



## **Aplicación para gestión de actividades deportivas de un club de padel**

### **Manual**

Miguel de la Torre Atrio 53191743B

Francisco López Alonso 45159031A

Carlos Somoza Martínez 45952345R

ABP\_46

## **1. Introducción**

Actualmente el Pádel es un deporte emergente; cada vez son más los jóvenes que empiezan a practicar dicho deporte y con ello son más las empresas que se empiezan a interesar en ofrecer sus servicios para su práctica.

Con ésto, será necesario que toda la información recogida por las empresas encargadas de ofrecer sus servicios para la práctica de la misma, sea gestionado por especialistas en el tema, con el fin de ofrecer al usuario una sencilla e interactiva vista que le facilitará el desarrollo de dicha actividad así como la posibilidad de participar en campeonatos y torneos ofrecidos por la empresa.

## **2. Resumen**

Para el desarrollo del programa hemos hecho un Proceso Unificado (PU), con deportista usuario, administrador y entrenador como actores. Todos ellos con sus respectivas vistas así como la posibilidad de darse de alta y de baja en el programa. Ofrecemos también una vista de las pistas disponibles así como la posibilidad de participar en torneos organizados. Distinguiremos entre diversas categorías (masculinas, femeninas o mixtas) y niveles con sus respectivos rankings.

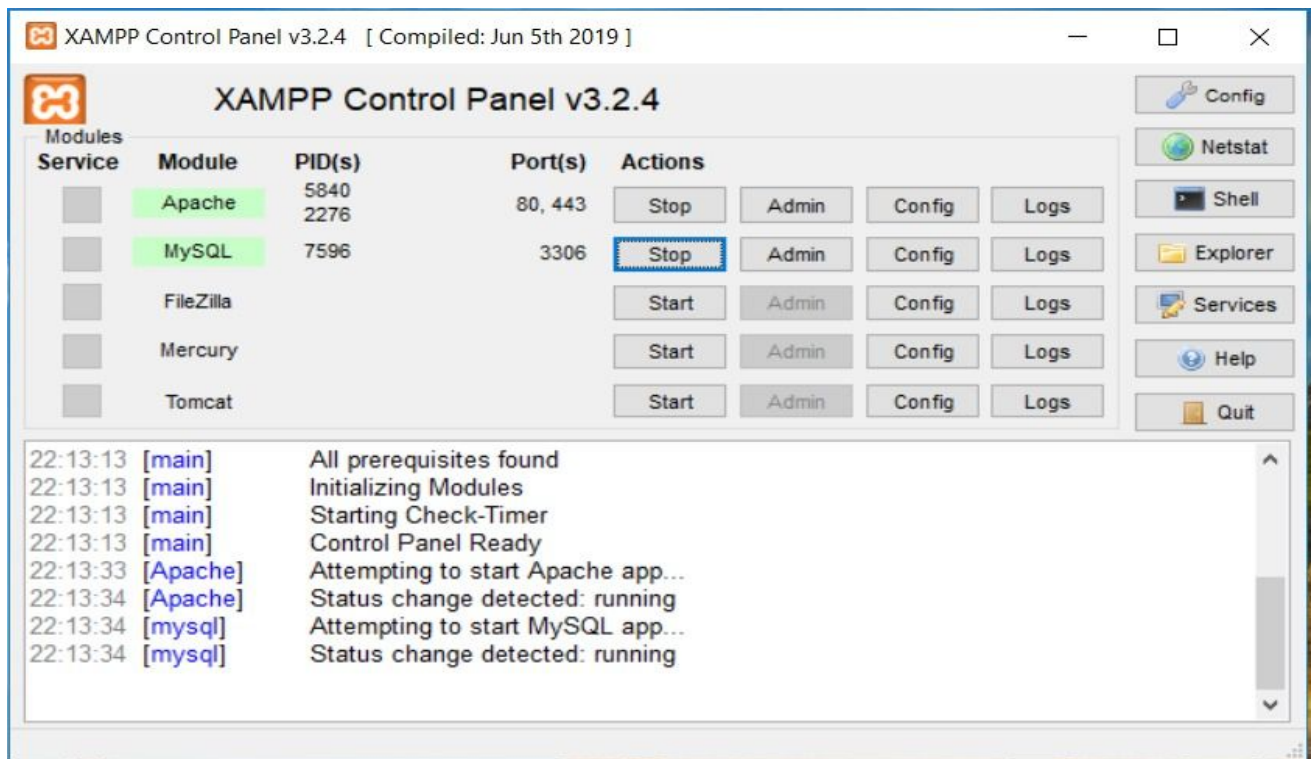
Para la formación de las parejas así como para la consulta del estado de las pistas o la hora de reserva se ha incorporado un chat para que así de forma interactiva los usuarios puedan contactar entre ellos y hacer comentarios y valoraciones de los eventos e instalaciones ofrecidos.

## **3. Primeros pasos e instalación**

Para obtener una visualización y un funcionamiento del proyecto correctos, se requiere una herramienta que permite desplegar un servidor *Apache* y acceder a *PHPMyAdmin* desde la máquina local. En nuestras pruebas hemos utilizado *XAMPP* debido a su sencillez y rapidez de

configuración.

Antes de comenzar es necesario iniciar los procesos *Apache* y *MySQL* en *XAMPP*, como se ilustra a continuación:



Una vez que los procesos se estén ejecutando, es necesario incluir la carpeta con los ficheros del proyecto (**Pádel ESEI**) en el directorio **htdocs** de *XAMPP*.

También hay que importar la base de datos **ABP\_46.sql** a la plataforma *PHPMYAdmin*. Para ello, accedemos al navegador e introducimos la dirección: “localhost/phpmyadmin”, que conduce a la página de gestión de bases de datos. Desde ahí se genera una nueva base de datos con el nombre “abp46” y, después de confirmar su creación, vamos al campo superior “Importar”, donde seleccionamos el archivo correspondiente “ABP\_46.sql”, que contiene las tablas y su información.

Importando al servidor actual

**Archivo a importar:**

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.  
Un archivo comprimido tiene que terminar en `[formato].[compresión]`. Por ejemplo: `.sql.zip`

Buscar en su ordenador:  `ABP_46.sql` (Máximo: 40MB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

**Importación parcial:**

☒ Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.)

Omitir esta cantidad de consultas (en SQL) desde la primera:

**Otras opciones:**

☒ Habilite la revisión de las claves foráneas

**Formato:**

**Opciones específicas al formato:**

Modalidad SQL compatible:

☒ No utilizar `AUTO_INCREMENT` con el valor 0

En cuanto a la información de los usuarios que se insertan en la BD; se les indica a mayores un número del 1 al 4 (`rol_id`) que nos indican la función que van a desempeñar en el sistema.

Se ha creado el usuario *admin* con su contraseña *admin* el cual es el que administra el sistema. En cuanto al resto serán los deportistas, entrenadores y usuarios registrados.

Por último, después de añadir el proyecto a `htdocs` (`C:\xampp\htdocs\Padel_Web`), basta con introducir “`localhost/Padel_Web`” en la barra de direcciones del navegador y se mostrará la página principal de la aplicación web.



## 4. Explicación de los ficheros del proyecto

La aplicación web “Pádel ESEI” se basa en el Modelo-Vista-Controlador (MVC), lo cual se ve reflejado en la estructura de directorios (Controllers, Models, Views,...). En la carpeta “Controllers” encontramos los archivos que responden a las acciones del usuario e

interactúan con el modelo; dentro de “Models”, figuran los archivos que gestionan el acceso a la información de la base de datos (mediante consultas), y por último, en “Views” se hallan los archivos que presentan el modelo al usuario para su utilización.

Para poder visualizar el contenido de estos ficheros y el sql es necesario un editor de texto o código fuente, en nuestro caso hemos utilizado Sublime Text.

- **Controllers**

### **Controllers de Login/Registro:**

-“*Index\_Controller*”: Si el usuario está autenticado, le muestra la página principal.

-“*Login\_Controller*”: Si la información introducida existe en la base de datos, el usuario se loguea. En cambio, si este no existe, se muestra un mensaje de error.

-“*Register\_Controller*”: Se recoge la información que el usuario introduce para registrarse. Si el usuario ya existe, es decir, ya se ha registrado anteriormente, se muestra un mensaje de error.

-“*User\_Controller*”: Posee la acción para visualizar todos los usuarios registrados.

### **Controller de Campeonato:**

-“*Championship\_Controller*”: Incluye las acciones pertinentes para añadir y eliminar un campeonato, además de visualizar todos los campeonatos existentes como acción predeterminada.

### **Controller de Chat:**

-“*Chat\_Controller*”: Si el usuario está logueado incluye las acciones pertinentes para añadir un mensaje así como para borrarlo.

### **Controller de Partido:**

-“*Match\_Controller*”: Incluye las acciones de insertar, promocionar

partido así como la muestra del calendario, promociones y la vista detallada. También nos permite inscribirnos así como editar la información o borrarla si fuese necesario.

### **Controller de Pago:**

-*"Payment\_Controller"*: Mediante el pago nos permite hacer una reserva así como editarla, borrarla y mostrarnos de ella una vista detallada.

### **Controller del Post:**

-*"Post\_Controller"*: Nos permite añadir, editar y borrar un post así como nos ofrece una vista detallada del mismo.

### **Controller del Enfrentamiento:**

-*"Clash\_Controller"*: Nos permite editar un enfrentamiento así como consultar el ranking para saber cómo va nuestra pareja.

### **Controller de la Pista:**

-*"Court\_Controller"*: Nos permite añadir una pista junto su descripción; así como buscar una mediante su id. También nos permite editar su información o borrarla si fuese necesario. A mayores también nos ofrece poder mostrarnos las pistas.

### **Controller de Pareja:**

-*"Couple\_Controller"*: Nos permite dar una muestra detallada de las parejas, así como los campeonatos en los que están inscritos.

## **● Models**

### **Model de Campeonato:**

-*"Championship\_Model"*: Nos permite insertar datos de la categoría a la que pertenece una pareja así como los niveles y los grupos. También nos permite añadir los datos de un nuevo campeonato, editarlo y borrarlo. A mayores éste model también se encarga de hacer las combinaciones de pareja; obtener información de dicho campeonato y las parejas; y controlar

el número máximo de parejas que pueden participar en éste.

### **Model de Chat:**

- "*Chat\_Model*": Nos permite insertar datos de un chat así como la posibilidad de acceder a todos ellos.

### **Model de Enfrentamientos:**

- "*Clash\_Model*": Nos permite añadir datos de un enfrentamiento; así como la posibilidad de consultarlos por categoría y nivel o consultarlos todos. A mayores estos datos se pueden editar y eliminar. En este model también se nos permite hacer una consulta la información sobre el ranking de la pareja (partidos jugados, ganados, perdidos) a partir de su categoría y nivel.

### **Model de Clase:**

- "*Clash\_Model*": Nos permite añadir los datos de una nueva clase; así como la posibilidad de borrar esos datos o editarlos.

### **Model de Pareja/Categoría:**

- "*Couple\_Categoria\_Model*": Nos permite añadir los datos de una nueva pareja con su respectiva categoría así como la posibilidad de consultar dichas parejas.

### **Model Pareja/Campeonato:**

- "*Couple\_Championship\_Model*": Nos permite añadir los datos de una nueva pareja para un campeonato; así como la posibilidad de consultar información de dichas parejas.

### **Model para Pareja/Grupo:**

- "*Couple\_Grupo\_Model*": Muestra los datos de los grupos establecidos; así como la posibilidad de añadir los datos de una nueva pareja al grupo. Con todo, también podemos consultar información sobre las parejas de los grupos.

### **Model para Pareja:**

-*"Couple\_Model"*: Nos ofrece la posibilidad de insertar datos de una nueva pareja; así como la posibilidad de consultar información sobre ellas. También nos permite consultar la información de sus respectivos campeonatos.

### **Model para Pareja/Nivel:**

-*"Couple\_Nivel\_Model"*: Nos ofrece la posibilidad de añadir los datos de una nueva pareja con un nivel determinado así como la posibilidad de consultar información de dichas parejas.

### **Model para las Pistas:**

-*"Court\_Model"*: Nos permite la posibilidad de insertar los datos de una nueva pista. También podemos editar los datos de la misma o eliminarlos.

### **Model para el Género:**

-*"Gender\_Model"*: Nos ofrece la posibilidad de añadir datos desde la base de datos, o desde la categoría de alguien de un género.

### **Model para Grupo:**

-*"Group\_Model"*: Nos ofrece la posibilidad de rellenar grupo desde la base de datos.

### **Model para Partido:**

-*"Match\_Model"*: Si el usuario está logueado nos permite introducir los datos de una nueva promoción y partido; así como editar y eliminarla en caso de que sea preciso. Además podemos hacer una consulta de los partidos disponibles y de los jugadores inscritos.

### **Model para Post:**

-*"Post\_Model"*: Nos permite añadir información de un nuevo post más la posibilidad de editarlo o borrarlo si fuera pertinente.

### **Model para Ranking:**



-*"Ranking\_Model"*: Nos ofrece la posibilidad de añadir la información sobre un nuevo ranking; así como la consulta de los mismos, en este model también se podrá modificar los resultados y establecer empates si fuese necesario.

### **Model para Reserva:**

-*"Reservation\_Model"*: Nos ofrece la posibilidad de añadir los datos sobre una nueva reserva; también podremos consultar las reservas y editarlas o eliminarlas por si fuera necesario.

### **Model para las Reglas:**

-*"Rule\_Model"*: Nos ofrece la posibilidad de ingresar infor sobre las reglas también con la ayuda de la BD que hemos implementado.

### **Model para Usuario Game:**

-*"User\_Game\_Model"*: Nos ofrece la posibilidad de borrar la promoción a la que estamos inscritos.

### **Model para Usuario:**

-*"USER\_MODEL"*: Nos ofrece la posibilidad de introducir los datos de un nuevo usuario; así como la posibilidad de editarlo y borrarlo.

## **● Vistas**

### **Vista Campeonato:**

-*"CHAMPIONSHIP\_VIEWS"*: Muestra una tabla de la clasificación.

-*ADD\_VIEW*: Vista para añadir campeonato.

-*DELETE\_CHAMPIONSHIP\_VIEW*: Vista para eliminar campeonato.

-*EDIT\_VIEW*: Vista para editar campeonato.

-*InscribirCampeonatoView*: Vista para inscribir en campeonato.

-*SHOW\_COUPLE*: Vista para mostrar la pareja del campeonato.

-*SHOWALL\_VIEW*: Vista completa del campeonato.

-*SHOWCURRENT\_COUPLE*: Vista detallada de las parejas del

campeonato.

- SHOWCURRENT\_VIEW*: Vista detallada del campeonato

- SHOWGRUPOS\_CAMPEONATO*: Vista de los grupos del campeonato.

### **Vista del Chat:**

- CHAT\_VIEW*: Muestra la vista del Chat.

- SHOWALL\_VIEW*: Vista completa del chat.

### **Vista de Enfrentamiento:**

- CLASH\_VIEWS*: Muestra una tabla de enfrentamiento.

- CLASH\_SHOWALL*: Vista completa del enfrentamiento.

- EDIT\_VIEW*: Vista para editar enfrentamiento.

- SHOWRANKING*: Vista del ranking en el enfrentamiento.

### **Vista de la Pista:**

- COURT\_VIEWS*: Muestra la tabla con las pistas.

- ADD\_VIEW*: Vista para añadir pista.

- DELETE\_VIEW*: Vista para borrar pista.

- EDIT\_VIEW*: Vista para editar pista.

- SEARCH\_VIEW*: Pista para buscar pista.

- SHOWALL\_VIEW*: Vista para ver la pista.

- SHOWCURRENT\_VIEW*: Vista detallada de la pista.

### **Vista del Partido:**

- MATCH\_VIEWS*: Muestra la tabla con los partidos.

- ADD\_VIEW*: Vista para añadir partido.

- DELETE\_MATCH\_VIEW*: Vista para eliminar partido

- EDIT\_VIEW*: Vista para editar partido.

- Inscripcion\_View*: Vista para inscribirse en partido.

- SCHEDULE\_PROMOTION*: Vista para ver el calendario del partido.

- SEARCH\_VIEW*: Vista para buscar partido.

- SHOWALL\_VIEW*: Vista para ver la partido.

- SHOWCURRENT\_PROMOCION*: Vista para ver en detalle la

promoción del partido.

-*SHOWCURRENR\_VIEW*: Vista detallada de la partido.

### **Vista del Post:**

-*"POST\_VIEWS"*: Muestra la vista de los post.

-*ADD\_VIEW*: Vista para añadir post.

-*EDIT\_VIEW*: Vista para editar post.

-*SEARCH\_VIEW*: Vista para buscar post.

-*SHOWALL\_VIEW*: Vista para ver la post.

-*SHOWCURRENT\_VIEW*:

### **Vista del Ranking:**

-*"RANKING\_VIEWS"*: Muestra la tabla con los rankings.

-*SHOWCURRENT.*: Vista detallada del ranking.

### **Vista de la Reserva:**

-*"RESERVATIONO\_VIEWS"*: Muestra una tabla con las reservas.

-*ADD\_VIEW*: Vista para añadir reserva

-*SEARCH\_VIEW*: Vista para buscar reserva.

-*EDIT\_VIEW*: Vista para editar la reserva.

-*SHOWALL\_VIEW*: Vista para ver todas las reservas.

-*SHOWCURRENT\_VIEW*: Vista detallada de la reserva

-*SHOWSCHEDULE*: Vista del calendario de la reserva.

### **Vista de las Reglas:**

-*"RULE\_VIEWS"*: Muestra la vista de las reglas.

-*SHOWRULE*: Vista de las reglas.

### **Vista de Usuario:**

-*"USER\_VIEWS"*: Muestra la tabla con los usuarios.

-*ADD\_VIEW*: Vista para añadir usuario.

-*DELETE\_VIEW*: Vista para borrar usuario.

-*SEARCH\_VIEW*: Vista para buscar usuario.

-*EDIT\_VIEW*: Vista para editar usuario.

-*SHOWALL\_VIEW*: Vista para ver los usuarios.

-*SHOWCURRENR\_VIEW*: Vista detallada del usuario.

## **Conclusión:**

Con este proyecto hemos mejorado nuestros conocimientos sobre manejo de repositorios por medio de GitHub; así como en los lenguajes de programación *mySQL*, *Javascript*, *HTML*, *CSS*. Así como el uso del paquete de software libre para la gestión de datos *XAMPP*.

También nos ha ayudado mucho a mejorar nuestra mecánica de trabajo con el reparto de tareas y la puesta en común de problemas para realizar el proyecto de manera más eficiente.