



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de una red social deportiva con el framework Yii

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Morillo Llorca, Joan Antoni

Tutor: Sáez Barona, Sergio

2015/2016

Resumen

El objetivo de este proyecto es diseñar e implementar una red social mediante un framework de PHP llamado Yii. La red social será similar a las ya existentes Facebook o Tuenti, pero incluyendo algunas secciones y funcionalidades que motiven a la práctica diaria de algún deporte. La red social estará compuesta de un conjunto de usuarios que deberán registrarse para poder disfrutar de los servicios de la red social. Cada usuario tendrá asociado un perfil en el que podrá editar ciertas características como datos personales, aficiones favoritas en su tiempo libre y, por supuesto, sus deportes favoritos y/o más practicados. La red social permitirá, aparte de la formación de relaciones de amistad, compartir aficiones, intenciones deportivas, sesiones de prácticas deportivas, etc.

Palabras clave: Yii, red, social, framework, php, deportes.

Abstract

The objective of this project is to design and implement a social network using a PHP framework called Yii. This social network will be similar to existing social networks as Facebook or Tuenti, but including some new functionalities in order to motivate people to introduce and a sport in their daily life. The social network will include a group of users previously registered so that they can start using it. Each user will have their own profile and they will be able to add some personal information, preferences and hobbies usually done in their spare time. Besides, the social network will let the users to start new friendships, to share hobbies, sport goals and meetings, etc.

Keywords : Yii, social network, framework, PHP, sports.



Tabla de contenidos

1.	Introducción.....	8
2.	Especificación de requisitos.....	9
2.1.	Introducción	9
2.1.1.	Propósito	9
2.1.2.	Ámbito	9
2.1.3.	Definiciones, Siglas y Abreviaturas	10
2.1.4.	Referencias	10
2.1.5.	Visión Global	11
2.2.	Descripción general	11
2.2.1.	Perspectiva del producto	11
2.2.2.	Funciones del producto	11
2.2.3.	Características de usuario	12
2.2.4.	Restricciones	13
2.2.5.	Supuestos y dependencias	13
2.3.	Requisitos específicos	13
2.3.1.	Requisitos de la interfaz externa	14
2.3.2.	Requisitos funcionales.....	14
2.3.3.	Otros requisitos.	16
3.	Análisis.....	17
3.1.	Diagrama de clases/entidades.....	17
3.2.	Modelado UML.....	19
3.3.	Casos de uso	21
3.3.1.	Usuario Jugador	21
3.3.2.	Entrenador.	26
3.3.3.	Administrador	27
4.	Diseño.....	38
4.1.	Introducción	38
4.2.	Modelo	39
4.3.	Vista	39
4.4.	Controlador.....	40
4.5.	Base de datos	40



4.6.	Mapa web.....	44
4.7.	Mockup (diseño inicial)	45
5.	Detalles de implementación.....	54
5.1.	Tecnologías usadas	54
5.2.	Estructura de los ficheros y directorios	57
6.	Pruebas.....	60
6.1.	Validación	60
6.2.	Pruebas de uso	64
7.	Conclusiones	82
8.	Bibliografía.....	83

1. Introducción

Internet es una poderosa herramienta que permite a las personas conectarse con otras sin importar la distancia que las separa. Es por esto, que Internet se ha convertido en la principal herramienta para comunicarse con la familia, amigos o compañeros de trabajo ya sea mediante un ordenador o un teléfono móvil.

Dentro de este gigante llamado Internet, tenemos la World Wide Web o WWW, compuesta por numerosos sitios de todo tipo. Uno de los mayores usos que se le da a la World Wide Web es la utilización de redes sociales (más del 80% de la gente que tiene acceso a internet ha usado o usa Facebook). Es por esto que se ha optado a realizar una red social deportiva, y por motivos académicos también, se ha decidido a implementarla con el lenguaje de programación PHP, el cuál es utilizado por el framework Yii.

En la red social deportiva, el usuario tendrá la capacidad de compartir sus experiencias, retos, logros, fotografías, etc. a sus amigos o gente que comparta sus mismas aficiones, y además podrá ver los mismos de sus amigos o allegados, logrando así una mayor motivación para la realización de ejercicio (intentar ser el que corre más rápido, el que nada más kilómetros o el que hace más puntos en un partido de baloncesto).

Por motivos de complejidad y tiempo, se realizará sobre un solo deporte, el baloncesto.

2. Especificación de requisitos

2.1. Introducción

A continuación se procederá a nombrar los requisitos que debe cumplir la aplicación a desarrollar. Para poder especificarlos correctamente, se seguirán una serie de pasos como guía, buscando primero obtener los requisitos más generales que debe cumplir la aplicación, para luego detallar los mismos.

2.1.1. Propósito

El propósito de la especificación de requisitos de software (ERS) es servir como medio de comunicación entre clientes, usuarios, desarrolladores y demás participantes del proyecto, y especificar los requisitos que debe cumplir la aplicación, de tal forma que cubran las necesidades del cliente, y mostrar la propuesta de solución de los desarrolladores, todo esto en un lenguaje que sea de fácil comprensión para todos los participantes del proyecto.

2.1.2. Ámbito

Es mucha la gente que necesita de un impulso para realizar deporte diariamente, debido a que necesitan una motivación extra. Éste es el motivo por el cual se ha decidido realizar este proyecto, para acercar la práctica del deporte a la gran cantidad de personas que tienen acceso a internet, motivándolas a practicarlo con el hecho de poder compartir con sus conocidos las experiencias deportivas realizadas, así como de facilitar la comunicación entre personas que comparten un mismo hobby o deporte, de tal forma que podrían quedar para practicar juntos, y así sociabilizarse.

Este proyecto se podría realizar de todos los deportes en general, pero se va a focalizar en el baloncesto ya que es un deporte practicado por muchísima gente, ya sea por hobby o de forma federada, y que no dispone de un gran número de redes sociales, y menos que cumplan con estos objetivos.

Como juego de equipo que es, el baloncesto necesita de más de una persona para poder practicarlo, pero muchas veces no se encuentran suficientes participantes en tu círculo de contactos para poder practicarlo. En la red social que se va a implementar, cualquiera podrá ofertar un partido/entrenamiento a una fecha y hora concreta y cualquiera podrá unirse para jugar. Es decir, podrán inscribirse para ese evento en particular solucionando así el problema de la falta de jugadores e incentivando a la práctica del deporte y a la sociabilidad.

Otro gran problema es la comunicación de clubes y entrenadores con los jugadores. Habitualmente, un equipo no se inscribe por falta de plantilla, y muchos jugadores no pueden jugar al baloncesto por no encontrar un equipo acorde a sus necesidades. Es por

esto, que los entrenadores podrán contactar con los jugadores que se encuentren sin equipo y estén registrados, solucionando así un problema de comunicación existente en este aspecto.

2.1.3. Definiciones, Siglas y Abreviaturas

WWW: La World Wide Web, comúnmente llamado “Web” es una red mundial de documentos de hipertexto conectados a los que se puede acceder a través de internet. El acceso a éstos se realiza a través de navegadores web, que solicitan las páginas web a un servidor, y las muestran por pantalla.

Red Social: Es un sitio web donde los usuarios se registran, establecen vínculos con otros usuarios del mismo sitio (como amistades o “followers”), y comparten información personal y contenidos multimedia.

Yii: Es un framework *open-source* usado para crear aplicaciones PHP que implementa el patrón de arquitectura MVC (Model-View-Controller).

MVC: Son las siglas de Modelo-Vista-Controlador, y es un patrón de arquitectura de software que separa los proyectos en éstos tres componentes, lo cual aumenta en gran medida la mantenibilidad del proyecto.

2.1.4. Referencias

BERNERS-LEE, TIM; BRAY, TIM; CONOLLY, DAN; COTTON, PAUL; FIELDING, ROY; JECKLE, MARIO; LILLEY, CHRIS; MENDELSON, NOAH; ORCHARD, DAVID; WALSH, NORMAN; WILLIAMS, STUART (15 de diciembre de 2004). *Architecture of the World Wide Web, Volume One*. Versión 2004-12-15. W3C. <<https://www.w3.org/TR/webarch/>> [Consulta: el 08 de enero de 2016].

CODE PROJECT (2008). *Simple example of MVC (Model View Controller) Design Pattern for Abstraction*. <<http://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design>> [Consulta: el 08 de enero de 2016].

SCRIBD. *Redes sociales*. <<https://www.scribd.com/doc/24658747/Redes-sociales-definicion>> [Consulta: el 08 de enero de 2016].

“Understanding MVC architecture”. *Youtube* <<https://www.youtube.com/watch?v=eTdVkgFSlo>> [Consulta: el 08 de enero de 2016].

WORLD WIDE WEB CONSORTIUM (2001). *About the World Wide Web*. <<https://www.w3.org/WWW/>> [Consulta: el 08 de enero de 2016].

YII SOFTWARE LLC. *About Yii*. <http://www.yiiframework.com/about/> [Consulta: el 08 de enero de 2016].

2.1.5. Visión Global

A continuación se procederá a explicar más detalladamente la perspectiva del producto, así como sus funciones. También se detallaran los distintos tipos de usuarios y sus funcionalidades y restricciones en el producto a desarrollar, sin olvidar los supuestos y dependencias.

Finalmente se llevará a cabo el estudio de los requisitos específicos, donde se detallarán todas y cada una de las funcionalidades que integrará el producto, indicando los requisitos de la interfaz, los requisitos funcionales y de rendimiento, así como los de diseño, de bases de datos lógica y de software.

2.2. Descripción general

Los puntos contenidos en este apartado van a introducir los requisitos necesarios que debe tener la aplicación, como la funcionalidad y características de los usuarios, así como las restricciones y dependencias de la aplicación web.

2.2.1. Perspectiva del producto

La aplicación a desarrollar constará de una red social de carácter deportivo, centrada en el baloncesto para jugadores de nivel amateur (o como hobby) que quieran disfrutar del deporte sin la necesidad de pertenecer a un equipo. También servirá para los entrenadores de distintos clubs, que podrán convocar entrenamientos y ojear a los jugadores registrados.

En esta aplicación, los usuarios tan sólo con formar parte de esta red social, serán capaces de relacionarse con otros que compartan mismas aficiones, y de mantenerse comunicados vía esta.

2.2.2. Funciones del producto

Las funciones que será capaz nuestro producto de realizar son las que suelen caracterizar una red social, como las ya conocidas Facebook, LinkedIn, etc. pero incluyendo otras específicas que cumplen con el motivo de la realización de esta red social. Las funciones son las que siguen:

- Registrar Usuario: para el acceso a la red social y a sus funcionalidades los usuarios deberán registrarse en ella.
- Login/Logout: una vez registrados, los usuarios serán capaces de entrar en la aplicación mediante un acceso de usuario/contraseña.



- Editar perfil: el usuario podrá editar cierta información personal, como son el nombre, email, fecha de cumpleaños, su nivel de baloncesto (categoría de liga aproximada) y posición, así como el lugar de residencia y su avatar.
- Ver/Enviar/Recibir mensajes: esta es la principal función de comunicación de cualquier red social, la capacidad de enviar/recibir mensajes de su círculo de amigos en la red social.
- Añadir/Borrar amigos: otra de las funcionalidades que caracterizan una red social es la de poder añadir y borrar amigos de nuestro círculo de amigos.
- Subir/Borrar fotos: el usuario registrado será capaz de subir fotos de partidos, entrenamientos y demás.
- Buscar jugadores: un usuario tendrá la capacidad de hacer una búsqueda de jugadores según los criterios que desee filtrar.
- Crear partidos/entrenamientos: un usuario podrá crear entrenamientos y partidos amateur de 3 contra 3.
- Unirse a un partido/entrenamiento: cualquier usuario podrá unirse a un partido ya creado, o a un entrenamiento.
- Subir estadísticas: Los usuarios podrán subir las estadísticas del partido 3 contra 3 que hayan jugado.
- Crear grupo/Borrar grupo: los usuarios tendrán la capacidad de crear un grupo y de borrarlo.
- Unirse/Quitarse de un grupo: otra funcionalidad disponible será la de unirse o desvincularse de un grupo.
- Enviar/Leer mensajes de grupo: los usuarios que formen parte de un grupo serán capaces de enviar y leer los mensajes subidos por los participantes del mismo grupo.

2.2.3. Características de usuario

Para poder mencionar las características de usuario que tendrá el sitio web, estos se dividirán entre registrados y no registrados. Los usuarios registrados podrán poseer distintos roles: administrador, entrenador y jugador.

- Usuario no registrado. Un usuario que no ha sido registrado tan sólo podrá acceder a la página inicial de la aplicación para registrarse o ver los campos de “Sobre nosotros” y “Contacto”.

- Administrador. El administrador es el “superusuario” o *superuser*. Es el usuario que controla todo lo que pasa en la aplicación, ya que tiene permisos para realizar cualquier acción en la red social, ya sea expulsar un usuario, borrar comentarios inapropiados o cambiar el rol de otros usuarios.
- Aficionado/Jugador. El usuario jugador o usuario típico es el usuario que abundará en la red social. Estos usuarios podrán editar sus perfiles, añadir o borrar amigos, así como enviar/recibir mensajes a/de sus amigos. Además podrá ver los tableros de sus amigos si así lo desea. También será capaz de crear un grupo, añadirse a uno existente y enviar y recibir mensajes dentro de un grupo si pertenece a él. Un usuario jugador podrá crear y unirse a pachangas o partidos de 3 contra 3, buscar a otros jugadores existentes en la red social y subir fotografías en su red. También será capaz de subir sus estadísticas de un partido 3 contra 3 que haya jugado.
- Entrenador. El entrenador es un tipo de usuario registrado. Un usuario entrenador será un usuario como el usuario jugador, a diferencia que éste tendrá el privilegio de crear entrenamientos y aceptar a los usuarios que quieran unirse a uno.

2.2.4. Restricciones

Para poder acceder a la red social y a todos sus contenidos será imprescindible tener a disposición un ordenador con acceso a internet, que venga con un navegador web de los habituales (Internet Explorer, Mozilla Firefox, Google Chrome, Safari, Opera entre otros). Se recomienda que el monitor tenga una resolución de 1280x800 como mínimo para la correcta visualización del proyecto.

2.2.5. Supuestos y dependencias

La aplicación web se realiza con la aplicación XAMPP. Esta aplicación se basa en un servidor web Apache. Si se quisiera utilizar otro servidor, este deberá poder dar soporte a PHP, lenguaje de programación usado para el desarrollo del producto, así como integrar correctamente MySQL como base de datos.

2.3. Requisitos específicos

En este apartado se procederá a entrar en más detalle sobre los requisitos que ha de cumplir el sitio web de forma obligatoria, como los requisitos de interfaz y los requisitos funcionales.



2.3.1. Requisitos de la interfaz externa

Interfaz de usuario.

La página de inicio de la red social debe estar compuesta por una barra de navegación situada a la parte superior de la pantalla, que tendrá los accesos al *Log-in* y *Register*.

Una vez el usuario ha hecho el log-in, esta barra de navegación pasará a tener accesos rápidos a las principales funcionalidades del proyecto, como son: los apartados de un usuario, su perfil, su tablón de mensajes, lista de amistades y *logout*.

Además incorporará un *layout* que será común en todas las páginas de la aplicación. Este *layout* se dividirá en tres partes claramente diferenciadas:

- Una parte vertical o menú a la parte izquierda de la pantalla que contendrá información del usuario conectado y links a diferentes apartados del sitio web.
- Una parte central o *container*, la cual almacenará las vistas específicas de cada apartado del sitio web.
- Una parte vertical o de información a la derecha donde se mostrará información útil para el usuario, y/o espacio para publicidad.

Interfaz de hardware.

Cualquier ordenador con acceso a internet y que cumpla con los requisitos antes mencionados para disponer de un servidor web y de un navegador.

Interfaz de software.

Cualquier ordenador cuyo servidor tenga soporte del lenguaje PHP, en su versión mínima de PHP5, e integre el sistema de bases de datos MySQL, como el ya mencionado Apache. El navegador web utilizado debe poder interpretar el lenguaje de marcas y el de estilos, HTML y CSS respectivamente.

2.3.2. Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema y especifican la manera en que éste debe reaccionar a determinadas entradas, es decir, a situaciones particulares. Se llevarán a cabo desde los diferentes puntos de vista. Los generales son aquellos realizados por el usuario jugador, que es el usuario predominante en la aplicación. El usuario entrenador extenderá las funcionalidades del usuario jugador, y además añadirá alguna. Por último tenemos las funcionalidades del administrador, que tendrá la capacidad de realizar todas las existentes en la aplicación.

Cabe decir que, un usuario jugador no podrá realizar las funciones a las que no esté autorizado. Por ejemplo, un usuario jugador no puede ser capaz de crear, editar o borrar un entrenamiento, pero sí de unirse a él. Por el mismo motivo, un usuario entrenador no debe ser capaz de realizar las funciones que no esté autorizado, en este caso las funciones del administrador. Por lo tanto, no puede acceder al *Admin Pannel* ni a sus funcionalidades.

Generales.

Funcionalidad: Ingresar en la aplicación
Actor: Todos
1) El usuario debe ser capaz de acceder en la aplicación si está registrado.
2) Si los datos no son correctos no debe poder entrar a la aplicación.

Funcionalidad: Registrarse en la aplicación
Actor: Todos
1) El usuario debe ser capaz de registrarse en la aplicación.
2) Si los datos no son correctos no debe poder registrarse en la aplicación.

Funcionalidad: Editar perfil
Actor: Todos
1) El usuario debe poder entrar en su perfil para editar su información
2) Si los datos no son correctos el sistema debe impedir la edición.

Funcionalidad: Subir imagen
Actor: Todos
1) El usuario debe poder subir una imagen a la aplicación.
2) Si la extensión no está permitida el sistema debe impedir la subida del fichero.

Funcionalidad: Gestionar amistades
Actor: Todos
1) El usuario debe poder enviar peticiones de amistad.
2) El usuario debe poder aceptar peticiones de amistad
3) El usuario debe poder eliminar amistades.

Funcionalidad: Gestionar mensajes
Actor: Todos
1) El usuario será capaz de enviar mensajes.
2) El usuario será capaz de recibir mensajes.

Funcionalidad: Gestionar grupos
Actor: Todos
1) El usuario podrá crear un nuevo grupo.
2) El usuario podrá ingresar en un grupo existente.
3) El usuario podrá ver el listado de grupos.
4) El usuario podrá borrar el grupo si ha sido creado por él.

Funcionalidad: Gestionar mensajes en un grupo
Actor: Todos
1) El usuario podrá crear un mensaje en el grupo si pertenece a él.
2) El usuario podrá leer los mensajes de un grupo si pertenece a él.

Funcionalidad: Gestionar partidos 3contra3
Actor: Todos
1) El usuario podrá crear un nuevo partido de 3contra3.
2) El usuario será capaz de editar información sobre un partido.
3) El usuario podrá unirse a un partido 3contra3.
4) El sistema podrá ver los usuarios registrados en un partido.
5) El usuario podrá borrar un partido si lo ha creado él.

Funcionalidad: Unirse a un entrenamiento
Actor: Todos
1) El usuario podrá unirse a un nuevo entrenamiento.
2) El usuario será capaz de ver la información de un entrenamiento.

Usuario entrenador.

Funcionalidad: Gestionar entrenamientos
Actor: Entrenador
1) El usuario entrenador podrá crear un nuevo entrenamiento.
2) El entrenador será capaz de editar la información de un entrenamiento.
3) El entrenador podrá eliminar un entrenamiento.

Usuario administrador.

Funcionalidad: Crear entrenamiento
Actor: Entrenador
1) El usuario entrenador accede a la página de entrenamientos.
2) El entrenador crea un nuevo entrenamiento introduciendo los datos necesarios.
3) El sistema comprueba que los datos introducidos son correctos.
4) El sistema avisa al usuario entrenador que el entrenamiento se creó correctamente.

2.3.3. Otros requisitos.

No se presentan otros requisitos de importancia.

3. Análisis

El análisis en el desarrollo de una aplicación web es una parte fundamental para una implementación exitosa de la misma. Son los cimientos de la aplicación, y un análisis incorrecto puede traer muchos problemas durante la implementación. Es por esto que aquí se van a estudiar las clases o entidades que se van a necesitar, así como los casos de uso que tendrán lugar en el sitio web, y su modelado UML.

3.1. Diagrama de clases/entidades

El diagrama de clases es el resultado de representar las diferentes entidades que van a ser necesarias en el producto a desarrollar. Para esta aplicación, se van a necesitar las siguientes entidades o tablas.

- *User*. Es la entidad principal de la aplicación, ya que en ella se registran los distintos usuarios que se registran en la red social. Además, en esta tabla se almacenarán distintas características o información de su perfil.
- *Message*. Tiene la función de almacenar los mensajes enviados entre los usuarios. Los mensajes enviados entre usuarios se podrán ver en el tablón de mensajes de cada uno.
- *Friendship*. En la tabla friendship se pretenden almacenar las amistades, así como su estado y los usuarios que componen esta amistad.
- *Match3v3*. La tabla encargada de almacenar los datos e información de los partiditos o pachangas de 3 contra 3 creados.
- *Training*. Se trata de la tabla encargada de guardar la información de los entrenamientos convocados por un usuario entrenador. Esta tabla es muy similar a la tabla match3v3.
- *Stats*. Esta tabla almacenará las estadísticas obtenidas por un jugador en un partido.
- *Usergroup*. La tabla *usergroup* guardará los distintos grupos creados por los usuarios. Estos grupos se compondrán de distintos usuarios que compartan las mismas aficiones o cosas en común.
- *UserToUsergroup*. Dado que la cardinalidad de la relación entre las tablas *user* y *usergroup* es de muchos a muchos, se requiere de una tabla *UserToUsergroup* para poder representarla, ya que un usuario puede pertenecer a muchos grupos, y un grupo puede tener muchos usuarios. Esta tabla contendrá un campo id, y dos campos donde cada uno haga referencia a una de las tablas: *user* y *usergroup*.
- *Usergroupmessage*. Esta tabla almacenará los mensajes enviados dentro de un grupo.



Las cardinalidades que tendrán lugar en la base de datos y que se van a detallar a continuación se pueden ver en la figura siguiente (Ilustración 1).

- Un usuario puede tener muchas amistades o no tener ninguna, pero una amistad está compuesta por dos usuarios.
- Un usuario puede enviar muchos mensajes o no enviar ninguno, pero un mensaje únicamente puede pertenecer a un usuario.
- Un usuario puede haber participado en muchos o ningún partido, y un partido debe tener exactamente 6 jugadores.
- Un usuario por lo tanto, puede tener muchas o ninguna estadística, sin embargo, una estadística no puede no tener usuario.
- Un usuario puede unirse a tantos entrenamientos como quiera, pero un entrenamiento puede tener un máximo de 8 jugadores.
- Un usuario puede pertenecer a tantos grupos como quiera, y un grupo tener muchos usuarios.

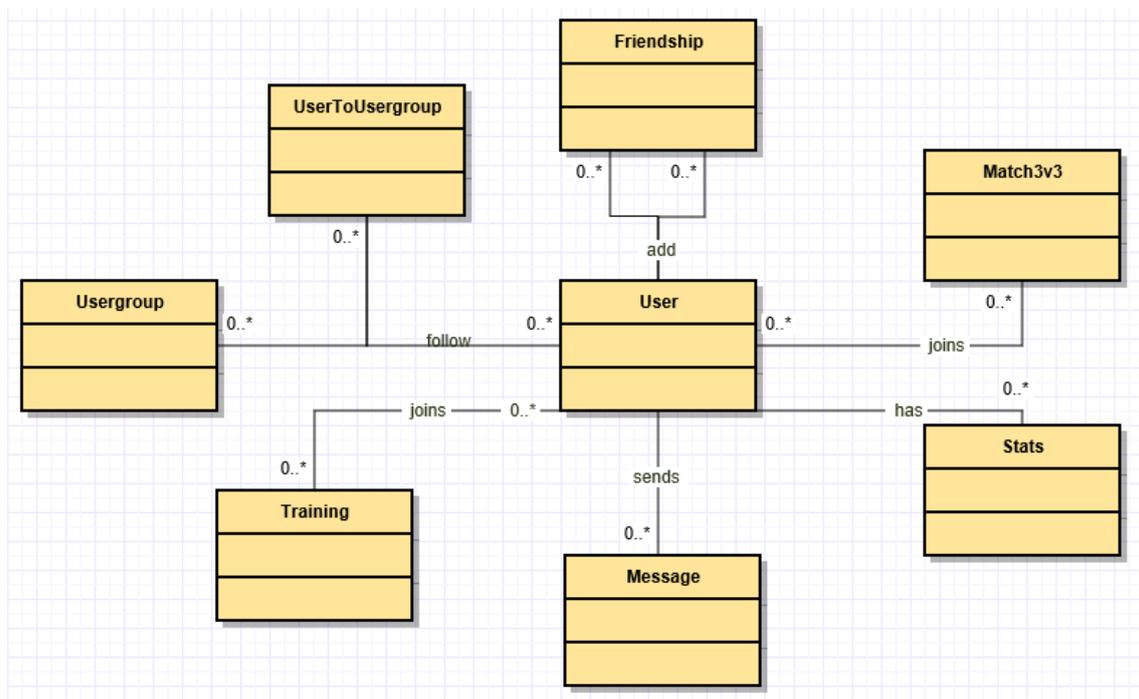


Ilustración 1. Diagrama de clases

3.2. Modelado UML.

El usuario jugador podrá realizar las siguientes acciones en la red social:

Usuario Jugador

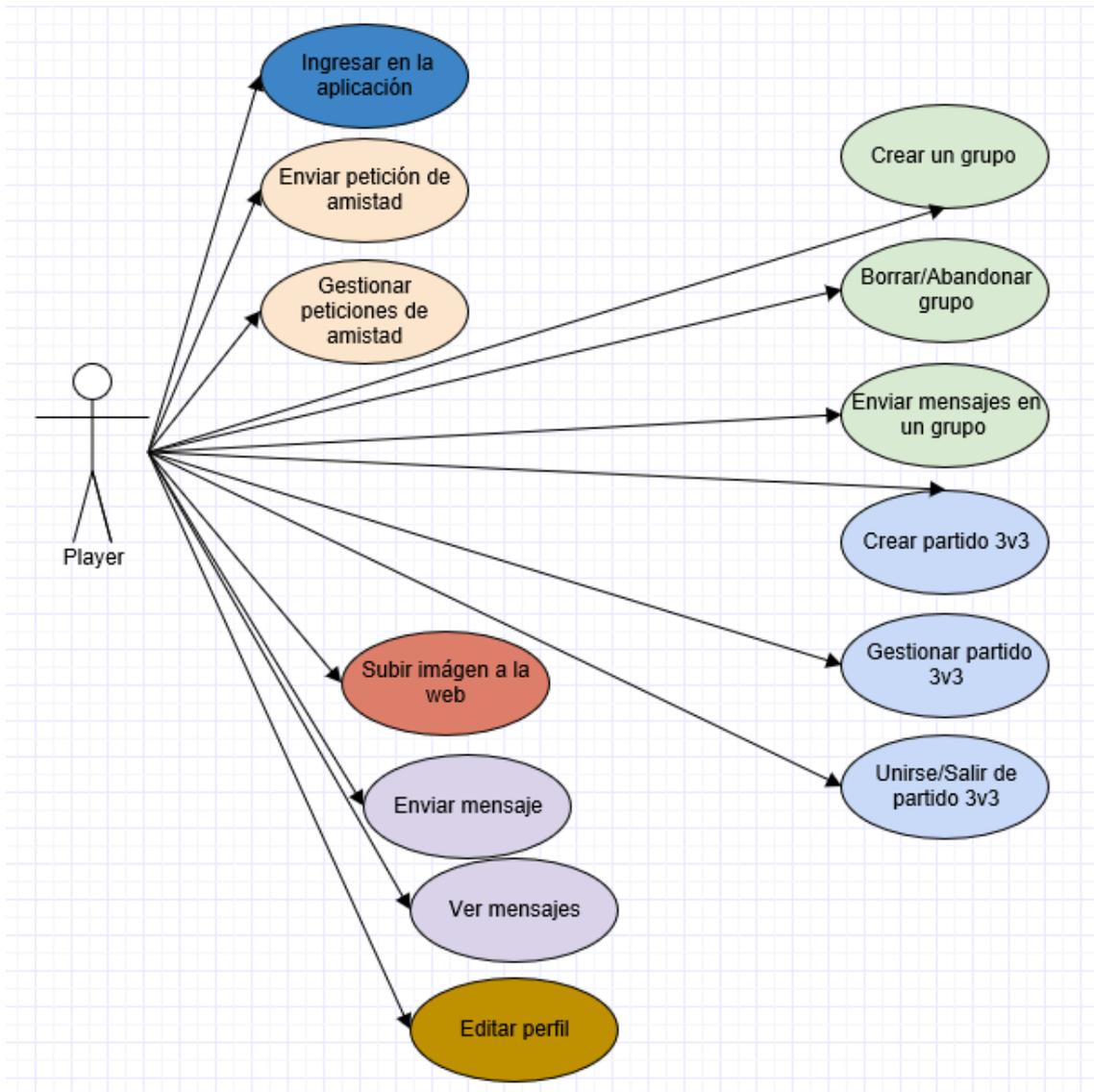


Figura 1. Casos de uso del usuario jugador

Usuario entrenador.

El usuario entrenador podrá realizar estas funciones y además heredará todos los casos de uso del usuario jugador.

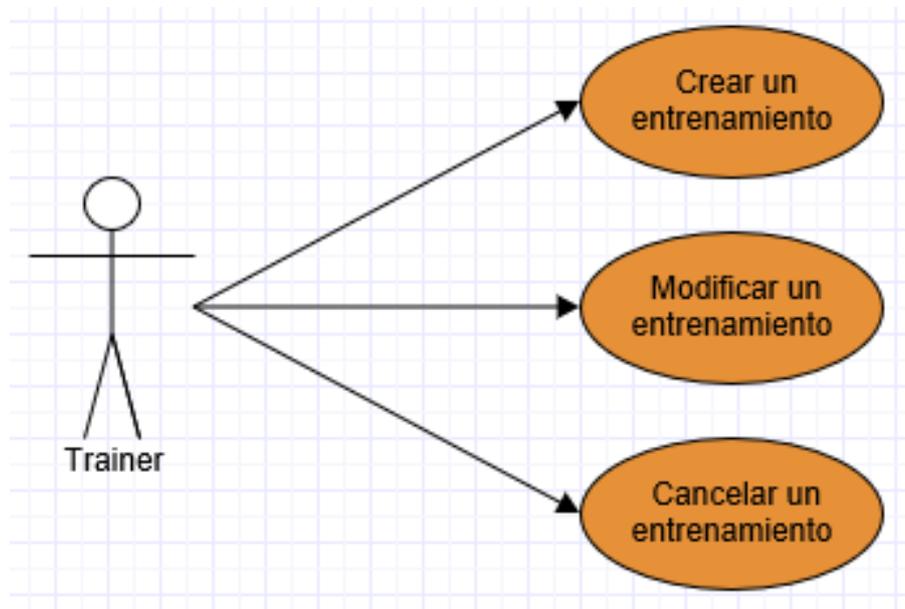


Figura 2. Casos de uso del usuario entrenador

Usuario administrador.

El usuario administrador podrá realizar todas las funciones. Heredará todos los casos de uso del usuario entrenador y jugador y además podrá gestionar los permisos de cada usuario.

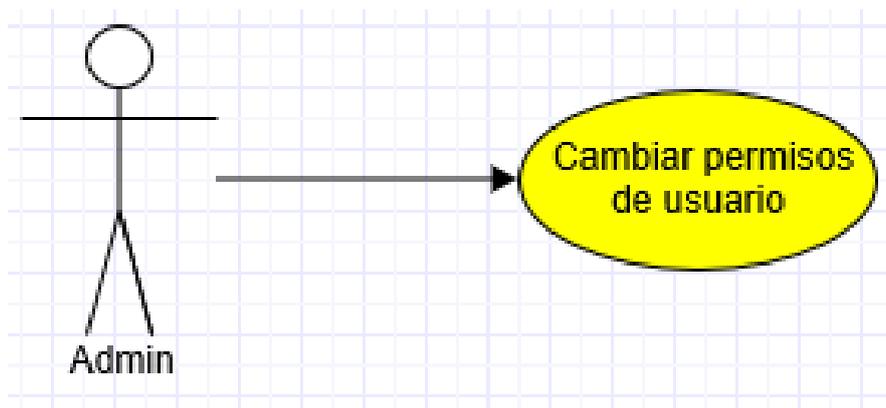


Figura 3. Casos de uso del usuario administrador

3.3. Casos de uso

Los casos de uso son una técnica necesaria para el desarrollo del producto ya que es utilizada para especificar el comportamiento del mismo. Es la secuencia de interacciones que ocurren entre un sistema y algo/alguien que usa sus servicios, normalmente los usuarios.

Se procederá a mostrar los casos de uso por cada tipo de usuario que puede existir en la aplicación.

Actores: Jugador, Entrenador, Administrador.

3.3.1. Usuario Jugador

Apartado	Descripción
Título	Log-in.
Propósito	Acceder a la aplicación mediante la identificación del usuario.
Entrada	Nombre de usuario (campo requerido). Contraseña (campo requerido).
Proceso	El usuario accede a la pantalla de log-in. El usuario introduce los datos requeridos. Validación de los campos requeridos. Validación del formato de los campos. Comprobación de los campos en la base de datos. Acceso a la aplicación.
Respuesta	Correcto: Mostrará por pantalla la página de perfil del usuario.

Apartado	Descripción
Título	Registro
Propósito	Registrarse en la base de datos como nuevo usuario
Entrada	Nombre (Campo requerido, único) Apellido (Campo requerido, único) Nombre de usuario (Campo requerido, único) Contraseña (Campo requerido) Email (Campo requerido) Fecha de nacimiento (Campo requerido) Posición (Campo requerido) Categoría (Campo requerido) Avatar
Proceso	El usuario accede a la pantalla de registro El usuario introduce los datos requeridos Validación de los campos requeridos Validación del correcto formato de los distintos campos Validación de campo único Registro de los datos en la base de datos



Respuesta	<p>Correcto: Mensaje al usuario confirmando registro</p> <p>Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio</p> <p>Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.</p> <p>Error dato duplicado: Mensaje al usuario indicando que el dato debe ser único, y el introducido ya existe.</p>
-----------	---

Apartado	Descripción
Título	Editar perfil del usuario
Propósito	Cambiar valores de campos del usuario en la base de datos
Entrada	<p>Nombre de usuario (Campo requerido, único)</p> <p>Contraseña (Campo requerido)</p> <p>E-mail (Campo requerido)</p> <p>Fecha de nacimiento (Campo requerido)</p> <p>Posición (Campo requerido)</p> <p>Categoría (Campo requerido)</p> <p>Avatar</p>
Proceso	<p>El usuario accede a la edición de su perfil</p> <p>El usuario introduce los datos</p> <p>El sistema valida los campos requeridos</p> <p>El sistema valida el correcto formato de los distintos campos</p> <p>El sistema valida los campos únicos</p> <p>Registro de los cambios en la base de datos</p>
Respuesta	<p>Correcto: Se mostrará una pantalla con los datos actualizados.</p> <p>Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio.</p> <p>Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.</p> <p>Error dato duplicado: Mensaje al usuario indicando que el dato debe ser único, y el introducido ya existe.</p>

Apartado	Descripción
Título	Enviar mensaje
Propósito	Insertar un nuevo mensaje en la base de datos
Entrada	<p>Título(Campo requerido)</p> <p>Mensaje(Campo requerido)</p>
Proceso	<p>El usuario introduce los datos requeridos</p> <p>Validación de los campos requeridos</p> <p>Validación del correcto formato de los distintos campos</p> <p>Registro del mensaje en la base de datos</p>
Respuesta	<p>Correcto: Mensaje al usuario confirmando el envío</p> <p>Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio.</p> <p>Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.</p>

Apartado	Descripción
Título	Ver mensajes recibidos por un usuario
Propósito	Ver todos los mensajes registrados en la base de datos de un usuario.
Entrada	Acceso a la página “My Wall” del usuario
Proceso	El usuario accede a la página “My Wall”. El sistema lee el parámetro id del usuario conectado. El sistema lee los mensajes registrados en la base de datos para ese id.
Respuesta	Correcto: Mensaje al usuario confirmando el envío Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Enviar petición de amistad
Propósito	Insertar una nueva amistad en la base de datos
Entrada	Nombre de usuario destinatario (campo requerido) Mensaje (campo requerido)
Proceso	El usuario envía una petición de amistad a otro usuario Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de la amistad en la base de datos
Respuesta	Correcto: Se mostrará la amistad por aceptar en la página del usuario. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Aceptar petición de amistad
Propósito	Modificar el valor del campo status de <i>Friendship</i> en la base de datos.
Entrada	
Proceso	El usuario invitado acepta petición de amistad Registro de la modificación de la amistad en la base de datos
Respuesta	Correcto: Se mostrará la amistad aceptada en la página del usuario. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Crear un nuevo grupo
Propósito	Insertar un nuevo grupo en la base de datos
Entrada	Nombre (campo requerido) Descripción(campo requerido)
Proceso	El usuario introduce los campos del nuevo grupo Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro del grupo en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la creación de un nuevo grupo. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Crear nuevo 3contra3 o Match3v3
Propósito	Insertar un nuevo partido 3contra3 en la base de datos
Entrada	Nombre (campo requerido) Fecha(campo requerido) Lugar(campo requerido) Jugador1 Jugador2 Jugador3 Jugador4 Jugador5 Jugador6
Proceso	El usuario introduce los campos para crear un nuevo partido Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro del 3contra3 en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la creación de un 3contra3. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Subir nueva fotografía o imagen
Propósito	Insertar una nueva fotografía en la aplicación
Entrada	Fichero con extensión .png, .jpg o .jpe
Proceso	El usuario selecciona una nueva imagen y la sube al sitio web Validación de la extensión del fichero Registro del fichero en el folder del usuario
Respuesta	Correcto: Mostrará por pantalla el listado de imágenes actualizada.

Apartado	Descripción
Título	Unirse a un partido
Propósito	Modificar campo “jugador X” para añadir un usuario al partido
Entrada	
Proceso	El usuario se une a un partido Registro de la modificación de los campos del partido en la base de datos.
Respuesta	Correcto: Mensaje al usuario confirmando la unión al partido.

Apartado	Descripción
Título	Unirse a un entrenamiento
Propósito	Modificar campo “jugador X” para añadir un usuario al entrenamiento.
Entrada	
Proceso	El usuario se une a un entrenamiento. Registro de la modificación de los campos del entrenamiento en la base de datos.
Respuesta	Correcto: Mensaje al usuario confirmando la unión al partido.

3.3.2. Entrenador.

El usuario entrenador heredará todos los casos de uso del usuario Jugador y además incorporará las siguientes interacciones:

Apartado	Descripción
Título	Crear nuevo entrenamiento
Propósito	Insertar un nuevo entrenamiento en la base de datos
Entrada	Nombre (campo requerido) Fecha(campo requerido) Lugar(campo requerido)
Proceso	El usuario accede a la pantalla de crear un entrenamiento El usuario introduce los datos requeridos Validación del rol de usuario Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro del entrenamiento en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la creación de un entrenamiento. Error rol de acceso. Mensaje al usuario indicando que no tiene permitido crear un entrenamiento. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Editar entrenamiento
Propósito	Modificar campos de un entrenamiento en la base de datos
Entrada	Nombre (campo requerido) Fecha(campo requerido) Lugar(campo requerido) Jugador1 Jugador2 Jugador3 Jugador4 Jugador5 Jugador6
Proceso	El usuario modifica los datos requeridos Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de la modificación de los campos del entrenamiento en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la modificación de un entrenamiento. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Ver los entrenamientos de un entrenador
Propósito	Ver todos los entrenamientos registrados en la base de datos por un entrenador.
Entrada	Acceso a la página “My trainings” de un entrenador.
Proceso	Acceso a la página “My trainings” de un entrenador. Leer el parámetro id del usuario conectado. Se mostrará el listado de los entrenamientos registrados en la base de datos por ese id.
Respuesta	Correcto: Se mostrará una pantalla con la lista de entrenamientos actualizada.

Apartado	Descripción
Título	Borrar entrenamiento
Propósito	Borrar entrenamiento de la base de datos
Entrada	ID del entrenamiento a eliminar
Proceso	El usuario accede al listado de entrenamientos Se mostrará el listado de los entrenamientos registrados Se eliminará el entrenamiento deseado si es suyo Mensaje de confirmación de borrado Borrado de los datos del entrenamiento en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con la lista de entrenamientos actualizada.

3.3.3. Administrador

El usuario administrador heredará todos los casos de uso del usuario Entrenador y además tendrá la potestad de realizar cambios sobre cualquier usuario, grupo, entrenamiento o partido. Para acceder a estas funcionalidades, el usuario administrador tendrá que acceder a su pantalla de “admin”.

Apartado	Descripción
Título	Insertar nuevo usuario
Propósito	Registrar en la base de datos un nuevo usuario
Entrada	Nombre (Campo requerido) Apellido (Campo requerido) Nombre de usuario (Campo requerido, único) Contraseña (Campo requerido) Email (Campo requerido) Fecha de nacimiento (Campo requerido) Posición (Campo requerido) Categoría (Campo requerido) Avatar

Proceso	El administrador accede a Insertar usuarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Validación de campo único Registro de los datos en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando registro. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo. Error dato duplicado: Mensaje al usuario indicando que el dato debe ser único, y el introducido ya existe.

Apartado	Descripción
Título	Editar usuario
Propósito	Cambiar valores de campos de un usuario en la base de datos
Entrada	Nombre (Campo requerido) Apellido (Campo requerido) Nombre de usuario (Campo requerido, único) Contraseña (Campo requerido) Email (Campo requerido) Fecha de nacimiento (Campo requerido) Posición (Campo requerido) Categoría (Campo requerido) Avatar
Proceso	El administrador accede a Editar usuarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Validación de campo único Registro de los cambios en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con los datos actualizados. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo. Error dato duplicado: Mensaje al usuario indicando que el dato debe ser único, y el introducido ya existe.

Apartado	Descripción
Título	Borrar usuario
Propósito	Borrar usuario de la base de datos
Entrada	ID del usuario a eliminar
Proceso	El administrador accede a Borrar usuarios Se mostrará el listado de usuarios registrados. Se introducirá el usuario a eliminar. Mensaje de confirmación de borrado

	Borrado de los datos del usuario en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con la lista de usuarios actualizado.

Apartado	Descripción
Título	Listado de los usuarios
Propósito	Listar usuarios registrados en el sistema
Entrada	
Proceso	Se mostrará el listado de usuarios registrados.
Respuesta	Correcto: Se mostrará una pantalla con la lista de usuarios actualizado.

Apartado	Descripción
Título	Crear nuevo mensaje
Propósito	Insertar un nuevo mensaje en la base de datos
Entrada	ID emisor (Campo requerido) ID receptor (Campo requerido) Titulo(Campo requerido) Mensaje(Campo requerido)
Proceso	El administrador accede a crear mensaje Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro del mensaje en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando el envío Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Editar mensaje
Propósito	Cambiar valores de campos de un mensaje en la base de datos
Entrada	ID emisor (campo requerido) ID receptor (campo requerido) Título (campo requerido) Mensaje (campo requerido)
Proceso	El administrador accede a editar mensaje Validación de los campos requeridos Validación del correcto formato de los distintos campos

	Registro de los cambios en los campos en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con los datos actualizados. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Borrar mensaje
Propósito	Borrar mensaje de la base de datos
Entrada	ID del mensaje a eliminar
Proceso	El administrador accede a mensajes Se mostrará el listado de los mensajes registrados Se introducirá el mensaje a eliminar Mensaje de confirmación de borrado Borrado de los datos del mensaje en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con la lista de mensajes actualizado.

Apartado	Descripción
Título	Listado de los mensajes
Propósito	Listar mensajes registrados en el sistema
Entrada	
Proceso	Se mostrará el listado de mensajes registrados.
Respuesta	Correcto: Se mostrará una pantalla con la lista de mensajes actualizado.

Apartado	Descripción
Título	Crear nueva amistad
Propósito	Insertar una nueva amistad en la base de datos
Entrada	ID emisor (Campo requerido) ID receptor (Campo requerido) Status(Campo requerido) Mensaje(Campo requerido)
Proceso	El administrador accede a <i>create friendship</i> Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de la amistad en la base de datos

Respuesta	<p>Correcto: Mensaje al usuario confirmando la nueva amistad</p> <p>Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio.</p> <p>Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.</p>
-----------	--

Apartado	Descripción
Título	Editar amistad
Propósito	Cambiar valores de campos de una amistad en la base de datos
Entrada	<p>ID emisor (campo requerido)</p> <p>ID receptor (campo requerido)</p> <p>Status (campo requerido)</p> <p>Mensaje (campo requerido)</p>
Proceso	<p>El administrador accede al listado de amistades</p> <p>Validación de los campos requeridos</p> <p>Validación del correcto formato de los distintos campos</p> <p>Registro de los cambios en los campos en la base de datos</p>
Respuesta	<p>Correcto: Se mostrará una pantalla con los datos actualizados.</p> <p>Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio.</p> <p>Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.</p>

Apartado	Descripción
Título	Borrar amistad
Propósito	Borrar amistad de la base de datos
Entrada	ID de la amistad a eliminar
Proceso	<p>El administrador accede a <i>friendship</i></p> <p>Se mostrará el listado de las amistades registradas.</p> <p>Se introducirá la amistad a eliminar.</p> <p>Mensaje de confirmación de borrado.</p> <p>Borrado de los datos de la amistad en la base de datos.</p>
Respuesta	Correcto: Se mostrará una pantalla con la lista de amistades actualizada.

Apartado	Descripción
Título	Listado de las amistades
Propósito	Listar amistades registradas en el sistema
Entrada	
Proceso	<p>El administrador accede a <i>friendship</i>.</p> <p>Se mostrará el listado de amistades registradas.</p>
Respuesta	Correcto: Se mostrará una pantalla con la lista de amistades actualizada.

Apartado	Descripción
Título	Crear nuevo grupo
Propósito	Insertar un nuevo grupo en la base de datos
Entrada	ID creador (campo requerido) Nombre (campo requerido) Descripción(campo requerido) Participantes
Proceso	El administrador accede a <i>create usergroup</i> Introduce los campos necesarios para crear un nuevo grupo Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de la amistad en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la nueva amistad. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Editar grupo
Propósito	Cambiar valores de campos de un grupo en la base de datos
Entrada	ID creador (campo requerido) Nombre(campo requerido) Descripción (campo requerido) Participantes
Proceso	El administrador accede a <i>usergroup</i> Modifica los campos necesarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de los cambios en los campos en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con los datos actualizados. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Borrar grupo
Propósito	Borrar grupo de la base de datos
Entrada	ID del grupo a eliminar
Proceso	El administrador accede a <i>usergroup</i> . Se introducirá el grupo a eliminar. Mensaje de confirmación de borrado. Borrado de los datos del grupo en la base de datos.
Respuesta	Correcto: Se mostrará una pantalla con la lista de amistades actualizada.

Apartado	Descripción
Título	Listado de grupos
Propósito	Listar grupos registrados en el sistema
Entrada	
Proceso	El administrador accede a <i>usergroup</i> . Se mostrará el listado de grupos registrados.
Respuesta	Correcto: Se mostrará una pantalla con la lista de grupos actualizado.

Apartado	Descripción
Título	Crear nueva estadística
Propósito	Insertar una nueva estadística en la base de datos
Entrada	ID 3contra3 (campo requerido) ID jugador (campo requerido) Puntos Rebotes Asistencias Tapones Robos Tiros anotados Tiros intentados Pérdidas
Proceso	El administrador accede a <i>stats</i> . Introduce los campos necesarios para crear un nuevo <i>stats</i> . Validación de los campos requeridos. Validación del correcto formato de los distintos campos. Registro de las estadísticas en la base de datos.
Respuesta	Correcto: Mensaje al usuario confirmando la nueva amistad. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Editar estadística
Propósito	Cambiar valores de campos de una estadística en la base de datos
Entrada	ID 3contra3 (campo requerido) ID jugador (campo requerido) Puntos Rebotes Asistencias Tapones

	Robos Tiros anotados Tiros intentados Pérdidas
Proceso	El administrador accede a <i>stats</i> Modifica los campos necesarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de los cambios en los campos en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con los datos actualizados. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Borrar estadística
Propósito	Borrar estadística de la base de datos
Entrada	ID de la estadística a eliminar
Proceso	El administrador accede a su página <i>stats</i> . Se mostrará el listado de las estadísticas registradas. Se introducirá la estadística a eliminar. Mensaje de confirmación de borrado Borrado de los datos del grupo en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con la lista de estadísticas actualizada.

Apartado	Descripción
Título	Listar estadísticas
Propósito	Mostrar todas las estadísticas almacenadas en la base de datos
Entrada	
Proceso	El administrador accede a su página <i>stats</i> Se mostrará el listado de las estadísticas registradas.
Respuesta	Correcto: Se mostrará una pantalla con la lista de estadísticas actualizada.

Apartado	Descripción
Título	Crear nuevo 3contra3
Propósito	Insertar un nuevo partido 3contra3 en la base de datos
Entrada	Nombre (campo requerido) Fecha(campo requerido) Lugar(campo requerido) Jugador1 Jugador2 Jugador3

	Jugador4 Jugador5 Jugador6
Proceso	El administrador entra a su página match3v3 El administrador introduce los datos necesarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro del 3contra3 en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la creación de un 3contra3. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Editar 3contra3
Propósito	Modificar campos de un partido 3contra3 en la base de datos
Entrada	Nombre (campo requerido) Fecha(campo requerido) Lugar(campo requerido) Jugador1 Jugador2 Jugador3 Jugador4 Jugador5 Jugador6
Proceso	El administrador entra a su página match3v3 El administrador modifica los datos necesarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de la modificación de los campos del 3contra3 en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la modificación de un 3contra3. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Borrar 3contra3
Propósito	Borrar partido 3contra3 de la base de datos
Entrada	ID del partido a eliminar

Proceso	El administrador entra a su página match3v3 Se mostrará el listado de los partidos registrados. Se introducirá el partido a eliminar. Mensaje de confirmación de borrado Borrado de los datos del partido en la base de datos
Respuesta	Correcto: Se mostrará una pantalla con la lista de partidos actualizada.

Apartado	Descripción
Título	Listar partidos 3contra3
Propósito	Mostrar todas los partidos 3contra3 almacenados en la base de datos
Entrada	
Proceso	El administrador entra a su página match3v3 Se mostrará el listado de los partidos registrados.
Respuesta	Correcto: Se mostrará una pantalla con la lista de partidos 3contra3 actualizada.

Apartado	Descripción
Título	Crear nuevo entrenamiento
Propósito	Insertar un nuevo entrenamiento en la base de datos
Entrada	Nombre (campo requerido) Fecha(campo requerido) Lugar(campo requerido) ID Entrenador (campo requerido) Participantes
Proceso	El administrador accede a su página <i>trainings</i> El administrador introduce los datos necesarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro del entrenamiento en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la creación de un entrenamiento. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Editar entrenamiento
Propósito	Modificar campos de un entrenamiento en la base de datos
Entrada	Nombre (campo requerido)

	Fecha(campo requerido) Lugar(campo requerido) Jugador1 Jugador2 Jugador3 Jugador4 Jugador5 Jugador6
Proceso	El administrador accede a su página <i>trainings</i> El administrador modifica los datos necesarios Validación de los campos requeridos Validación del correcto formato de los distintos campos Registro de la modificación de los campos del entrenamiento en la base de datos
Respuesta	Correcto: Mensaje al usuario confirmando la modificación de un entrenamiento. Error campo requerido: Mensaje al usuario indicando que falta un campo obligatorio. Error formato del campo: Mensaje al usuario indicando un formato no válido introducido en un campo.

Apartado	Descripción
Título	Borrar entrenamiento
Propósito	Borrar entrenamiento de la base de datos
Entrada	ID del entrenamiento a eliminar
Proceso	El administrador accede a su página <i>trainings</i> Se mostrará el listado de los entrenamientos registrados. Se introducirá el entrenamiento a eliminar. Mensaje de confirmación de borrado. Borrado de los datos del entrenamiento en la base de datos.
Respuesta	Correcto: Se mostrará una pantalla con la lista de entrenamientos actualizada.

Apartado	Descripción
Título	Listar entrenamientos
Propósito	Mostrar todos los entrenamientos almacenados en la base de datos
Entrada	
Proceso	El administrador accede a su página <i>trainings</i> . Se mostrará el listado de entrenamientos registrados.
Respuesta	Correcto: Se mostrará una pantalla con la lista de entrenamientos actualizada.

4. Diseño

El siguiente paso consiste en realizar un diseño inicial para poder empezar a implementar la red social. Se va a proceder a explicar el patrón seguido (Modelo-Vista-Controlador), la base de datos utilizada y el *mockup* o plantilla del aspecto que deben tener las páginas en el sitio web.

4.1. Introducción

Hace unos años, para desarrollar sitios web con PHP sin la posibilidad de un *framework*, se seguía la idea de crear un archivo PHP por cada página que iba a tener tu sitio. Estos archivos PHP debían contener la inicialización y configuración global, la lógica de negocio, la obtención de los registros de la base de datos y el código PHP encargado de generar la página.

El resultado de esto eran archivos con muchísimo código, y por lo tanto, muy difícil de mantener, haciendo considerablemente costoso modificar o añadir nuevas características y/o funcionalidades al sitio.

Es por este motivo que surgió una solución (introducida por Trygve Reenskaug en Smalltalk), con la finalidad de estructurar el código según su función, facilitando mucho la modificación del sitio. El patrón de diseño se llama Model-View-Controller o MVC, y separa el código en tres capas (imagen inferior):

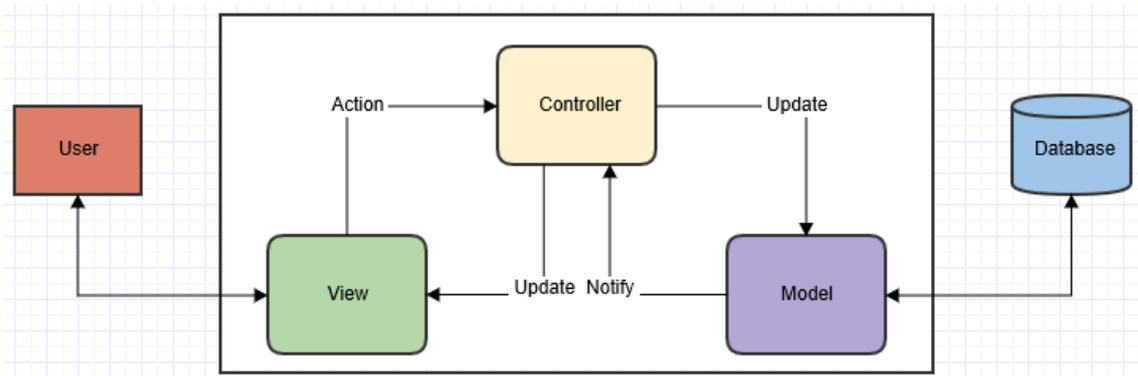


Figura 4. Diagrama basado en www.patricksoftwareblog.com

En este tipo de patrón de diseño, la mayor parte del peso recae sobre el modelo. De esta forma, podemos reutilizar la lógica cuando implementamos una nueva vista. Tan solo tenemos que decirle al controlador que renderice una nueva vista pasándole el modelo conveniente. Idealmente, se debe intentar conseguir acercarse lo máximo posible al significado de la famosa frase *"We need SMART Models, THIN Controllers, and DUMB Views"*.

4.2. Modelo

El modelo encapsula la lógica de negocio de una aplicación y éste debe ser independiente, es decir, el modelo no puede mantener variables directas que refieran a la vista o al controlador. En el modelo se representa la información con la que opera el sistema, ya sean consultas como actualizaciones de la base de datos. Se encarga de enviar a la vista la información solicitada para ser mostrada a través del controlador. Las peticiones al modelo también llegan a través del controlador.

En Yii2, el modelo es usado para mantener los datos y sus reglas de negocio. Un modelo representa un solo objeto de datos, ya sea una tabla de la base de datos, una fila o un formulario con campos para que los usuarios introduzcan datos. Cada uno de los campos del objeto es representado por un atributo del modelo.

Los modelos en Yii2 se encuentran bajo la carpeta *app/models* y sus clases se llaman normalmente como la entidad a la que hacen referencia, empezando la primera letra en mayúsculas. Por ejemplo: *app/models/Message.php*.

Yii2 implementa dos tipos de modelos, los modelos de formularios y los Active Record (AR). Los modelos de formularios son usados para almacenar los datos introducidos por el usuario, por ejemplo, una página de *log-in*. Por otra parte, tenemos los AR que usan la popular técnica *Object-Relational Mapping (ORM)*. Cada clase AR representa una tabla de la base de datos y sus atributos son representados como propiedades en la clase AR. AR es de gran ayuda para acceder a la base de datos relacional de forma simple y práctica.

4.3. Vista

La vista obtiene los datos del modelo y los muestra al usuario, es decir, es la salida por pantalla de la aplicación. Estos datos los obtiene mediante métodos de consulta (comúnmente llamados *getters*) implementados en el modelo. Es importante recalcar que una vista no debería nunca modificar los datos obtenidos por el modelo, sólo debería leerlos.

En el *framework* Yii2, las vistas son scripts o documentos PHP que contienen sentencias HTML y PHP. Estos ficheros deben ser lo más ligeros posibles, por lo tanto, un buen uso es que no deben contener grandes cantidades de lógica. Las vistas son renderizadas desde el controlador, y es este el que le pasa los atributos que la vista vaya a necesitar.

En Yii2, también hay un tipo especial de vista llamado *Layout*, que es una vista especial encargada de decorar las demás vistas. Las vistas en Yii2 se encuentran en la carpeta *app/views/controllerName/nombreDeVista.php*.



4.4. Controlador

El controlador es el responsable de llamar a los métodos del modelo que cambian su estado, como escrituras y modificaciones de las tablas, y también se encarga de renderizar una vista determinada pasándole unos atributos. Es por esto por lo que se le conoce como el pegamento de la aplicación, porque básicamente se encarga de unir los modelos y las vistas.

En Yii2, los controladores se ejecutan cuando son llamados por la URL o por una acción. Los controladores tienen métodos llamados *action*, y son los que empiezan la acción determinada. Por ejemplo, si una URL llama al controlador *PostController* y a la acción *create* tal que la URL quedara así: <http://myServer/myApp/Post/create>, se ejecutará la acción *actionCreate* de *PostController*.

Los archivos PHP de los controladores los podemos encontrar dentro de la carpeta `app/controllers` de nuestro proyecto.

4.5. Base de datos

A continuación, se va a mostrar una imagen de la estructura de la base de datos utilizada, así como a explicar cada entidad y su contenido y las relaciones existentes entre las distintas clases.

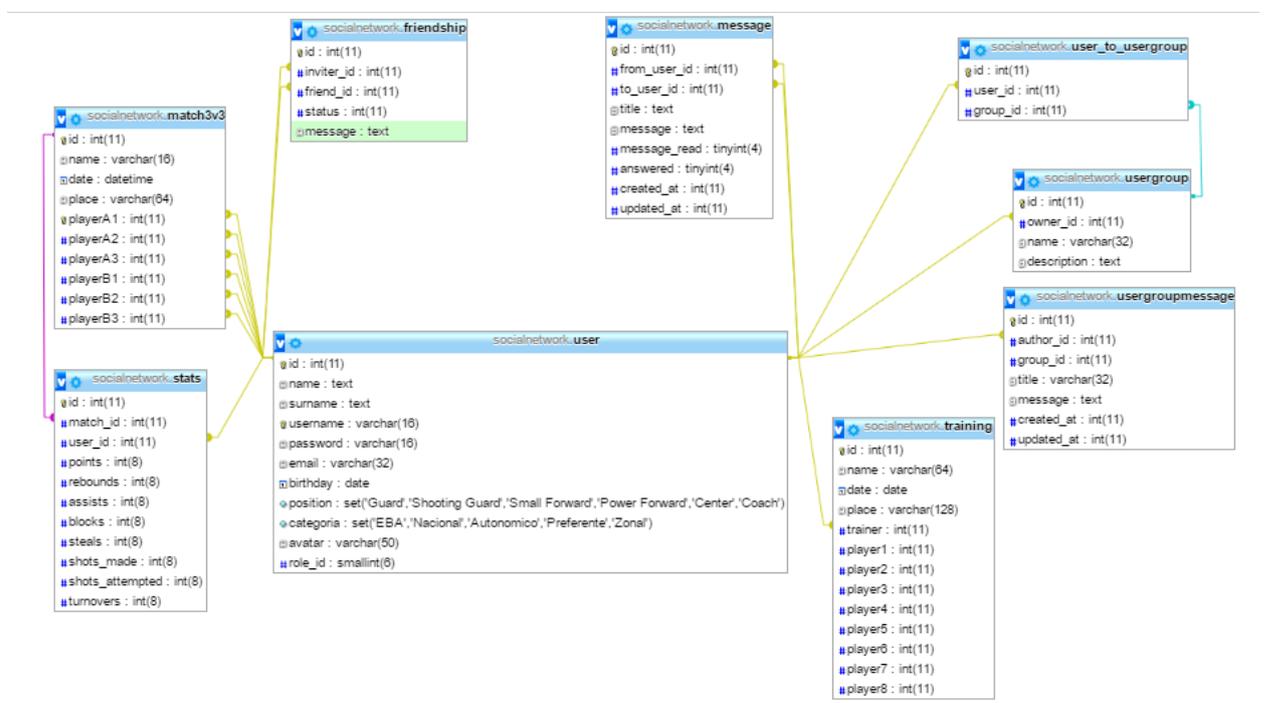


Figure 5. Base de datos MySQL

User.

- **id.** Es el campo identificador del usuario. Se trata de una variable de tipo *Integer* y su valor se escribe por *autoincrement*, esto es, cada usuario nuevo registrado tendrá el valor del id del usuario anterior registrado más una unidad. Es también la clave primaria de la tabla *user* (PK).
- **name.** Tal y como se deduce, este campo almacena el nombre del usuario, de tipo *String*.
- **surname.** Al igual que el campo *name*, almacena el apellido del usuario con un *String*.
- **username.** Es el campo usado para almacenar el nombre de usuario. Este será el nombre que usaremos para poder entrar en la aplicación. Es también una cadena de caracteres con longitud máxima de 16 caracteres.
- **password.** Almacena la contraseña del usuario. Es la contraseña usada para validar la autenticidad de un usuario.
- **email.** Guarda la dirección de correo electrónico del usuario. Será mostrado a otros usuarios para facilitar el contacto con éste. Al igual que *username* y *password* se guarda en un campo de tipo *varchar*.
- **birthday.** En este campo se registra la fecha de nacimiento del usuario. Sirve para poder filtrar por edades en la búsqueda de jugadores. Es un campo de tipo fecha.
- **position.** Guarda la posición natural de un jugador, que sólo puede ser un valor entre los siguientes: *Guard, Shooting Guard, Small Forward, Power Forward, Center y Coach*. Sirve para poder filtrar por posiciones en la búsqueda de jugadores.
- **categoría.** En este campo sólo se puede almacenar un valor entre los siguientes: EBA, Nacional, Autonómico, Preferente y Zonal. También sirve para poder filtrar por nivel de liga a los jugadores.
- **avatar.** Se trata de un campo de tipo *String* que almacenará el nombre de la imagen subida. Su uso es el típico de un avatar en cualquier red social.
- **role_id:** Por último, el campo *role_id* almacena un valor que puede ser 10, 20 o 30. Estos valores son variables estáticas que hacen referencia al tipo de rol del usuario, *admin, user* y *trainer* respectivamente.

Message.

- **id.** Campo identificador del mensaje. Como todos los id que hay en esta base de datos, son de tipo *Integer*, su valor se autoincrementa cuando se crea y son la clave primaria de la tabla (PK).
- **from_user_id.** Se trata de un campo que almacena el id del usuario que envía el mensaje. Éste campo tiene una restricción de clave foránea con el campo *user.id*.
- **to_user_id.** Igual que el anterior, pero almacena el id del usuario que recibe el mensaje. También tiene una restricción de clave foránea con el campo *user.id*.
- **title.** Es un campo de tipo *String* que almacena el título que el usuario da al mensaje que quiere enviar.
- **message.** Exactamente igual que el título, pero almacena el contenido del mensaje que se envía.



- **message_read.** Podrá almacenar dos valores, 0 y 1, haciendo la función del boolean, $0 \rightarrow false$ y $1 \rightarrow true$, dependiendo de si el mensaje ha sido leído o no (no implementado).
- **message_answered.** Del mismo tipo que *message_read*, nos dará la información de si el mensaje ha sido respondido o no (no implementado).
- **created_at.** En este campo se almacenará el *timestamp* del mensaje. Esto es, cuando ha sido creado. Se trata de un campo de tipo *Integer* que almacena los segundos que han pasado desde el 1 de Enero de 1970.
- **updated_at.** Similar a *created_at* pero almacena el momento en el que el mensaje ha sido editado.

Friendship.

- **id.** Mismo tipo de campo que en las otras tablas. También es la llave primaria de esta tabla (PK).
- **inviter_id.** Almacena el identificador del usuario que envía la petición de amistad. Tiene una restricción de clave foránea con el campo *user.id*.
- **friend_id.** Guarda el identificador del usuario que recibe la petición de amistad. Al igual que *inviter_id*, tiene una restricción de clave foránea con el campo *user.id*.
- **status.** Es un campo de tipo *Integer* que tiene la función de guardar el estado de la amistad, esto es, si ha sido aceptada o sigue en trámite. Cuando una nueva amistad es creada, este campo adquiere el valor por defecto “0”, es decir, petición en trámite. Cuando el usuario que recibe la petición la acepta, este campo cambia a “1”.
- **message.** El uso de este campo es para poder enviar un pequeño mensaje junto a la petición de amistad.

Match3v3.

- **id.** Como en las tablas anteriores. (PK)
- **name.** Es el campo encargado de almacenar un nombre para el partido. Este nombre lo pondrá el usuario que cree el partido. Campo de tipo *String*.
- **date.** Guardará la fecha en un campo de tipo *Date* en la que tendrá lugar el partido.
- **place.** Campo de tipo *String* que contendrá la dirección donde se realizará el partido.
- **playerA1, playerA2, playerA3.** Se trata de los campos que almacenarán los identificadores de los jugadores de un equipo. Cada uno de estos campos, de tipo *Integer*, tendrá una restricción de clave foránea al campo *user.id*.
- **playerB1, playerB2, playerB3.** Exactamente igual que los campos anteriores pero para el equipo rival.

Stats.

- **id.** Como en las tablas anteriores (PK).
- **user_id.** Contiene el identificador del usuario al que pertenecen las estadísticas. Tiene una restricción de clave foránea con *user.id*.

- **match_id.** Contiene el identificador del partido en el que se han conseguido las estadísticas. Tiene una restricción de clave foránea con *match.id*.
- **points, rebounds, assists, blocks, steals, shots_attempted, shots_made, turnovers.** Se trata de campos de tipo *int* que guardarán el número de puntos, rebotes, asistencias, tapones, robos, tiros intentados y anotados y perdidas respectivamente.

Training.

- **id.** Misma utilidad que en los otros campos id de las otras tablas.
- **name, date, place.** Campos muy similares a los de mismo nombre de la tabla *match3v3* pero que almacenará el nombre, fecha y dirección del entrenamiento convocado.
- **trainer.** Este campo contiene el identificador del usuario entrenador que ha convocado el entrenamiento. Este campo tiene una restricción de clave foránea con el campo id de la tabla *user* (*user.id*).
- **player1, player2, player3, player4, player5, player6, player7, player8.** Estos campos contienen los identificadores de los usuarios jugadores que se apuntan al entrenamiento.

Usergroup.

- **id.** Igual que en tablas anteriores.
- **owner_id.** Almacena el identificador del usuario que ha creado el grupo.
- **name.** Este campo guarda una cadena de caracteres que darán nombre al grupo creado.
- **description.** El campo *description* almacenará una breve descripción del grupo. Como por ejemplo, describiendo el tipo de grupo que será, que aficiones se comparten en ese grupo, equipos afines, etc.

Usergroupmessage.

- **id.** Igual que en tablas anteriores.
- **author_id.** Guarda el identificador del usuario que ha enviado el mensaje. Tiene una restricción de clave foránea con el campo *user.id*.
- **group_id.** Guarda el identificador del grupo al que se envía el mensaje. Tiene una restricción de clave foránea con el campo *usergroup.id*.
- **title, message.** Campos de tipo *String* que almacenan respectivamente el título y el mensaje que el usuario quiere enviar dentro de un grupo.
- **created_at, updated_at.** En este campo se almacenará el *timestamp* del mensaje cuando ha sido creado o editado. Se trata de un campo de tipo *Integer* que almacena los segundos que han pasado desde el 1 de Enero de 1970.

User to usergroup.

- **id.** Igual que en tablas anteriores.
- **user_id.** Campo identificador del usuario. Tiene una restricción de clave foránea con el campo id de la tabla *user*.



- **Group_id.** Campo identificador del grupo. Tiene una restricción de clave foránea con el campo *id* de la tabla *usergroup*.

4.6. Mapa web

En la siguiente figura podemos ver como se distribuyen los enlaces y las páginas en la red social:

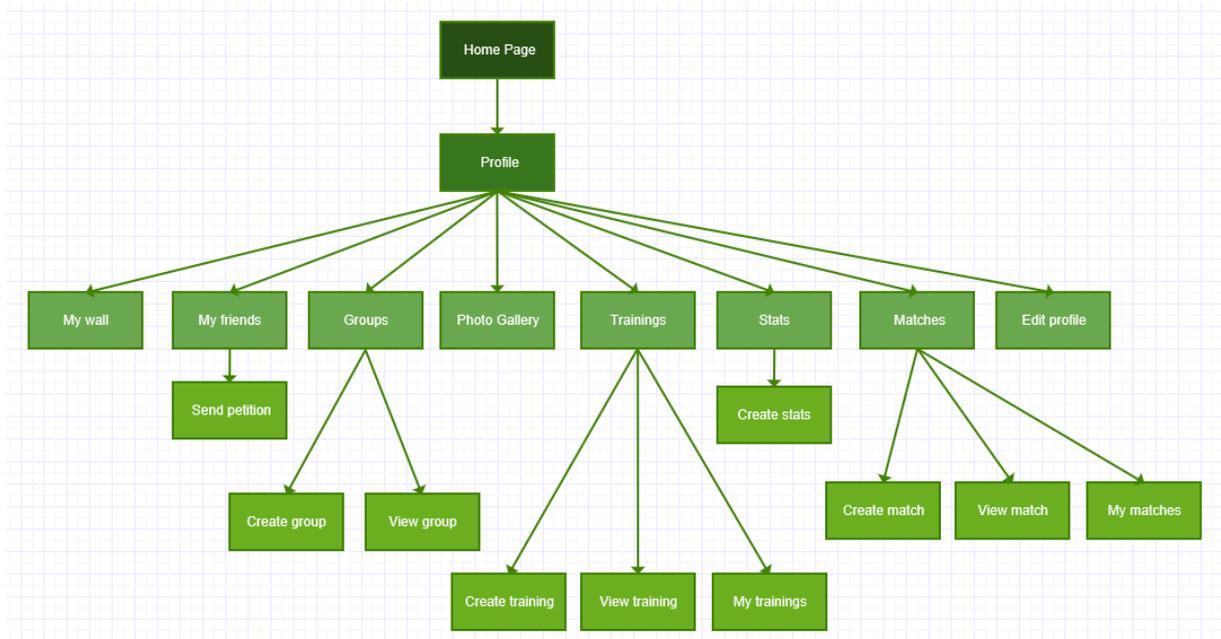
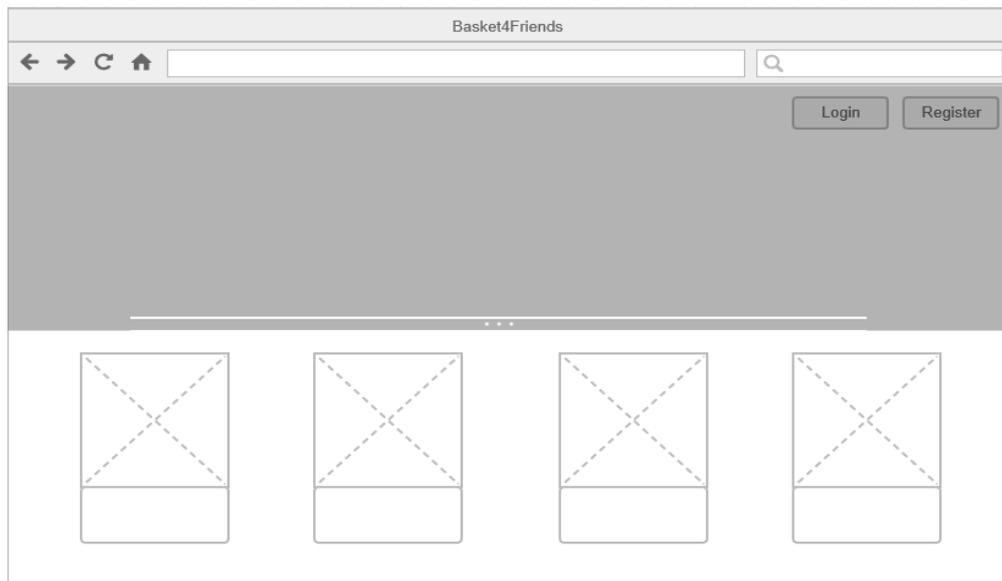


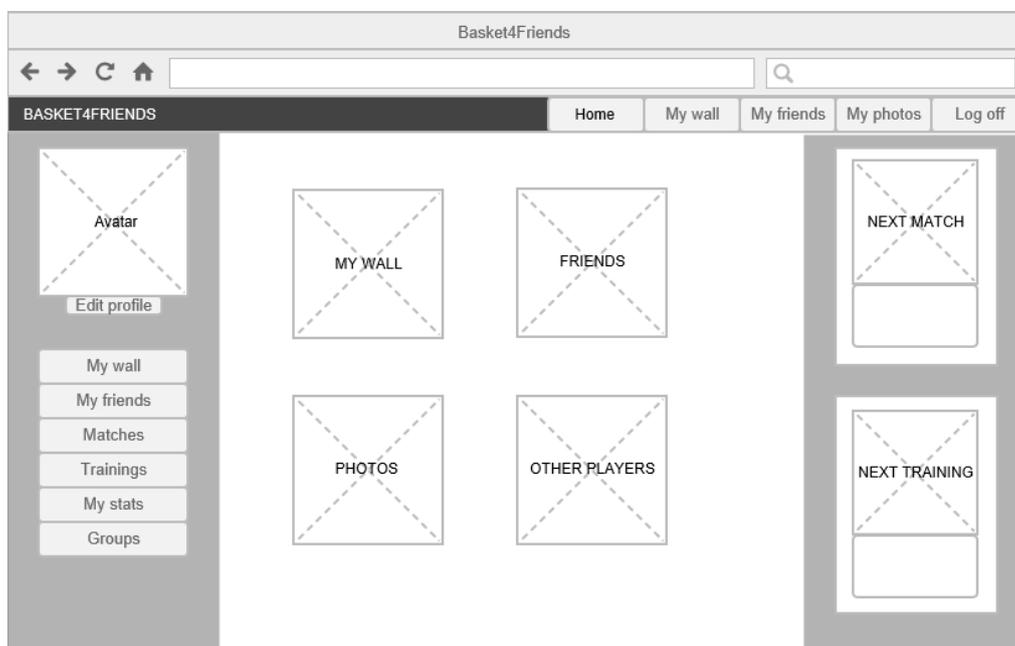
Figura 6. Sitemap

El usuario, al hacer log-in, accederá a su página principal, la cuál será *Profile*, donde tendrá accesos directos a las principales funcionalidades de la red social, como son, ver el muro, los amigos, grupos, fotografías, etc. de uno mismo. Dentro de cada una de estas funcionalidades, se podrá acceder al resto mediante otro acceso directo o *button*.

4.7. Mockup (diseño inicial)



1. Página de inicio.



2. Profile.

Basket4Friends

← → ↻ 🏠 🔍

Edit profile

Name

Surname

Username

Password

Email

Birthday / / 📅

Position ▼

Category ▼

3. Edit profile.

Basket4Friends

← → ↻ 🏠 🔍

BASKET4FRIENDS Home My wall My friends My photos Log off

Edit profile

My wall

My friends

Matches

Trainings

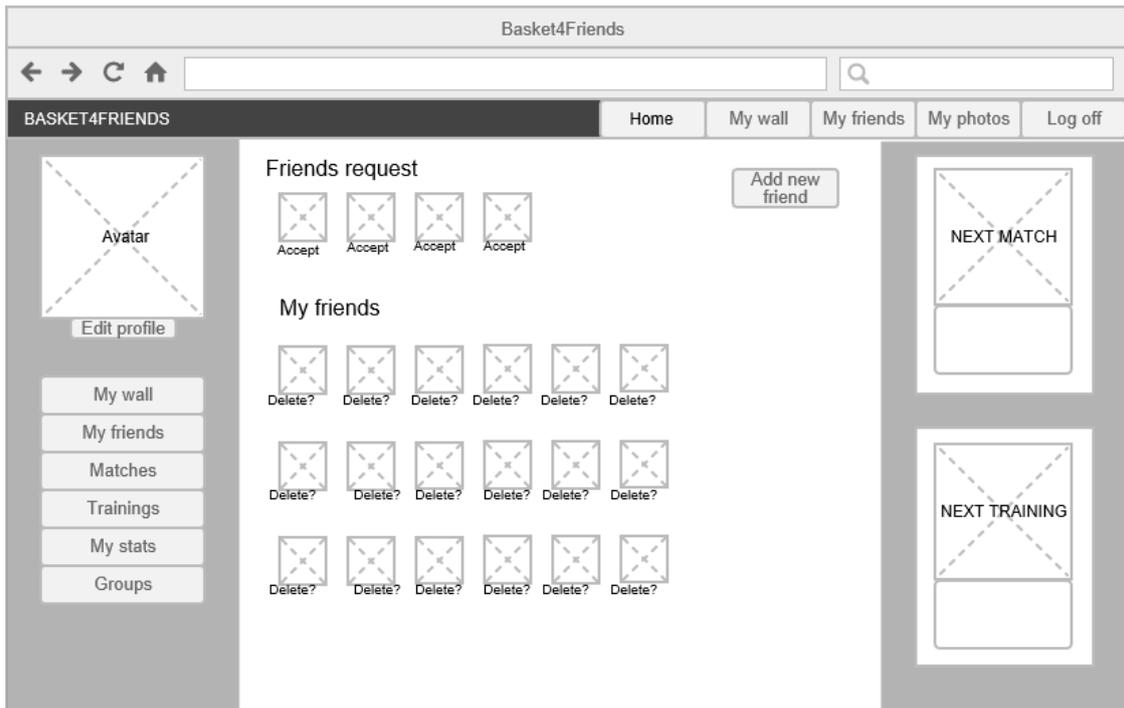
My stats

Groups

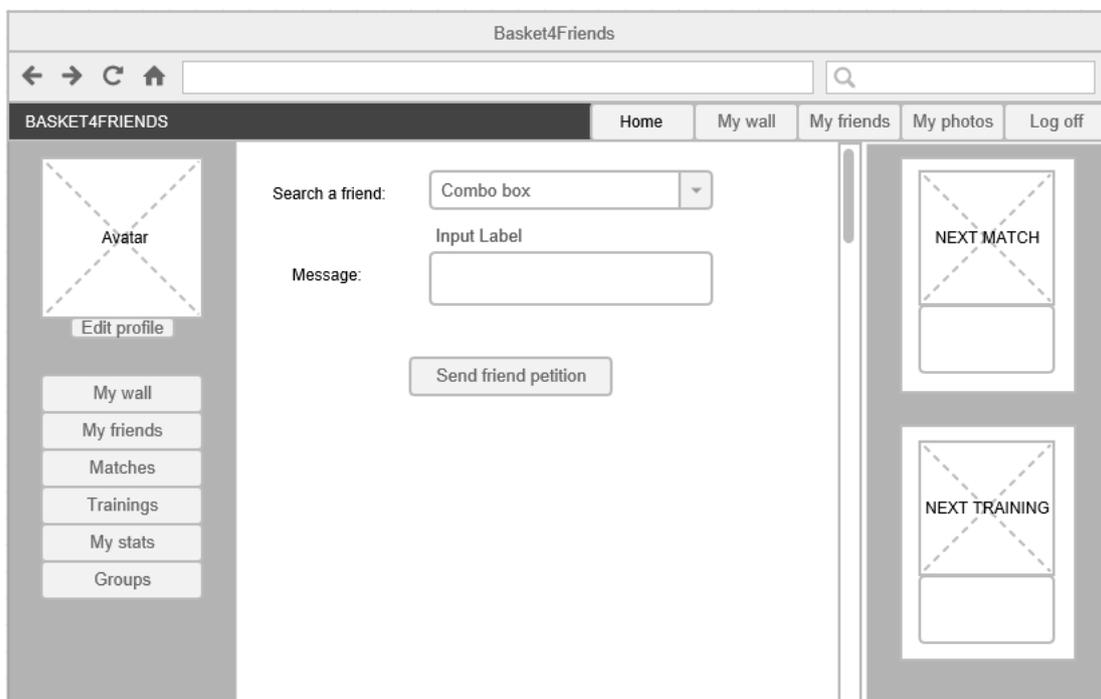
New message

My wall

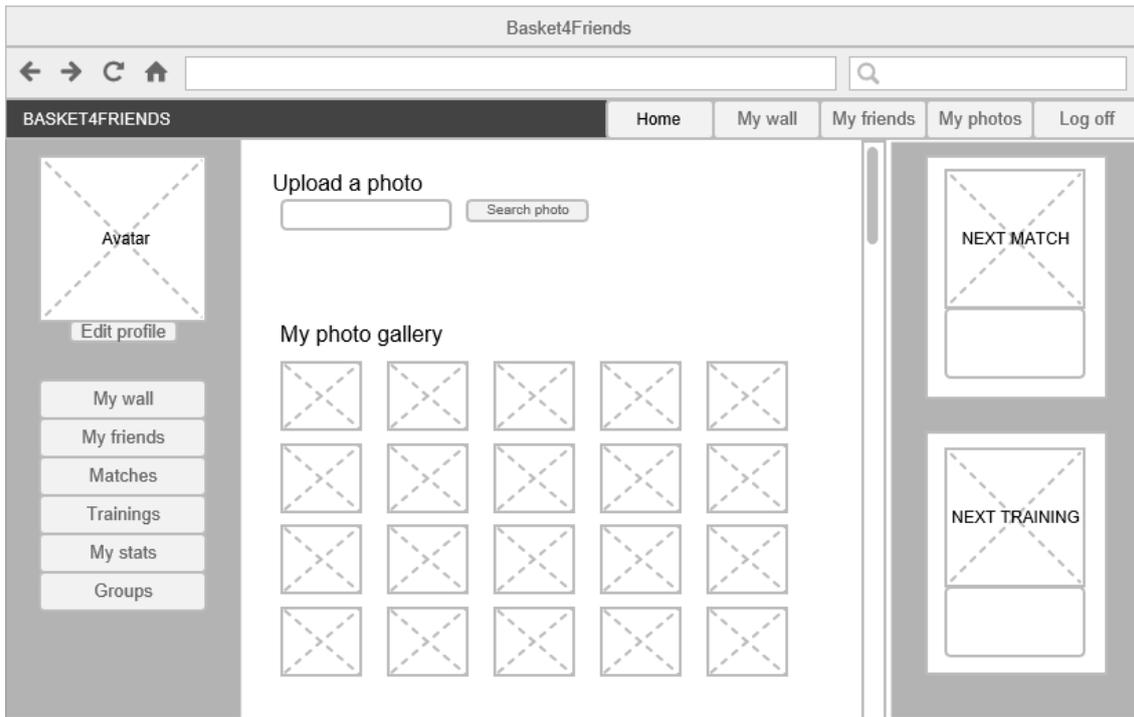
4. My wall.



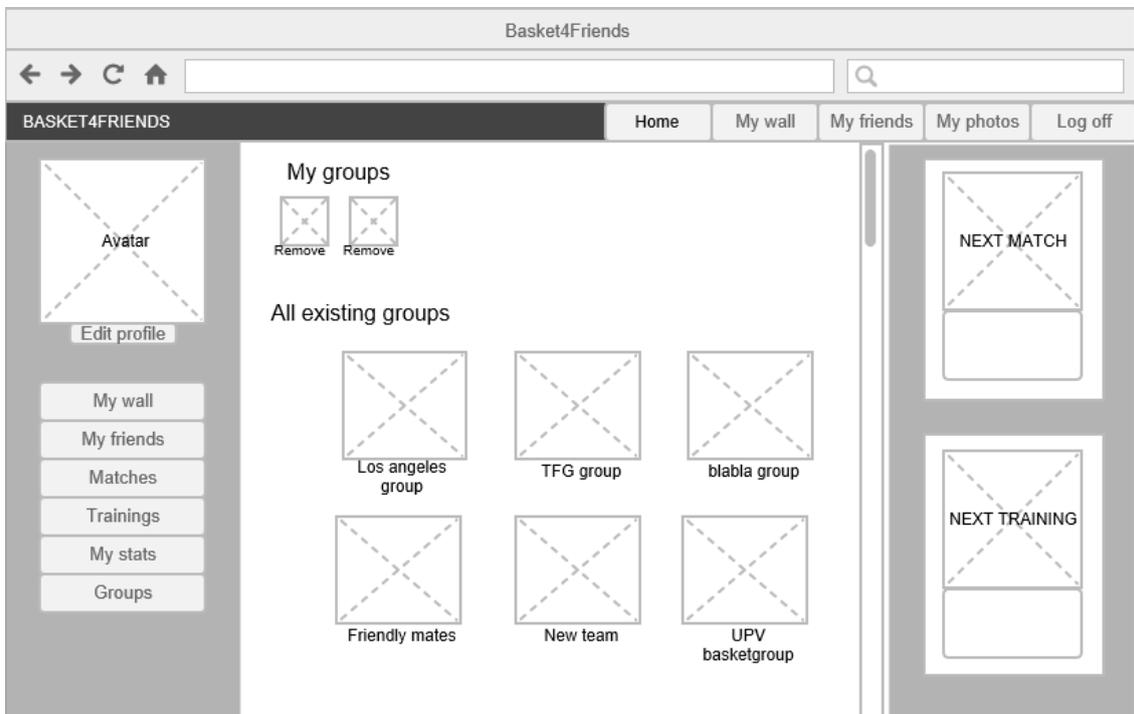
5. My friends.



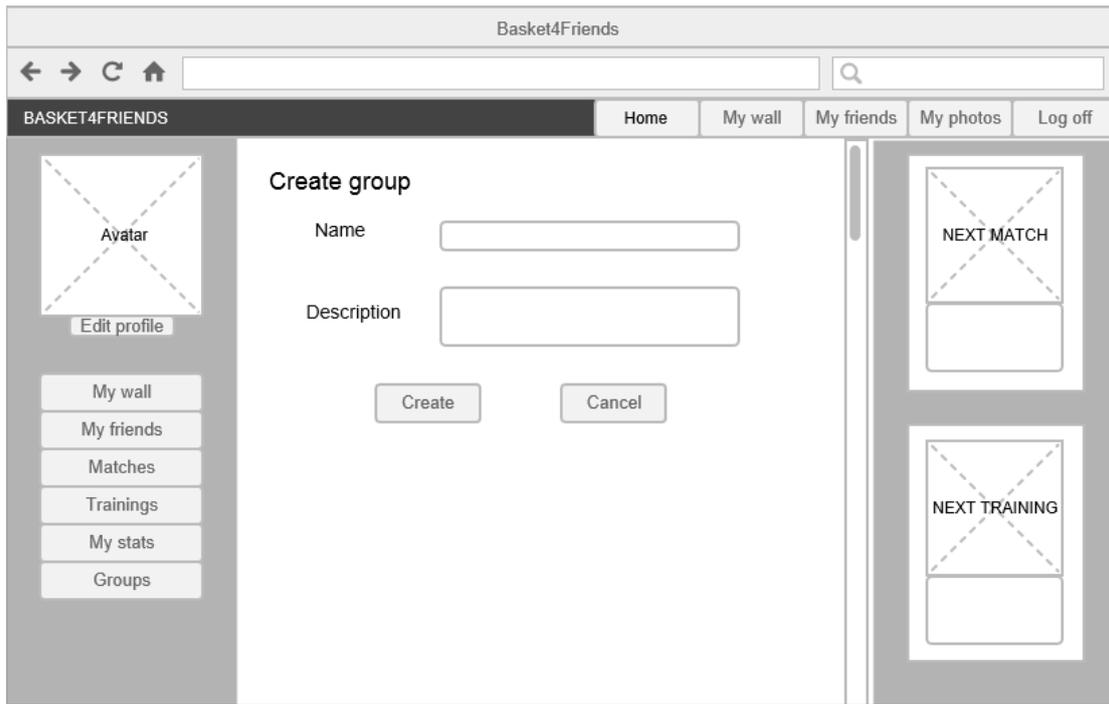
6. Friend petitions



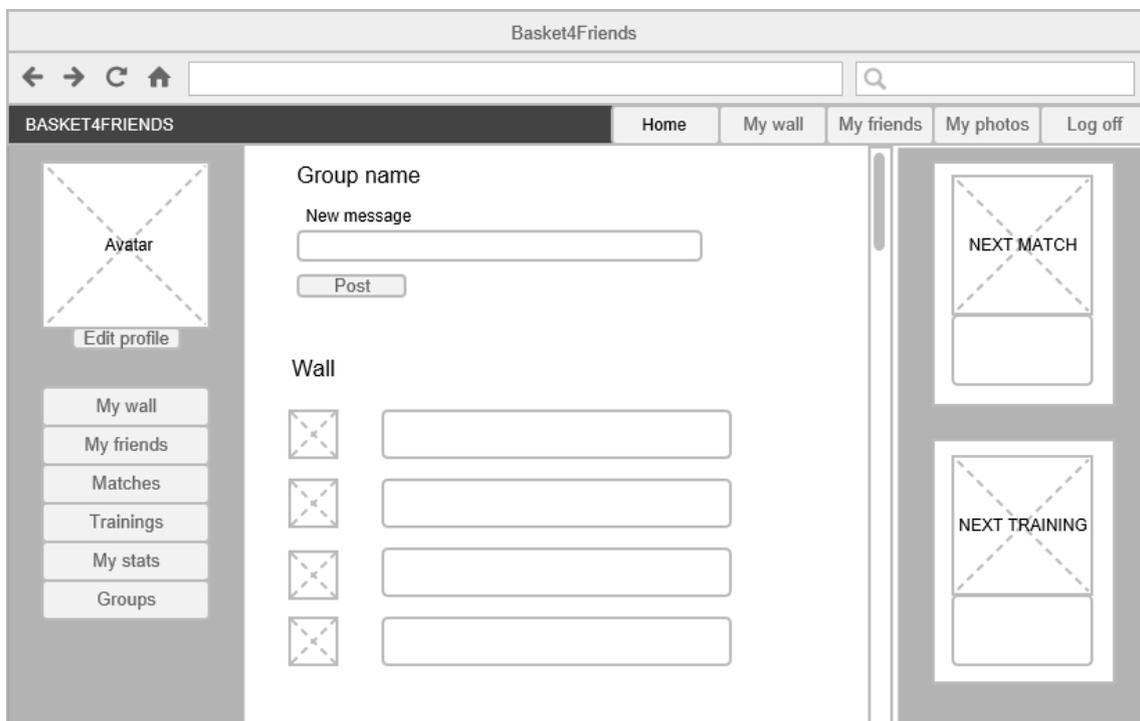
7. Photo gallery.



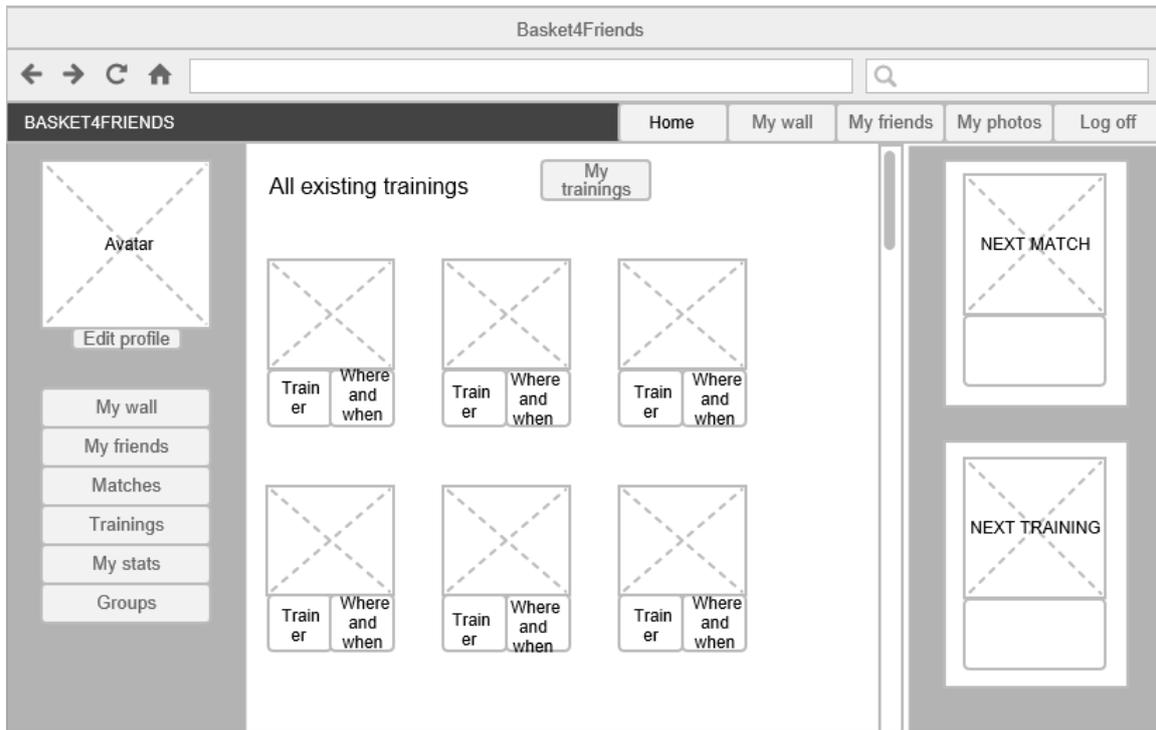
8. View all groups.



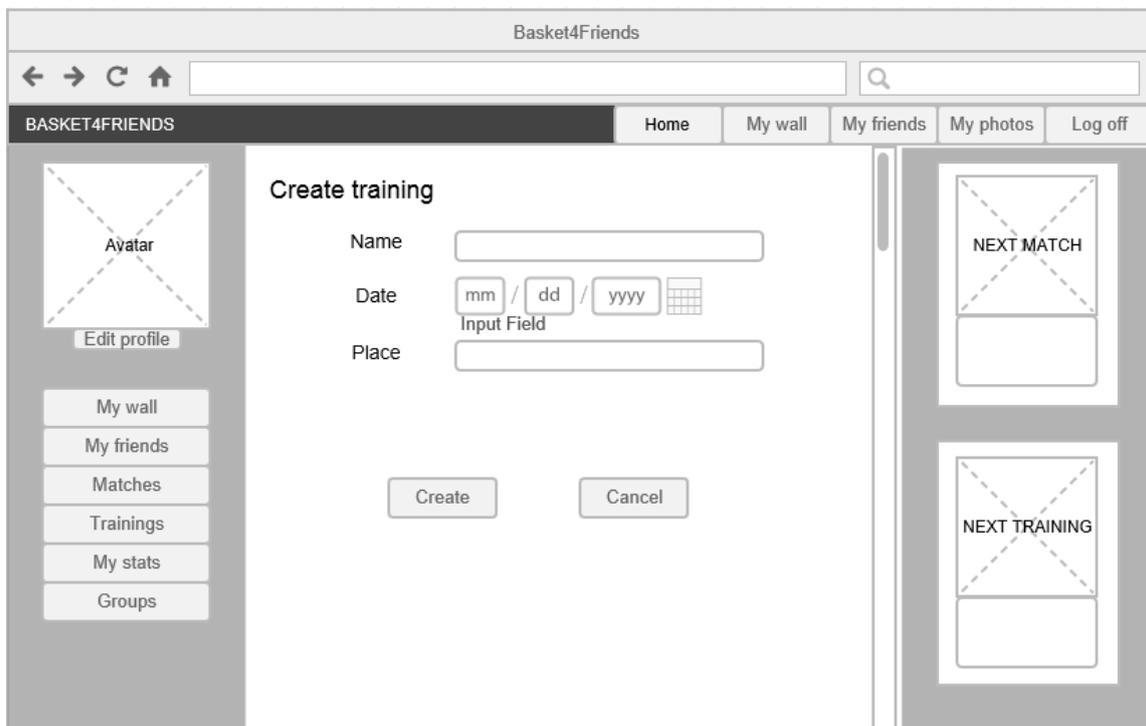
9. Create new group.



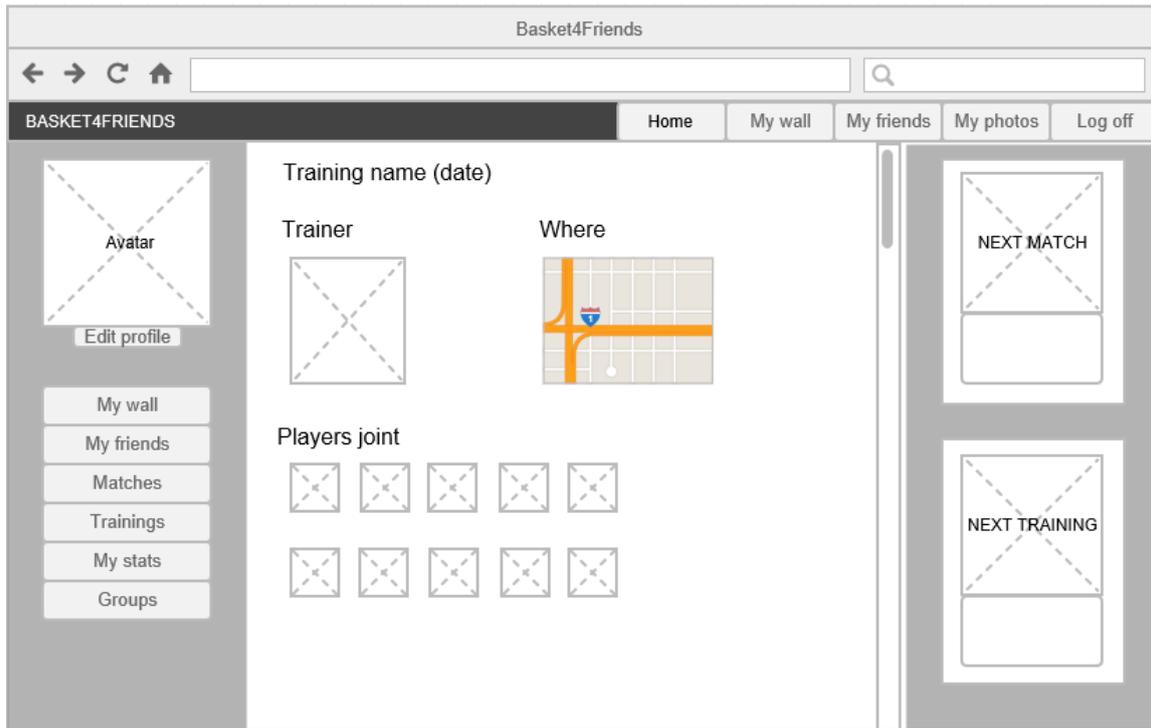
10. View group.



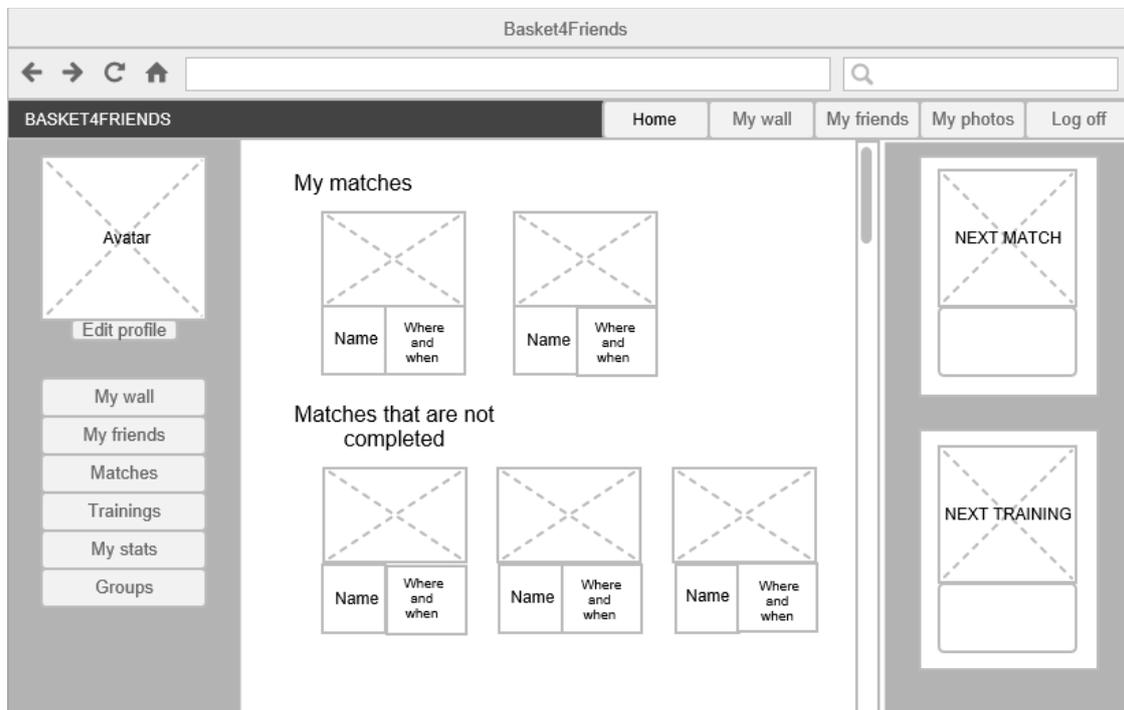
11. View all trainings.



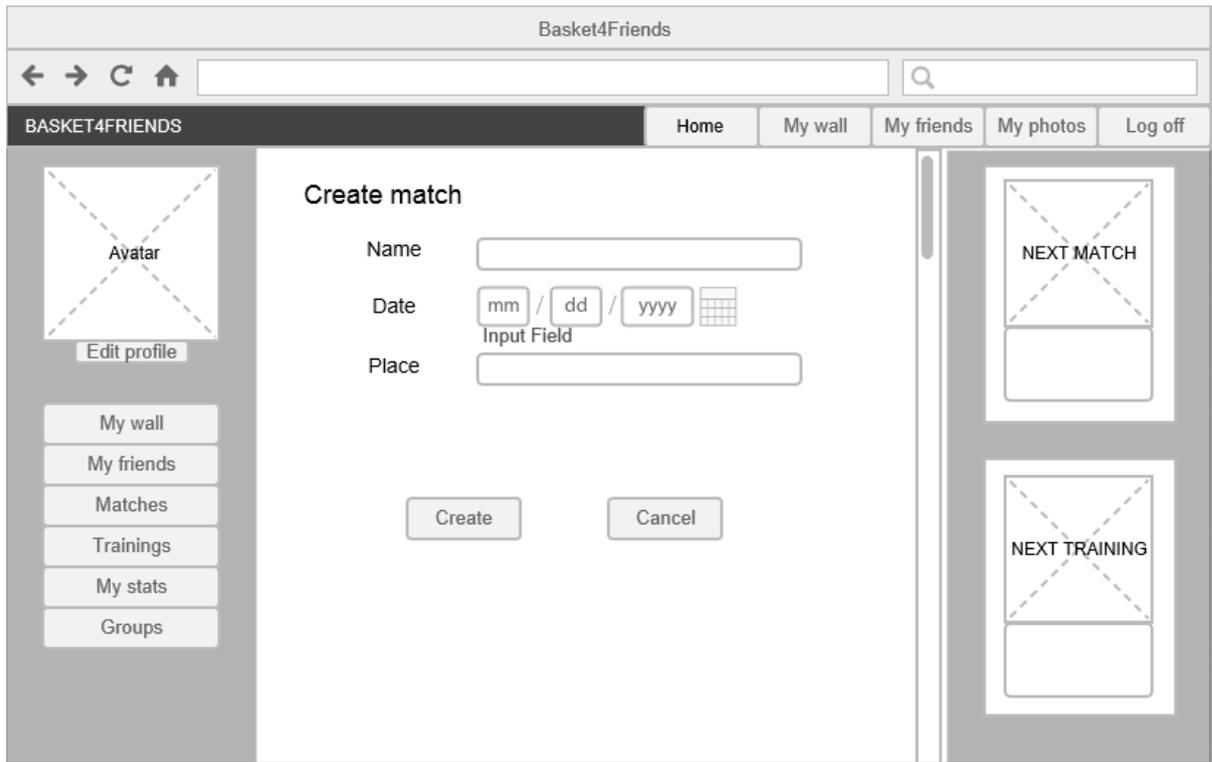
12. Create new training.



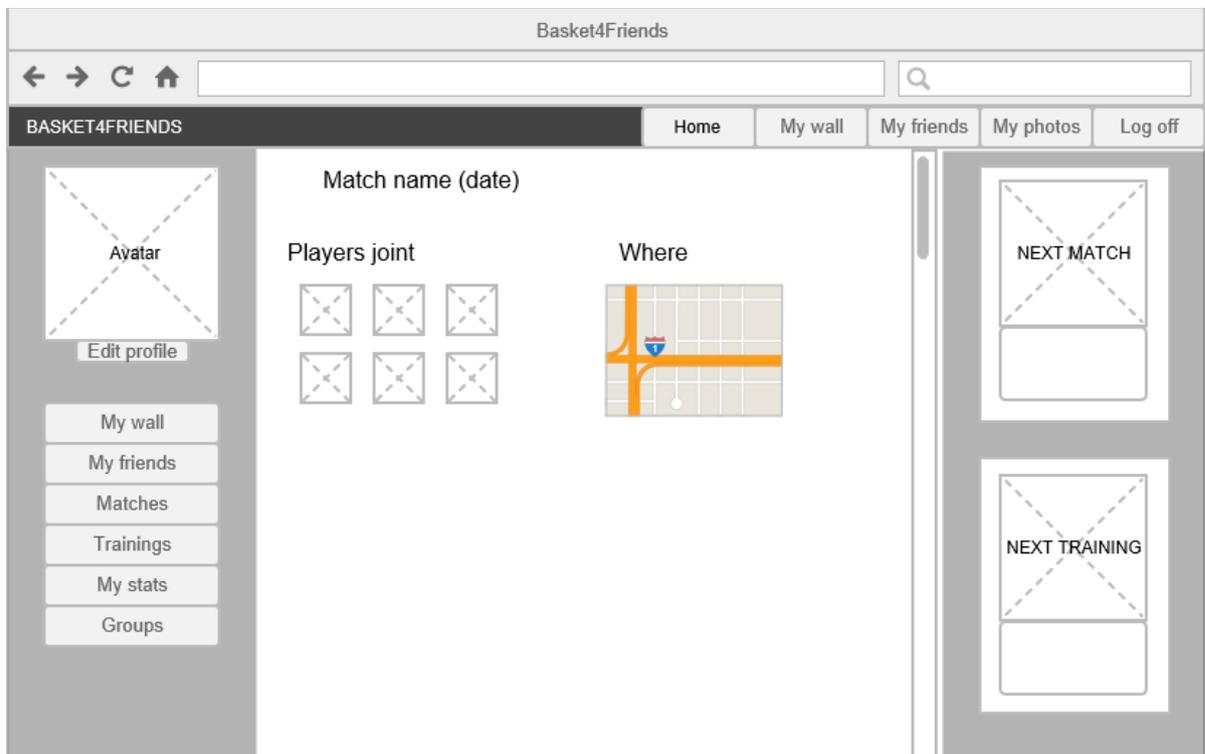
13. View training.



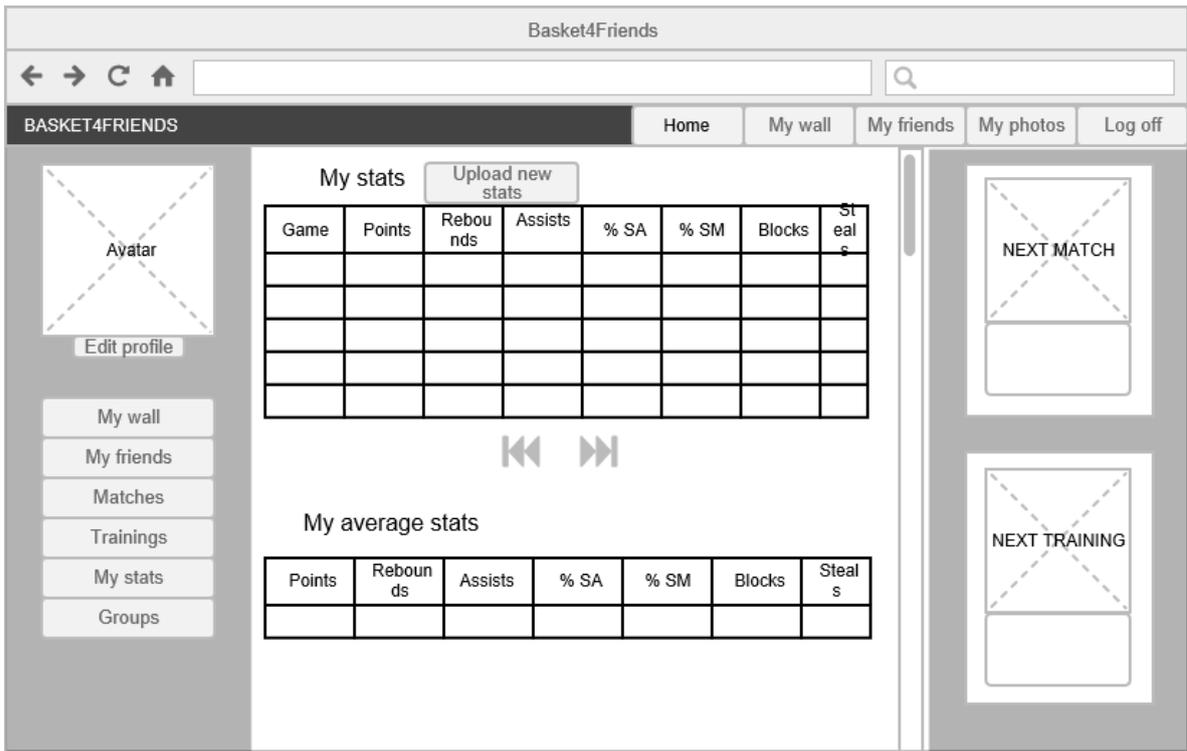
14. View all matches.



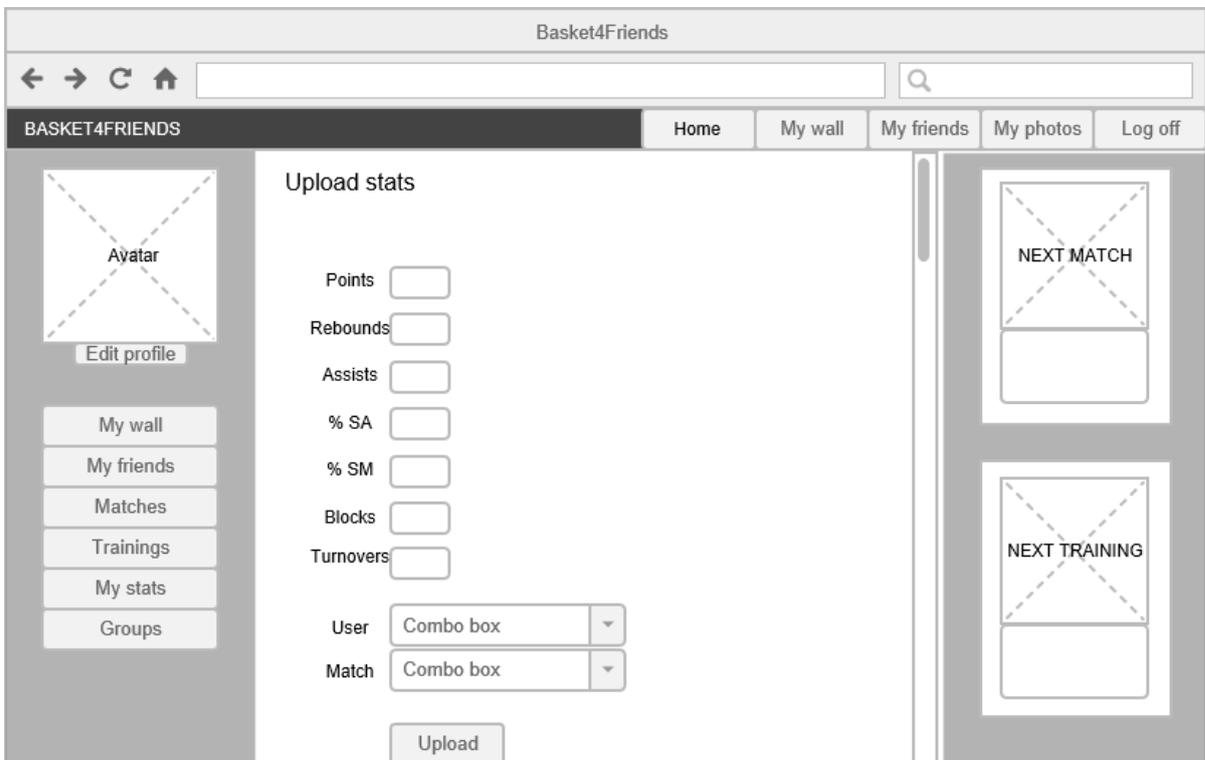
15. Create new match.



16. View match.



17. View stats.



18. Upload stats.

5. Detalles de implementación

A continuación se va a proceder a explicar los detalles de la implementación. Esto es, las herramientas y tecnologías usadas, así como la estructura de los ficheros y directorios llevada a cabo durante el desarrollo de la aplicación.

5.1. Tecnologías usadas

Para el desarrollo de la aplicación se ha elegido la utilización de tecnologías desconocidas para el desarrollador, todas estas de carácter *open-source*, a fin de aprenderlas, para ampliar el conocimiento en la rama de desarrollo web.

Yii Framework

Yii es un *framework* usado para crear aplicaciones PHP que implementa el patrón de arquitectura por excelencia MVC, o *Model-View-Controller*. Se trata de un *framework* similar al famoso *Symphony*, pero finalmente se decantó por Yii por haber sacado recientemente la versión 2.0 y por mejor curva de aprendizaje. Ya que es la primera vez que desarrollo una aplicación con un *framework*, éste no debía ser demasiado complejo. El funcionamiento de Yii2 queda reflejado en este diagrama.

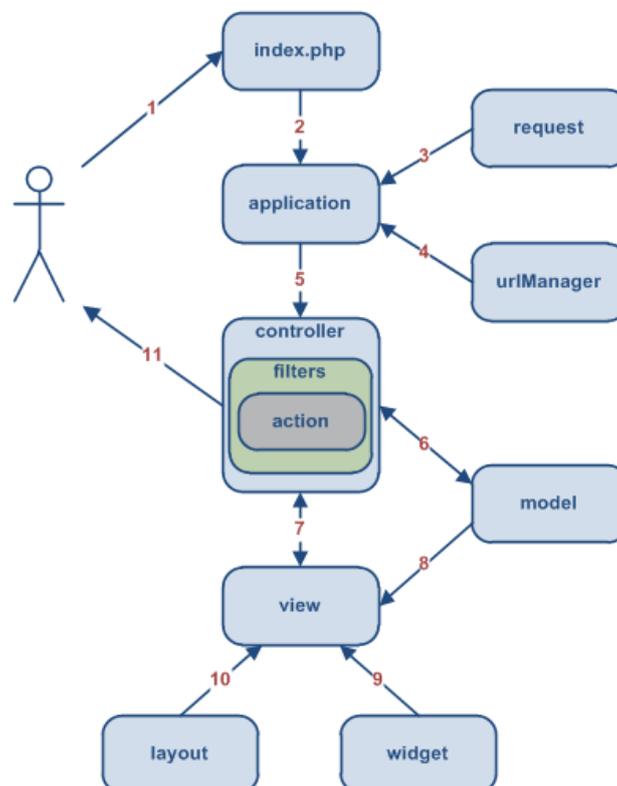


Figure 7. Image from Yiiframework.com

La funcionalidad que sigue este *framework* cuando un usuario realiza una petición mediante un navegador consta de estos pasos:

1. Un usuario hace una petición a la url `.../index.php?r=post/show&id=1` y el servidor maneja la petición ejecutando el script `index.php`.
2. El script crea una instancia de aplicación y la arranca.
3. La aplicación obtiene información detallada de la petición del usuario a través de un componente de la aplicación llamado `request`.
4. La aplicación determina que controlador y que acción van a ser ejecutados con la ayuda del `urlManager`. Para este ejemplo, la url nos indica que *route* es `post`, que refiere a `PostController` y que la acción a ejecutar es `show`, pasándole un parámetro `id`, que refiere a `actionShow($id)`. En términos generales, las rutas se forman con `r=controller/action`.
5. La aplicación crea y ejecuta los filtros, como el control de acceso o *benchmarking* asociados a esta acción. La acción es ejecutada si pasa todos los filtros satisfactoriamente.
6. La acción lee un modelo `Post` cuyo ID es 1.
7. La acción renderiza una vista llamada `show` con el modelo `Post`.
8. La vista lee y muestra los atributos del modelo `Post`.
9. La vista ejecuta los widgets correspondientes.
10. El resultado de la vista es incrustado con el *layout*.
11. La acción completa el renderizado y lo muestra al usuario.

Yii Framework también permite acceder a la base de datos mediante *Query Builders* o Active Records, que evita la necesidad de escribir consultas SQL repetidas. Además viene con la funcionalidad de autenticación ya incluida. Otras características de este *framework* las podemos encontrar en su [sitio web](#).

XAMPP

Es una herramienta muy útil para el desarrollo de aplicaciones ya que con su instalación ya dispones de un servidor APACHE, una base de datos MySQL, y de los lenguajes de script PHP y Perl.

Una vez instalado, basta con arrancar los servicios de Apache, para poder acceder al servidor mediante <http://localhost> y al servicio de MySQL. Para acceder a este último servicio, basta con escribir a través de un navegador web la url <http://localhost/phpmyadmin>. Aquí podemos crear nuevas bases de datos.

GitHub

Es una plataforma utilizada para alojar proyectos de forma remota en la nube, que utiliza el *framework* Ruby On Rails.

Mediante el sistema de control de versiones Git, un desarrollador puede subir su versión del código a la nube, así como descargarse otras versiones del código que ya se encuentren subidas por él o por otros usuarios. Esta herramienta es muy útil a la hora de



desarrollar proyectos grandes, donde cada desarrollador puede ir realizando su parte del proyecto, para luego subirlo y hacer un *merge* o integración de todos los cambios que se han realizado. Mediante los *commits* (guardar en repositorio local el proyecto) se puede también llevar un control de los cambios realizados en cada momento.

En el caso de este proyecto, se ha utilizado GitHub para dar mayor control sobre las versiones, así como para tener una versión actualizada del proyecto en la nube, aumentando así la persistencia del código.

NetBeans

NetBeans IDE es un entorno de desarrollo que permite crear aplicaciones web con PHP5. Además viene con soporte para aplicaciones con patrón de diseño MVC y con soporte de Git. Otro entorno de desarrollo similar muy conocido es Eclipse.

PHP

Es un lenguaje de programación que se ejecuta en el lado del servidor diseñado para el desarrollo web. Destaca por la capacidad de ser incorporado directamente en los ficheros HTML, mediante las sentencias `<?php ... ¿>`.

Cuando el cliente realiza una petición al servidor para que éste le envíe la página web solicitada, el servidor ejecuta el intérprete de PHP, el cual procesa el script y así, genera el contenido de forma dinámica. El intérprete envía el resultado al servidor, y éste al cliente.

PHP destaca por su capacidad de ser usado en la mayoría de servidores, así como de sistemas operativos. Otras de sus ventajas son la facilidad de acceso a las bases de datos y su facilidad de aprendizaje.

Algunos ejemplos de sitios desarrollados con PHP son Facebook y WordPress.

MySQL

Es un sistema de gestión de bases de datos relacional muy utilizado en las aplicaciones web. Una base de datos relacional archiva datos en tablas separadas en lugar de colocar éstos en un único archivo. Esto le da a MySQL mucha velocidad y flexibilidad, dos características vitales para el entorno web. Las tablas están conectadas mediante relaciones definidas que permiten combinar datos de distintas tablas. Destaca por su simplicidad y facilidad de aprendizaje.

HTML

Se trata de un lenguaje de marcas usado para la elaboración de páginas web que todos los navegadores interpretan. Se compone de *tags* de apertura y de cierre ('<'

elemento' />') que incluyen elementos que indican que tipo de contenido se va a visualizar, como por ejemplo, párrafos, tablas y listas entre otros muchos.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico, como el bloc de notas de Windows, o con entornos de desarrollo más avanzados, como en este proyecto, en NetBeans.

CSS

Es el lenguaje utilizado para la presentación de las páginas web. *CSS* o *Cascading Style Sheet* se encarga de dar estilo a las páginas web o HTML, ya sea mediante colores, rellenos, márgenes, etc. La información para el estilo de estas páginas la obtienen los navegadores de los ficheros HTML, dónde el código CSS está incrustado, o de los ficheros CSS invocados en los ficheros HTML. Algunas de las ventajas de utilizar CSS son el control centralizado de la presentación de un sitio web completo, la optimización del ancho de banda de la conexión o la mejora de la accesibilidad del documento.

Bootstrap

Es un conjunto de herramientas para el diseño de sitios y aplicaciones web creado por los mismos desarrolladores de Twitter que contiene plantillas ya creadas de diseño basados en HTML, CSS y JavaScript como diferentes tipos de tipografía, botones, formularios, alertas, menús de navegación, etc.

Para poder utilizar esta herramienta, solo hace falta descargar la hoja de estilo *Bootstrap CSS* y enlazarla al archivo HTML, como haríamos con cualquier fichero .css creado. A partir de aquí ya podemos usar *Bootstrap* aplicando a los elementos las clases de estilos predeterminados.

En el *framework Yii2*, *Bootstrap* viene incorporado por defecto.

Gliffy

Es una herramienta online que sirve para dibujar diagramas de muchos tipos. Se ha utilizado en esta aplicación para dibujar las tablas de bases de datos, así como el modelado UML y sus casos de uso.

5.2. Estructura de los ficheros y directorios

Para la estructuración de los ficheros del proyecto se ha llevado a cabo la estructura natural que viene en el *framework* Yii. Esta sigue tal y como muestra el diagrama siguiente:

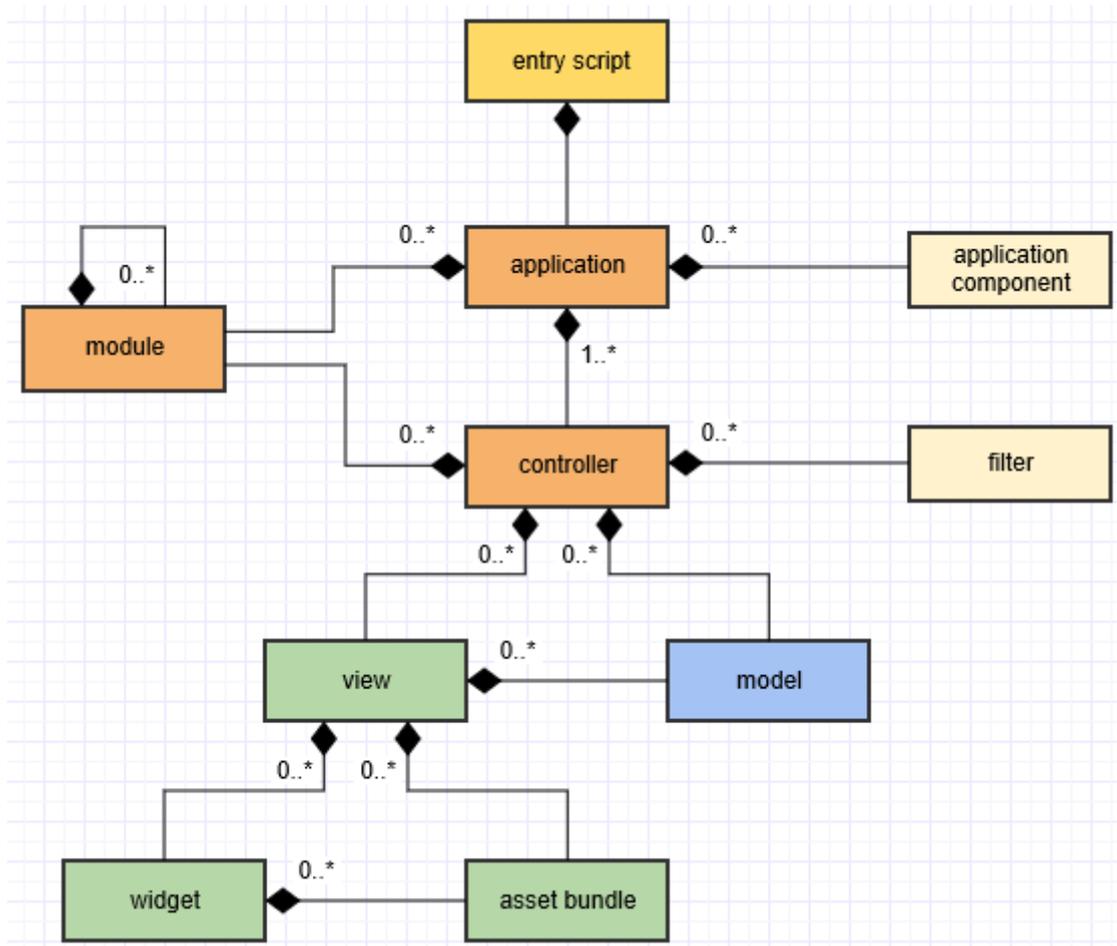


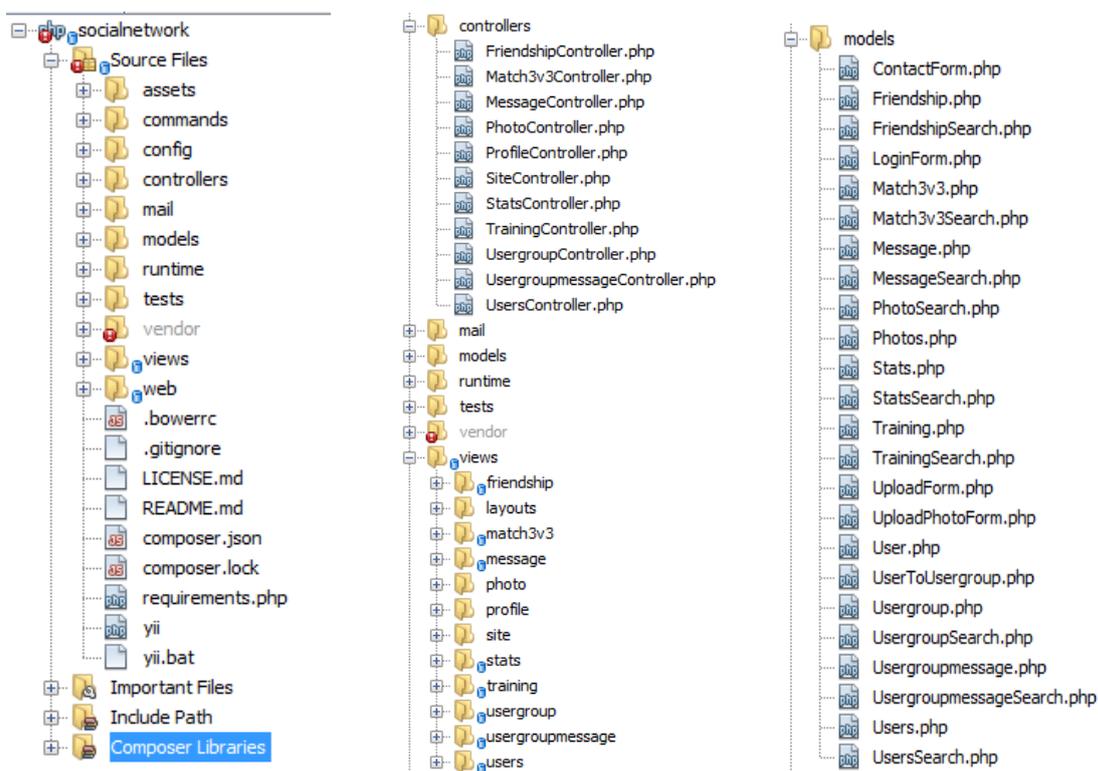
Figure 8.

Algunas de las funcionalidades que trae este *framework* no han sido utilizadas, por lo tanto, no se van a encontrar en nuestra aplicación los directorios de filtros y módulos, aunque en el caso de los filtros sí se han usado pero aplicados directamente en los *controllers*.

De esta manera, el directorio de la aplicación web tiene las siguientes carpetas y ficheros.

- **assets/** : Los *assets* son ficheros que pueden ser referenciados en una página web. Estos pueden ser ficheros CSS, Javascript, imágenes, videos, etc.
- **commands/** : Contiene los comandos de consola (no se aplica en nuestra aplicación).
- **config/** : Esta carpeta contiene los ficheros de configuración de la aplicación. Aquí se pueden configurar los componentes de la aplicación, las bases de datos utilizadas, los parámetros, etc.

- **controllers/** : Las clases Controller de la aplicación web se sitúan aquí. Se ha añadido un nuevo XyzController.php por cada grupo de funcionalidades, quedándonos así de forma más estructurada según la funcionalidad buscada. El formato seguido para nombrarlos ha sido el recomendado por Yii. “*NombredelmodeloController.php*”, excepto para *ProfileController*, que engloba una serie de acciones básicas de una red social.
- **mail/** : Contiene los ficheros de las vistas para los emails. En nuestra aplicación no se ha aplicado ningún cambio.
- **models/** : Este directorio almacena los ficheros .php de los modelos. Cada tabla de la base de datos tiene un modelo con su mismo nombre (pero empezando por mayúscula) y otro modelo de búsqueda o *queries* a la tabla que para nombrarlo sigue el formato “*NombredemodeloSearch.php*”.
- **modules/** : Aquí se sitúan los diferentes módulos que se creen o instalen. En nuestro caso, no ha sido necesario la creación de ninguno.
- **runtime/** : Contiene los ficheros generados durante el tiempo de ejecución. En nuestro escenario, tampoco se ha usado.
- **tests/** : Contiene tests básicos para la aplicación.
- **views/** : El directorio *views* contiene los ficheros de las vistas y los *layouts* utilizados. Para una mayor organización, dentro de *views* se ha creado un folder por cada controlador, además de un folder para los *layouts*.
- **web/** : Finalmente tenemos la carpeta web. Esta carpeta es la utilizada para almacenar los entry scripts y los recursos utilizados en la aplicación, como por ejemplo, las imágenes, los assets de bootstrap y de css, etc). Dentro de web, destacar la carpeta *uploads*, donde se almacenan los avatares de los usuarios, así como la de *photoUpload*, que almacena las imágenes que los usuarios suben al sitio web.



6. Pruebas

Para finalizar el desarrollo de la aplicación, se va a someter al sitio web a una serie de pruebas de validación para ver que la aplicación funciona correctamente. Esto es, que nuestro sitio web tenga todas las funcionalidades requeridas anteriormente funcionando, que todos los enlaces funcionan correctamente, así como las vistas renderizadas. También se va a someter la aplicación a pruebas para ver la resolución en distintos navegadores.

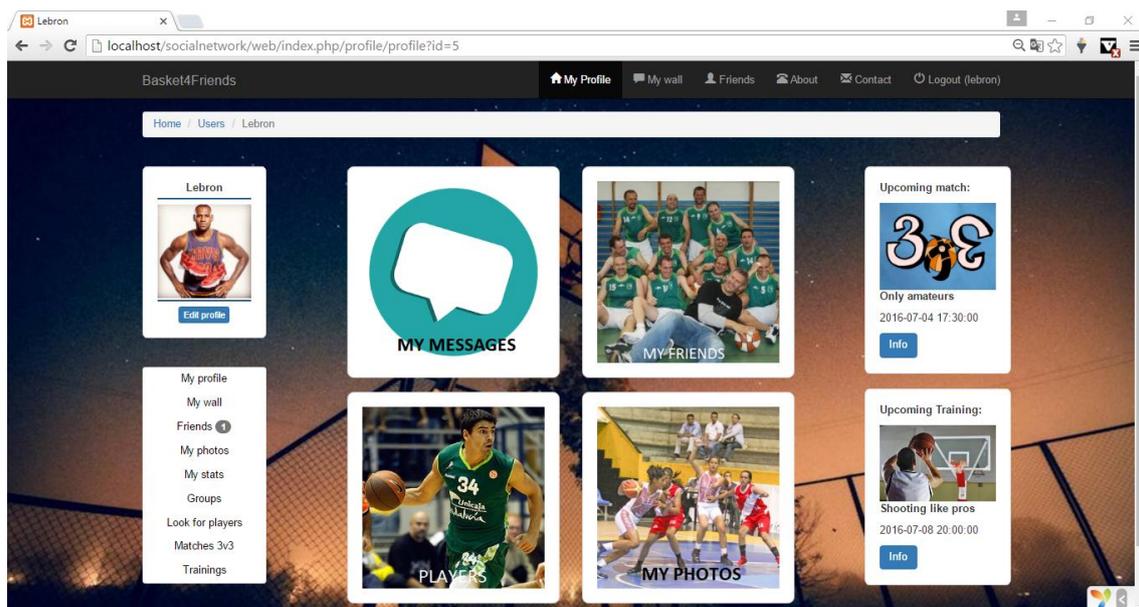
Finalmente, se hará una demostración del funcionamiento de la aplicación mediante capturas de pantalla.

6.1. Validación

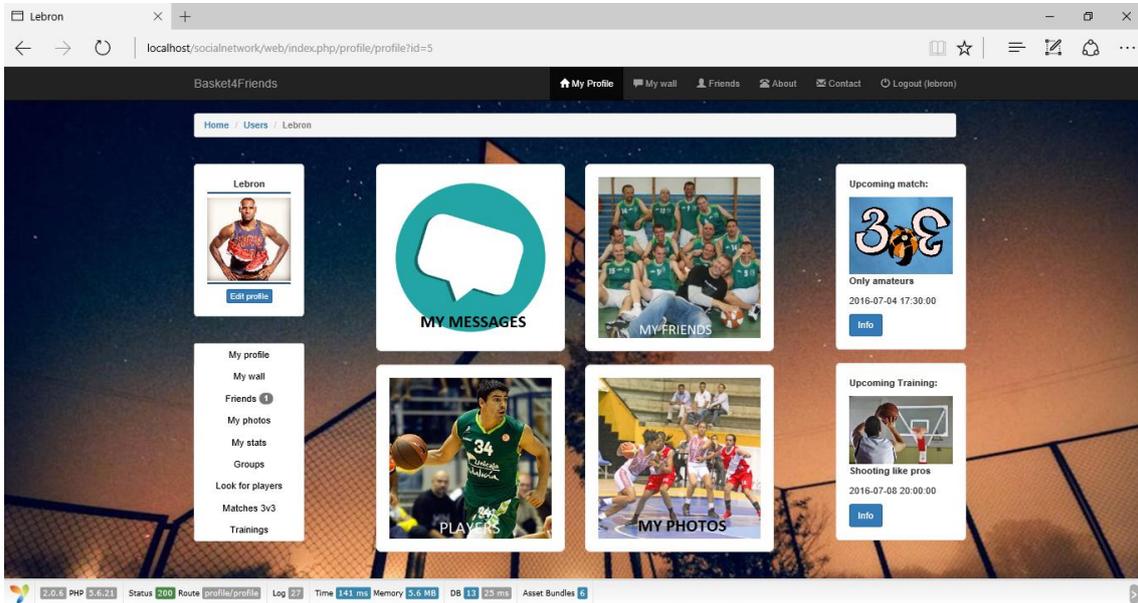
Se ha procedido a la validación del comportamiento de la aplicación en diferentes navegadores, usando Internet Explorer, Google Chrome, Mozilla Firefox y Blisk, así como en dispositivos de distinto tamaño. También se ha comprobado la validación HTML y CSS mediante el consorcio W3C. Como la aplicación se encuentra en local y no en la web, se ha procedido a comprobar los enlaces uno por uno para ver si había algún enlace roto. El resultado ha sido satisfactorio por esta parte también.

Navegadores.

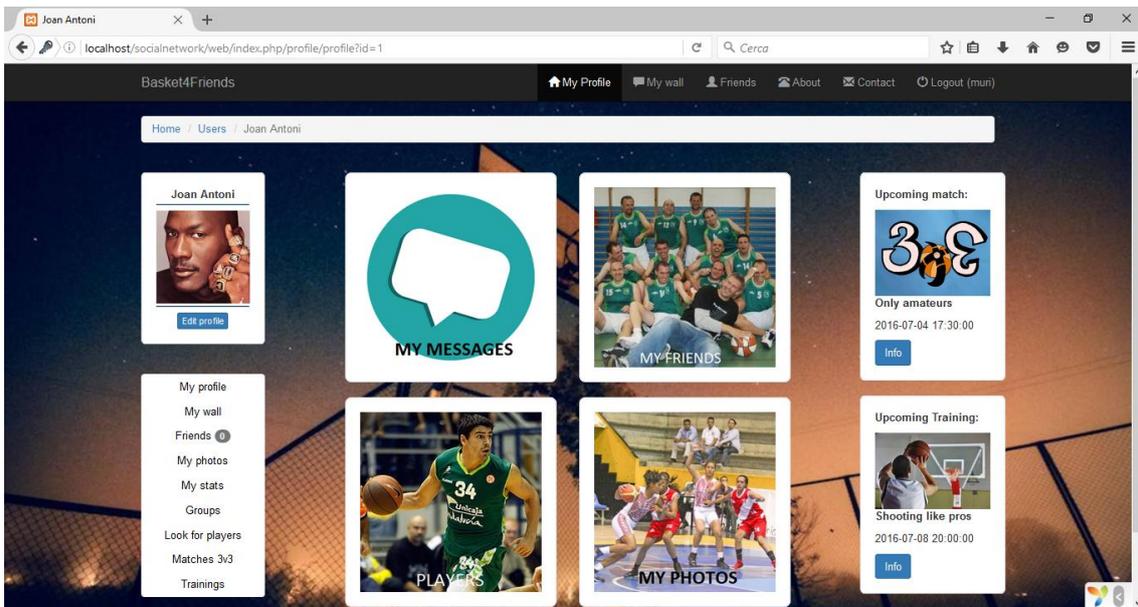
Chrome.



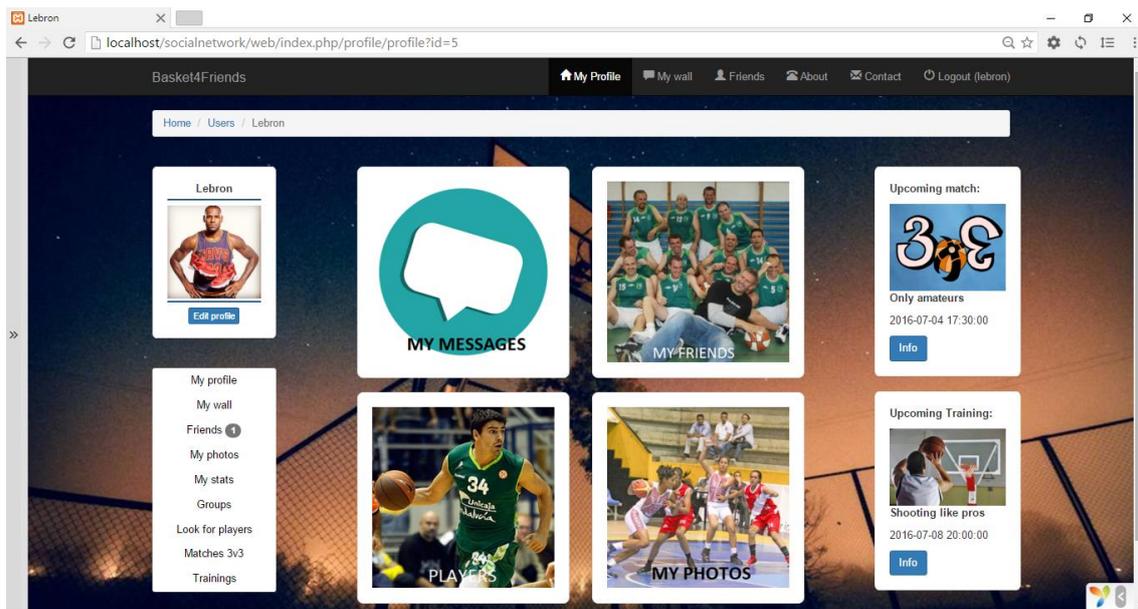
Internet Explorer



Mozilla Firefox



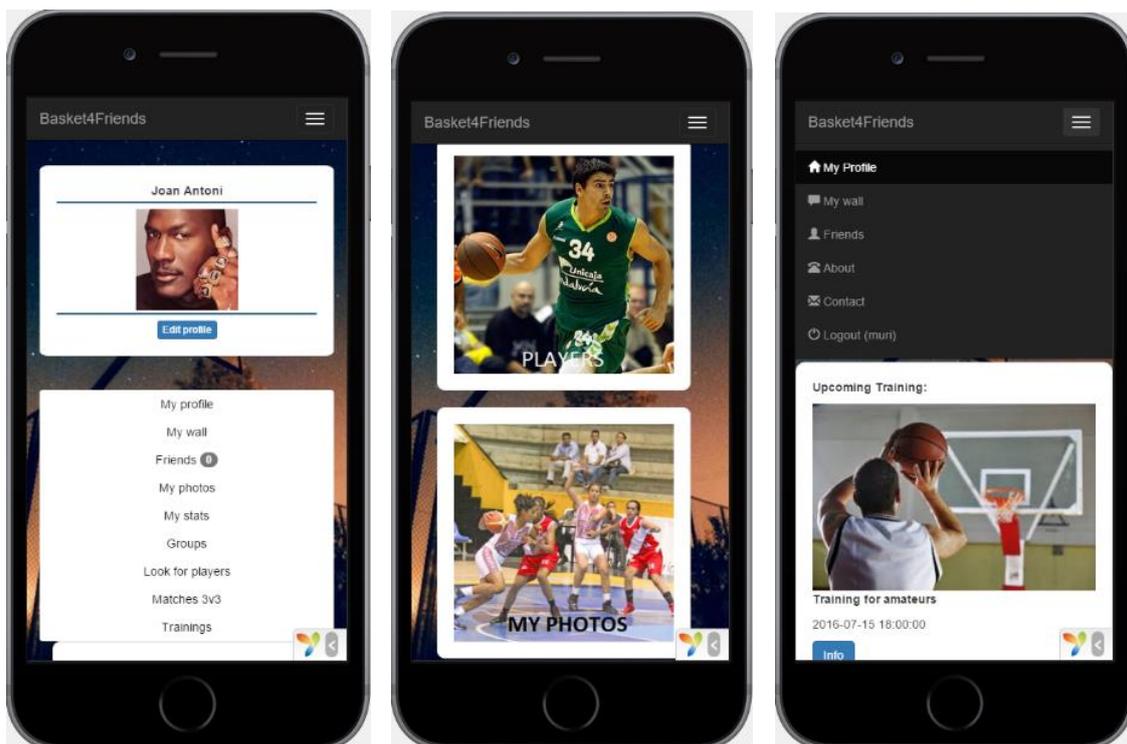
Blisk.



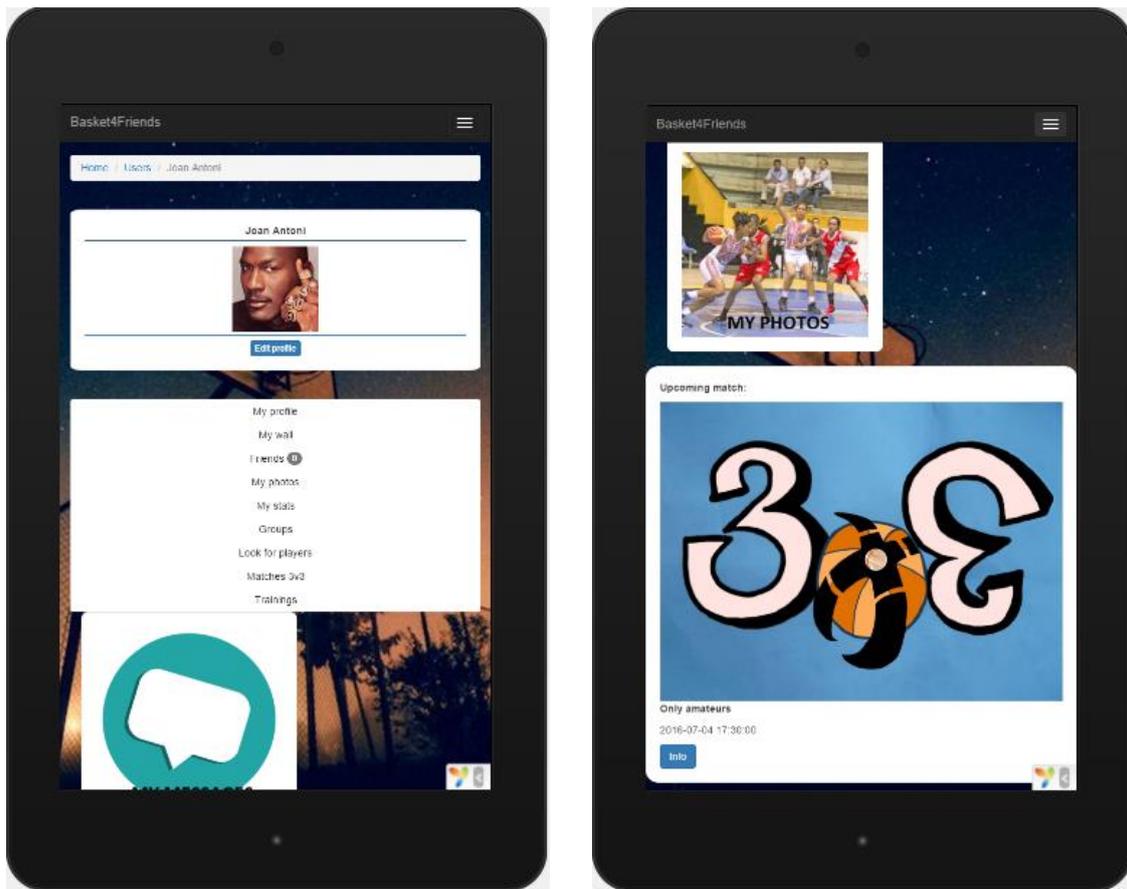
Dispositivos.

Las pruebas anteriores se han realizado con un ordenador portátil de 15 pulgadas. Aunque no entraba en los requisitos, se está procediendo a mejorar la aplicación para que sea 100% responsive. De momento, estos son los resultados obtenidos en la visualización de la aplicación con un dispositivo móvil y una Tablet.

Iphone 6.



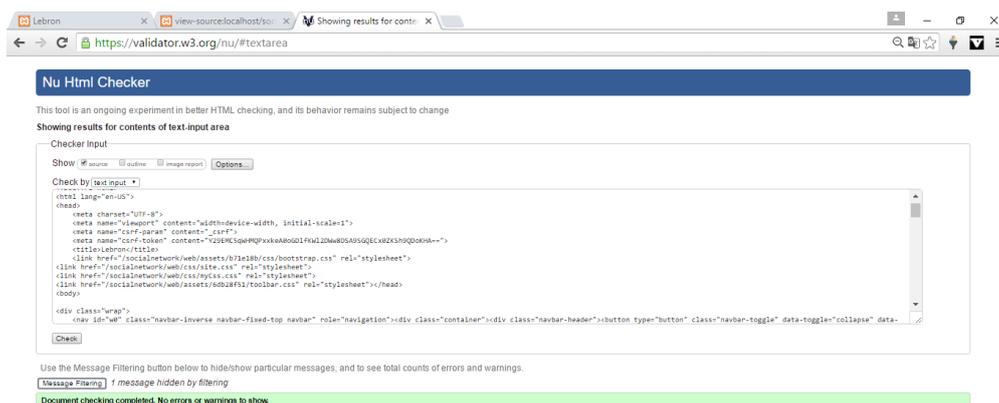
Tablet de 7”.



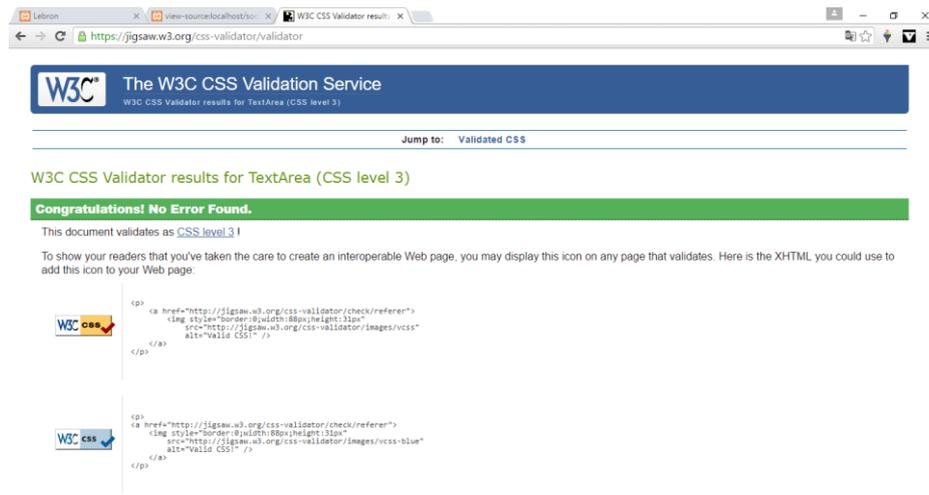
Otras validaciones.

Los resultados de las validaciones HTML y CSS de la aplicación han sido un éxito. Se han comprobado todas las páginas de la aplicación y han pasado las pruebas. Sólo destacar un warning que se produce porque el framework Yii pone un role al nav superior, y esto se encuentra obsoleto según el consorcio W3C y otros problemas menores de la transformación que hace Yii a HTML.

Validaciones HTML.



Validaciones CSS.



6.2. Pruebas de uso

Una vez finalizada la aplicación, se va a proceder a mostrar el resultado de la red social implementada, con cada una de sus pantallas (por cuestión de tamaño, se ha procedido a reducir el zoom y quitar los *toolbar* para que pueda obtenerse una mejor visualización de las pantallas).



1. Start page.

Basket4Friends

Home / Users / Registration

Registration

Name

Surname

Username

Password

Email

Birthday

Position

Categoria

Avatar

Tira un fibret? No s'ha llegit cap fibret

Create

2. Registration.

Basket4Friends

Home / Login

Login

Please fill out the following fields to login:

Username

Password

Remember Me

Login

You may login with admin/admin or demo/demo.
To modify the username/password, please check out the code: `app\models\User::users`

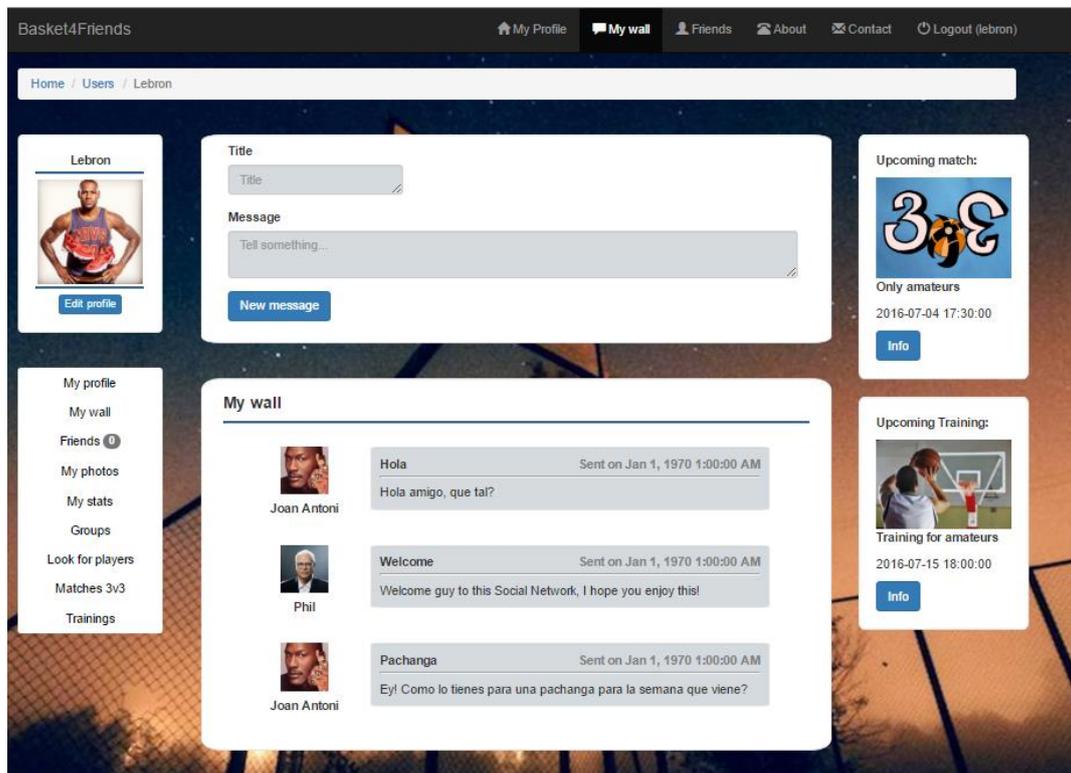
© Basket4Friends 2016

Powered by Yii Framework

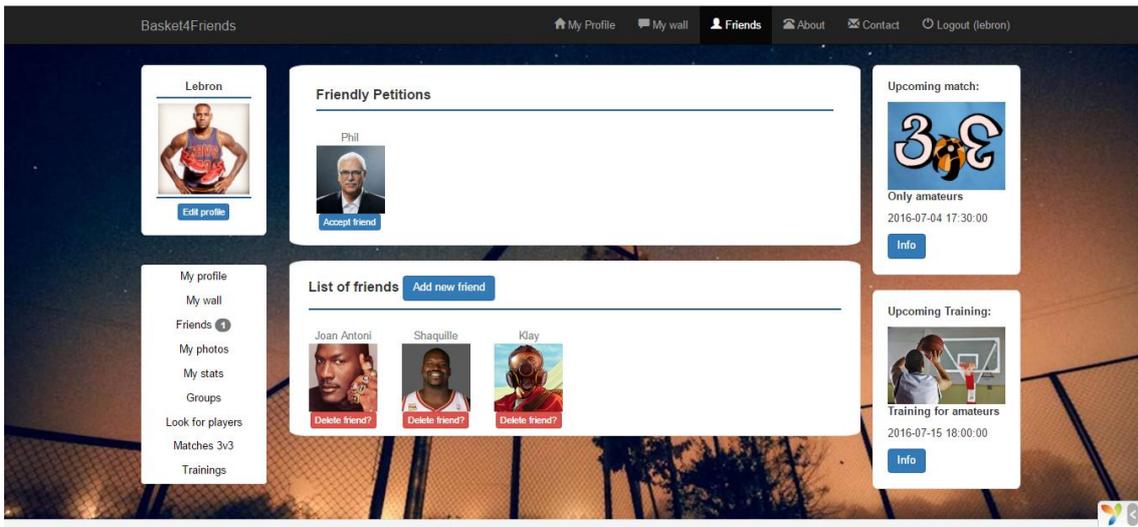
3. Login.



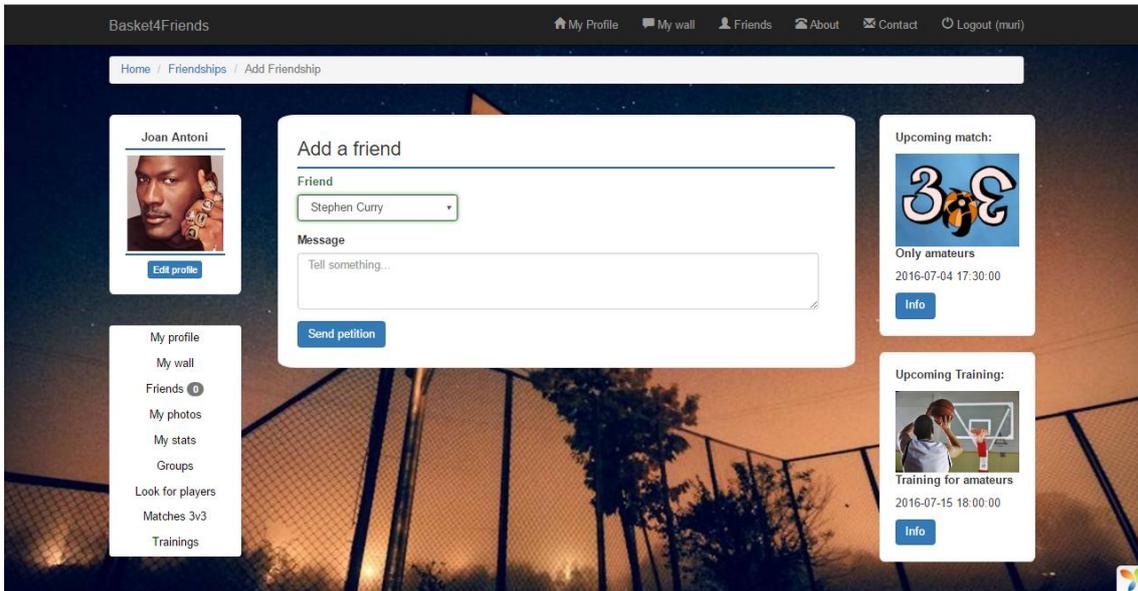
4. Profile.



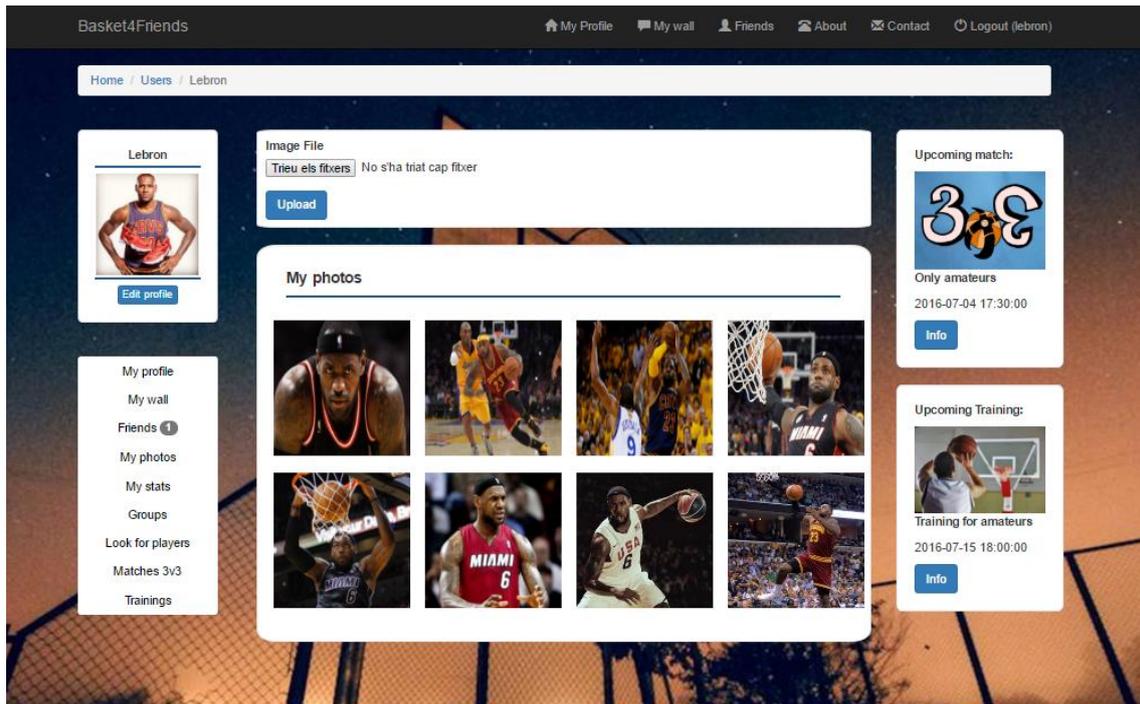
5. My wall.



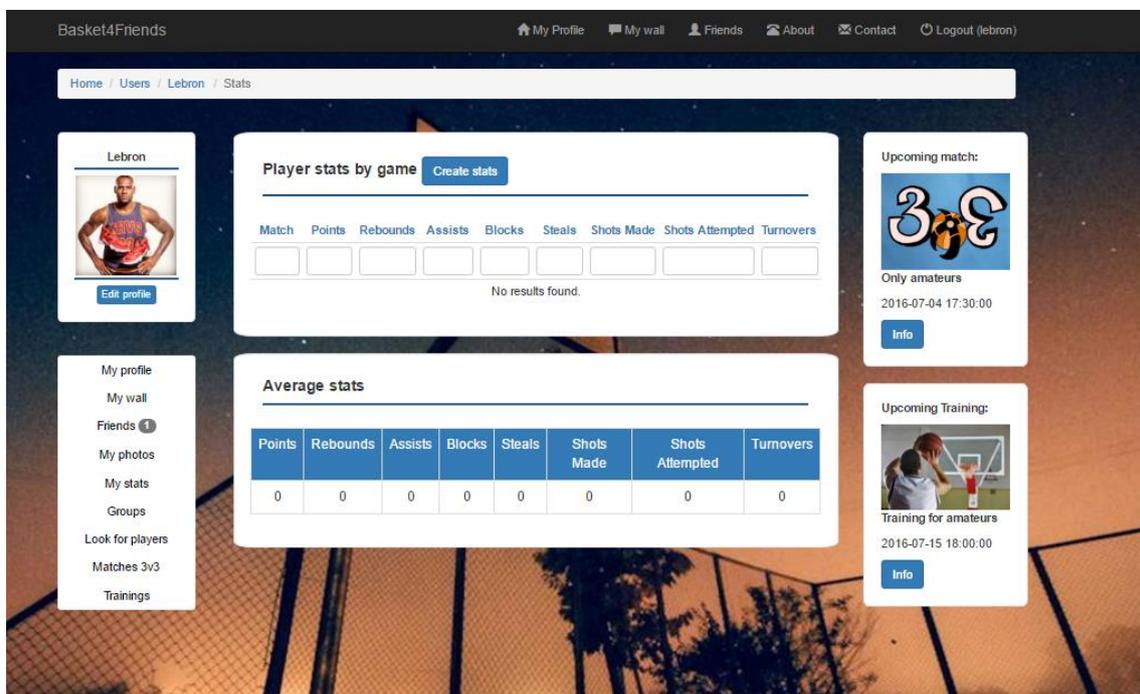
6. My friends.



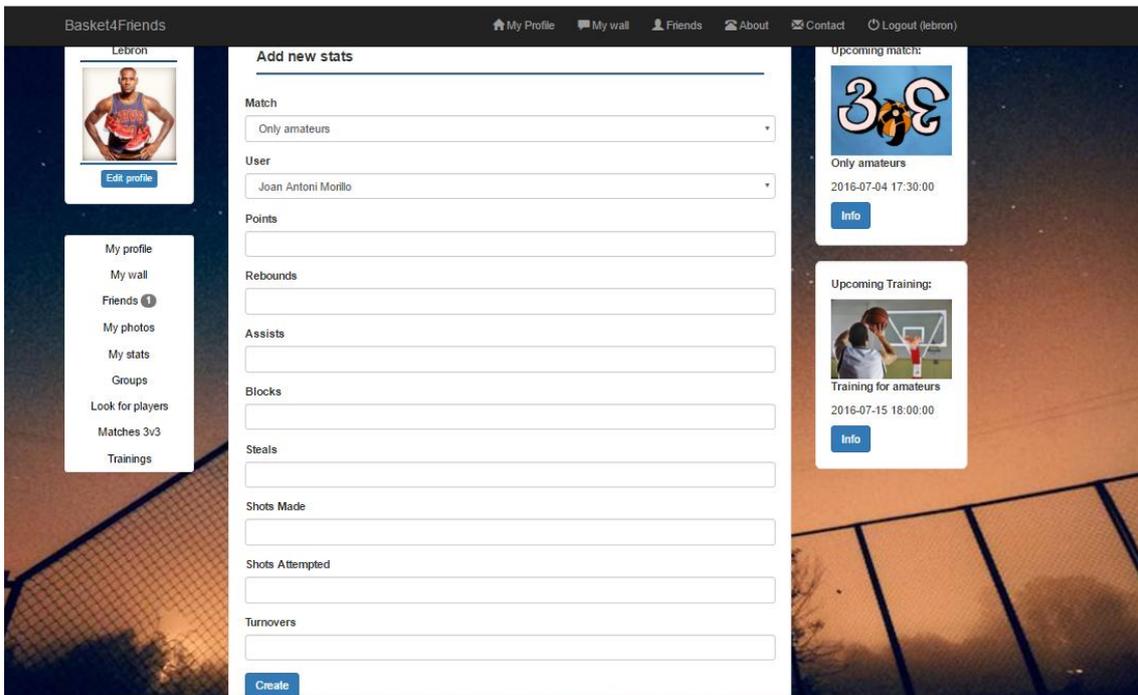
7. Add new friend.



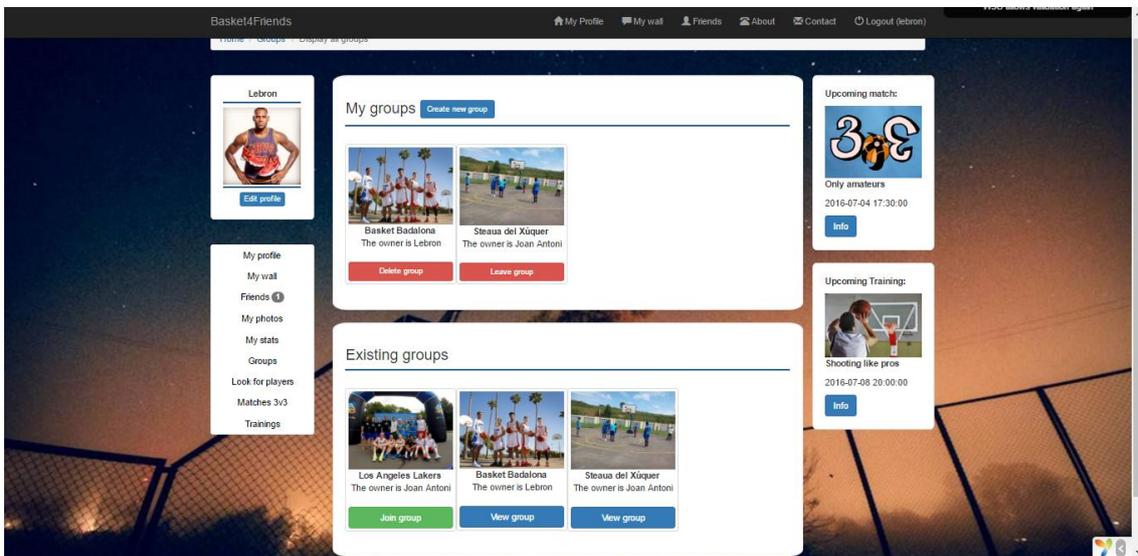
8. My photos.



9. My stats.

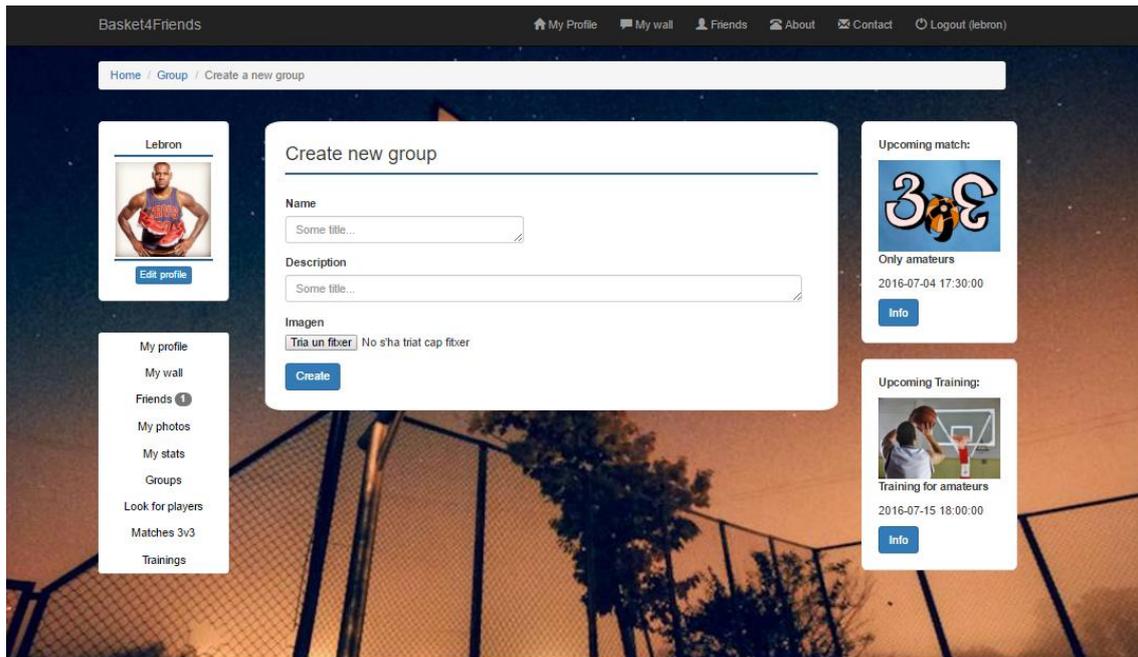


10. Create stats.

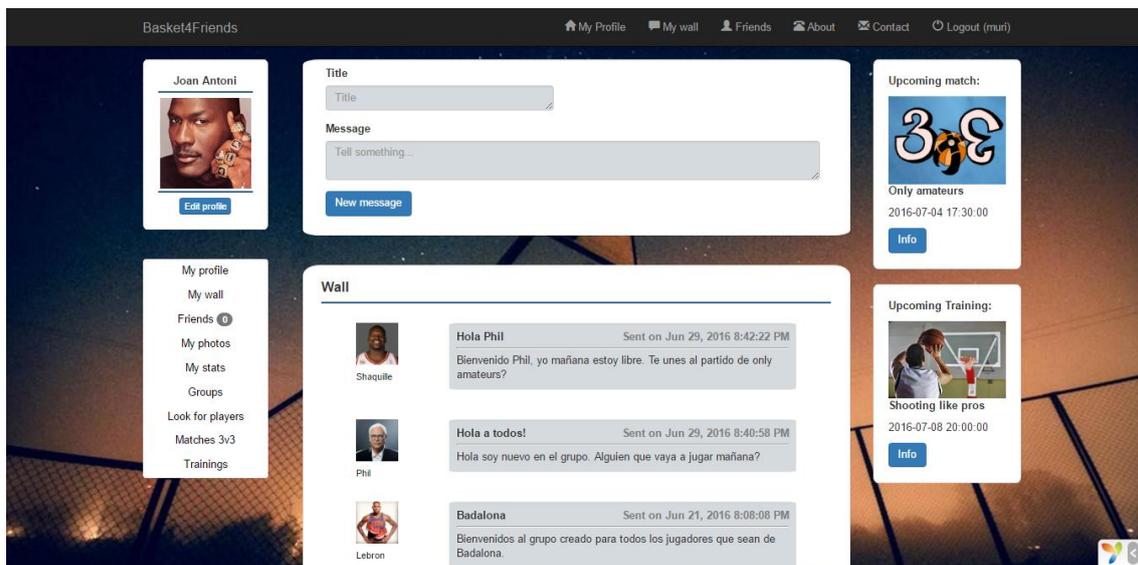


11. Groups.

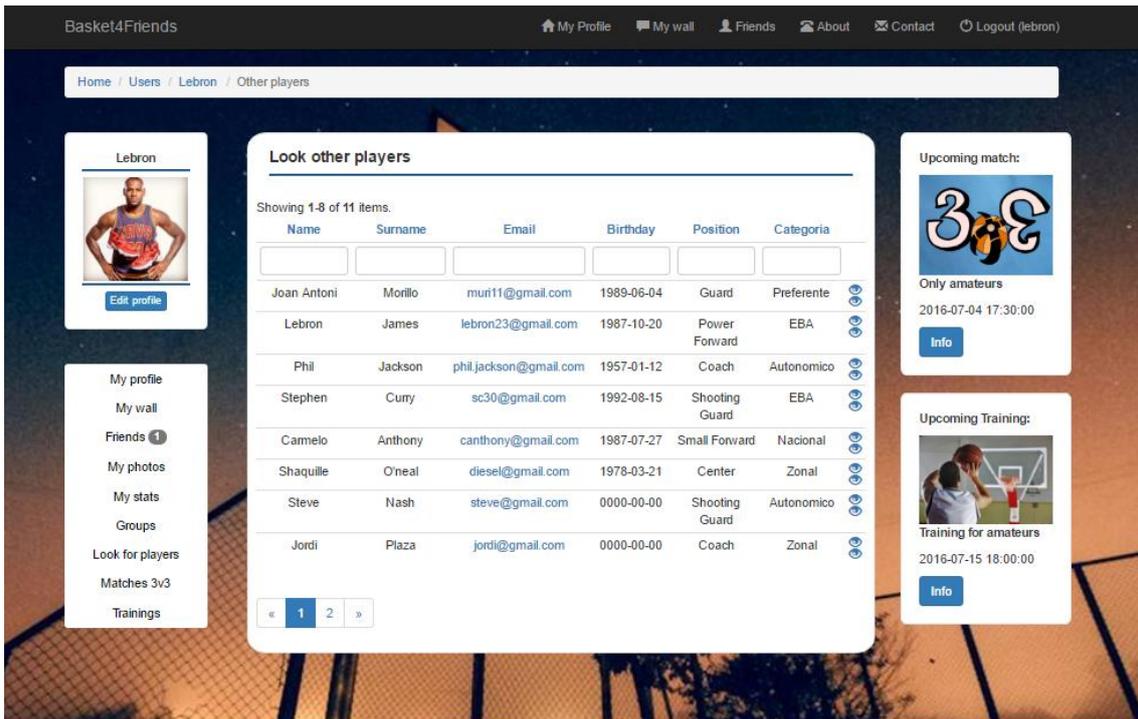
Diseño e implementación de una red social deportiva con el framework Yii



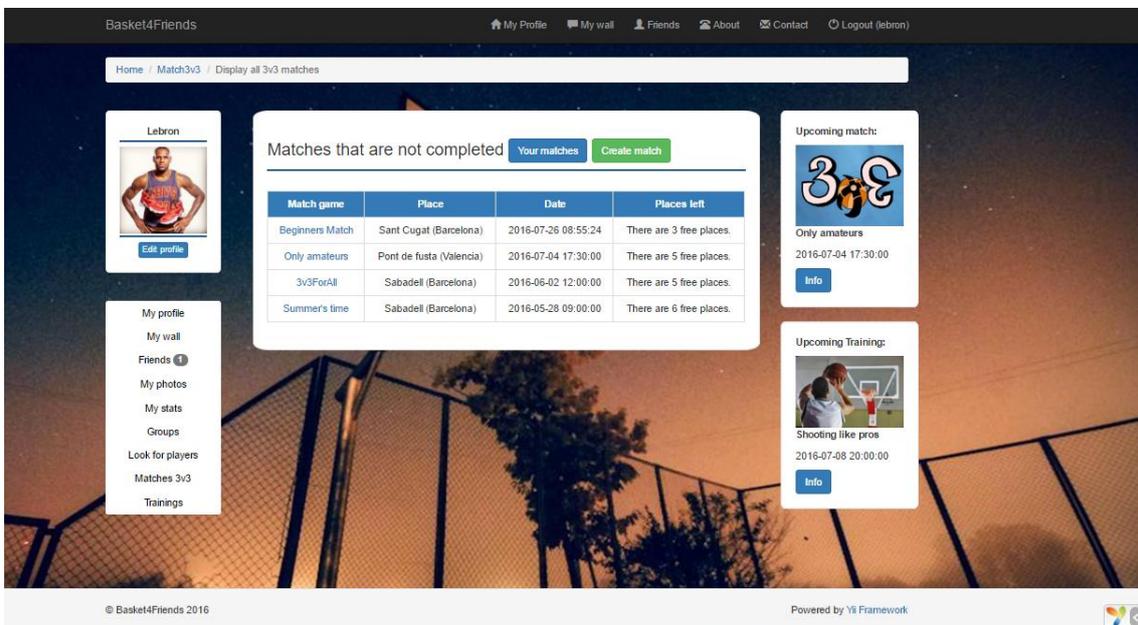
12. Create new group.



13. View group.

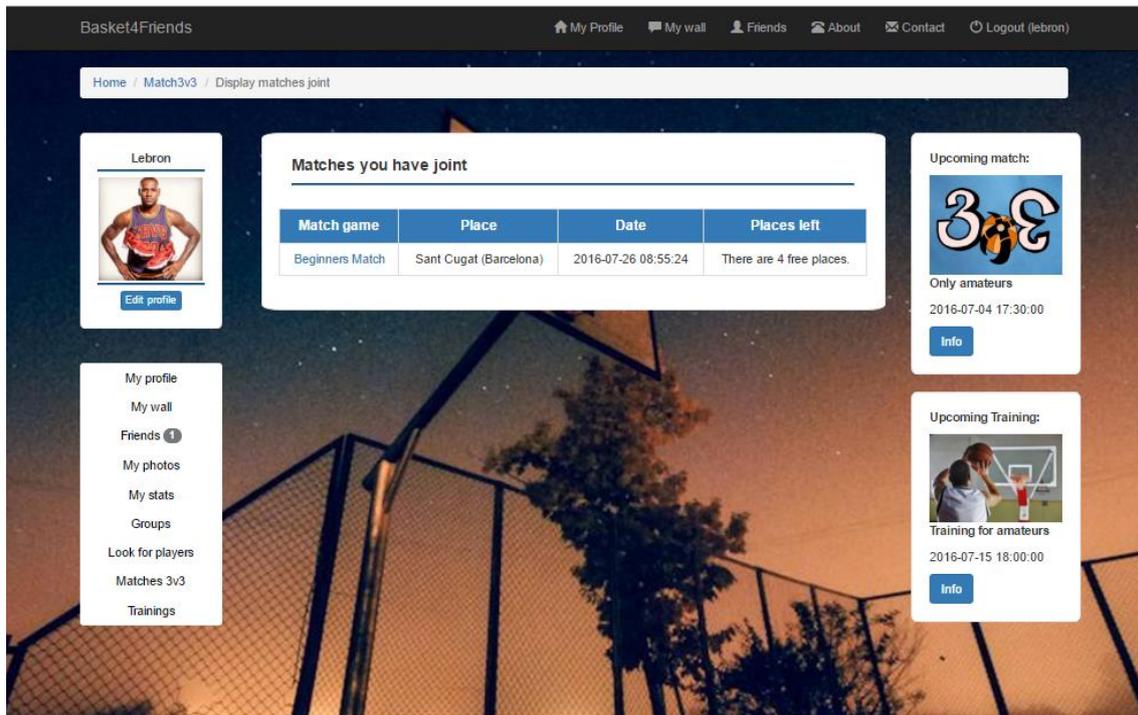


14. Look for players.

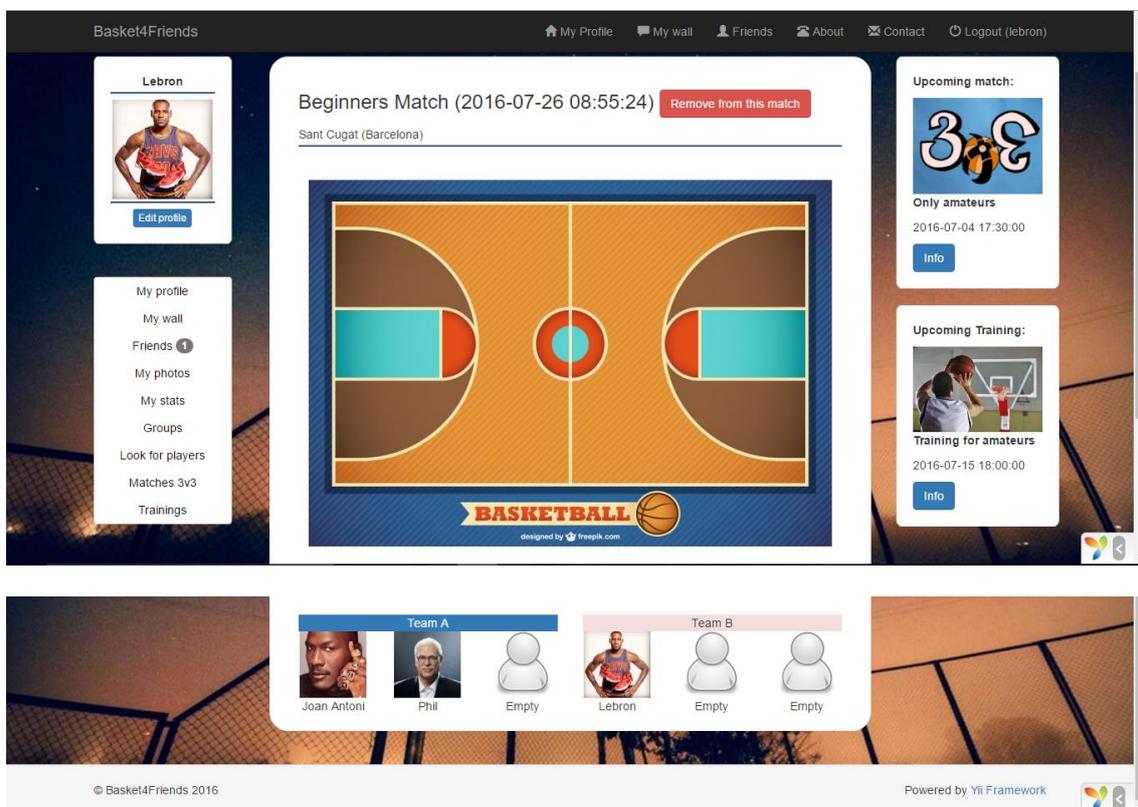


15. View matches.

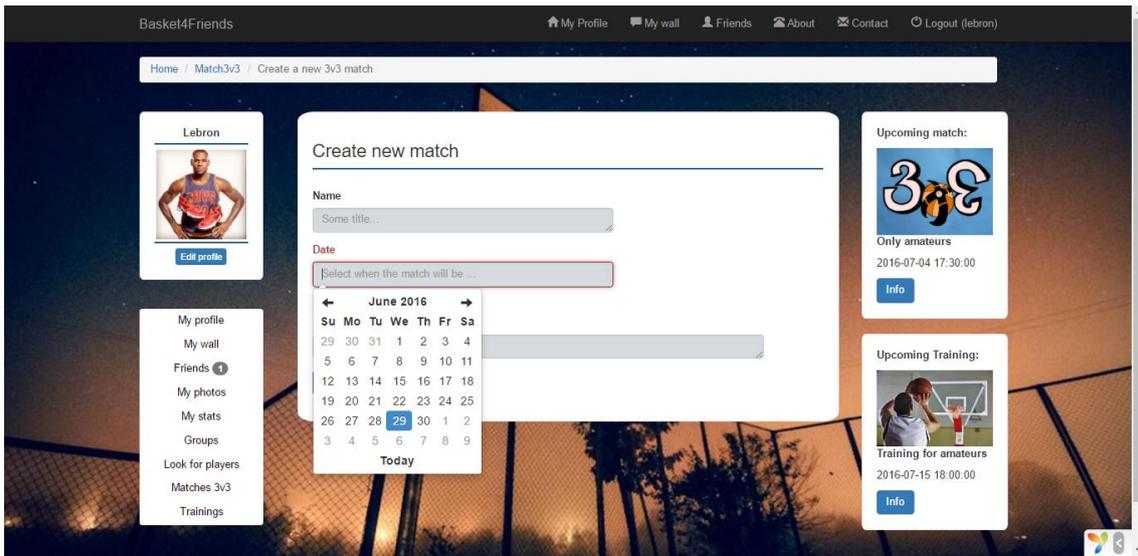
Diseño e implementación de una red social deportiva con el framework Yii



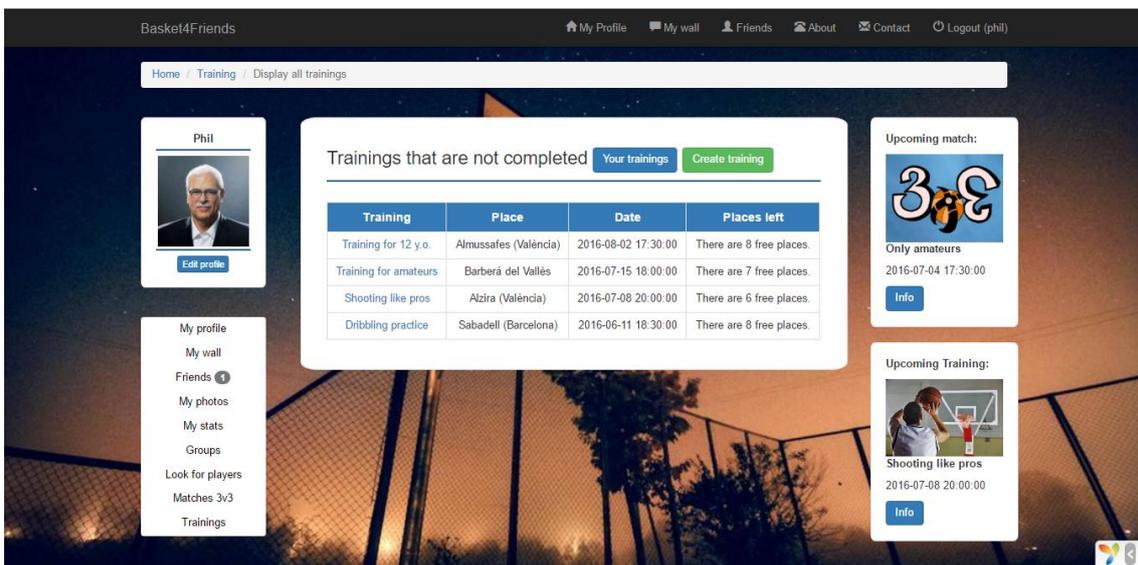
16. Your matches.



17. View match.

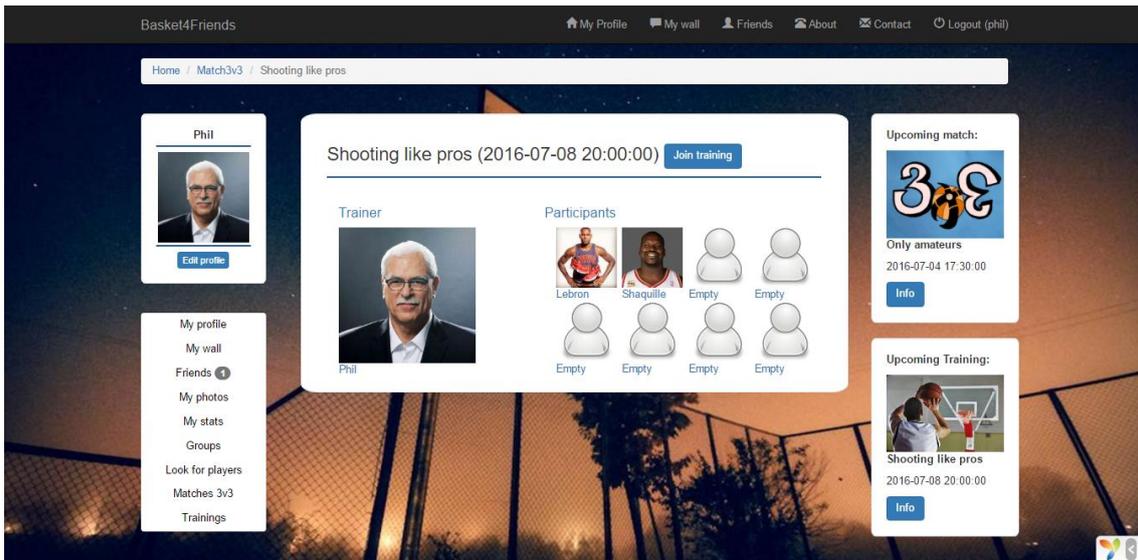


18. Create new match.

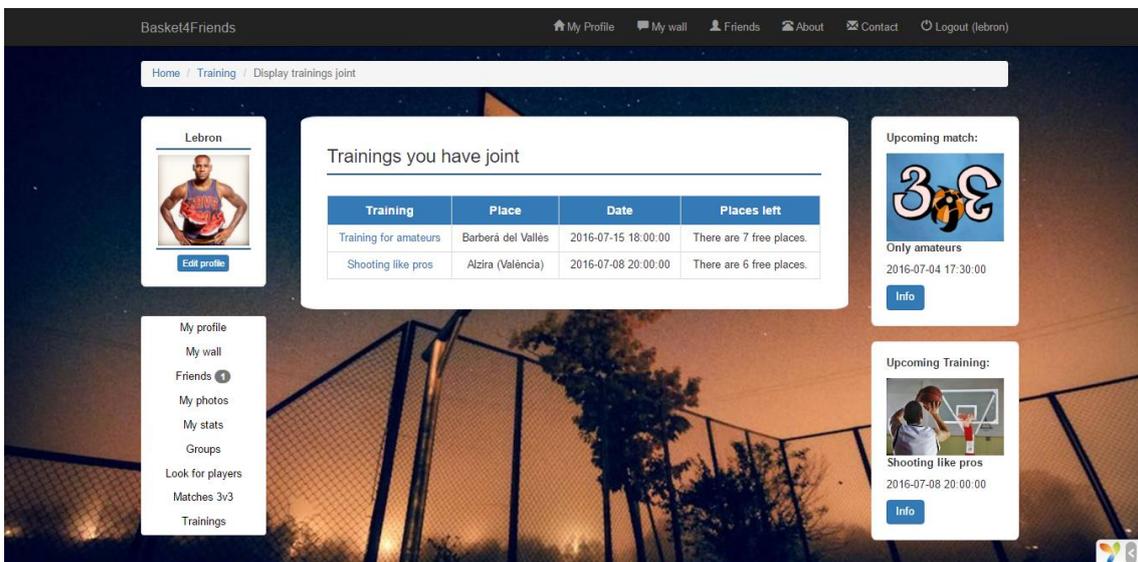


19. View all trainings.

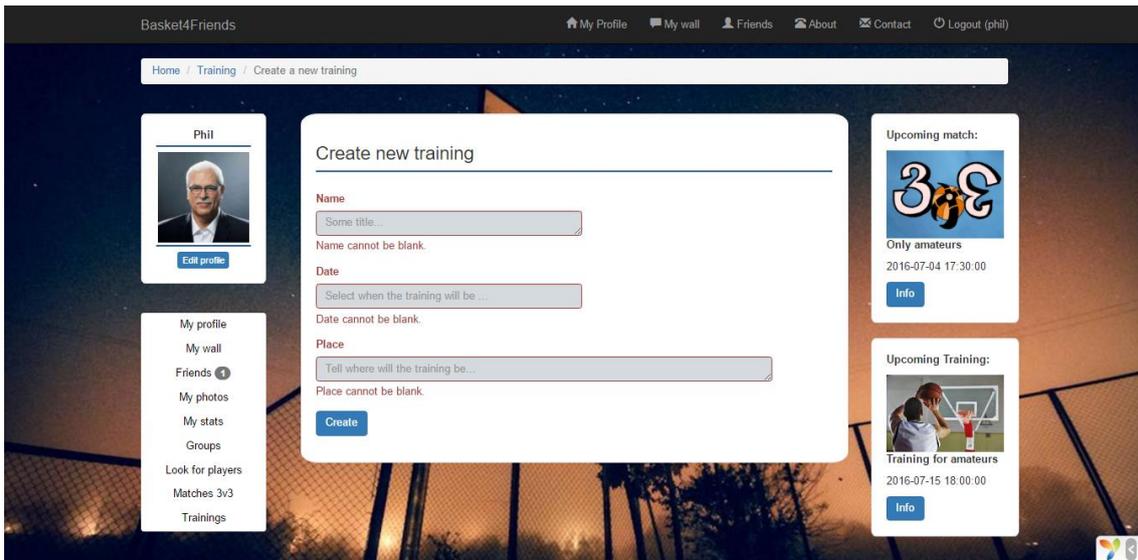
Diseño e implementación de una red social deportiva con el framework Yii



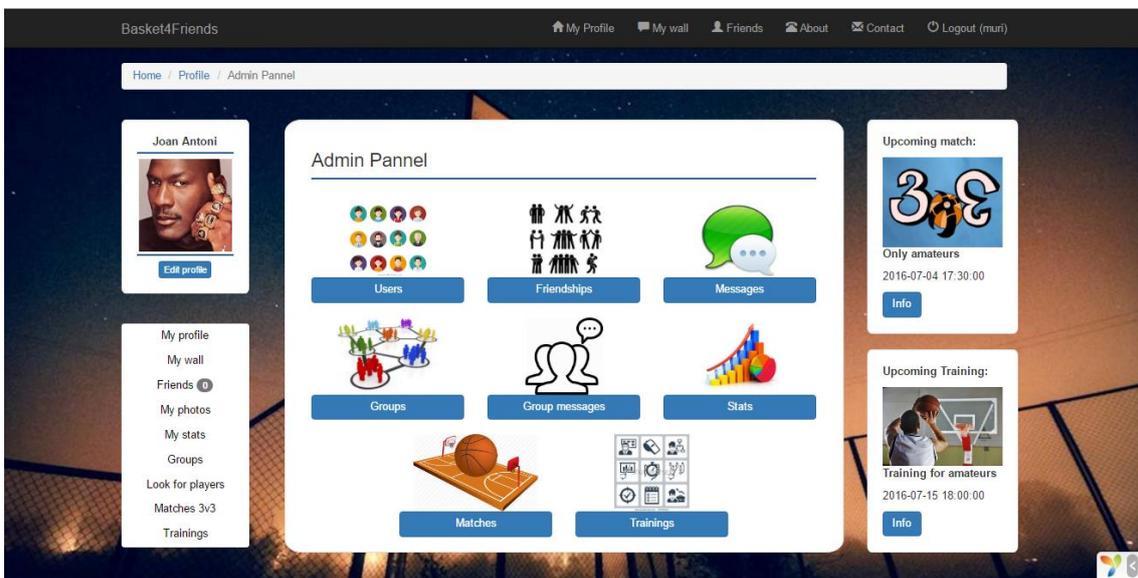
20. View training.



21. My trainings.

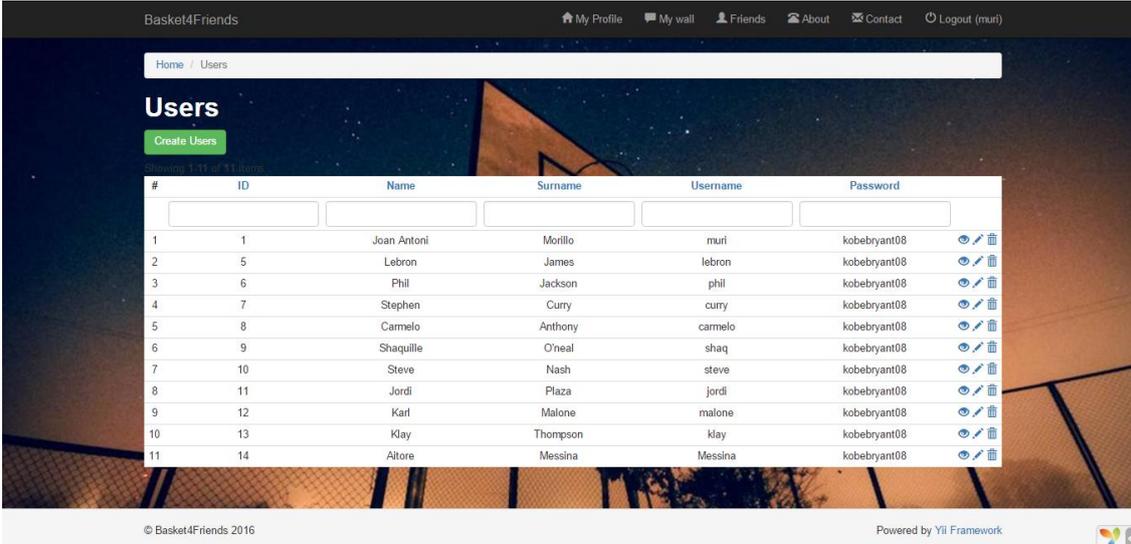


22. Create new training.



23. Admin pannel.

Diseño e implementación de una red social deportiva con el framework Yii

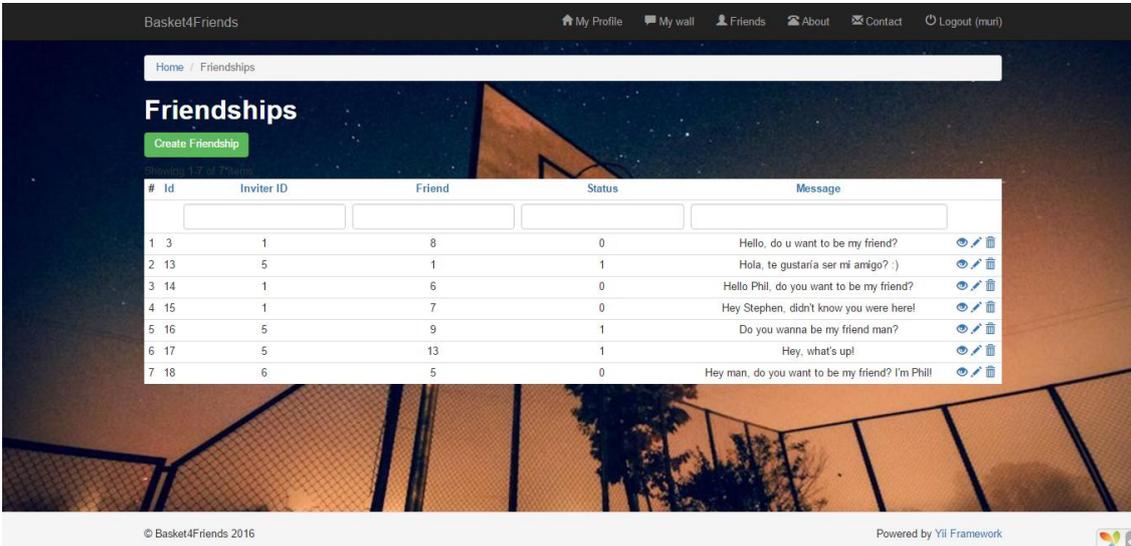


The screenshot shows the 'Users' admin panel in the Basket4Friends application. The page has a dark header with navigation links: My Profile, My wall, Friends, About, Contact, and Logout (muri). Below the header is a breadcrumb trail 'Home / Users' and a 'Create Users' button. The main content is a table with the following data:

#	ID	Name	Surname	Username	Password	
1	1	Joan Antoni	Morillo	muri	kobebryant08	 
2	5	Lebron	James	lebron	kobebryant08	 
3	6	Phil	Jackson	phil	kobebryant08	 
4	7	Stephen	Curry	curry	kobebryant08	 
5	8	Carmelo	Anthony	carmelo	kobebryant08	 
6	9	Shaquille	O'neal	shaq	kobebryant08	 
7	10	Steve	Nash	steve	kobebryant08	 
8	11	Jordi	Plaza	jordi	kobebryant08	 
9	12	Karl	Malone	malone	kobebryant08	 
10	13	Klay	Thompson	klay	kobebryant08	 
11	14	Alto	Messina	Messina	kobebryant08	 

At the bottom of the page, there is a copyright notice '© Basket4Friends 2016' and a footer 'Powered by Yii Framework' with the Yii logo.

24. Admin pannel (Users).

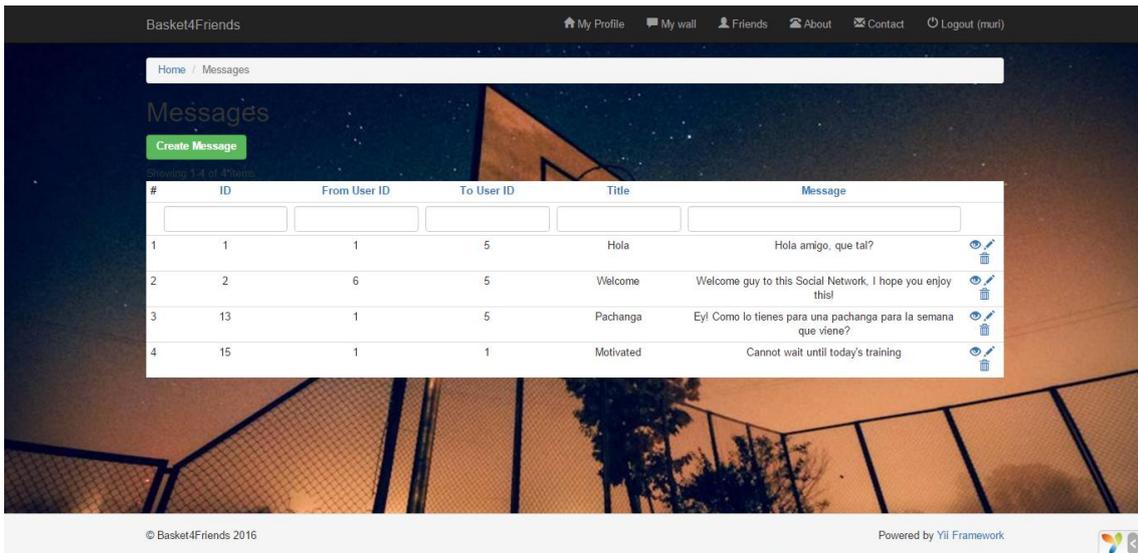


The screenshot shows the 'Friendships' admin panel in the Basket4Friends application. The page has a dark header with navigation links: My Profile, My wall, Friends, About, Contact, and Logout (muri). Below the header is a breadcrumb trail 'Home / Friendships' and a 'Create Friendship' button. The main content is a table with the following data:

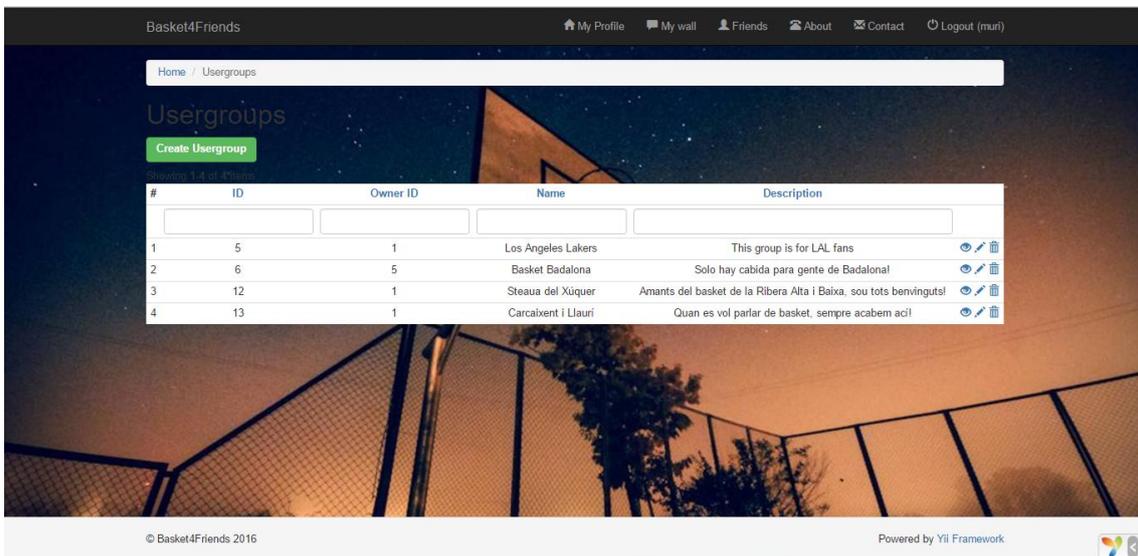
#	Id	Inviter ID	Friend	Status	Message	
1	3	1	8	0	Hello, do u want to be my friend?	 
2	13	5	1	1	Hola, te gustaria ser mi amigo? :)	 
3	14	1	6	0	Hello Phil, do you want to be my friend?	 
4	15	1	7	0	Hey Stephen, didn't know you were here!	 
5	16	5	9	1	Do you wanna be my friend man?	 
6	17	5	13	1	Hey, what's up!	 
7	18	6	5	0	Hey man, do you want to be my friend? I'm Phill!	 

At the bottom of the page, there is a copyright notice '© Basket4Friends 2016' and a footer 'Powered by Yii Framework' with the Yii logo.

25. Admin pannel (Friendships).



26. Admin pannel (Messages).



27. Admin pannel (Groups).

Diseño e implementación de una red social deportiva con el framework Yii

The screenshot shows the 'Usergroupmessages' admin interface. At the top, there is a navigation bar with 'Basket4Friends' and links for 'My Profile', 'My wall', 'Friends', 'About', 'Contact', and 'Logout (muri)'. Below the navigation bar is a breadcrumb trail 'Home / Usergroupmessages' and a 'Create Usergroupmessage' button. The main content area displays a table with columns: '#', 'ID', 'Author ID', 'Group ID', 'Title', and 'Message'. The table contains four rows of data. At the bottom of the page, there is a footer with '© Basket4Friends 2016', 'Powered by Yii Framework', and a small logo.

#	ID	Author ID	Group ID	Title	Message
1	2	1	5	Hello	Hello all, I'm new here!
2	3	5	6	Badalona	Bienvenidos al grupo creado para todos los jugadores que sean de Badalona.
3	4	6	6	Hola a todos!	Hola soy nuevo en el grupo. Alguien que vaya a jugar mañana?
4	5	9	6	Hola Phil	Bienvenido Phil, yo mañana estoy libre. Te unes al partido de only amateurs?

28. Admin pannel (Group messages).

The screenshot shows the 'Stats' admin interface. At the top, there is a navigation bar with 'Basket4Friends' and links for 'My Profile', 'My wall', 'Friends', 'About', 'Contact', and 'Logout (muri)'. Below the navigation bar is a breadcrumb trail 'Home / Stats' and a 'Create Stats' button. The main content area displays a table with columns: '#', 'ID', 'Match', 'User', 'Points', and 'Rebounds'. The table contains six rows of data. At the bottom of the page, there is a footer with '© Basket4Friends 2016', 'Powered by Yii Framework', and a small logo.

#	ID	Match	User	Points	Rebounds
1	1	5	1	9	6
2	2	5	1	6	5
3	3	9	5	9	8
4	4	10	5	11	6
5	5	5	9	8	9
6	6	5	5	6	11

29. Admin pannel (Stats).

Basket4Friends

My Profile My wall Friends About Contact Logout (muri)

Home / Matches3v3

Matches3v3

Create Match3v3

Showing 1-4 of 4 Matches

#	ID	Name	Date	Place
1	5	Only amateurs	2016-07-04 17:30:00	Pont de fusta (Valencia)
2	7	3v3ForAll	2016-06-02 12:00:00	Sabadell (Barcelona)
3	9	Summer's time	2016-05-28 09:00:00	Sabadell (Barcelona)
4	10	Beginners Match	2016-07-26 08:55:24	Sant Cugat (Barcelona)

© Basket4Friends 2016

Powered by Yii Framework

30. Admin pannel (Matches3v3).

Basket4Friends

My Profile My wall Friends About Contact Logout (muri)

Home / Trainings

Trainings

Create Training

Showing 1-4 of 4 Trainings

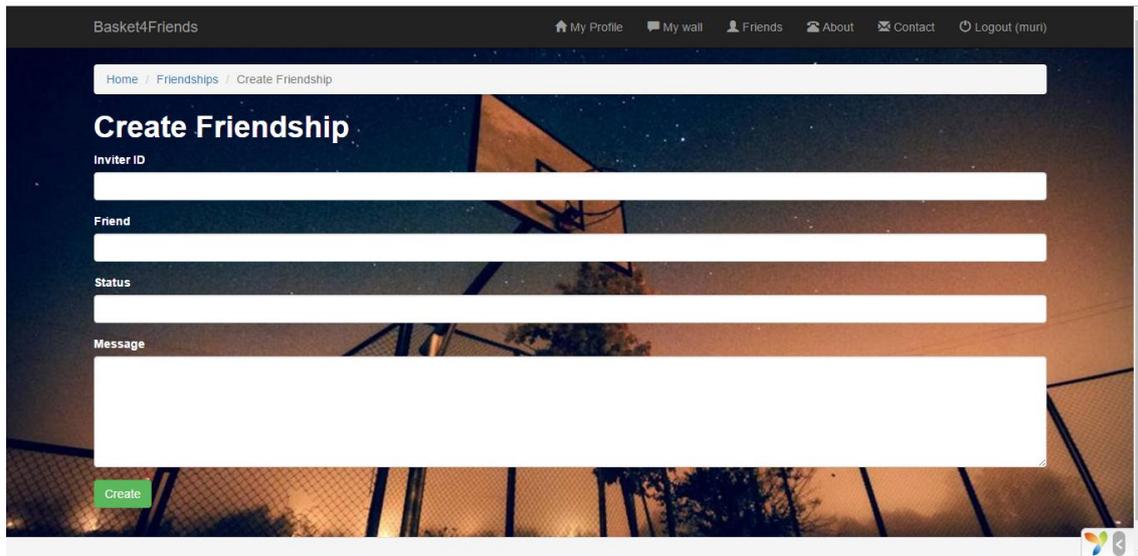
#	ID	Name	Date	Place	Trainer
1	1	Training for amateurs	2016-07-15 18:00:00	Barberá del Vallès	1
2	6	Dribbling practice	2016-06-11 18:30:00	Sabadell (Barcelona)	6
3	7	Shooting like pros	2016-07-08 20:00:00	Alzira (València)	6
4	8	Training for 12 y.o.	2016-08-02 17:30:00	Almussafes (València)	6

© Basket4Friends 2016

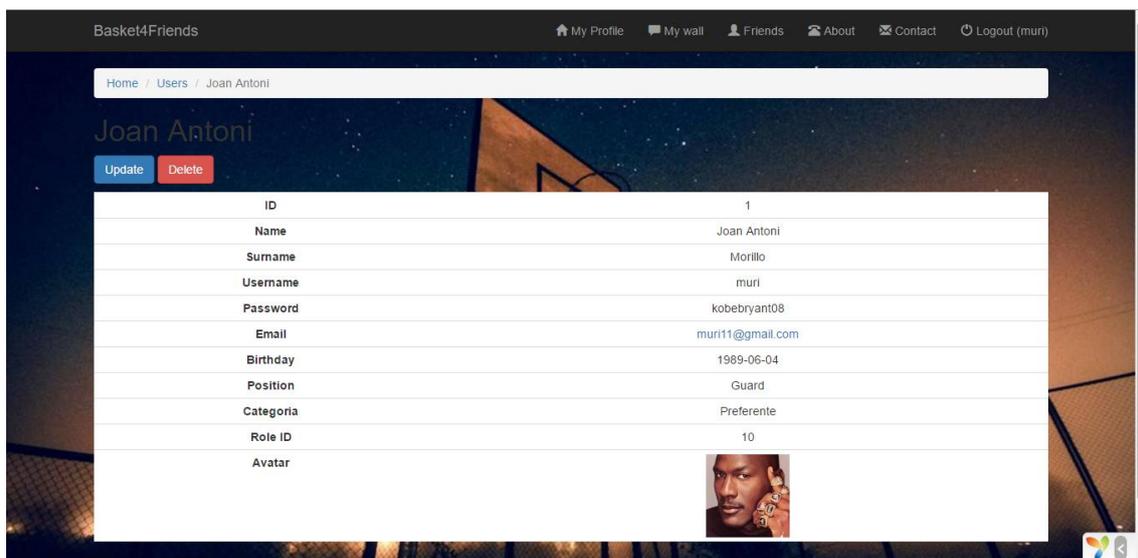
Powered by Yii Framework

31. Admin pannel (Trainings).

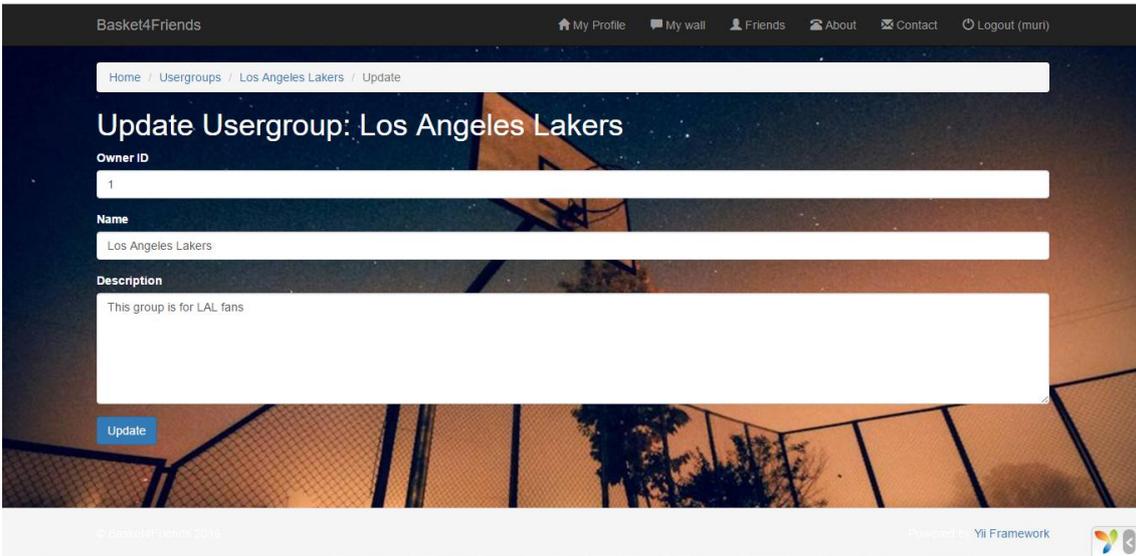
Para evitar la redundancia, a partir de ahora solo se mostrará una imagen para la funcionalidad “create”, “edit” y “view”, ya que siguen la misma estructura (layout y form embebido). Por tanto, podemos imaginar las mismas funcionalidades pero para las demás entidades.



32. Create (Admin).



33. View (Admin).



34. Edit (Admin).

7. Conclusiones

La realización de este proyecto me ha sido de gran utilidad para afianzar los conocimientos adquiridos durante estos años en la universidad. Mi objetivo era profundizar un poco más en el aprendizaje del desarrollo web, dado que lo encuentro un campo muy interesante y extenso. Es por esto que elegí un framework PHP, para poder aprender este lenguaje de programación que hasta ahora era desconocido para mí, en un framework que usara la arquitectura MVC. Creo que ha sido una buena idea elegir dentro de los múltiples campos que hay en el desarrollo web, la creación de una red social, pues pienso que son, hoy en día, una gran herramienta de comunicación entre usuarios. Además, una red social extiende muchas funcionalidades importantes en la red, la adición de amigos, intercambio de mensajes, etc.

Después de haber terminado el proyecto, puedo decir que estoy muy satisfecho con el trabajo realizado y los conocimientos adquiridos, así como del resultado final, que se asemeja bastante a la idea inicial. Ahora me propongo seguir aprendiendo de éstas y más tecnologías, para poder seguir haciendo crecer este proyecto, para que algún día, subirlo a la red y poder servir de llave de comunicación entre todos esos deportistas que necesitan de más gente para poder practicar sus hobbies.

8. Bibliografía

Los elementos bibliográficos que se han consultado para realizar esta memoria han sido mayormente consultas en páginas web, así como de libros sobre el framework Yii en especial. Se va a proceder al listado siguiendo el orden alfabético.

CÓDIGO FACILITO – “Curso de PHP” . *Youtube*

<https://www.youtube.com/playlist?list=PLFo2659CF8418C229> [Consulta: el 10 de diciembre de 2015].

DOINGITEASYCHANNEL – “Yii2 Lessons”. *Youtube*

<https://www.youtube.com/playlist?list=PLRdozhQj3CBmusDbBzFgg3H20VxLx2mkF> [Consulta: el 25 de enero de 2016].

HERREROS RINCÓN, J. (2010) - Desarrollo del portal web de un colegio. Trabajo Final de Carrera. Valencia: Universidad Politécnica de Valencia,

https://riunet.upv.es/bitstream/handle/10251/8576/MemoriaPFC_AntonioMachado_FINAL.pdf, [Consulta: el 18 de mayo de 2016].

JQUERY MANUAL - [Tutorial de Yii2 en español](http://jquery-manual.blogspot.com.es/2015/03/tutorial-de-yii2-en-espanol.html). <[http://jquery-](http://jquery-manual.blogspot.com.es/2015/03/tutorial-de-yii2-en-espanol.html)

[manual.blogspot.com.es/2015/03/tutorial-de-yii2-en-espanol.html](http://jquery-manual.blogspot.com.es/2015/03/tutorial-de-yii2-en-espanol.html)> [Consulta: el 07 de abril de 2016].

KRAJEE (2013) - [DateTimePicker Widget](http://demos.krajee.com/widget-details/datetimepicker). <[http://demos.krajee.com/widget-](http://demos.krajee.com/widget-details/datetimepicker)

[details/datetimepicker](http://demos.krajee.com/widget-details/datetimepicker)> [Consulta: el 05 de abril de 2016].

LINARES PÉREZ, J - Aplicación Web para la gestión de entrenos de deportistas. Trabajo

Final de Carrera. Valencia: Universidad Politécnica de Valencia,

<https://riunet.upv.es/bitstream/handle/10251/9114/memoria.pdf> [Consulta: el 17 de mayo de 2016].

MIELGO OLCINA, L. (2012). Red social de reparaciones de coches. Trabajo Final de Carrera. Valencia: Universidad Politécnica de Valencia,

<https://riunet.upv.es/bitstream/handle/10251/17514/Memoria.pdf>, [Consulta: el 17 de mayo de 2016].

MAKAROV, ALEXANDER (2013). *Yii Application Development Cookbook*, 2nd Edition.

PACKT Publishing

MYBB GROUP. [YiiFrameworkenespañol](http://www.yiiframeworkenespanol.org/foro/). <<http://www.yiiframeworkenespanol.org/foro/>>

[Consulta: el 07 de abril de 2016].

PORTWOOD, CHARLES R. (2014). *Yii Project Blueprints*. Packt Publishing.

<<http://it-ebooks.info/book/4661/>> [Consulta: el 30 de marzo de 2016]

REFSNES DATA. [W3Schools, Bootstrap3 Tutorial](http://www.w3schools.com/bootstrap/default.asp).

<<http://www.w3schools.com/bootstrap/default.asp>> [Consulta: el 06 de mayo de 2016].

REFSNES DATA. [W3Schools, CSS Tutorial](http://www.w3schools.com/css/default.asp). <<http://www.w3schools.com/css/default.asp>>

[Consulta: el 06 de mayo de 2016].

- REFSNES DATA. W3Schools, HTML(5) Tutorial. <<http://www.w3schools.com/html/default.asp>> [Consulta: el 06 de mayo de 2016].
- REFSNES DATA. W3Schools, PHP5 Tutorial. <<http://www.w3schools.com/php/default.asp>> [Consulta: el 18 de diciembre de 2016].
- SAFRONOV, MARK; WINESETT, JEFFREY (2014). Web Application Development with Yii2 and PHP. <<http://it-ebooks.info/book/4641/>> [Consulta: el 09 de abril de 2016].
- STACK EXCHANGE INC. Stack Overflow. <<http://stackoverflow.com/>> [Consulta: el 24 de mayo de 2016].
- STACK EXCHANGE INC. How to upload a file to directory in Yii2. <<http://stackoverflow.com/questions/23592125/how-to-upload-a-file-to-directory-in-yii2>> [Consulta: el 02 de abril de 2016].
- YII SOFTWARE LLC. The definitive guide to Yii 2.0. <<http://www.yiiframework.com/doc-2.0/guide-index.html>> [Consulta: el 22 de enero de 2016].
- YII SOFTWARE LLC. Yiiframework Forum. <<http://www.yiiframework.com/forum/>> [Consulta: el 02 de marzo de 2016].