Imagen que contiene texto, dibujo, señal

Descripción generada automáticamente

**SmartFit: sistema integral de servicios y entrenamiento personalizado**

SMART FIT

Práctica Profesional III

Alumnos: Farias Ramiro,  
 Fernández Bryan,  
 Lino Milagros,

Morales Ignacio

Descripción breve del proyecto

2024

Contenido

[1.](#_gjdgxs) Estado del arte 2

[2.](#_30j0zll) Descripción del proyecto 3

[3.](#_1fob9te) Objetivo 3

[4.](#_3znysh7) Alcance 3

[5.](#_2et92p0) Casos de uso 4

[6.](#_tyjcwt) Especificación de requerimientos 6

[7.](#_3dy6vkm) Recursos disponibles 13

[a.](#_1t3h5sf) Personal 13

[b.](#_4d34og8) Material 14

[c.](#_2s8eyo1) Tiempo 14

[8.](#_17dp8vu) Plan de trabajo 14

[9.](#_3rdcrjn) Gestión de riesgos 15

[10.](#_26in1rg) Diseño del sistema 18

[a.](#_lnxbz9) Arquitectura 18

[b.](#_35nkun2) Diagrama de componentes 19

[c.](#_1ksv4uv) DER 19

[d.](#_44sinio) Diagrama de actividades 20

e[.](#_z337ya) Diagrama de [clases](#_44sinio)21

f[.](#_3j2qqm3) Interfaces de usuarios 22

11. Desarrollo 25

12. Implementación y Prueba 27

13. Conclusiones finales 29

# Estado del arte

Los avances en tecnología han ido redefiniendo la forma en que abordamos el deporte y la salud. Hoy día, existen aplicaciones para hacer deporte que nos ayudan a monitorizar resultados y lograr el éxito.

Cada vez más deportistas y aficionados al deporte y a la vida saludable optan por estar acompañados de aplicaciones que controlen el entrenamiento y de esta forma alcanzar de una forma más eficaz, las metas. A su vez, éstas destacan en cuanto al aspecto motivacional que tiene en los usuarios que las utilizan.

En la actualidad, existen múltiples aplicaciones relacionadas al deporte. Algunas que podemos destacar son:

**Adidas Running App**: es una plataforma diseñada para corredores que ofrece seguimiento de rutas utilizando el GPS integrado en los dispositivos móviles, proporciona entrenamientos guiados adaptados a corredores de todos los niveles. Los usuarios pueden unirse a una comunidad en línea, participar en desafíos y compartir sus logros con otros corredores.

**Entrenamiento en casa**: **Fitness**: es una aplicación integral que ofrece planes de entrenamiento personalizados para lograr objetivos de fitness específicos, ya sea desarrollar músculo, quemar grasa o mantenerse en forma. Con más de 100 entrenamientos guiados, seguimiento de datos, y la posibilidad de entrenar en casa sin equipamiento, es la herramienta perfecta para usuarios de todos los niveles.

## Tecnologías

En la creación de esta aplicación web, se decidió la utilización de diversas tecnologías que pueden ofrecer un desarrollo simple, objetivo y escalable.

* **Python**: lenguaje de programación flexible y fácil de aprender, ampliamente utilizado en el desarrollo web debido a su versatilidad y amplio ecosistema de herramientas
* **Flask**: framework ligero y minimalista para Python que facilita la creación de aplicaciones web simples y elegantes, proporcionando las herramientas necesarias sin imponer una estructura rígida.
* **MySQL**: sistema de gestión de bases de datos relacional confiable y eficiente, ideal para almacenar y gestionar datos en aplicaciones web, especialmente aquellas que requieren manejar grandes volúmenes de información.
* **Frontend**: la interfaz de usuario será desarrollada utilizando tecnologías como HTML, CSS y JavaScript, junto con frameworks y bibliotecas modernas para garantizar una experiencia visualmente atractiva e interactiva para los usuarios.

# Descripción del proyecto

**SmartFit** es una aplicación web diseñada para combinar lo mejor del mundo del entrenamiento y la salud con la conveniencia de la tecnología moderna. Esta aplicación ofrece una solución completa e integral para aquellos que desean mejorar su bienestar físico, proporcionando herramientas de entrenamiento personalizado, seguimiento del progreso y, por otro lado, un sistema especializado para acercar de manera efectiva al usuario con profesionales de la salud y el fitness en tiempo real.

Con un desarrollo adecuado, esta aplicación puede ser una herramienta invaluable para alcanzar los objetivos deportivos, mientras que se encarga de ofrecer una experiencia más conveniente y ágil para acceder a servicios de salud y deporte.

# Objetivo

El objetivo principal de la aplicación web es **brindar una solución integral para el entrenamiento personal**, permitiendo a los usuarios **alcanzar sus metas de fitness de manera efectiva, personalizada y motivante.**

Esta aplicación resuelve las siguientes cuestiones:

* Dificultad para monitorear el progreso
* Inaccesibilidad a entrenadores personales

El programa tiene como objetivo eliminar las barreras comunes para el éxito en el fitness y brindar a los usuarios las herramientas y el apoyo que necesitan para alcanzar sus objetivos de manera efectiva y agradable.

# Alcance

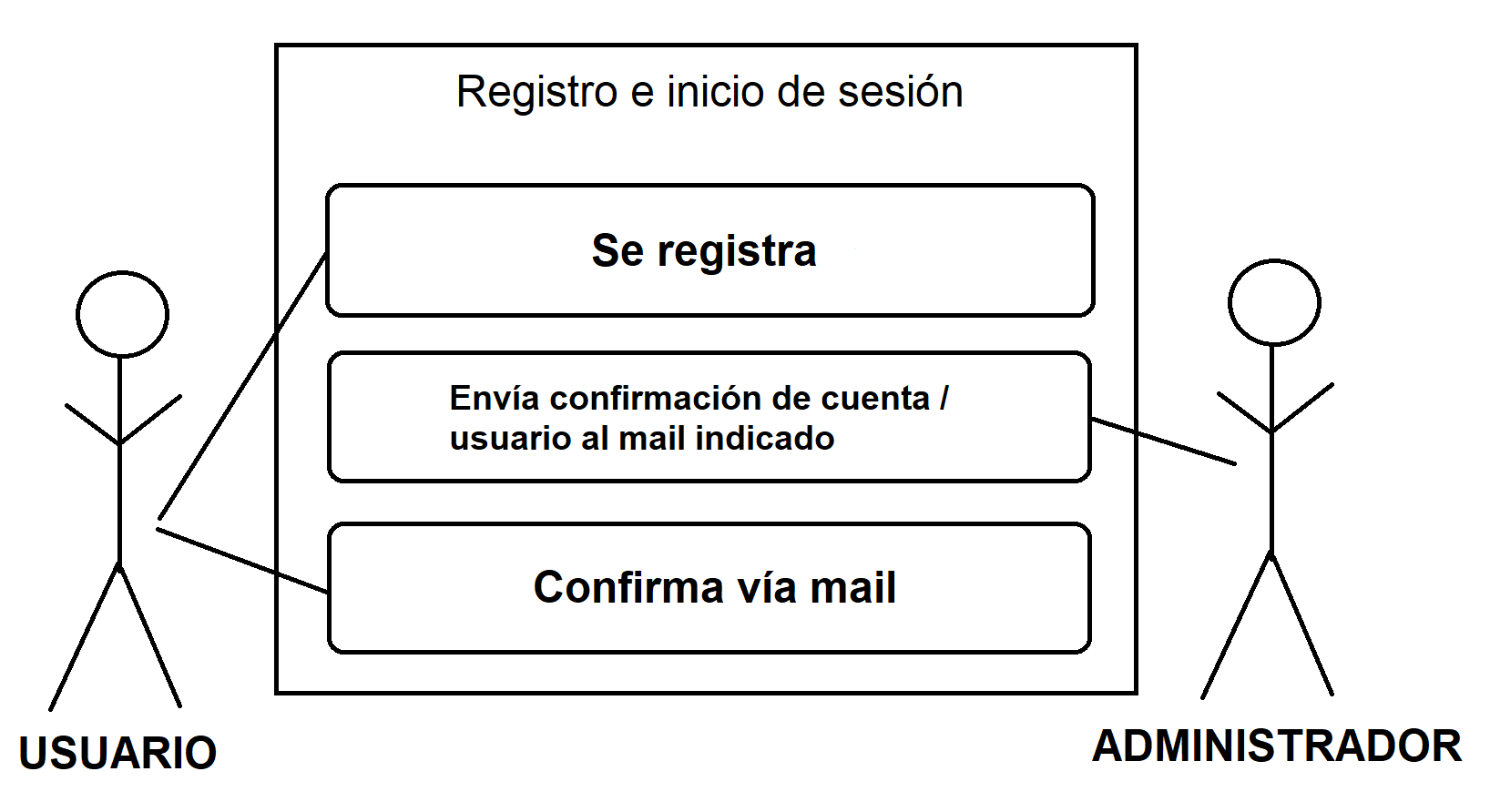
* **Entrenamiento Personalizado**:
  + En este módulo se desarrolla la evaluación inicial para determinar condición física, objetivos y preferencias.
  + También se contempla la creación de planes de entrenamiento personalizados con variedad de ejercicios.
* **Seguimiento del progreso**:
  + Este módulo se encarga del registro de los datos y del progreso.
  + Calorías quemadas e historial detallado de entrenamientos previos.
* **Consultoría personalizada**
  + La página web permite la interacción directa con un entrenador personal, nutricionista, deportólogo, kinesiólogo para necesidades o condiciones específicas.

***Funciones No incluidas:***

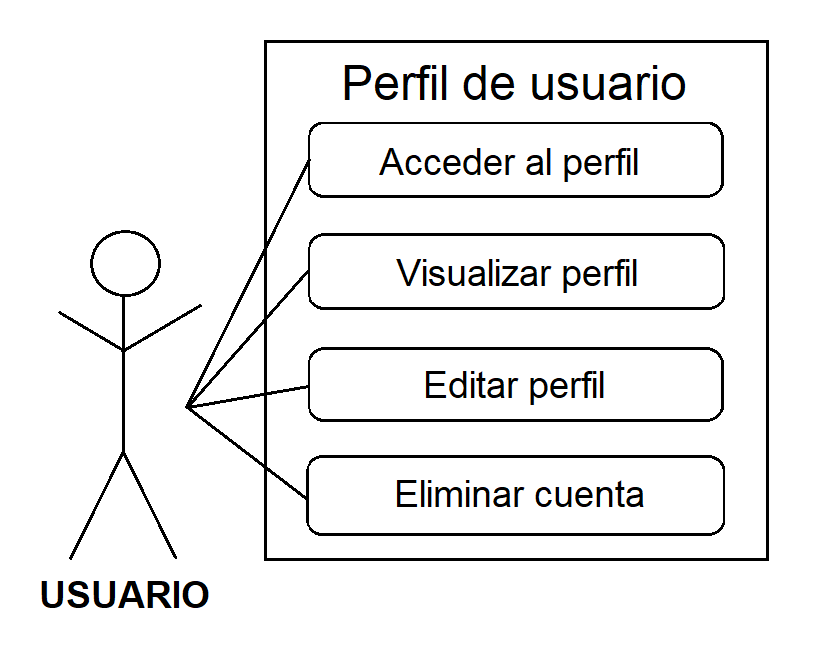
* **Creación de planes nutricionales**: La app se centra en el entrenamiento físico y no ofrece asesoramiento nutricional personalizado.
* **Funciones de diagnóstico médico**: La app no está diseñada para diagnosticar o tratar condiciones médicas.
* **Integración con otras apps de salud**: La app no se integra directamente con otras apps de salud o fitness que el usuario pueda estar utilizando.

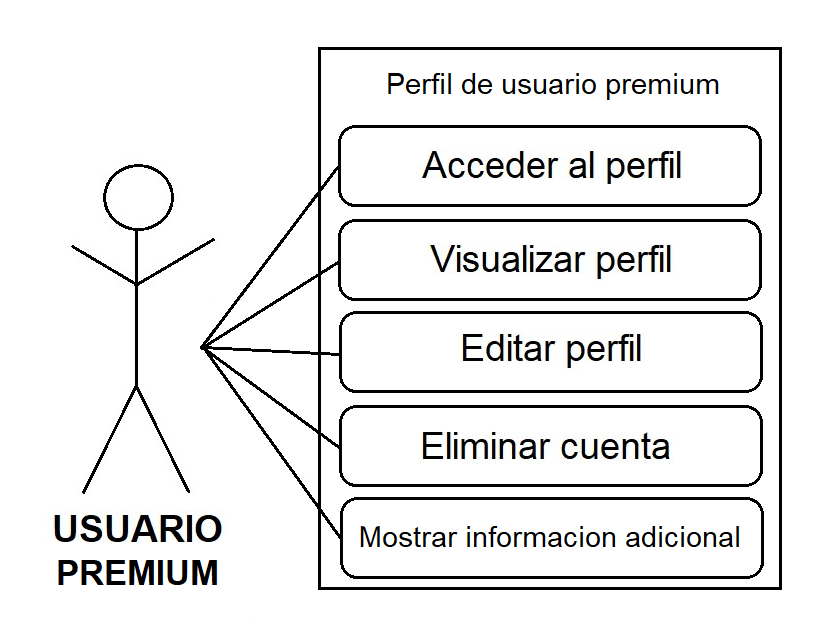
# Casos de uso

#### Caso de Uso para el registro e inicio de sesión

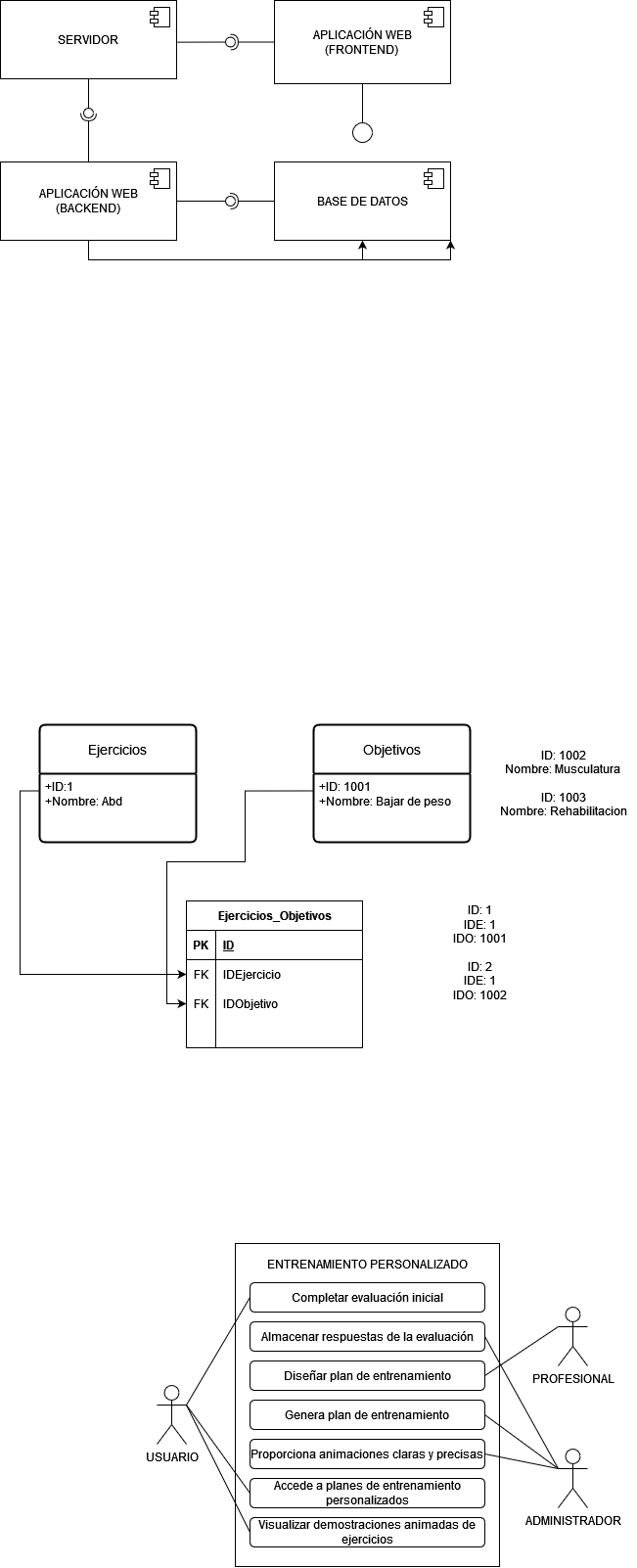


#### Casos de uso para la configuración de Perfil de usuario normal y premium

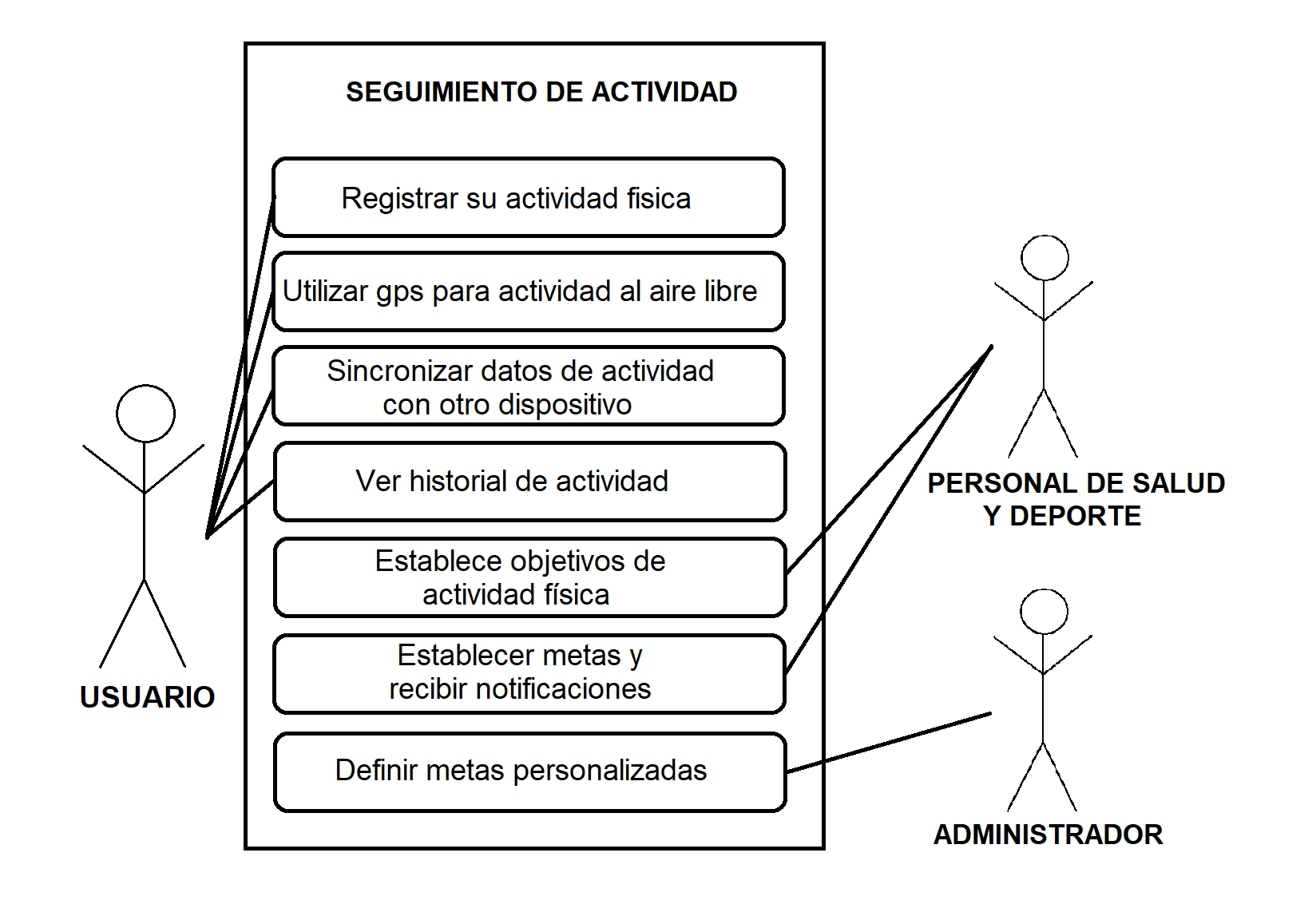




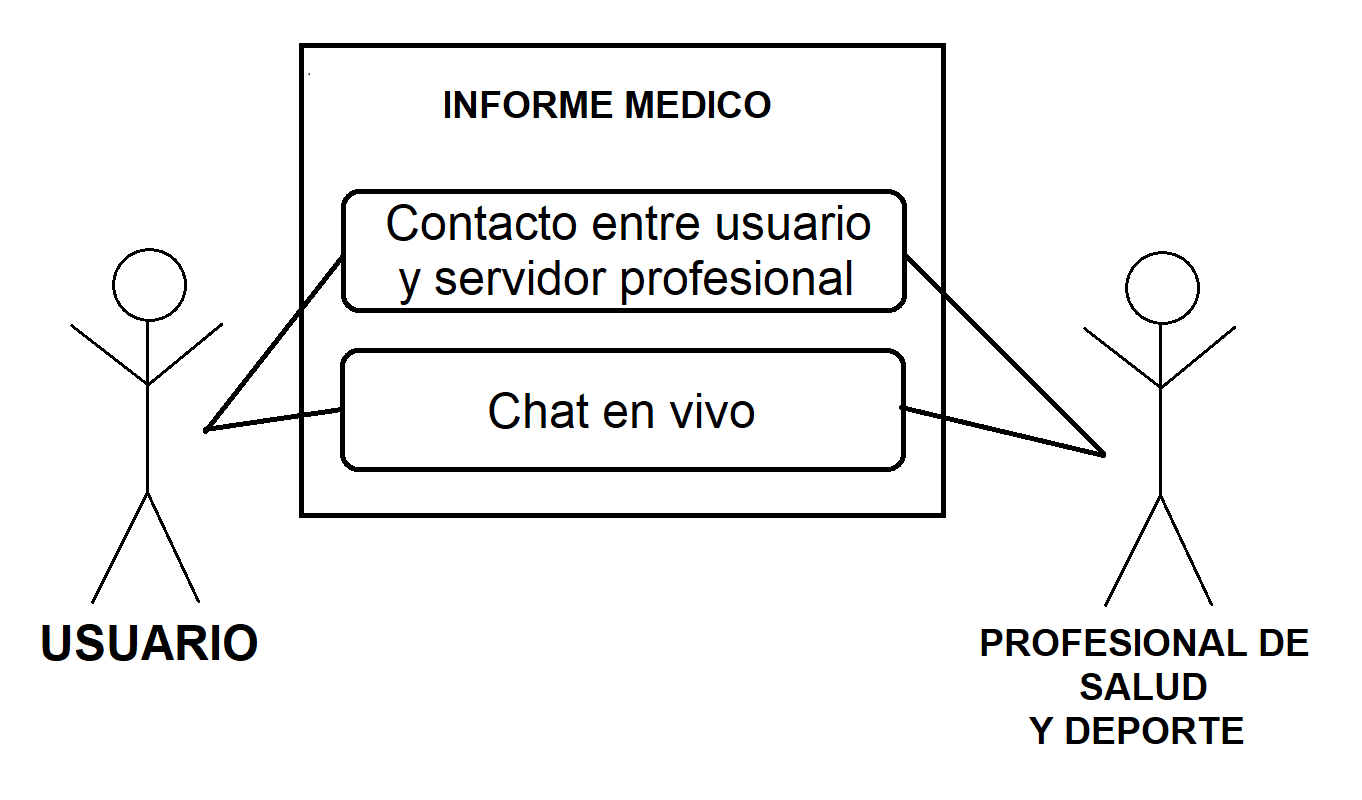
#### Caso de uso para funcionalidad de entrenamiento personalizado



#### Casos de uso para la funcionalidad de seguimiento de actividad



#### Casos de uso para el Informe profesional privado



# Especificación de requerimientos

## Para el módulo 1: Registro e Inicio de Sesión

**1.1 Creación de Cuenta SmartFit (Registro)**

**RF1.1.1:** El usuario debe poder acceder a la página de registro desde la página principal de la aplicación o sitio web.

**RF1.1.2:** El usuario completa con sus datos el formulario de registro.

**RF1.1.3:** El administrador debe enviar un correo electrónico de confirmación al usuario después de registrarse.

**RF1.1.4:** El usuario debe poder activar su cuenta haciendo clic en el enlace provisto en el correo electrónico de confirmación.

**1.2 Inicio de Sesión en SmartFit**

**RF1.2.1:** El usuario debe poder acceder a la página de inicio de sesión desde cualquier página de la aplicación o sitio web.

**RF1.2.2:** El formulario de inicio de sesión debe solicitar la dirección de correo electrónico y la contraseña del usuario.

**RF1.2.3:** El administrador debe validar las credenciales ingresadas por el usuario mediante la base de datos de usuarios en alta.

## Para el módulo 2: Perfiles de Usuario

**2.1 Perfil de Usuario Común**

**RF2.1.1:** El usuario debe poder acceder a su perfil desde cualquier página de la aplicación o sitio web.

**RF2.1.2:** El perfil del usuario debe mostrar su información.

**RF2.1.3:** El usuario debe poder editar su información de perfil.

**RF2.1.4:** El usuario debe poder eliminar su cuenta.

**2.2 Perfil de Usuario Premium**

**RF2.2.1:** El usuario premium debe tener acceso a las mismas funcionalidades que el usuario común.

**RF2.2.2:** El perfil del usuario premium debe mostrar la funcionalidad de Chat con el profesional.

## Para el módulo 3: Entrenamientos personalizados

**3.1. Entrenamientos Personalizados**

**RF3.1.1:** El usuario completa un formulario para determinar condición física, objetivos y así la evaluación del entrenamiento personalizado más adecuado, ajustando la duración y repeticiones de entrenamientos.

**RF3.1.2:** El administrador genera un plan de entrenamiento utilizando datos y objetivos, incluyendo variedad de ejercicios.

**RF3.1.3:** El usuario debe poder acceder a planes de entrenamiento personalizados según sus objetivos y condición física para usuario común y premium.

**RF3.1.4:** El administrador proporciona animaciones claras y precisas para cada ejercicio en la página web.

**RF3.1.5:** El usuario visualiza demostraciones animadas de cada ejercicio para asegurar una correcta ejecución.

**3.2. Seguimiento de Actividad**

**RF3.2.1:** La aplicación debe permitir al usuario registrar su actividad física, como entrenamientos realizados previamente y las calorías totales quemadas.

## Para el módulo 4: Servicio de intermediación entre usuario premium y profesional médico

**RF4.1. Chat en vivo:** El usuario premium debe poder chatear en vivo con un profesional de deporte o de salud.

**RF4.2. Acercamiento con profesional:** El usuario premium debe poder contactar con servicios relacionados de los profesionales de salud de nuestra base de datos, según el tipo de necesidad que tenga.

## Requerimientos No Funcionales

**1. Rendimiento**

· **Tiempo de respuesta:**

o El sistema debe responder a las solicitudes del usuario en menos de 3 segundos en condiciones normales de red.

o El tiempo de carga de las pantallas principales no debe superar los 5 segundos.

· **Uso de recursos:**

o La memoria del sistema no debe exceder el 50 % de la memoria RAM disponible en el dispositivo.

o El sistema debe optimizar el uso de batería para permitir sesiones de entrenamiento prolongadas sin un consumo excesivo.

· **Carga de datos:**

o El sistema debe procesar y almacenar datos de entrenamiento en tiempo real sin afectar el rendimiento general.

o La carga de historiales de entrenamiento o planes personalizados debe completarse en un máximo de 15 segundos.

· **Pruebas de rendimiento bajo carga:**

o El sistema debe someterse a pruebas de rendimiento bajo carga para evaluar su capacidad para manejar un gran número de usuarios simultáneos. Las pruebas deben identificar cuellos de botella y áreas de mejora para garantizar un rendimiento escalable.

· **Monitoreo del rendimiento en tiempo real:**

o El sistema debe implementar un sistema de monitoreo del rendimiento en tiempo real que recopile métricas clave como el tiempo de respuesta, el uso de recursos y la tasa de errores para identificar y resolver problemas de rendimiento de manera proactiva.

**2. Escalabilidad**

· **Capacidad de usuarios:**

o El sistema debe soportar un número creciente de usuarios simultáneos a medida que SmartFit crece. Se debe planificar una arquitectura escalable que pueda adaptarse a la demanda.

· **Volumen de datos:**

o El sistema debe ser capaz de manejar un volumen cada vez mayor de datos de usuarios, incluyendo historiales de entrenamiento, registros de actividad y objetivos personales.

o Se debe implementar una base de datos escalable con estrategias de almacenamiento eficientes para garantizar un acceso rápido y seguro.

**3. Seguridad**

· **Autenticación y autorización:**

o El sistema debe utilizar un mecanismo de autenticación seguro para verificar la identidad de los usuarios, como el inicio de sesión con correo electrónico y contraseña o integración con redes sociales.

o El sistema debe implementar control de acceso basado en roles para restringir el acceso a las funcionalidades según el tipo de usuario (gratuito, premium, etc.).

· **Protección de datos:**

o La información personal de los usuarios, como historiales de entrenamiento, objetivos y datos de salud, debe estar cifrada tanto en el sistema como durante la transmisión al servidor. Se debe utilizar protocolos de cifrado robustos como TLS/SSL.

o SmartFit debe cumplir con las leyes y regulaciones de privacidad de datos vigentes en su región.

· **Prevención de ataques:**

o El sistema debe ser resistente a ataques cibernéticos comunes, como ataques de inyección de código, ataques de denegación de servicio y ataques de phishing. Se debe utilizar herramientas de análisis de malware y actualizaciones de seguridad regulares para mantener la protección contra las últimas amenazas.

o Se deben implementar medidas de seguridad preventivas y mecanismos de detección de intrusiones para proteger el sistema y los datos de los usuarios.

· **Pruebas de seguridad regulares:**

o El sistema debe someterse a pruebas de seguridad regulares realizadas por expertos independientes para identificar y corregir vulnerabilidades de seguridad. Las pruebas deben incluir análisis de código estático, pruebas de penetración y evaluaciones de vulnerabilidades.

**4. Usabilidad**

· **Facilidad de uso:**

o La interfaz de usuario debe ser intuitiva y fácil de navegar para personas con diferentes niveles de experiencia tecnológica.

o EL sistema debe ofrecer tutoriales e instrucciones claras para guiar a los usuarios en la configuración y uso de todas las funcionalidades.

· **Accesibilidad:**

o El sistema debe ser accesible para personas con discapacidades, incluyendo opciones para usuarios con problemas de visión, audición o movilidad.

o Se deben seguir las pautas de accesibilidad WCAG para garantizar una experiencia inclusiva.

· **Pruebas de usabilidad con usuarios reales:**

o El sistema debe ser evaluado por usuarios reales a través de pruebas de usabilidad para identificar problemas de diseño, navegación y usabilidad. Los resultados de las pruebas deben utilizarse para mejorar la experiencia del usuario siendo más fácil de usar.

**5. Mantenibilidad**

· **Código modular y documentado:**

o El código fuente del sistema debe estar organizado en módulos bien definidos y documentados con comentarios claros y descriptivos.

o La documentación debe incluir información sobre la arquitectura del sistema, las funcionalidades implementadas y las decisiones de diseño tomadas.

· **Soporte técnico:**

o El sistema debe ofrecer soporte técnico de alta calidad a los usuarios para ayudarlos a resolver problemas y utilizar el sistema de manera efectiva. El soporte técnico debe estar disponible a través de correo electrónico.

· **Guías, tutoriales y referencias técnicas:**

o El sistema debe proporcionar guías de usuario, tutoriales y referencias técnicas. La documentación debe estar disponible en línea y ser fácil de encontrar y utilizar.

· **Pruebas unitarias y de integración:**

o El sistema debe contar con un conjunto completo de pruebas unitarias y de integración que automaticen la verificación del correcto funcionamiento de las funcionalidades.

o Las pruebas deben cubrir todos los componentes del sistema, incluyendo la lógica de negocio, la interfaz de usuario y la interacción con la base de datos.

· **Herramientas de seguimiento de errores:**

o Se debe utilizar un sistema de seguimiento de errores para registrar y gestionar eventos y métricas de rendimiento clave, además de permitir el reporte por los usuarios o los detectados durante las pruebas del sistema.

o El sistema debe permitir la asignación de prioridades, la asignación de responsables y el seguimiento del progreso de la resolución de errores.

· **Facilidad de despliegue:**

o El proceso de despliegue de nuevas versiones del sistema debe ser automatizado, regular en tiempo y sencillo, minimizando el tiempo de inactividad y los errores de implementación. Las actualizaciones deben ser fáciles de instalar.

o Se deben utilizar herramientas de integración y entrega continuas para agilizar el proceso de desarrollo y despliegue.

**6. Accesibilidad**

· **Conformidad con WCAG 2.1:**

o El sistema debe cumplir con las pautas de accesibilidad WCAG 2.1 para garantizar que sea accesible para personas con discapacidades.

o Se deben evaluar y corregir cualquier problema de accesibilidad que se identifique durante el desarrollo o pruebas.

· **Soporte para lectores de pantalla:**

o El sistema debe proporcionar una experiencia de usuario fluida y completa para usuarios que utilizan lectores de pantalla.

o El contenido del sistema debe estar adecuadamente etiquetado y estructurado para facilitar la lectura por parte de los lectores de pantalla.

· **Contraste de color adecuado:**

o El sistema debe utilizar una paleta de colores con un contraste adecuado para garantizar la legibilidad del texto y los elementos gráficos para personas con problemas de visión.

o Se deben ofrecer opciones para ajustar el contraste de la interfaz según las preferencias del usuario.

**7. Confiabilidad:**

· **Disponibilidad:**

o El sistema debe estar disponible para los usuarios la mayor parte del tiempo posible, con un objetivo de tiempo de actividad del 99,9%. Se deben implementar estrategias de alta disponibilidad y tolerancia a fallos para minimizar el tiempo de inactividad.

· **Recuperación ante desastres:**

o El sistema debe contar con un plan de recuperación ante desastres para garantizar la continuidad del negocio en caso de eventos inesperados, como cortes de energía o ataques cibernéticos. El plan debe incluir procedimientos para la copia de seguridad de datos, la restauración del sistema y la recuperación rápida de la funcionalidad.

# Recursos disponibles

## Personal

Este proyecto contará con un equipo multidisciplinario en desarrollo de software, diseño y gestión de proyectos. El equipo deberá cubrir estos roles para llevar a cabo las tareas en cada área.

**Product Owner**: definiremos las necesidades del usuario y guiaremos el desarrollo del proyecto según esas especificaciones.

**Programadores UX/UI**: Crearemos la interfaz de usuario con la premisa de dar una experiencia intuitiva y atractiva para los usuarios.

**Analistas funcionales**: deberemos analizar los requisitos del sistema y elaborar a partir de los mismos casos de uso.

**Programadores Backend**: Desarrollaremos la lógica del sistema del lado del servidor.

**Administradores de Bases de Datos**: Gestionaremos y mantendremos la base de datos del sistema, sobre todo para garantizar la integridad y disponibilidad de la información.

## Material

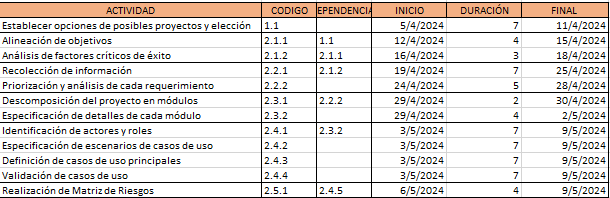
El material necesario para el diseño, desarrollo, implementación y prueba del sistema tenido en cuenta por el personal para abordar el alcance del proyecto es el siguiente:

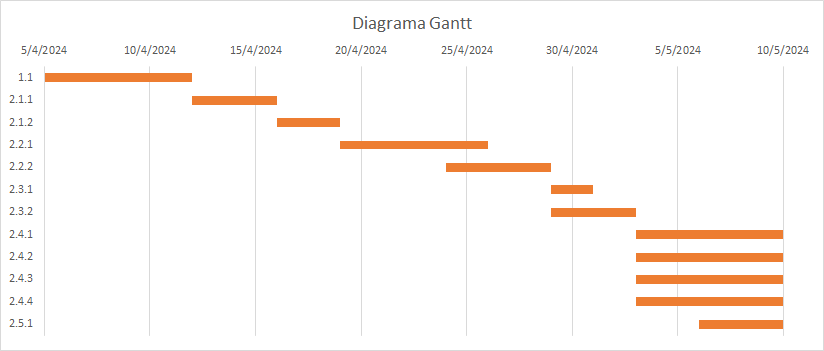
* **Computadoras**: deberán estar equipadas con el software necesario para cada etapa del proyecto. Las siguientes son:
  + **Visual Studio Code**: editor de código con una amplia variedad de extensiones que facilitan el desarrollo y la depuración de aplicaciones web.
  + **DB Browser**: para visualizar y manipular bases de datos SQLite, comunes para el almacenamiento de datos estructurados.
  + **Navegadores Web**: necesarios para probar y depurar el sistema en diferentes entornos de usuario, asegurando la compatibilidad y el rendimiento en distintos navegadores
  + **Software de dibujo de gráficos**: Facilita la creación de diagramas, prototipos de interfaz de usuario y otros elementos visuales importantes.

## Tiempo

El tiempo del proyecto será de una cursada completa, la cual se divide en dos partes principales. Dentro del diagrama de Gantt, se especifica qué tareas se llevarán a cabo a lo largo del tiempo del desarrollo proyecto.

# Plan de trabajo





# Gestión de riesgos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Severidad X Probabilidad** | **Nada Probable** | **Ligeramente Probable** | **Un poco Probable** | **Muy Probable** | **Extremadamente Probable** |
| **Insignificante** | Riesgo trivial | Riesgo trivial | Riesgo bajo | Riesgo moderado | Riesgo moderado |
| **Menor** | Riesgo trivial | Riesgo bajo | Riesgo moderado | Riesgo moderado | Riesgo alto |
| **Moderado** | Riesgo bajo | Riesgo moderado | Riesgo alto | Riesgo alto | Riesgo crítico |
| **Crítico** | Riesgo moderado | Riesgo alto | Riesgo crítico | Riesgo crítico | Riesgo catastrófico |
| **Catastrófico** | Riesgo alto | Riesgo crítico | Riesgo catastrófico | Riesgo catastrófico | Riesgo catastrófico |

**Tabla de Riesgos:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **Tipo de riesgo** | **Probabilidad** | **Severidad** | **Medidas de Mitigación** |
| Retrasos en la planificación y definición de los alcances del proyecto:  - Falta de claridad en los objetivos y entregables.  - Cambios constantes en los requisitos.  - Estimaciones de tiempo y recursos imprecisas. | Alto | Muy probable | Moderado | \* Definir objetivos y entregables claros y medibles al inicio del proyecto.  \* Establecer un proceso de comunicación efectivo para gestionar cambios en los requisitos.  \* Realizar una planificación detallada del proyecto, incluyendo estimaciones de tiempo y recursos realistas.  \* Utilizar herramientas de gestión de proyectos para monitorear el progreso y detectar desviaciones tempranamente. |
| Problemas de comunicación y colaboración entre los miembros del equipo:  - Falta de coordinación y trabajo en equipo.  - Dificultades para compartir información y conocimiento.  - Conflictos y desacuerdos entre los miembros del equipo. | Alto | Un poco probable | Moderado | \* Definir roles y responsabilidades claras para cada miembro del equipo.  \* Establecer canales de comunicación efectivos y fomentar la comunicación abierta y transparente.  \* Realizar reuniones periódicas para discutir el progreso del proyecto y resolver problemas.  \* Implementar herramientas de colaboración que faciliten el intercambio de información y la gestión conjunta de tareas. |
| Falta de habilidades o experiencia del equipo para llevar a cabo las tareas del proyecto:  - Dificultades para completar las tareas asignadas.  - Retrasos en la entrega de resultados.  - Errores o defectos en el trabajo realizado. | Moderado | Un poco probable | Moderado | \* Evaluar las habilidades y experiencia del equipo al inicio del proyecto.  \* Brindar capacitación y entrenamiento al equipo para desarrollar las habilidades necesarias.  \* Considerar la contratación de personal externo con las habilidades y experiencia requeridas.  \* Fomentar el aprendizaje continuo y el intercambio de conocimientos entre los miembros del equipo. |
| Gestión inadecuada de los riesgos del proyecto:  - Falta de identificación y evaluación de los riesgos potenciales.  - Ausencia de planes de mitigación para los riesgos identificados.  - Impacto negativo de los riesgos en el proyecto sin planes de contingencia. | Moderado | Ligeramente probable | Moderado | \* Implementar un proceso formal de gestión de riesgos para identificar, evaluar y mitigar los riesgos potenciales.  \* Desarrollar planes de contingencia para responder a los riesgos que se materialicen.  \* Monitorear y actualizar el plan de gestión de riesgos a lo largo del proyecto. |
| Motivación y compromiso insuficientes del equipo:  - Baja participación y entusiasmo de los miembros del equipo.  - Dificultades para cumplir con los plazos y objetivos establecidos.  - Aumento del absentismo o la rotación del personal. | Bajo | Ligeramente probable | Menor | \* Definir objetivos claros y desafiantes para el equipo.  \* Brindar reconocimiento y recompensas por el trabajo bien realizado.  \* Fomentar un ambiente de trabajo positivo y colaborativo.  \* Ofrecer oportunidades de crecimiento y desarrollo profesional a los miembros del equipo. |

# Diseño del sistema

## Arquitectura

1. Capa de presentación (UI/UX):

Responsable de la interacción con el usuario.

Implementada utilizando tecnologías web: HTML, CSS y JavaScript.

Desarrollada por los programadores UX/UI.

1. Capa de negocio (Backend):

Contiene la lógica del sistema y las reglas de negocio.

Implementada utilizando lenguajes de programación: Python y Flask.

Desarrollada por los programadores backend.

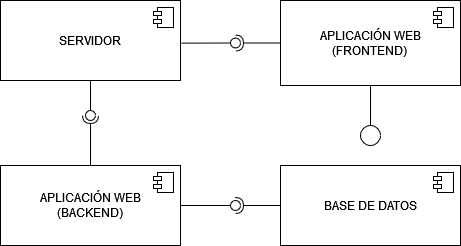
1. Capa de datos:

Almacena y administra los datos del sistema.

Utiliza una base de datos relacional: MySQL.

Gestionada por los administradores de bases de datos.

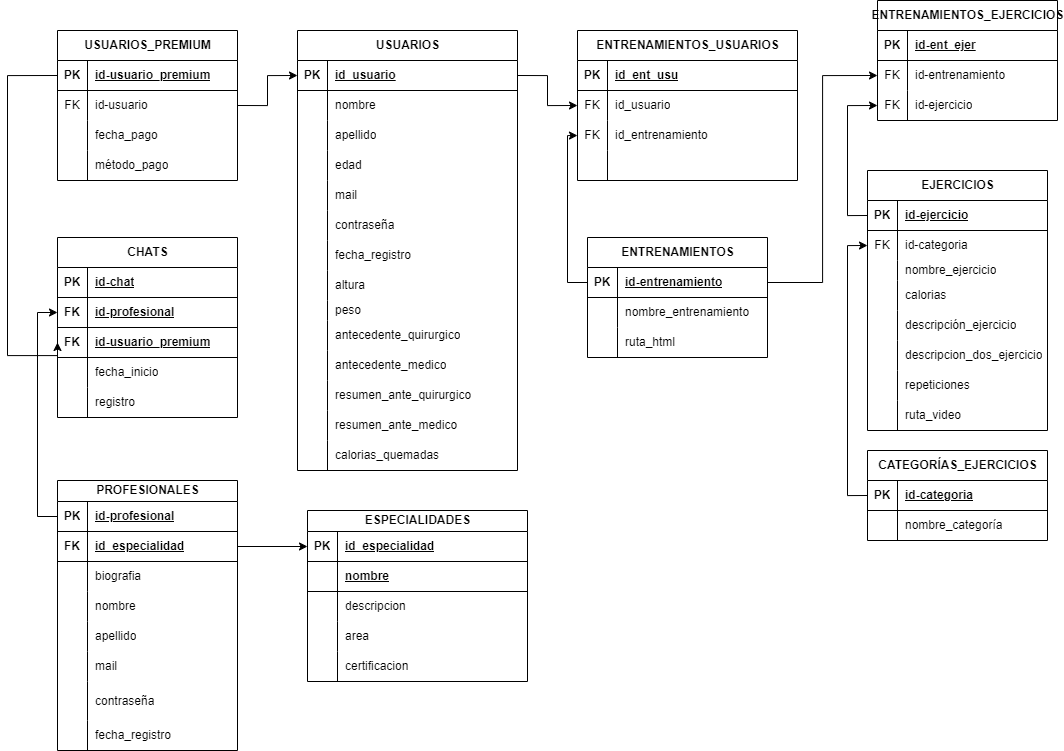
## Diagrama de Componentes



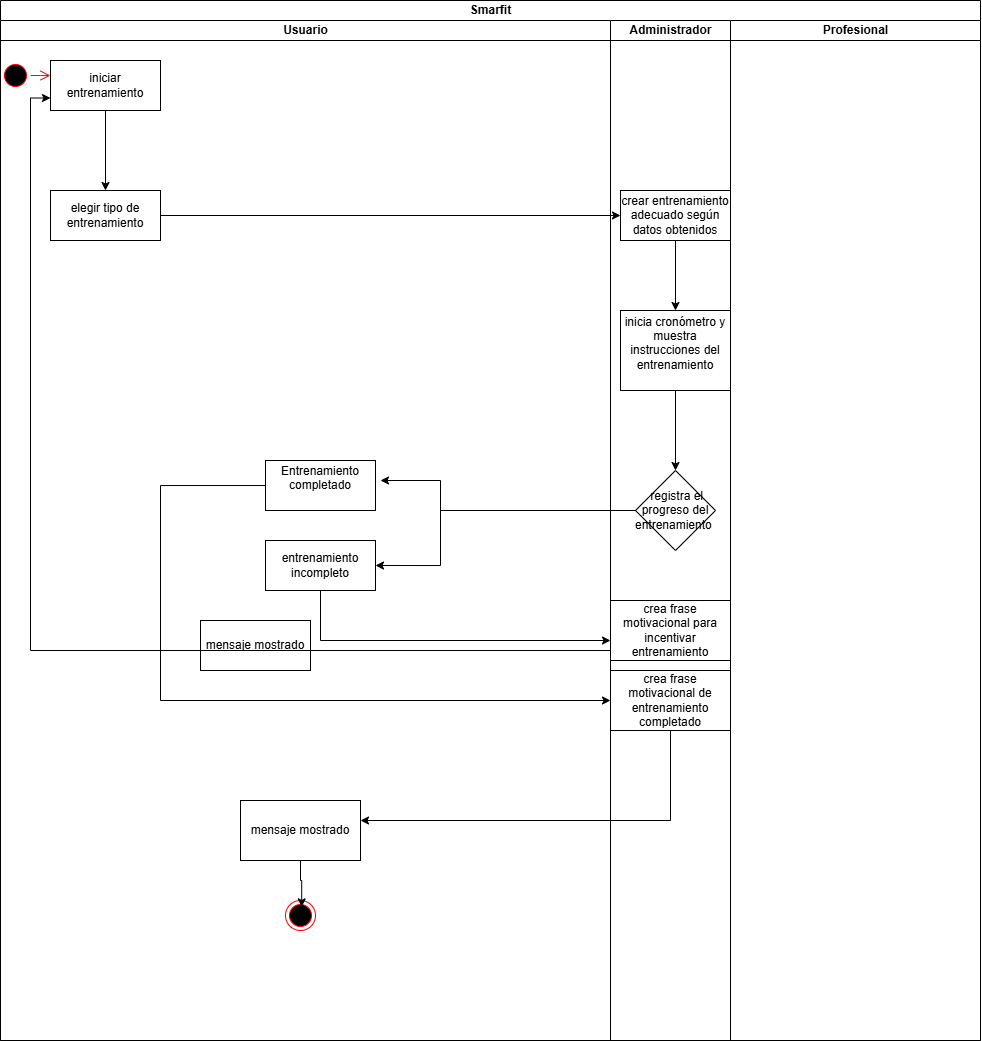
Este diagrama nos permite representar cuatro componentes:

* El Servidor, que sirve los recursos estáticos y maneja las solicitudes HTTP.
* Frontend de la Aplicación Web: Representa la interfaz de usuario, usualmente desarrollada con HTML, CSS, y JavaScript
* Backend de la Aplicación Web: Maneja la lógica del negocio y las reglas de la aplicación.
* Base de Datos: Almacena y gestiona los datos de la aplicación.

## Diagrama de Entidad-Relación



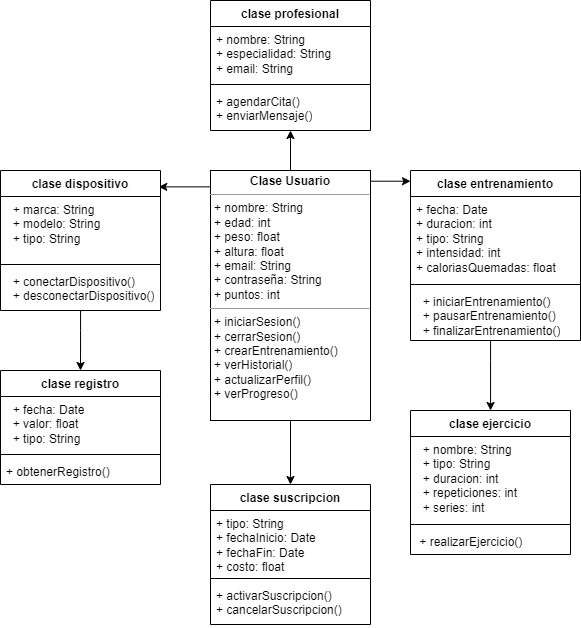
## Diagramas de Actividades



## 



## Diagrama de Clases



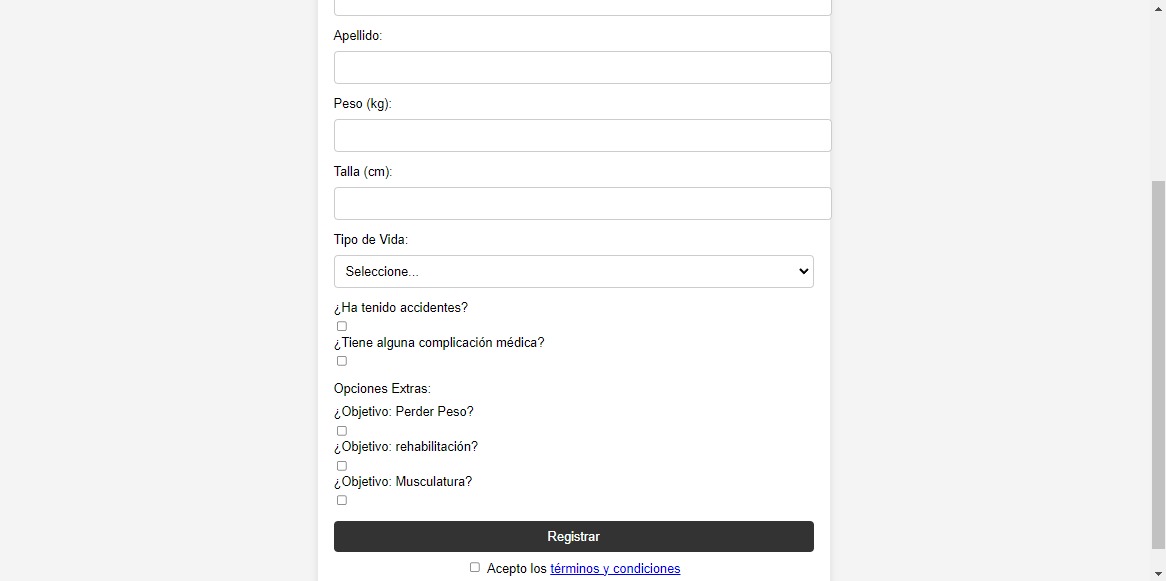
## Interfaces de Usuario

Página de Inicio de SmartFit

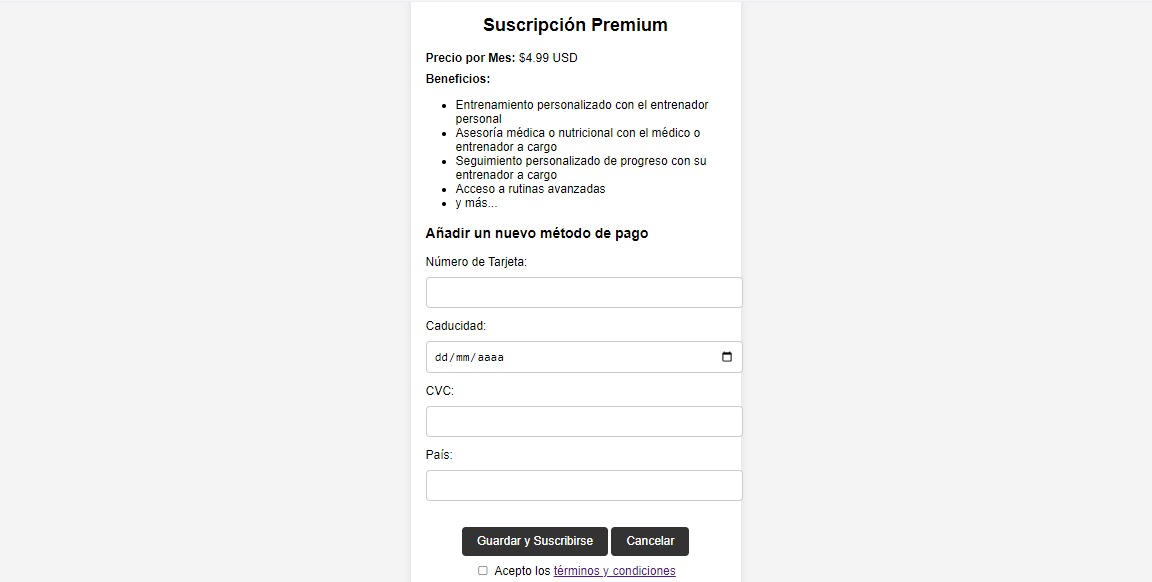


Página de Registro de Usuario

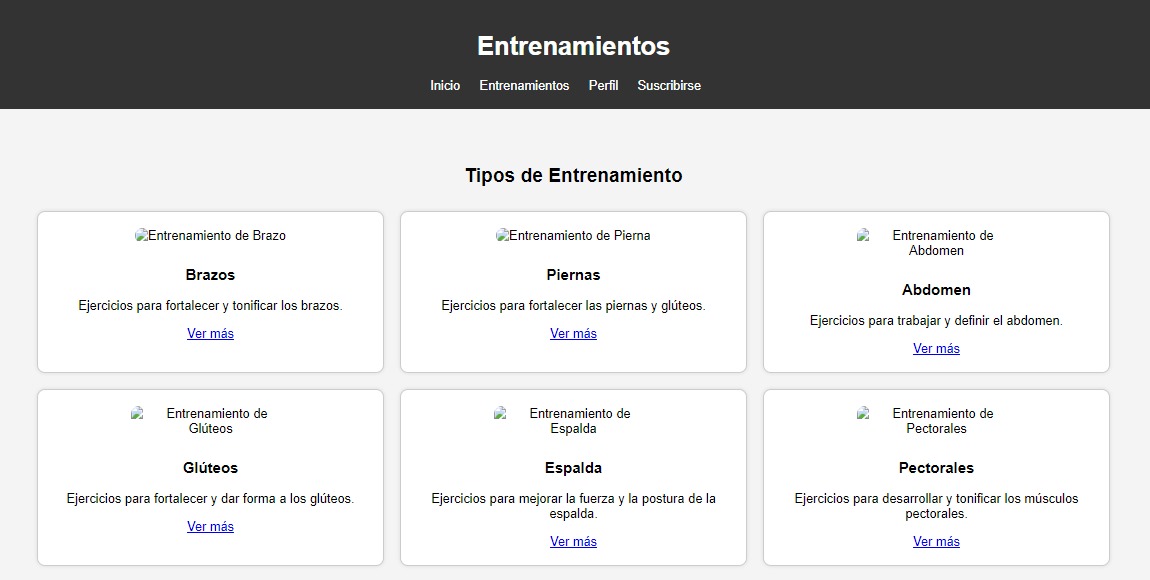




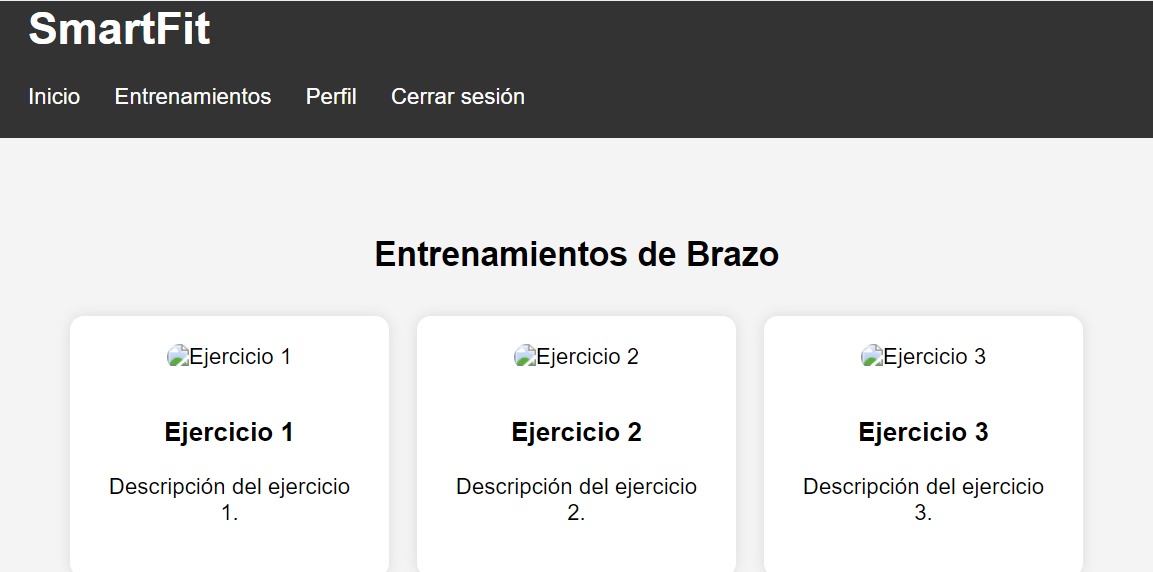
Página del Formulario de Suscripción a Premium



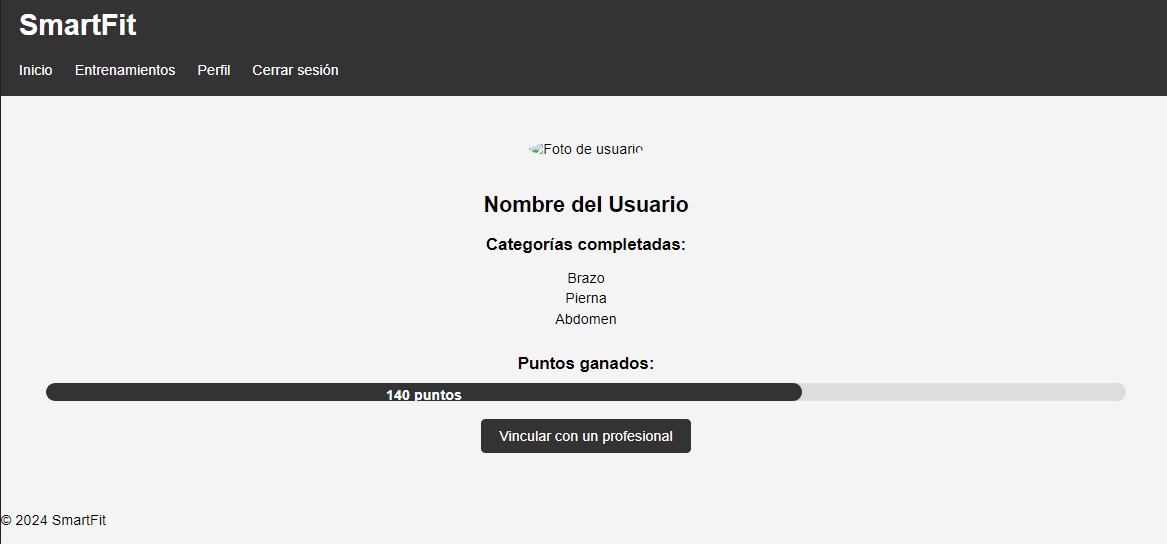
Página de Entrenamiento



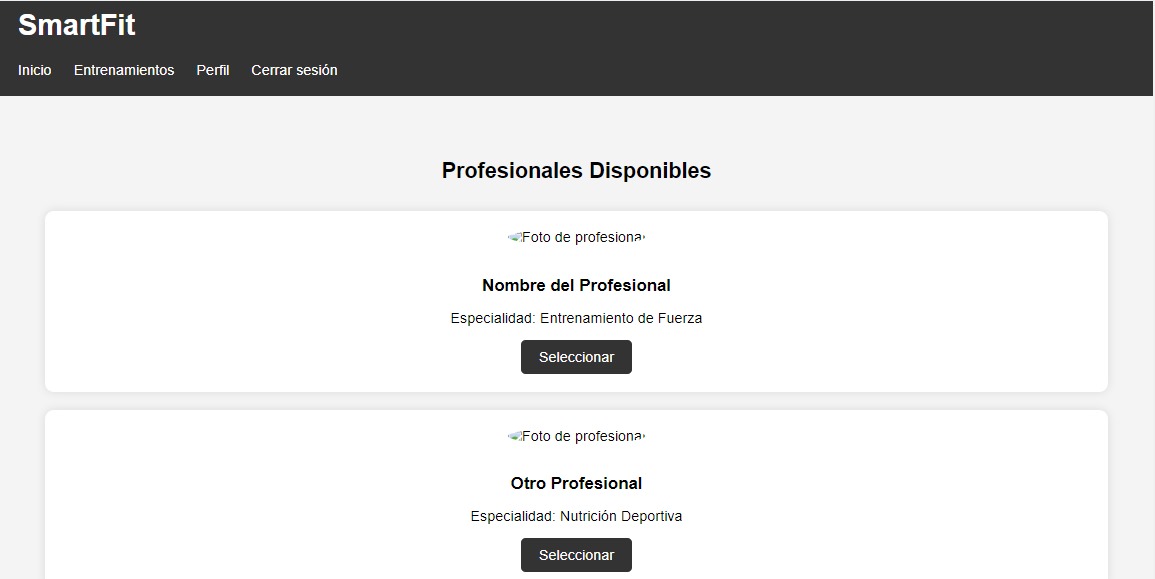
Página de Entrenamiento Específico



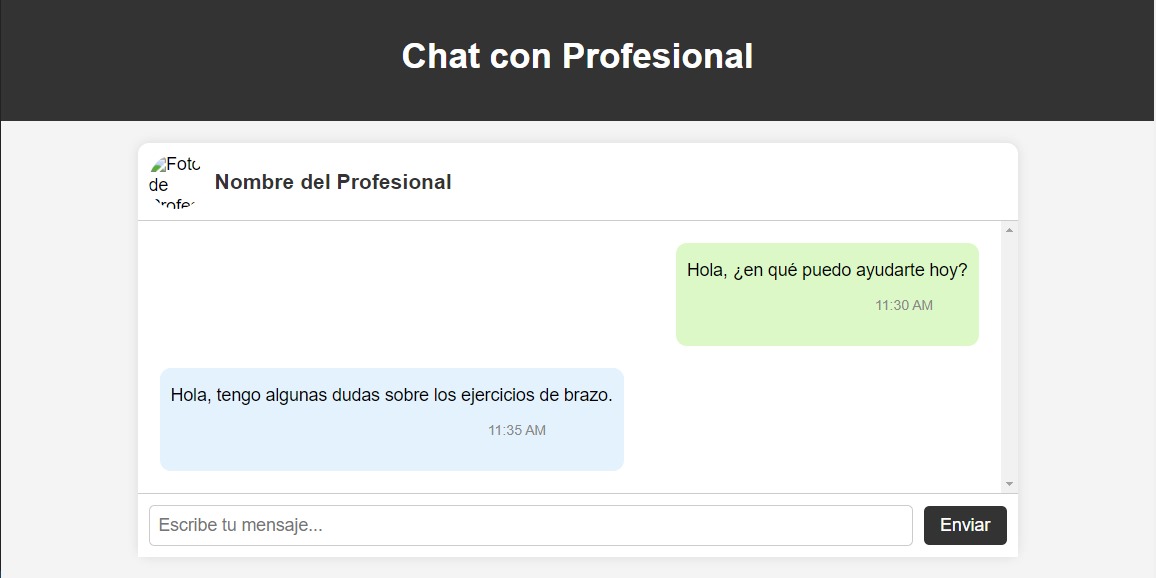
Página de Perfil de Usuario



Página de Servicios Profesionales



Página de Chat con Profesional Elegido



# Desarrollo

La función siguiente función permite que un usuario logueado registre su progreso de calorías y entrenamientos, actualizando así su historial. Esto refleja uno de los objetivos del proyecto, que es brindar al usuario un seguimiento continuo de su progreso en cada entrenamiento.

@app.route('/actualizar\_calorias\_y\_entrenamiento', methods=['POST'])

def actualizar\_calorias\_y\_entrenamiento():

    if 'id\_usuario' in session:

        datos = request.get\_json()

        nuevas\_calorias = datos.get('calorias', 0)

        entrenamiento\_id = datos.get('entrenamiento\_id')

        conn = sqlite3.connect('smartfit.db')

        c = conn.cursor()

        c.execute("UPDATE Usuarios SET calorias\_quemadas = calorias\_quemadas + ? WHERE id\_usuario = ?",

                  (nuevas\_calorias, session['id\_usuario']))

        c.execute("INSERT OR IGNORE INTO Entrenamientos\_usuarios (id\_usuario, id\_entrenamiento) VALUES (?, ?)",

                  (session['id\_usuario'], entrenamiento\_id))

        conn.commit()

        conn.close()

        return jsonify({"success": True}), 200

    else:

        return jsonify({"error": "Usuario no autenticado"}), 401

El siguiente proceso es parte de un conjunto de funciones que filtran entrenamientos específicos según nivel y tipo. Primero conecta a la base de datos, selecciona ejercicios específicos según el nivel, con detalles como nombre, descripción, repeticiones, calorías y ruta del video, y luego cierra la conexión. La página indicada se renderiza con estos datos y el nombre del usuario logueado si está en sesión.

@app.route('/P\_A')

def principiante\_abdomen():

    conn = sqlite3.connect('smartfit.db')

    cursor = conn.cursor()

    cursor.execute("SELECT nombre\_ejercicio, descripcion\_ejercicio, descripcion\_dos\_ejercicio, repeticiones, calorias, ruta\_video FROM Ejercicios WHERE ruta\_video LIKE '%1.ABDOMINALES%' AND nombre\_ejercicio LIKE '%(Principiante)'")

    ejercicios = cursor.fetchall()

    conn.close()

    usuario\_logueado = session.get('nombre', None)

    return render\_template('entrenamientos/P\_A.html', ejercicios=ejercicios, usuario\_logueado=usuario\_logueado)

Este código (JavaScript) calcula el IMC del usuario y determina su estado de peso al hacer clic en "Calcular". Según el IMC, asigna un rango y sugiere una meta diaria de calorías. Luego, muestra un mensaje con el IMC y la recomendación de calorías

document.getElementById('calcularBtn').addEventListener('click', function () {

    const nombre = document.getElementById('nombre').value;

    const altura = parseFloat(document.getElementById('altura').value) / 100; // Convertimos de cm a metros

    const peso = parseFloat(document.getElementById('peso').value);

    if (isNaN(altura) || isNaN(peso) || !nombre) {

        alert("Por favor, ingresa todos los campos.");

        return;

    }

    const imc = peso / (altura \* altura);

    let estadoPeso = '';

    let caloriasQuemar = 0;

    if (imc < 18.5) {

        estadoPeso = 'Bajo peso';

        caloriasQuemar = 1500; // Ejemplo

    } else if (imc >= 18.5 && imc <= 24.9) {

        estadoPeso = 'Saludable';

        caloriasQuemar = 2000; // Ejemplo

    } else if (imc >= 25 && imc <= 29.9) {

        estadoPeso = 'Exceso de peso';

        caloriasQuemar = 2500; // Ejemplo

    } else {

        estadoPeso = 'Obeso';

        caloriasQuemar = 3000; // Ejemplo

    }

    document.getElementById('resultadoIMC').innerHTML = `

        <p>Hola ${nombre}, tu IMC es de ${imc.toFixed(2)} y tu estado de peso es ${estadoPeso}.

        Necesitarías quemar aproximadamente ${caloriasQuemar} calorías al día.</p>`;

});

# Implementación y prueba

## Prueba unitaria: Gestión de Calorías

Esta prueba unitaria resume la lógica lo que hemos diseñado para gestionar el cálculo total de las calorías quemadas en cada entrenamiento. Las calorías quemadas por el usuario se registran en su perfil, permitiendo llevar seguimiento de todas las calorías quemadas a lo largo de las sesiones de entrenamiento que realice.

calorias\_usuario = 0  # calorías iniciales del usuario

calorias\_ejercicios\_entrenamiento = [20, 20, 15]  # calorías por cada ejercicio del entrenamiento

def calcular\_calorias\_entrenamiento(): # Calcula las calorías quemadas en el entrenamiento

    calorias\_entrenamiento = sum(calorias\_ejercicios\_entrenamiento)

    print(f"Calorías que se queman en este entrenamiento: {calorias\_entrenamiento} kcal")

    return calorias\_entrenamiento

def actualizar\_calorias\_usuario(calorias): #actualiza el perfil del usuario con las calorías que quemo en el entrenamiento

    global calorias\_usuario

    calorias\_usuario += calorias

    print(f"Calorías totales quemadas en el perfil del usuario: {calorias\_usuario} kcal") # El total de calorías se muestra en el Perfil

calorias\_sesion = calcular\_calorias\_entrenamiento()

entrenamiento\_finalizado = input("Ingrese 'Finalizado' para terminar el entrenamiento: ")

if entrenamiento\_finalizado == "finalizado":

    actualizar\_calorias\_usuario(calorias\_sesion)  # Si el usuario finaliza entrenamiento, se actualiza las calorías quemadas por el usuario

    print("Entrenamiento completado: ya puede ver el total de calorías quemadas en su  perfil.")

else:

    print("Entrenamiento no fue finalizado")

Hacer esta prueba nos facilitó la organización de los elementos en diferentes archivos: Las calorías quemadas por cada ejercicio se registran en la base de datos, al igual que las calorías totales que el usuario acumula. El frontend muestra los ejercicios disponibles y permite al usuario indicar cuáles ha completado, mientras que el backend se encarga de gestionar qué usuario está realizando esos entrenamientos y, por ende, cuántas calorías quemó.

## Prueba unitaria: Gestión de Entrenamientos

Esta prueba unitaria demuestra la intención de ofrecer a los usuarios la posibilidad de almacenar y gestionar sus entrenamientos desde la facilidad de su Perfil. Buscamos que al permitir a los usuarios agregar el nombre de cada entrenamiento realizado, que se facilite el acceso a sus ejercicios preferidos desde su perfil. Con ello mejoramos la experiencia general del usuario y hacemos una personalización a su rutina de ejercicios.

entrenamientos\_usuario = []  # Variable que almacena los entrenamientos del usuario

def agregar\_entrenamiento(nombre\_entrenamiento): # Agrega entrenamiento a la lista

    entrenamientos\_usuario.append(nombre\_entrenamiento)

    print(f"Entrenamiento '{nombre\_entrenamiento}' agregado al perfil del usuario")

def mostrar\_entrenamientos(): #Muestra todos los entrenamientos que tenga guardados el usuario en el perfil

    if entrenamientos\_usuario:

        print("Entrenamientos guardados en el perfil:")

        for i, entrenamiento in enumerate(entrenamientos\_usuario, start=1):

            print(f"{i}. {entrenamiento}")

    else:

        print("No hay entrenamientos guardado")

agregar\_entrenamiento("Entrenamiento de Fuerza")

agregar\_entrenamiento("Entrenamiento Cardio")

agregar\_entrenamiento("Entrenamiento de Flexibilidad")

mostrar\_entrenamientos()

## Prueba unitaria: Cálculo de IMC

Esta prueba unitaria resume la lógica que hemos diseñado para gestionar el cálculo del índice de masa corporal (IMC) del usuario de manera precisa. El cálculo del IMC se basa en el peso y la altura del usuario, lo que permite obtener un indicador general de su condición física y controlar si se encuentra en un rango saludable.

Esta estructura también es compatible con posibles ampliaciones del proyecto, como el registro de resultados en una base de datos o en el perfil del usuario. Además, los datos pueden analizarse para ofrecer recomendaciones personalizadas basadas en el IMC del usuario, facilitando un seguimiento de su progreso de salud y condición física.

# Función para calcular el IMC

def calcular\_imc(peso, altura):

    # Calcular el IMC

    imc = peso / (altura \*\* 2)

    return round(imc, 2)  # Redondear a dos decimales

# Ejemplos de uso

peso = 70  # en kg

altura = 1.75  # en metros

imc\_resultado = calcular\_imc(peso, altura)

print("El IMC es:",imc\_resultado)

# Conclusiones finales

El desarrollo de esta aplicación web ha sido una experiencia enriquecedora que ha permitido consolidar conocimientos en diversas áreas de la ingeniería de software. La implementación de una metodología ágil, en particular Scrum, ha sido fundamental para alcanzar los objetivos planteados y adaptarnos a los cambios que surgieron durante el proyecto. Aspectos Positivos y Aprendizajes:

* **Trabajo en Equipo y Comunicación**: A lo largo del proyecto, se llevaron a cabo un total de 3 reuniones semanales, las cuales fueron fundamentales para mantener al equipo alineado y tomar decisiones oportunas. La organización de reuniones periódicas y la documentación exhaustiva de las decisiones han sido claves para mantener un trabajo coordinado y eficiente, incluso con la incorporación de un nuevo miembro al equipo a mitad del proyecto. La presentación elaborada para dar a conocer el proyecto al nuevo integrante demostró la importancia de una comunicación clara y concisa.
* **Adaptabilidad y Flexibilidad**: La elección de una metodología ágil nos permitió ser flexibles ante los cambios y ajustar el alcance del proyecto según las necesidades. La etapa de codificación, aunque demandó un mayor esfuerzo debido a la adaptación del código a las funcionalidades, fue una oportunidad para afinar nuestras habilidades de desarrollo y depuración.
* **Gestión de la Documentación**: La experiencia de no reflejar adecuadamente el trabajo realizado en una entrega nos llevó a implementar un proceso de revisión constante de la documentación, asegurando que esta estuviera siempre actualizada y alineada con el avance del proyecto. Gracias a la documentación actualizada, se estima que se ahorró aproximadamente un 20% del tiempo dedicado a la resolución de dudas y la búsqueda de información ya que esta práctica es fundamental para garantizar la trazabilidad y la transparencia en el desarrollo de software.
* **Definición de Requisitos**: La elección de la programación estructurada permitió definir de manera clara y concisa el proceso de codificación, facilitando la comprensión y el mantenimiento del código. Sin embargo, también evidenció la importancia de una constante revisión y ajuste de los requisitos, ya que estos pueden evolucionar a lo largo del proyecto.
* **Herramientas de Gestión de Proyectos**: La utilización de herramientas especializadas para la gestión de proyectos ágiles, como Jira o Trello, optimizaron el seguimiento de las tareas y la colaboración entre los miembros del equipo.

**Áreas de Mejora**:

* **Gestión de Cambios**: Si bien la metodología ágil permite adaptarse a los cambios, es fundamental establecer un proceso claro para gestionar estos cambios y evaluar su impacto en el alcance y el cronograma del proyecto.
* **Automatización de Pruebas**: La implementación de pruebas automatizadas podría mejorar la calidad del software y reducir el tiempo de detección de errores.

**Reflexiones Finales**:

Los factores que influyeron en el éxito del proyecto son el compromiso del equipo, adopción de la metodología ágil y la documentación detallada. A partir de esta experiencia, hemos aprendido la importancia de priorizar las funcionalidades que tengan un mayor valor de impacto en el usuario final, realizar pruebas exhaustivas e invertir tiempo en la mejora de una interfaz intuitiva y atractiva Este proyecto ha sido una valiosa oportunidad para aplicar los conocimientos adquiridos en la formación académica y desarrollar nuevas habilidades en el ámbito de la ingeniería de software. La experiencia adquirida en la gestión de proyectos, la comunicación efectiva y la resolución de problemas será fundamental para afrontar futuros desafíos en el desarrollo de software