellos y así ayudar a los navegadores a renderizar la página web correctamente. Este hecho también ayuda al SEO de la página.

Por otra parte he tenido que sincronizarme con el trabajo de Back-End para enlazar correctamente la parte de lógica de la aplicación con la interfaz.

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 ^[0], se define como un conjunto de subrutinas, funciones y procedimientos.

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

- Google Maps^[5]: Aplicación web desarrollada por la compañía estadounidense Google, y que ofrece la posibilidad de desplazarte por los mapas de todo el mundo.
- HTML: (Lenguaje de marcas de hipertexto), es un lenguaje de marcado para la elaboración de páginas web.
- CSS: Es un lenguaje usado para definir y crear la presentación de un documento escrito en el lenguaje HTML
- JavaScript: 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: 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^[8]: Es un Framework utilizado para la creación de aplicaciones web.
- SEO: 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^[1]. 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^[2], 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^[3]. 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^[4], 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^[5], 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^[6].

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 tablets. 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	<p>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.</p> <p>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.</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 noticia
Introducción	El usuario puede editar noticias ya creadas en el club.
Entradas	Identificador de la noticia.
Proceso	<p>El usuario se dirige al apartado de administración de noticias y selecciona la noticia que desea editar.</p> <p>El usuario puede modificar el título, la privacidad o el contenido de la noticia.</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 noticia 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.

Desarrollo de una aplicación web para la gestión de clubs deportivos y red social para sus miembros (Front-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 (Front-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 (Front-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 (Front-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 (Front-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 (Front-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 (Front-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 (Front-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 (Front-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 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> <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 (Front-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 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> <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.

2.3.2.9. Registro e inicio de sesión

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 4GB 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 ^[7].

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.

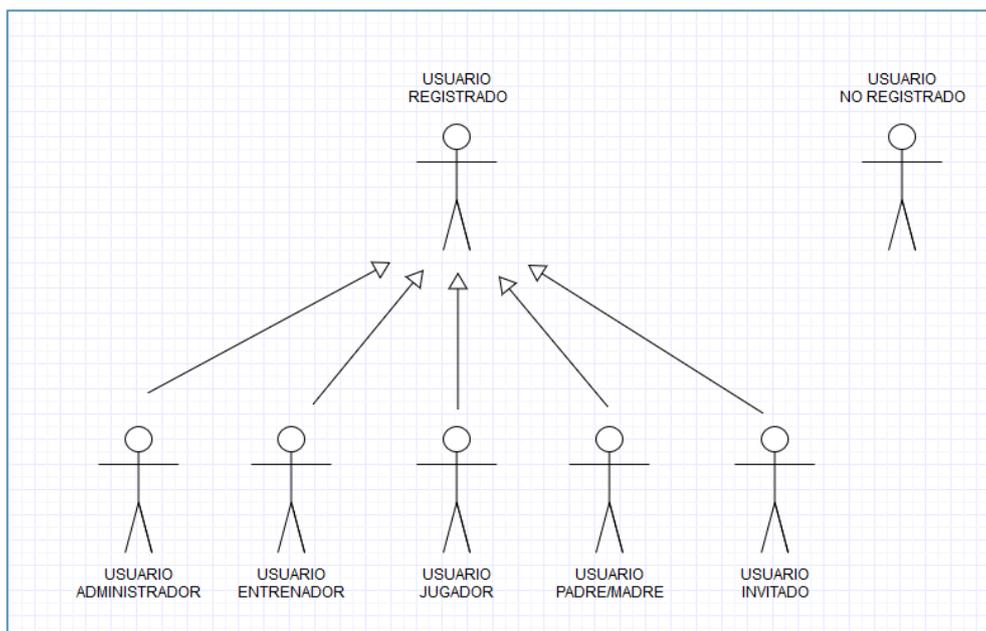


Figura 1. Actores de la aplicación

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

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.

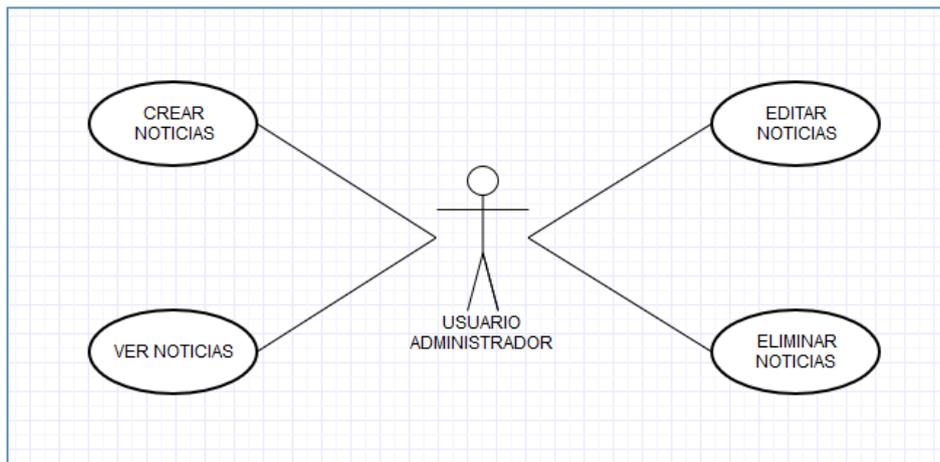


Figura 2. Gestión de noticias del usuario administrador.

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. Figura 3.

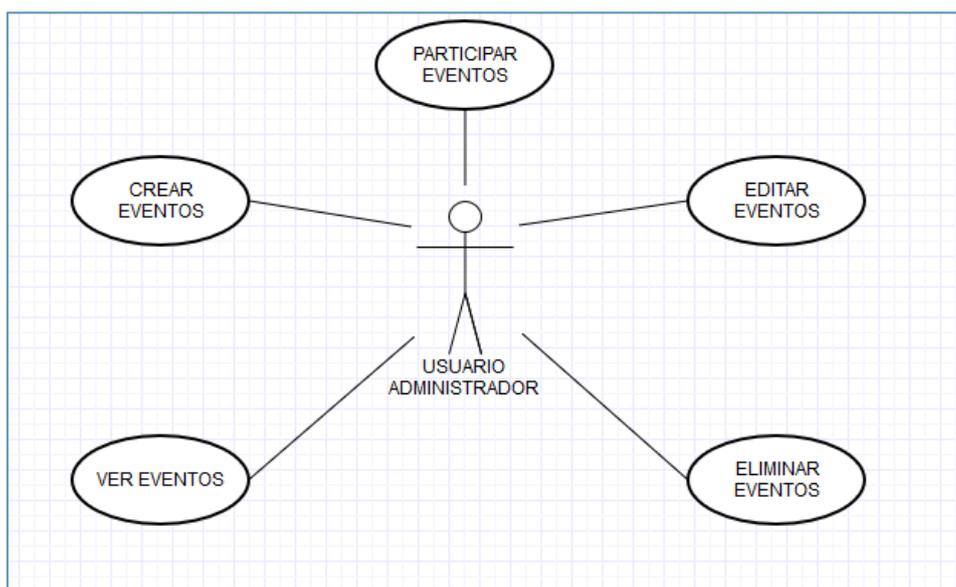


Figura 3. Gestión de eventos del usuario administrador.

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. Figura 4.

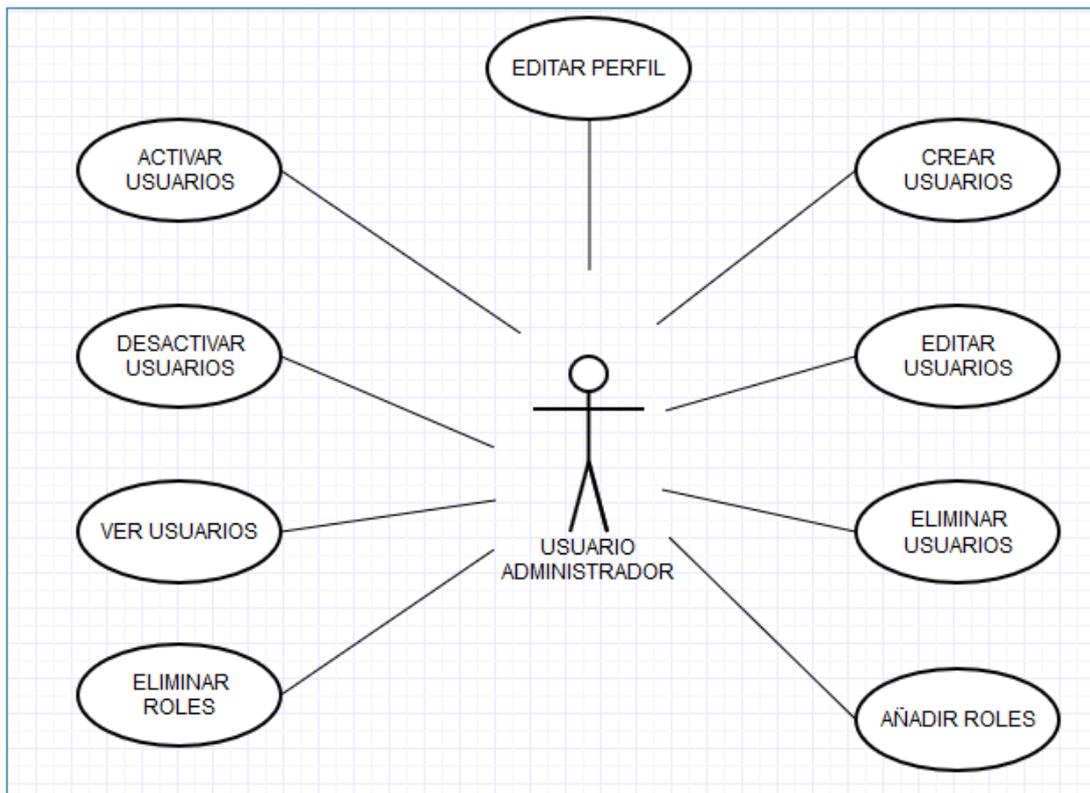


Figura 4. Gestión de usuarios del usuario administrador.

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. Figura 5.

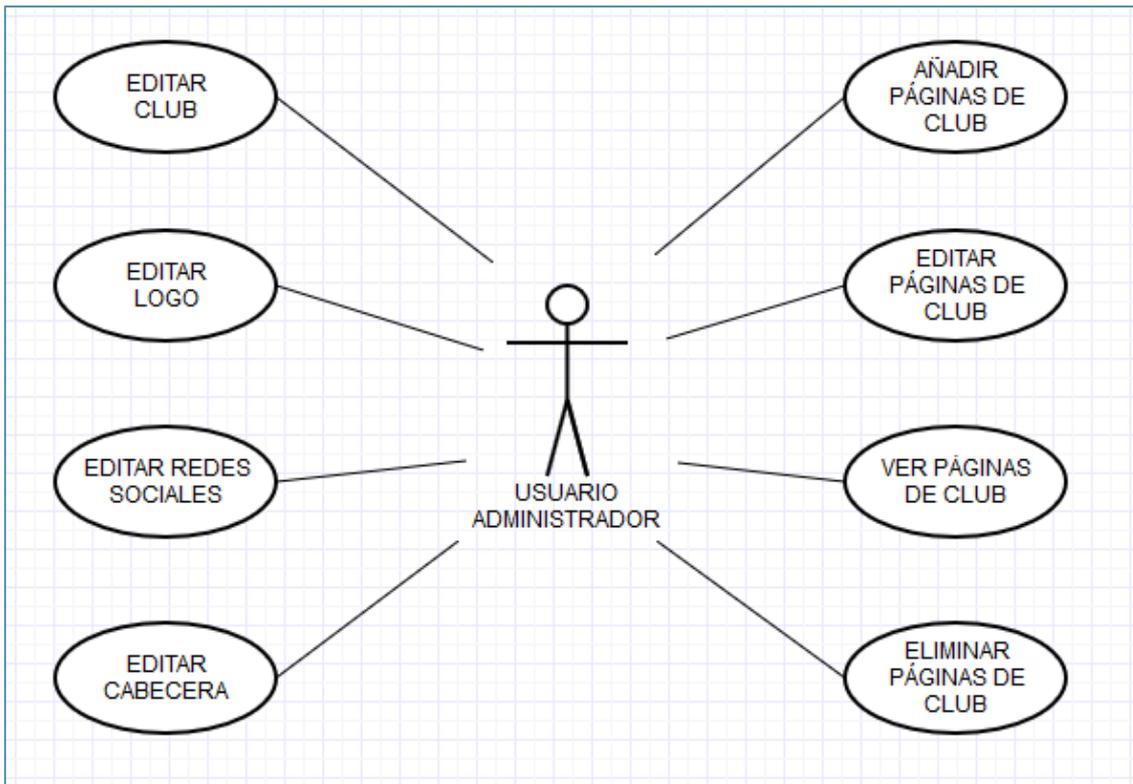


Figura 5. Editar club.

3.2.2.5. Gestión de instalaciones

El usuario administrador puede visualizar, crear, editar y eliminar las instalaciones. Dichas instalaciones pueden ser propias del club, o de los rivales. Figura 6.

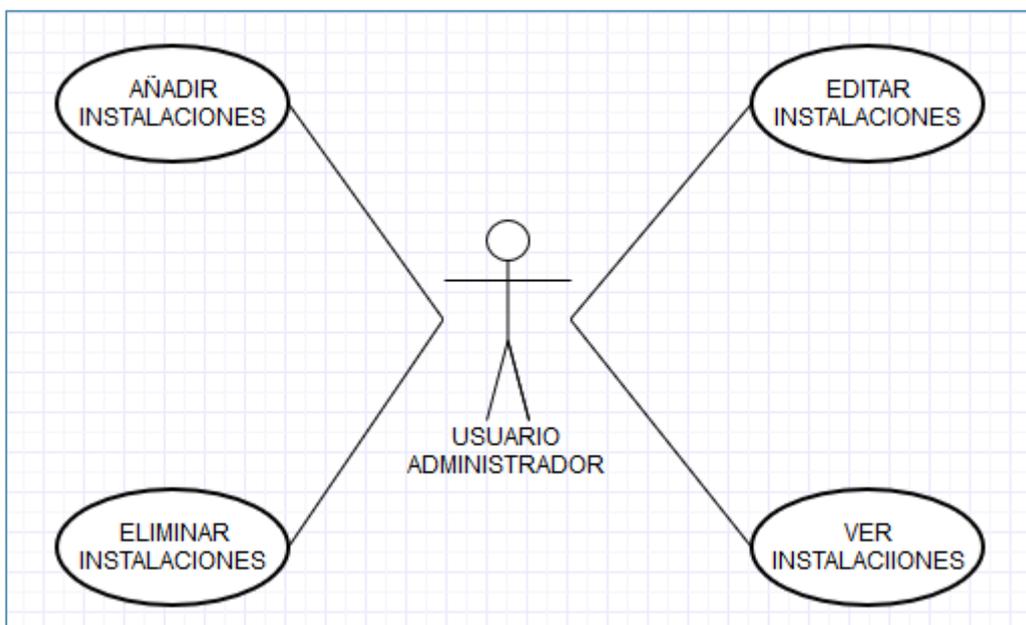


Figura 6. Gestión de instalaciones del usuario administrador.

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. Figura 7.

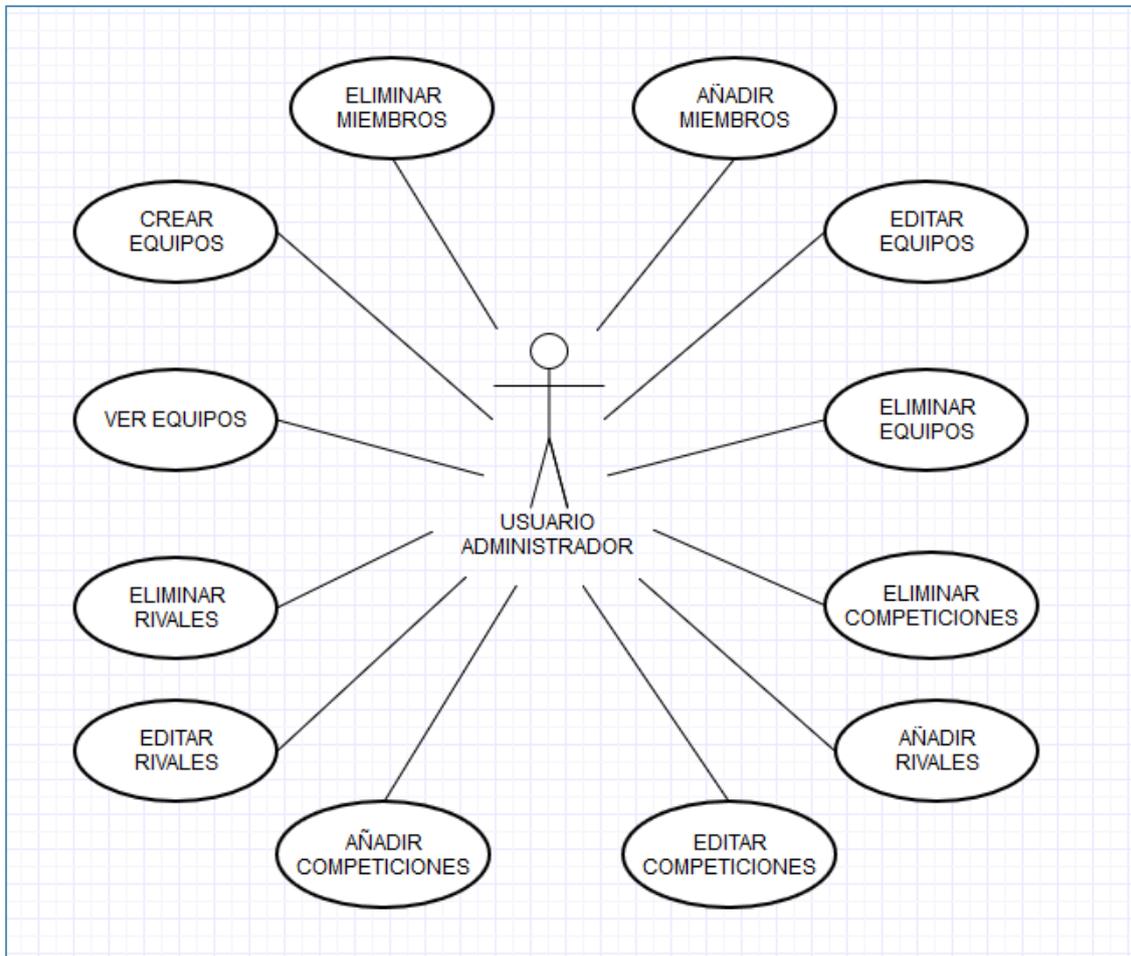


Figura 7. Gestión de equipos del usuario administrador.

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. Figura 8.

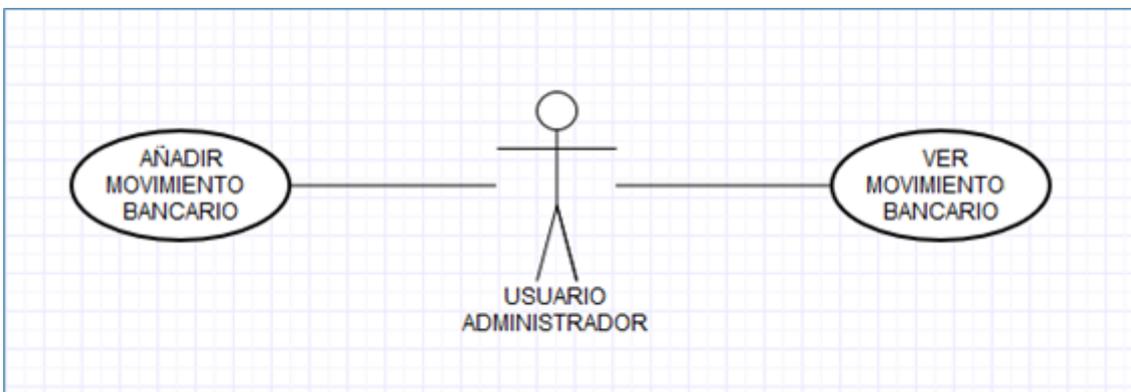


Figura 8. Gestión bancaria del usuario administrador.

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. Figura 9.

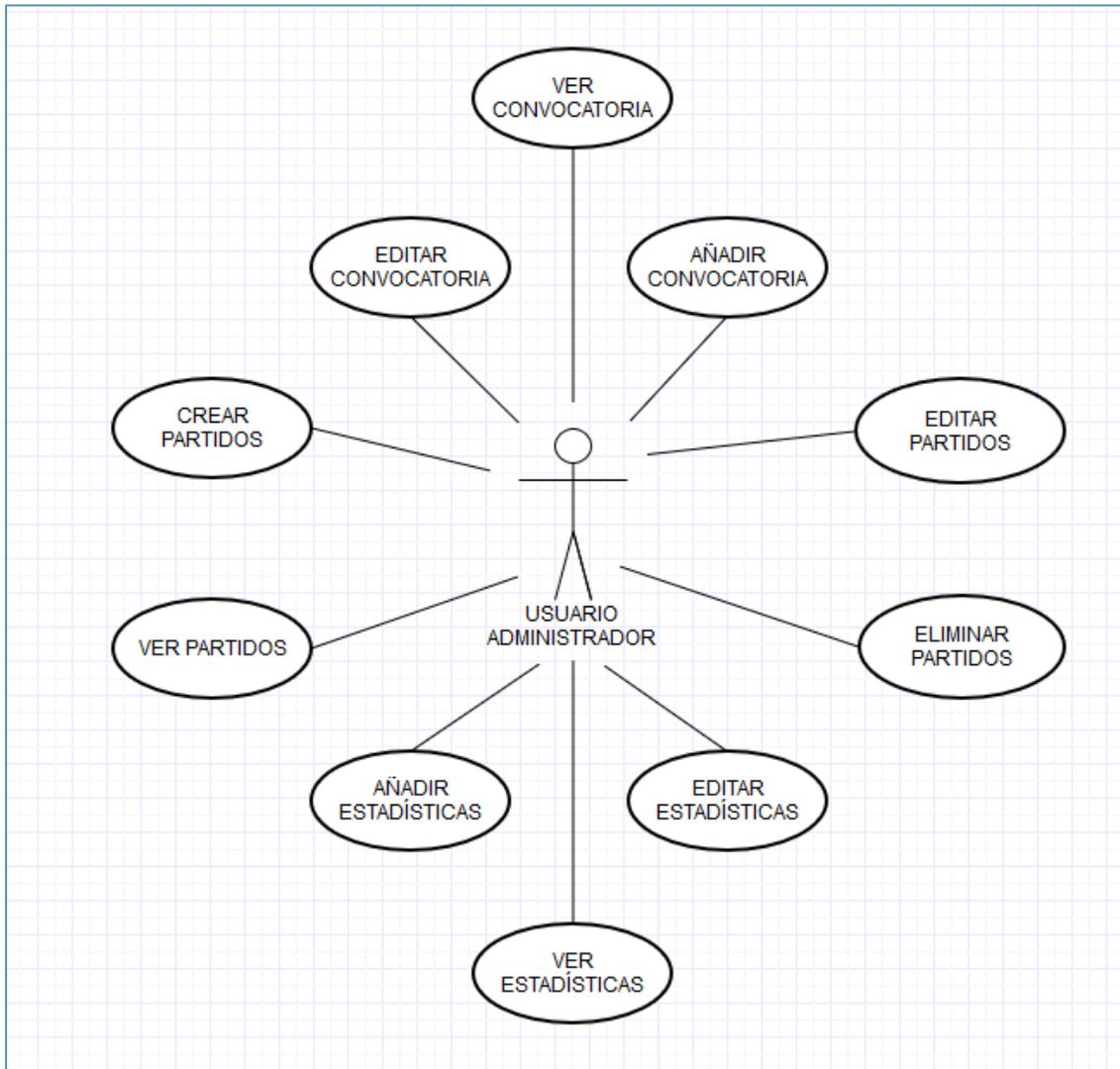


Figura 9. Gestión de partidos del usuario administrador.

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. Figura 10.

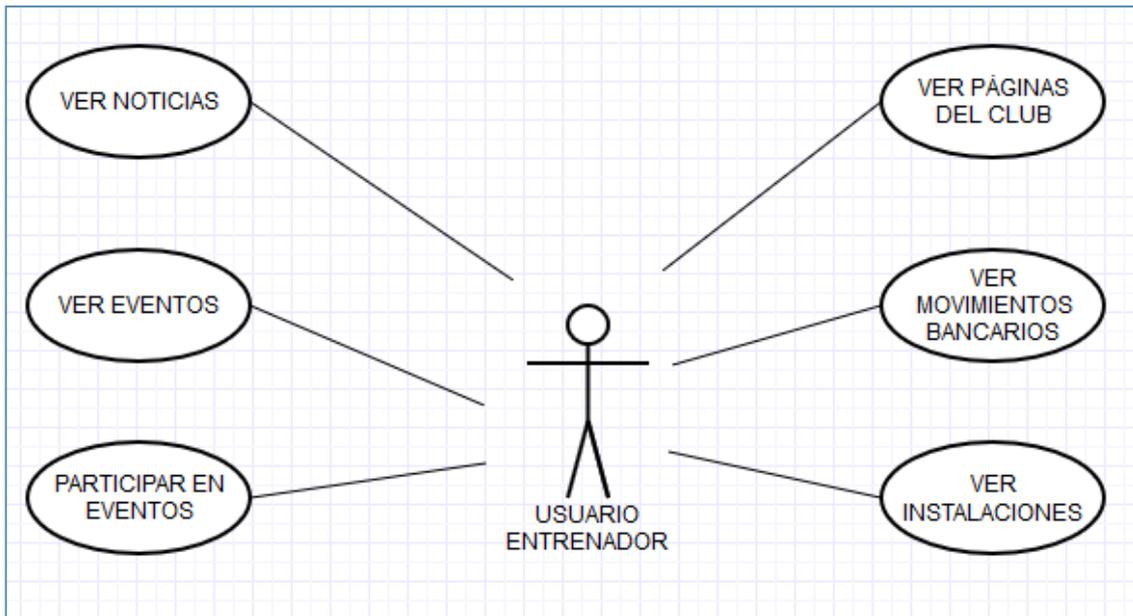


Figura 10. Visualizaciones disponibles para el usuario entrenador.

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. Figura 11.

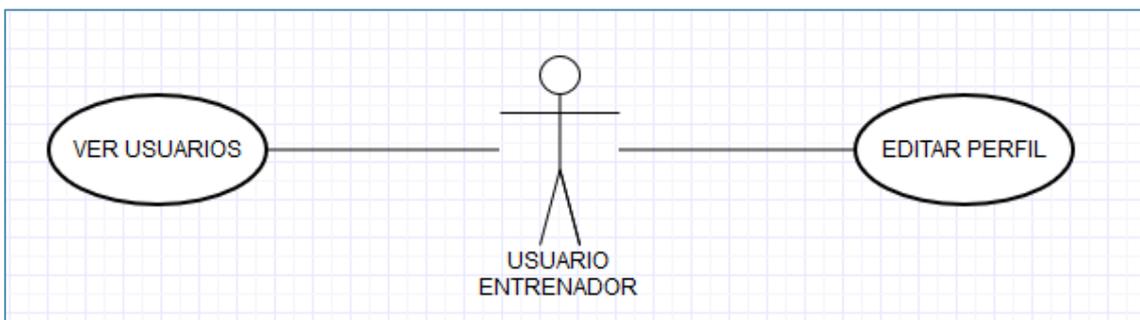


Figura 11. Gestión de perfil para el usuario entrenador.

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. Figura 12.

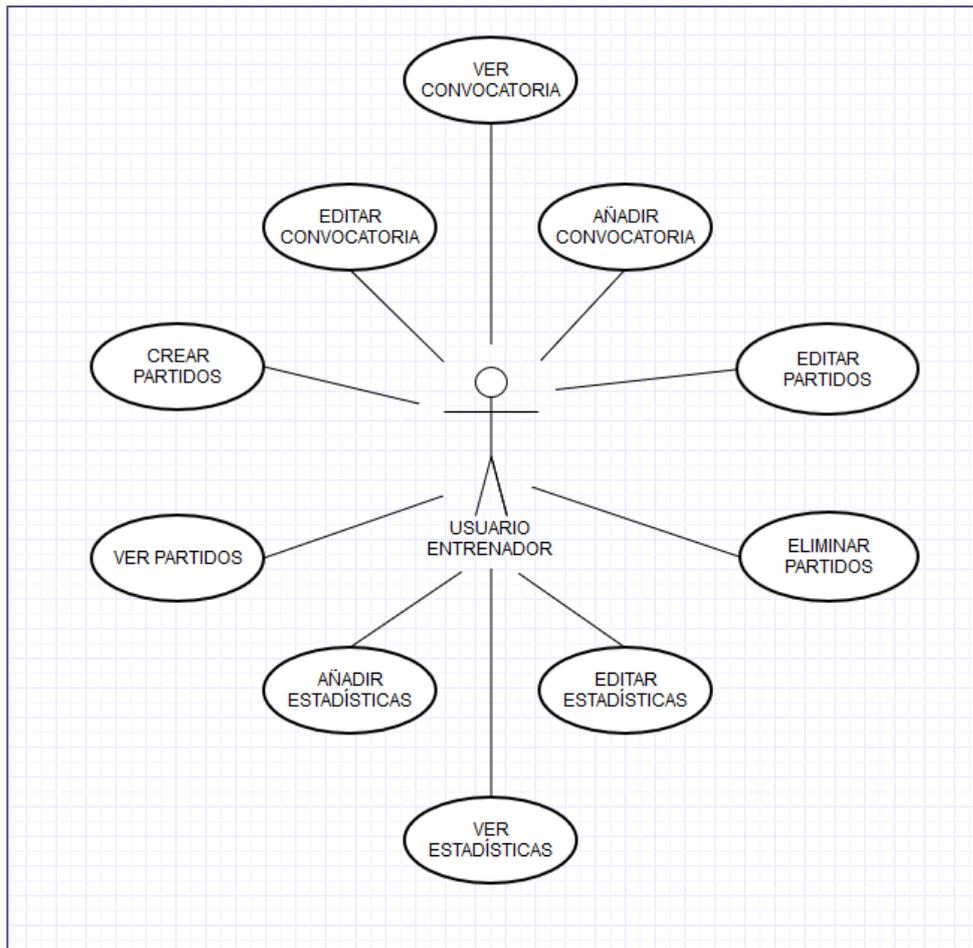


Figura 12. Gestión de partidos para el usuario entrenador.

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. Figura 13.

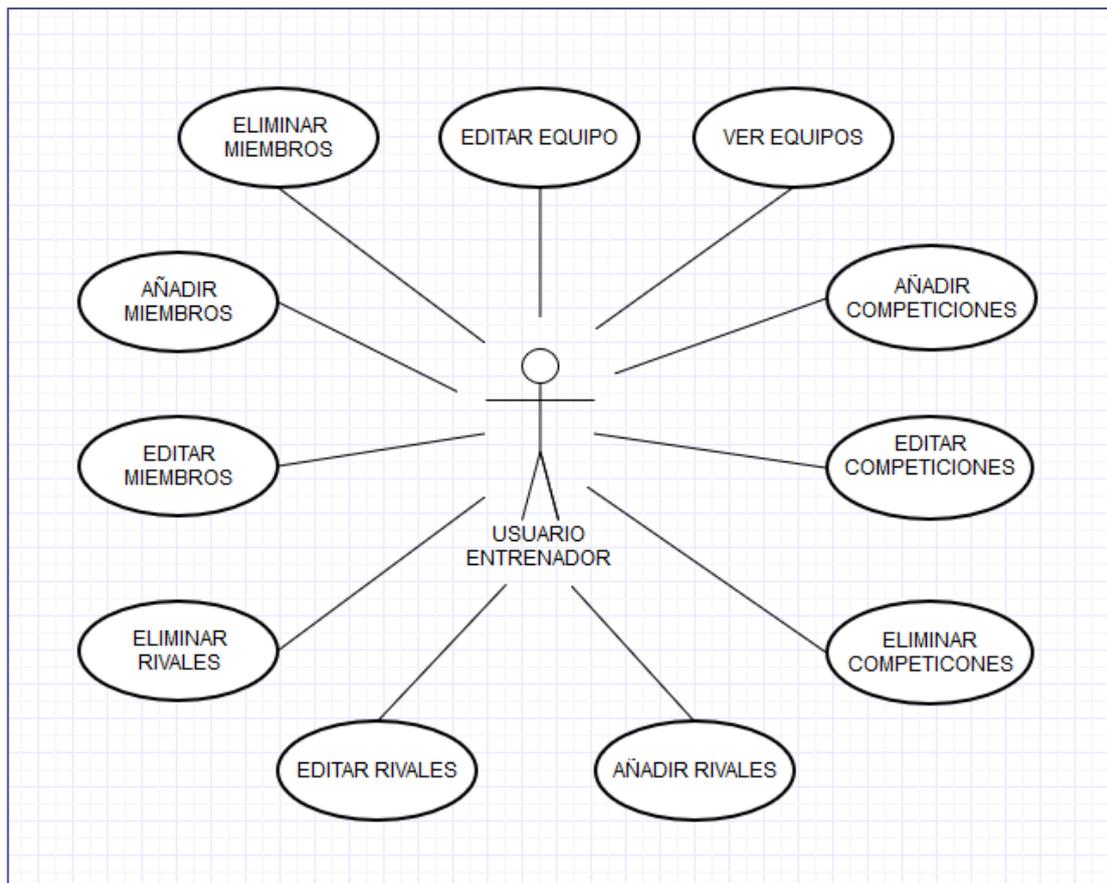


Figura 13. Gestión de equipos para el usuario entrenador.

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. Figura 14.

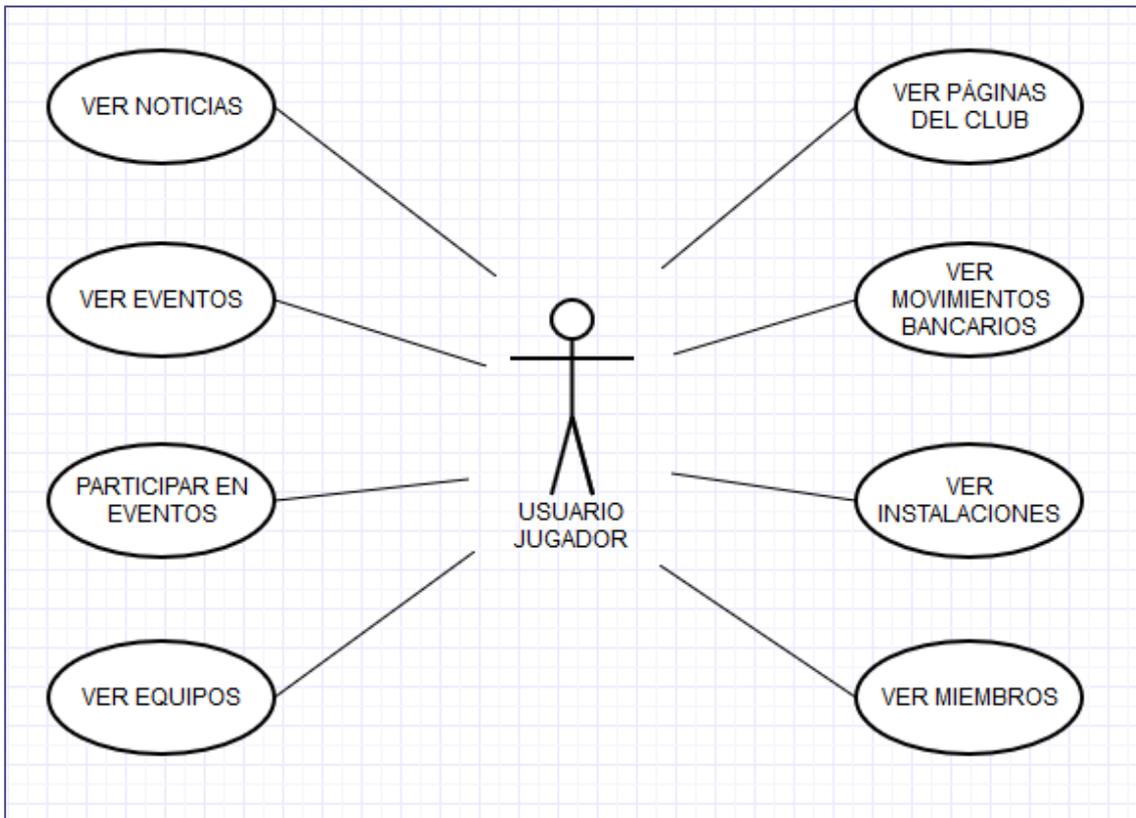


Figura 14. Visualizaciones del usuario jugador.

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. Figura 15.

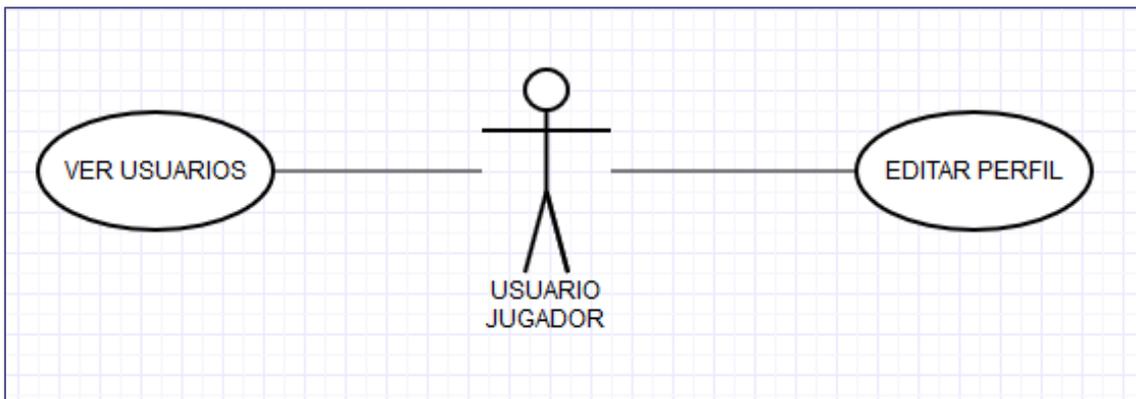


Figura 15. Gestión de perfil del usuario jugador.

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. Figura 16.

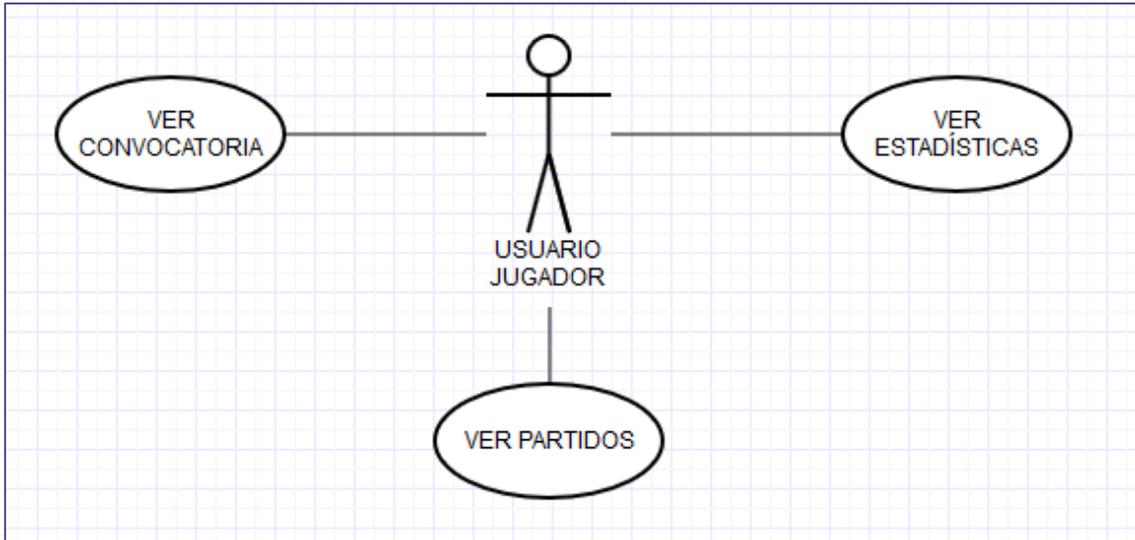


Figura 16. Gestión de partidos para el usuario jugador.

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. Figura 17.

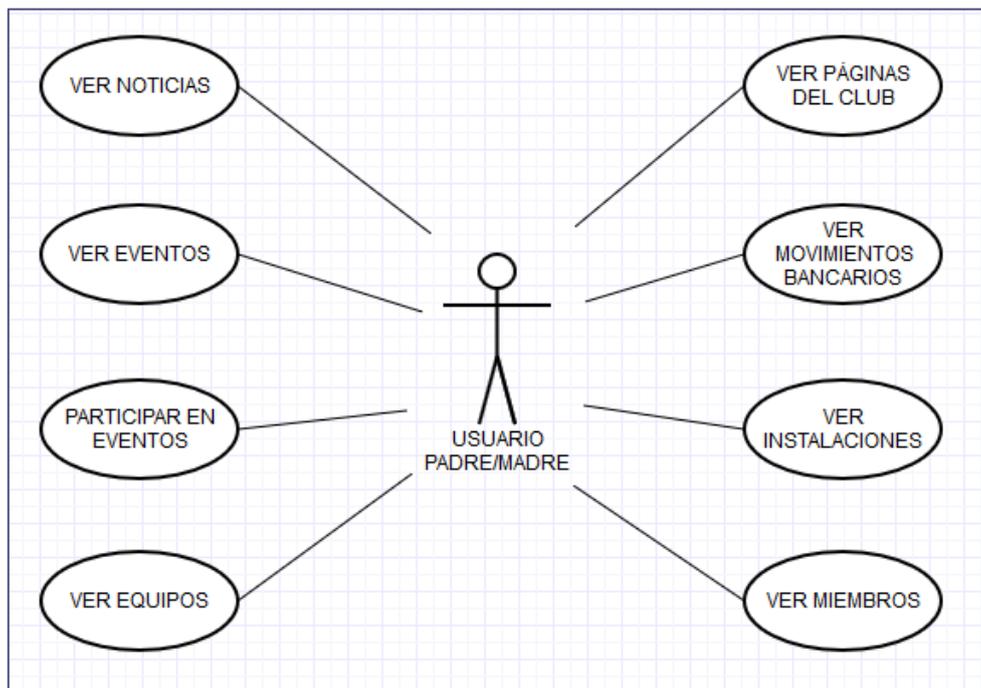


Figura 17. Visualizaciones del usuario padre/madre.

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. Figura 18.

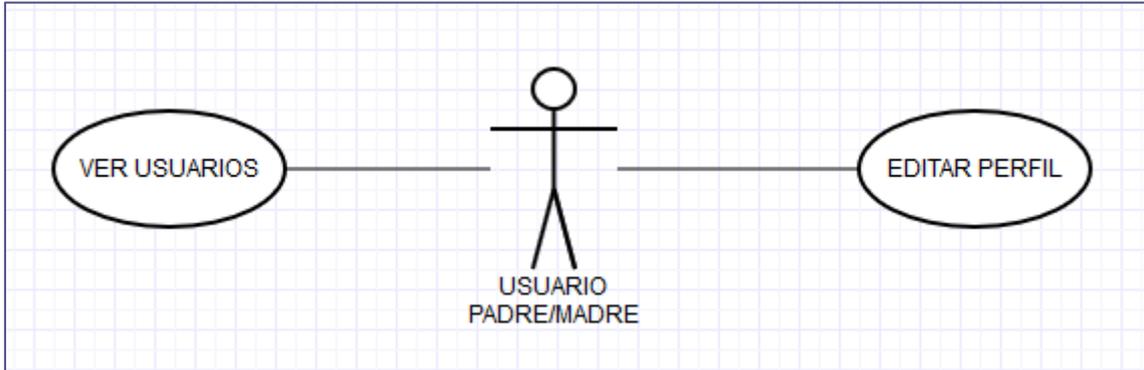


Figura 18. Gestión de perfil del usuario padre/madre.

3.2.4.3. Gestión de partido

El usuario padre/madre puede visualizar todos los partidos del club. Figura 19.

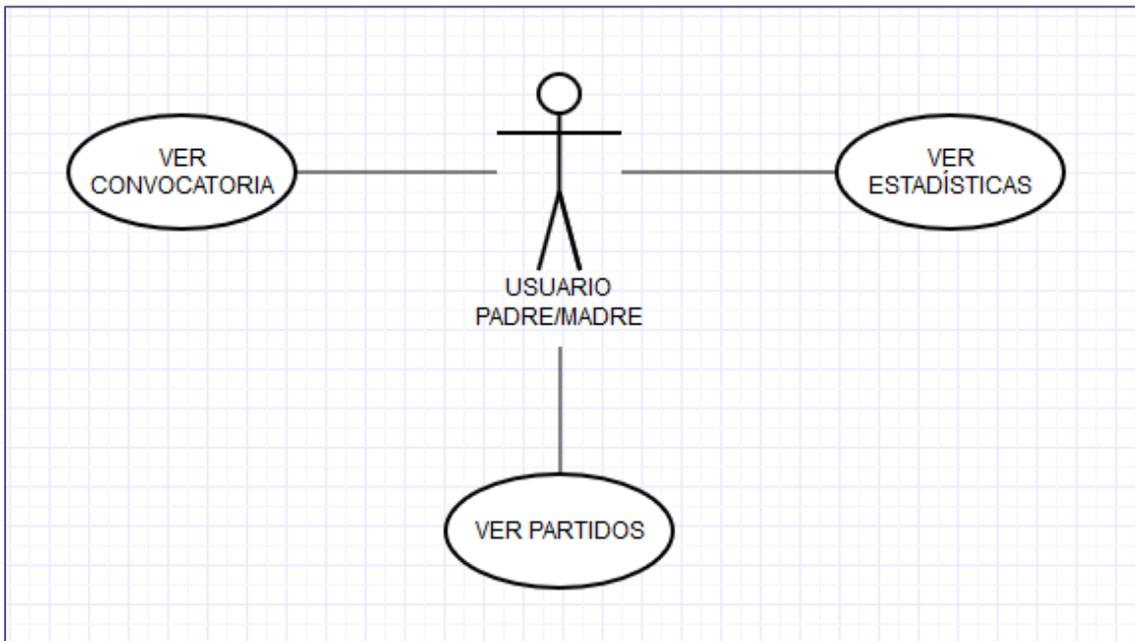


Figura 18. Gestión de partidos del usuario padre/madre.

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. Figura 19.

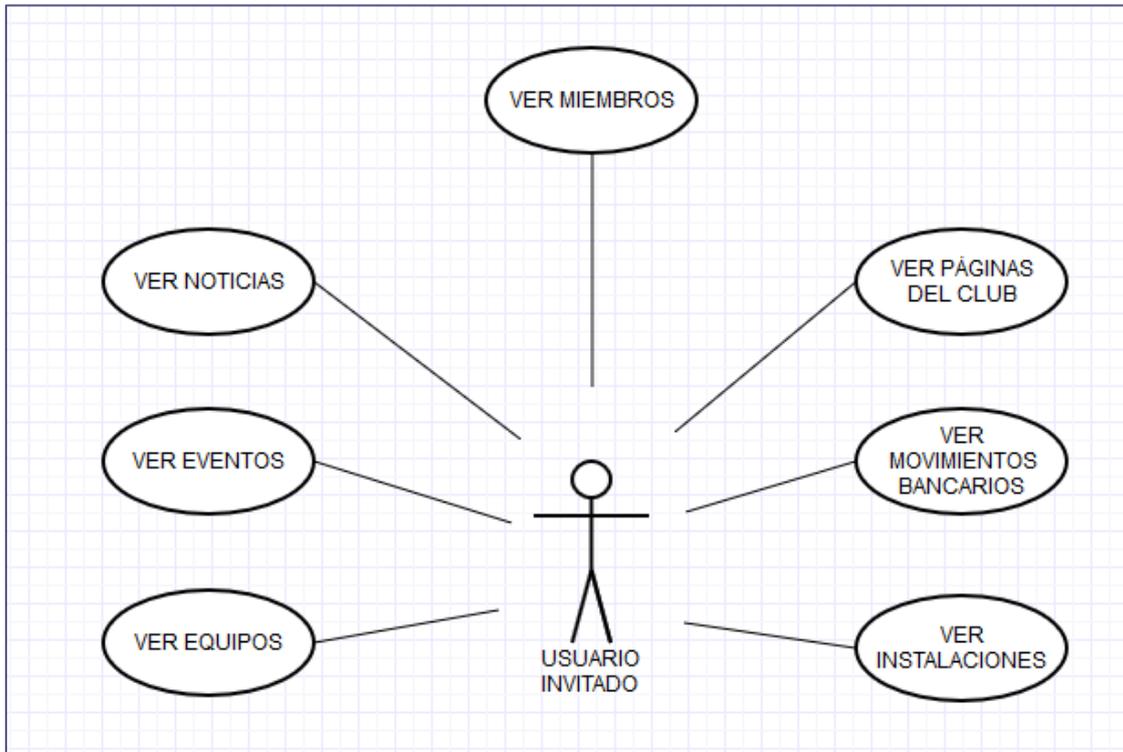


Figura 19. Visualizaciones para el usuario invitado.

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. Figura 20.

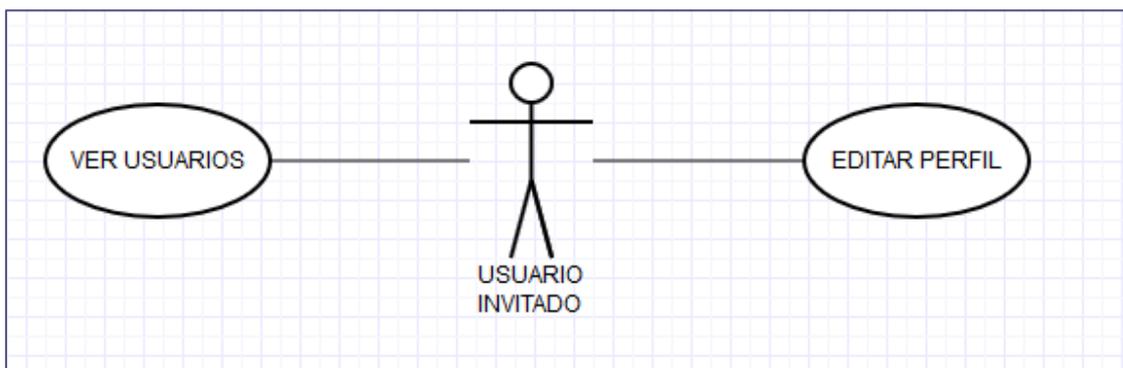


Figura 20. Gestión de perfil del usuario invitado.

3.2.5.3. Gestión de partido

El usuario invitado puede visualizar todos los partidos del club. Figura 21.

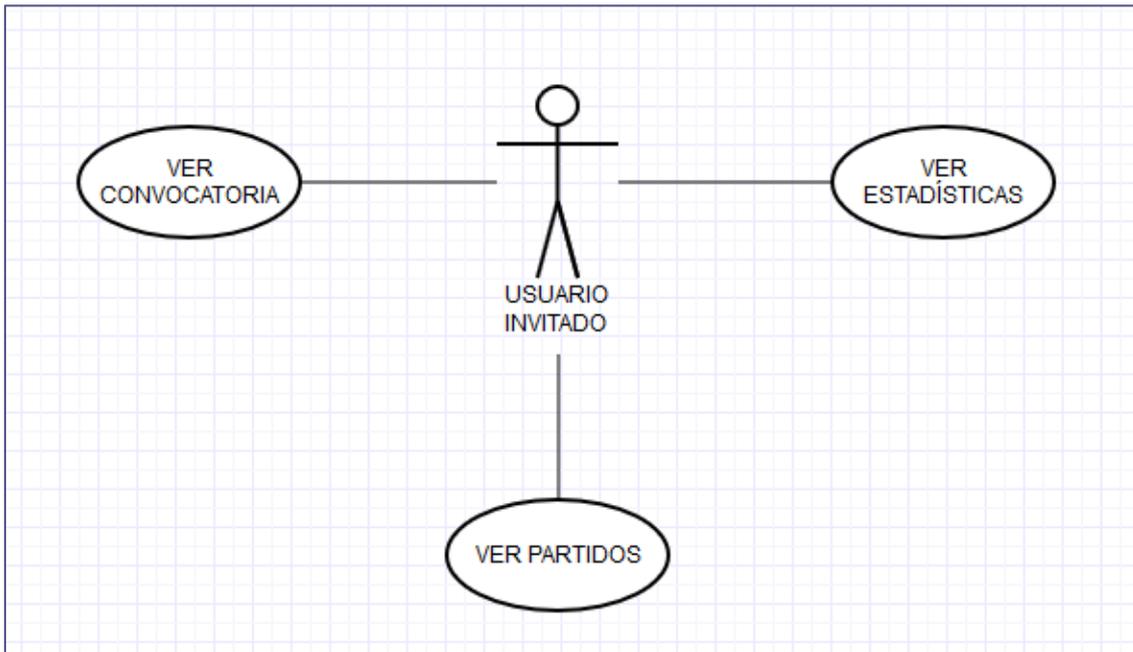


Figura 21. Gestión de partidos del usuario invitado.

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. Figura 22.



Figura 22. 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. Figura 23.



Figura 23. 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. Figura 25.

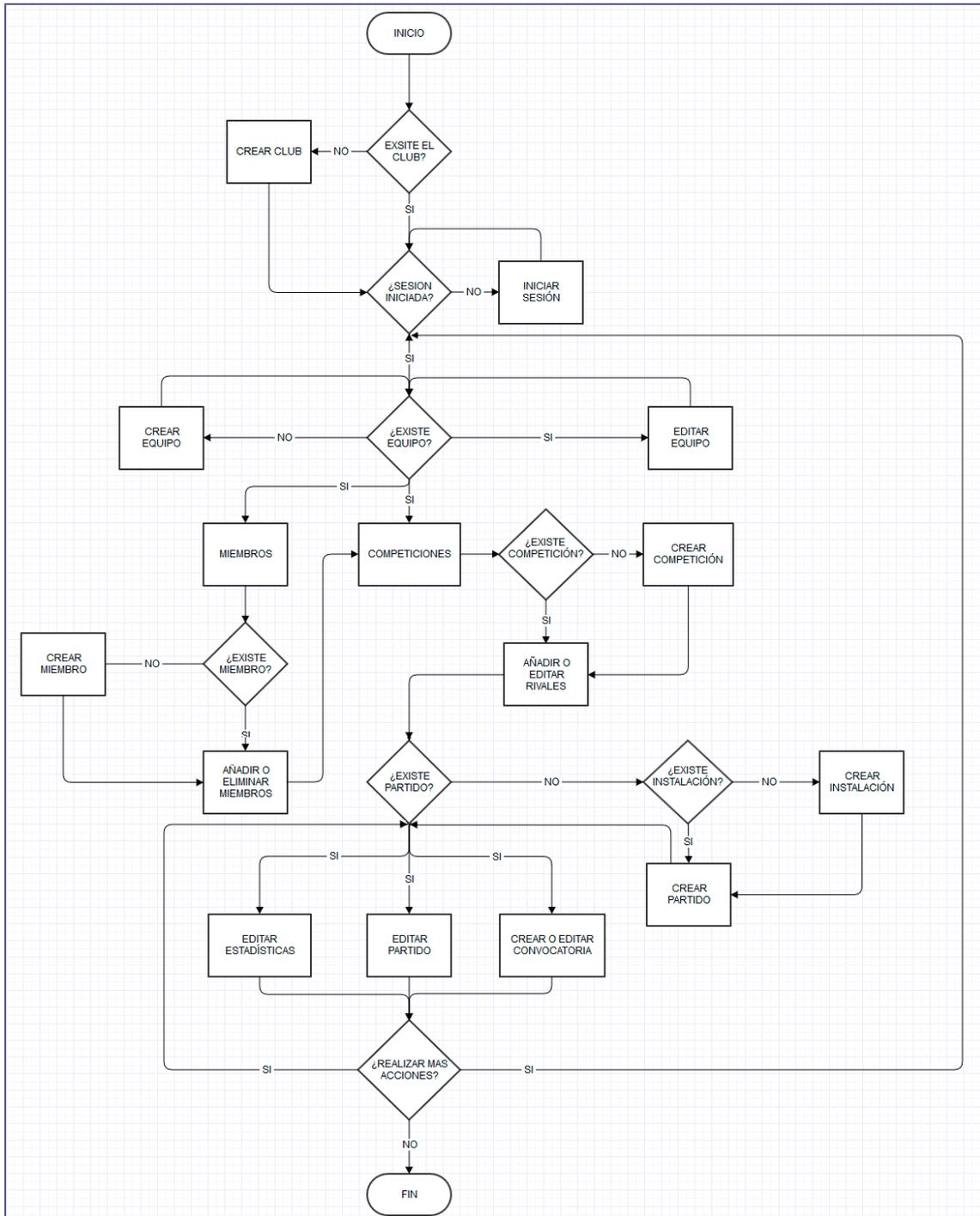


Figura 25. Diagrama de flujo para la gestión de equipos.



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. Figura 26.

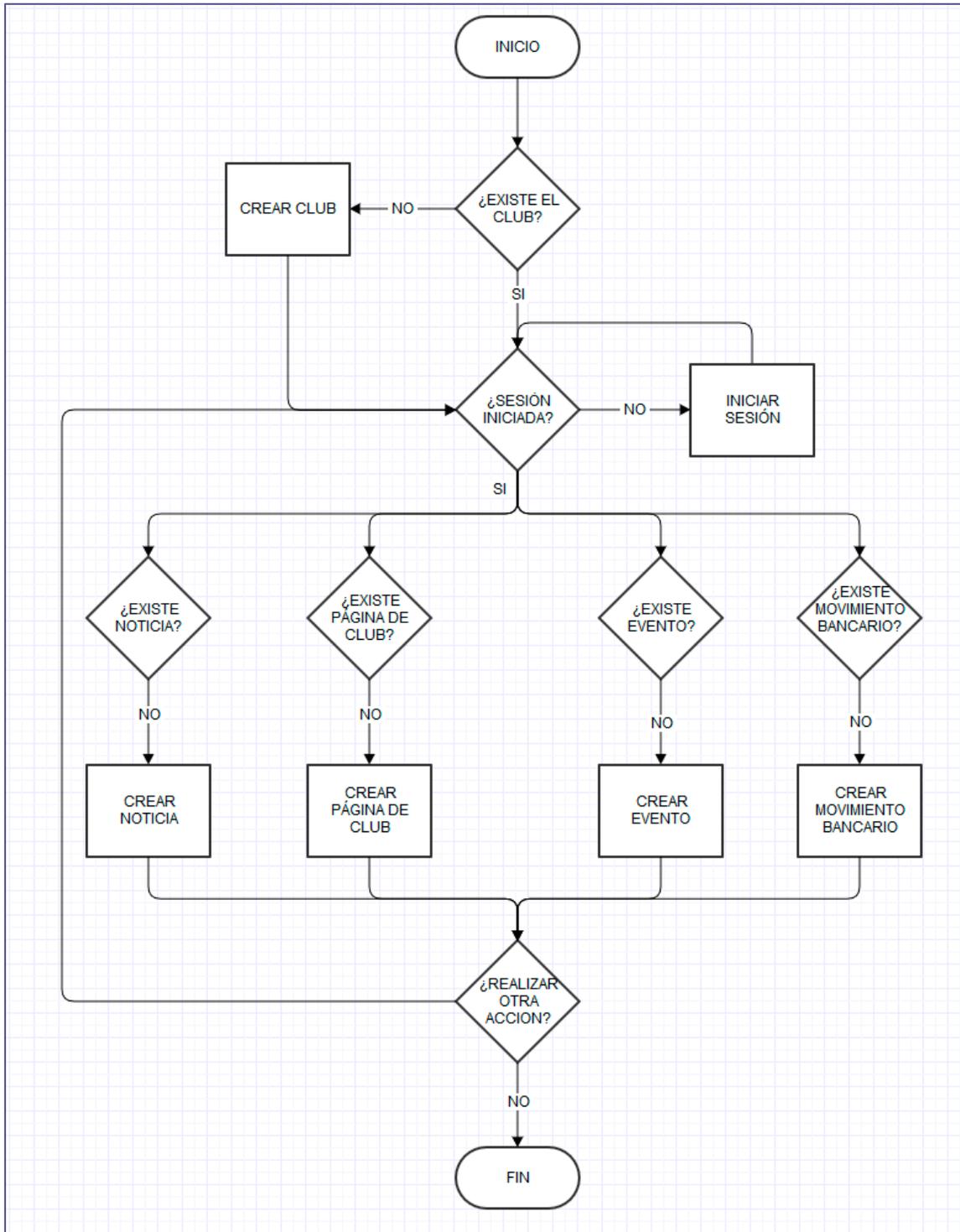


Figura 26. Diagrama de flujo para la gestión de la web.

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 el de 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 más 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 27 se muestra un ejemplo de la arquitectura MVC utilizada por CodeIgniter.

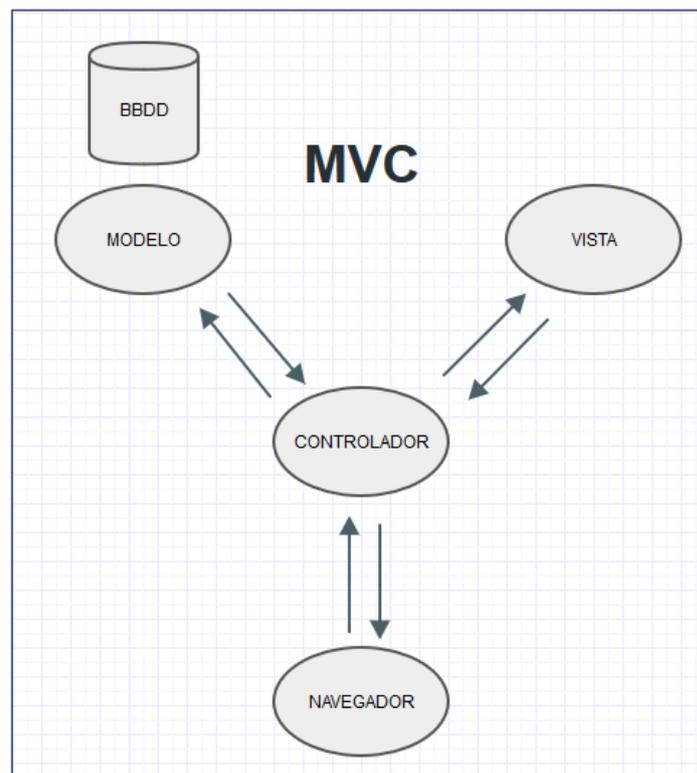


Figura 27. Arquitectura MVC.

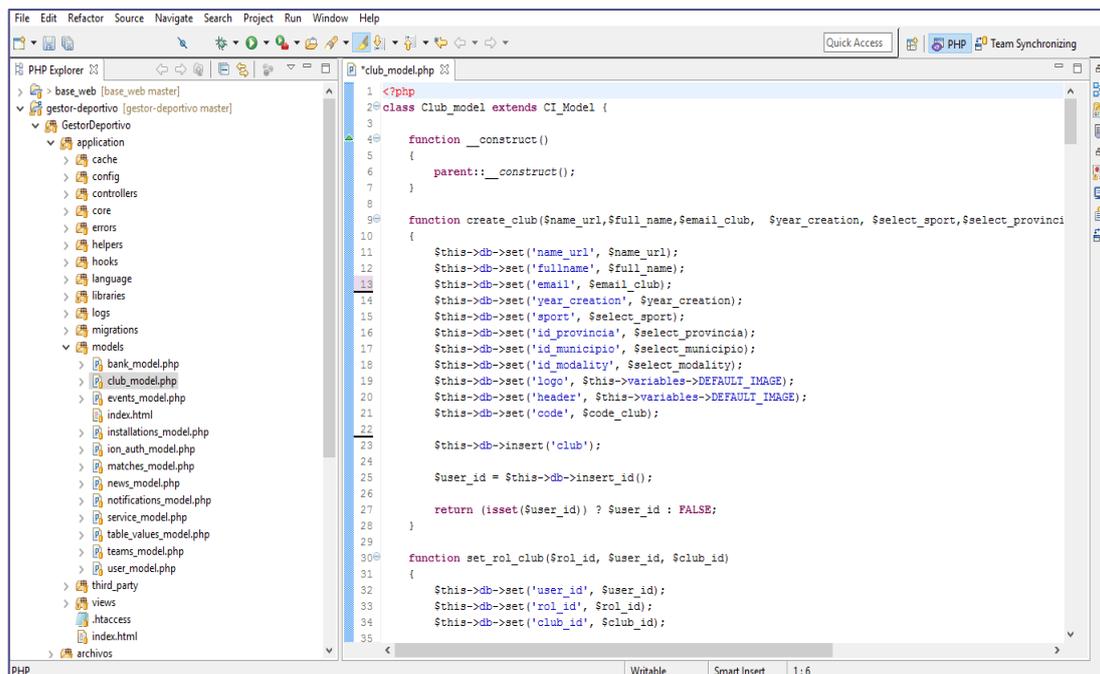
4.3. Modelo

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 28) 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 28. Ejemplo de modelo.

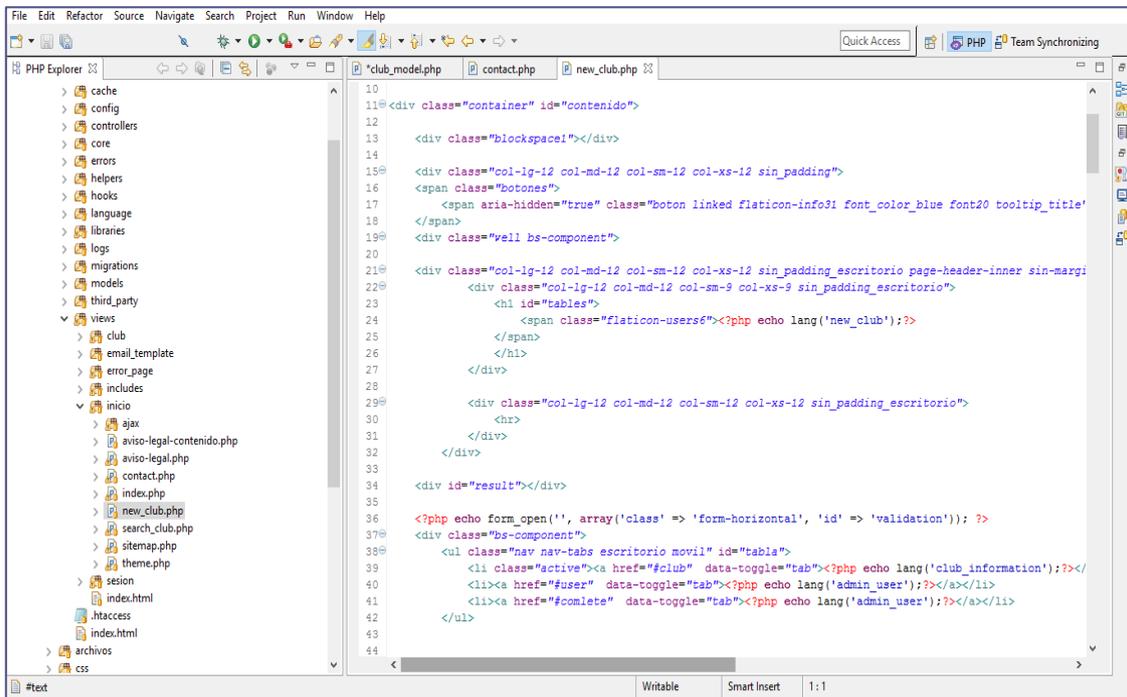
4.4. Vista

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.

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

La figura 29 muestra un ejemplo del código de la vista.



```
10
11 <div class="container" id="contenido">
12
13     <div class="blockspace1"></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_margi">
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="tabla">
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="#comlete" data-toggle="tab"><?php echo lang('admin_user');?></a></li>
42     </ul>
43
44
```

Figura 29. 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.

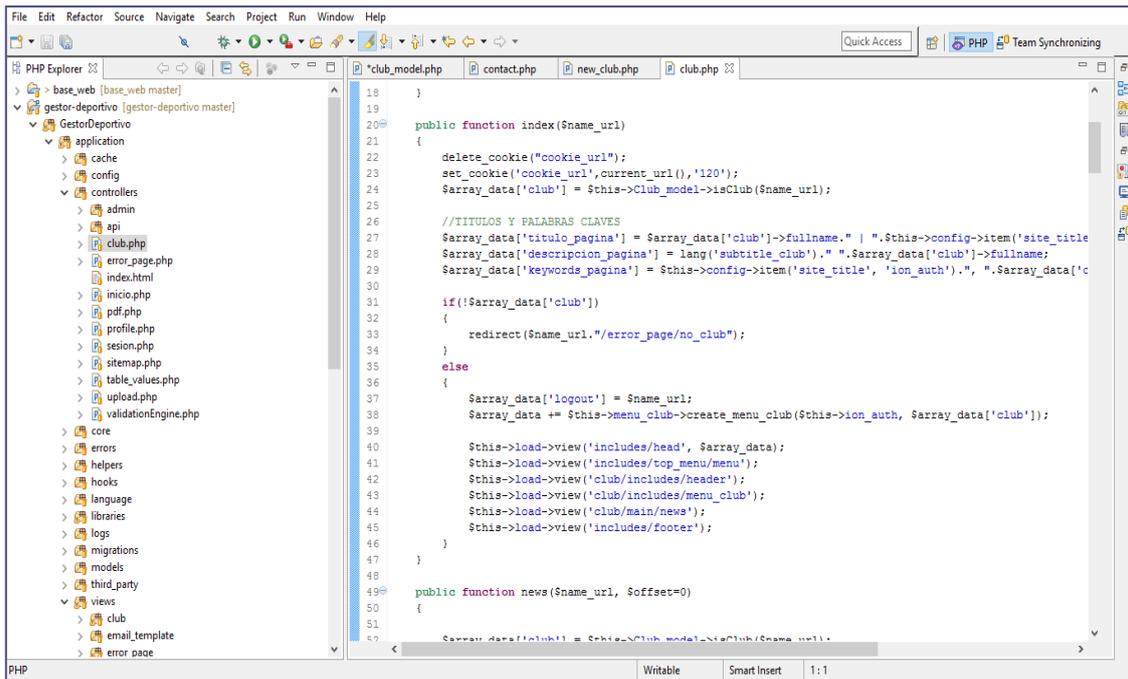
También en esta capa se comprueba que el usuario tenga los permisos necesarios para poder visualizar el contenido solicitado o incluso realiza las peticiones de datos al modelo para después transmitirlos a la vista.

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 incluimos una imagen de ejemplo de la capa de controlador (Figura 30).



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



```
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     }
53 }
```

Figura 30. 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^[8] 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 venían 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 31 se muestra un ejemplo del contenido del fichero idioma en español. Asimismo, en la figura 32 se puede observar como la capa de vista hace uso de las variables que se encuentran en el fichero anteriormente mencionado.

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

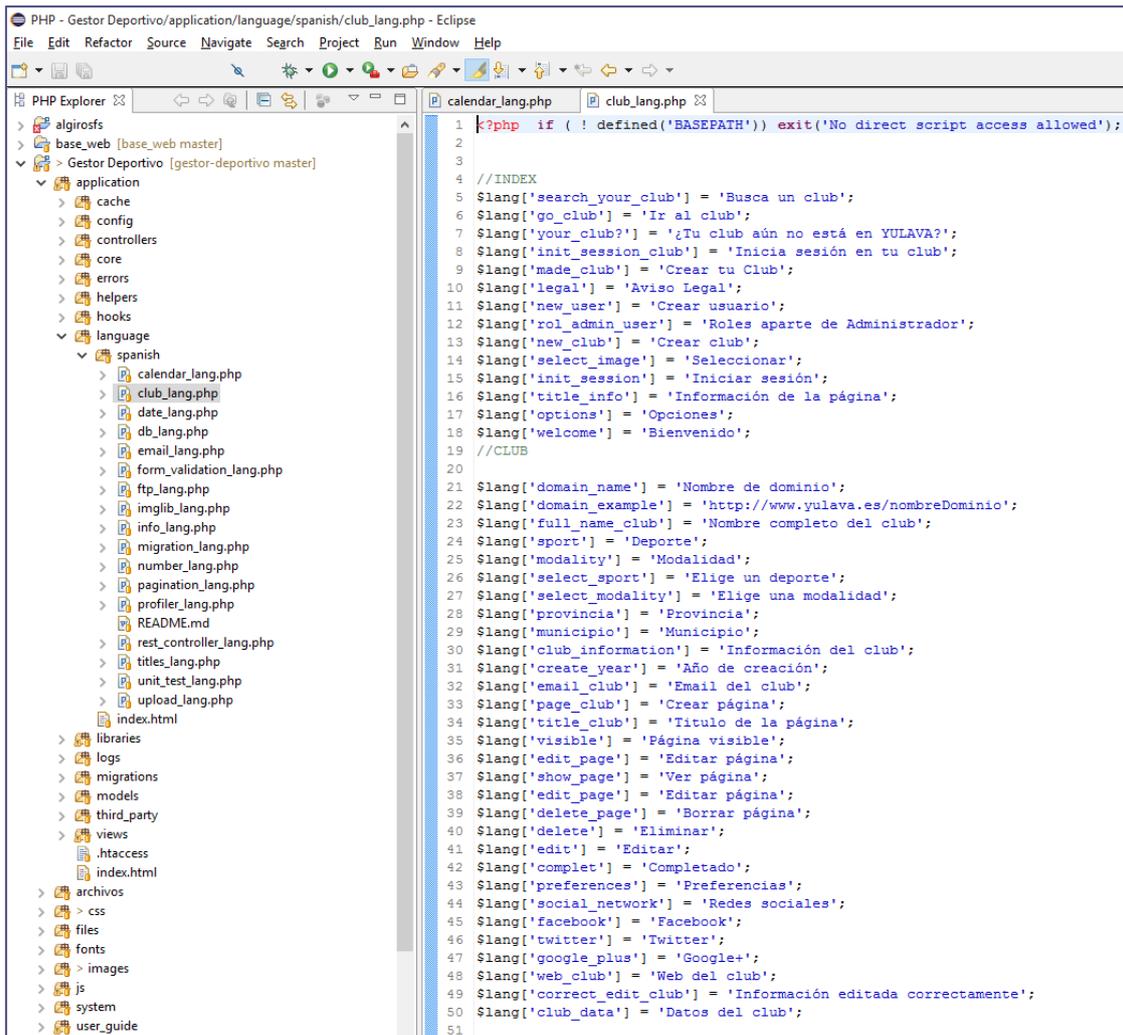


Figura 31. Ejemplo de un fichero Language.

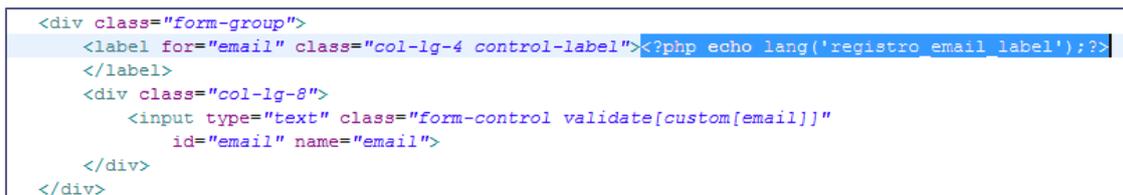


Figura 32. Ejemplo de uso de las variables de idioma.

5.2.2 PHP5

PHP^[9] 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 Back-End y en las vistas se ha utilizado para mostrar los datos dinámicos de la página.

5.2.3. MYSQL

MySQL^[10] es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado (SQL).

5.2.4. HTML

Para la implementación de las vistas de la aplicación se ha utilizado HTML^[11] en la versión 5.

Esta versión ya está soportada en todos los navegadores actuales y presenta unos avances muy importantes con respecto a la versión anterior.

Las mejoras más importantes de este nuevo estándar son:

- Nuevas etiquetas para formar la estructura de la página: article, footer, header, nav y section. De este modo se puede dar una mejor estructuración a la página y el código queda mucho más claro y limpio.
- Etiquetas de audio y video: Ya es posible reproducir audios y videos con simples etiquetas.
- Etiqueta <progress>: Para añadir una barra de progreso a cualquier proceso.
- Etiqueta <canvas>: Para poder crear gráficos multimedia e interactivos.
- Etiqueta <device>: Para tener acceso a determinados dispositivos (webcam, micrófonos...).
- Nuevos atributos para las etiquetas: Estos atributos pueden proporcionar más funcionalidad a las etiquetas.

5.2.5. CSS

Para dar estilo a las páginas HTML se han usado hojas de estilo CSS^[12]. Dichas hojas de estilo son externas a los documentos HTML. De esta manera separamos la estructura de la página del estilo de la misma, proporcionando una mejor ordenación del código. Además muchos estilos pueden ser utilizados en muchas páginas y de esta manera también mejoramos la reutilización de código.

Otra ventaja de utilizar hojas externas es la facilidad de mantenimiento de la web porque si se cambia un estilo de la hoja del CSS se cambiará en todas las páginas de la web. En caso de no utilizar hojas externas habría que modificar el estilo página por página.

Para la aplicación se ha utilizado la última versión de CSS (CSS3).

5.2.6. BOOTSTRAP

Bootstrap^[13] es un framework Open Source para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, botones, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript. Además las plantillas predeterminadas también incluyen los estilos para crear la interfaz responsive.



Aun habiendo utilizado Bootstrap como base, hemos realizado cambios en los estilos y añadidos nuevas hojas de estilo para mejorar y adaptar a nuestras necesidades la interfaz de usuario.

5.2.7. AJAX

En algunas partes de la web hemos necesitado realizar consultas de datos desde la propia interfaz, pero sin tener que recargar la página. Para ello hemos hecho uso de Ajax^[14].

Ajax es una técnica que permite realizar comunicaciones asíncronas entre cliente y servidor. Gracias a estas conexiones se pueden actualizar datos de la página sin recargar la misma. Además ofrece una mejora de velocidad y usabilidad.

5.2.8. JAVASCRIPT

Para incluir funcionalidad en la parte del cliente se ha utilizado JavaScript^[15]. Actualmente todos los navegadores son capaces de interpretar el código JavaScript. Para que el código JavaScript interactúe con la interfaz se hace uso del DOM (Document Object Model)^[16].

Para nuestro proyecto hemos utilizado una biblioteca JavaScript llamada jQuery^[17]. JQuery es una biblioteca que facilita y simplifica la interacción con la interfaz, proporcionando distintos eventos y una amplia funcionalidad ya implementada.

5.2.9. PLUGINS Y LIBRERÍAS

Muchas de las funcionalidades que hemos necesitado en la aplicación web ya se podían conseguir gracias a librerías y plugins jQuery. Nosotros hemos hecho uso de ellos para obtener una buena base en la funcionalidad, pero también hemos tenido que modificar alguna funcionalidad en concreto o el estilo para que se adecuara a nuestras necesidades.

Las librerías y plugins utilizados han sido:

- Validation engine^[18]: Para realizar la validación de los formularios en la parte del cliente.
- Select2^[19]: Para disponer de componentes <select> en los formularios, donde se puedan realizar búsqueda de valores. Se muestra un ejemplo en la figura 33.

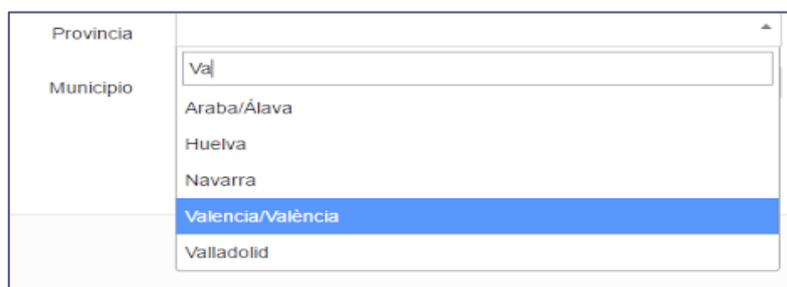


Figura 33. Ejemplo de Select2.

- MaskedInput^[20]: Para crear máscaras de datos en los formularios.

- HighCarts^[21]: Para la creación de gráficas interactivas.
- Google Maps: Para hacer uso de la API de Google Maps y poder ofrecer mapas con ubicaciones.
- Full calendar^[22]: Para ofrecer un completo calendario.
- DataTables^[23]: Para mostrar los datos en tablas, ofreciendo funcionalidades de buscar, ordenar... Se muestra un ejemplo en la figura 34.

The screenshot shows a web interface titled 'Instalación'. It features a search bar and a dropdown menu set to '10 Entradas'. Below is a table with the following data:

Nombre de la instalación	Teléfono fijo	Teléfono movil	Instalación propia	Acciones
CEIP Rodriguez Fornos			🏠	Acciones ▾
Coelgio Pio XII			🏠	Acciones ▾
Colegio Salesianos San Juan Bosco			🏠	Acciones ▾
Colegio Agustinos			🏠	Acciones ▾
Colegio Cumbres			🏠	Acciones ▾
Colegio El Armelar			🏠	Acciones ▾
Colegio el Pilar			🏠	Acciones ▾
Colegio Esclavas Sagrado Corazon			🏠	Acciones ▾
Colegio Jesús María Fuensanta			🏠	Acciones ▾
Colegio Juan Comenius			🏠	Acciones ▾

At the bottom of the table, there is a pagination control showing page numbers 1, 2, 3, 4 and navigation arrows.

Figura 34. Ejemplo de Datatable.

- CkEditor^[24]: Es un editor wysiwyg que hemos utilizado para la redacción de noticias y eventos.
- Cropper^[25]: Es una librería jQuery que permite cortar y voltear imágenes. Ha sido utilizado para editar las fotos de perfil, el escudo del equipo y las imágenes de cabecera.

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^[26], uno de los motivos de su elección es que esta 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).

5.3.2. Control de versiones

Como se ha mencionado anteriormente, Eclipse incluye GIT como control de versión. Se llama control de versiones^[27] 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.

5.3.2.1 GIT

Git^[28] 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.

5.3.2.2 Bitbucket

Bitbucket^[29] 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.

5.3.3. Servidor web

Como servidor Http se ha usado Apache^[30]. 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.

5.3.4 Base de datos

Para el almacenamiento de la información se ha utilizado una base de datos MySQL^[31]. MySQL es un sistema de gestión de bases de datos relacional. Además es una base de datos Open Source.

5.4. Descripción de la implementación

5.4.1. Estructura de directorios

Para la implementación del código hemos usado un framework PHP llamado CodeIgniter. Dicho framework ya nos crea un directorio estructurado que se describe en la figura 35.

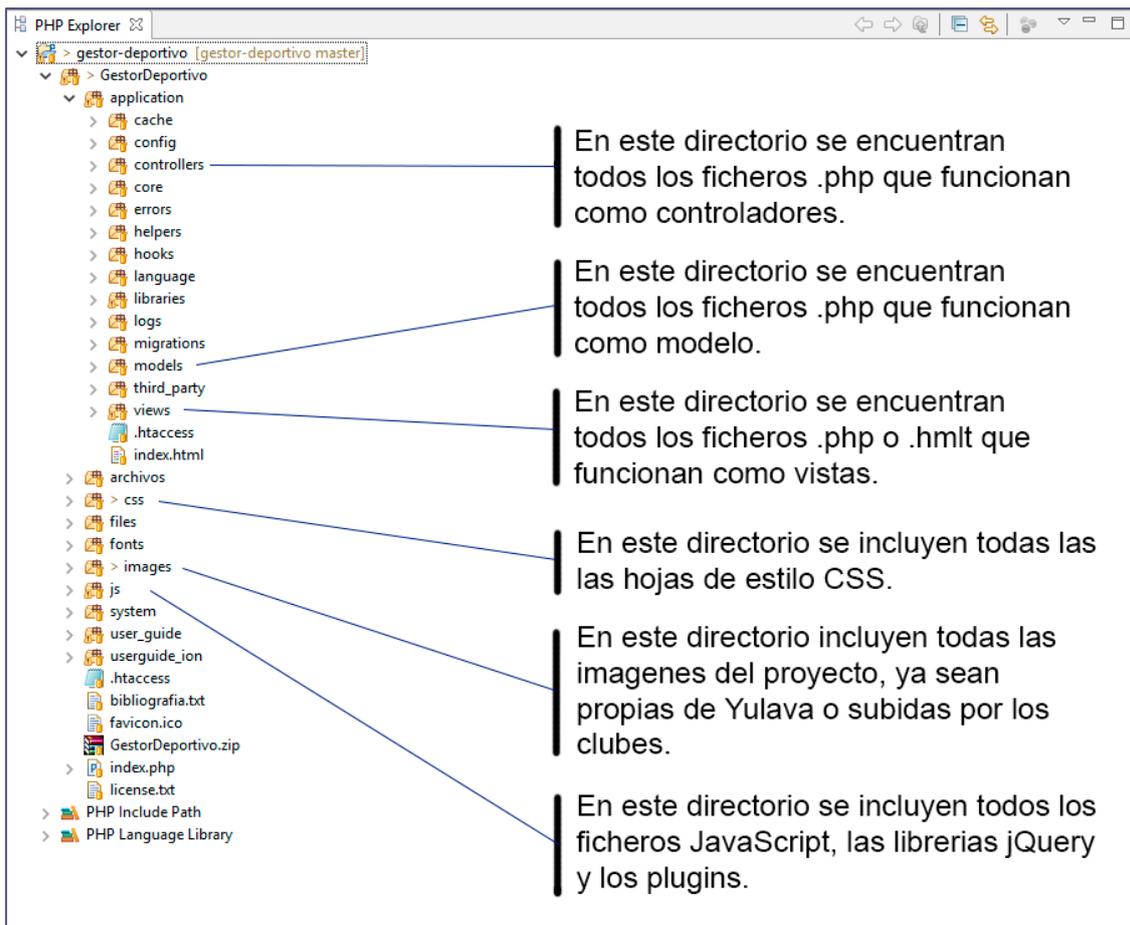


Figura 35. Estructura de directorios.

En la imagen anterior se han descrito los principales directorios, pero también hacemos uso de los siguientes directorios:

- **Cache:** Directorio donde se guardan los archivos cacheables del programa así como las consultas a base de datos.
- **Config:** En este directorio se encuentran todos los archivos de configuración.
- **Languages:** En este directorio se añaden los ficheros con las traducciones.
- **Libraries:** Directorio donde se encuentran las librerías php utilizadas en el proyecto.
- **Logs:** En este directorio se guardan todos los archivos Log que tengamos habilitados en el sistema.

5.5. Interfaces de usuario

En este apartado vamos a mostrar las principales interfaces de la aplicación. Como la web es responsive, vamos a mostrar también algunas capturas de la web desde un Smartphone.

La página principal de la aplicación corresponde a lo mostrado en la figura 36. En ella se puede ver que desde la página principal se puede crear un nuevo club o iniciar sesión. También dispone en la parte superior de un buscador de clubes.

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

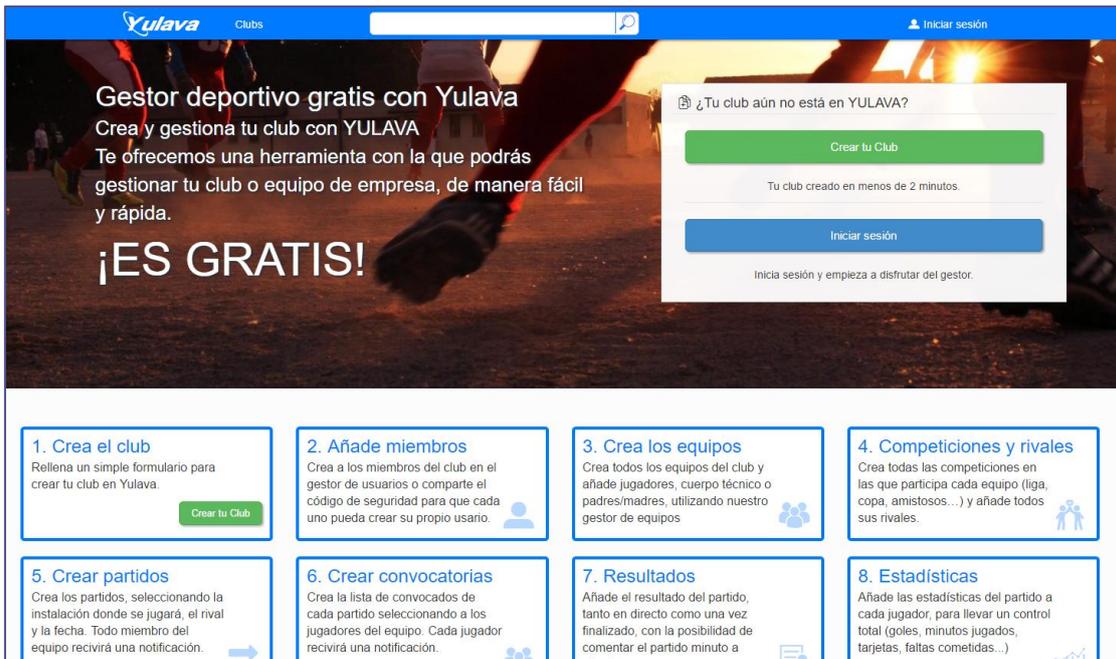


Figura 36. Página principal.

Al pulsar sobre el botón de iniciar sesión aparece una ventana donde introducir los datos de usuario. Se muestra en la figura 37.

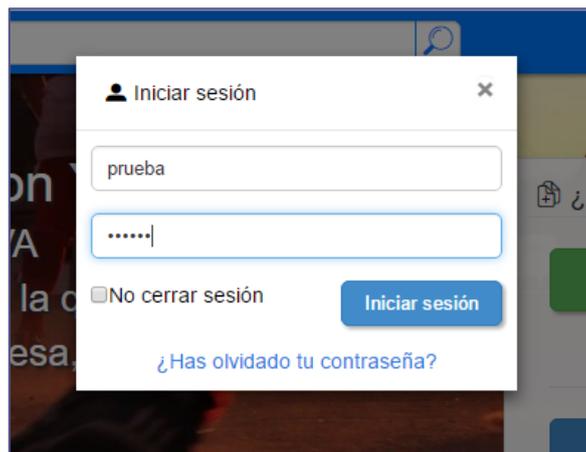


Figura 37. Inicio de sesión.

Al pulsar sobre crear club se muestra el formulario de creación del club tal y como se puede observar en la figura 38.

Figura 38. Formulario de crear club.

En las páginas de un club se muestra la cabecera con el escudo del club y una imagen. También se observa a la parte izquierda el menú del club y la parte derecha encontraremos el contenido de cada página. Figura 39.

Figura 39. Página de un club.

A continuación expondremos las capturas anteriores pero esta vez con un tamaño de pantallas de Smartphone para que se pueda comprobar cómo se adapta la interfaz dependiendo del dispositivo. Figura 40.

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

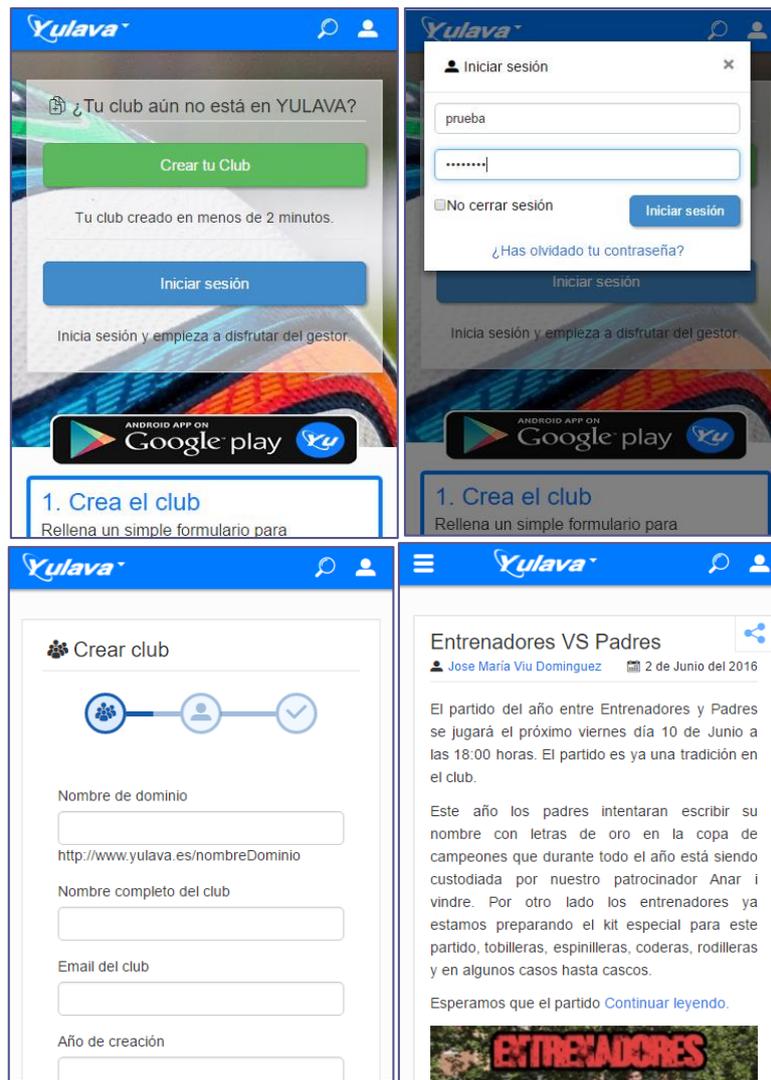


Figura 40. Versión responsive de la web.

Como hemos dicho anteriormente, el proyecto web hace uso de Bootstrap. Para realizar uso de los estilos y componentes que incluyen Bootstrap, lo primero necesario es enlazar nuestras páginas con los ficheros externos necesarios. En nuestro caso utilizaremos los ficheros CSS y JS de Bootstrap.

Los links a los ficheros CSS se han incluido en el <head> y se muestra un ejemplo en la figura 41.

```

1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" lang="es">
3 <head>
4 <meta charset="<?php echo $this->config->item('charset');?>">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1.0, user-scal
7 <link rel="icon" type="image/x-icon" href="<?php echo base_url('favicon.ico')?>" />
8 <title><?php if(isset($titulo_pagina)) echo $titulo_pagina ?></title>
9 <meta name="keywords" content="<?php if(isset($keywords_pagina)) echo $keywords_pagina ?>">
10 <meta name="description" content="<?php if(isset($descripcion_pagina)) echo $descripcion_pagina
11 <script>
12     (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
13     (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),
14     m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
15     })(window,document,'script','https://www.google-analytics.com/analytics.js','ga');
16
17     ga('create', 'UA-35330214-5', 'auto');
18     ga('send', 'pageview');
19
20 </script>
21
22
23 <?php
24 $this->carabiner->css('theme/bootstrap.css');
25 $this->carabiner->css('changestheme.css');
26 $this->carabiner->css('imagesfilters.css');
27 $this->carabiner->css('other.css');
28 $this->carabiner->css('pageclub.css');
29 $this->carabiner->css('versions.css');
30 $this->carabiner->css('flaticon.css');
31 $this->carabiner->css('admin_club.css');
32 $this->carabiner->css('../js/libraries/dataTables/css/jquery.dataTables.css');
33 $this->carabiner->css('../js/libraries/validationEngine/validationEngine.jquery.css');
34 $this->carabiner->css('../js/libraries/select2/select2.min.css');
35 $this->carabiner->css('../js/libraries/cropper/dist/cropper.css');
36 $this->carabiner->css('../js/libraries/cropper/crop-avatar/css/main.css');
37 $this->carabiner->css('../js/libraries/unslider-master/dist/css/unslider.css');
38 $this->carabiner->css('../js/libraries/datetetimepicker/css/bootstrap-datetimepicker.css');
39 $this->carabiner->css('../js/libraries/toggle/bootstrap-switch.css');
40 $this->carabiner->css('../js/libraries/bootstrap-slider/bootstrap-slider.css');
41 $this->carabiner->css('../js/libraries/PhotoSwipe-master/dist/photoswipe.css');
42 $this->carabiner->css('../js/libraries/AutomaticImageMontage/css/style.css');
43 $this->carabiner->css('../js/libraries/sweet-alert/sweetalert.css');
44 $this->carabiner->display('css');
45 ?>
46
47 <link rel="stylesheet" href="<?php echo base_url('js/libraries/PhotoSwipe-master/dist/default-sk
48
49 <!--
50 <link href="<?php echo base_url('css/theme/bootstrap.css')?>" rel="stylesheet">
51 <link href="<?php echo base_url('css/changestheme.css')?>" rel="stylesheet">
52 <link href="<?php echo base_url('css/imagesfilters.css')?>" rel="stylesheet">

```

Figura 41. Ejemplo de carga CSS.

Los archivos JavaScript se han incluido al final del <footer>, de este modo la carga de la interfaz se realiza más rápido. Figura 42.

```
3
4 <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 sin_padding_escritorio">
5   <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12 sin_padding_escritorio footer">
6
7     <?php $this->load->view('includes/footer_content');?>
8   </div>
9 </div>
10
11
12
13 <?php
14
15 $this->carabiner->js('bootstrap/bootstrap.js');
16 $this->carabiner->js('libraries/select2/select2.js');
17 $this->carabiner->js('libraries/maskedInput/jquery.maskedinput.js');
18 $this->carabiner->js('libraries/validationEngine/jquery.validationEngine.js');
19 $this->carabiner->js('libraries/validationEngine/jquery.validationEngine-es.js');
20 $this->carabiner->js('libraries/fancybox/jquery.fancybox.js');
21 $this->carabiner->js('libraries/dataTables/js/jquery.dataTables.js');
22 $this->carabiner->js('libraries/SlidePushMenus/classie.js');
23 $this->carabiner->js('libraries/SlidePushMenus/modernizr.custom.js');
24 $this->carabiner->js('libraries/sweet-alert/sweetalert.min.js');
25 $this->carabiner->js('libraries/toggle/bootstrap-switch.js');
26 $this->carabiner->js('libraries/bootstrap-slider/bootstrap-slider.js');
27 $this->carabiner->js('libraries/cropper/crop-avatar/js/main.js');
28 $this->carabiner->js('libraries/cropper/dist/cropper.min.js');
29 $this->carabiner->js('libraries/unslider-master/src/js/unslider.js');
30 $this->carabiner->js('libraries/PhotoSwipe-master/dist/photoswipe.min.js');
31 $this->carabiner->js('libraries/PhotoSwipe-master/dist/photoswipe-ui-default.min.js');
32 $this->carabiner->js('libraries/AutomaticImageMontage/js/jquery.montage.js');
33 $this->carabiner->js('libraries/datettimepicker/js/bootstrap-datettimepicker.js');
34 $this->carabiner->js('libraries/datettimepicker/js/locales/bootstrap-datettimepicker.es.js');
35 $this->carabiner->display('js');
36 ?>
37
38
39
40 <!--
41 <script src="<?php echo base_url('js/libraries/plugins.js');?>"></script>
42 <script src="<?php echo base_url('js/bootstrap/bootstrap.js');?>"></script>
43 <script src="<?php echo base_url('js/libraries/select2/select2.js');?>"></script>
44 <script src="<?php echo base_url('js/libraries/maskedInput/jquery.maskedinput.js');?>"></script>
45 <script src="<?php echo base_url('js/libraries/validationEngine/jquery.validationEngine.js');?>"></script>
46 <script src="<?php echo base_url('js/libraries/validationEngine/jquery.validationEngine-es.js');?>"></script>
47 <script src="<?php echo base_url('js/libraries/fancybox/jquery.fancybox.js');?>"></script>
48 <script src="<?php echo base_url('js/libraries/dataTables/js/jquery.dataTables.js');?>"></script>
49 <script src="<?php echo base_url('js/libraries/SlidePushMenus/classie.js');?>"></script>
50 <script src="<?php echo base_url('js/libraries/SlidePushMenus/modernizr.custom.js');?>"></script>
51 <script src="<?php echo base_url('js/libraries/sweet-alert/sweetalert.min.js');?>"></script>
52 <script src="<?php echo base_url('js/libraries/toggle/bootstrap-switch.js');?>"></script>
```

Figura 42. Ejemplo de carga JS.

Tal y como se muestra en las figuras 41 y 42, los ficheros CSS han sido incluidos utilizando la librería Carabiner, utilizada para unir y minificar los ficheros CSS y JS. De esta forma mejoramos la velocidad de la página.

Una vez tenemos los ficheros de Bootstrap relacionados con nuestras web, podemos hacer uso de componentes o estilos utilizando las clases. En la figura 43 se muestra un ejemplo de utilización de clases haciendo uso del atributo “class”. En este caso estamos utilizando un componente de Bootstrap que nos permite abrir ventanas emergentes.

```
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-variable-initsession">
    <div class="modal-content">
      <form class="black" id="validationlogin">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-label="close">
            <span aria-hidden="true">&times;</span>
          </button>
          <span class="modal-title flaticon-silhouette78" id="myModalLabel"><?php echo lang('init_session');?></span>
        </div>
```

Figura 43. Ejemplo de uso del atributo class.

Otro detalle importante de Bootstrap es hacer uso de él para crear páginas responsive. Dependiendo del tamaño de pantalla se mostrara de una forma u otra. Por ejemplo en la figura 44 se muestran dos etiquetas div en las cuales encontramos diferentes clases que hacen referencia a columnas.

Dependiendo del tamaño de pantalla serán válidas unas clases u otras. En este caso si la pantalla es grande (col-lg-4) o mediana (col-md-4) el div ocupará un espacio de 4/12 del total. En caso de ser pequeña (col-sm-6) o muy pequeña (col-xs-6) ocupará 6/12 del total.

```
<div class="col-lg-4 col-md-4 col-sm-6 col-xs-6 sin padding escritorio right">  
  <div class="menu buttons escritorio">
```

Figura 44. Ejemplo de uso responsive.

6. Evaluación

6.1 Introducción

En este capítulo vamos a validar que cumplimos en la medida de lo posible con todos los estándares web. Para ello utilizaremos las herramientas que nos ofrece el World Wide Web Consortium^[32] (W3C). En nuestro caso haremos uso del validador de HTML, CSS y de enlaces rotos. También realizaremos un análisis del SEO de la web y otro de la velocidad de la página.

6.2. Validación de estándares

6.2.1 Validación HTML

La validación HTML la hemos realizado de las principales páginas de la aplicación obteniendo los siguientes resultados:

En la figura 45 se muestra el resultado de la validación HTML de la página principal de Yulava.

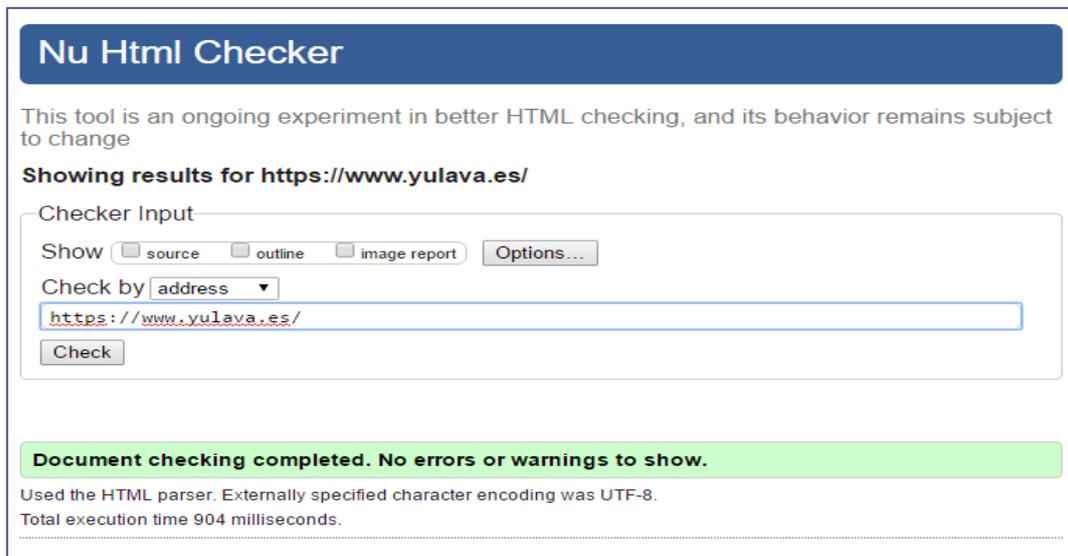


Figura 45. Página de inicio de Yulava.

En la figura 46 se muestra el resultado de la validación HTML del buscador de clubes.

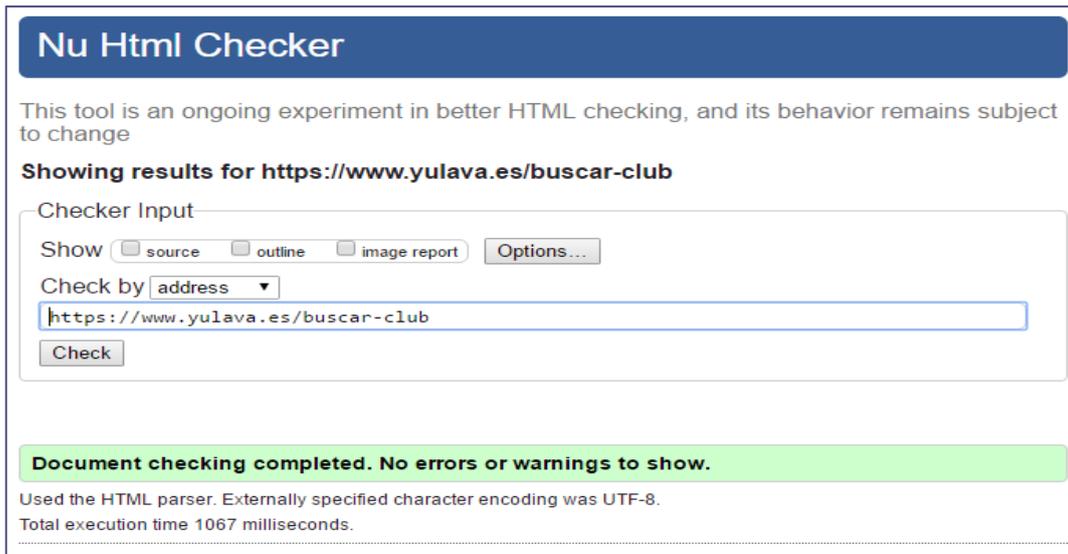


Figura 46. Página de buscar clubes.

En la figura 47 se muestra el resultado de la validación HTML de la página de contacto de Yulava.

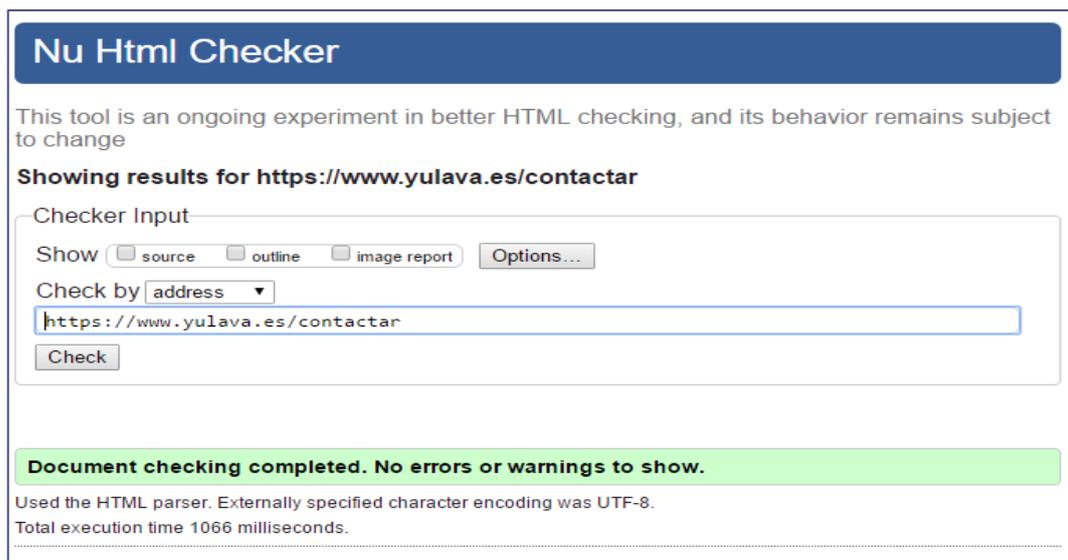


Figura 47. Página de contactar.

En la figura 48 se muestra el resultado de la validación HTML del aviso legal.

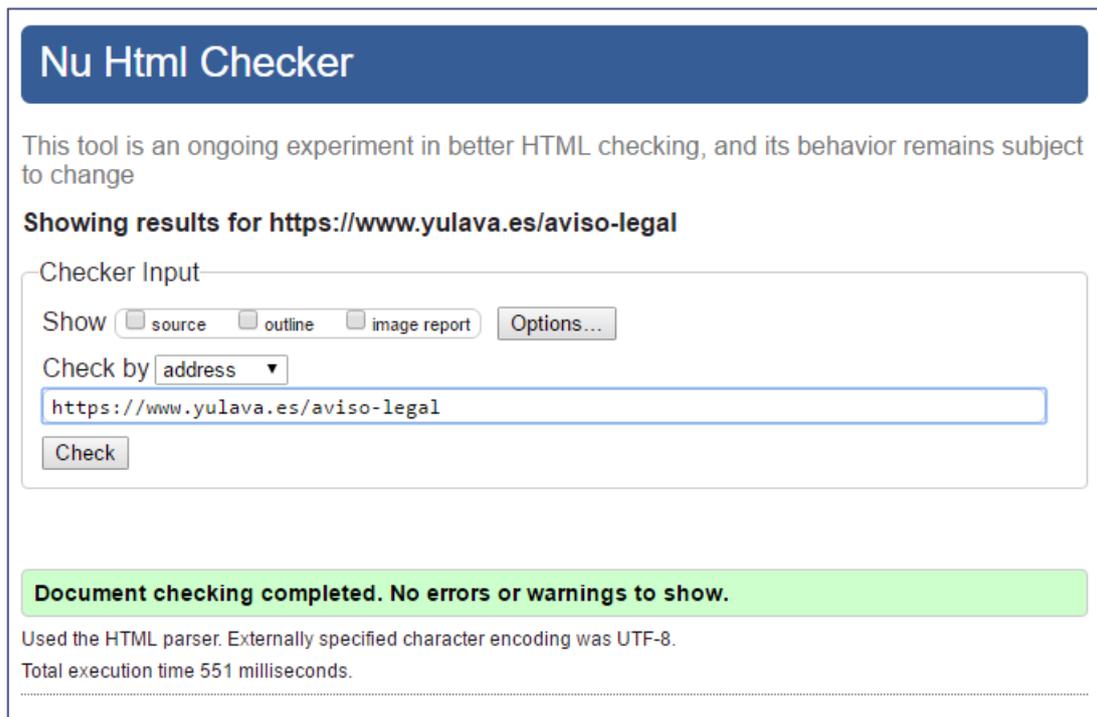


Figura 48. Página de aviso legal.

En la figura 49 se muestra el resultado de la validación HTML de la página de recuperar contraseña.

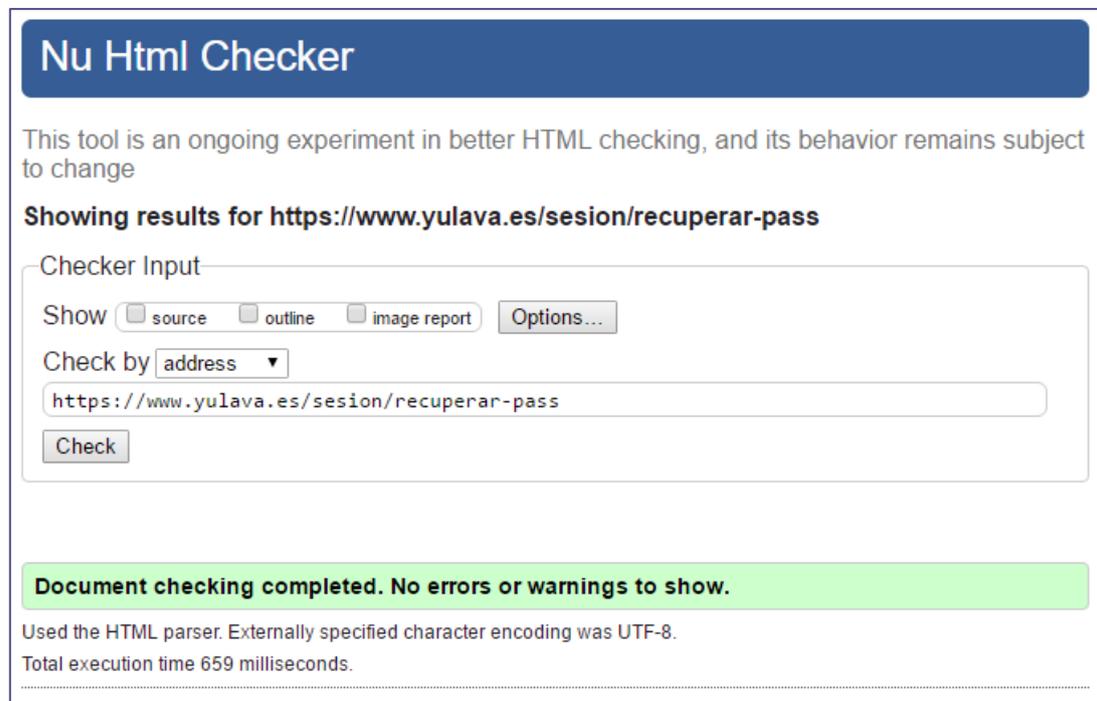


Figura 49. Página de recuperar contraseña.

Como se puede comprobar con las validaciones, se puede decir que hemos seguido el estándar HTML5, ya que no se detecta ningún error en las páginas comprobadas.

6.2.2. Validación CSS

Los archivos CSS han sido unificados en uno solo por esta razón, únicamente será necesario realizar la validación a la página principal de Yulava.

Los resultados obtenidos se muestran en la figura 50.



W3C[®] El Servicio de Validación de CSS del W3C
Resultados del Validador CSS del W3C para <https://www.yulava.es> (CSS versión 3)

Ir a: [Los Errores \(261\)](#) [Las Advertencias \(571\)](#) [Su Hoja de Estilo validada](#)

Resultados del Validador CSS del W3C para <https://www.yulava.es> (CSS versión 3)

Disculpas! Hemos encontrado las siguientes errores (261)

URI : <https://www.yulava.es/css/1463408448d8405030ae6affc59f268e492384f763.css>

1	<code>input[type="radio"], input[type="checkbox"]</code>	Propiedad no válida : margin-top Faltan valores o no se rec
1		La propiedad -webkit-min-device-pixel-ratio no existe en el {input[type="date"],input[type="time"],input[type="datetime- height:43px;}input[type="date"].input-sm,input[type="time"] sm,input[type="month"].input-sm,.input-group-sm input[typ input[type="time"],.input-group-sm input[type="datetime-loc height:31px;}input[type="date"].input-lg,input[type="time"].i

Figura 50. Resultado de la validación de CSS.

El número de errores y advertencias son elevados, por lo que se deberían reparar e intentar solventar cada uno de ellos.

Después de realizar un análisis de los errores, hemos detectado que todos proceden de hojas de estilo de plugins o de Bootstrap.

6.2.3. Validación de enlaces rotos

El test pasado a la página principal de Yulava nos ha devuelto el siguiente informe. Figura 51.

Processing <https://www.yulava.es/>

Go to [the results](#). Settings used:

- **Accept:** text/html, application/xhtml+xml;q=0.9, application/vnd.wap.xhtml+xml;q=0.6, */*;q=0.5
- **Accept-Language:** es-ES,es;q=0.8
- **Referer:** sending
- Sleeping 1 second between requests to each server

For reliable link checking results, check [HTML validity](#) and [CSS validity](#) first.

Back to the [link checker](#).

Status: Done. Document processed in 40.49 seconds.

```
Checking link https://www.yulava.es/session/recuperar-pass
HEAD https://www.yulava.es/session/recuperar-pass fetched in 1.46 seconds

Checking link https://www.yulava.es/images/theme/logo_yulava_white.png
HEAD https://www.yulava.es/images/theme/logo_yulava_white.png fetched in 1.42 seconds

Checking link https://www.yulava.es/css/146315020191a3e1857eae3ef6ca78759c835ebc98.js
HEAD https://www.yulava.es/css/146315020191a3e1857eae3ef6ca78759c835ebc98.js fetched in 1.44 seconds

Checking link https://www.yulava.es/js/views/inicio/index.js
HEAD https://www.yulava.es/js/views/inicio/index.js fetched in 1.43 seconds
```

Results

List of broken links and other issues

There are issues with the URLs listed below. The table summarizes the issues and suggested actions by HTTP response status code.

Code	Occurrences	What to do
(N/A)	1	Accessing links with this URI scheme has been disabled in link checker.

◆ Line: 416 **javascript:void(0);**

Status: (N/A) Access to 'javascript' URIs has been disabled

You must change this link: people using a browser without JavaScript support will not be able to follow this link. See the [Web Content Accessibility Guidelines on the use of scripting on the Web](#) and the [techniques on how to solve this](#).

Figura 51. Resultado de la validación de enlaces rotos.

Tal y como se muestra en la figura 51, existe una ocurrencia con código desconocido, pero tal y como se aclara, no es un error, simplemente el validador no comprueba los enlaces creados con JavaScript. Por tanto no existe ningún enlace roto.

6.3. Análisis SEO

Para el análisis del SEO de la aplicación hemos utilizado la herramienta online WooRank^[33]. Dicha herramienta realiza un análisis completo de la web y te ofrece unos consejos para solucionar o mejorar los puntos débiles de la web.

El resultado obtenido se muestra en la figura 42.

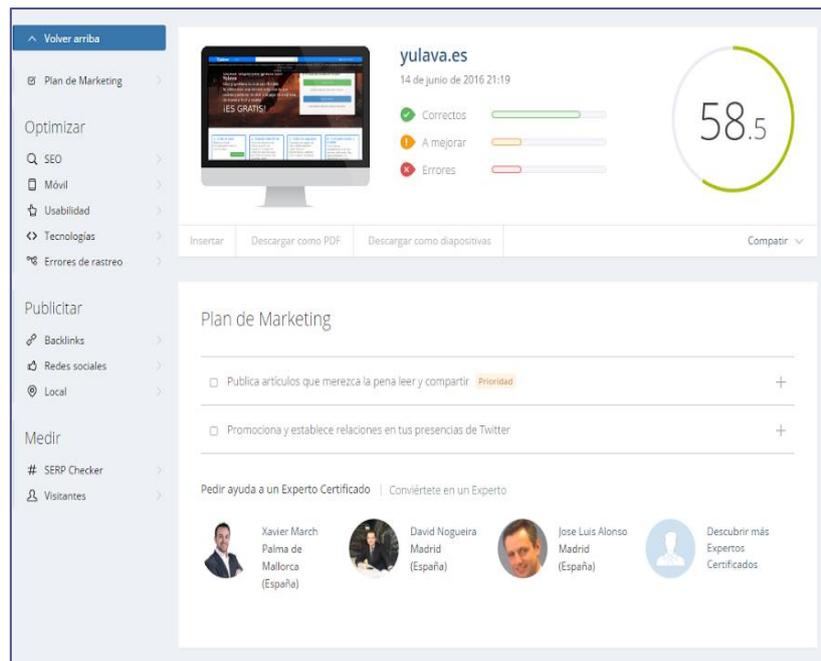


Figura 52. Resultado del análisis SEO.

Los errores detectados por la herramienta están los siguientes:

- **No se ha encontrado ningún blog en este sitio web.** Este apartado es muy importante para el posicionamiento en Google, pero por el momento no hemos implementado un apartado en la aplicación web que sirva de Blog.
- **Ningún Marcado de Datos Estructurados ha sido detectado.** Dichos marcadores ayudan a los buscadores a detectar los diferentes tipos de datos que contiene una web (noticia, galería de imágenes, productos, opiniones...). No detecta los marcadores de datos estructurados porque en la página principal no existe ningún dato estructurado, pero si entramos en las páginas de los clubes, las noticias y las galerías de imágenes sí que implementan los marcadores de datos.
- **Estimación del tráfico.** Este es un análisis que realiza WooRank estimando el tráfico que puedes tener. En nuestro caso el tráfico es muy bajo. Esto es debido a que nuestra web lleva muy pocos meses publicada y aún no tiene la autoridad suficiente en los buscadores.
- **Puntuación Backlinks.** Este apartado hace referencia a las páginas web que tienen un link apuntando a tu web. Cuanto mayor sea el número de links y mejor sea la calidad de estos links mayor autoridad tendremos en los buscadores. En nuestro caso solo tenemos redirecciones desde algunas webs de clubes que ya disponen de Yulava.

6.4. Análisis de rendimiento

Para realizar el análisis de rendimiento de la web hemos utilizado la herramienta online GTMetrix^[34]. Dicha herramienta realiza un análisis completo de la web y te ofrece unos consejos para solucionar o mejorar los puntos débiles de la web.

El resultado obtenido se muestra en la figura 53.

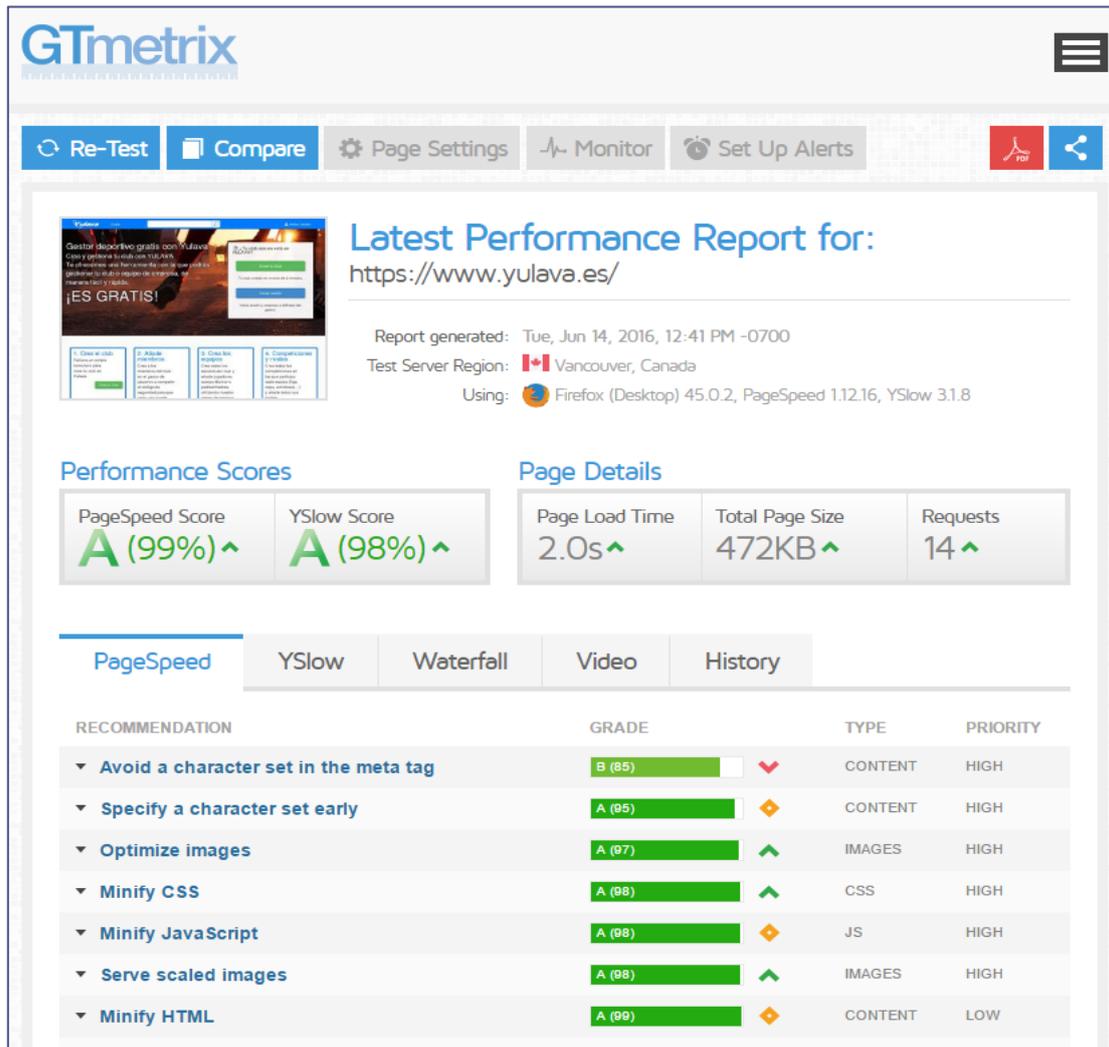


Figura 53. Resultado del análisis de rendimiento.

Tal y como se muestra se ha realizado una buena optimización, entre las que destacan:

- **Habilitar compresión gzip.** Es un sistema muy utilizado y permite comprimir los archivos en el servidor, enviarlos y por último el cliente los descomprime.
- **Habilitar caché.** En caso de habilitar la caché para archivos estáticos se consigue ahorrar en el número de peticiones al servidor, ya que muy posiblemente el cliente ya disponga de algunos archivos y no será necesario volver a pedirlos.
- **Optimizar las imágenes.** Gracias a la optimización de imágenes se consigue un ahorro en el peso de las imágenes mejorando así el tiempo de carga de dichas imágenes.
- **Minificar y unir los css y js.** De esta manera conseguimos reducir el número de archivos que se tienen que descargar y el tamaño del mismo.

7. Conclusiones

7.1 Resumen

Este proyecto ha consistido en la elaboración de una aplicación web para la gestión de clubes deportivos. En dicho proyecto se han seguido las etapas propias de la ingeniería del software: especificación de requisitos, análisis, diseño, implementación y evaluación.

Las primeras etapas del proyecto (especificación de requisitos, análisis y diseño) se han realizado conjuntamente entre la parte encargada del Front-End y la parte encargada del Back-End. Las siguientes etapas se han diversificado en Front-End y Back-End.

En la etapa de Implementación, las dos partes del proyecto estaban muy relacionadas; por tanto, debíamos estar compenetrados para determinar las conexiones entre la lógica del programa y la interfaz de usuario.

Para realizar un buen proyecto, hemos hecho uso de multitud de documentación de todas las tecnologías usadas. Dicha documentación aparece reflejada en la bibliografía de la memoria.

Una vez acabada la Implementación, se realizó una evaluación de la aplicación. Para la parte de Front-End se realizó un análisis de la interfaz, donde se midió la velocidad de carga, el SEO de la página y se pasaron diferentes pruebas para comprobar que habíamos seguido los estándares web correctamente.

7.2. Valoración

En este proyecto se han utilizado habilidades y conocimientos adquiridos durante los cuatro años de titulación en Ingeniería Informática. A parte de dichos conocimientos, al realizar este proyecto, se ha podido profundizar más en aspectos de ingeniería del software y de programación web.

La etapa de implementación ha sido en la que más documentación hemos necesitado, ya que se ha utilizado gran cantidad de tecnologías que en un principio desconocíamos.

Al finalizar el proyecto, hemos adquirido gran cantidad de conocimientos en programación web, configuración del servidor, seguridad web y SEO.

La valoración que tengo después de la realización del trabajo ha sido muy positiva, ya que es un proyecto de gran envergadura y nos ha servido para obtener una amplia experiencia en este sector.

7.3. Trabajo futuro

Una vez acabada la primera versión del proyecto, fuimos a ofertarlo a diferentes clubes de Valencia. Actualmente, Yulava contiene seis clubes deportivos. Dicho clubes nos han dado ideas y mejoras para obtener así una herramienta más completa.

Para futuras versiones vamos a añadir las siguientes funcionalidades:

- **Gestor de archivos para los miembros del club:** En este gestor, por ejemplo, los entrenadores podrán subir documentos que contengan ejercicios de entrenamiento, documentos con normativas...
- **Más niveles de privilegio:** Se añadirá un nuevo perfil de usuario. Dicho perfil será el de coordinador, que se encontrará entre el administrador y los entrenadores.
- **Mejoras en privilegios:** De forma individual, el usuario administrador podrá añadir o quitar permisos específicos a cada usuario.
- **Gestor de comunicaciones:** Se creará un nuevo apartado que servirá como chat dentro de Yulava, donde los miembros del club podrán contactar rápidamente con otros miembros. Por ejemplo, los entrenadores lo pueden utilizar para ponerse en contacto con los padres de los jugadores.
- **Aplicaciones Android e IOs:** Se desarrollarán aplicaciones nativas para los sistemas móviles más utilizados. En dichas aplicaciones, existirán todas las opciones que se encuentran en la web. Además, se dotará a estas aplicaciones con notificaciones instantáneas.

8. Referencias bibliográficas

- [1] Futbolvista. Junio 2016. <http://www.futbolvista.com>
- [2] Miclub-manager. Junio 2016. <http://www.miclub-manager.es>
- [3] Myequipo. Junio 2016. <http://www.myequipo.com>
- [4] Ley Orgánica 15/1999, de 13 de Diciembre.
<https://www.boe.es/buscar/doc.php?id=BOE-A-1999-23750>
- [5] Google Maps. Junio 2016. <https://developers.google.com/maps/?hl=es>
- [6] API de Google for Work. Junio 2016. <https://developers.google.com/maps/premium/>
- [7] UML (Unified Modeling Language) Junio 2016. <http://www.uml.org/>
- [8] CodeIgniter Web Framework. Junio 2016. <https://www.codeigniter.com/>
- [9] PHP. Junio 2016. <https://secure.php.net/manual/es/index.php>
- [10] MySQL language. Junio 2016. <http://dev.mysql.com/doc/refman/5.7/en/language-structure.html>
- [11] HTML5. Junio 2016. <https://www.w3.org/html/>
- [12] CSS3. Junio 2016. <https://www.w3.org/Style/CSS/>
- [13] Framework BootStrap. Junio 2016. <http://getbootstrap.com/>
- [14] AJAX. Junio 2016. <https://es.wikipedia.org/wiki/AJAX>
- [15] JavaScript. Junio 2016. <http://www.w3schools.com/js/>
- [16] DOM (Document Object Model). Junio 2016.
https://es.wikipedia.org/wiki/Document_Object_Model
- [17] JQuery. Junio 2016. <https://jquery.com/>
- [18] Validation-Engine. Junio 2016. <https://github.com/posabsolute/jquery-Validation-Engine>
- [19] Select2. Junio 2016. <https://github.com/select2/select2>
- [20] MaskedInput. Junio 2016. <https://github.com/digitalBush/jquery.maskedinput>
- [21] HighCharts. Junio 2016. <http://www.highcharts.com/>
- [22] FullCalendar. Junio 2016. <http://fullcalendar.io/docs/usage/>
- [23] Datatables. Junio 2016. <https://datatables.net/>



- [24] CKEditor. Junio 2016. <http://ckeditor.com/about>
- [25] Cropper. Junio 2016. <https://github.com/fengyuanchen/cropper>
- [26] Eclipse for PHP Developers -Version: Mars Release (4.5.0). Junio 2016
<http://www.eclipse.org/downloads/packages/release/mars/r>
- [27] Control de versiones. Junio 2016.
https://es.wikipedia.org/wiki/Control_de_versiones
- [28] GIT. Junio 2016 <https://git-scm.com/>
- [29] BitBucket. Junio 2016 <https://bitbucket.org/>
- [30] Apache. Junio 2016. <http://www.apache.org/>
- [31] MySQL Database. <https://www.mysql.com>
- [32] W3C (World Wide Web Consortium). Junio 2016. <http://www.w3c.es/>
- [33] WooRank. Junio 2016. <https://www.woorank.com/es/>
- [34] GTmetrix. Junio 2016. <https://gtmetrix.com/>