

IES La Arboleda

Proyecto final de ciclo

**Desarrollo de una aplicación
para la gestión de un club de
baloncesto**

Autor: Javier Cordero Toscano

Ciclo: Desarrollo de Aplicaciones Web

Curso: 2020-2021

Resumen

El objetivo del proyecto es crear una aplicación que permita la gestión de la cantera de un club de baloncesto, teniendo dos aplicaciones claramente diferenciadas dentro del proyecto, una página web pública a todo el mundo y una aplicación de gestión del club.

La página web pública tiene como objetivo informar sobre el club, mostrando los equipos, entrenadores y jugadores que pertenecen al club. Además, se muestran los partidos ya jugados, con las estadísticas, y los partidos por jugar.

La aplicación administrativa permite gestionar las diferentes entidades que forman un club: los equipos, los entrenadores y los jugadores.

La aplicación permite, también, registrar los partidos que juegan los equipos del club y registrar las estadísticas de los jugadores que han participado en ellos.

Además, guarda un registro histórico de todos los equipos por los que han ido pasando los diferentes jugadores y entrenadores que pertenecen al club.

La aplicación tiene tres tipos de usuarios diferentes: administrador, entrenador y jugador.

El administrador es el encargado de gestionar las altas, bajas y modificaciones de los equipos, jugadores y entrenadores. Además, podrá gestionar los partidos de todos los equipos del club.

El entrenador es el encargado de gestionar sus equipos, con todo lo que conlleva: convocatorias a partidos, evolución de jugadores, calendario de entrenamientos, ...

El jugador puede enviar y recibir mensajes del resto de usuarios para estar informado de las novedades.

Índice

Resumen	2
1. Introducción	5
1.1. Motivación del proyecto	5
1.2. Objetivos	5
2. Estado del arte	6
2.1. Aplicaciones similares	6
2.2. Conclusiones	6
3. Análisis	8
3.1. Usuarios de la aplicación	8
3.2. Requisitos funcionales	8
3.3. Casos de uso	9
4. Diseño	24
4.1. Base de datos	24
4.2. Diagrama de clases	27
5. Tecnologías utilizadas	28
5.1. Lenguajes	28
5.2. Base de datos	28
5.3. Frameworks y librerías	29
5.4. Control de versiones	29
5.5. Software	30
6. Desarrollo	32
6.1. Estructura	32
6.2. Diseño de las interfaces	38
7. Conclusiones	43
7.1. Aprendizaje	43
8. Trabajo futuro	44
9. Bibliografía	45
Anexo I: Manual de instalación	46
1. Instalación de Laragon	46

2.	Instalación de la base de datos	47
3.	Ejecutar la aplicación.....	49
Anexo II: Manual de usuario		50
1.	Página web pública	50
2.	Aplicación administrativa.....	57

1. Introducción

1.1. Motivación del proyecto

La idea del proyecto nace de mi trabajo como entrenador en la cantera del Club Baloncesto Lepe Alius. Durante mi estancia en el club he observado que la gestión se lleva mediante unos métodos bastante anticuados: hojas de Excel, documentos de Word y anotaciones a mano. Gracias a la aplicación desarrollada podremos aunar toda la documentación, evitando así la pérdida de información que puede suponer el tener esta documentación en diferentes archivos. Además, modernizar y hacer más sencillas las tareas que conllevan la gestión de un club.

1.2. Objetivos

El objetivo del proyecto es crear una aplicación que permita la gestión de la cantera de un club de baloncesto y una página web pública que informe sobre el club. Los principales objetivos del proyecto son los siguientes:

- Gestionar los datos de las entidades que componen el club: equipos, entrenadores y jugadores.
- Asignar jugadores y entrenadores a los equipos.
- Guardar un histórico de los equipos por los que han pasado los jugadores y entrenadores.
- Gestionar los horarios de entrenamientos de los equipos.
- Gestionar el calendario de partidos de los equipos.
- Guardar las estadísticas de los jugadores en los partidos disputados por los equipos.
- Registrar la evolución de los jugadores.
- Dotar a los componentes del club de una herramienta de comunicación interna.
- Mostrar la información del club en una página web estética y accesible.

2. Estado del arte

2.1. Aplicaciones similares

Se ha realizado una búsqueda exhaustiva por la web para lograr encontrar aplicaciones similares a la planteada en el proyecto. A continuación, se detallan algunas de las que se han encontrado.

2.2.1. Esportter

Es una aplicación de gestión de clubes deportivos, no está especificada para ningún deporte en concreto, sino que abarca diferentes deportes como el fútbol, el baloncesto y el fitness.

Se enfoca mucho en la inscripción de los usuarios, la gestión del pago de cuotas y la comunicación con los usuarios.

Además, cuenta con una herramienta para controlar los calendarios de entrenamientos y partidos de los diferentes equipos, y permite integrar una tienda online dentro de la aplicación.

El precio oscila de 35€ a 115€ según el número de equipos y deportistas.

2.2.1. SportMember

Es una aplicación de gestión de clubes deportivos, aunque no está enfocada a ningún deporte específico parece más centrada en el fútbol que en el resto de deportes.

Permite la gestión de múltiples equipos y jugadores, además de otorgar un sistema de gestión de cuotas y emisión de facturas de manera rápida.

Cuenta también con la gestión de calendarios personalizables que permiten añadir actividades y entrenamientos al mismo.

Por último, cabe destacar que tiene una aplicación móvil disponible tanto para Android como para Apple.

Tiene diferentes planes de precios, desde uno básico que no permite muchas de las funcionalidades interesantes, hasta el plan Pro, con un precio de 0,15€ al mes por usuario + 0,40€ + 1.0% de comisión por los pagos hechos desde la aplicación.

2.2. Conclusiones

Al buscar aplicaciones similares a la del proyecto se ha podido observar que las aplicaciones encontradas no están enfocadas a ningún deporte en concreto, sino que se ofrecen como aplicaciones de gestión de clubes deportivos. Estas aplicaciones están muy centradas también en la gestión de cuotas y de la tienda online del club, dándole demasiada importancia a un tema que, desde el punto de vista del proyecto,

sería secundario a la hora de gestionar un club, pues la mayor preocupación debería ser los jugadores y equipos que conforman el club. Las aplicaciones encontradas son bastantes potentes y cuentan con multitud de funcionalidades, pero quizás los precios que ofertan se pueden ir un poco de mano para clubes humildes como son los de nuestro entorno.

3. Análisis

El análisis del proyecto nos sirve para examinar cada uno de los puntos que componen el proyecto teniendo en cuenta las necesidades del mismo.

3.1. Usuarios de la aplicación

La necesidad de diferenciar roles dentro de una aplicación se debe, sobre todo, a la seguridad de la aplicación. Necesitamos delimitar los niveles de acceso, de manera que cada usuario pueda manipular la parte del programa que debe. A continuación, detallamos los diferentes roles que existen en la aplicación.

Administrador	Tiene permiso total para gestionar las altas, bajas y modificaciones de las diferentes entidades del proyecto. Encargado de crear el resto de usuarios de la aplicación
Entrenador	Tiene permiso para gestionar sus equipos.
Jugador	Tiene permiso para enviar y recibir mensajes.

3.2. Requisitos funcionales

3.2.1. Gestión de entrenadores

- Un administrador podrá dar de alta y modificar los entrenadores.
- Un administrador podrá asignar entrenadores a diferentes equipos.

3.2.2. Gestión de jugadores

- Un administrador podrá dar de alta y modificar los jugadores.
- Un entrenador podrá actualizar los atributos de los jugadores pertenecientes a sus equipos.

3.2.4. Gestión de equipos

- Un administrador podrá dar de alta, baja y modificar los equipos.
- Un administrador podrá gestionar los horarios de todos los equipos del club.
- Un entrenador podrá gestionar los horarios de sus equipos.
- Un administrador podrá asignar y quitar jugadores de un equipo.
- Un administrador podrá mover jugadores de un equipo a otro.
- Un entrenador podrá asignar y quitar jugadores a sus equipos.

3.2.4. Gestión de partidos

- Un administrador podrá dar de alta, baja y modificar partidos de todos los equipos del club.
- Un entrenador podrá dar de alta partidos de sus equipos.
- Un entrenador podrá hacer la convocatoria de jugadores para los partidos de sus equipos.

3.3. Casos de uso

El modelo de casos de uso especifica la funcionalidad que el sistema ha de ofrecer desde la perspectiva de los usuarios y lo que el sistema ha de realizar para satisfacer las peticiones de estos usuarios. Se han identificado los siguientes Casos de Uso según los tipos de usuario que pueden acceder a la aplicación.

3.1.1. Actor Visitante

No es un usuario de los definidos anteriormente, sino que corresponde a cualquier persona que visite la página web.

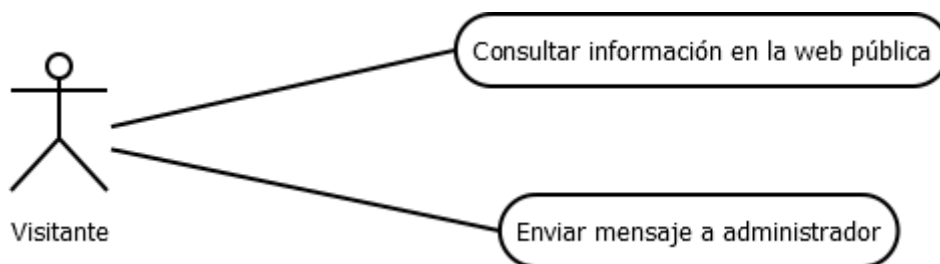


Figura 1 – Casos de uso Visitante

Descripción de los casos de uso

- **Consultar información en la web pública:** El usuario navegar por la página web pública, pudiendo ver datos como el calendario de partidos, resultados de partidos, listado de equipos y listado de jugadores.
 - **Precondiciones:** No aplica.
 - **Postcondiciones:** No aplica.
- **Enviar mensaje a administrador:** El usuario puede enviar un mensaje al administrador desde el formulario de contacto, aportando su nombre y email.
 - **Precondiciones:** No aplica.

- **Postcondiciones:** Se guarda un registro en la tabla Usuario-Mensajes.

3.1.2. Actor Jugador

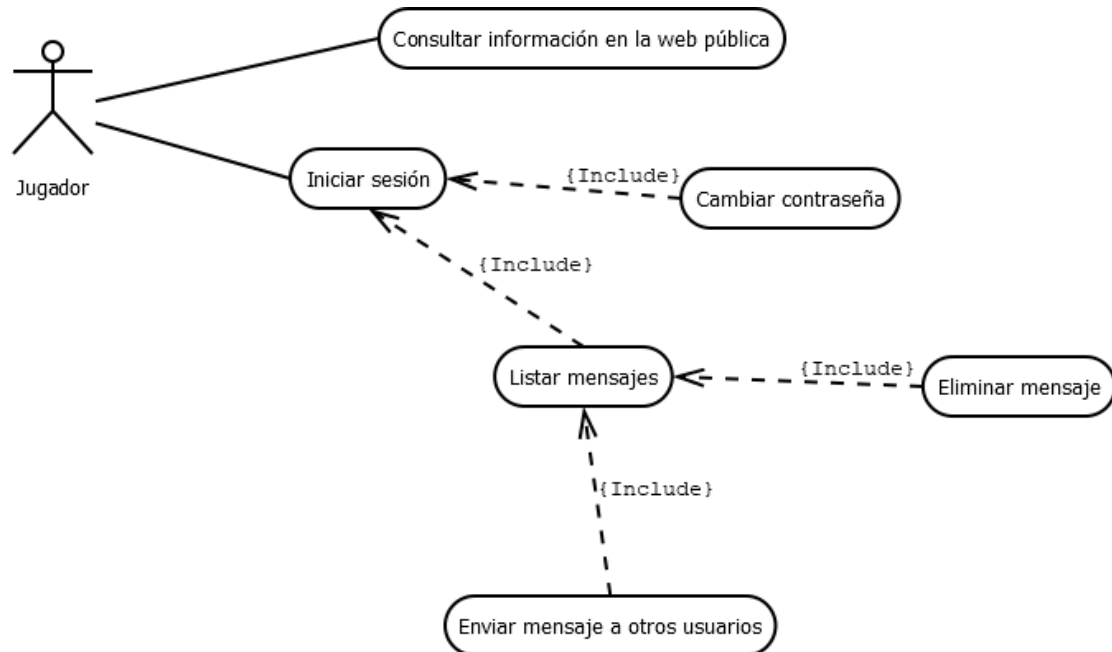


Figura 2 – Casos de uso Jugador

Descripción de los casos de uso

- **Consultar información en la web pública:** El usuario navegar por la página web pública, pudiendo ver datos como el calendario de partidos, resultados de partidos, listado de equipos y listado de jugadores.
 - **Precondiciones:** No aplica.
 - **Postcondiciones:** No aplica.
- **Iniciar sesión:** El usuario puede iniciar sesión en la aplicación con su nombre de usuario y contraseña.
 - **Precondiciones:** No aplica.
 - **Postcondiciones:** El usuario queda logueado en la aplicación.
- **Cambiar contraseña:** El usuario puede cambiar su contraseña de usuario.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - **Postcondiciones:** Se actualiza el campo contraseña para el usuario en la tabla Usuarios.
- **Listar mensajes:** El usuario puede listar sus mensajes recibidos.
 - **Precondiciones:**

- El usuario debe estar logueado.
 - Deben existir mensajes recibidos por el usuario.
 - **Postcondiciones:** No aplica.
- **Enviar mensaje a otros usuarios:** El usuario puede enviar mensaje a uno o varios usuarios de la aplicación.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir usuarios en la aplicación.
 - **Postcondiciones:** Se guarda un registro en la tabla Usuario-Mensajes por cada usuario seleccionado.
- **Eliminar mensaje:** El usuario puede eliminar cualquiera de sus mensajes recibidos.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir mensajes recibidos por el usuario.
 - **Postcondiciones:** Se elimina el registro de la tabla Usuario-Mensajes.

3.1.3. Actor Entrenador

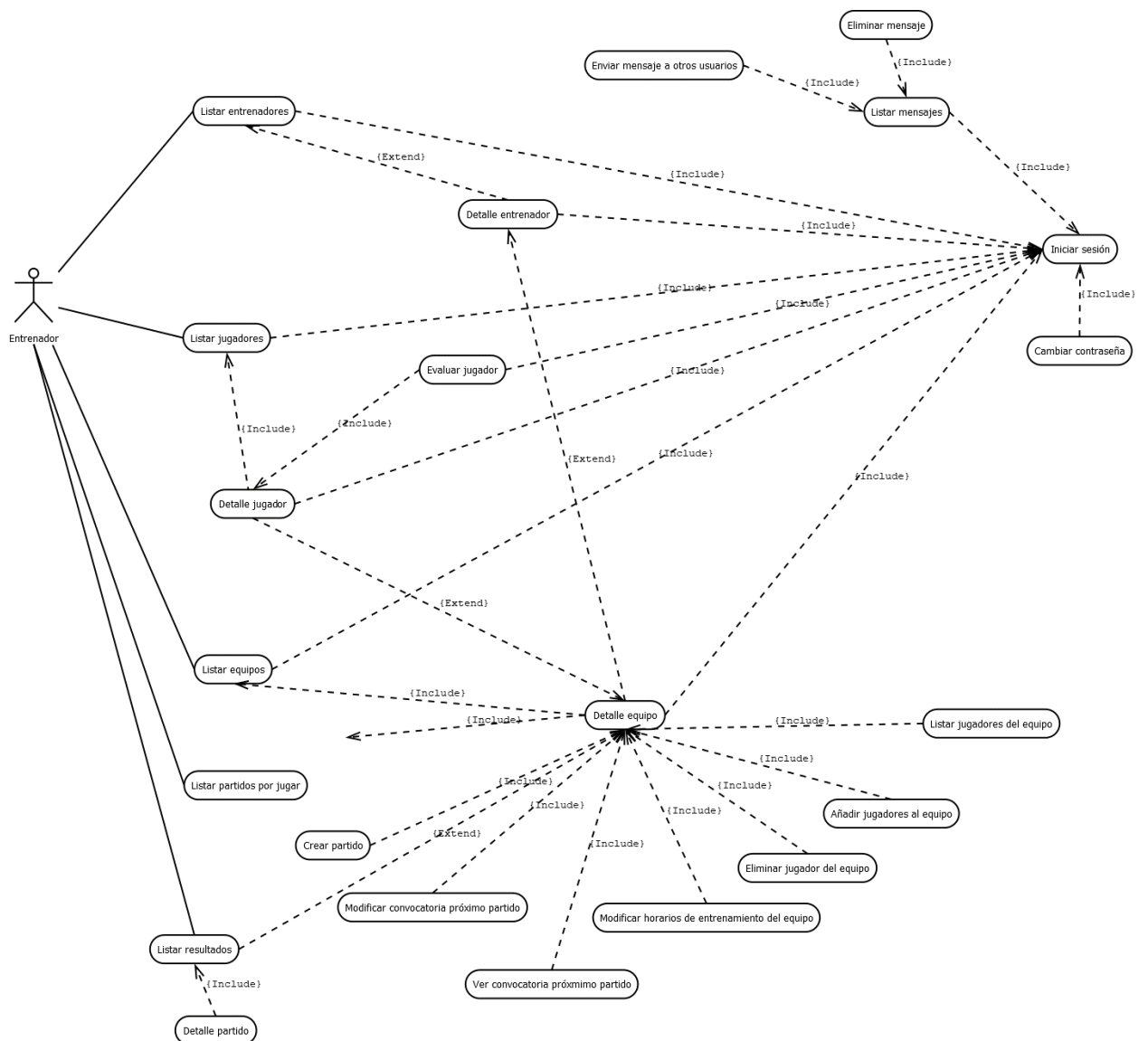


Figura 3 – Casos de uso Entrenador

Descripción de los casos de uso

- **Iniciar sesión:** El usuario puede iniciar sesión en la aplicación con su nombre de usuario y contraseña.
 - **Precondiciones:** No aplica.
 - **Postcondiciones:** El usuario queda logueado en la aplicación.
- **Cambiar contraseña:** El usuario puede cambiar su contraseña de usuario.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - **Postcondiciones:** Se actualiza el campo contraseña para el usuario en la tabla Usuarios.

- **Listar mensajes:** El usuario puede listar sus mensajes recibidos.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir mensajes recibidos por el usuario.
 - **Postcondiciones:** No aplica.
- **Enviar mensaje a otros usuarios:** El usuario puede enviar mensaje a uno o varios usuarios de la aplicación.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir usuarios en la aplicación.
 - **Postcondiciones:** Se guarda un registro en la tabla Usuario-Mensajes por cada usuario seleccionado.
- **Eliminar mensaje:** El usuario puede eliminar cualquiera de sus mensajes recibidos.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir mensajes recibidos por el usuario.
 - **Postcondiciones:** Se elimina el registro de la tabla Usuario-Mensajes.
- **Listar entrenadores:** El usuario puede obtener un listado de los entrenadores, con los siguientes datos:
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir entrenadores.
 - **Postcondiciones:** No aplica.
- **Detalle entrenador:** El usuario puede acceder a la página de detalle de un entrenador. En la página de detalle se muestran los datos personales del entrenador, un historial de los equipos que ha entrenado y un listado de los equipos que ha entrenado.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir entrenadores.
 - **Postcondiciones:** No aplica.
- **Listar jugadores:** El usuario puede obtener un listado de los jugadores.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir jugadores.

- **Postcondiciones:** No aplica.
- **Detalle jugador:** El usuario puede acceder a la página de detalle de un jugador. En la página de detalle se muestran los datos personales del entrenador, un historial de los equipos que ha entrenado y un listado de los equipos que ha entrenado.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir jugadores.
 - **Postcondiciones:** No aplica.
- **Evaluar jugador:** El usuario puede realizar una evaluación de todos los atributos del jugador.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir jugadores.
 - El jugador debe pertenecer a uno de los equipos que entrena el usuario.
 - **Postcondiciones:** Se crean nuevos registros en la tabla Jugadores-Atributos con fecha del momento de la evaluación.
- **Listar equipos:** El usuario puede obtener un listado de los equipos
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - **Postcondiciones:** No aplica.
- **Detalle equipo:** El usuario puede acceder a la página de detalle de un equipo. En la página de detalle se muestra un listado de los jugadores que pertenecen al equipo, los horarios de entrenamiento, la convocatoria para el próximo partido, los últimos resultados y los próximos partidos.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - **Postcondiciones:** No aplica.
- **Listar partidos:** El usuario puede obtener un listado de los próximos partidos del club.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - Deben existir partidos por jugar.

- **Postcondiciones:** No aplica.
- **Listar resultados:** El usuario puede obtener un listado de los últimos resultados del club.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - Deben existir partidos jugados.
 - **Postcondiciones:** No aplica.
- **Detalle partido:** El usuario puede acceder a la página de detalle de un partido en la que se muestra el resultado y las estadísticas de los jugadores que participaron en él.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener partidos jugados.
 - **Postcondiciones:** No aplica.
- **Crear partido:** El usuario puede crear un partido para el equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - Deben existir rivales.
 - El usuario debe ser entrenador del equipo.
 - **Postcondiciones:** Se crea un nuevo registro en la tabla partidos.
- **Ver convocatoria próximo partido:** El usuario puede ver un listado de los jugadores convocados para el próximo partido.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener partidos por jugar.
 - Debe haber registros en la tabla Jugadores-Partidos para el próximo partido.
 - **Postcondiciones:** No aplica.
- **Modificar convocatoria próximo partido:** El usuario puede realizar y modificar la convocatoria de jugadores para el próximo partido del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.

- El equipo debe tener partidos por jugar.
 - El usuario debe ser entrenador del equipo.
- **Postcondiciones:** Se crea un registro en la tabla Jugadores-Partidos por cada jugador seleccionado para el próximo partido. Si el jugador ya tuviera un registro para un partido el mismo día se elimina el registro anterior.
- **Modificar horarios de entrenamiento del equipo:** El usuario puede modificar los horarios de entrenamiento del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El usuario debe ser entrenador del equipo.
 - **Postcondiciones:** Se actualizan los registros en la tabla Equipo-Dias_de_entrenamiento.
- **Listar jugadores del equipo:** El usuario puede obtener un listado de los jugadores que pertenecen actualmente al equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener jugadores.
 - **Postcondiciones:** No aplica.
- **Eliminar jugador del equipo:** El usuario puede eliminar un jugador del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener jugadores.
 - El usuario debe ser entrenador del equipo.
 - **Postcondiciones:** Se actualiza el registro de la tabla Contratos para el contrato actual del jugador, asignándole fecha final al momento de la acción.
- **Añadir jugadores al equipo:** El usuario puede realizar y modificar la convocatoria de jugadores para el próximo partido del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.

- Deben existir jugadores del mismo género y la misma categoría del equipo que no pertenezcan a ningún equipo
- El usuario debe ser entrenador del equipo.
- **Postcondiciones:** Se crea un registro en la tabla Contratos para los jugadores seleccionados, con fecha de comienzo del momento de la acción.

3.1.4. Actor Administrador

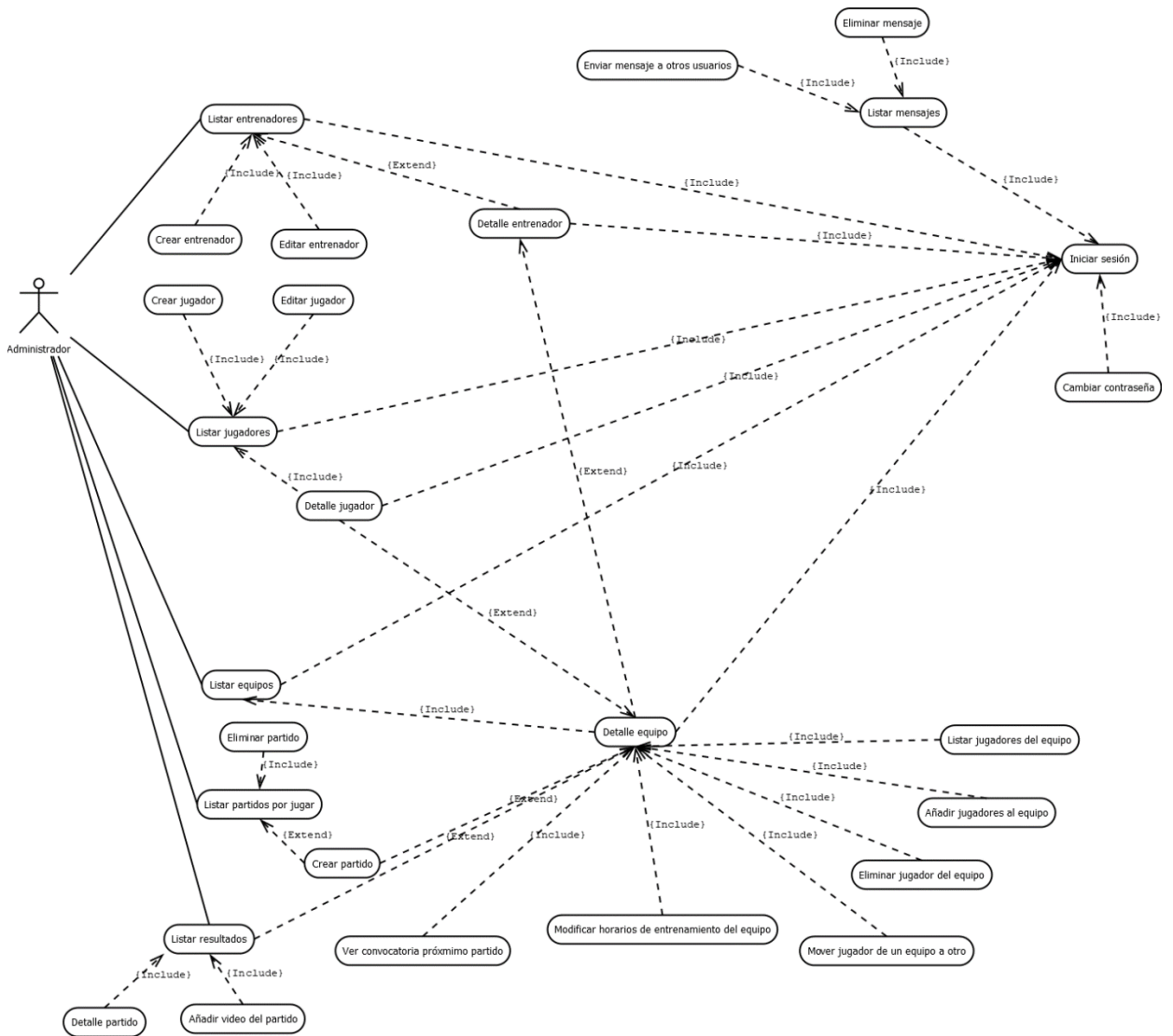


Figura 4 – Casos de uso administrador

Descripción de los casos de uso

- **Iniciar sesión:** El usuario puede iniciar sesión en la aplicación con su nombre de usuario y contraseña.
 - **Precondiciones:** No aplica.
 - **Postcondiciones:** El usuario queda logueado en la aplicación.
- **Cambiar contraseña:** El usuario puede cambiar su contraseña de usuario.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - **Postcondiciones:** Se actualiza el campo contraseña para el usuario en la tabla Usuarios.
- **Listar mensajes:** El usuario puede listar sus mensajes recibidos.

- **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir mensajes recibidos por el usuario.
 - **Postcondiciones:** No aplica.
- **Enviar mensaje a otros usuarios:** El usuario puede enviar mensaje a uno o varios usuarios de la aplicación.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir usuarios en la aplicación.
 - **Postcondiciones:** Se guarda un registro en la tabla Usuario-Mensajes por cada usuario seleccionado.
- **Eliminar mensaje:** El usuario puede eliminar cualquiera de sus mensajes recibidos.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir mensajes recibidos por el usuario.
 - **Postcondiciones:** Se elimina el registro de la tabla Usuario-Mensajes.
- **Listar entrenadores:** El usuario puede obtener un listado de los entrenadores, con los siguientes datos:
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir entrenadores.
 - **Postcondiciones:** No aplica.
- **Crear entrenador:** El usuario puede crear un entrenador.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - **Postcondiciones:**
 - Se crea un registro en la tabla Entrenadores.
 - Se crea un registro en la tabla Personas.
 - Se crea un registro en la tabla Usuarios.
 - Se envía un email con los datos de inicio de sesión al usuario.
- **Editar entrenador:** El usuario puede editar los campos de un entrenador.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir entrenadores.
 - **Postcondiciones:**

- Se modifican los valores del registro en la tabla Entrenadores.
- **Detalle entrenador:** El usuario puede acceder a la página de detalle de un entrenador. En la página de detalle se muestran los datos personales del entrenador, un historial de los equipos que ha entrenado y un listado de los equipos que ha entrenado.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir entrenadores.
 - **Postcondiciones:** No aplica.
- **Listar jugadores:** El usuario puede obtener un listado de los jugadores.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir jugadores.
 - **Postcondiciones:** No aplica.
- **Crear jugador:** El usuario puede crear un jugador.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - **Postcondiciones:**
 - Se crea un registro en la tabla Entrenadores.
 - Se crea un registro en la tabla Personas.
 - Se crea un registro en la tabla Usuarios.
 - Se envía un email con los datos de inicio de sesión al usuario.
- **Editar jugador:** El usuario puede editar los campos de un jugador.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir jugadores.
 - **Postcondiciones:**
 - Se modifican los valores del registro en la tabla Jugadores.
- **Detalle jugador:** El usuario puede acceder a la página de detalle de un jugador. En la página de detalle se muestran los datos personales del entrenador, un historial de los equipos que ha entrenado y un listado de los equipos que ha entrenado.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir jugadores.
 - **Postcondiciones:** No aplica.
- **Listar equipos:** El usuario puede obtener un listado de los equipos

- **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - **Postcondiciones:** No aplica.
- **Detalle equipo:** El usuario puede acceder a la página de detalle de un equipo. En la página de detalle se muestra un listado de los jugadores que pertenecen al equipo, los horarios de entrenamiento, la convocatoria para el próximo partido, los últimos resultados y los próximos partidos.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - **Postcondiciones:** No aplica.
- **Listar partidos:** El usuario puede obtener un listado de los próximos partidos del club.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - Deben existir partidos por jugar.
 - **Postcondiciones:** No aplica.
- **Listar resultados:** El usuario puede obtener un listado de los últimos resultados del club.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - Deben existir partidos jugados.
 - **Postcondiciones:** No aplica.
- **Detalle partido:** El usuario puede acceder a la página de detalle de un partido en la que se muestra el resultado y las estadísticas de los jugadores que participaron en él.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir partidos en estado jugado.
 - El partido ha de estar en estado jugado.
 - **Postcondiciones:** No aplica.
- **Añadir video del partido:** El usuario puede incluir un enlace de YouTube con el video del partido.
 - **Precondiciones:**

- El usuario debe estar logueado.
 - Deben existir partidos en estado jugado.
- **Postcondiciones:** No aplica.
- **Crear partido:** El usuario puede crear un partido.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - Deben existir rivales.
 - **Postcondiciones:** Se crea un nuevo registro en la tabla partidos con estado no jugado.
- **Ver convocatoria próximo partido:** El usuario puede ver un listado de los jugadores convocados para el próximo partido.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener partidos por jugar.
 - Debe haber registros en la tabla Jugadores-Partidos para el próximo partido.
 - **Postcondiciones:** No aplica.
- **Modificar convocatoria próximo partido:** El usuario puede realizar y modificar la convocatoria de jugadores para el próximo partido del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener partidos por jugar.
 - El usuario debe ser entrenador del equipo.
 - **Postcondiciones:** Se crea un registro en la tabla Jugadores-Partidos por cada jugador seleccionado para el próximo partido.
- **Modificar horarios de entrenamiento del equipo:** El usuario puede modificar los horarios de entrenamiento del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El usuario debe ser entrenador del equipo.
 - **Postcondiciones:** Se actualizan los registros en la tabla Equipo-Dias_de_entrenamiento.

- **Listar jugadores del equipo:** El usuario puede obtener un listado de los jugadores que pertenecen actualmente al equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener jugadores.
 - **Postcondiciones:** No aplica.
- **Eliminar jugador del equipo:** El usuario puede eliminar un jugador del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - El equipo debe tener jugadores.
 - El usuario debe ser entrenador del equipo.
 - **Postcondiciones:** Se actualiza el registro de la tabla Contratos para el contrato actual del jugador, asignándole fecha final al momento de la acción.
- **Añadir jugadores al equipo:** El usuario puede realizar y modificar la convocatoria de jugadores para el próximo partido del equipo.
 - **Precondiciones:**
 - El usuario debe estar logueado.
 - Deben existir equipos.
 - Deben existir jugadores del mismo género y la misma categoría del equipo que no pertenezcan a ningún equipo
 - El usuario debe ser entrenador del equipo.
 - **Postcondiciones:** Se crea un registro en la tabla Contratos para los jugadores seleccionados, con fecha de comienzo del momento de la acción.

4. Diseño

4.1. Base de datos

Entendemos como base de datos al conjunto de datos que pertenecen a un mismo concepto y sistemáticamente almacenado para posteriormente usarlo. En caso de una aplicación, se usan las bases de datos para organizar y operar con los datos guardados en el servidor, en nuestro caso hemos escogido una base de datos MySQL. A la hora de realizar una aplicación es muy importante realizar un buen diseño de la base de datos, para obtener una aplicación de calidad y, fácilmente escalable y mantenible en el tiempo.

4.1.1. Diagrama entidad-relación

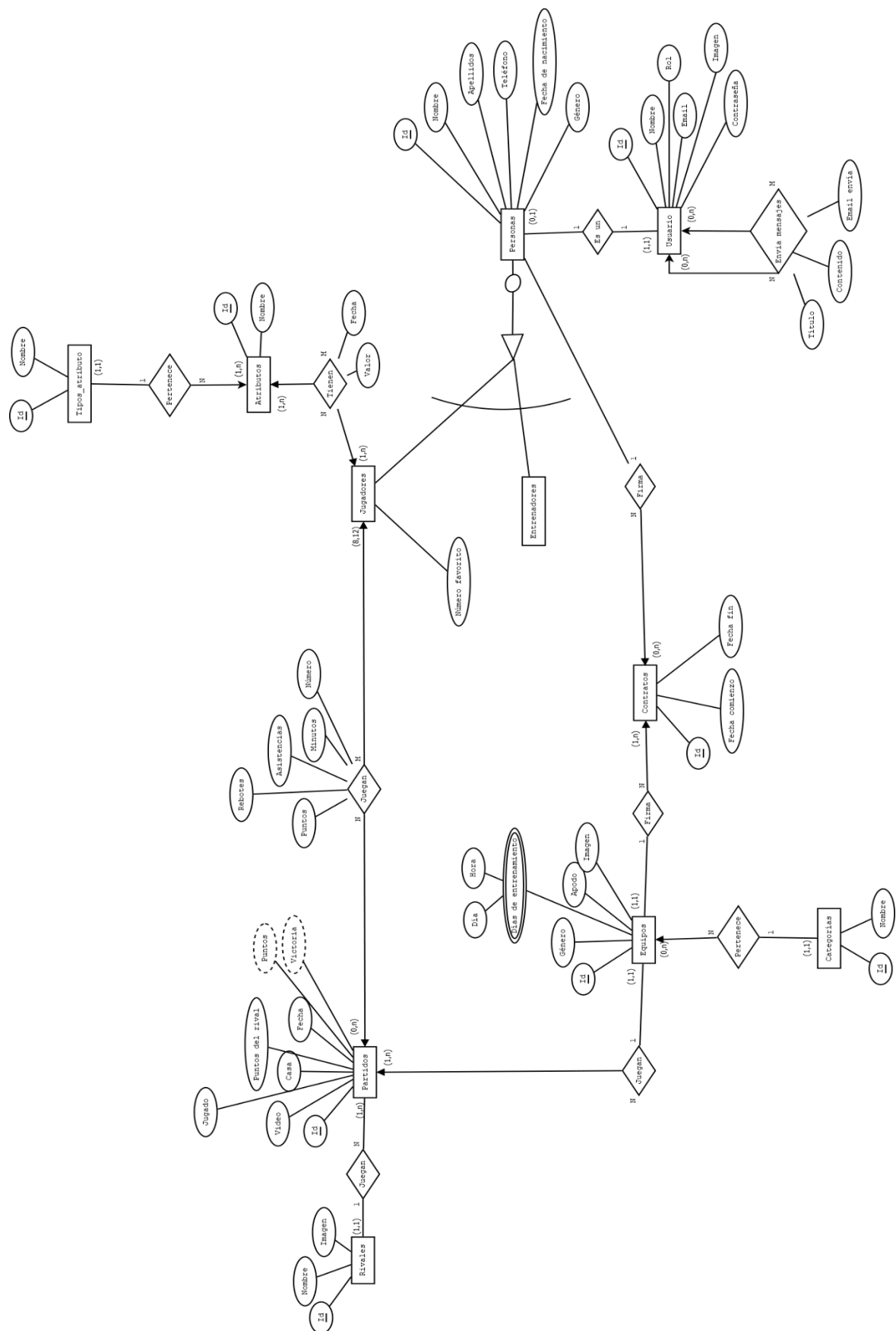


Figura 5 – Diagrama entidad relación

4.1.2. Diagrama relacional



Figura 6 – Diagrama relacional

4.2. Diagrama de clases

Un diagrama de clases de diseño recoge tanto los conceptos del dominio del sistema propios del diagrama de clases de análisis como aquellos conceptos que forman parte de la implementación de la propia aplicación del Sistema. A continuación, se muestra el diagrama de clases del proyecto.

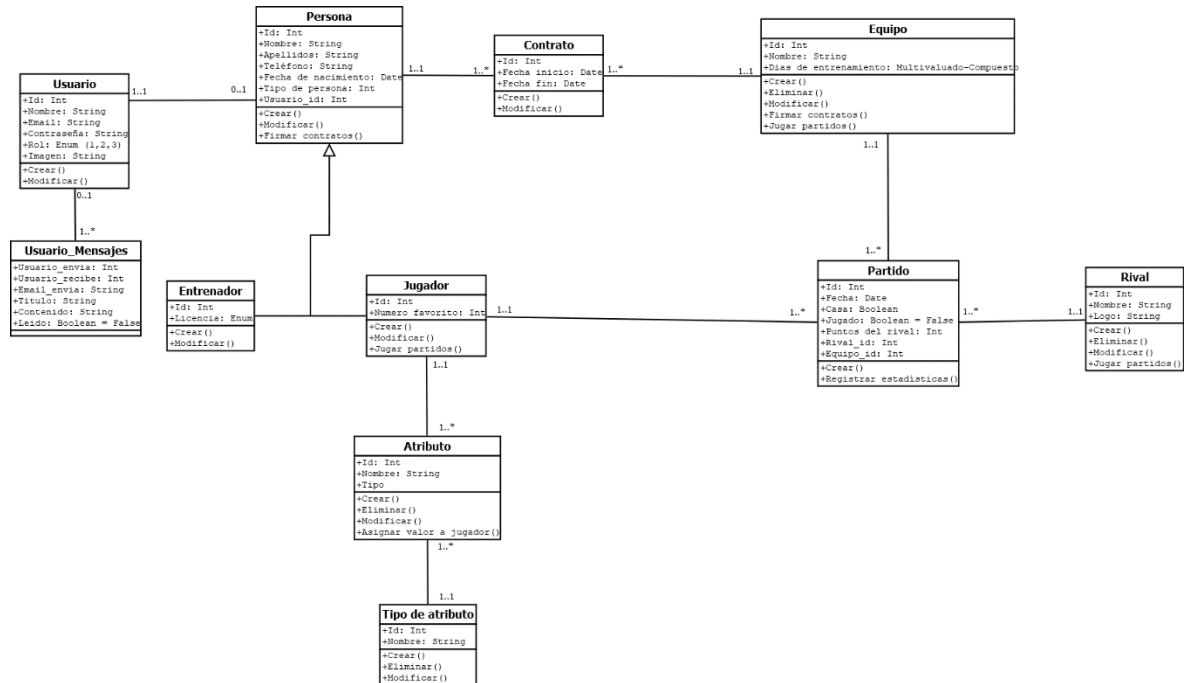


Figura 7 – Diagrama de clases

5. Tecnologías utilizadas

A continuación, se detallan las diferentes tecnologías que se han usado para la realización del proyecto.

5.1. Lenguajes

Para desarrollar una aplicación web debemos comunicarnos con el ordenador mediante una serie de instrucciones escritas en una sintaxis que la máquina entienda. A continuación, se detallan los lenguajes utilizados en el proyecto.

HTML5

Siglas correspondientes a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto. Es un lenguaje de marcado basado en etiquetas que el navegador interpreta para dar estructura a la página web.

CSS3

Siglas correspondientes a Cascading Style Sheets, es decir, Hojas de Estilo en Cascada. Es un lenguaje de diseño gráfico para dotar de estilo a un documento escrito en un lenguaje de marcado.

PHP

Es un lenguaje de programación diseñado especialmente para el desarrollo de aplicaciones web. El código PHP se ejecuta en el servidor y es este el que envía la página HML al cliente.

JavaScript

Es un lenguaje de programación diseñado especialmente para el desarrollo de aplicaciones web. Se ejecuta en el lado cliente, aunque, gracias a frameworks como NodeJs, puede ejecutarse también en el servidor.

5.2. Base de datos

Para el desarrollo de una aplicación web necesitamos, además, una base de datos que guarde la información de manera organizada para luego poder acceder a ella y utilizarla de manera fácil. A continuación, se detalla el sistema de base de datos utilizado en el proyecto.

MySQL

Es un sistema de gestión de base de datos relacionales de código abierto, siendo considerado como el sistema de gestión de base de datos más popular para el desarrollo web.

5.3. Frameworks y librerías

A la hora de desarrollar una aplicación web podemos apoyarnos en frameworks (entornos de trabajo) y librerías que nos permiten escribir código de una manera más rápida, fácil y mejor estructurada.

Laravel

Es un framework del lenguaje PHP que sigue el patrón MVC (Modelo, Vista, Controlador). Permite escribir código de una manera sencilla, rápida y con una buena organización de carpetas.

Bootstrap

Es un framework del lenguaje CSS que favorece el desarrollo de webs de una manera más cómoda y rápida. Se ha usado, sobre todo, para conseguir la estructura básica de la web y hacer esta misma responsive.

JQuery

Es una biblioteca del lenguaje JavaScript que permite simplificar la manera de interactuar con los elementos de un documento HTML. Además, simplifica el desarrollo de animaciones, el manejo de eventos y las llamadas al servidor mediante AJAX.

Composer

Es un gestor de dependencias para PHP, similar a lo que npm es para JavaScript. Es una aplicación PHP que ayuda a administrar las librerías desarrolladas por terceros que vas a incorporar a tu proyecto.

5.4. Control de versiones

Cuando desarrollamos una aplicación web, se considera buena práctica el contar con un sistema de control de versiones que registre los cambios realizados en los archivos de la aplicación a lo largo del tiempo, de modo que nos permite recuperar

versiones específicas más adelante. A continuación, se detallan las herramientas utilizadas para el control de versiones.

GIT

Es un software de control de versiones, diseñado por Linus Trovalds y de software libre. Es el más famoso y utilizado en el desarrollo de aplicaciones.

GitHub

Es una plataforma de desarrollo colaborativo, que nos permite alojar el proyecto utilizando el sistema de control de versiones Git. Además, funciona como una red social pensada para desarrolladores. Aquí está el enlace del repositorio del proyecto <https://github.com/javicordero/proyecto>

5.5. Software

Para desarrollar una aplicación necesitamos utilizar una serie de programas que nos permitan ejecutar diferentes tareas en nuestro ordenador. A continuación, se detallan los programas utilizados en la realización del proyecto.

Visual Studio Code

Es un editor de código desarrollado por Microsoft y de código abierto. Es el editor más utilizado en la actualidad, permite trabajar con múltiples lenguajes, es muy fácil de personalizar y consta de miles de plugins gratuitos fácilmente instalables.

XAMPP

Es un paquete que contiene el servidor web Apache, un gestor de base de datos MySQL y los intérpretes para los lenguajes de programación PHP y Perl. En el proyecto se ha usado para alojar la aplicación en un servidor local desde el que trabajar.

phpMyAdmin

Es una herramienta que permite manejar la administración de bases de datos MySQL a través de un navegador.

Dia

Es un programa diseñado para el dibujo de diagramas y esquemas. Permite realizar diferentes tipos de esquemas con una interfaz muy fácil de utilizar.

Sourcetree

Es un programa que nos permite trabajar con Git de una manera muy cómoda y sencilla. Permite clonar el repositorio de GitHub y trabajar con el desde una interfaz bastante clara.

6. Desarrollo

Para desarrollar la aplicación se ha usado el paradigma de programación MVC (Modelo, vista, controlador), apoyado en el framework de PHP Laravel. El uso de Laravel ha permitido que el desarrollo de la aplicación haya sido ágil, facilitando la creación de los modelos, vistas y controladores necesarios; además de aportar diferentes herramientas como Seeders (semillas) y Factories (factorías). Se ha optado por el uso de Laravel por lo fácil que es crear un proyecto desde cero, únicamente se necesitan una serie de líneas de comandos.

```
composer global require laravel/installer

laravel new example-app

cd example-app

php artisan serve
```

Figura 8 – Comandos para crear un proyecto con Laravel y lanzar el servidor

A continuación, se detalla cómo ha sido el desarrollo del proyecto.

6.1. Estructura

Una vez creado el proyecto Laravel obtenemos la siguiente estructura de carpetas:

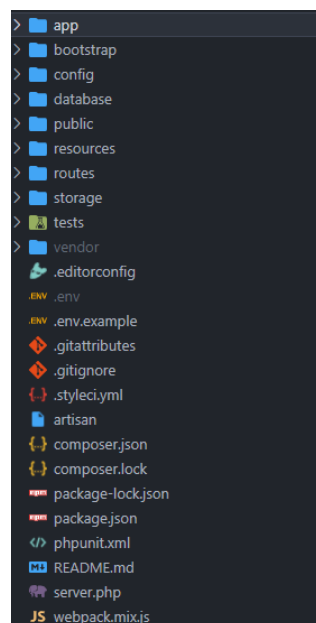


Figura 9 – Estructura de carpetas Laravel

A continuación, se detallan los ficheros y directorios más importantes de la aplicación.

6.1.1. Directorio App

En este directorio se pueden encontrar diferentes directorios entre las que destacan los siguientes.

Directorio Models

Incluye todos los modelos de la aplicación. (Incluir foto de la carpeta)

Los modelos, además de mapear la base de datos permiten relacionarse entre ellos y obtener los datos de las tablas sin necesidad de escribir código SQL. En las imágenes siguientes se muestran cómo se define una relación en el modelo y cómo se utiliza en el controlador. He querido destacar la relación entre jugadores y personas al ser una relación polimórfica.

```
public function person(){
    return $this->morphOne(People::class, 'personable');
}
```

Figura 10 – Definición de la relación con personas en el modelo jugadores

Con el siguiente código obtenemos dos instancias de dos clases diferentes: la variable `$player` contiene una instancia de la clase Jugador y la variable `$person` contiene una instancia de la clase Persona.

```
public function show($id){
    $player = Player::find($id);
    $person = $player->person;

    $attribute_types = AttributeType::all();
    $data = compact('attribute_types', 'person', 'player');

    return view('admin.people.profile', compact('data'));
}
```

Figura 11 –Método show (mostrar) del controlador de jugadores

Además, los modelos nos aportan la posibilidad de crear accessors. Un accessor es un método que permite generar atributos extras en el modelo. Son muy útiles para los atributos derivados y para formatear fechas y horas. El siguiente accessor devuelve la ruta relativa de la imagen preparada para todos los usuarios y para que funcione correctamente en el atributo `'src'` de una etiqueta HTML ``.

```
//Atributo image_path
public function getImagePathAttribute(){
    if($this->image){
        if(substr($this->image, 0, 4) != 'http'){
            return '/images/users/' . $this->image;
        }
        else{
            return $this->image;
        }
    }
    else{
        return '/images/users/default.jpg';
    }
}
```

Figura 12 – Accesor `getImagePathAttribute()` del modelo `Usuarios`

Directorio Traits

Contiene los traits (rasgos) de la aplicación. Un trait es un mecanismo de reutilización de código en lenguajes de herencia simple, como PHP. Es similar a una clase, pero con el objetivo de agrupar funcionalidades específicas de manera coherente. El trait a continuación se usa para formatear una fecha pasando de “21/06/2021” a “Lunes 21 de junio de 2021”.

```
trait FormatearDate
{
    public static function formatDate($date){
        return __(strftime("%A", $date->getTimestamp())).
            ' '.
            strftime("%d", $date->getTimestamp()).
            ' de '.
            __(strftime("%B", $date->getTimestamp())).
            ' de '.
            strftime("%Y", $date->getTimestamp());
    }
}
```

Figura 13 – Trait `FormatearDate`

Directorio Http/Middleware

Contiene los middlewares (lógica de intercambio de información entre aplicaciones) de la aplicación. Un middleware es un archivo que filtra las peticiones HTTP en un sistema, es un archivo adicional que va en el medio de la petición y del resultado final en la vista. En el caso de la aplicación hemos creado el `AdminMiddleware` que gestiona el acceso a las rutas de la aplicación de los diferentes roles de usuario. (Incluir foto del archivo)

Directorio Http/Controllers

Contienes los controladores de la aplicación. En el proyecto se han agrupado los controladores en tres carpetas diferentes:

- Directorio raíz: Contiene los controladores necesarios para la aplicación administrativa.
- Publics: Contiene los controladores necesarios para la página web pública de la aplicación.
- Auth: Contiene los controladores del paquete de autenticación de Laravel. En el proyecto solo se ha usado el LoginController para gestionar el inicio de sesión de los usuarios, aunque no se han eliminado el resto de controladores por si en un futuro se desea implementar alguna otra funcionalidad del paquete.

(Incluir foto de la carpeta controllers)

6.1.2. Directorio database

En este directorio se pueden encontrar diferentes directorios entre las que destacan los siguientes.

Directorio migrations

Contiene las migraciones de la aplicación. Las migraciones son las encargadas de crear las tablas y campos de la base de datos. En la imagen se muestra la migración

encargada de crear la tabla personas, he querido destacarla porque es la tabla padre de una jerarquía y para implementarla hay que usar una relación polimórfica.

```
class CreatePeopleTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('people', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('surname');
            $table->string('phone');
            $table->date('birth_date')->nullable();
            $table->enum('gender', ['masculino', 'femenino']);
            $table->string('personable_type');
            $table->unsignedBigInteger('personable_id')->constrained()->cascadeOnDelete();
            $table->foreignId('user_id')->constrained()->cascadeOnDelete();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('people');
    }
}
```

Figura 14 – Migración tabla personas

Directorio seeders

Contiene los seeders (semillas) de la aplicación. Los seeders son componentes que nos permiten rellenar las tablas de datos. Al comienzo del desarrollo se utilizaron para llenar las datas con datos ficticios y poder hacer pruebas. Al final se han seguido usando algunos de ellos, principalmente el UserSeeder, encargado de crear usuarios y de enlazarlos con las tablas personas, jugadores y entrenadores, rellenando todas las tablas con datos ficticios, apoyándose en la librería faker.

6.1.3. Directorio public

Contiene todos los assets (activos digitales) necesarios para usar la aplicación. Un asset es una representación de cualquier item utilizado en el proyecto. El directorio incluye los ficheros JavaScript, CSS y las imágenes. Está dividido en dos directorios:

- Directorio raíz: Contiene los assets necesarios para la aplicación de administrativa.
- Public-template: Contiene los assets necesarios para la página web pública de la aplicación.

6.1.4. Directorio resources

Contiene varios directorios entre los que destaca el directorio views.

Directorio views

El directorio views (vistas) contiene todas las vistas de la aplicación. Las vistas están escritas en sintaxis Blade, el sistema de plantillas predeterminado de Laravel. Blade nos permite integrar código PHP en el HTML de una manera sencilla y clara. La imagen siguiente nos muestra una tabla recorriendo la variable `$data['players']`, incluyendo varios condicionales.

```
<table class="table table-striped table-bordered" id="team-players-table">
  <thead>
    <tr>
      <th class="text-center">#</th>
      <th class="text-center">Nombre</th>
      <th class="text-center">Acciones</th>
    </tr>
  </thead>
  <tbody>
    @foreach ($data['players'] as $person)
      <tr>
        <td>{{ $person->personable->number }}</td>
        <td>{{ $person->full_name }}</td>
        <td>
          <form class="form-inline" id="del_event{{ $person->id }}"
            action="{{ route('admin.players.removeFromTeam', $person->id) }}" method="POST">
            @if ($data['team']->current_coach && $data['team']->current_coach->user == Auth::user() || Auth::user()->role == 1)
              <div class="form-group">
                @csrf
                <button class="del_id table-btn table-btn-danger" type="submit"
                  data-attrId="{{ $person->id }}" data-toggle="tooltip" data-placement="top" title="Eliminar del equipo"><i class="fa fa-trash"></i></button>
              </div>
            @endif
            <div class="form-group">
              <a href="{{ route('admin.players.show', $person->personable_id) }}"
                class="pull-right table-btn table-btn-success">
                <i class="fa fa-eye"></i>
              </a>
            </div>
            @if ($data['team']->current_coach->user == Auth::user() || Auth::user()->role == 1)
              <div class="form-group">
                <button
                  class="pull-right modal_id table-btn move-player-modal table-btn-warning"
                  data-personId="{{ $person->id }}" data-csrf="{{ csrf_token() }}"
                  type="button" data-toggle="tooltip" data-placement="top" title="Mover a otro equipo">
                  <i class="fa fa-send"></i>
                </button>
              </div>
            @endif
          </form>
        </td>
      </tr>
    @endforeach
  </tbody>
</table>
```

Figura 15 – Parte de la vista `admin.teams.players.blade.php`

Está dividido en dos directorios:

- Admin: Contiene las vistas de la aplicación administrativa.
- Directorio raíz: Contiene las vistas de la página web pública de la aplicación.

6.1.5. Directorio routes

Incluye diferentes ficheros entre los que destaca el fichero web.php. Este fichero es el encargado de gestionar las rutas de la aplicación de una manera intuitiva y fácil de utilizar.

```
//TEAMS
Route::put('teams/{id}/update', [TeamController::class, 'update'])->name('teams.update');
Route::delete('teams/{id}', [TeamController::class, 'destroy'])->name('teams.destroy');
Route::post('teams/store', [TeamController::class, 'store'])->name('teams.store');
Route::get('teams/{id}', [TeamController::class, 'show'])->name('teams.show');
Route::get('teams', [TeamController::class, 'index'])->name('teams.index');
Route::post('teams/getViewForPracticesModal', [TeamController::class, 'practicesView'])->name('teams.practicesView');
Route::post('teams/create', [TeamController::class, 'create'])->name('teams.create');
Route::post('teams/edit', [TeamController::class, 'edit'])->name('teams.edit');
Route::post('teams/getPossibleTeamsForPlayer', [TeamController::class, 'getPossibleTeamsForPlayer'])->name('teams.getPossibleTeamsForPlayer');
Route::post('teams/getAllListablePlayers', [TeamController::class, 'getAllListablePlayers'])->name('teams.getAllListablePlayers');
Route::post('teams/savePlayersForNextGame', [TeamController::class, 'savePlayersForNextGame'])->name('teams.savePlayersForNextGame');
Route::post('teams/{id}/savePracticesDays', [TeamController::class, 'savePracticesDays'])->name('teams.savePracticesDays');
Route::post('teams/freePlayersForTeam', [TeamController::class, 'freePlayersForTeam'])->name('teams.freePlayersForTeam');
Route::post('teams/{id}/savePlayersOnTeam', [TeamController::class, 'savePlayersOnTeam'])->name('teams.savePlayersOnTeam');
```

Figura 16 – Rutas del controlador de equipos

6.2. Diseño de las interfaces

Para el diseño de las diferentes interfaces se ha dividido el proyecto en las dos aplicaciones que aclaramos anteriormente. Por ello, se han escogido dos plantillas claramente diferenciadas, una para la página web pública y otra para la aplicación administrativa.

6.2.1. Aplicación administrativa

Para la aplicación administrativa se ha escogido la plantilla Gentelella Alela, <https://colorlib.com/polygon/gentelella/index.html> (enlace a la demo). Es una plantilla administrativa realizada con Bootstrap que se ajusta perfectamente a las necesidades del proyecto. Contiene una estructura básica bastante estética, es responsive y cuenta con una serie de plugins fácilmente incorporables a la aplicación. A continuación, se muestra la vista de detalle de un equipo para mostrar el acabado final de la plantilla con las modificaciones que se han llevado a cabo.

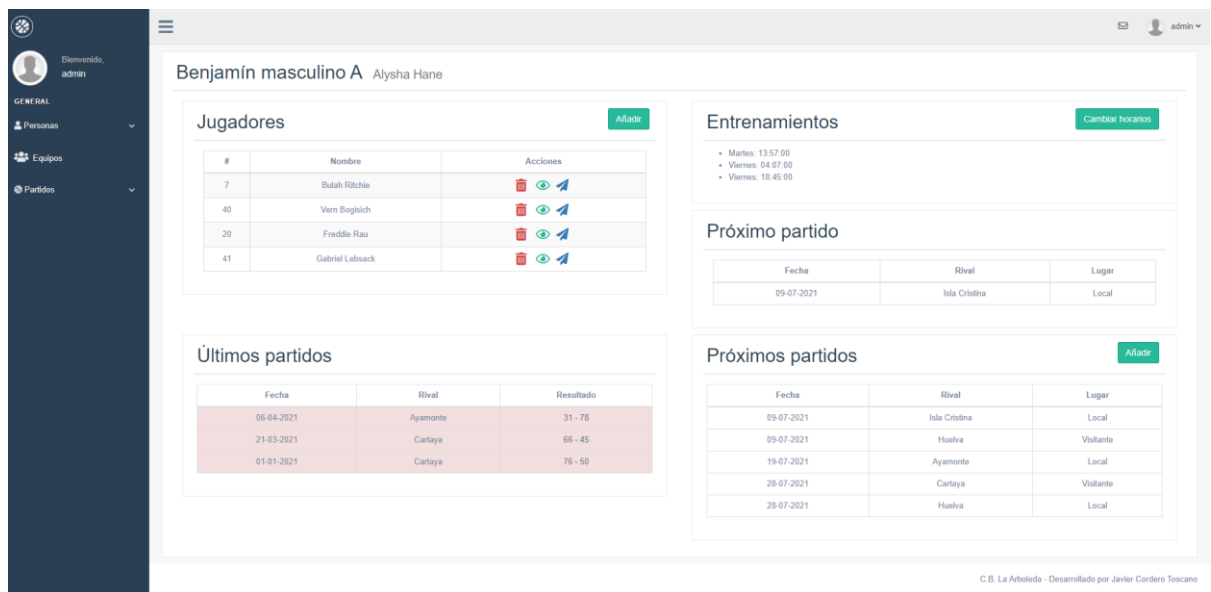


Figura 17 – Vista detalle de Equipo

Como se aprecia en las imágenes se ha optado por colocar los diferentes componentes de una vista en cajas, con el título y las acciones disponibles para cada componente.

A continuación, se detallan los plugins y herramientas de terceros más destacables que se han usado en el proyecto.

DataTables

Es un plugin de jQuery que construye tablas dinámicas de una manera muy simple, pero potente. Permite ordenar, filtrar y paginar los datos introducidos. Además de hacer responsive la tabla, todo esto con unas pocas líneas de código. A continuación, se muestra como se ha usado en la aplicación.

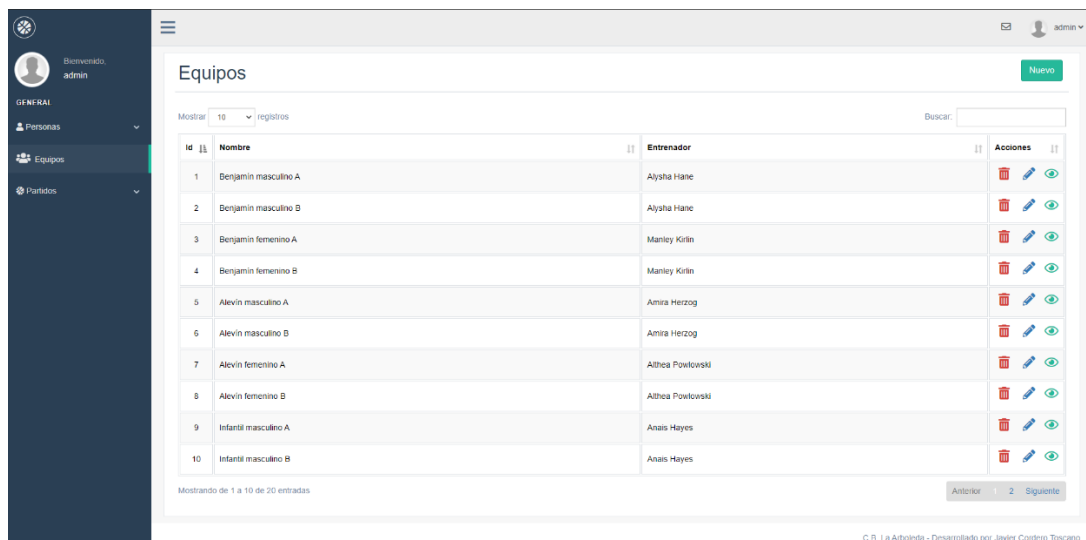


Figura 18 – Vista listado de Equipos

Charts.js

Es una librería de JavaScript diseñada para crear diferentes tipos de gráficas. Chart.js se representa en HTML5 Canvas y está ampliamente considerado como una de las mejores bibliotecas de visualización de datos. Se ha usado para realizar el gráfico de evolución del jugador. Para pasarle los datos a la función JavaScript que pinta la gráfica se realiza una petición Ajax al servidor.

Evolución del jugador

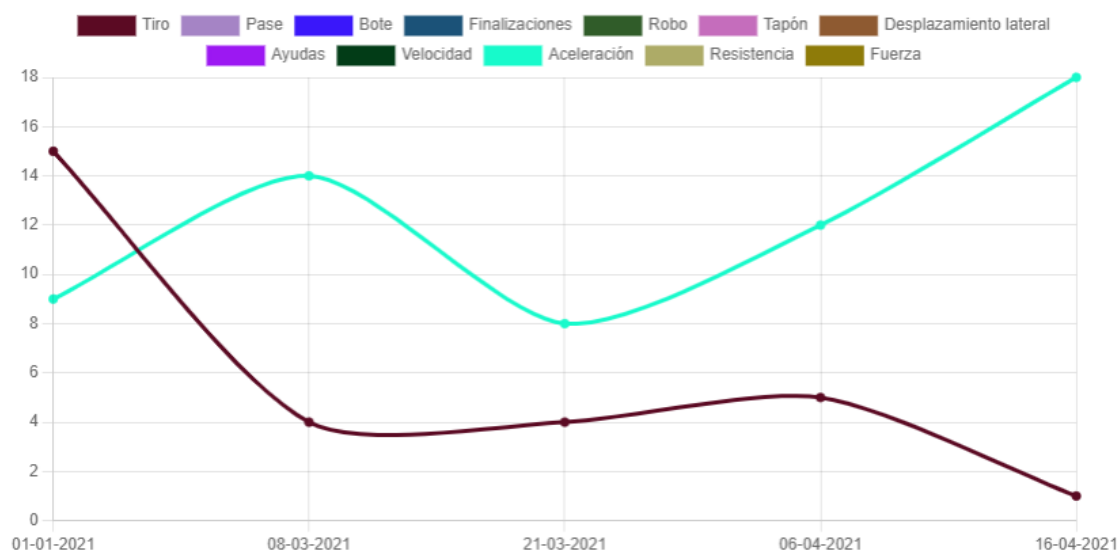


Figura 19 – Gráfico evolución del Jugador

SweetAlert2

Es una herramienta que permite crear diferentes alertas de una manera sencilla y estética. Se ha usado para informar al usuario de los cambios que se producen al realizar ciertas acciones dentro de la aplicación.

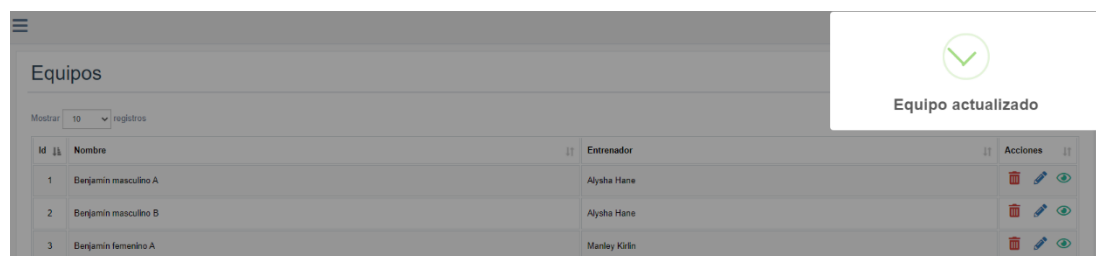


Figura 20 – Muestra de uso de SweetAlert2

6.2.2. Página web pública

Para la página web pública se ha escogido la plantilla Phoenix, <https://preview.colorlib.com/#phoenix> (enlace a la demo). Es una plantilla realizada con Bootstrap, diseñada para un equipo de baseball. Presenta una gama de colores bastante simple, basada en tres colores principales y cuenta con componentes prediseñados, como calendarios y resultados, que han sido muy útiles en el proyecto. A continuación, se muestran como han quedado algunos componentes tras adaptar la plantilla a las necesidades del proyecto.



Figura 21 – Componente resultado de partido



Figura 22 – Carrusel de componentes de partidos por jugar

También se destaca una card modal que muestra las estadísticas y los atributos de un jugador, respetando la gama de colores de la página.

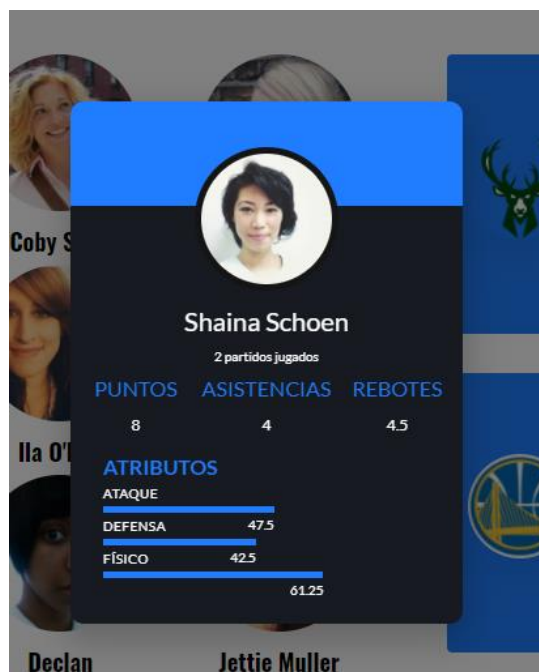


Figura 23 – Card modal estadísticas de jugador

7. Conclusiones

Durante el desarrollo del proyecto, me he dado cuenta de la importancia de la organización para poder construir una aplicación de calidad. Ha sido clave, en primer momento, un buen diseño de la base de datos para poder empezar a construir sobre una buena base.

En un principio se pensó en crear una aplicación de gestión para clubes deportivos sin enfocarse a ningún deporte en especial, pero finalmente se decidió centrarse en el baloncesto y ha sido un gran acierto. La posibilidad de abarcar todos los deportes era muy interesante, pero se podían perder funcionalidades específicas interesantes que, finalmente, se han podido obtener gracias a centrarnos en un único deporte. Durante el proceso he notado, todavía más de lo que creía, la importancia de usar frameworks, librerías y herramientas de terceros para agilizar el desarrollo de la aplicación. Cuanto mayor es el tamaño de la aplicación a desarrollar más necesario se hace el uso de estas herramientas.

Para terminar, ha sido interesante enfrentarme a la construcción de una aplicación desde cero, además relacionada con un tema que me interesa. Estoy seguro de que seguiré mejorando la aplicación para poder darle uso fuera de lo académico.

7.1. Aprendizaje

La realización del proyecto me ha servido, sobre todo, para ampliar mis conocimientos en el framework Laravel, apoyándome en la gran comunidad que existe en internet alrededor del framework y en la documentación del mismo. Además, el enfrentarme, por primera vez, a un proyecto desde cero y en solitario han conseguido mejorar mis capacidades, tanto de análisis, como de resolución de problemas dentro del desarrollo.

8. Trabajo futuro

Una vez finalizado el proyecto se ha obtenido una aplicación, no muy grande, pero bastante sencilla de escalar y de mantener en el tiempo. Se observan varias líneas de mejoras de cara al futuro, entre las que se encuentran las siguientes:

- Diseñar un sistema de control de pagos de cuotas y realización de facturas.
- Diseñar un sistema de control de asistencia a entrenamientos.
- Integrar mejor la web pública y la aplicación administrativa.
- Mejorar la modularización de las vistas.
- Desarrollar un sistema para validar los datos introducidos por los usuarios.

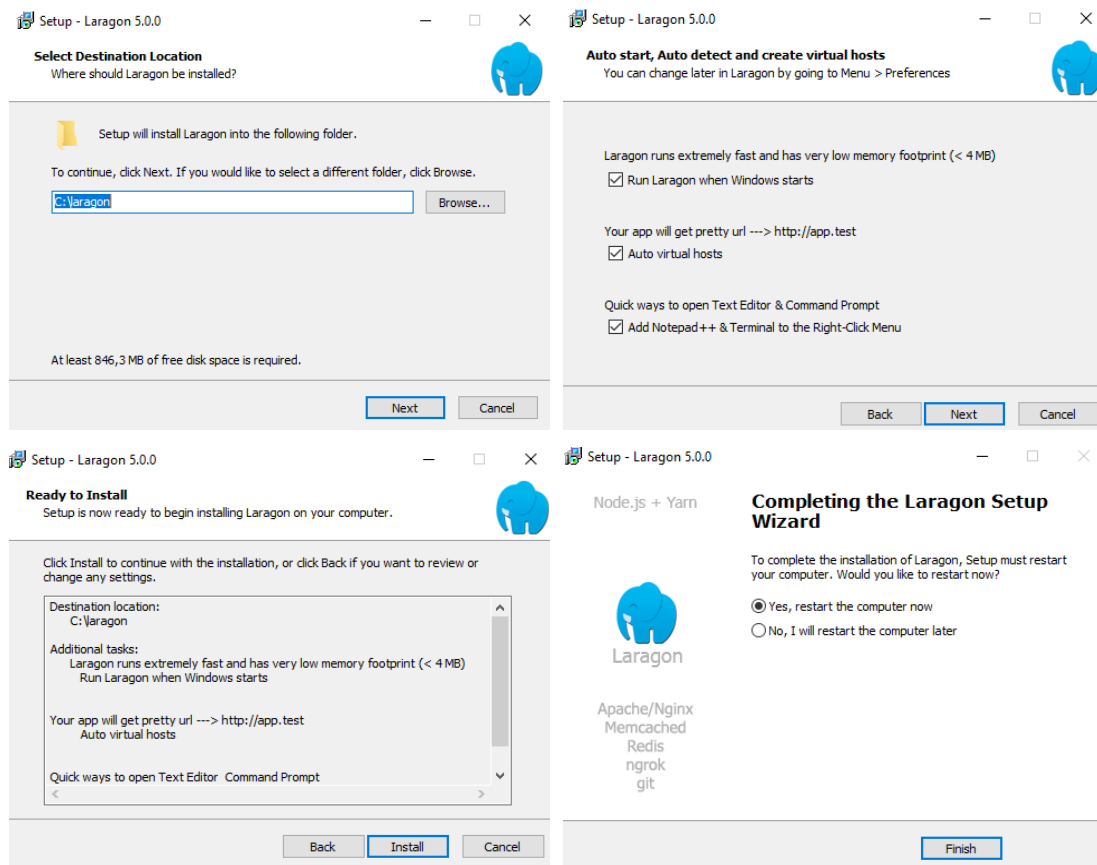
9. Bibliografía

- PHP - Manual: <https://www.php.net/manual/es/>
- Laravel – Documentación: <https://laravel.com/docs/8.x>
- Bootstrap – Documentación:
<https://getbootstrap.com/docs/5.0/getting-started/introduction/>
- GitHub - Documentación: <https://docs.github.com/es>
- Datatables - Manual: <https://datatables.net/manual/>
- Charts.js – Documentación: <https://www.chartjs.org/docs/latest/>
- SweetAlert2 - Documentación: <https://sweetalert2.github.io/>
- Select2 – Documentación: <https://select2.org/>

Anexo I: Manual de instalación

1. Instalación de Laragon

Descargar el programa desde su web <https://laragon.org/download/index.html>.
Abrir el ejecutable, dejar todas las opciones que vienen por defecto e ir avanzando dándole al botón Next.



Figuras 24, 25, 26 y 27 – Ventanas de instalación de Laragon

2. Instalación de la base de datos

Abrir Laragon, pulsar sobre el botón Start All y aceptar todos los permisos que se piden.

Pulsar sobre el botón Database para abrir el gestor de bases de datos que trae Laragon incorporado.



Figura 28 - Menú principal Laragon

Pulsar sobre el botón Nueva de la esquina inferior izquierda, dejar las opciones por defecto y pulsar sobre Abrir.

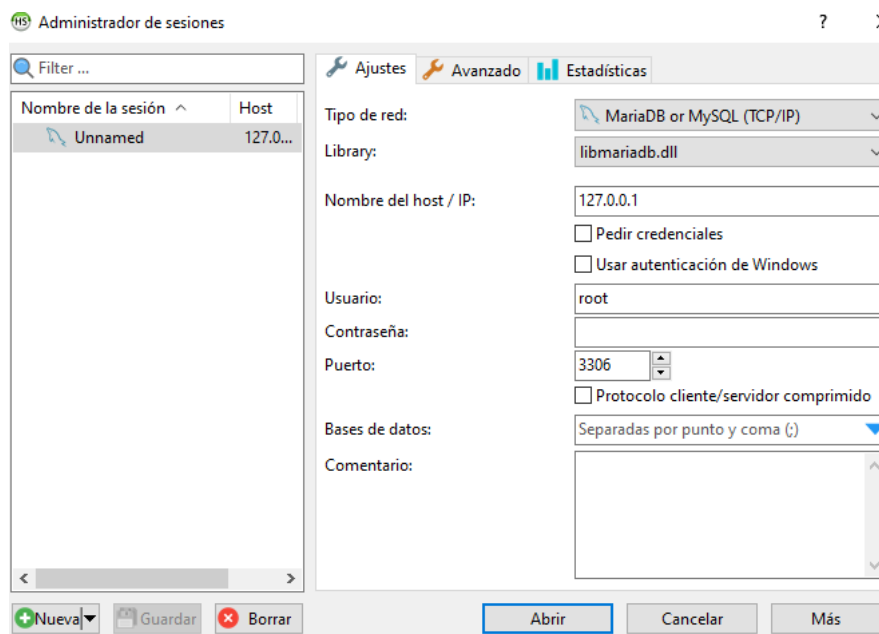


Figura 29 – Ventana inicial del Gestor de base de datos de Laragon

Una vez se ha abierto el gestor de base de datos, hacer click con el botón derecho del ratón en el menú lateral derecho y seleccionar Crear nuevo -> Base de datos.

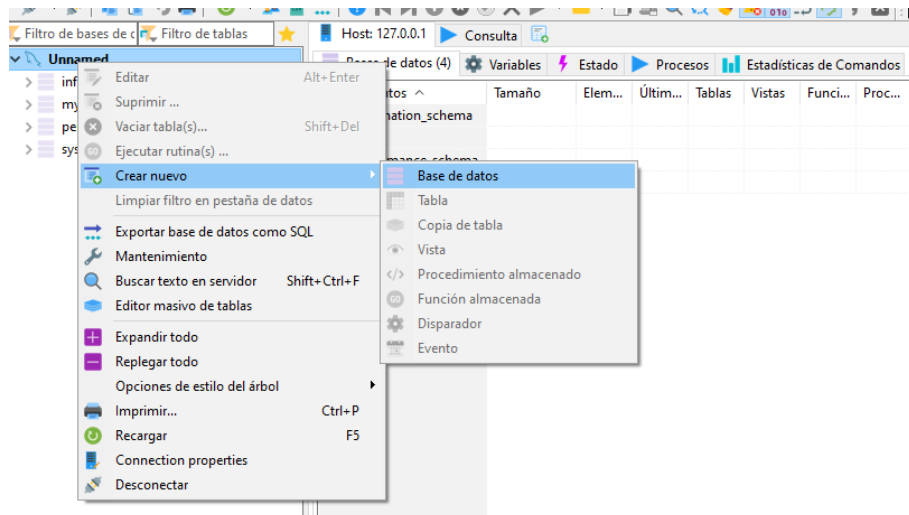


Figura 30 – Ventana principal del Gestor de base de datos

Poner como nombre “proyecto” y en Collation seleccionar “utf8_unicode_ci”

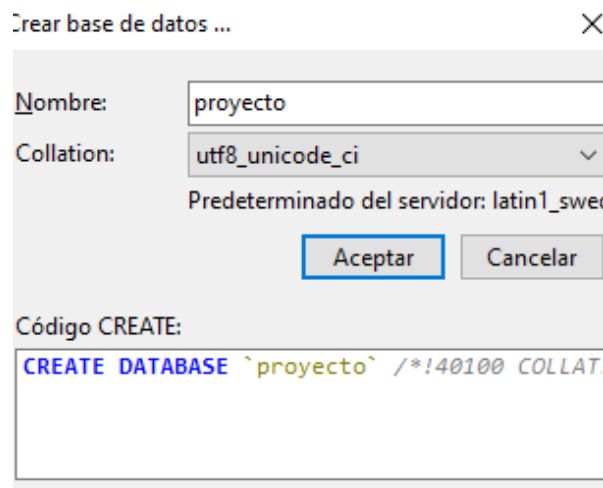


Figura 31 -Ventana crear base de datos

Seleccionar la base de datos creada haciendo doble click sobre ella. En la esquina superior izquierda desplegar las opciones de Archivo, seleccionar Ejecutar archivo SQL y escoger el archivo SQL que viene dentro de la carpeta Base de Datos en el usb.

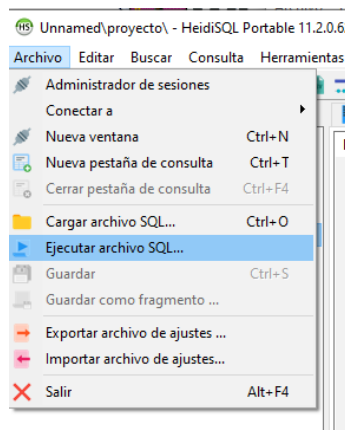


Figura 32 – Desplegable con Ejecutar Archivo SQL seleccionado

3. Ejecutar la aplicación

Copiar la carpeta APP desde el usb al directorio laragon/www.

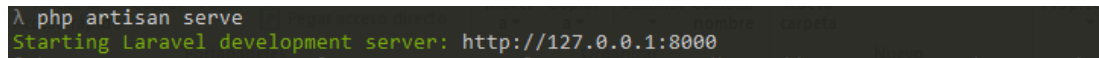
Volver al menú principal de Laragon y pulsar el botón Terminal.

En la terminal que se abre introducir el siguiente código para lanzar el servidor:

`php artisan serve`

Manetener el control izquierdo del teclado pulsado y hacer click sobre la ruta

`http://127.0.0.1:8000`



```
λ php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

Figura 33 – Consola con comando `php artisan serve`

Anexo II: Manual de usuario

El manual de usuario está dividido en dos partes, una para la página web pública y otra para la aplicación de gestión administrativa.

1. Página web pública

1.1. Inicio

En la ventana de inicio se muestra un resumen del club. Se puede interactuar con el botón Detalles en cualquiera de los resultados de partidos que aparecen en la. Al pulsar sobre él se redirige a la pagina de detalles del partido.



Figura 34 – Módulo Último partido en ventana Inicio

1.2. Equipos

En esta ventana se muestra un listado de los equipos paginados. El usuario tiene varios elementos de interacción. Puede interactuar con los botones inferiores para moverse entre las diferentes páginas de la vista.

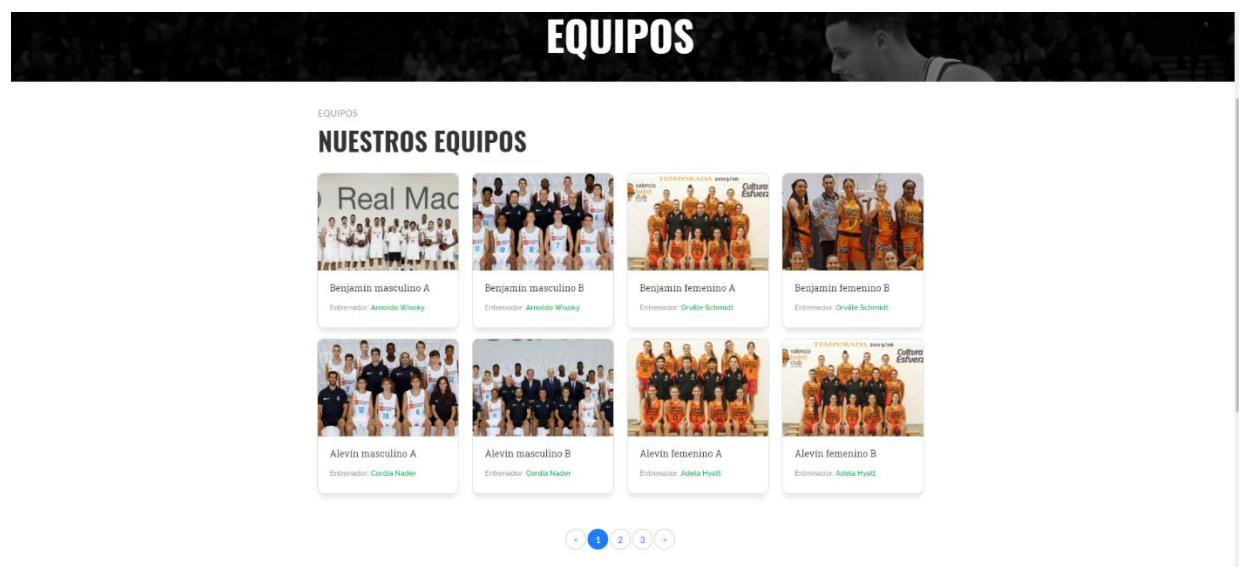


Figura 35 – Ventana equipos

Al pasar el ratón por una de las cards de los equipos se muestran los horarios de entrenamientos del equipo en cuestión.



Figura 36 – Card Equipo cursor encima

Y al hacer click sobre una de ellas lleva a la ventana de detalle del equipo.



Figura 37 -Ventana detalle Equipo

Existen varios elementos de interacción en esta ventana, al hacer click sobre un jugador se muestra una tarjeta con la información del jugador.



Figura 38 - Card

Al hacer click sobre el botón Detalles en cualquiera de los resultados se redirige a la página de detalles del partido.

1.3. Resultados

En esta ventana se muestran los resultados de todos los partidos ya disputados por los equipos y los máximos anotadores, asistentes y reboteadores en esos partidos. Existen varios elementos de interacción. Puede interactuar con los botones inferiores para moverse entre las diferentes páginas de la vista.

The screenshot displays a web interface titled "RESULTADOS DE LA TEMPORADA". It features a sidebar on the right with filters for "Categoría" and "Sexo", and a "Filtrar" button. The main content area lists four basketball games, each with a date, team logos, scores, and a "Detalles" button. To the right of the games, there are three sections: "Máximos Anotadores", "Máximos Asistentes", and "Máximos Reboteadores", each showing player photos and statistics. At the bottom, there is a pagination bar with numbers 1 through 15.

Fecha	Equipo 1	Puntuación 1	Equipo 2	Puntuación 2
MARTES 15 DE JUNIO DE 2021	INF FEM B	96	AYAMONTE	71
JUEVES 10 DE JUNIO DE 2021	ISLA CRISTINA	72	BEN MAS B	116
LUNES 07 DE JUNIO DE 2021	BEN MAS A	112	ISLA CRISTINA	95
VIERNES 04 DE JUNIO DE 2021	CAD MAS B	137	HUELVA	60

Máximos Anotadores

Nombre	Puntos	Partidos
Jeff Murazik	25 pts	1 partidos
Camron Pfannerstill	24 pts	2 partidos
Joan Borer	24 pts	2 partidos
Shyann Walsh	23 pts	1 partidos

Máximos Asistentes

Nombre	Asistencias	Partidos
Jennyfer Streich	15 ast	1 partidos
Elijah Wolff	12.5 ast	4 partidos
Stefan Vandervort	11 ast	2 partidos
Jered Leuschke	10.67 ast	2 partidos

Máximos Reboteadores

Nombre	Rebotes	Partidos
Jennyfer Streich	20 rts	1 partidos
Gladycce Bashirian	13 rts	1 partidos
Elijah Wolff	12 rts	4 partidos
Jackeline Farrell	11 rts	2 partidos

Figura 39 – Ventana Resultados

En el filtro de la esquina superior derecha se puede elegir un sexo y un género para filtrar por ellos. Para realizar el filtrado se deben escoger los valores deseados y hacer click en el botón Filtrar. Si se desea restablecer alguno de los filtros se deben retornar a sus valores iniciales “Categoría” y “Sexo”.

Al hacer click sobre el botón Detalles en cualquiera de los resultados se redirige a la página de detalles del partido. En esta ventana se puede consultar las estadísticas del partido y hacer click para ver el video del partido desde Youtube.

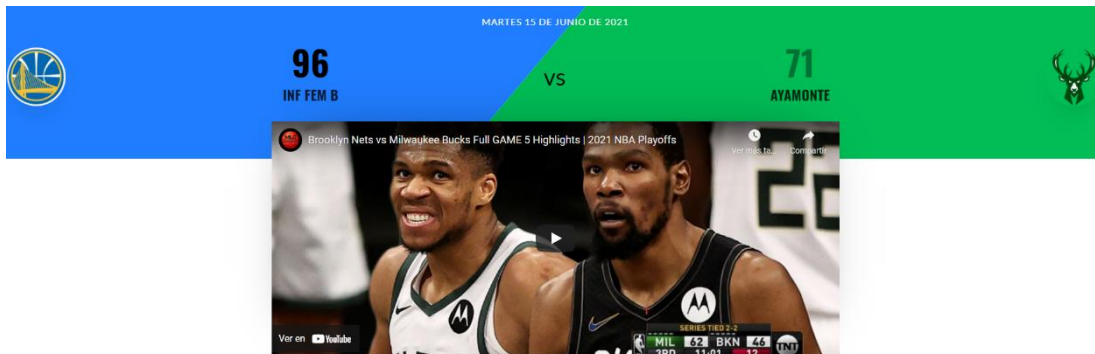


Figura 40 -Ventana detalles de partido (primera parte)

GAME REPORT					
ESTADÍSTICAS					
#	Jugador	Minutos	Puntos	Asistencias	Rebotes
20	DORA WILLIAMSON	24	13	9	7
24	LEONE SCHNEIDER	26	12	1	12
32	GERTRUDE WYMAN	1	10	2	5
36	GILBERT KUB	6	2	1	5
44	MACI LAKIN	9	3	4	4
49	JERROD BOSCO	2	4	2	4
54	EBONY BAILEY	25	0	6	2
57	WATSON ABSHIRE	30	25	2	0
70	DURWARD SWANIAWSKI	5	0	3	1
82	VERLA WILDERMAN	6	9	2	4
90	GERDA ORTIZ	23	2	2	10
92	JOAQUIN CONROY	21	16	1	13
#		200	96	35	67







Máximos anotadores	
	
Watson Abshire 25 puntos	Joaquin Conroy 16 puntos
Máximos asistentes	
	
Dora Williamson 9 asistencias	Ebony Bailey 6 asistencias
Máximos reboteadores	
	
Joaquin Conroy 13 rebotes	Leone Schneider 12 rebotes

Figura 41 – Ventana detalles de partido (segunda parte)

1.4. Contacto

En esta ventana se muestra un formulario de contacto básico, debe introducir los datos correctamente, imprescindible introducir un email válido para poder enviar el mensaje.

CONTACTO

CONTACTA CON NOSOTROS

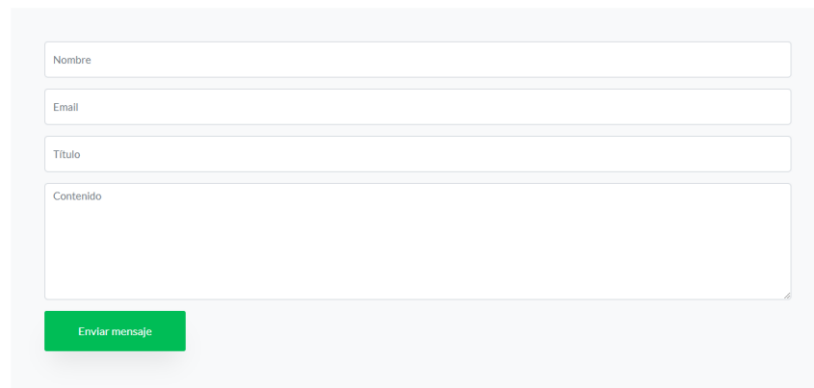
Un formulario de contacto con un fondo gris claro. En la parte superior, el texto "CONTACTO" en gris y "CONTACTA CON NOSOTROS" en negro. El formulario contiene cuatro campos de entrada blancos con bordes grises: "Nombre", "Email", "Titulo" y "Contenido" (un área de texto más grande). Debajo de los campos hay un botón rectangular de color verde con el texto "Enviar mensaje" en blanco.

Figura 42 – Ventana Contacto

1.5. Iniciar sesión

Se muestra una ventana con un formulario para introducir el usuario y la contraseña.

Una ventana de inicio de sesión con un fondo gris claro. En la parte superior, el título "Iniciar Sesión" en azul, flanqueado por líneas horizontales. Debajo hay dos campos de entrada blancos con bordes grises: "Usuario" y "Contraseña". Debajo de estos campos hay un botón rectangular con el texto "Acceder". En la parte inferior, hay un logo de "C.B. La Arboleda" (un icono de árbol) y el texto "©2021 Desarrollado por Javier Cordero Toscano".

Figura 43 – Ventana Iniciar sesión

En caso de que los datos introducidos sean correctos, se inicia sesión, cambia el menú y se muestra el nombre e imagen del usuario. Sobre esta imagen se puede pasar el cursor lo que despliega un listado de opciones. El de cerrar sesión cierra la sesión del usuario y, los otros dos, redirigen a nuevas ventanas.

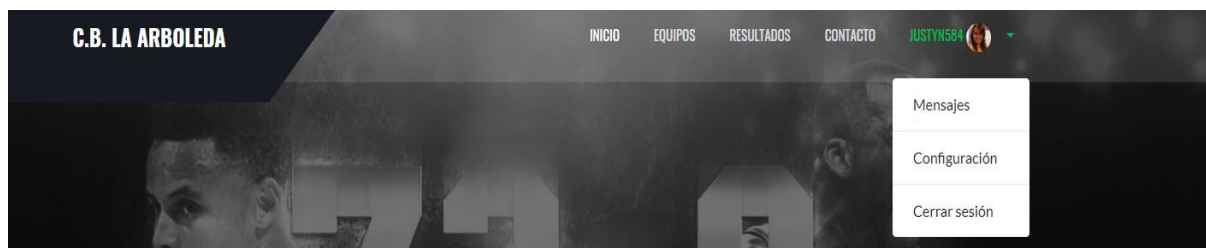


Figura 44 – Desplegable menú principal

1.6. Mensajes

Se abre una ventana nueva en la que aparece un listado de los mensajes recibidos por el usuario y un botón para escribir un nuevo mensaje.

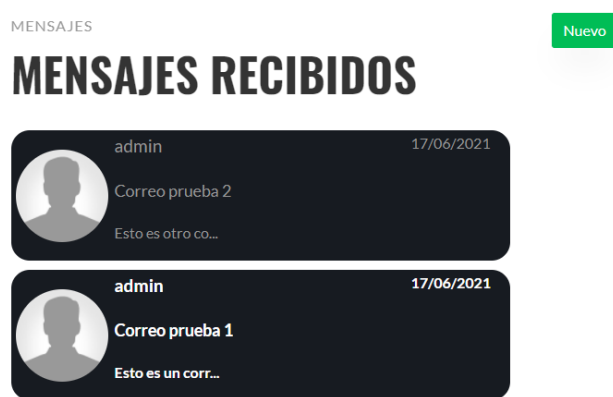


Figura 45 – Mensajes

Si se hace click sobre alguno de los mensajes se muestra al lado el contenido del mismo detallado.




Figura 46 – Mensaje detallado

Si se hace click sobre el icono rojo de la papelera se elimina el mensaje.

Si se hace click sobre el botón Nuevo se muestra un formulario para enviar mensaje a uno, o varios usuarios. Al darle al botón enviar se envía el mensaje.

MENSAJES

MENSAJES RECIBIDOS




admin

17/06/2021

Correo prueba 2

Esto es otro co...



admin

17/06/2021

Correo prueba 1

Esto es un corr...

Nuevo

Destinatarios:

Título

Cuerpo

Enviar

Figura 47 – Formulario nuevo mensaje

1.7. Configuración

Se muestra una nueva ventana con un formulario para introducir la contraseña actual del usuario y la nueva contraseña que se desea. Al introducir los datos deseados se ha hacer click sobre el botón Guardar, si la contraseña actual coincide se realiza el cambio de contraseña.

CONFIGURACIÓN

CONFIGURACIÓN DEL USUARIO

Contraseña actualizada

Contraseña acutal:

Contraseña nueva:

Enviar

Figura 48 – Formulario cambio de contraseña

2. Aplicación administrativa

Para acceder a la aplicación administrativa se debe pasar por la página web pública y hacer click en el botón Iniciar Sesión del menú principal. Al hacerlo se redirige al formulario de iniciar sesión y, una vez introducidos los datos correctamente, se redirige a la aplicación.

2.1. Menú superior

El menú superior cuenta con varios elementos de interacción: un icono de bacadillo que permite hacer más pequeño o agrandar el menú lateral, un icono de correo electrónico que redirige a la ventana de mensajes y el nombre e imagen del usuario que despliega diferentes opciones al hacer click sobre él.

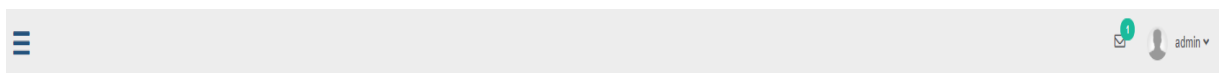


Figura 49 – Menú superior sin desplegar

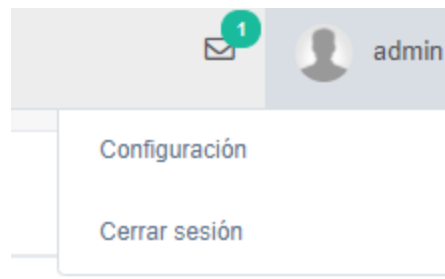


Ilustración 2.2 – Desplegable de opciones menú principal

Al hacer click sobre Cerrar Sesión se cierra la sesión del usuario y se redirige a la página web pública.

Al hacer click sobre Configuración se redirige a una nueva ventana para cambiar la configuración del usuario.

2.2. Mensajes

Se abre una ventana nueva en la que aparece un listado de los mensajes recibidos por el usuario y un botón para escribir un nuevo mensaje.

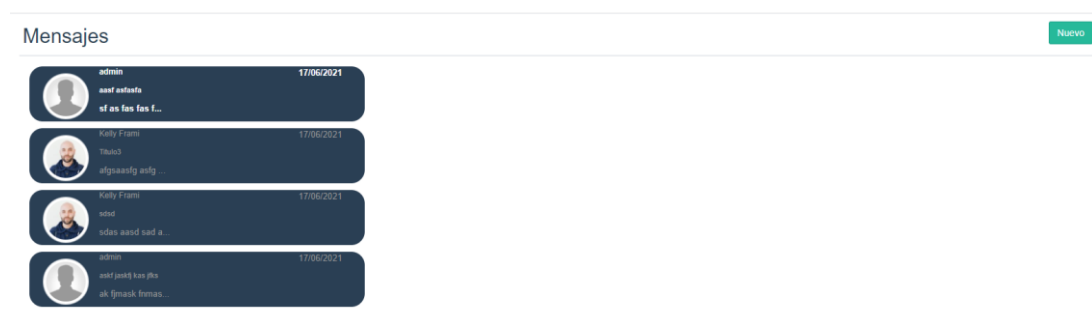


Figura 50 – Mensajes

Si se hace click sobre alguno de los mensajes se muestra al lado el contenido del mismo detallado.



Figura 51 – Mensaje detallado

Si se hace click sobre el icono rojo de la papelera se elimina el mensaje.

Si se hace click sobre el botón Nuevo aparece una ventana modal con un formulario para enviar mensaje a uno, o varios usuarios. Al darle al botón enviar se envía el mensaje.

The image shows a modal window titled 'Nuevo mensaje' with a close button (X) in the top right corner. The form contains three input fields: 'Para:' (To), 'Titulo:' (Title), and 'Contenido:' (Content). The 'Contenido' field is a larger text area. A blue button labeled 'Guardar' (Save) is positioned at the bottom right of the form.

Figura 52– Formulario modal nuevo mensaje

2.3. Configuración

Se muestra una nueva ventana con un formulario para introducir la contraseña actual del usuario y la nueva contraseña que se desea. Al introducir los datos deseados se ha hacer click sobre el botón Guardar, si la contraseña actual coincide se realiza el cambio de contraseña.

Configuración

Contraseña actual:

Contraseña nueva:

Enviar

Figura 53 – Formulario de ventana Configuración

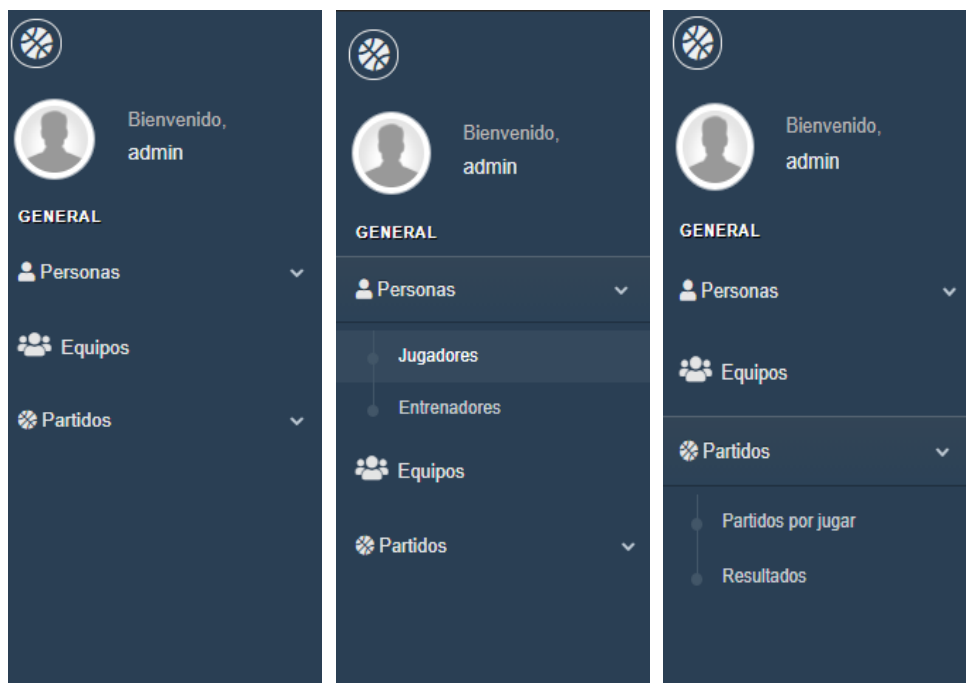
2.4. Menú lateral

El menú lateral cuenta con tres elementos.

Al hacer click sobre Personas se despliegan otras dos opciones: Jugadores y Entrenadores, las cuales, al hacer click sobre alguna de ellas redirige a la ventana del listado correspondiente.

Al hacer click sobre Equipos redirige a la ventana del listado de equipos.

Al hacer click sobre Partidos se despliegan otras dos opciones: Partidos por jugar y Resultados, las cuales, al hacer click sobre alguna de ellas redirige a la ventana del listado correspondiente.



Figuras 54 , 55 y 56 – Menú lateral sin desplegar y desplegando opciones

**Nota: Las funcionalidades que se han visto hasta ahora son independientes del rol de usuario que inicie sesión. A partir de aquí, las ventanas pueden tener más o*

menos funcionalidades activas dependiendo del rol del usuario y de los requisitos de cada ventana.

2.5. Elementos comunes en todos los listados

A continuación, se detallan los elementos comunes con los que se pueden interactuar en cualquiera de los listados que se abren desde el menú lateral descrito anteriormente.

Al abrir un listado se muestra una ventana como esta:





















Id	Nombre	Apellidos	Teléfono	Fecha de nacimiento	Número favorito	Acciones
1	Elijah	Wolff	205 614 5665	2004-05-19	74	 
2	Alicia	Pollich	+1 (680) 478-4424	2013-05-18	97	 
3	Shaina	Schoen	520 842 0379	2013-02-27	31	 
4	Niya	Swaniawski	502-823-8793	2013-08-23	89	 
5	Hershel	Bins	+1 (641) 484-1529	2006-08-07	58	 
6	Jackeline	Farrell	+1-351-570-2599	2012-05-20	4	 
7	Coby	Swift	669 213 4288	2012-08-27	99	 
8	Sherwood	Hudson	(312) 476-3759	2005-09-04	55	 
9	Diamond	Schoen	(419) 873-9913	2006-05-03	97	 
10	Stefan	Vandervort	+1-715-729-8263	2003-09-24	82	 

Figura 57 – Ventana Listado Jugadores

En la tabla se puede interactuar con varios elementos

Mostrar [10] registros

Es un desplegable para cambiar el número de registros por los que se pagina la tabla.

Al escoger una de las opciones se cambia automáticamente.

Cuadro de texto Buscar

Permite filtrar por cualquier columna de la tabla con solo introducir una letra, o borrarla, la tabla se va filtrando automáticamente.

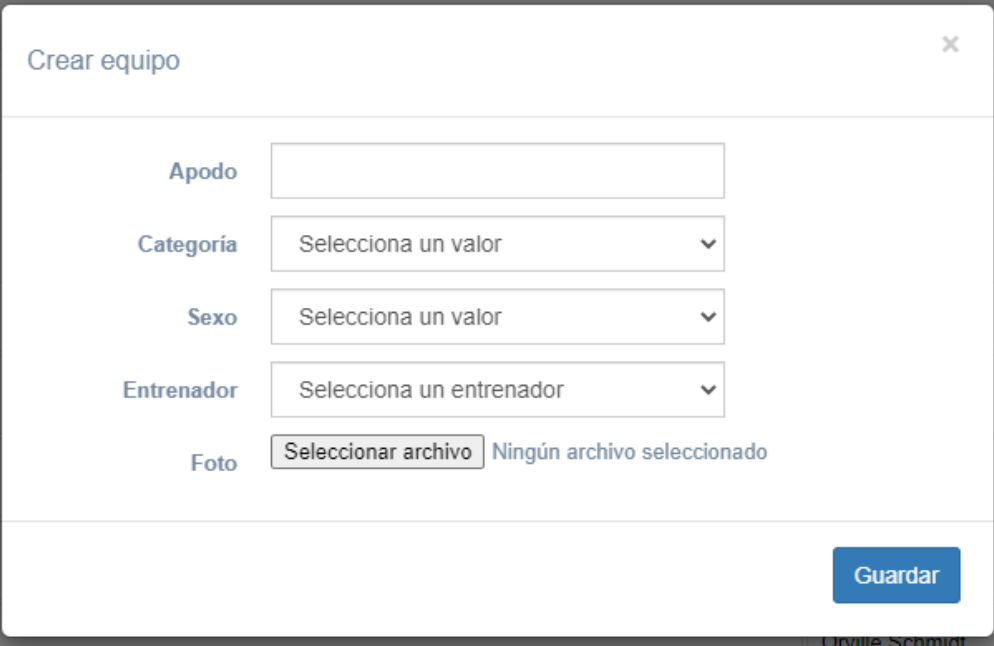
Enlaces de paginación

En la esquina inferior derecha de la tabla se muestran unos enlaces que permiten cambiar la página de registros de la tabla con solo hacer click sobre uno de estos enlaces.

Botón Nuevo

Al hacer click sobre el botón nuevo se abre una ventana modal con un formulario para añadir un nuevo registro a la tabla correspondiente. Se deben rellenar todos los

datos que el formulario va requiriendo (la imagen es opcional en todos los formularios).



Crear equipo

Apodo

Categoría

Sexo

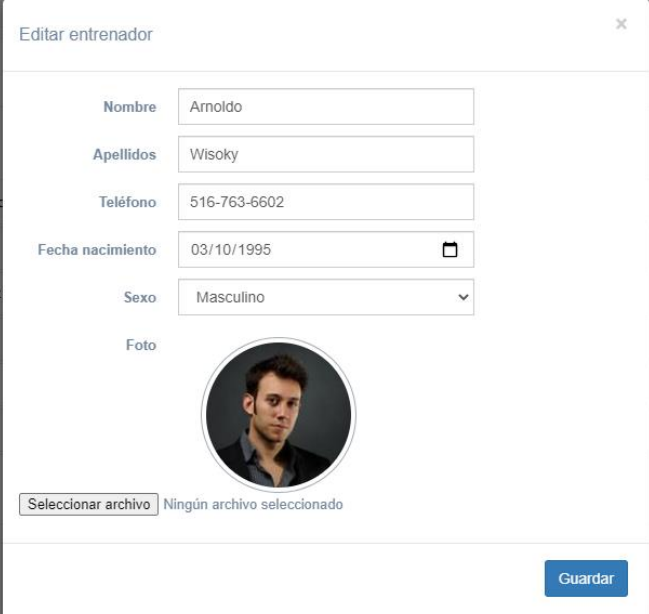
Entrenador

Foto Ningún archivo seleccionado

Figura 58 – Ventana modal Crear Equipo

Icono del lápiz

Al hacer click sobre el icono se abre una ventana modal con un formulario para editar el registro seleccionado. Se deben rellenar todos los datos que el formulario va requiriendo (la imagen es opcional en todos los formularios).



Editar entrenador

Nombre

Apellidos

Teléfono

Fecha nacimiento

Sexo

Foto Ningún archivo seleccionado

Figura 59 – Ventana modal Editar entrenador

Icono del ojo

Al hacer click sobre el icono del ojo se abre la página de detalle del registro seleccionado.

Icono de la papelera

Al hacer click sobre el icono de la papelera se muestra una alerta con dos opciones. Una para cancelar la operación y otra para confirmarla y eliminar el registro definitivamente.

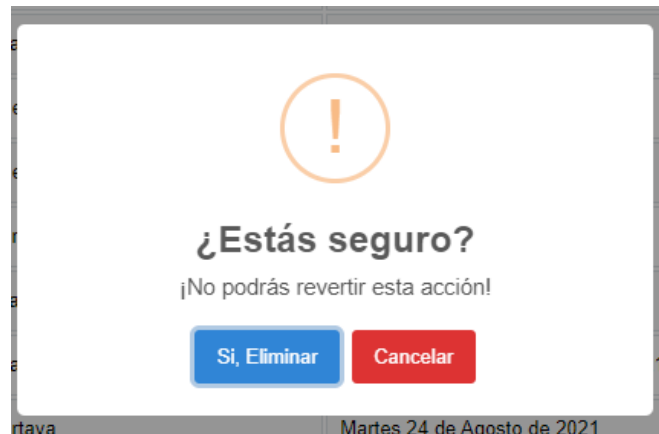


Figura 60 – Modal Confirmar eliminar partido

Icono de la suma (Listado resultados)

Al hacer click sobre el icono de la suma se muestra un modal para añadir el enlace de Youtube del partido. Se debe copiar en enlace en el cuadro de texto y pulsar sobre el botón Guardar.

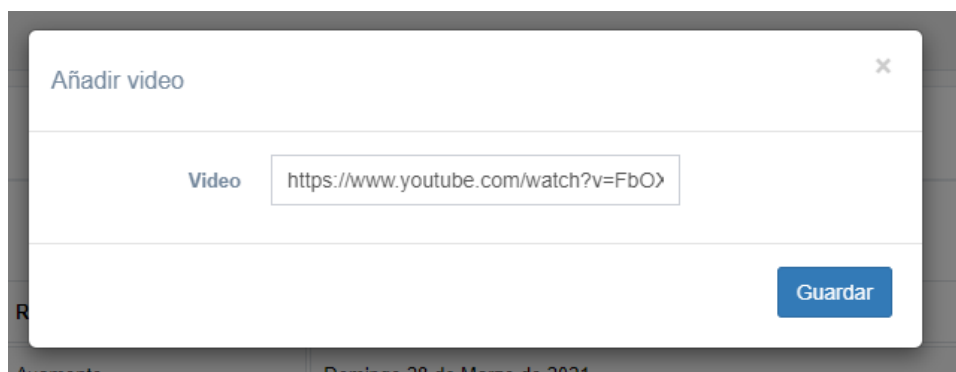


Figura 61 – Modal Añadir video

Es indispensable para que el enlace funcione correctamente que copie el enlace desde el botón compartir que aparece en el video de Youtube.

Últimos partidos


Fecha	Resultado	Minutos	Puntos	Asistencias	Rebotes
15-02-2021	76 - 83	30	5	15	20
02-02-2021	84 - 95	33	4	15	5
24-01-2021	82 - 70	33	18	9	14
09-01-2021	80 - 98	25	17	11	9

Figura 65 – Componente Últimos partidos en detalle de Jugador

2.7. Detalle de entrenador

En la ventana de detalle de un entrenador se puede interactuar únicamente con el listado de equipos que aparece. Si se hace click sobre uno de los equipos se redirige a la página de detalle del equipo seleccionado.

Arnoldo Wisoky



Arnoldo Wisoky

📞 516-763-6602

✉️ kub.jamel@example.com

Equipos

Nombre	Días de entrenamiento
Benjamín masculino A	Viernes: 02:00:00 Viernes: 02:55:00 Viernes: 09:31:00
Benjamín masculino B	Lunes: 10:27:00 Jueves: 09:37:00 Jueves: 02:20:00
Infantil femenino A	Martes: 21:22:00 Miércoles: 12:27:00 Miércoles: 08:44:00
Infantil femenino B	Miércoles: 05:30:00 Viernes: 09:22:00 Viernes: 16:32:00

Historial

Equipo	Comienzo	Fin
Benjamín masculino A	15-09-2020	Actual
Benjamín masculino B	15-09-2020	Actual
Infantil femenino A	15-09-2020	Actual
Infantil femenino B	15-09-2020	Actual

Figura 66 – Ventana Detalle de Entrenador

2.8. Detalle de equipo

En la ventana de detalle de un equipo se encuentran múltiples elementos con los que se puede interactuar. A continuación, se detallan divididos por los módulos diferenciados de la ventana.

6.1.1. Módulo jugadores

Jugadores

[Añadir](#)


































#	Nombre	Acciones
65	Tomasa Cumberata	  
76	Hortense Heaney	  
84	Stanton Mosciski	  
90	Adolf Conroy	  
97	Justice Feeney	  
80	Lonnie Bode	  
72	Dayna Roberts	  
2	Angela Volkman	  
93	Madelynn Funk	  
58	Audrey Pfeffer	  
6	Aiyana Rempel	  

Figura 67 – Módulo Jugadores en Detalle de equipo

Dentro del módulo jugadores se puede interactuar con múltiples elementos. A continuación, se detallan.

Icono de la papelera

Al hacer click sobre el icono de la papelera se muestra una alerta con dos opciones. Una para cancelar la operación y otra para confirmarla y eliminar al jugador del equipo.

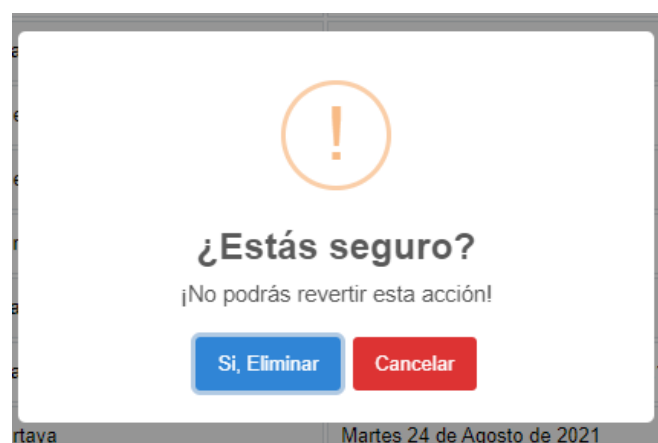


Figura 68 – Modal Confirmar eliminar partido

Icono del ojo

Al hacer click sobre el icono del ojo se abre la página de detalle del jugador seleccionado.

Icono del avión de papel

Al hacer click sobre el icono del avión de papel se abre una ventana modal que permite escoger un equipo. Se debe pulsar sobre el botón Guardar para confirmar la operación.

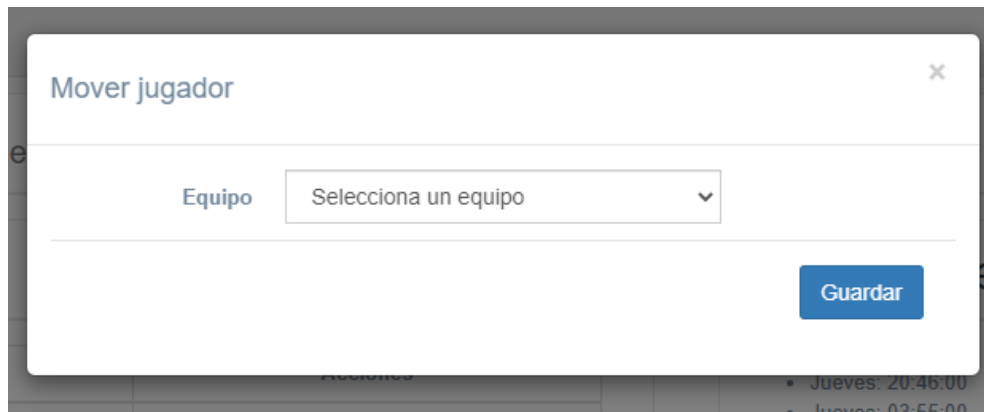


Figura 69 – Modal mover Jugador

6.1.2. Módulo Entrenamientos



Figura 70 – Módulo Entrenamientos en Detalle de Equipo

En el modulo entrenamientos se puede interactuar con el botón Cambiar horarios. Al hacerlo, se muestra una ventana modal con un formulario para seleccionar los días y horas de entrenamiento. Se deben rellenar todos los campos y pulsar sobre el botón Guardar.

Modificar días y horarios de entrenamiento

Lunes	▼	20:00	🕒
Miércoles	▼	20:00	🕒
Jueves	▼	20:00	🕒

Guardar

Figura 71 – Modal modificar días y horarios de entrenamiento

6.1.3. Módulo Próximo partido

Próximo partido

Hacer convocatoria

Fecha	Rival	Lugar
25-06-2021	Cartaya	Visitante

Figura 72 – Módulo Próximo partido en Detalle de equipo

Al pulsar sobre el botón Hacer convocatoria se muestra una ventana modal en la que se muestran dos listados: Jugadores propios y Jugadores convocables. Si se pulsa sobre alguno de los dos títulos se despliega o se encoge el listado según la forma que tuviera anteriormente. Se deben seleccionar los jugadores y pulsar sobre el botón Guardar.

Hacer convocatoria

Jugadores propios:

- ☐ Donnell Boehm
- ☐ Trever Jenkins
- ☒ Gayle Johns
- ☒ Zella Kuvalis
- ☒ Ayana Schinner

Jugadores convocables:

Guardar

Figura 73 – Modal Hacer convocatoria

6.1.3. Módulo Próximos partidos

Próximos partidos

Añadir

Fecha	Rival	Lugar
25-06-2021	Cartaya	Visitante
26-06-2021	Aljaraque	Local
01-08-2021	Huelva	Visitante
14-08-2021	Cartaya	Visitante
18-08-2021	Huelva	Local

Figura 74 – Módulo Próximos partidos en Detalle de equipo

Al hacer click sobre el botón Añadir se muestra una ventana modal con un formulario. Deben rellenarse todos los datos y pulsar sobre el botón Guardar.

The modal window titled "Nuevo partido" contains the following fields:

- Rival:** A dropdown menu with the placeholder text "Selecciona un valor".
- Fecha:** A date input field with the placeholder text "dd/mm/aaaa" and a calendar icon.
- Lugar:** A dropdown menu with the placeholder text "Selecciona un valor".

A blue button labeled "Guardar" is located at the bottom right of the modal.

Figura 75 – Modal Nuevo partido