



# **INSTITUTO POLITÉCNICO NACIONAL**

---

**UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y  
CIENCIAS SOCIALES Y ADMINISTRATIVAS**

## **MANUAL TÉCNICO VERSIÓN:3**

### **PROTOTIPO DE UNA PÁGINA WEB PARA EL CONTROL DE ACCESO A UN GIMNASIO**



#### **Equipo Integrantes:**

**Mora Flores Omar  
Lugo Mora José Yael  
Ramírez Ramos Ángel Jair  
Ramírez Blas Luis Angel  
Trejo Monroy Stefany**



# Índice

## Contenido

Índice.....	2
Objetivo .....	3
<b>General</b> .....	<b>3</b>
<b>Específicos</b> .....	<b>3</b>
<b>Requerimientos Técnicos</b> .....	<b>3</b>
<b>Instalación</b> .....	<b>3</b>
Creación de la Base de Datos .....	9
<b>Diseño de arquitectura</b> .....	<b>11</b>
Arquitectura .....	11
Tecnologías .....	12
Estructura del proyecto .....	13
Diagrama de clases .....	14
Diagrama entidad Relación .....	15
Diagrama de Despliegue .....	15
Diagrama de Estados .....	16

# Objetivo

## General

Desarrollar un prototipo de un sistema de reconocimiento facial de control de acceso para gimnasios, modernizando el proceso de registro y control de acceso de los usuarios, este sistema tendrá una interfaz que permita a los administradores del gimnasio registrar nuevos usuarios, gestionar membresías y revisar registros de acceso, utilizando una base de datos relacional para almacenar la información de los usuarios y datos relacionados al gimnasio, contará con un control de acceso por medio de un sistema de reconocimiento facial para identificar y autenticar a los usuarios. Se busca con el desarrollo del prototipo del sistema ofrecer una alternativa moderna y más eficiente a la que se utiliza en la actualidad en los gimnasios, debido a que sigue manejando métodos de control de acceso y registro de los miembros en formatos impresos en papel o en Excel, el nuevo sistema permitirá que esté proceso sea rápido y sencillo tanto para el administrador como para el miembro del gimnasio.

## Específicos

- Implementar un nuevo sistema de gestión y control que reemplace los métodos basados en papel y Excel, mejorando así la eficiencia operativa y la experiencia del cliente en el gimnasio.
- Establecer un procedimiento biométrico para la autenticación del usuario.
- Agilizar el proceso de registro de clientes y el control de acceso al gimnasio.

## Requerimientos Técnicos

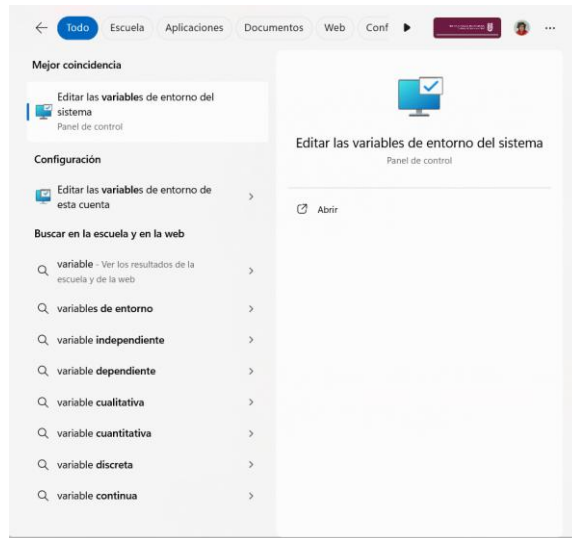
Compatible con SO windows 10 y 11, MAC os

requisitos mínimos de hardware

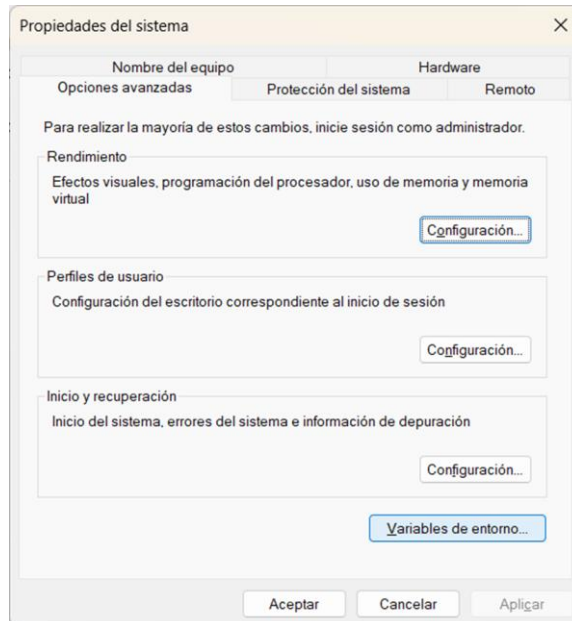
- Teclado y ratón.
- Procesador Intel core i3
- 4 GB de Ram.
- Una cámara con un mínimo de 10 megapíxeles

## Instalación

- Instalar Python 3.10 o superior
  - [Download Python | Python.org](https://www.python.org/downloads/)
    - asegurar que este la ruta de instalación de Python en la variable path dentro de las variables de entorno



- 
- Dar click en “Variables de entorno”



- Variables de entorno

Variables de usuario para kivop

Variable	Valor
OneDrive	C:\Users\kivop\OneDrive
OneDriveConsumer	C:\Users\kivop\OneDrive
Path	C:\Users\kivop\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\kivop\AppData\Local\Temp
TMP	C:\Users\kivop\AppData\Local\Temp

Nuevo...

Editar...

Eliminar

Variables del sistema

Variable	Valor
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	8
OnlineServices	Online Services
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System3...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

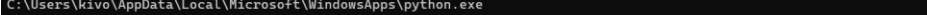
Nueva...

Editar...

Eliminar

Aceptar

Cancelar

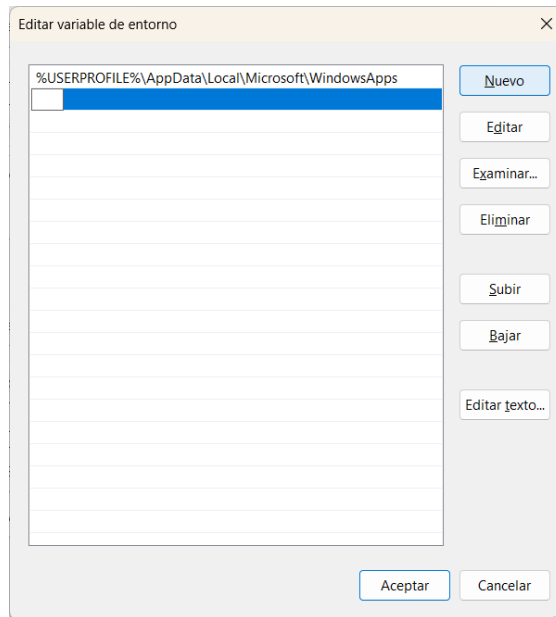
- 
- ```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22631.4317]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\kivo>where python

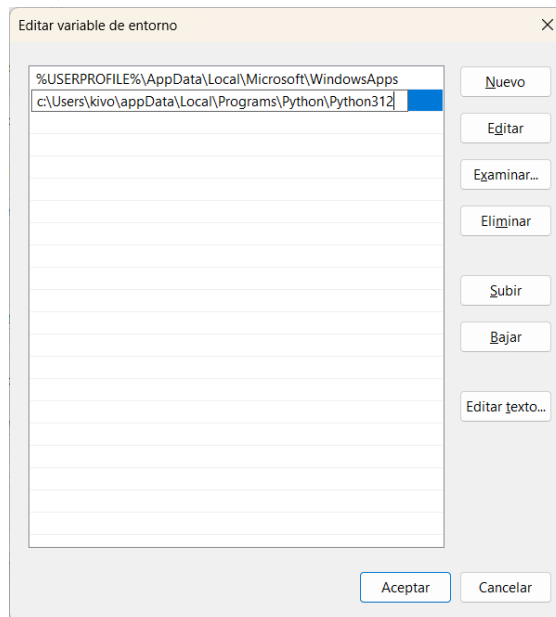
C:\Users\kivo\AppData\Local\Programs\Python\Python312\python.exe
C:\Users\kivo\AppData\Local\Microsoft\WindowsApps\python.exe

C:\Users\kivo>
```

- Dar click en “nuevo”



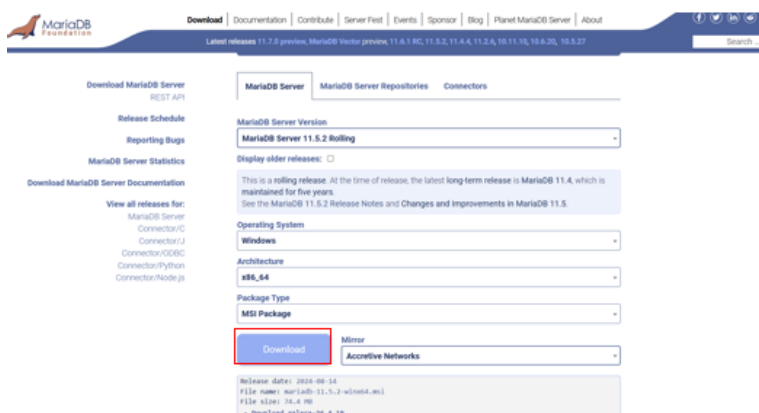
- 
- Pegar la ruta previamente copiada y dar click en “aceptar” a las 3 ventanas



- Instalar Maria DB
  - [MariaDB Foundation - MariaDB.org](https://mariadb.org/)



- no cambiar tipo de paquete y el apartado de mirror

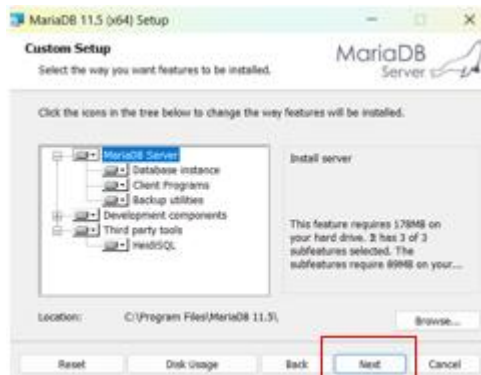


- ejecutar el instalador

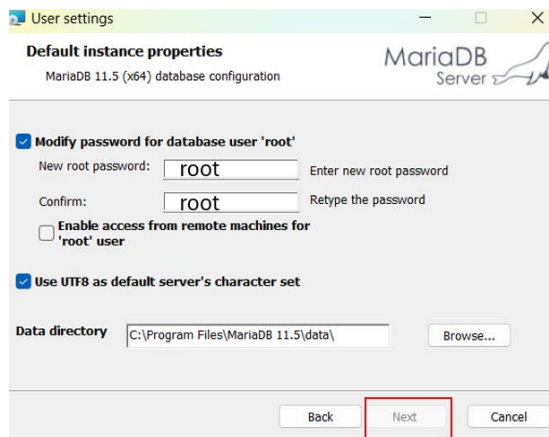


- En caso de instalar en otra ruta cambiar, si no instalar en la ruta predefinida





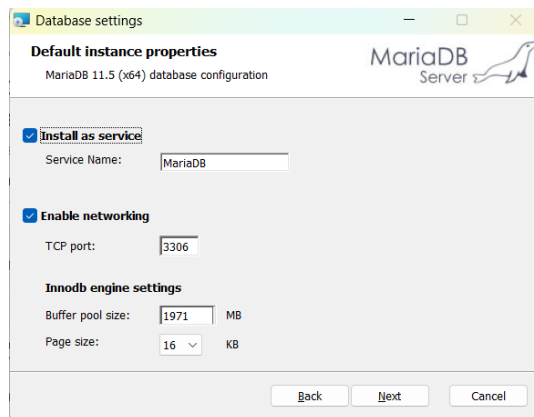
- Colocar la contraseña “root” y activar “Use UTF8 as default server’s character set



**NOTA:** asegurarse que el puerto 3306 no esté siendo ocupado por otro gestor de mysql

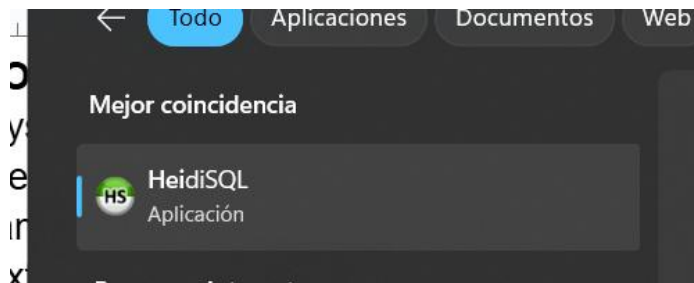
Ejemplo si se tiene instalado XAMPP y no se quiere borrar, momentáneamente cambias el puerto, inicias XAMPP, en SQL, click en config>my.ini , en el editor de texto en las secciones [client] y [mysqld] cambias el puerto, guardas el editor de texto y reinicias XAMPP

No mover nada de la siguiente ventana, si el puerto esta ocupado no se instalara

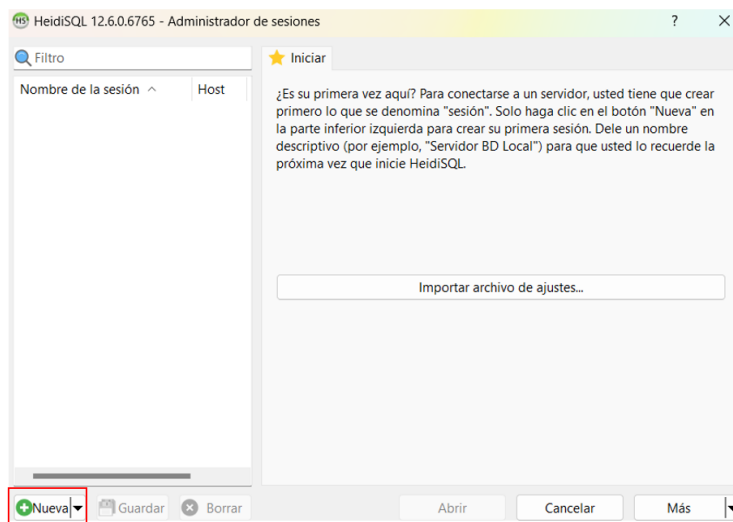


- IDE que ejecute python
  - recomendado Visual Studio Code
    - [Download Visual Studio Code - Mac, Linux, Windows](#)
- Instalar las siguientes bibliotecas dentro de la consola
  - instalar flask
    - pip install flask
  - instalar flask-sql
    - pip install flask-mysqldb
  - instalar openCV
    - pip install opencv-python
    - pip install opencv-contrib-python
  - instalar imutils
    - pip install imutils

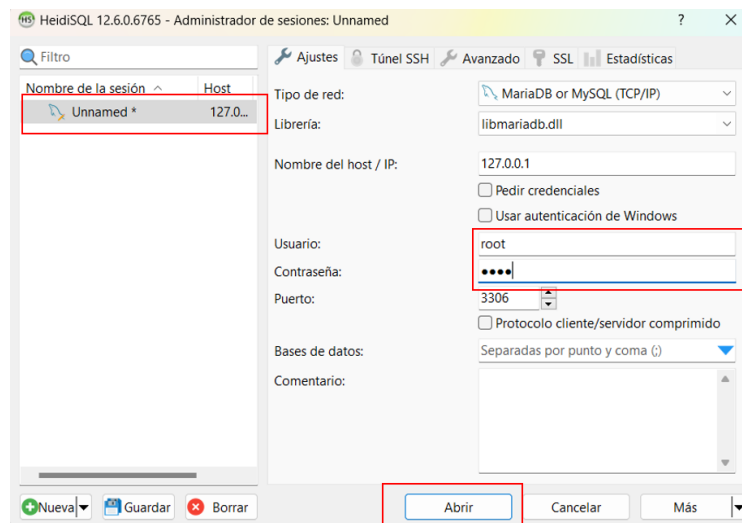
## Creación de la Base de Datos



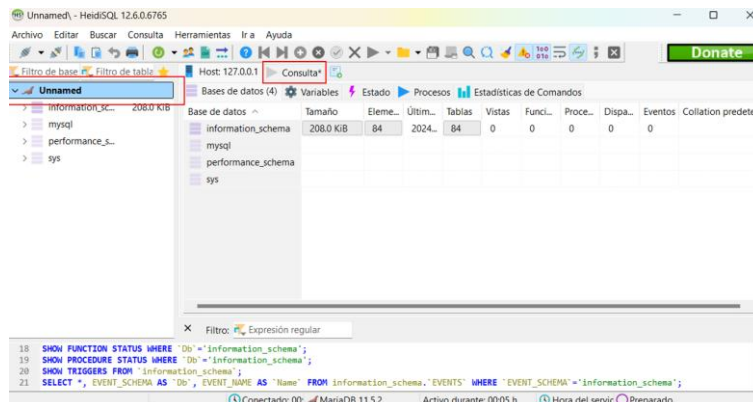
Dar click en “nueva



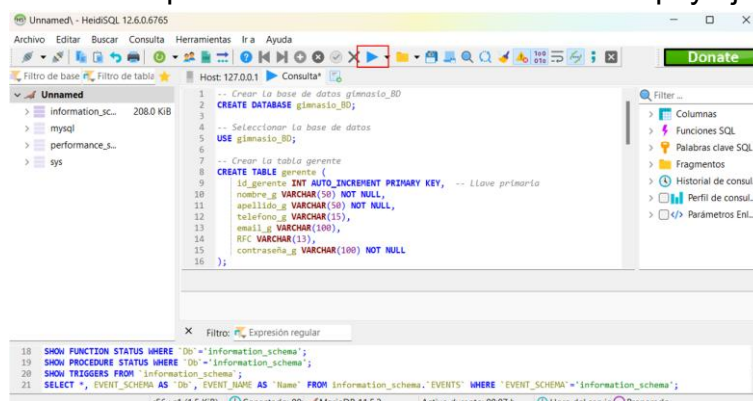
El nombre de la sesión puede ser el de su preferencia, es para identificar la sesión  
Asignar usuario y contraseña “root”  
Abrir



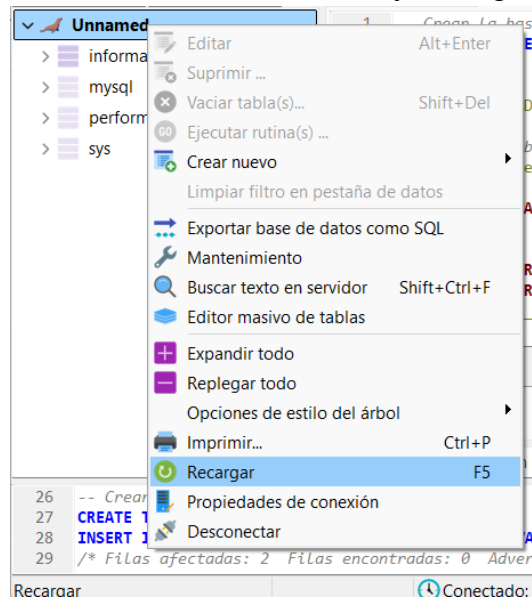
Seleccionando el la sesión en este caso “Unnamed” dar click en consulta



Pegar la base de datos que está en el archivo “database.sql” y ejecutar



Click derecho en la sesión y recargar



Si desea agregar nuevas contraseñas de admin hacerlo directamente a la tabla de admin con la instrucción

```
INSERT INTO admin (id_admin, contraseña_admin)
VALUES (1, '123'); #modificar a preferencia VALUES
```

## Diseño de arquitectura

### Arquitectura

La arquitectura de la aplicación se basa en el patrón MVC, que separa las responsabilidades del código para mejorar la organización y la mantenibilidad. En esta estructura, el Modelo representa la lógica de negocio y la interacción con la base de datos, gestionando las operaciones CRUD. Esto se maneja a través del archivo config.py, que configura la conexión a la base de datos, y los métodos dentro de app.py, que gestionan las interacciones con los datos.

La **Vista** se compone de los archivos HTML ubicados en la carpeta templates, donde se define la interfaz de usuario. Estas vistas permiten la presentación de la información al usuario y la captura de datos a través de formularios. Las vistas son renderizadas dinámicamente por Flask, lo que facilita la creación de contenido dinámico.

El **Controlador**, que se encuentra en app.py, actúa como intermediario entre el modelo y la vista. Se encarga de recibir las solicitudes del cliente, procesar la lógica de negocio y devolver las respuestas adecuadas, ya sea renderizando una vista o redirigiendo a otra página.

Además, la carpeta static almacena los archivos CSS, imágenes y otros recursos estáticos, lo que permite una mejor gestión de los activos del proyecto.

## Tecnologías

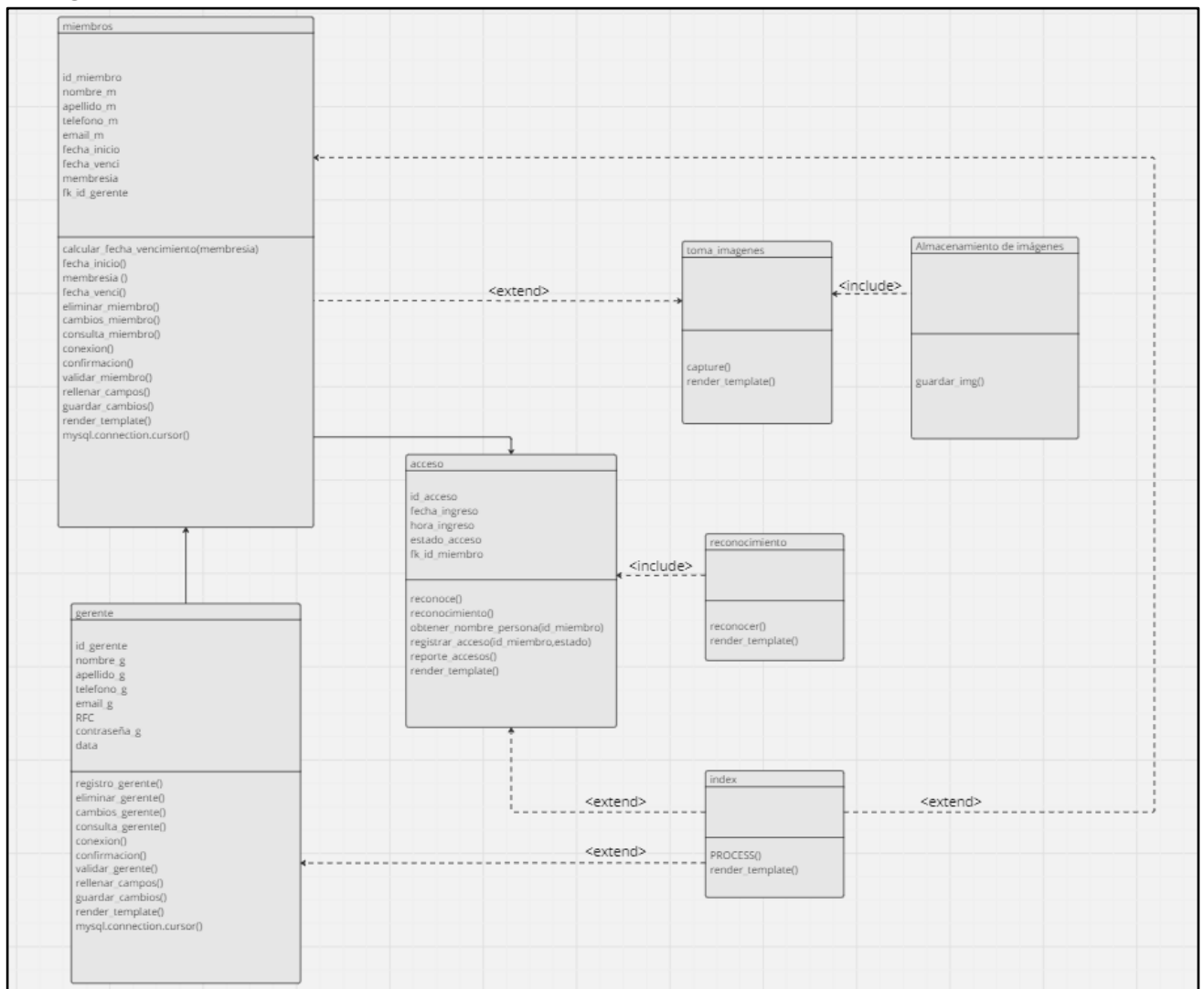
- python
- openCV
- flask
- html
- css
- Bootstrap
- JavaScript
- MySQL

## Estructura del proyecto

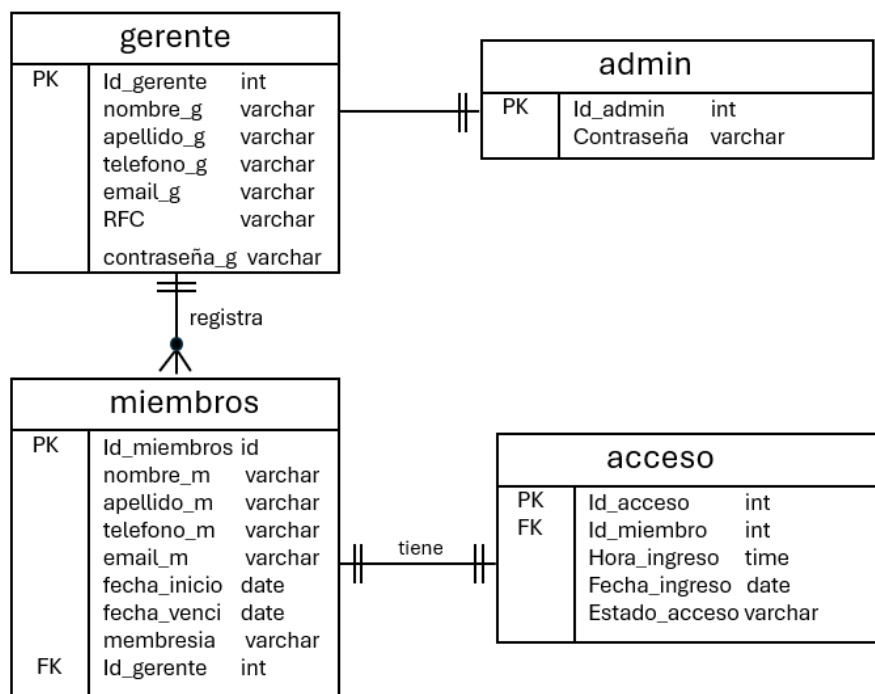
### Prototipo Gimnasio/

- |— \_\_pycache\_\_/
- |— miembros/
- |— static/
  - |— css/
  - |— fronts/
  - |— js/
  - |— Logo.jpg
- |— Templates/
  - |— acceso.html
  - |— admin\_gerente.html
  - |— admin\_miem.html
  - |— cambios\_g.html
  - |— cambios\_m.html
  - |— index.html/
  - |— reconocimiento.html /
  - |— registro\_g.html
  - |— registro .html
  - |— Toma\_imagenes.html
- |— app.py
- |— config.py
- |— Database.sql
- |— Haarcascade\_frontalface\_default.xml
- |— modeloLBPHFace.xml
- |— README.md

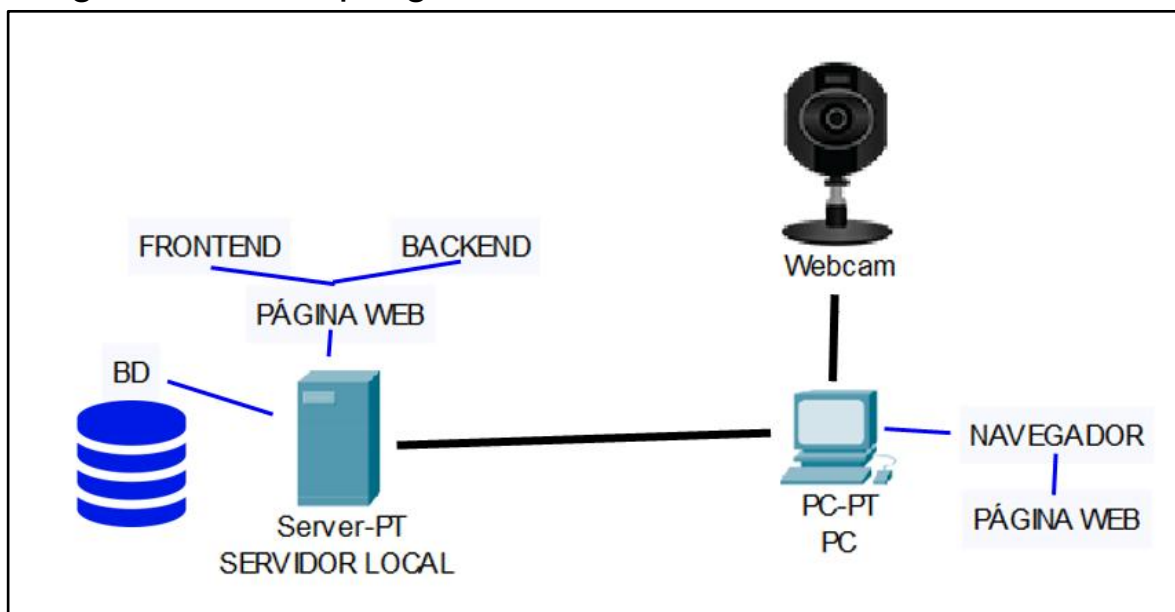
## Diagrama de clases



## Diagrama entidad Relación



## Diagrama de Despliegue





[illegible]