

MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Web

YOLISPORT

Autor: Alberto León González

Tutor: Fernando Méndez Castellar

Fecha de entrega: 18/06/2023

Convocatoria: 2S – 2223

Documentos del proyecto: [carpeta drive](#)



Índice de contenidos

1. INTRODUCCIÓN	3
1.1. Motivación	3
1.2. Abstract.....	4
1.3. Objetivos propuestos (generales y específicos)	5
2. METODOLOGÍA USADA.....	6
3. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO	7
4. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN.....	9
5. ANÁLISIS DEL PROYECTO	10
6. DISEÑO DEL PROYECTO	17
7. DESPLIEGUE Y PRUEBAS	22
8. CONTEXTO LABORAL.....	26
9. INSTALACIÓN Y CONFIGURACIÓN	27
10. POSICIONAMIENTO SEO	29
11. CONCLUSIONES	30
12. VÍAS FUTURAS.....	30
13. GLOSARIO	31
14. BIBLIOGRAFÍA/WEBGRAFÍA.....	31
15. ANEXOS	33

12.1 Anexo I	35
---------------------------	-----------

12.2 Anexo II	35
----------------------------	-----------

En la normativa de proyectos vigente encontrarás una breve descripción de cada uno de estos apartados para saber qué información debes incluir en ellos

1. Introducción

Durante este proyecto, mostraré, o al menos intentaré hacerlo, todo el proceso de desarrollo de una aplicación desde el principio hasta el final, incluyendo su implementación y explicando y profundizando en cada etapa.

Este proyecto abarca todos los módulos que hemos aprendido a lo largo de este ciclo, dando más énfasis, por supuesto, a algunos módulos más que a otros, según las características del proyecto.

Como es natural en cualquier proyecto de esta índole, surgieron varios contratiempos durante el desarrollo, pero gracias a los recursos disponibles, pudimos resolverlos de la mejor manera posible. Esto tiene dos beneficios: en primer lugar, identificar los errores, y, en segundo lugar, ser capaces de buscar información, hacer preguntas y documentarnos para solucionarlos por nosotros mismos.

Todo esto está debidamente documentado y justificado en las siguientes secciones, utilizando un lenguaje y expresiones lo más neutrales posible para que sea comprensible para la mayoría de las personas involucradas en el desarrollo web, desde principiantes hasta expertos.

Inicialmente, el tema de la aplicación iba a ser una aplicación web de gestión de citas previas, ya sea en talleres, peluquerías y demás servicios, pero tras consultas con mi alrededor, me di cuenta que tenían una necesidad, la cual desarrollaré en el siguiente apartado.

1.1. Motivación

Los precedentes que me han motivado a realizar este proyecto vienen fundamentados en querer aplicar y ampliar los conocimientos adquiridos durante el ciclo formativo poniendo en común todas las asignaturas y tratar de conseguir desplegar la aplicación de manera correcta y sin inconvenientes.

El trasfondo de esta temática es para ayudar a una compañera que desea llevar un control de los entrenamientos que envía a sus clientes realizados en hojas de cálculo y enviados plataformas de mensajería.

Con este proyecto intentaremos que toda la comunicación de estos entrenamientos y el registro de los mismos se haga a través de una aplicación web. Para hacerlo más accesible y cómodo para todos.

En ella, en función del usuario que acceda tendrá una serie de privilegios como administrador o como usuario.

- **Administrador:** tendrá la posibilidad de crear, modificar o eliminar los usuarios. Puede crear los tipos de entrenamientos que desee generar, así como los ejercicios para cada tipo. Estos ejercicios estarán compuestos por el nombre el ejercicio, las series a realizar y las repeticiones que se deben hacer en cada serie. También se indicará el tiempo de descanso entre series. Se habilitará un espacio para la inclusión de comentarios que el entrenador desee comunicar al entrenado.
- **Usuario:** tendrá la posibilidad de visualizar e interactuar con la web, sin tener los privilegios de los que dispone el administrador.

1.2. Abstract

The project consists of a web application focused on the management of physical training. The trainer will set up sessions for his students, including the date they are to perform the exercise, the type of exercises to be performed for that specific day, and also the type of exercise, including sets, repetitions, and rest time.

For this reason, the site administrator will have various CRUD to handle with the users, type of workouts and exercises for each workout.

The application is developed in html, css, javascript and php, mainly.

The objective is to obtain a simplicity in the communication between user and trainer.

1.3. Objetivos propuestos (generales y específicos)

Desde una perspectiva general, queremos proponernos como objetivo principal mejorar la fluidez y la comunicación entre el entrenador, que en nuestro caso también será la persona encargada de tener acceso al panel de administrador, y los usuarios, que serán los clientes.

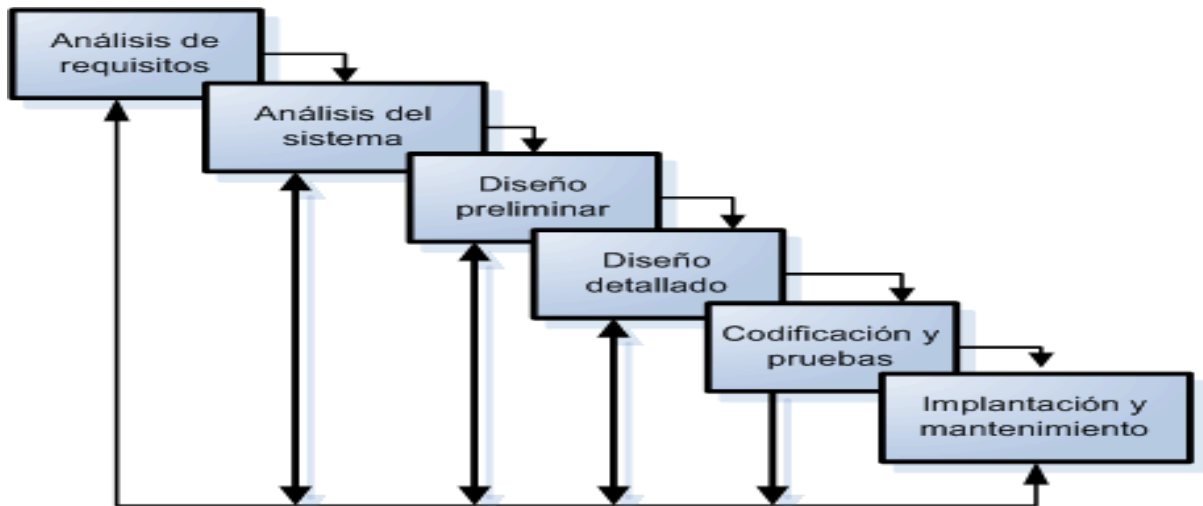
Mediante esta web, dejaremos atrás programas de terceros y que cada vez más usuarios sean conocedores de nuestra web, y, sobre todo, fidelizar los clientes, es decir, una vez hagan uso de nuestra aplicación, retener el tráfico. Ese es nuestro objetivo principal.

De una manera más específica:

- El entrenador/administrador podrá asignar entrenamientos específicos para cada usuario registrado previamente en la web, en relación a su condición física y sus objetivos.
- El entrenador/administrador podrá insertar ejercicios para posteriormente crear los entrenamientos para los usuarios.
- Ofrecer una experiencia para el usuario fácil e intuitiva.
- Realizar un seguimiento de las estadísticas internas de la web, ya sea del tráfico, uso diario, género...
- Implementar una base de datos segura tanto para el usuario como el administrador.
- Asegurarme de que la web está potenciada y optimizada para los motores de búsqueda (posicionamiento SEO).

2. Metodología usada

Como metodología en principio íbamos a usar *SCRUM*, pero tras revisar el proyecto creemos que lo más idóneo es usar el método de **Cascada con retroalimentación**.



Esta metodología se ajusta más a un proyecto pequeño y bien definido en la cual podemos avanzar entre etapas sin demasiados contratiempos, pero nos permite volver atrás si detectamos algún error en alguna etapa ya finalizada.

Las diferentes etapas de las que está compuesta esta metodología son:

- Análisis donde se analizarán los puntos importantes del proyecto, así como su viabilidad.
- Diseño en la cual se realizará toda la parte teórica del proyecto incluyendo los diferentes diagramas: de casos de uso, entidad-relación, modelo relacional y de clases.
- Codificación donde se implementará toda la aplicación siguiendo todo el diseño generado con anterioridad.
- Pruebas para verificar que todo funciona correctamente.
- Mantenimiento donde podremos incluir mejoras o solucionar errores detectados en el proyecto.

3. Tecnologías y herramientas utilizadas en el proyecto

Las tecnologías que utilizaremos en el proyecto serán las siguientes:

Primero para crear la estructura y la apariencia usaremos lo aprendido en la asignatura *Lenguaje de marcas*.

- **HyperText Markup Language** (en adelante HTML) es un lenguaje de marcado que define la estructura de tu contenido. HTML consiste en una serie de elementos que usarás para encerrar diferentes partes del contenido para que se vean o comporten de una determinada manera. (Mozilla, s.f.)
- **Cascading Style Sheet** (en adelante CSS) es el lenguaje de estilos que se usa para crear la presentación y dotar de un diseño agradable a los documentos HTML o XML. (Mozilla, s.f.)
- **Bootstrap**, que está permitido según la normativa actual 1S2223. Aportará que la app web sea responsive y el diseño goce de modernidad, además de reducir nuestro CSS.



Para dotar de interactividad al proyecto y para una parte de la comunicación entre cliente y servidor usaremos lo aprendido en la asignatura *Desarrollo web entorno cliente*.

- **Javascript** (en adelante JS) es un lenguaje ligero, interpretado y orientado a objetos con funciones de primera clase, y mejor conocido como el lenguaje de programación para las páginas Web, pero también se utiliza en muchos entornos que no son de navegador. (Mozilla, s.f.)



Toda la parte del backend será realizado con **PHP** con todo lo visto en la asignatura *Desarrollo web entorno servidor*. Con el realizaremos todas las interacciones con la base de datos.



Para el almacenamiento y persistencia de datos usaremos una base de datos SQL con los conocimientos adquiridos en la asignatura *Bases de datos*. Para solicitar las distintas operaciones a realizarse en la base de datos usaremos el **Structured Query Language** (en

adelante SQL) un lenguaje de programación diseñado para actualizar, obtener, y calcular información en bases de datos relacionales.

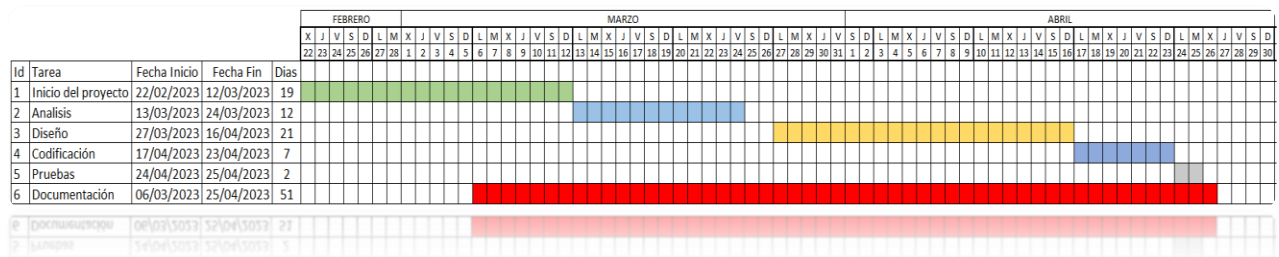
Las herramientas a usar en el proyecto serán las siguientes:

- **XAMPP** que nos proporciona un servidor **Apache** para poder ejecutar nuestra aplicación y **PHPMyAdmin** como sistema gestor de base de datos (en adelante SGBD). **XAMPP**
- **Apache** estudiado en la asignatura *Despliegue de aplicaciones web*. El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. (Wikipedia, s.f.) **APACHE**
HTTP SERVER PROJECT
- **Sublime Text 3** como editor de texto sin muchas extensiones para no generar archivos de configuración no deseados.
- **Microsoft Edge y Google Chrome** como navegadores para testear el correcto funcionamiento y compatibilidades.
- **Git** como herramienta de control de versiones. Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. (Wikipedia, s.f.)
- **GitHub** es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. (Wikipedia, s.f.)

4. Estimación de recursos y planificación

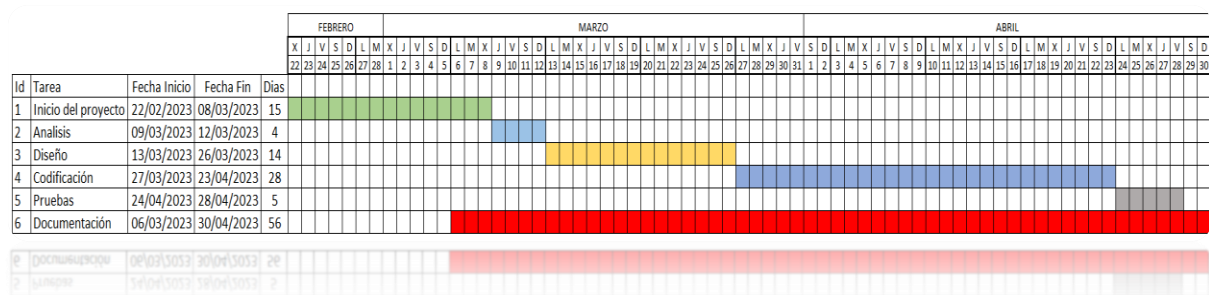
Incluimos el diagrama de Gantt inicial en el cual hemos dado más peso a la parte de análisis y diseño al considerar que si ambas fases quedan perfectamente definidas la codificación resultará mucho más sencilla. La tarea de documentación es la que más ocupa debido a que debemos registrar cada acción que vayamos completando. No se comienza a documentar desde el inicio del proyecto al estar pendiente de la aceptación del tema propuesto.

En todas las tareas hemos sobreestimado el tiempo para cubrir posibles imprevistos, así como por las festividades acaecidas durante la elaboración del proyecto.



En el diagrama de Gantt final podemos comprobar como habíamos sobreestimado en demasía la tarea de análisis y subestimado bastante la parte de codificación.

Esta última parte nos ha llevado más de la cuenta al tener que realizar la parte de backend usando PDO y POO. También nos encontramos con errores a la hora de hacer las peticiones asíncronas entre cliente y servidor por lo que tuvimos que rehacer bastante parte de código.



5. Análisis del proyecto

REQUISITOS FUNCIONALES:

➤ RF001: El cliente debe ser capaz de iniciar sesión en el sistema.	
➤ RF002: El cliente debe ser capaz de cerrar sesión en el sistema.	
➤ RF003: El usuario debe ser capaz de poder visualizar su entrenamiento.	
➤ RF004: El administrador debe ser capaz de eliminar un usuario.	
➤ RF005: El administrador debe ser capaz de crear un nuevo entrenamiento.	
➤ RF006: El administrador debe ser capaz de editar un entrenamiento.	
➤ RF007: El administrador debe ser capaz de eliminar un entrenamiento.	
➤ RF008: El administrador debe ser capaz de crear un nuevo ejercicio asociado a un tipo de entrenamiento	
➤ RF009: El administrador debe ser capaz de editar un ejercicio.	
➤ RF010: El administrador debe ser capaz de eliminar un ejercicio.	
➤ RF011: El usuario debe ser capaz de visualizar todos los entrenamientos creados por el administrador.	
➤ RF012: El administrador debe ser capaz de cerrar sesión en el sistema.	

REQUISITOS NO FUNCIONALES:

➤ RNF001: La aplicación web debe resultar fácil de utilizar para los usuarios.
➤ RNF002: La aplicación web debe ser rápida y optimizada para no tener tiempos de carga.
➤ RNF003: No se podrán generar usuarios con nombre o email que ya existan en la base de datos.
➤ RNF004: Se comprobará en todo momento que el usuario está conectado y en ningún caso se podrá acceder sin haberse hecho la comprobación en la base de datos.
➤ RNF005: El password quedará encriptado en la base de datos.
➤ RNF006: La aplicación mostrará mensajes de error cuando se produzcan.
➤ RNF007: La aplicación funcionará en los navegadores Microsoft Edge y Google Chrome.



En los anexos podemos ver el [diagrama de casos de uso](#). A continuación, tenemos la tabla de especificación de casos de uso:

Caso de Uso	Descripción	Actor Principal	Precondiciones	Flujo Principal	Postcondiciones
RF001	Iniciar sesión	Cliente	El sistema está disponible	1. El cliente ingresa su nombre de usuario y contraseña. 2. El sistema verifica las credenciales proporcionadas. 3. El sistema inicia sesión para el cliente.	El cliente ha iniciado sesión en el sistema.
RF002	Cerrar sesión	Cliente	El cliente ha iniciado sesión en el sistema	1. El cliente solicita cerrar sesión. 2. El sistema cierra la sesión del cliente.	El cliente ha cerrado sesión en el sistema.
RF003	Visualizar entrenamiento	Usuario	El usuario ha iniciado sesión en el sistema	1. El usuario selecciona la opción de visualizar su entrenamiento.	El usuario visualiza su entrenamiento.

Caso de Uso	Descripción	Actor Principal	Precondiciones	Flujo Principal	Postcondiciones
				2. El sistema muestra al usuario su entrenamiento.	
RF004	Eliminar usuario	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador selecciona la opción de eliminar un usuario. 2. El administrador selecciona el usuario a eliminar. 3. El sistema elimina al usuario seleccionado.	El usuario seleccionado ha sido eliminado del sistema.
RF005	Crear nuevo entrenamiento	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador selecciona la opción de crear un nuevo entrenamiento. 2. El administrador ingresa los detalles del entrenamiento. 3. El sistema crea el nuevo entrenamiento.	Se ha creado un nuevo entrenamiento en el sistema.

Caso de Uso	Descripción	Actor Principal	Precondiciones	Flujo Principal	Postcondiciones
RF006	Editar entrenamiento	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador selecciona la opción de editar un entrenamiento. 2. El administrador selecciona el entrenamiento a editar. 3. El administrador modifica los detalles del entrenamiento. 4. El sistema guarda los cambios realizados.	El entrenamiento ha sido editado y los cambios se han guardado.
RF007	Eliminar entrenamiento	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador selecciona la opción de eliminar un entrenamiento. 2. El administrador selecciona el entrenamiento a eliminar. 3. El sistema elimina el entrenamiento seleccionado.	El entrenamiento seleccionado ha sido eliminado del sistema.

Caso de Uso	Descripción	Actor Principal	Precondiciones	Flujo Principal	Postcondiciones
RF008	Crear nuevo ejercicio	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador selecciona la opción de crear un nuevo ejercicio. 2. El administrador ingresa los detalles del ejercicio y lo asocia a un tipo de entrenamiento. 3. El sistema crea el nuevo ejercicio.	Se ha creado un nuevo ejercicio asociado a un tipo de entrenamiento en el sistema.
RF009	Editar ejercicio	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador selecciona la opción de editar un ejercicio. 2. El administrador selecciona el ejercicio a editar. 3. El administrador modifica los detalles del ejercicio. 4. El sistema guarda los cambios realizados.	El ejercicio ha sido editado y los cambios se han guardado.

Caso de Uso	Descripción	Actor Principal	Precondiciones	Flujo Principal	Postcondiciones
RF010	Eliminar ejercicio	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador selecciona la opción de eliminar un ejercicio. 2. El administrador selecciona el ejercicio a eliminar. 3. El sistema elimina el ejercicio seleccionado.	El ejercicio seleccionado ha sido eliminado del sistema.
RF011	Visualizar entrenamientos del administrador	Usuario	El usuario ha iniciado sesión en el sistema	1. El usuario selecciona la opción de visualizar todos los entrenamientos. 2. El sistema muestra al usuario todos los entrenamientos creados por el administrador.	El usuario visualiza todos los entrenamientos creados por el administrador.
RF012	Cerrar sesión	Administrador	El administrador ha iniciado sesión en el sistema	1. El administrador solicita cerrar sesión. 2. El sistema cierra la sesión del administrador.	El administrador ha cerrado sesión en el sistema.

y el [diagrama entidad-relación](#), en los anexos también, donde podemos ver las entidades, los atributos, sus relaciones y sus cardinalidades.

Tras realizar el diagrama entidad relación hemos procedido a la normalización de las tablas para hacer que la base de datos sea consistente y escalable. Hemos obtenido las siguientes tablas:

Las claves primarias están subrayadas y negrita y las claves foráneas en cursiva.

Admin (**Id_PK**, *Id_Usuario_FK*)

Usuario (**Id_PK**, Nombre, Email, Password, *Id_Admin_FK*)

Entrenamiento (**Id_PK**, Tipo, *Id_Usuario_FK*, *Id_Admin_FK*)

Ejercicio (**Id_PK**, Fecha, Nombre, Series, Repeticiones, Descanso, *Id_Admin_FK*, *Id_Entrenamiento_FK*)

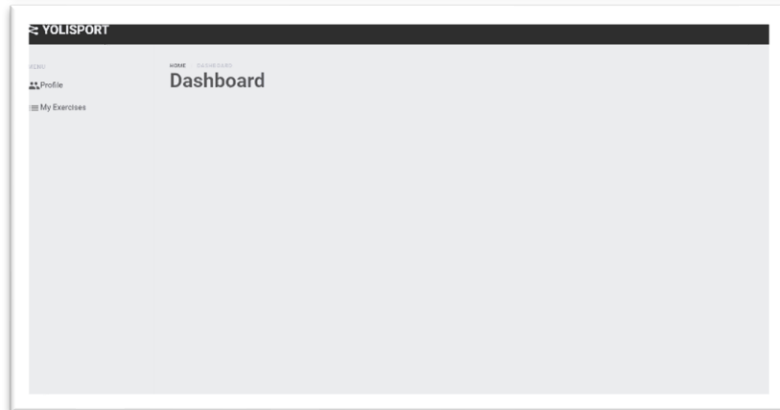
En Anexo podemos encontrar el [Diagrama relacional](#) y el diagrama de clases.

6. Diseño del proyecto

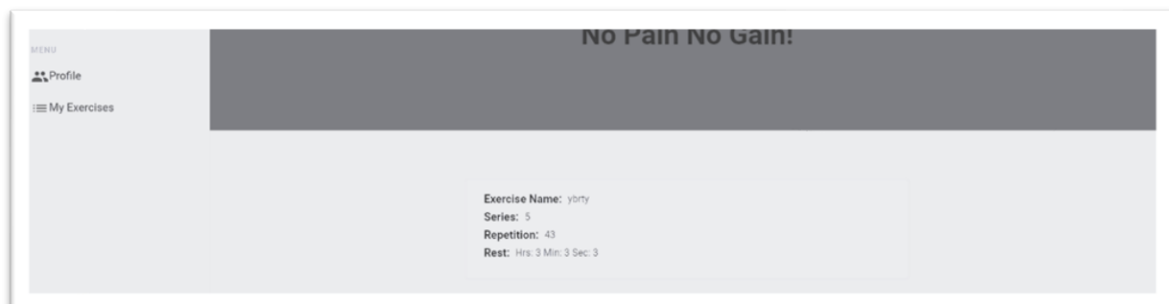
Un mockup, también conocido como maqueta o prototipo de baja fidelidad, es una representación visual de un diseño o idea con el propósito de mostrar una posible apariencia del producto final.

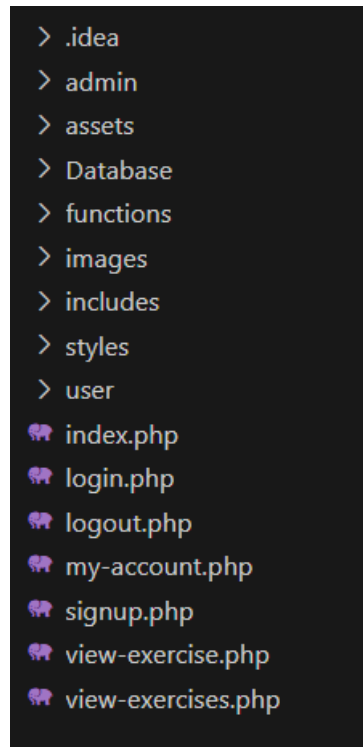
Estos modelos pueden ser creados utilizando diversos formatos, como papel, programas de diseño gráfico o herramientas especializadas de prototipado. Se utilizan comúnmente en el diseño de sitios web, aplicaciones móviles, productos físicos y otros proyectos de diseño.

Los mockups son una herramienta sumamente útil para comunicar ideas de diseño y obtener una visión visual de cómo será el producto final y su funcionamiento. También son una forma efectiva para que diseñadores y desarrolladores experimenten con diferentes opciones de diseño y reciban comentarios valiosos de las partes interesadas antes de invertir grandes recursos y tiempo en la construcción del producto final.



HOME EXERCISE					
Exercise Lists					
Exercise No	Exercise Name	Series	Repetition	Rest	Action
1	ybrty	5	43	3-3-3	View





Aquí tenemos la estructura básica del proyecto, donde tenemos todos los directorios y archivos necesarios para el funcionamiento la de web.

Dentro de functions, nos encontramos con

Respecto a las funciones más importantes, tenemos lo siguiente:

1. Función **getDays ()**: Esta función obtiene los días de la semana desde la tabla **days** en la base de datos y muestra enlaces para cada día. Cada enlace tiene el formato `$day_name`, donde **\$day_id** y **\$day_name** son valores obtenidos de la base de datos. Estos enlaces permiten filtrar los ejercicios según el día seleccionado.
2. Función **getExercise()**: Esta función obtiene los ejercicios desde la tabla **exercises** en la base de datos y muestra enlaces para cada ejercicio. Al igual que en la función **getDays()**, cada enlace tiene la estructura `$exer_name`, donde **\$exer_id** y **\$exer_name** son valores obtenidos de la base de datos. Estos enlaces permiten filtrar los ejercicios según el ejercicio seleccionado.

3. Función **get_Day_Exercises()**: Esta función muestra los ejercicios de entrenamiento según el día seleccionado. Utiliza el parámetro **\$_GET['day']** para obtener el día seleccionado a través de la URL. Luego, realiza una consulta en la base de datos para obtener los ejercicios asociados a ese día y los muestra en forma de productos con información como el nombre del ejercicio, una imagen y el número de series.
4. Función **get_Exer_Exercises()**: Esta función muestra los detalles de un ejercicio específico seleccionado. Utiliza el parámetro **\$_GET['exer']** para obtener el ejercicio seleccionado a través de la URL. Luego, realiza una consulta en la base de datos para obtener los detalles del ejercicio y los muestra en forma de producto con el nombre del ejercicio y una imagen.
5. Función **get_All_Exercises()**: Esta función muestra una selección aleatoria de ejercicios de la base de datos. Si no se ha seleccionado un día o un ejercicio específico, muestra una selección aleatoria de ejercicios utilizando la consulta **SELECT * FROM exercises ORDER BY rand() LIMIT 0,5**.

Por la parte de los estilos, tenemos:

1. El primer bloque de código establece un conjunto de reglas para restablecer los estilos predeterminados de los elementos HTML y proporcionar una base consistente para todos los navegadores.
2. El bloque **body** establece el color de fondo de la página en **#39C** (un tono de azul claro).
3. **.main_wrapper** define un contenedor principal que tiene un ancho de **1000px** y se centra en la página.
4. **.header_wrapper** establece un contenedor para la cabecera de la página con un ancho de **1000px** y una altura de **100px**. El fondo es blanco (**#FFF**).
5. **#navbar** establece el estilo de la barra de navegación, con un ancho de **1000px**, una altura de **50px** y un fondo de color gris (**#999**). El texto es de color blanco (**#FFF**).
6. **#menu** y **#menu li** definen el estilo del menú de navegación. Los elementos de la lista (**li**) se muestran en línea y no tienen viñetas. Los enlaces (**a**) dentro del menú tienen un estilo específico, incluido el color de texto y los espacios alrededor.

7. **#login-btn-signup** establece el estilo del botón de inicio de sesión o registro en la barra de navegación.
8. **.content_wrapper** define un contenedor para el contenido principal de la página, con un ancho de **1000px** y un fondo de color rosa claro (**#F96**).
9. **#right_content** y **#left_sidebar** establecen el estilo del contenido principal y la barra lateral izquierda respectivamente. Tienen anchos y colores de fondo específicos.
10. **.footer** define el estilo del pie de página de la página web, con un ancho de **1000px**, una altura de **75px**, un fondo gris (**#999**) y color de texto blanco (**#FFF**).
11. **#sidebar_title** establece el estilo del título de la barra lateral izquierda.
12. **#days**, **#days li** y **#days a** establecen el estilo de una lista de días de la semana en la barra lateral izquierda, que se utiliza para filtrar los ejercicios según el día seleccionado.
13. **#headline** y **#headline_content** definen el estilo del encabezado principal de la página, que puede contener algún tipo de contenido destacado.
14. **#products_box** establece el estilo de un cuadro que contiene productos o elementos relacionados.
15. **#frm** define el estilo de un formulario con campos de entrada de texto y un botón de envío.
16. **#single_product** establece el estilo de un contenedor que muestra un producto o elemento individual, como una imagen y una descripción.

7. Despliegue y pruebas

Verificar el inicio de sesión del cliente	RF001: El cliente debe ser capaz de iniciar sesión en el sistema	<ol style="list-style-type: none"> 1. Ingresar un nombre de usuario y contraseña válidos. 2. Ingresar un nombre de usuario válido pero una contraseña incorrecta. 3. Intentar iniciar sesión sin ingresar ninguna información. 4. Intentar iniciar sesión con un nombre de usuario no registrado.
Verificar el cierre de sesión del cliente	RF002: El cliente debe ser capaz de cerrar sesión en el sistema	<ol style="list-style-type: none"> 1. Hacer clic en el botón de cerrar sesión. 2. Verificar que el cliente sea redirigido a la página de inicio de sesión después de cerrar sesión.
Verificar la visualización del entrenamiento por parte del usuario	RF003: El usuario debe ser capaz de poder visualizar su entrenamiento	<ol style="list-style-type: none"> 1. Iniciar sesión como usuario. 2. Navegar a la sección de visualización de entrenamientos. 3. Verificar que se muestre correctamente la información del entrenamiento del usuario.

Verificar la eliminación de un usuario por parte del administrador	RF004: El administrador debe ser capaz de eliminar un usuario	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador. 2. Acceder a la lista de usuarios. 3. Seleccionar un usuario para eliminar. 4. Confirmar la eliminación y verificar que el usuario sea eliminado del sistema.
Verificar la creación de un nuevo entrenamiento por parte del administrador	RF005: El administrador debe ser capaz de crear un nuevo entrenamiento	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador. 2. Acceder a la opción de crear un nuevo entrenamiento. 3. Completar los campos requeridos del formulario de creación. 4. Guardar el entrenamiento y verificar que se haya creado correctamente.
Verificar la edición de un entrenamiento por parte del administrador	RF006: El administrador debe ser capaz de editar un entrenamiento	<ol style="list-style-type: none"> 1. Iniciar sesión como administrador. 2. Acceder a la lista de entrenamientos. 3. Seleccionar un entrenamiento para editar.

		<p>4. Realizar los cambios deseados en el entrenamiento.</p> <p>5. Guardar los cambios y verificar que se hayan aplicado correctamente.</p>
Verificar la eliminación de un entrenamiento por parte del administrador	RF007: El administrador debe ser capaz de eliminar un entrenamiento	<p>1. Iniciar sesión como administrador.</p> <p>2. Acceder a la lista de entrenamientos.</p> <p>3. Seleccionar un entrenamiento para eliminar.</p> <p>4. Confirmar la eliminación y verificar que el entrenamiento sea eliminado del sistema.</p>
Verificar la creación de un nuevo ejercicio asociado a un tipo de entrenamiento por parte del administrador	RF008: El administrador debe ser capaz de crear un nuevo ejercicio asociado a un tipo de entrenamiento	<p>1. Iniciar sesión como administrador.</p> <p>2. Acceder a la sección de administración de ejercicios.</p> <p>3. Seleccionar el tipo de entrenamiento al que se desea agregar un nuevo ejercicio.</p> <p>4. Completar los campos requeridos del formulario de creación de ejercicio.</p> <p>5. Guardar el ejercicio y verificar que se haya creado correctamente y esté</p>

		asociado al tipo de entrenamiento correspondiente.
Verificar la eliminación de un ejercicio asociado a un tipo de entrenamiento por parte del administrador	RF009: El administrador debe ser capaz de eliminar un ejercicio asociado a un tipo de entrenamiento	<ol style="list-style-type: none">1. Iniciar sesión como administrador.2. Acceder a la sección de administración de ejercicios.3. Seleccionar el tipo de entrenamiento del cual se desea eliminar un ejercicio.4. Seleccionar el ejercicio a

8. Contexto laboral

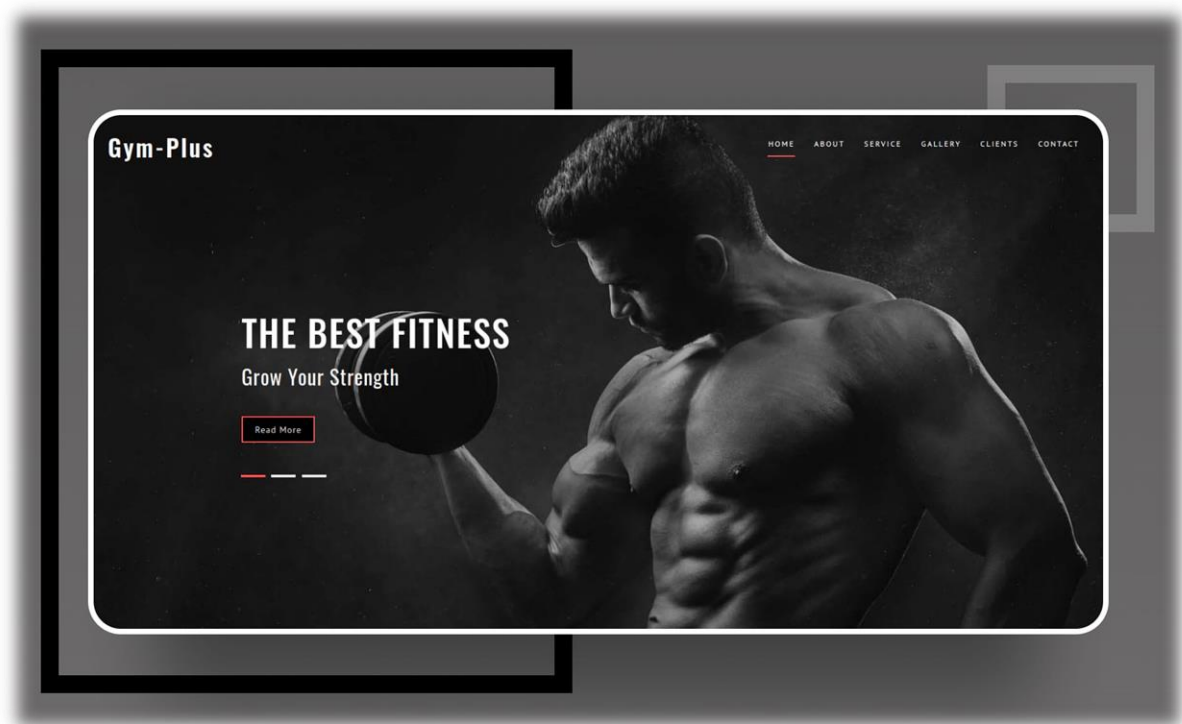
Esta aplicación web, a priori tiene mucho futuro laboral por una serie de motivos. Principalmente porque a medida que la web vaya siendo más conocida, el tráfico aumentará progresivamente, hasta llegar a un punto donde será necesario expandir la empresa, principalmente por programadores, y en un segundo plano, sin dejar de ser menos importante, marketing, recursos humanos, administración...

Y también, entrenadores querrán formar parte de nuestra web a medida que va creciendo. Por lo tanto, sería buena idea añadir algún tipo de funcionalidad premium, por ejemplo, que el entrenador y el alumno puedan comunicarse a través de la web, por ejemplo, con una caja de comentarios.

Esta funcionalidad premium lógicamente sería de pago.

También podría instaurarse algún tipo de suscripción mensual, para que los usuarios puedan acceder a vídeos colgados o vídeos en directo por parte del profesor.

Salidas y futuro laboral tiene mucho, pero para ello, primero se necesita una buena base.



9. Instalación y configuración

Para poder instalar y acceder a la web de manera local sin ningún tipo de inconveniente, debemos tener instalado XAMPP.

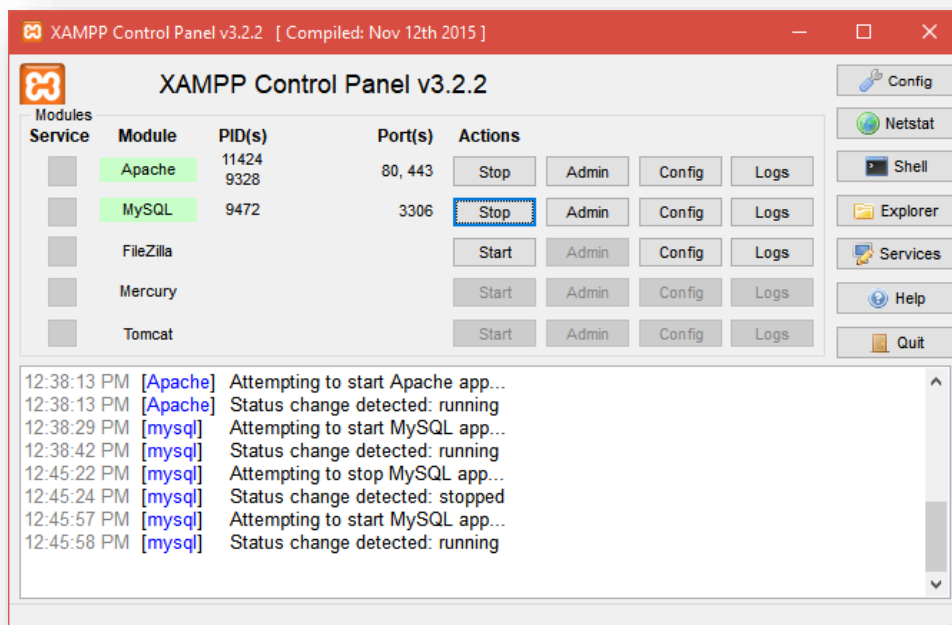
XAMPP cuenta con un servidor independiente, la base de datos MySQL, el servidor web Apache y el intérprete para PHP.

Una vez instalado y configurado (revisar documentación oficial), debemos acceder a la carpeta de XAMPP, normalmente se puede localizar en el disco duro C, y dentro de la carpeta XAMPP, debemos localizar la carpeta htdocs.

En dicha carpeta, debemos pegar la carpeta con nuestro código.

Después, abrimos XAMPP y activamos las dos primeras casillas:

- Apache
- MySQL




Si no tenemos ningún mensaje en rojo, está todo correcto.

Y ya solo queda abrir la web en el navegador, para ello:


Para acceder como usuario normal:

- <http://localhost/YOLISPORT/login.php>



Sign In!

EMAIL ADDRESS:

 admin@admin.com

PASSWORD:

.....

Login

[Create a new Account](#)

Para acceder al panel de control de administrador:

- <http://localhost/YOLISPORT/admin/login.php>

Usuario: admin@admin.com
Password: admin

10. Posicionamiento SEO

Para mejorar el posicionamiento SEO de YOLISPORT, haré lo siguiente:

Investigación de palabras clave: Realizaré un análisis intenso de las palabras clave relevantes para mi web. Utilizaré herramientas como Google Keyword Planner, para identificar las palabras clave más buscadas en mi temática.

Contenido de relevancia: Crearé contenido original, relevante y de alta calidad relacionado con los entrenamientos personalizados que ofrezco. Utilizaré las palabras clave de manera natural en el contenido, incluyendo títulos, encabezados y párrafos.

Optimización de metaetiquetas: Me Aseguraré de que cada página de mi sitio web tenga una etiqueta de título única y descriptiva que contenga palabras clave relevantes. Además, optimizaré las meta descripciones de cada página para que sean atractivas y alentadoras para los usuarios a hacer click.

Optimización de imágenes: Me Aseguraré de que las imágenes que utilices en mi sitio web tengan nombres de archivo descriptivos y utilicen el atributo "alt" para proporcionar una descripción de la imagen. Esto ayuda a los motores de búsqueda a entender de qué trata la imagen y mejora la accesibilidad.

Enlaces internos y externos: Crearé enlaces internos dentro de mi sitio web para ayudar a los motores de búsqueda a descubrir y rastrear todas las páginas. Además, buscaré oportunidades para obtener enlaces externos de calidad hacia mi sitio web, ya sea a través de colaboraciones, directorios o contenido compartido en redes sociales.

Velocidad de carga del sitio web: Optimizaré la velocidad de carga de mi sitio web utilizando técnicas como la compresión de imágenes, el almacenamiento en caché, la minimización de archivos CSS y JavaScript, entre otros. Una página rápida mejora la experiencia del usuario y puede beneficiar al SEO.

Etiquetas de encabezado: Utilizaré las etiquetas de encabezado HTML (H1, H2, H3, etc.) para estructurar jerárquicamente mi contenido. Esto ayuda a los motores de búsqueda a entender la organización del contenido y su relevancia.

Monitoreo y análisis: Utilizaré herramientas como Google Analytics para monitorear el tráfico de mi sitio web, las palabras clave que generan más visitas y otras métricas relevantes. Esto me ayudará a identificar áreas de mejora y ajustar mi estrategia de SEO en consecuencia.

11. Conclusiones

En definitiva, creo que el proyecto cumple con mis expectativas, tanto en diseño, como en funcionalidad. El acabado es correcto, y funciona como me esperaba.

Ahora, toca esperar que los usuarios lo utilicen, y que me proporcionan un feedback para comprobar qué problemas y/o necesidades no están cubiertas aún. Con todos los datos recopilados, habrá que hacer un análisis de la situación y ver que es necesario realmente ampliar o mejorar, y si esto lo realizaremos dentro del contexto laboral o no, refiriéndome a la capacidad de expansión de la empresa, o solo a nivel local.

Sin duda alguna, esta asignatura a favorecido mucho a terminar de encajar todas las piezas que hemos ido aprendiendo a lo largo del ciclo, y conseguir desarrollar una aplicación web de principio a fin. Por supuesto esto continúa, y no va a ser ni mucho menos la última aplicación realizada.

12. Vías futuras

Cómo he comentado en otros apartados, me gustaría instaurar algún que otro sistema de feedback entre el entrenador y el usuario, como una caja de comentarios o incluso algún tipo de chat en directo, pero antes de realizar esa gran funcionalidad, es mejor comprobar la viabilidad de la aplicación una vez lanzada al mercado.

Ahora el siguiente paso sería comprar los dominios y servidores para alojar la web, y hacer un estudio de por donde proviene el tráfico, la edad, el género, el peso, los objetivos que buscan los usuarios, y tratar de explotar al máximo esa vía.

Lo más importante también es fidelizar los usuarios. No nos interesa que una vez se registran, no vuelvan a utilizar la web, porque eso desvirtúa un poco los datos.

13. Glosario

- **SEO:** Search Engine Optimization (Optimización para motores de búsqueda).
- **SCRUM:** Marco de gestión de proyectos.
- **PDO:** PHP Data Objects.
- **POO:** Programación orientada a objetos.
- **RF:** Requisito funcional.
- **RNF:** Requisito no funcional.

14. Bibliografía/Webgrafía

Referencias

Mozilla. (s.f.). CSS. Obtenido de <https://developer.mozilla.org/es/docs/Web/CSS>.

Mozilla. (s.f.). *HTML*. Obtenido de Mozilla: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics

Mozilla. (s.f.). *JavaScript*. Obtenido de <https://developer.mozilla.org/es/docs/conflicting/Web/JavaScript>.

Piattini, G. (2004). *cursos.clavijero.edu.mx*. Obtenido de https://cursos.clavijero.edu.mx/cursos/130_asio/modulo1/contenido/tema1.3.html

Wikipedia. (s.f.). *Git*. Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/Git>

Wikipedia. (s.f.). *GitHub*. Obtenido de Wikipedia: <https://es.wikipedia.org/wiki/GitHub>

Wikipedia. (s.f.). *Servidor HTTP Apache*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Servidor_HTTP_Apache

Documentación de Bootstrap:

[Get started with Bootstrap · Bootstrap v5.3 \(getbootstrap.com\)](https://getbootstrap.com/docs/5.3/getting-started/)

Ayudas varias en problemas concretos:

[Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/)

Herramienta para diagramas:

<https://www.lucidchart.com/pages/es>

15. Anexos

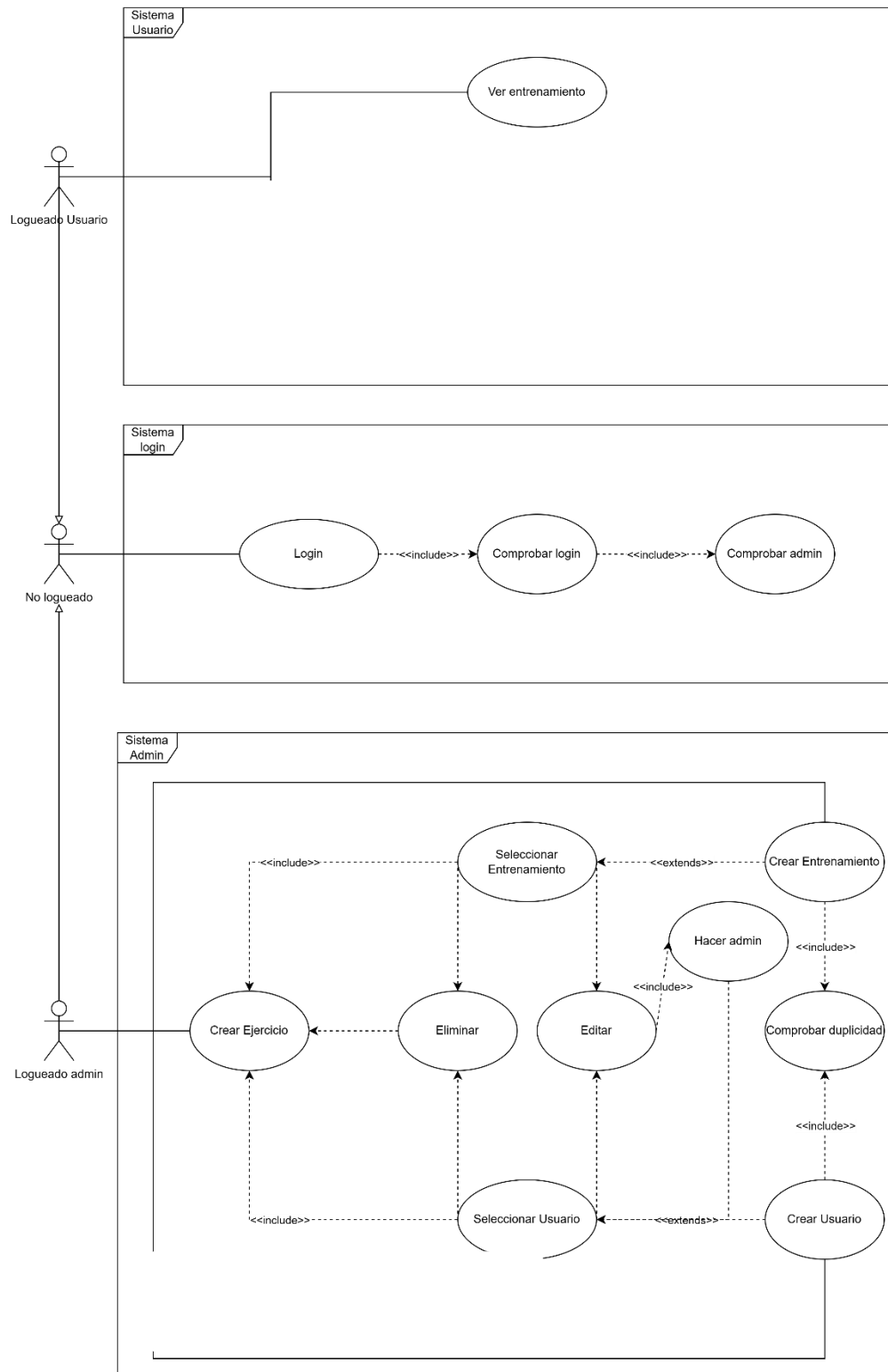


Ilustración 1-Diagrama de casos de uso

Ilustración 2-Diagrama entidad-relación

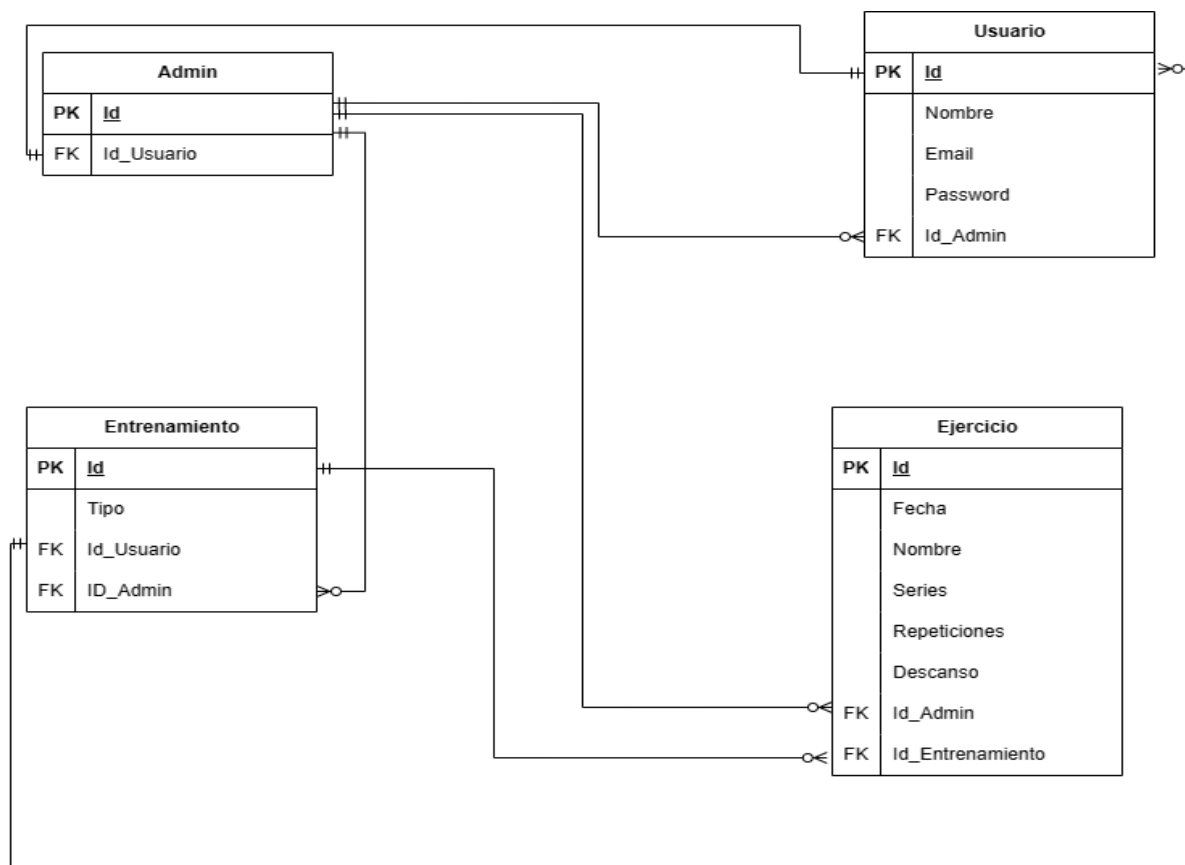
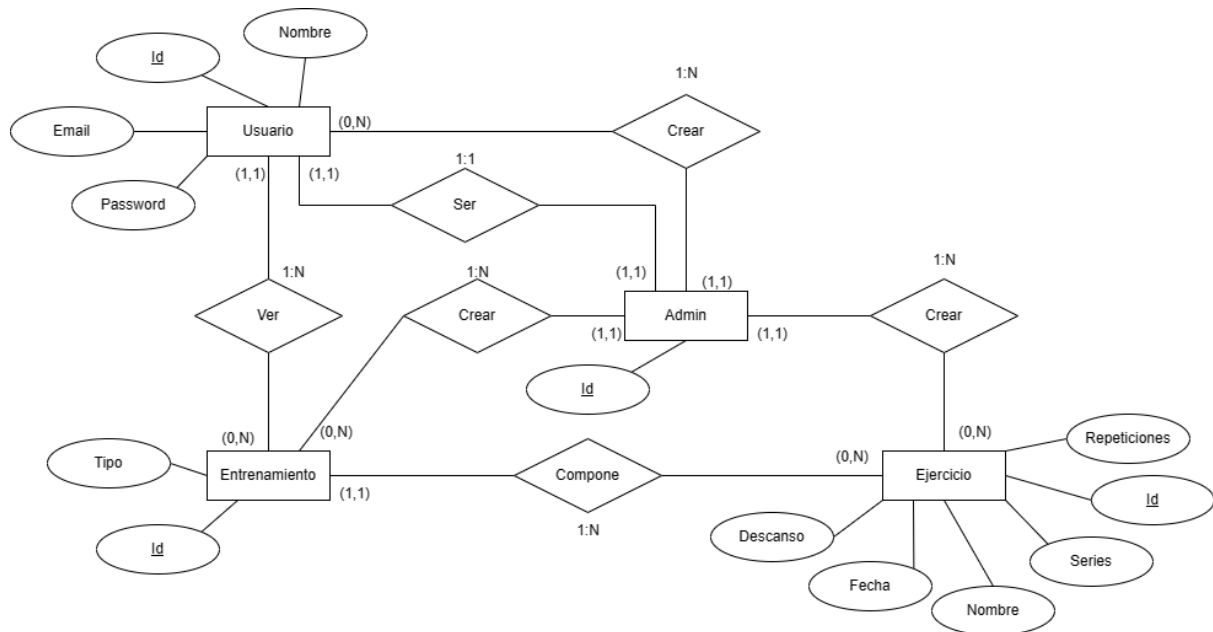


Ilustración 3-Diagrama modelo relacional

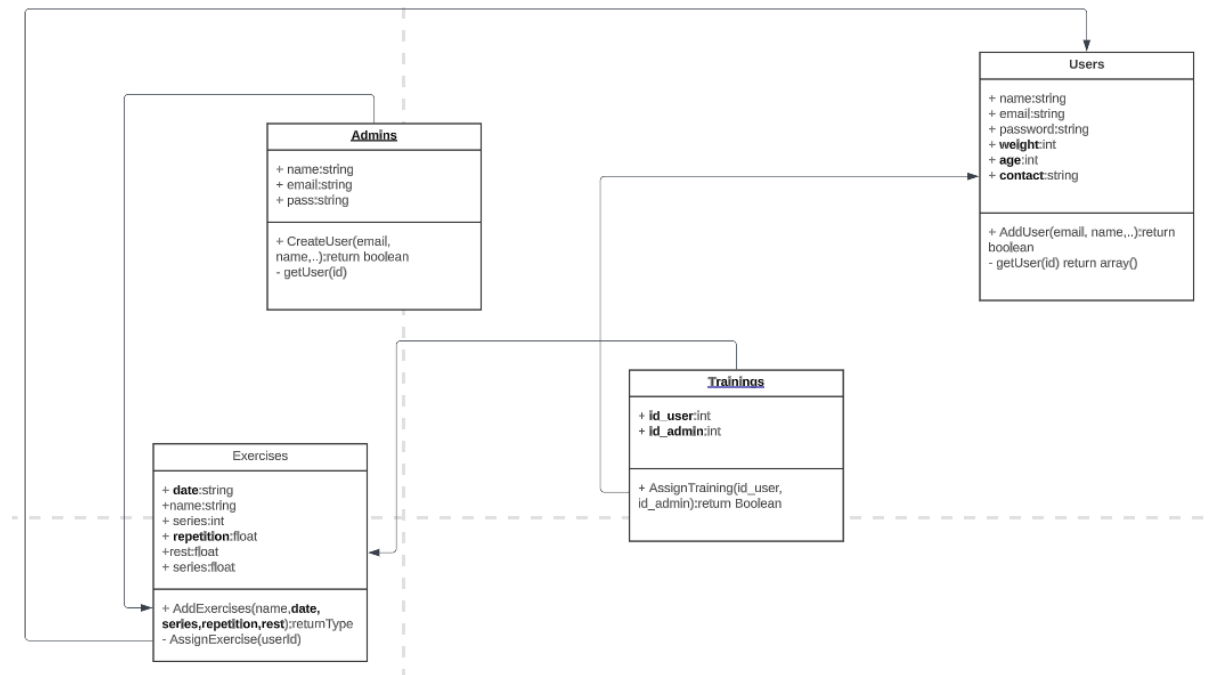


Ilustración 5-Diagrama de clases

