

Memoria proyecto: Gimnasio Online



C.I.F.P. “JUAN DE COLONIA”

Departamento de Informática

2º DAW

Autor Raúl García Arnaiz

Tutor Ana Rosa Álvarez

Fecha 19-05-2025

1. Contenido

Índice de figuras.....	4
Índice de tablas.....	5
2. Introducción.....	6
2.1. Descripción.....	6
2.2. Justificación.....	7
3. Planificación.....	7
3.1. Requisitos.....	7
3.1.1. Requisitos funcionales.....	7
3.1.2. Requisitos no funcionales.....	8
3.2. Recursos.....	9
3.2.1. Recursos hardware.....	9
3.2.2. Recursos software.....	9
3.3. Planificación temporal.....	10
3.4. Planificación económica.....	10
4. Tecnologías.....	11
5. Desarrollo y secuenciación temporal.....	14
5.1. Diseño.....	14
5.1.1. Diagrama de aplicación.....	14
5.1.2. Diagrama E/R (entidad/relación) y esquema de BBDD.....	16
5.1.3. Diseño de interface.....	17
5.2. Pruebas.....	21
5.2.1. Tipos de Pruebas Realizadas.....	21
5.2.2. Conclusión.....	23
6. Conclusiones finales.....	24
6.1. Grado de cumplimiento de los requisitos fijados.....	24
6.2. Propuestas de mejora o ampliaciones futuras.....	26
7. Guías.....	28
7.1. Guía para Usuarios.....	28
7.2. Guía para Administradores.....	29
8. Referencias.....	30

Índice de figuras

<u>Fig. Introducción-1-Pantalla inicial de la aplicación.....</u>	<u>6</u>
<u>Fig. Introducción-2-Pantalla inicial de la aplicación (móvil).....</u>	<u>6</u>

Índice de tablas

<u>Tabla Planificación-1: Planificación temporal del proyecto.....</u>	<u>10</u>
<u>Tabla Planificación-2- Presupuesto del proyecto.....</u>	<u>10</u>

2. Introducción

2.1. Descripción

Mi proyecto es una aplicación web en la que poder crear tus rutinas del gimnasio, guardarlas y consultarlas en cualquier momento, se proporcionaran el número de series, repeticiones y tiempo de descanso entre series, separando las rutinas en tres tipos avanzado, intermedio y principiante

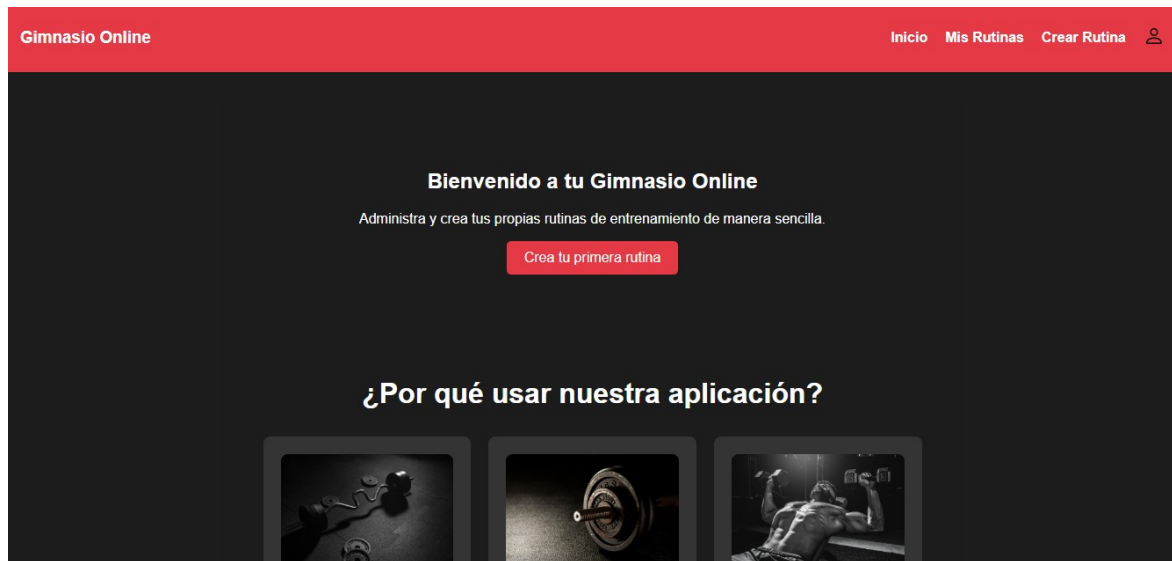


Fig. Introducción-1-Pantalla inicial de la aplicación

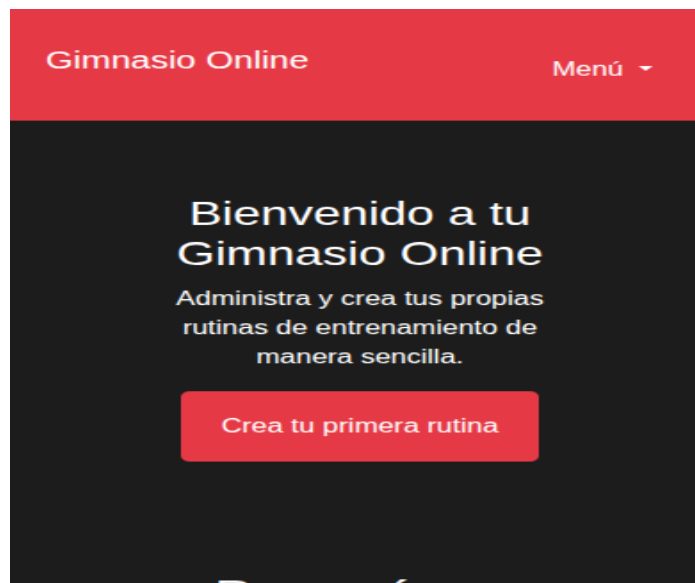


Fig. Introducción-2-Pantalla inicial de la aplicación (móvil)

2.2. Justificación

Porque unos de los problemas del gimnasio es no acordarse de las rutinas, de los ejercicios, sobre todo la gente nueva que está muy perdida, no sabe dónde hacer los ejercicios, cuanto hacer, cuanto descansar y como hacerlos, esta aplicación también va a servir un poco de guía

3. Planificación

3.1. Requisitos

3.1.1. Requisitos funcionales

- ✓ *Los requisitos funcionales de mi aplicación web son los siguientes:*
 1. Crear una cuenta: Para crear una cuenta el usuario tendrá que introducir su nombre junto con sus apellidos, su nombre de usuario, correo electrónico y una contraseña que tendrá que ser entre 8 y 16 caracteres
 2. Iniciar sesión: También se podrá iniciar sesión una vez el usuario se ha registrado, para iniciar sesión lo que se tendrá que hacer será introducir su correo electrónico y su contraseña.
 3. Cambio de contraseña: En el caso de que el usuario olvide la contraseña, podrá acceder en la parte de iniciar sesión, para cambiar la contraseña el usuario tendrá que introducir el correo electrónico y la nueva contraseña
 4. Creación de rutinas: El usuario una vez registrado podrá crear sus propias rutinas de ejercicios eligiendo los ejercicios que se le ofrecerán
 5. Consulta de rutinas: El usuario podrá consultar sus rutinas creadas en cualquier momento.
 6. Edición de las rutinas: El usuario podrá editar las rutinas, podrá cambiar ejercicios, añadir etc...

3.1.2. Requisitos no funcionales

1. Usabilidad

La aplicación ha sido diseñada para que sea fácil de entender y utilizar tanto por usuarios novatos como por administradores. La interfaz es intuitiva, con una navegación clara y formularios sencillos que guían al usuario paso a paso.

2. Accesibilidad

He procurado que los elementos visuales como botones, formularios y tablas sean legibles y estén bien contrastados, facilitando el uso en pantallas pequeñas o condiciones de baja visibilidad. Se ha utilizado un esquema de colores con buen contraste.

3. Rendimiento

La carga de ejercicios se hace mediante paginación, lo que mejora el rendimiento al evitar mostrar todos los datos a la vez. Además, el uso de peticiones asíncronas (fetch) permite que las acciones del usuario no requieran recargar toda la página.

4. Escalabilidad

La estructura de la base de datos y del código está pensada para que sea fácil añadir nuevas funciones en el futuro, como más niveles de dificultad, más tipos de usuarios, o una sección de seguimiento de progreso físico.

5. Compatibilidad

La aplicación funciona correctamente en los principales navegadores modernos (Chrome, Firefox, Edge). Además, el diseño es responsive, por lo que se adapta bien a diferentes tamaños de pantalla, incluyendo móviles.

6. Mantenibilidad

He organizado el código de manera modular, con funciones separadas para cada acción, y he utilizado nombres descriptivos en variables y funciones. Esto facilita futuras modificaciones o ampliaciones del sistema.

3.2. Recursos

3.2.1. Recursos hardware

Arquitectura: x86_64.

modo(s) de operación de las CPUs: 32-bit, 64-bit.

Address sizes: 39 bits physical, 48 bits virtual.

Orden de los bytes: Little Endian.

CPU(s): 8.

Lista de la(s) CPU(s) en línea: 0-7.

ID de fabricante: GenuineIntel.

Nombre del modelo: Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz.

Servidor en el que se desplegará: El servidor en el que desplegaré la aplicación web será infinityserver, ya que es bastante sencillo, te enseña gráficamente donde subir los archivos y lo más importante que es gratis.

3.2.2. Recursos software

Lista de recursos software

- ✓ WamppServer.
- ✓ Visual Studio.
- ✓ Chrome, firefox, edge.

3.3. Planificación temporal

Fases en que se divide y sus fechas.

Tabla Planificación-1: Planificación temporal del proyecto

Descripción de la tarea	N.º de horas
Diseño de aplicación	5
Frontend	20
Creación BD	2
Backend	25
Pruebas	5
Despliegue web	3
TOTAL HORAS	60

3.4. Planificación económica

El salario promedio anual de un desarrollador full-stack en España es de aproximadamente 32.500 euros. Esto equivale a un salario por hora de aproximadamente 16,25 euros (suponiendo una jornada laboral de 2.000 horas anuales).

Para 60 horas de trabajo:

- **Coste total de desarrollo:** 60 horas \times 16,25 euros/hora = 975 euros.
- **Coste despliegue:** 0 euros

Total estimado: 975 euros.

Tabla Planificación-2- Presupuesto del proyecto

Descripción de la tarea	Precio (€)
Coste de desarrollo	975
Coste despliegue	0
TOTAL (€)	975

4. Tecnologías

Tecnologías utilizadas:

- ✓ *HTML [2]: El HTML ha sido la base estructural de la aplicación web, permitiendo la visualización y organización de los distintos elementos de la interfaz. Su única función ha sido gráfica, proporcionando la estructura del sitio mediante etiquetas y elementos semánticos. Gracias a HTML, se han definido secciones como encabezados, párrafos, listas, formularios y botones*
- ✓ *CSS: El uso de CSS ha sido fundamental para la estilización de la aplicación, permitiendo la creación de una interfaz visual atractiva y funcional. Se ha utilizado para:*
 - o **Definir el diseño y la distribución de los elementos en la pantalla**, asegurando una disposición clara y ordenada.
 - o **Establecer combinaciones de colores**, alineándose con la identidad visual de la aplicación.
 - o **Gestionar la tipografía y los estilos de texto**, mejorando la legibilidad y estética.
 - o **Optimizar la apariencia en distintos dispositivos**, mediante técnicas de diseño responsive, adaptando la visualización a pantallas de diferentes tamaños.
- ✓ *MYSQL: El sistema de gestión de bases de datos MySQL ha sido utilizado para la creación, gestión y modificación de la base de datos de la aplicación. Se ha empleado para:*
 - o **Definir la estructura de la base de datos**, organizando la información en tablas optimizadas.
 - o **Gestionar la persistencia de los datos**, permitiendo el almacenamiento seguro de la información de los usuarios, rutinas, ejercicios y otros elementos clave del sistema.
 - o **Ejecutar consultas SQL eficientes**, para realizar operaciones como inserción, actualización, eliminación y búsqueda de datos.
 - o **Garantizar la integridad de los datos**, estableciendo relaciones entre tablas y aplicando restricciones para evitar inconsistencias.

- ✓ *PHP: El lenguaje PHP ha sido la pieza clave en la implementación del backend de la aplicación. Su uso se ha centrado principalmente en las capas de modelo y controlador dentro de la arquitectura MVC, proporcionando la lógica necesaria para gestionar las operaciones de la aplicación. Sus principales funciones han sido:*
 - o ***Procesar las solicitudes del usuario**, actuando como intermediario entre la base de datos y la interfaz de usuario.*
 - o ***Gestionar la conexión con MySQL**, utilizando PDO (PHP Data Objects) para establecer una conexión segura y minimizar riesgos de inyección SQL.*
 - o ***Ejecutar consultas para la creación y gestión de usuarios**, permitiendo el registro, autenticación y administración de cuentas.*
 - o ***Implementar la lógica del negocio**, como la validación de datos, el control de acceso y la manipulación de información enviada desde el frontend.*

- ✓ *JAVASCRIPT: El lenguaje JavaScript ha sido fundamental en la parte frontend, permitiendo dotar a la aplicación de dinamismo e interactividad. Su uso se ha centrado en:*
 - o ***Crear y gestionar eventos en la interfaz**, como clics en botones, envíos de formularios y cambios en la página.*
 - o ***Facilitar la comunicación con el backend**, mediante AJAX para el envío y recepción de datos sin necesidad de recargar la página.*
 - o ***Actualizar dinámicamente el contenido**, permitiendo la modificación de secciones en tiempo real sin afectar la navegación.*

- ✓ *JQUERY: Aunque jQuery forma parte de JavaScript, su uso en la aplicación ha sido lo suficientemente relevante como para destacarlo de manera individual. Se ha utilizado principalmente en:*
 - o ***Manejo de formularios**, optimizando su funcionamiento y validación.*
 - o ***Gestión de eventos**, facilitando la detección y respuesta a interacciones del usuario.*
 - o ***Mostrar y ocultar formularios dinámicamente**, mejorando la usabilidad en ciertos procesos, como la autenticación o la edición de información.*
 - o ***Simplificar el código JavaScript**, permitiendo manipular el DOM de manera más eficiente y reduciendo la complejidad del desarrollo.*

- ✓ WAMPP: El paquete de software WAMPP ha sido utilizado para el desarrollo y prueba de la aplicación en un entorno local antes de su despliegue en producción. Se ha empleado para:
 - o **Configurar un servidor local**, replicando las condiciones de un servidor en línea para simular su funcionamiento real.
 - o **Probar la conexión con la base de datos**, asegurando su correcto rendimiento y evitando fallos en la consulta de datos.
 - o **Ejecutar pruebas de funcionalidad**, permitiendo la depuración y corrección de errores antes de lanzar la aplicación.

- ✓ BOOTSTRAP: El framework Bootstrap ha sido incorporado en el desarrollo para mejorar el diseño y la responsividad de la aplicación. Sus principales aportaciones han sido:
 - o **Implementación de un diseño responsive**, permitiendo que la interfaz se adapte automáticamente a diferentes tamaños de pantalla.
 - o **Uso de componentes predefinidos**, como menús desplegables, botones estilizados, formularios y modales, optimizando el tiempo de desarrollo.
 - o **Facilidad en la personalización de estilos**, combinando CSS personalizado con las clases de Bootstrap para una mejor integración visual.

5. Desarrollo y secuenciación temporal

5.1. Diseño

5.1.1. Diagrama de aplicación

Mapa del sitio:

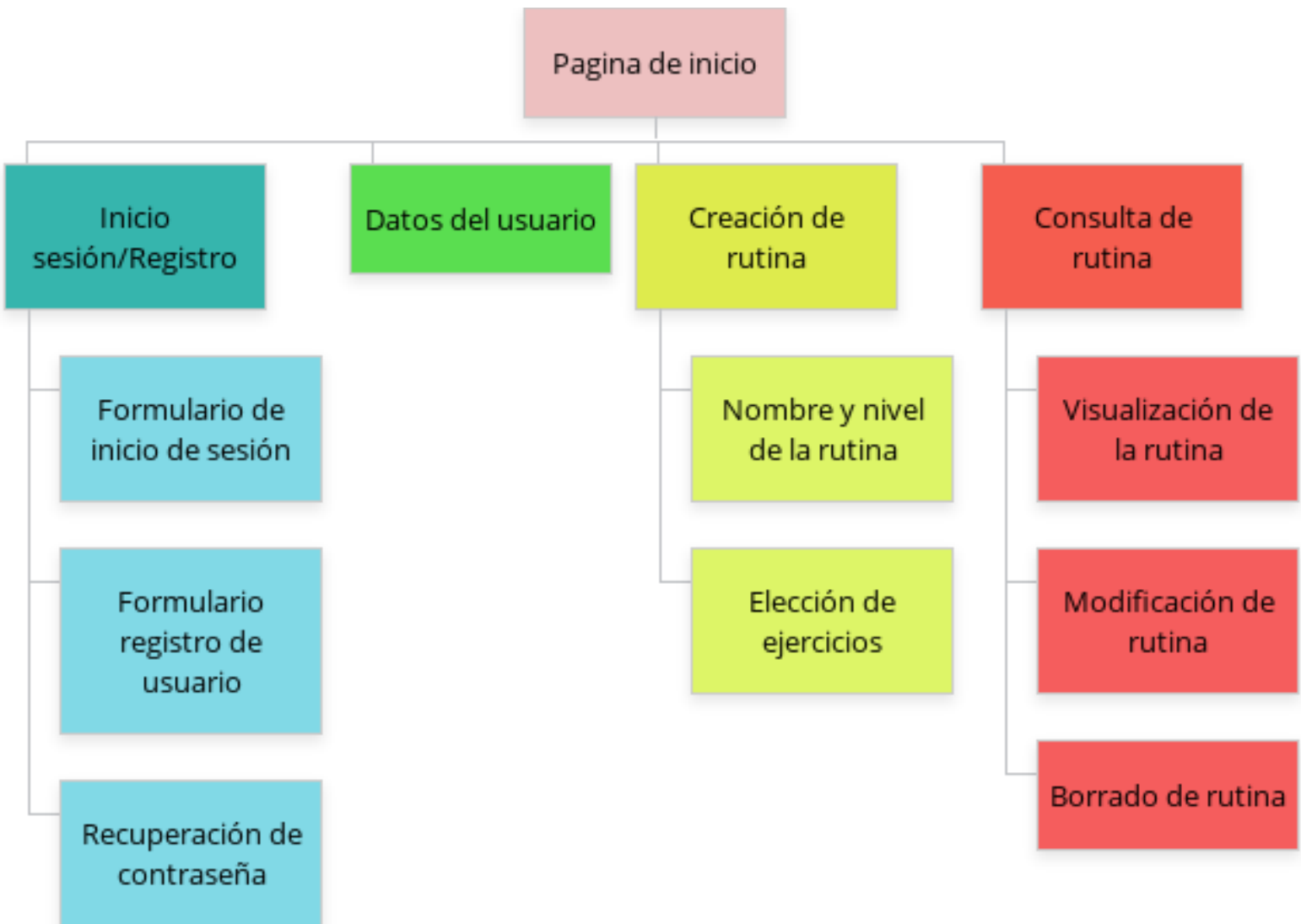


Fig. Desarrollo y secuenciación temporal-3-Diagrama de aplicación

Diagrama casos de uso

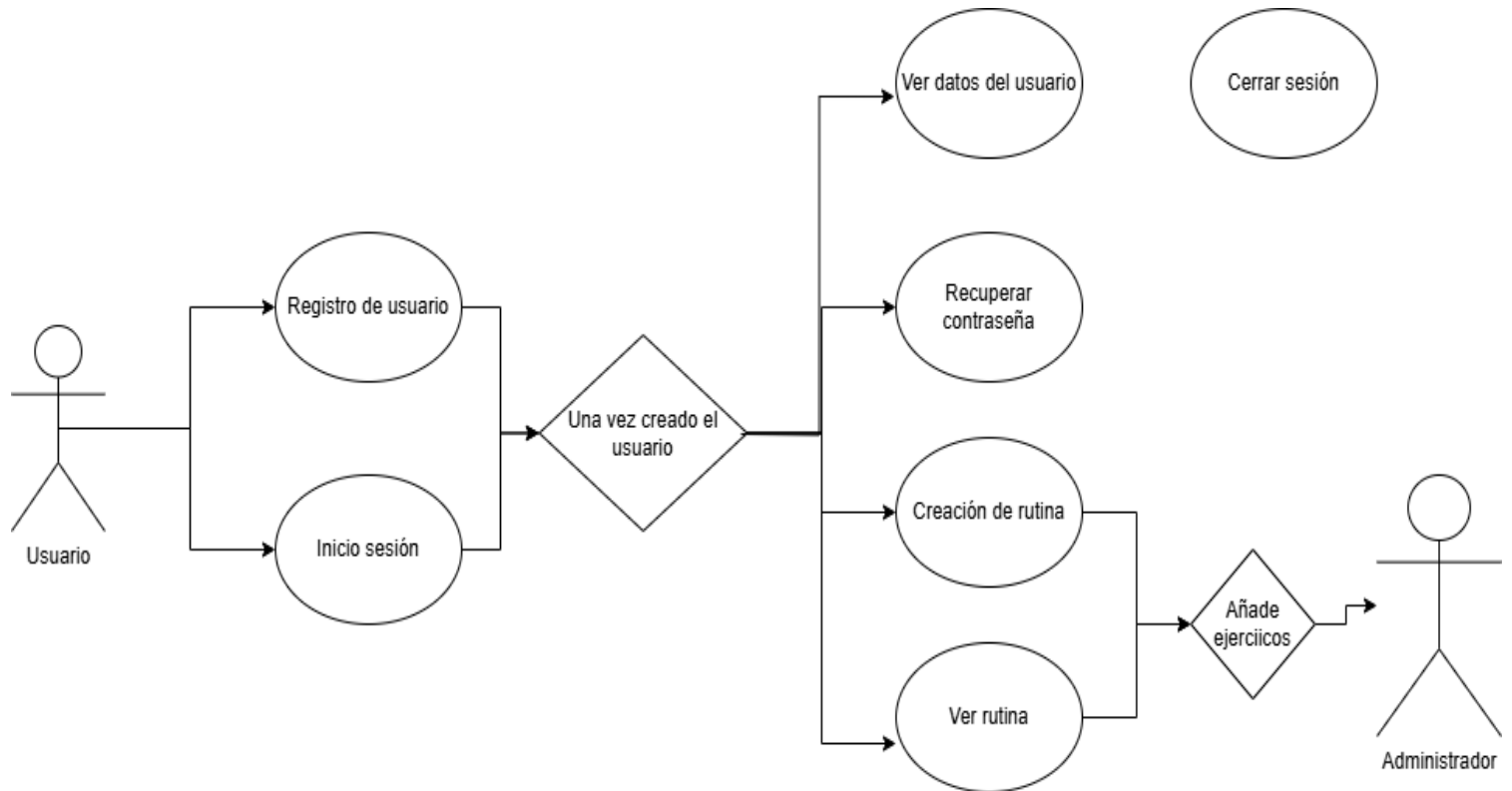


Fig. Desarrollo y secuenciación temporal-4-Diagrama casos de uso

5.1.2. Diagrama E/R (entidad/relación) y esquema de BBDD

Esquema BBDD:

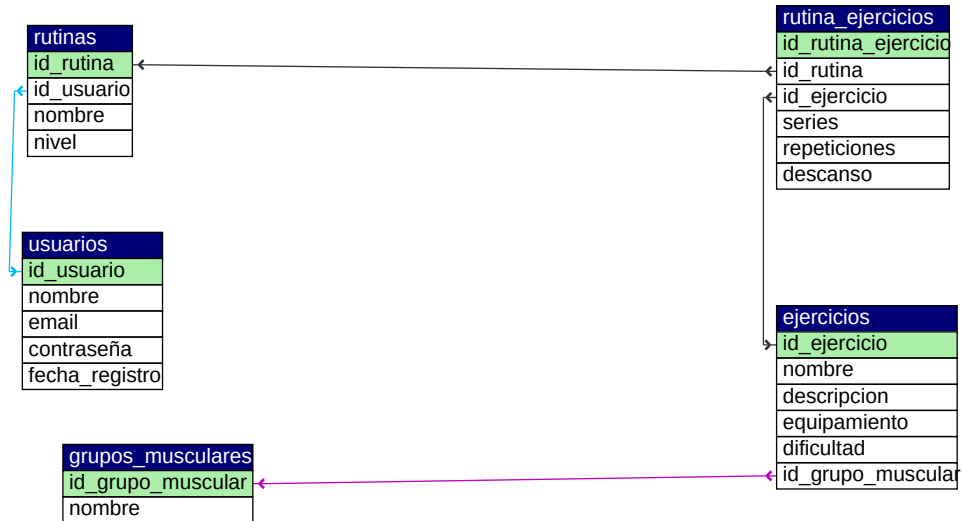


Fig. Desarrollo y secuenciación temporal-5-Diagrama E/R esquema BBDD

5.1.3. Diseño de interface

Página de inicio (index.html)

Objetivo: Introducir la aplicación al usuario, invitar al registro o inicio de sesión, y condicionar el acceso a la funcionalidad si no está autenticado.

Estructura:

- **Header:**
 - Nombre de la aplicación.
 - Menú de navegación con enlaces a "Iniciar sesión" y "Registrarse".
- **Cuerpo principal:**
 - Mensaje central del tipo: "Crea y gestiona tus rutinas personalizadas de entrenamiento".
 - Imagen ilustrativa o de fondo relacionada con el fitness.
 - Botón de llamada a la acción: "Comienza ahora" (redirecciona al registro o al login).
- **Condición:** Si el usuario ya está autenticado, se muestra un botón "Ir a mis rutinas" que lleva a la página de gestión.

Página de inicio de sesión (inicio_sesion.html)

Objetivo: Permitir que los usuarios accedan a su cuenta

Componentes:

- Formulario con campos:
 - Email
 - Contraseña
- Botón "Iniciar sesión".
- Enlace "¿Olvidaste tu contraseña?" que lleva a la página de recuperación.
- Mensajes de error en caso de datos incorrectos.
- Redirección a "Mis rutinas" en caso de inicio exitoso.

Objetivo: Registrar nuevos usuarios.

Componentes:

- Formulario con campos:
 - Nombre de usuario
 - Email
 - Contraseña
 - Confirmar contraseña
- Validación básica (contraseñas coinciden, email válido, etc.).
- Botón "Registrarse".
- Mensaje de éxito que sugiere iniciar sesión.

Objetivo: Cambiar la contraseña en caso de olvido, incluso si el sistema no envía correos.

Componentes:

- Campo de email (para confirmar la cuenta).
- Nueva contraseña
- Confirmación de contraseña
- Botón "Cambiar contraseña"
- Mensaje de confirmación o error.

Página para crear rutina (crear_rutina.html)

Objetivo: Permitir al usuario crear una rutina personalizada con ejercicios, niveles y datos asociados.

Componentes:

- Formulario inicial con:
 - Campo para el nombre de la rutina
 - Selección del nivel (novato, intermedio, avanzado)
- Tabla de ejercicios (con paginación si hay muchos registros), mostrando:
 - Nombre del ejercicio
 - Grupo muscular
 - Nivel de dificultad
 - Botón "Seleccionar"
- Al seleccionar un ejercicio:
 - Aparece en una tabla inferior con campos editables:
 - Número de series
 - Número de repeticiones
 - Tiempo de descanso
 - Botón para eliminar ejercicio
- Botón final "Guardar rutina" que envía toda la información.
- Condicional: si no está autenticado, se muestra un section con mensaje "Debes iniciar sesión o registrarte para crear una rutina" junto con botones de acceso.

Página de gestión de rutinas (rutinas.html)

Objetivo: Permitir al usuario ver, modificar y eliminar sus rutinas.

Componentes:

- Lista de rutinas en tarjetas o tabla:
 - Nombre de la rutina
 - Nivel
 - Número de ejercicios
- Botones por rutina:
 - "Editar" (abre un formulario precargado como en crear_rutina.html)
 - "Eliminar" (con confirmación)
- Botón adicional "Crear nueva rutina"

Objetivo: Permitir modificar una rutina ya existente.

Componentes:

- Mismo formulario que en crear_rutina.html, pero con los datos precargados.
- Posibilidad de:
 - Eliminar ejercicios
 - Cambiar series, repeticiones, descanso
 - Agregar nuevos ejercicios desde la tabla
- Botón "Guardar cambios"

Cabecera y navegación (incluida en todas las páginas)

Objetivo: Navegación clara, contextual según si el usuario está autenticado o no.

Componentes:

- Logotipo o nombre de la aplicación.
- Si el usuario no ha iniciado sesión:
 - Enlaces a "Inicio", "Iniciar sesión", "Registrarse"
- Si el usuario ha iniciado sesión:
 - Enlaces a "Mis rutinas", "Crear rutina", "Cerrar sesión"

Flujo general del usuario

1. Usuario entra a index.html.
2. Decide registrarse o iniciar sesión.
3. Una vez autenticado, puede:
 - Crear nuevas rutinas desde cero.
 - Ver y modificar sus rutinas existentes.
 - Eliminar rutinas no deseadas.

5.2. Pruebas

5.2.1. Tipos de Pruebas Realizadas

1. Pruebas funcionales

A lo largo del desarrollo del proyecto se han llevado a cabo distintas pruebas funcionales y de usabilidad para asegurar que todas las funcionalidades requeridas se comportan como se espera y que la experiencia del usuario sea fluida e intuitiva.

Pruebas funcionales realizadas:

- Registro de usuario: se ha verificado que se pueden registrar nuevos usuarios correctamente y que no se permiten duplicados de email.
- Inicio de sesión: se ha comprobado el acceso con credenciales correctas y la gestión de errores en caso de credenciales inválidas.
- Recuperación de contraseña: se ha probado el formulario de recuperación y su validación
- Creación de rutina: se ha probado que un usuario puede seleccionar ejercicios, introducir series, repeticiones y descanso, y guardar correctamente una rutina personalizada en la base de datos.
- Visualización de ejercicios: se ha validado que los ejercicios se cargan con paginación y se muestran correctamente en la tabla interactiva.
- Edición y eliminación de rutinas: se ha comprobado que las rutinas creadas se pueden modificar correctamente, así como eliminar si es necesario.
- Comunicación cliente-servidor: se ha probado el correcto funcionamiento del envío y recepción de datos mediante fetch y controladores PHP en el servidor remoto.
- Gestión de sesiones: se ha validado que el sistema gestiona la sesión del usuario correctamente tras iniciar y cerrar sesión.

2. Pruebas de usabilidad (usuarios no técnicos)

Se invitó a varios familiares y amigos a utilizar la aplicación sin instrucciones detalladas, observando su interacción:

- **Aspectos positivos:**
 - La navegación fue considerada intuitiva.
 - El diseño fue calificado como moderno y visualmente claro.
 - Los formularios fueron comprendidos sin necesidad de explicación.
- **Mejoras implementadas tras feedback:**
 - Se agregó un aviso cuando un ejercicio ya ha sido seleccionado.
 - Se mejoró el botón para eliminar ejercicios de la rutina.
 - Se ajustaron algunos colores para mejor contraste visual.

5.2.2. Conclusión

Las pruebas demostraron que la aplicación funciona correctamente en condiciones normales y se comporta de manera predecible ante entradas no válidas. La colaboración con usuarios reales fue clave para detectar problemas de experiencia de usuario y mejorar la interfaz.

6. Conclusiones finales

6.1. Grado de cumplimiento de los requisitos fijados

Base de datos con al menos 4 tablas relacionadas

El proyecto supera este requisito, ya que la base de datos contiene al menos 5 tablas interrelacionadas, las cuales son fundamentales para la gestión de rutinas de entrenamiento personalizadas. Las tablas más relevantes incluyen:

- usuarios: almacena los datos de los usuarios registrados (ID, nombre, email, contraseña).
- rutinas: contiene la información de cada rutina creada (ID, nombre, nivel, ID del usuario).
- ejercicios: incluye el catálogo de ejercicios disponibles (ID, nombre, grupo muscular, nivel).
- rutinas_ejercicios: tabla intermedia que relaciona rutinas con ejercicios, incluyendo series, repeticiones y descanso.
- grupos_musculares: guarda los grupos musculares.

Estas tablas están relacionadas mediante claves foráneas que aseguran la integridad referencial y permiten operaciones como consultar qué rutinas pertenecen a un usuario, qué ejercicios contiene cada rutina y más.

Despliegue en un servidor

El proyecto ha sido desplegado exitosamente en un servidor gratuito de hosting mediante InfinityFree. Esto implica:

- La aplicación está accesible públicamente mediante una URL del hosting.
- Los archivos .php, .html, .js y .css han sido subidos al servidor mediante el panel de control o gestor de archivos.

Este despliegue permite que cualquier usuario pueda registrarse, iniciar sesión y utilizar la aplicación desde cualquier navegador conectado a Internet.

Diseño responsive

La aplicación ha sido desarrollada con un enfoque responsive, lo que garantiza una correcta visualización y funcionalidad en distintos tamaños de pantalla (PC, tablets y móviles). Esto se logra mediante:

- Uso de Bootstrap, un framework CSS que incluye una amplia variedad de componentes y clases responsivas
- Aplicación de reglas CSS adaptativas.
- Interfaz que se adapta automáticamente al tamaño de pantalla, ocultando o reorganizando elementos para mejorar la experiencia de usuario.

Modelo Vista Controlador (MVC)

El proyecto ha sido estructurado respetando el patrón Modelo Vista Controlador, que permite una separación clara de responsabilidades entre los distintos componentes de la aplicación.

- El modelo está representado por clases PHP como usuarios_class.php, rutinas_class.php, entre otras, que se encargan de acceder a la base de datos y gestionar la lógica de negocio.
- Los controladores están compuestos por archivos como controlador_inicio_sesion.php, controlador_crear_rutina.php o controlador_actualizar_rutina.php, que reciben las solicitudes del usuario (por ejemplo, desde fetch() en JavaScript), procesan la información, consultan el modelo y devuelven la respuesta adecuada.
- Las vistas están representadas por las páginas HTML como index.html o crear_rutina.html, que presentan la información al usuario. Estas vistas se complementan con JavaScript para hacer la aplicación dinámica y reactiva mediante llamadas asincrónicas a los controladores.

Tecnologías no vistas en clase

Se ha incorporado la utilización de tecnologías no vistas en clase, lo cual aporta valor al desarrollo y demuestra iniciativa por parte del desarrollador.

- jQuery: aunque opcional, se puede haber usado inicialmente o de forma complementaria para manejo del DOM o AJAX

6.2. Propuestas de mejora o ampliaciones futuras

Durante el desarrollo de la aplicación web de gimnasio, he identificado varias áreas en las que podría mejorarse la experiencia de usuario y ampliarse la funcionalidad general del sistema. A continuación, voy a exponerlas:

1. Sistema de progreso y seguimiento

Una posible ampliación sería permitir que los usuarios registren su progreso (por ejemplo, peso levantado, repeticiones realizadas o duración del entrenamiento). Estos datos podrían visualizarse en gráficas para fomentar la motivación y el seguimiento de resultados a lo largo del tiempo.

2. Rutinas sugeridas por IA

Se podría implementar un sistema de recomendaciones inteligentes que sugiera rutinas en base al historial del usuario, sus objetivos y su nivel. Esto aumentaría la personalización y el valor percibido de la aplicación.

3. Mejoras en la gestión de rutinas

Actualmente, los usuarios pueden crear y modificar rutinas, pero una mejora sería permitirles:

- Clonar rutinas anteriores para editarlas rápidamente.
- Compartir rutinas con otros usuarios mediante un código o enlace.
- Marcar rutinas como favoritas.

4. Modo claro

La interfaz ya tiene un esquema de colores oscuros, pero ofrecer al usuario un selector de tema claro/oscuro permitiría mayor accesibilidad y comodidad visual.

5. Paginación y filtrado más avanzado

Ya se utiliza paginación para mostrar los ejercicios, pero podría complementarse con filtros dinámicos (por grupo muscular, nivel, tipo de ejercicio) y una búsqueda instantánea para mejorar la usabilidad.

6. Mejoras cambio de contraseña

Ya se utiliza un cambio de contraseña, pero no es seguro, si alguien sabe tu correo puede cambiar la contraseña y entrar, la mejora que se implementaría sería el envío de un correo al correo introducido.

7. Inserccion de videos explicativos sobre ejercicios

Aunque los ejercicios tienen descripciones siempre es mejor ver algo en imágenes o videos, se añadirán videos explicativos cortos para que el usuario entienda como se realiza el ejercicios con mayor precisión.

7. Guías

7.1. Guía para Usuarios

El usuario es cualquier persona registrada que desea acceder a rutinas de entrenamiento, crear las suyas propias y consultar ejercicios. El uso general es el siguiente:

1. Registro e inicio de sesión

- El usuario accede a la página de registro para crear su cuenta introduciendo su nombre, email y contraseña.
- Una vez registrado, podrá iniciar sesión con su email y contraseña.

2. Visualización de ejercicios

- Desde la sección correspondiente, el usuario puede ver una lista de ejercicios divididos por páginas.
- Puede consultar información sobre cada ejercicio, incluyendo grupo muscular y nivel de dificultad.

3. Creación de rutinas personalizadas

- En la página “Crear rutina”, el usuario puede:
 - Seleccionar ejercicios de una tabla paginada.
 - Introducir el número de series, repeticiones y tiempo de descanso.
 - Darle un nombre y nivel a la rutina creada.
- Al enviar el formulario, la rutina se guarda asociada a su cuenta.

4. Gestión de rutinas

- Puede ver todas las rutinas que ha creado.
- Tiene la opción de modificar una rutina (cambiar ejercicios o parámetros) o eliminarla.

5. Cierre de sesión

- Desde el menú, puede cerrar sesión en cualquier momento.

7.2. Guía para Administradores

El administrador tiene acceso a funciones adicionales para gestionar los contenidos de la plataforma. En esta versión inicial, sus funciones principales incluyen:

1. Gestión de ejercicios

- Puede añadir nuevos ejercicios a la base de datos especificando su nombre, descripción, grupo muscular y nivel.
- También puede editar o eliminar ejercicios existentes si hay errores o actualizaciones necesarias.

2. Supervisión de usuarios y rutinas

- Puede consultar las rutinas creadas por los usuarios, lo que permite detectar contenido inapropiado o duplicado.
- Se pueden implementar futuras funciones para desactivar cuentas o eliminar rutinas si se considera necesario.

3. Mantenimiento general

- El administrador puede encargarse del mantenimiento de la base de datos, comprobando que las relaciones entre usuarios, rutinas y ejercicios estén bien estructuradas.

8. Referencias

- 1: <https://pixabay.com/es/>
- 2: <https://www.gloomaps.com/>
- 3: <https://app.diagrams.net/>
- 4: <https://php-myadmin.net>
- 5: <https://ideogram.ai/>