



INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y
CIENCIAS SOCIALES Y ADMINISTRATIVAS

Prototipo de Sistema de Control de Acceso para Gimnasio con Reconocimiento Facial

P R E S E N T A

Mora Flores Omar

Lugo Mora José Yael

Ramírez Ramos Ángel Jair

Ramírez Blas Luis Ángel

Trejo Monroy Stefany



Profesor

MTRO. Martínez Vázquez Gustavo



CIUDAD DE MÉXICO, MÉXICO

2025

b) Contenido

b) Contenido.....	2
c) Objetivo	3
1.1 Objetivo General.....	3
1.2 Objetivos Particulares	3
d) Hipótesis	4
e) Justificación.....	4
f) Marco Teórico	4
4.1 Planteamiento del Problema	4
4.2 Alcance del Proyecto	5
4.3 Referencia con otros proyectos o iniciativas	5
g) Marco Metodológico Documental	6
5.1 Propuesta de solución:.....	6
5.2 Relación costo-beneficio:	6
5.3 Herramientas, técnicas y formas de trabajo	6
5.4 Plan de trabajo:.....	6
h) Conclusiones.....	11
6.1 Individual	11
6.2 Colectiva.....	14
i) Anexos.....	15
7.1 Backlog	15
7.2 Prototipos y/o Wireframes	15
7.3 Diagramas	23
7.3.1 Procesos.....	23
7.3.2 Clases	24
7.3.3 Actividades	26
7.4 Diccionario de Datos	27
7.5 Manual Técnico	27
7.6 Manual de Usuario.....	27
7.7 Manual de Identidad	27
7.8 Arquitectura de software.....	27

7.8.1. Modelo Arquitectónico General	27
7.8.2. Arquitectura del Backend	29
7.8.3. Arquitectura del Frontend	30
7.8.4. Arquitectura de Datos y Persistencia	30
7.9 Carta de Presentación de Proyecto --	¡Error! Marcador no definido.
7.10 Carta Compromiso --	¡Error! Marcador no definido.
7.11 Matriz de Pruebas	31
7.12 Plan de recuperación	31

c) Objetivo

1.1 Objetivo General

Diseñar y construir un prototipo de sistema de control de acceso para gimnasios. Este sistema empleará la tecnología de reconocimiento facial para optimizar y modernizar los procedimientos de registro y acceso de los usuarios. El proyecto se materializará en una plataforma web que integrará tres módulos esenciales: el primero, para la captura eficiente de imágenes de los usuarios; el segundo, para el entrenamiento y ajuste de un modelo de reconocimiento facial; y el tercero, para la automatización del proceso de autenticación de usuarios al ingresar al gimnasio. Se espera que este sistema proporcione una alternativa más ágil, precisa y segura a los métodos convencionales de control de acceso, basados en registros manuales o tarjetas de identificación.

1.2 Objetivos Particulares

- Implementar un formulario de registro de miembros y gerentes: Se diseñará e implementará un formulario web que permita a los usuarios (miembros y gerentes) ingresar sus datos personales y de contacto. Este formulario almacenará la información de manera estructurada en una base de datos MySQL, garantizando la integridad y disponibilidad de los datos.
- Desarrollar un módulo de captura de imágenes: Se desarrollará un módulo de software que permita capturar imágenes de los rostros de los usuarios. Este módulo utilizará una cámara web y algoritmos de procesamiento de imágenes para obtener fotografías de alta calidad que serán almacenadas en una base de datos de rostros. Esta base de datos será utilizada posteriormente para el entrenamiento del modelo de reconocimiento facial.

- Integrar y entrenar un modelo LBPHFaceRecognizer: Se integrará y configurará el algoritmo LBPHFaceRecognizer (Local Binary Patterns Histograms) para el reconocimiento facial. Se utilizará la base de datos de rostros generada en el módulo anterior para entrenar el modelo, ajustando sus parámetros para lograr una identificación confiable y precisa de los usuarios.
- Crear una interfaz de acceso: Se diseñará y desarrollará una interfaz web que permita registrar de forma automática y eficiente la entrada y salida de los miembros del gimnasio. Esta interfaz utilizará el modelo de reconocimiento facial entrenado para identificar a los usuarios y registrar su hora de acceso en tiempo real.

d) Hipótesis

Si se implementa un sistema de autenticación biométrica mediante reconocimiento facial en el control de acceso de un gimnasio, entonces se reducirá el tiempo de validación de ingreso en al menos un 70% y se disminuirán los errores de registro manual en un 90%, en comparación con métodos tradicionales basados en papel o credenciales físicas.

- La precisión de reconocimiento facial en entornos de baja iluminación superará el 90% tras optimizaciones de preprocesamiento de imagen.
- La adopción del sistema por parte de los miembros del gimnasio mostrará un nivel de satisfacción superior al 85% en encuestas de usabilidad.
- El sistema permitirá detectar y bloquear intentos de acceso no autorizados, logrando reducir incidentes de seguridad en un 95%

e) Justificación

El sistema propuesto busca modernizar los métodos de control de acceso y registro de miembros, que actualmente se basan en papel o Excel, ofreciendo una alternativa más eficiente y ágil. La implementación de un sistema biométrico de autenticación agilizará el proceso de registro y control de acceso, mejorando la experiencia del cliente y la eficiencia operativa del gimnasio.

f) Marco Teórico

4.1 Planteamiento del Problema

Los gimnasios tradicionales utilizan registros manuales que generan cuellos de botella en horas pico y errores humanos al anotar datos. Esto repercute en una mala experiencia del

cliente y dificulta el control de asistencia, así como el análisis estadístico de uso de las instalaciones.

4.2 Alcance del Proyecto

- Registro de miembros y gerentes a través de formularios web interactivos
- Captura y almacenamiento digital de hasta 300 imágenes faciales por usuario, destinadas al entrenamiento del modelo de reconocimiento.
- Integración y entrenamiento de un modelo LBPH (Local Binary Patterns Histograms) para la autenticación facial confiable de los usuarios.
- Desarrollo de una interfaz de acceso en tiempo real que automatiza el registro de la entrada y salida de los miembros.
- Implementación de una herramienta de consulta para generar reportes detallados de acceso.

4.3 Referencia con otros proyectos o iniciativas

- Vaca Piña y Rivera Rodríguez desarrollaron un sistema de control de acceso basado en reconocimiento facial para la Academia Titanes Cuenca, demostrando viabilidad académica en un entorno educativo
- Jacinto et al. presentaron un prototipo en el "International Congress on Innovation and Sustainable" que emplea Haarcascade y redes neuronales para mejorar la precisión de reconocimiento facial.
- Compañías como Brivo ofrecen soluciones comerciales de acceso biométrico en gimnasios con interfaces en la nube y supervisión remota, evidenciando la tendencia de adopción de sistemas basados en la web.
- **OpenCV:** Es una biblioteca de visión artificial de código abierto ampliamente utilizada en proyectos de reconocimiento facial. Proporciona herramientas y algoritmos para el procesamiento de imágenes, la detección de rostros y el entrenamiento de modelos de reconocimiento.
- **FaceNet:** Desarrollado por Google, FaceNet es un modelo de red neuronal que aprende incrustaciones faciales, representaciones numéricas de los rostros que facilitan la comparación y el reconocimiento. Se ha utilizado en diversas aplicaciones de control de acceso y autenticación.

- **Amazon Rekognition:** Es un servicio de reconocimiento de imágenes y video basado en la nube que ofrece capacidades de reconocimiento facial. Puede utilizarse para verificar la identidad de los usuarios, controlar el acceso a instalaciones y personalizar experiencias.
- **Microsoft Azure Face API:** Similar a Amazon Rekognition, Azure Face API proporciona servicios de reconocimiento facial en la nube con diversas funcionalidades, como la detección de rostros, la identificación y la verificación.

g) Marco Metodológico Documental

5.1 Propuesta de solución:

Emplear una metodología de investigación documental combinada con diseño e implementación ágil (Scrum) para desarrollar un prototipo modular en Flask y Python. El sistema constará de los siguientes módulos: registro de usuarios, captura y preprocesamiento de imágenes, entrenamiento del modelo LBPHFaceRecognizer, reconocimiento en tiempo real y generación de reportes. Cada módulo se documentará y validará mediante pruebas unitarias y de integración.

5.2 Relación costo-beneficio:

Costos: Hardware mínimo (PC con cámara HD, 4 GB RAM), licencias de software libres (Python, Flask, OpenCV, MySQL), tiempo de desarrollo estimado en 8 semanas por un equipo de 5 integrantes.

Beneficios: Reducción de tiempos de acceso y errores manuales (70% y 90% respectivamente), aumento en la satisfacción de usuarios (>85%), ahorro en materiales de control (papel, fichas), mayor seguridad y trazabilidad de accesos.

5.3 Herramientas, técnicas y formas de trabajo:

- Herramientas de desarrollo: Python 3.11, Flask, OpenCV, MySQL, Git/GitHub, Visual Studio Code.
- Técnicas de investigación: Revisión de literatura académica, benchmark de soluciones existentes, análisis de requisitos mediante encuestas y entrevistas.
- Formas de trabajo: Scrum con sprints de dos semanas, reuniones diarias de seguimiento (daily stand-ups), revisiones de sprint y retrospectivas. Uso de tableros Kanban en Trello para gestión de tareas.

5.4 Plan de trabajo:

Metodología Ágil: Scrum

El desarrollo del prototipo se gestionará utilizando la metodología ágil Scrum. El trabajo se organizará en iteraciones de duración fija, denominadas Sprints. Cada Sprint tendrá como objetivo entregar una porción funcional del producto.

- **Roles de Scrum:**

- **Product Owner:** (Mora Flores Omar) Responsable de definir la funcionalidad del producto y priorizar las necesidades del negocio.
- **Scrum Master:** (Mora Flores Omar, Ramírez Ramos Ángel Jair) Facilitador del equipo Scrum, encargado de eliminar impedimentos y asegurar que se sigan las prácticas de Scrum.
- **Equipo de Desarrollo:** (Lugo Mora José Yael, Ramirez Blas Luis Angel, Trejo Monroy Stefany) Responsables de desarrollar el producto.

- **Eventos de Scrum:**

- **Sprint Planning:** Se realiza al inicio de cada Sprint para definir el trabajo a realizar.
- **Daily Scrum:** Reunión diaria de 15 minutos para sincronizar actividades y detectar impedimentos.
- **Sprint Review:** Demostración del incremento de producto al final del Sprint.
- **Sprint Retrospective:** Reflexión sobre el Sprint para identificar mejoras.

- **Artefactos de Scrum:**

- **Product Backlog:** Lista priorizada de funcionalidades.
- **Sprint Backlog:** Lista de tareas a realizar en el Sprint.
- **Incremento:** Funcionalidad entregable al final del Sprint.

Semana	Actividad Principal	Duración	Fecha de Inicio	Fecha de Fin
1-2	Levantamiento de requisitos, investigación documental y definición del alcance.	2 semanas	2023-10-23	2023-11-03

3	Diseño de la arquitectura, modelado de la base de datos y prototipos de interfaz (wireframes).	1 semana	2023-11-06	2023-11-10
4-5	Implementación de módulos de registro y captura de imágenes; pruebas unitarias de preprocesamiento.	2 semanas	2023-11-13	2023-11-24
6	Desarrollo del módulo de entrenamiento del modelo y pruebas de precisión en distintos escenarios.	1 semana	2023-11-27	2023-12-01
7	Implementación del reconocimiento en tiempo real, integración de todos los módulos y pruebas E2E.	1 semana	2023-12-04	2023-12-08
8	Generación de reportes, sistema de notificaciones, pruebas de usabilidad y ajustes finales.	1 semana	2023-12-11	2023-12-15
9	Elaboración de documentación técnica y de usuario, preparación de la presentación del prototipo.	1 semana	2023-12-18	2023-12-22

Semana 1-2: Levantamiento de requisitos, investigación documental y definición del alcance.

- Recopilación de información sobre los procesos actuales de control de acceso en gimnasios.
- Entrevistas con administradores y usuarios de gimnasios para identificar necesidades y expectativas.

- Revisión de literatura académica y proyectos similares para identificar mejores prácticas y tecnologías relevantes.
- Análisis de los requisitos funcionales y no funcionales del sistema.
- Definición del alcance del proyecto, incluyendo las funcionalidades que se implementarán en el prototipo.
- Elaboración de un documento de especificación de requisitos.

Semana 3: Diseño de la arquitectura, modelado de la base de datos y prototipos de interfaz (wireframes).

- Diseño de la arquitectura del sistema, incluyendo la selección de componentes y tecnologías.
- Modelado de la base de datos MySQL, definiendo las tablas y relaciones necesarias para almacenar la información del sistema.
- Creación de prototipos de interfaz de usuario (wireframes) para las diferentes pantallas del sistema.
- Revisión y aprobación del diseño de la arquitectura, el modelo de la base de datos y los prototipos de interfaz.

Semana 4-5: Implementación de módulos de registro y captura de imágenes; pruebas unitarias de preprocesamiento.

- Desarrollo del módulo de registro de miembros y gerentes, incluyendo la creación de formularios web y la lógica para almacenar la información en la base de datos.
- Implementación del módulo de captura de imágenes, utilizando OpenCV y una cámara web para capturar fotografías de los rostros de los usuarios.
- Desarrollo de funciones para el preprocesamiento de las imágenes capturadas, incluyendo el ajuste de la iluminación, el recorte y el escalado.
- Creación de pruebas unitarias para verificar la correcta funcionalidad de los módulos de registro y captura de imágenes, así como de las funciones de preprocesamiento.

Semana 6: Desarrollo del módulo de entrenamiento del modelo y pruebas de precisión en distintos escenarios.

- Implementación del módulo de entrenamiento del modelo de reconocimiento facial, utilizando el algoritmo LBPHFaceRecognizer.

- Preparación de la base de datos de rostros capturadas en la etapa anterior para el entrenamiento del modelo.
- Entrenamiento del modelo con diferentes configuraciones y parámetros para optimizar su rendimiento.
- Diseño y ejecución de pruebas de precisión del modelo en distintos escenarios, incluyendo variaciones en la iluminación, la pose y la expresión facial.

Semana 7: Implementación del reconocimiento en tiempo real, integración de todos los módulos y pruebas E2E.

- Desarrollo de la interfaz de reconocimiento en tiempo real, utilizando la cámara web y el modelo entrenado para identificar a los usuarios al ingresar o salir del gimnasio.
- Integración de todos los módulos desarrollados en las etapas anteriores, incluyendo el registro, la captura de imágenes, el entrenamiento del modelo y el reconocimiento en tiempo real.
- Realización de pruebas de extremo a extremo (E2E) para verificar el correcto funcionamiento del sistema en su conjunto.

Semana 8: Generación de reportes, sistema de notificaciones, pruebas de usabilidad y ajustes finales.

- Desarrollo de la funcionalidad de generación de reportes, permitiendo a los administradores del gimnasio acceder a información detallada sobre la actividad de los usuarios.
- Implementación de un sistema de notificaciones para informar a los usuarios sobre eventos relevantes, como el registro de su acceso o posibles errores en el reconocimiento.
- Realización de pruebas de usabilidad con usuarios reales para evaluar la facilidad de uso y la experiencia del sistema.
- Realización de ajustes finales al sistema en función de los resultados de las pruebas de usabilidad y las observaciones del equipo de desarrollo.

Semana 9: Elaboración de documentación técnica y de usuario, preparación de la presentación del prototipo.

- Elaboración de la documentación técnica del sistema, incluyendo la descripción de la arquitectura, el diseño de la base de datos, el código fuente y las instrucciones de instalación y configuración.

- Creación de la documentación de usuario, incluyendo manuales y tutoriales que expliquen cómo utilizar el sistema.
- Preparación de la presentación del prototipo, incluyendo la creación de diapositivas y la práctica de la demostración.

h) Conclusiones

6.1 Individual

Mora Flores Omar

El sistema cumple con todo lo que nos propusimos, tiene funciones avanzadas como reconocimiento facial y envío de QR por WhatsApp, y quedó bastante sólido. Aún hay espacio para mejorar, sobre todo en seguridad y optimización, pero lo que hicimos ya es una base muy buena para algo más grande en el futuro.



Participar en este proyecto me permitió aplicar conocimientos técnicos en una solución real. Desde el desarrollo hasta la integración del reconocimiento facial, pude fortalecer mis habilidades en backend, arquitectura de software y manejo de datos. Aprendí a trabajar bajo metodologías ágiles y a colaborar efectivamente con mi equipo para superar retos técnicos y de integración.





Trejo Monroy Stefany

Este proyecto me permitio poner en práctica conocimientos previos adquiridos en otras materias como ingenieria de requerimientos, de diseño, programación, y a su vez aplicar lo aprendido en esta materia, tambien reforzó mis habilidades en modelado de datos y consultas SQL aplicadas a un sistema real.



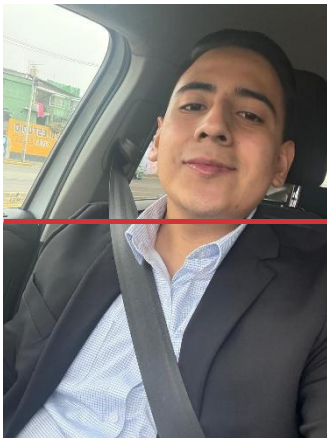
Lugo Mora José Yael

Desarrollar este proyecto fue una experiencia muy enriquecedora, ya que pude aplicar los conocimientos adquiridos en clases para crear un sistema funcional. Me permitió entender cómo se puede mejorar un proceso cotidiano, como el control de acceso a un gimnasio, utilizando la tecnología. Además, me ayudó a mejorar mi organización y mi capacidad para resolver problemas.



Ramirez Ramos Angel Jair

Este proyecto fue una experiencia enriquecedora que me permitió aplicar conocimientos adquiridos en distintas materias para desarrollar un sistema funcional. Fortalecí habilidades técnicas como el modelado de datos, consultas SQL, desarrollo backend e integración de reconocimiento facial. Además, mejoré mi organización, resolución de problemas y trabajo en equipo bajo metodologías ágiles.



6.2 Colectiva

Como equipo, este proyecto fue todo un reto, pero también una gran oportunidad para aplicar lo que hemos aprendido y enfrentarnos a problemas reales. Cada quien aportó desde su rol, y al final logramos construir un sistema funcional que cumple con lo que nos propusimos: facilitar la gestión de un gimnasio de forma segura y práctica.

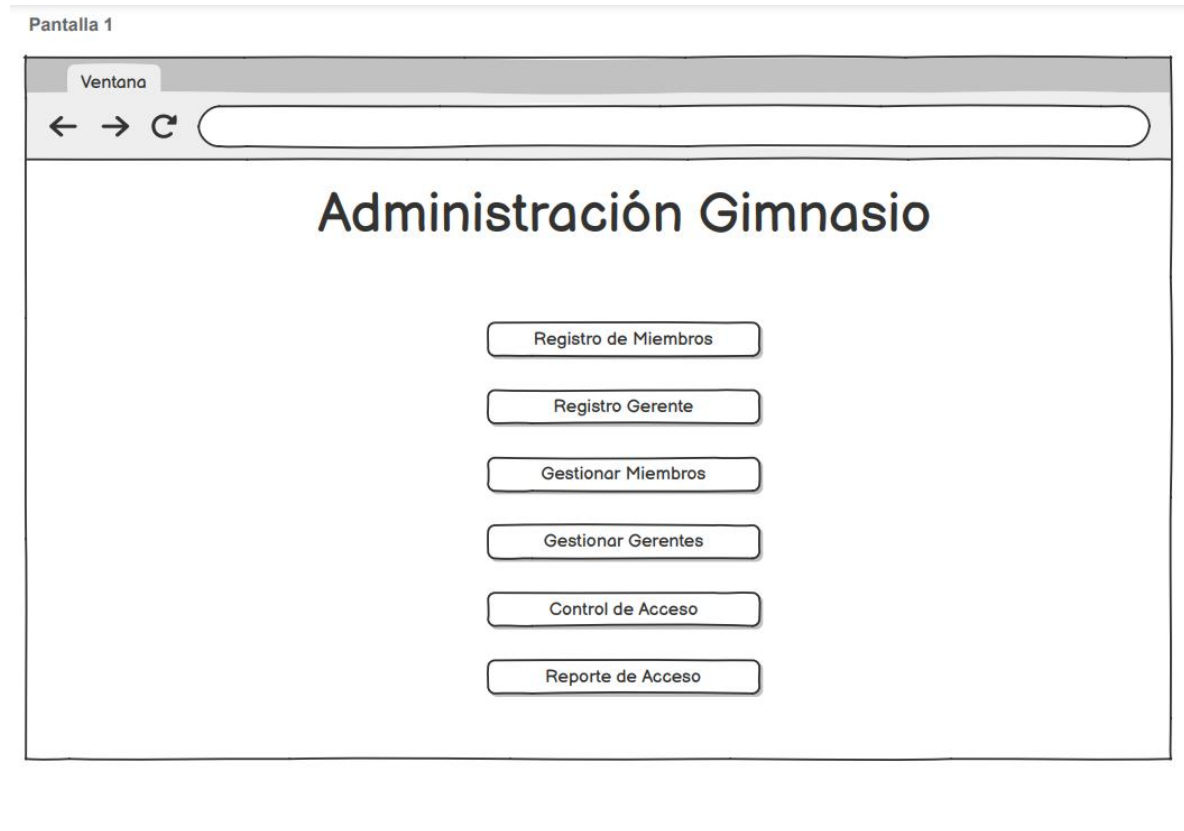
Nos quedamos con muchos aprendizajes técnicos, pero también con la satisfacción de haber trabajado en algo completo, útil y que realmente podría usarse. Sabemos que aún se puede mejorar, pero estamos orgullosos del resultado y de cómo crecimos en el proceso.

i) Anexos

7.1 Backlog

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/BAC KLOG.xlsx

7.2 Prototipos y/o Wireframes



Pantalla 2

Ventana

←

→

↻

Registro de Nuevo Miembro

volver

Nombre

Apellido

Telefono

Correo

Membresia

1 Mes

3 Meses

6 Meses

Registrar

Pantalla 3

Ventana

←

→

↻

Registro de Nuevo Miembro

volver

Nombre

Apellido

Telefono

Correo

Membresia

Registrar

Validar Gerente

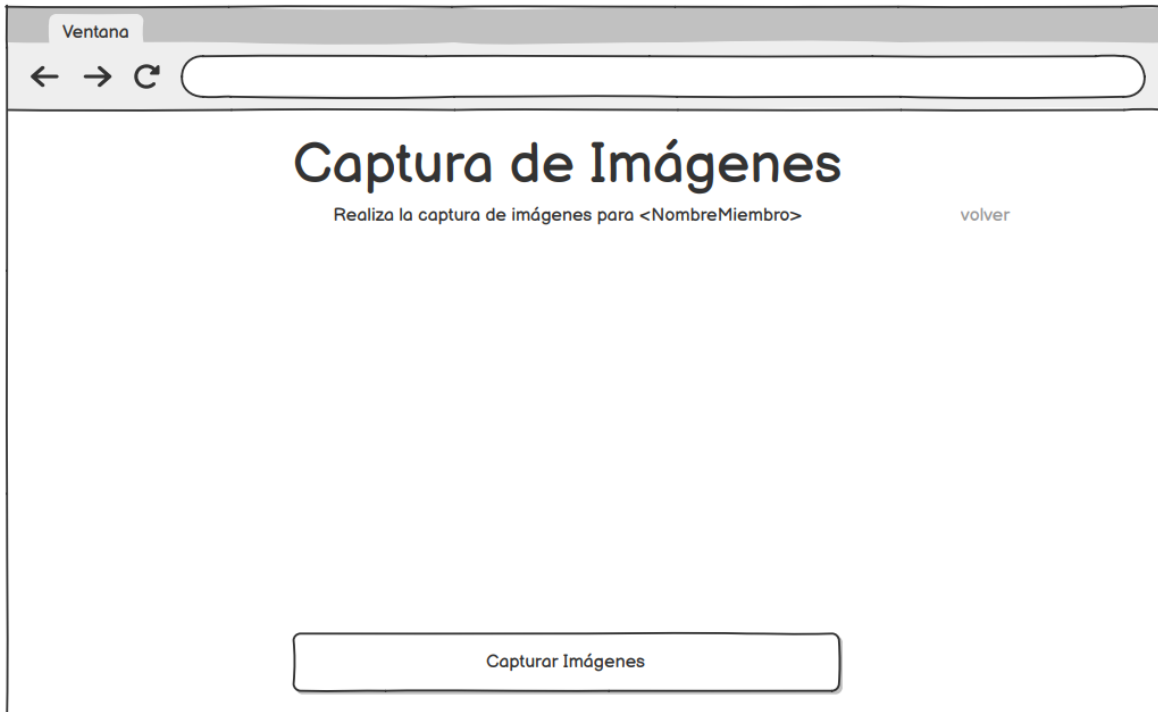
Id del Gerente

Contraseña

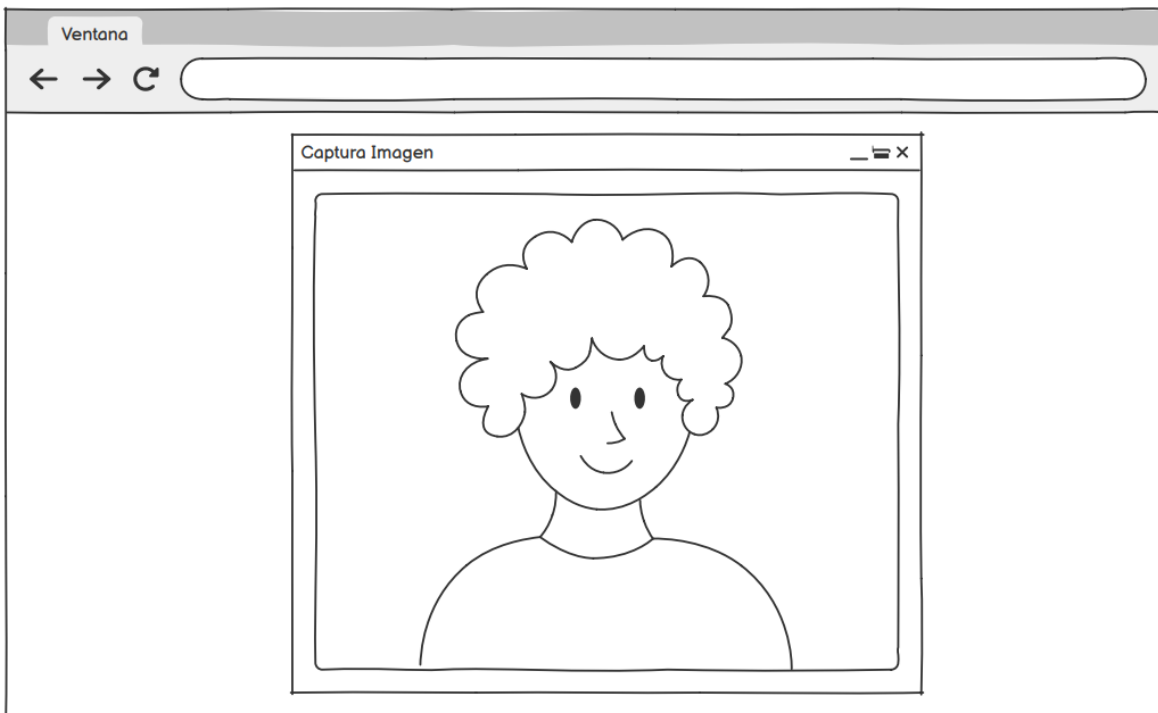
Cancelar

Confirmar

Pantalla 4



Pantalla 5



Ventana

←

→

↺

Registro de Gerente

volver

Nombre

Apellido

Telefono

Correo

RFC

Contraseña

Confirmar contraseña

Registrar

[illegible]

Pantalla 8

Ventana

←

→

↺

Cambios de Datos Miembros

volver

Nombre

Juan

Apellido

Perez

Telefono

5580880800

Correo

example@mail.com

Fecha de Vencimiento

2024-12-31

Duracion de Membresia

1 Mes

3 Meses

6 Meses

Actualizar

Pantalla 8

Ventana

←

→

↺

Cambios de Datos Miembros

volver

Nombre

Juan

Apellido

Perez

Telefono

5580880800

Correo

example@mail.com

Fecha de Vencimiento

2024-12-31

Duracion de Membresia

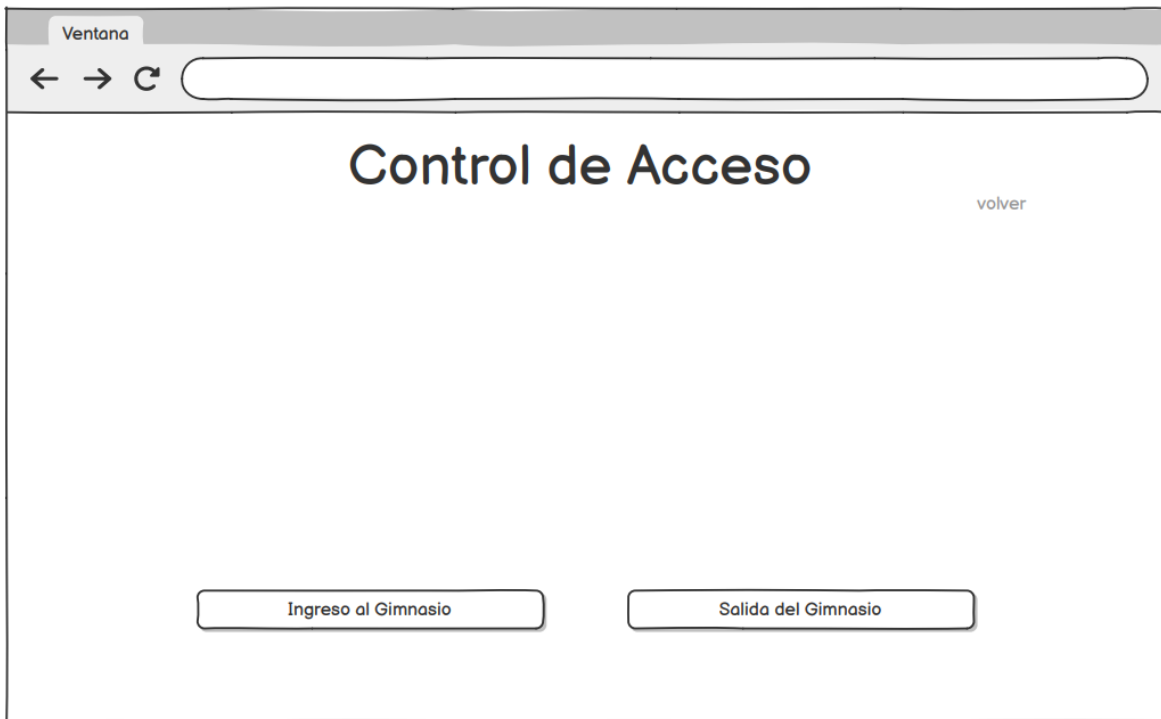
1 Mes

3 Meses

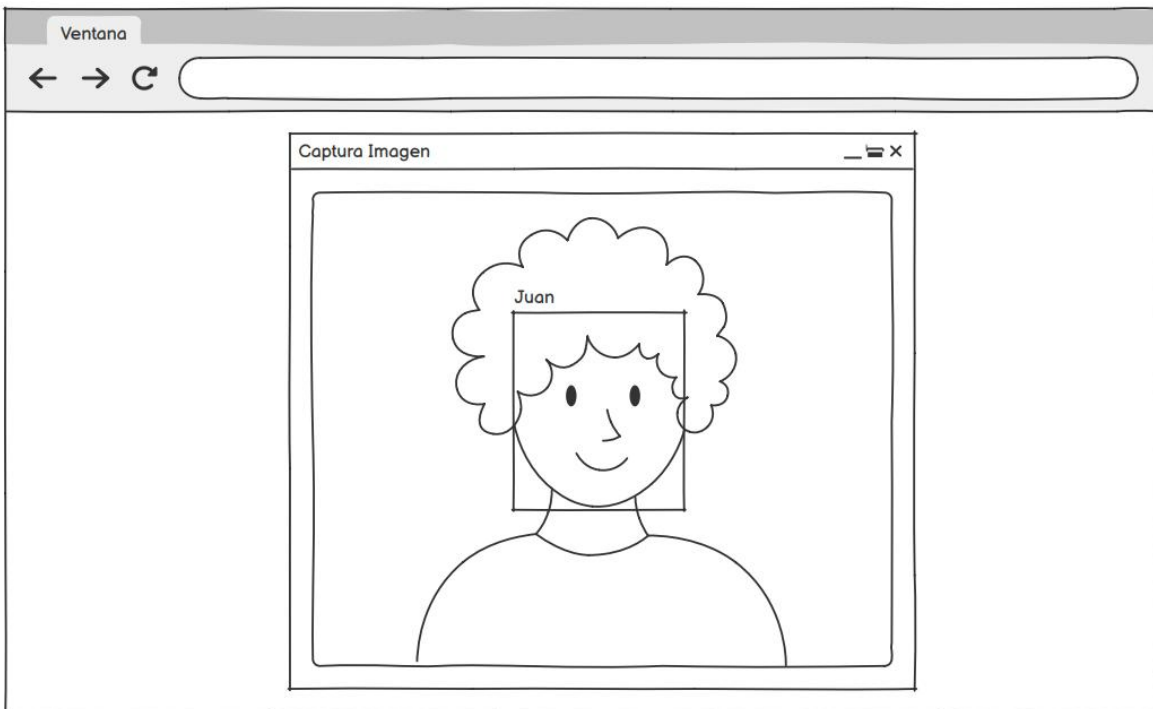
6 Meses

Actualizar

Pantalla 10



Pantalla 11



Ventana

←

→

C

Reporte de Acceso

volver

Acesos

No. Acceso	Fecha de Ingreso	Hora de Ingreso	Estado de Acceso	ID Miembro	Nombre del Miembro
1	2024-08-08	20:12:00	Ingreso	10	Kevin Pimmental
2	2024-08-08	20:59:00	Salida	10	Kevin Pimmental
3	2024-08-11	20:01:00	Ingreso	10	Kevin Pimmental
4	2024-08-11	21:08:00	Salida	10	Kevin Pimmental

[illegible]

Pantalla 14

Ventana

←

→

↺

Cambios de Datos de Gerentes

volver

Nombre

Juan

Apellido

Perez

Telefono

5580880800

Correo

example@mail.com

RFC

EXAM913

Actualizar

Pantalla 15

Ventana

←

→

↺

Registrar Nuevo Empleado

Inicio

Confirmar Eliminacion

Esta seguro de que desea eliminar este registro?

Aceptar

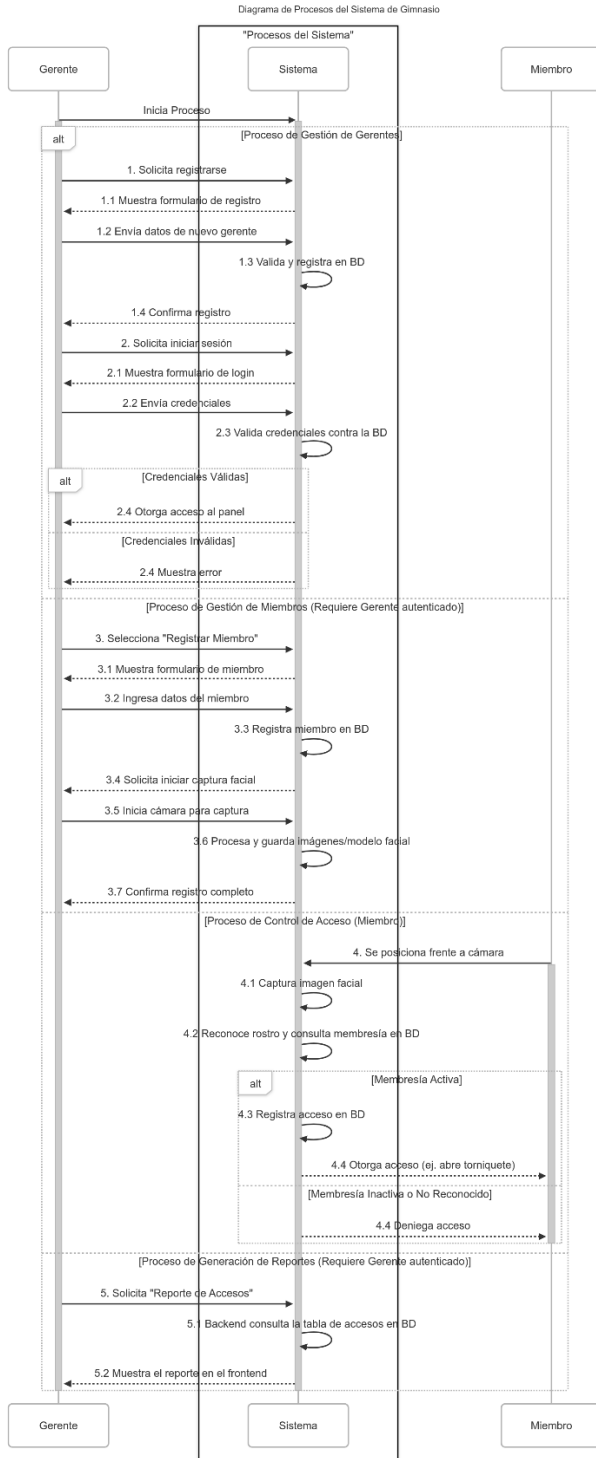
Cancelar

Administracion de Empleados

	Id	Nombre	Apellido	Telefono	Correo	Fecha de Vencimiento	Estado	Membresia	Accion
1	Kevin	Pimmentel	5544889955	kevin@mail.com	2024-06-01	2024-09-01	vigente	Cambios	
2	Zabdiel	Reyes	5511880088	zabdiel@mail.com	2024-06-01	2024-12-01	vigente	Eliminar	
3	Fernanda	Gomez	5544892314	fernanda@mail.com	2024-06-01	2024-12-01	vigente	Cambios	
								Eliminar	

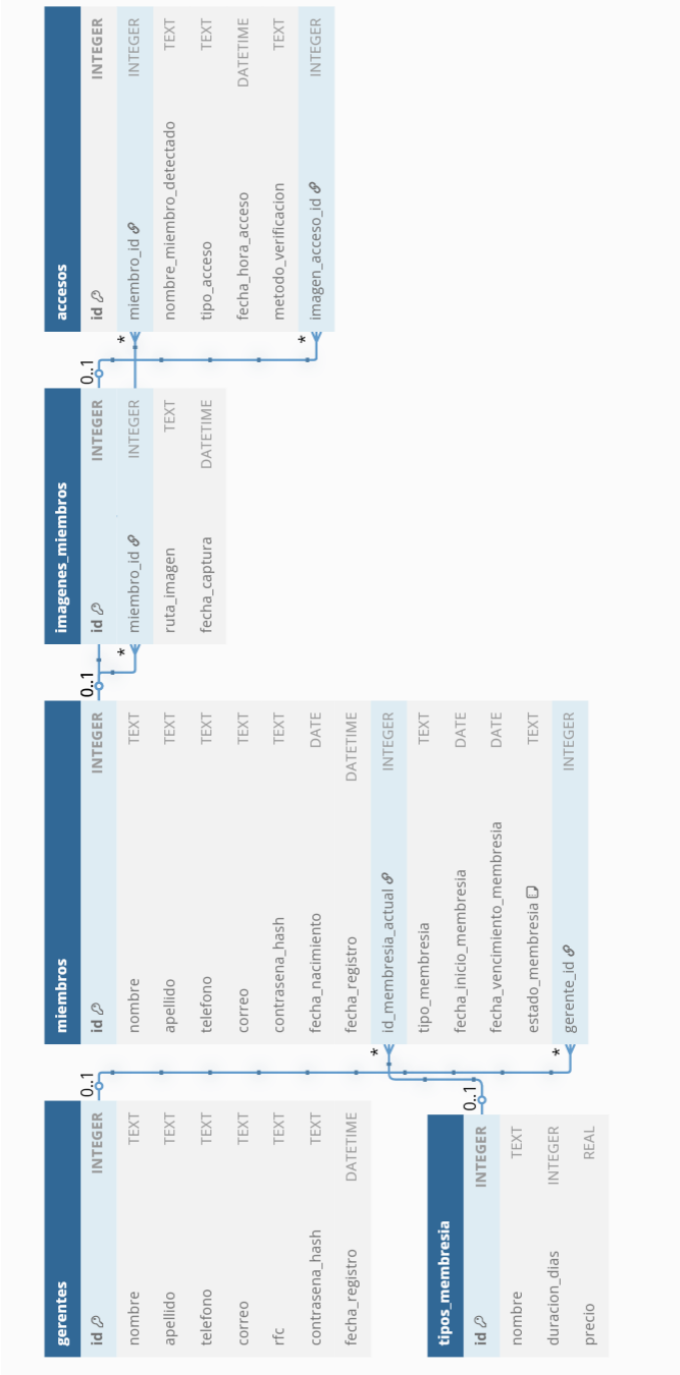
7.3 Diagramas

7.3.1 Procesos

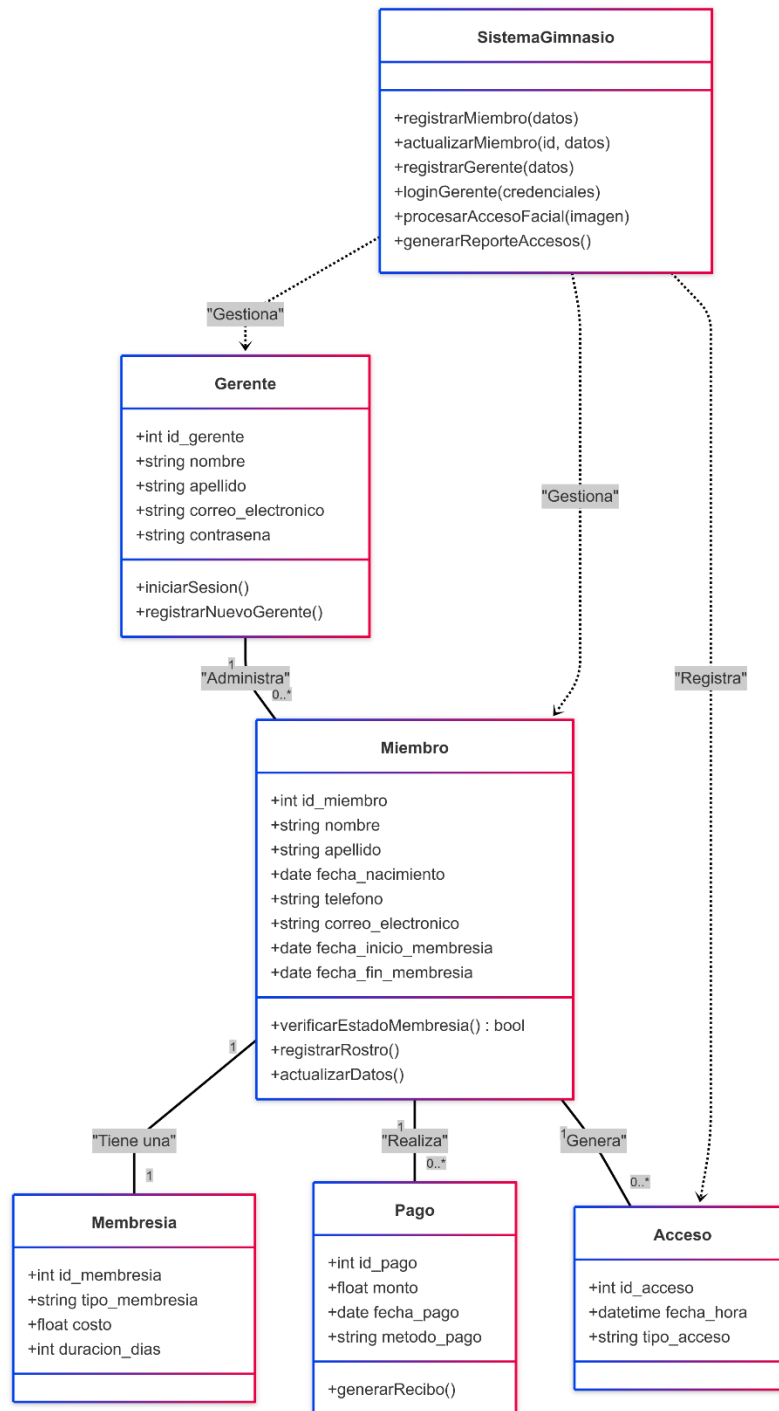


7.3.2 Clases

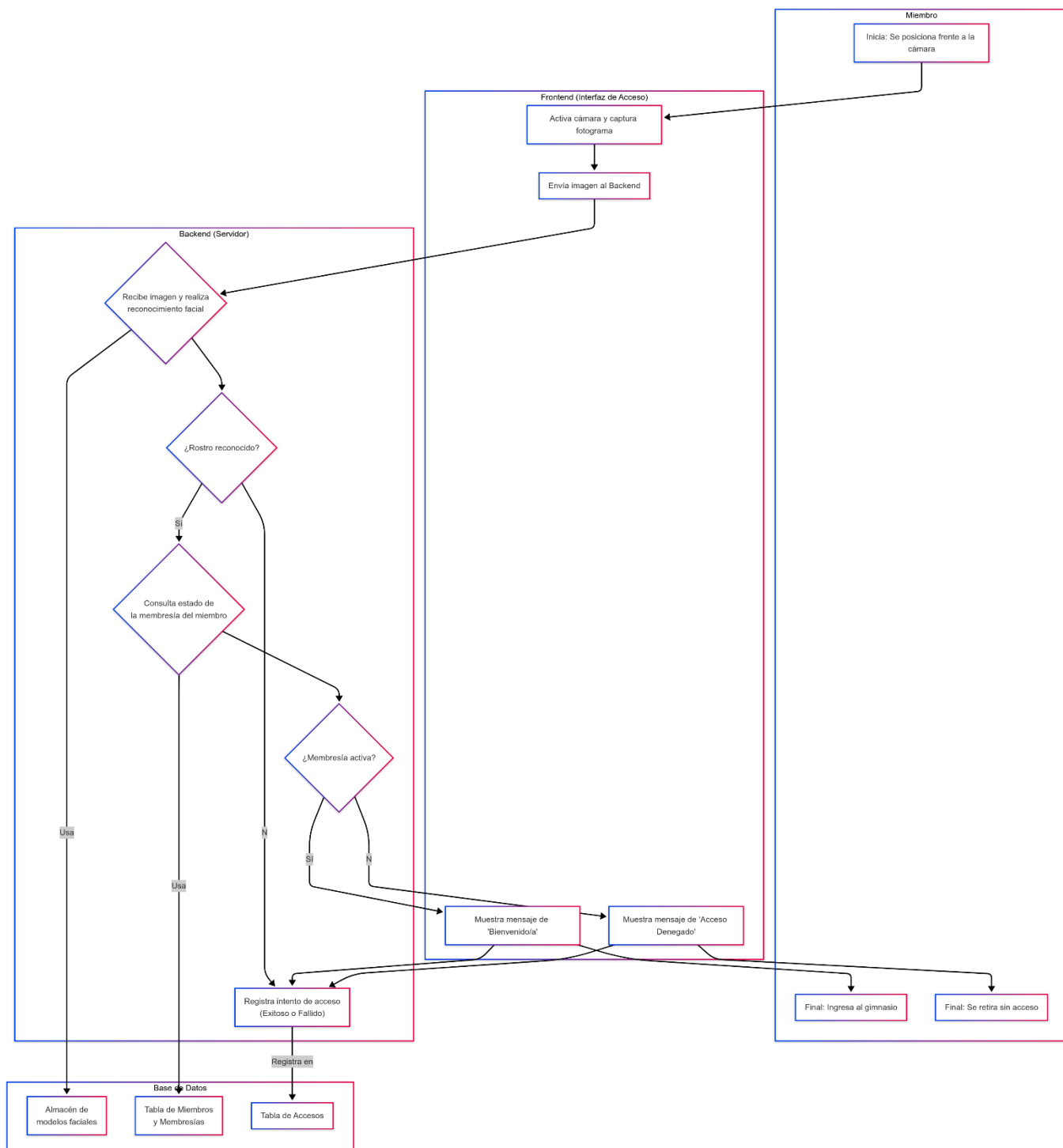
7.3.2.1 Base de Datos



7.3.2. Sistema



7.3.3 Actividades



7.4 Diccionario de Datos

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Diccionario%20de%20datos%20.xlsx

7.5 Manual Técnico

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Manual%20Tecnico.pdf

7.6 Manual de Usuario

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Manual%20de%20Usuario.pdf

7.7 Manual de Identidad

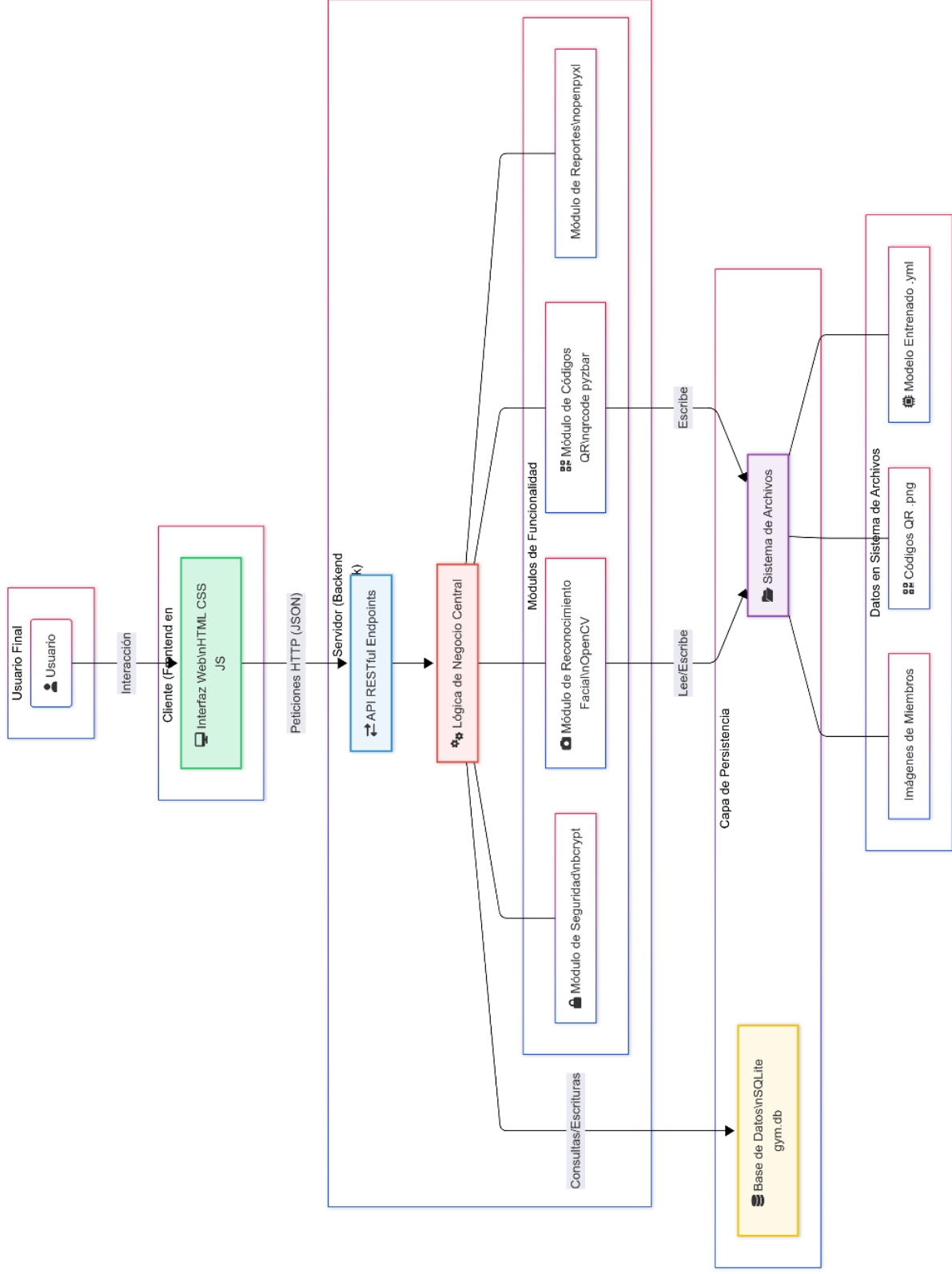
https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Manual%20de%20identidad%20.pdf

7.8 Arquitectura de software

7.8.1. Modelo Arquitectónico General

El sistema se ha diseñado siguiendo un **modelo cliente-servidor**. Esta arquitectura desacopla la presentación y la lógica de negocio, permitiendo una mayor flexibilidad y escalabilidad. La comunicación entre el cliente (frontend) y el servidor (backend) se realiza a través de una **API RESTful**, utilizando el protocolo HTTP(S) y el formato de intercambio de datos JSON.

- **Cliente (Frontend):** Es una aplicación web construida con tecnologías estándar (HTML5, CSS3 y JavaScript). Se encarga de toda la interacción con el usuario, la presentación de los datos y la captura de información (formularios, imágenes de cámara, etc.). No contiene lógica de negocio crítica; su función principal es consumir los servicios expuestos por el backend.
- **Servidor (Backend):** Es una aplicación monolítica desarrollada en Python con el microframework Flask. Alberga toda la lógica de negocio, la gestión de la base de datos, la seguridad, y los procesos de cómputo intensivo como el reconocimiento facial y la generación de reportes.



7.8.2. Arquitectura del Backend

El backend es el núcleo del sistema, implementado como una aplicación Flask que expone una serie de endpoints para ser consumidos por el cliente.

- **Framework:** Flask. Se eligió por su simplicidad, flexibilidad y el amplio ecosistema de librerías de Python disponibles.
- **API RESTful:** La aplicación define rutas (endpoints) para cada una de las funcionalidades principales (CRUD de gerentes, CRUD de miembros, control de acceso, etc.). Estas rutas manejan las peticiones HTTP (GET, POST, PUT, DELETE) y retornan respuestas en formato JSON.
- **Módulo de Seguridad:** La seguridad de las contraseñas se maneja mediante el hashing con bcrypt. Cada contraseña se convierte en un hash irreversible antes de ser almacenada en la base de datos, protegiendo las credenciales de los usuarios. La validación de gerentes para operaciones críticas se realiza en cada endpoint correspondiente.
- **Módulo de Reconocimiento Facial (OpenCV):**
 1. **Enrolamiento:** Se capturan imágenes del rostro de un miembro a través de un endpoint. Estas imágenes se guardan en el sistema de archivos (/static/uploads/member_images/).
 2. **Entrenamiento:** Un endpoint específico invoca un proceso que lee todas las imágenes de los miembros, utiliza el clasificador en cascada de Haar (haarcascade_frontalface_default.xml) para detectar rostros y entrena un modelo **LBPH (Local Binary Patterns Histograms)**. El modelo entrenado se guarda como un archivo modelo_lbph.yml.
 3. **Verificación:** Durante el control de acceso, se envía una imagen. El backend la procesa, detecta el rostro y utiliza el modelo modelo_lbph.yml para predecir la identidad del miembro.
- **Módulo de Códigos QR:**
 1. **Generación (qrcode):** Se genera una imagen QR que codifica el ID único del miembro. Esta imagen se almacena en el sistema de archivos (/static/uploads/qrs/).
 2. **Verificación (pyzbar):** Durante el acceso, el backend puede recibir una imagen, de la cual extrae y decodifica el código QR para obtener el ID del miembro.
- **Módulo de Reportes (openpyxl):** Permite generar dinámicamente archivos de Microsoft Excel (.xlsx) a partir de los registros de acceso almacenados en la base de datos, filtrados por un rango de fechas.

7.8.3. Arquitectura del Frontend

El frontend es una aplicación web clásica, sin un framework de JavaScript complejo, lo que simplifica su desarrollo y despliegue.

- **Tecnologías:** HTML5, CSS3 (con Tailwind CSS) y JavaScript (Vanilla).
- **Estructura:** Consiste en un conjunto de archivos .html interconectados que representan las diferentes vistas del sistema (login, registro de miembro, gestión, etc.).
- **Comunicación con Backend:** La lógica de la interfaz se implementa en archivos JavaScript. Se utiliza la función fetch() del navegador para realizar llamadas asíncronas a la API REST del backend, enviar datos de formularios y recibir la información para mostrarla dinámicamente en la página.
- **Gestión de Estado:** La sesión del gerente autenticado se gestiona de forma simple a través del localStorage del navegador, almacenando el ID y nombre del gerente para su uso en peticiones posteriores.

7.8.4. Arquitectura de Datos y Persistencia

La persistencia de los datos del sistema se gestiona a través de dos componentes principales: una base de datos relacional para los datos estructurados y el sistema de archivos para los datos no estructurados.

- **Sistema Gestor de Base de Datos (SGBD):** Se utiliza **SQLite 3**, un SGBD embebido en la aplicación que almacena toda la base de datos en un único archivo (gym.db). Fue seleccionado por su facilidad de configuración (no requiere un servidor de base de datos independiente), lo cual es ideal para un prototipo.
- **Modelo de Datos:** El esquema de la base de datos es relacional y está diseñado para garantizar la integridad y consistencia de la información. Las tablas principales inferidas del código son:
 - **gerentes:** Almacena la información de los administradores del gimnasio (ID, nombre, correo, contraseña hasheada, etc.).
 - **miembros:** Contiene los datos de los clientes del gimnasio (ID, nombre, datos de contacto, tipo de membresía, fechas de vigencia, contraseña hasheada).
 - **accesos:** Registra cada evento de entrada o salida (ID de acceso, ID del miembro asociado, tipo de acceso, fecha y hora, método de verificación).
 - **imagenes_miembros:** Asocia a los miembros con las rutas de las imágenes faciales almacenadas para el entrenamiento.
- **Sistema de Archivos:** Se utiliza para almacenar datos binarios o no estructurados:
 - **Imágenes de Miembros:** Las fotos tomadas para el enrolamiento facial.
 - **Códigos QR:** Las imágenes .png generadas para cada miembro.

- **Modelo de Reconocimiento:** El archivo modelo_lbph.yml generado tras el entrenamiento.

Esta arquitectura dual (base de datos + sistema de archivos) es eficiente, ya que evita almacenar grandes objetos binarios (BLOBs) en la base de datos, lo cual podría degradar su rendimiento.

7.9 Carta de Presentación de Proyecto

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Carta%20de%20presentaci3n.docx.pdf

7.10 Carta Compromiso

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Carta%20compromiso.docx.pdf

7.11 Matriz de Pruebas

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Matriz%20de%20Pruebas.xlsx

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Anexos%20Matriz%20de%20Pruebas.pdf

7.12 Plan de recuperación

https://github.com/YaelMora/Gimnasio_AdmonBD/blob/main/Reporte_General/Anexos/Plan%20de%20Recuperacion.pdf