

Trabajo Práctico - Proyecto de Desarrollo en Python

Cada alumno o grupo deberá redactar un documento formal donde se describa su proyecto, incluyendo los siguientes puntos:

1. Introducción y Descripción del Proyecto:

- Explicar qué desean desarrollar y cuál es el propósito del proyecto.
- Definir qué problema o necesidad aborda la solución.

2. Alcance del Proyecto:

- Determinar los límites del sistema, especificando qué funcionalidades incluirá y cuáles no.
- Definir quiénes serán los usuarios finales y cómo interactuarán con el sistema.

3. Justificación y Necesidad:

- Explicar por qué el proyecto es relevante y útil.
- Contextualizar el problema dentro de un área específica (ejemplo: automatización, análisis de datos, web scraping, etc.).

4. Tecnologías a Utilizar:

- Especificar qué tecnologías y herramientas se emplearán para el desarrollo (Ejemplo: Python, Django, Flask, Pandas, FastAPI, PostgreSQL, etc.).
- Justificar la elección de cada tecnología.

5. Estructura del Proyecto:

- Proporcionar un esquema o diagrama que represente la estructura del proyecto (puede ser un diagrama de clases, de base de datos, o de flujo de trabajo).

6. Casos de Uso o Ejemplo de Funcionamiento:

- Incluir algunos ejemplos de cómo funcionará el sistema.
- Opcional: Incluir mockups (Bocetos) o interfaces si corresponde.

7. Conclusión y Futuras Mejoras:

- Reflexionar sobre el impacto del proyecto y posibles mejoras futuras.

Criterios de Evaluación:

- Claridad y coherencia en la redacción del documento.
- Aplicación de los conceptos aprendidos en el curso.

- Viabilidad del proyecto dentro del alcance definido.
- Uso adecuado de tecnologías y justificación de su elección.
- Presentación ordenada y profesional del documento.

Entrega:

- El trabajo debe entregarse en formato PDF.
- Se valorará la presentación de un repositorio en GitHub con la estructura del proyecto iniciada.

Desarrollo:

1. Introducción y Descripción del Proyecto:

- **Desarrollo:**

La plataforma propuesta es un sistema integral para la gestión de gimnasios, que incluye registro de usuarios, seguimiento de rutinas, reserva de clases, control de acceso y análisis de datos de entrenamiento.

- **Propósito:**

Simplificar la administración del gimnasio, mejorar la experiencia del cliente y optimizar el rendimiento de los usuarios.

- **Problema/Necesidad:**

La falta de una solución centralizada para la gestión de gimnasios dificulta la organización, la comunicación con los usuarios y el seguimiento de su progreso.

2. Alcance del Proyecto:

- **Funcionalidades:**

- Registro y gestión de usuarios (miembros y entrenadores).
- Creación y asignación de rutinas de entrenamiento.
- Reserva de clases grupales y sesiones individuales.
- Control de acceso al gimnasio (tarjetas o códigos).
- Análisis de datos de entrenamiento (progreso, rendimiento).
- Gestión de pagos y suscripciones.

- **Limitaciones:**

- No se incluye la gestión de inventario de productos del gimnasio.

- No se implementa la comunicación directa entre usuarios y entrenadores dentro de la plataforma (se utilizará WhatsApp como canal externo).
- **Usuarios Finales:**
 - Miembros del gimnasio (acceso a su información, reservas).
 - Entrenadores (gestión de usuarios, rutinas, clases).
 - Administradores del gimnasio (control total del sistema).

3. Justificación y Necesidad:

- **Relevancia:**

La plataforma es relevante para gimnasios de cualquier tamaño, ofreciendo una solución eficiente para la gestión de sus operaciones y la mejora de la experiencia del cliente.

- **Contexto:**

Automatización de tareas administrativas, análisis de datos para mejorar el entrenamiento y seguimiento del progreso de los usuarios.

4. Tecnologías a Utilizar:

- **Frontend:**

[React](#) (o [Vue.js](#)) para la interfaz de usuario, garantizando una experiencia interactiva y responsive.

- **Backend:**

Python con Django (o Flask) para la lógica del servidor y la gestión de la base de datos.

- **Base de Datos:**

PostgreSQL, una base de datos relacional robusta y escalable.

- **Justificación:**

Estas tecnologías ofrecen un equilibrio entre eficiencia, escalabilidad y facilidad de desarrollo. React y Django son populares y bien documentados, facilitando el desarrollo y mantenimiento del sistema. PostgreSQL es una base de datos confiable y potente para aplicaciones web.

5. Estructura del Proyecto:

- Diagrama de clases: Representa las entidades principales del sistema (Usuario, Rutina, Clase, etc.) y sus relaciones.
- Diagrama de flujo: Muestra el flujo de usuarios al utilizar las funcionalidades principales (ej: reserva de clase).

6. Casos de Uso/Ejemplo de Funcionamiento:

- **Caso 1:**

Un miembro inicia sesión, busca una clase de yoga, verifica la disponibilidad y realiza la reserva. Recibe una confirmación por correo electrónico.

- **Caso 2:**

Un entrenador crea una nueva rutina de entrenamiento para un miembro. El miembro recibe una notificación y puede acceder a la rutina en su perfil.

- **Mockups (opcional):**

Bocetos de la interfaz de usuario, mostrando el diseño de la página de inicio, el perfil de usuario, la página de reserva de clases, etc.

7. Conclusión y Futuras Mejoras:

- **Impacto:**

La plataforma optimizará la gestión del gimnasio, mejorando la experiencia del cliente y permitiendo un mejor seguimiento del progreso de los usuarios.

- **Mejoras Futuras:**

- Implementación de la comunicación interna entre usuarios y entrenadores dentro de la plataforma.
- Integración con dispositivos de seguimiento de actividad física (smartwatches).
- Implementación de un sistema de recompensas y desafíos para motivar a los usuarios.