



Facultad de
Ciencias Informáticas

MALLA: Sistemas de información 2019 (presencial) (rediseño 2019)

FACULTAD DE CIENCIAS INFORMÁTICAS

DOCENTE TUTOR: Vargas Nolivos Hernan Patricio

MATERIA: Programación avanzada

NIVEL: sexto

PARALELO: "A"

ESTUDIANTES:

Figueroa Reyes Jorge Leonardo

Quimis Piguave Ricardo Rene

Tipo: Proyecto Medio Ciclo

Documento SRS: Sistema de Administración de Gimnasio (ADMIN GYM)

1. Introducción

1.1 Propósito del Documento

Este documento especifica los requisitos funcionales y no funcionales del sistema ADMIN GYM, una aplicación para la gestión de gimnasios que permite manejar usuarios, rutinas, planes de membresía, pagos, y acceso al sistema a través de credenciales.

1.2 Público Objetivo

Este documento está dirigido a:

- Propietarios o administradores de gimnasios.

1.3 Alcance del Sistema

ADMIN GYM permite:

- Registrar y autenticar usuarios.
- Administrar rutinas personalizadas.
- Gestionar pagos y planes.
- Controlar accesos mediante roles (administrador).

2. Descripción General

2.1 Perspectiva del Producto

ADMIN GYM es una aplicación cliente-servidor construida con Python usando el framework Flask, junto con SQLAlchemy para el manejo de base de datos. La interfaz de usuario se sirve mediante templates HTML.

2.2 Funciones del Producto

- Registro de usuarios.
- Gestión de rutinas (crear, eliminar).
- Visualización de perfiles y planes.

- Control de pagos.
- Panel de administración.

2.3 Características del Usuario

- Administrador: Tiene acceso a todo el sistema, incluyendo gestión de usuarios, rutinas, pagos y reportes.

2.4 Restricciones

- El sistema depende de Flask y SQLAlchemy.
- La base de datos debe estar disponible para la persistencia de datos.
- El acceso está protegido por un sistema de login básico con tokens.

3. Requisitos Específicos

3.1 Requisitos Funcionales

ID	Requisito	Descripción
RF1	Registro de Usuario	El sistema debe permitir el registro de nuevos usuarios.
RF3	Gestión de Rutinas	El administrador puede crear, editar y eliminar rutinas.
RF4	Asignación de Planes	El administrador puede asignar planes de entrenamiento a los usuarios.
RF5	Gestión de Pagos	El sistema debe registrar pagos y vincularlos con los usuarios.

3.2 Requisitos No Funcionales

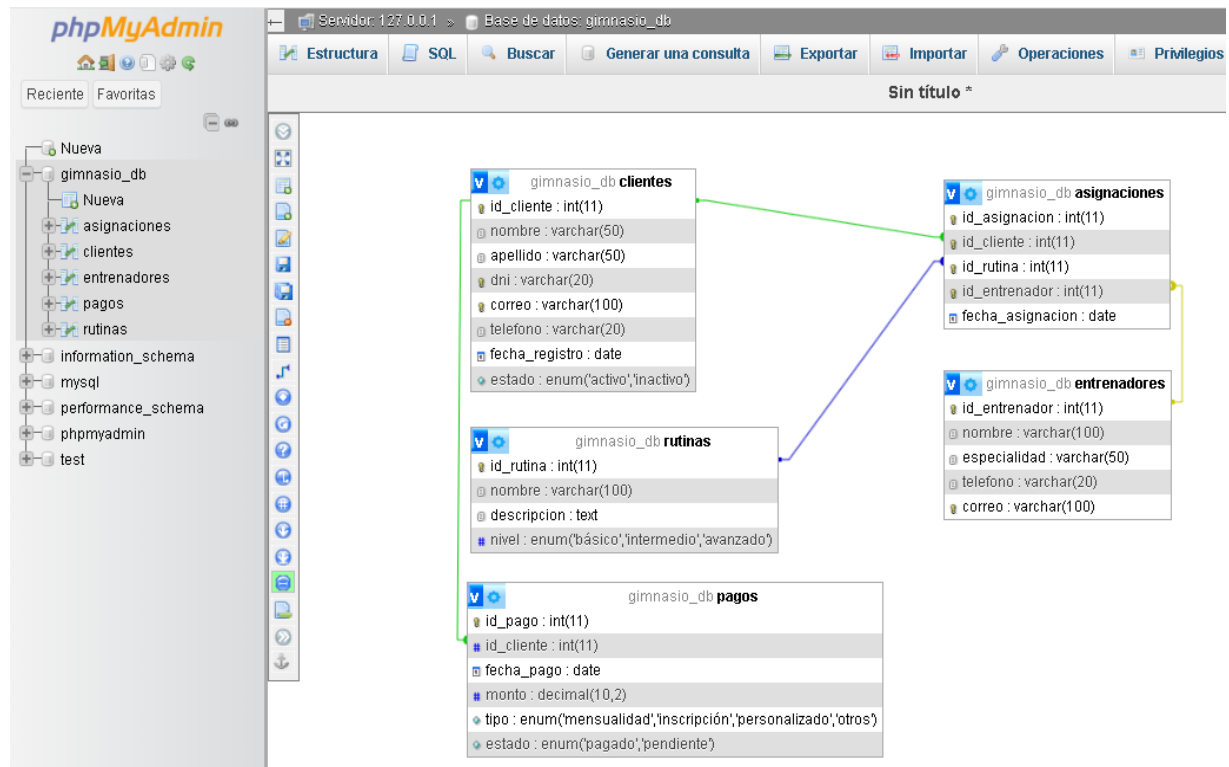
ID	Requisito	Descripción
RNF1	Usabilidad	La interfaz debe ser clara y fácil de usar.
RNF2	Rendimiento	El sistema debe cargar las páginas principales en menos de 2 segundos.
RNF3	Seguridad	Las contraseñas deben estar cifradas. Los accesos deben estar protegidos por tokens.
RNF4	Escalabilidad	El sistema debe permitir el aumento del número de usuarios sin cambios en la arquitectura.

3.3 Reglas del Negocio

- Los administradores pueden registrar, modificar y eliminar información sobre clientes, planes y rutinas.
- Cada cliente debe estar asociado a un plan de membresía y una rutina asignada.
- El sistema debe registrar pagos y permitir el seguimiento del estado de pago de cada cliente.
- Los pagos vencidos deben marcarse para su seguimiento, pero no afectan el acceso del administrador al sistema.

4. Diagramas

- Diagrama Entidad-Relación de la base de datos.



5. Apéndices

- Tecnologías utilizadas: Python, Flask, SQLAlchemy, SQLite.
- Repositorio del código: Local (App.py, models.py, etc.).
- Estructura de carpetas:

