



UNIVERSIDAD
DA VINCI
DE GUATEMALA

Evaluación Final

Nombre: Lourdes María Cámara Illescas

Carné: 202403284

Curso: Programación Web

Docente: Ing. Brandon Chitay

Guatemala, 7 de diciembre de 2025

Índice:

Introducción.....	3
Fit Access: Sistema de Gestión de Socios para Gimnasios	4
Definición del problema y solución	4
Propuesta técnica (Arquitectura).....	7
Persistencia de Datos (Base de Datos).....	10
Diseño de interfaz de programación (API).....	13
Planificación y costos	16
Prototipo de la interfaz de Fit Access	19
Conclusión.....	22
Links.....	23

Introducción:

En el presente documento se describe el diseño, la arquitectura y la planificación del desarrollo de Fit Access, una aplicación web orientada a la gestión de socios y membresías en gimnasios. El objetivo principal del proyecto es aplicar de forma integrada los conceptos vistos en el curso: análisis de problemas reales, toma de decisiones técnicas justificadas, diseño de arquitectura, definición de APIs y estimación de esfuerzo y costos.

Más que enfocarse únicamente en la implementación de código, este proyecto busca que el estudiante piense como un ingeniero de software: capaz de entender una necesidad del entorno, proponer una solución tecnológicamente viable, elegir las herramientas adecuadas y dimensionar el trabajo requerido para llevar un prototipo funcional (v0) a producción en un contexto realista.

Fit Access: Sistema de Gestión de Socios para Gimnasios:

Fit Access es un sistema web que permite a un gimnasio administrar la información de sus socios, sus membresías y el estado de cada una (activa, inactiva, suspendida), así como visualizar un resumen rápido de la situación general del gimnasio. La aplicación está pensada como una herramienta sencilla pero escalable, que facilite el trabajo del personal de recepción e instructores, quienes muchas veces llevan el control en hojas de cálculo, libretas o sistemas informales que no permiten un seguimiento ordenado del progreso de los usuarios.

1. Definición del problema y solución:

a) El problema:

Muchos gimnasios pequeños administran a sus socios de forma manual: listas en papel, archivos de Excel sin actualizar o mensajes dispersos en WhatsApp. Con este tipo de manejo es difícil saber con precisión qué membresías están activas, cuáles ya vencieron y qué socios están por perder el acceso si no renuevan. Esto genera desorden en el control de entrada, confusiones con las fechas y poca claridad para el dueño del gimnasio sobre cuántos socios activos tiene realmente.

Además, en algunos gimnasios existen instructores muy comprometidos con sus alumnos, que los acompañan, les dan seguimiento y diseñan rutinas personalizadas. Sin embargo, muchas veces no tienen una herramienta donde guardar de forma ordenada el historial de entrenamiento, los planes asignados y el progreso de cada persona.

Las rutinas y cambios de progreso se quedan en hojas sueltas o notas en el celular, lo que dificulta dar un seguimiento continuo y estructurado

al avance de cada socio.

b) La solución:

FitAccess es una aplicación web pensada para gimnasios pequeños que necesitan un control más profesional de sus socios, sus membresías y el seguimiento básico de sus entrenamientos.

La plataforma permitirá registrar a cada socio, asignarle un plan de membresía con fechas de inicio y vencimiento, y visualizar de forma clara el estado de su acceso al gimnasio.

Asimismo, FitAccess incluirá un apartado para que los instructores puedan registrar información clave del entrenamiento del socio, como:

- Rutinas o planes de ejercicio asignados.
- Objetivos básicos (por ejemplo: bajar de peso, ganar masa muscular, mejorar resistencia).
- Notas o actualizaciones sobre su progreso.

De forma general, FitAccess permitirá:

- Registrar y editar la información básica de los socios.
- Asignar planes de membresía (mensual, trimestral, anual, etc.) con sus fechas.

- Consultar el estado de la membresía (activa, próxima a vencer, vencida).
- Buscar socios por nombre, documento u otro identificador.
- Registrar rutinas y notas de progreso para que los instructores tengan un historial ordenado de cada socio.

c) Justificación:

La mayoría de sistemas de gestión para gimnasios están pensados para cadenas grandes y suelen ser complejos o costosos. Muchos gimnasios de barrio o pequeños negocios deportivos no pueden acceder a estas herramientas y terminan trabajando con métodos manuales que se vuelven difíciles de mantener. FitAccess se enfoca específicamente en este tipo de gimnasios pequeños, ofreciendo una solución ligera y centrada en dos necesidades muy concretas:

- Saber quién es socio y hasta cuándo es válida su membresía.
- Dar a los instructores un lugar sencillo donde consultar y actualizar la información básica de rutinas y progreso.

Esta solución aporta valor porque:

- Reduce errores al validar el acceso de los socios en la entrada.
- Facilita identificar a quién contactar para renovar su plan antes de que venza.

- Permite que el trabajo de los buenos instructores no se pierda en papeles o notas aisladas, sino que quede registrado de una forma organizada.
- Mejora la experiencia del socio al sentir que hay un seguimiento más personalizado y consistente de su avance.
- Sirve como base para extensiones a futuro, como control de pagos, registro de asistencias o reportes más detallados.

2. Propuesta técnica (Arquitectura):

d) Frontend:

Para el desarrollo del frontend de FitAccess se consideraron tres opciones principales: **React**, **Vue** y **Angular**.

- **React:** Biblioteca muy popular para construir interfaces de usuario. Tiene una gran comunidad, abundante documentación y se integra muy bien con herramientas modernas de despliegue como Vercel. Además, muchos generadores de interfaces basados en IA, como v0, producen componentes en React.
- **Vue:** Framework progresivo, sencillo de aprender y con una sintaxis clara. Es una buena opción para proyectos pequeños y medianos, pero tiene un ecosistema algo más pequeño comparado con React.

- **Angular:** Framework completo mantenido por Google, con una arquitectura bastante estructurada. Es potente, pero su curva de aprendizaje es más alta y puede resultar pesado para proyectos pequeños como FitAccess.

e) Elección:

Para FitAccess se utilizará React a través del framework Next.js, porque:

- Permite construir aplicaciones web modernas con manejo de rutas, páginas y componentes reutilizables.
- Se integra de forma nativa con Vercel para el despliegue.
- Es compatible con el uso de v0, que generará parte del prototipo de la interfaz en forma de componentes React que luego pueden ajustarse manualmente.

f) Uso de v0 en el prototipo:

El prototipo inicial de la interfaz de FitAccess se construirá utilizando v0, describiendo en lenguaje natural las pantallas principales, por ejemplo:

- Pantalla de listado de socios con tabla, buscador y botón para agregar nuevo socio.
- Vista de detalle de socio con información personal, membresías y rutinas.

- Pantalla para que el instructor registre una nueva rutina de entrenamiento.

El código generado por v0 servirá como base para el frontend y se integrará dentro del proyecto Next.js, realizando los ajustes necesarios para adaptarlo a la lógica específica del sistema.

g) Backend y publicación:

Para el backend de FitAccess se utilizará también Next.js, aprovechando sus API Routes para exponer los endpoints que consumirá el frontend. Estas rutas se implementarán en Node.js, utilizando JavaScript o TypeScript, y se conectarán a la base de datos mediante un ORM o biblioteca de acceso a datos.

De esta forma:

- El frontend y el backend comparten el mismo proyecto.
- La comunicación entre la interfaz y el servidor se realiza mediante peticiones HTTP a los endpoints definidos (por ejemplo, /api/socios, /api/membresias, /api/rutinas).
- Se simplifica la arquitectura y el despliegue, ya que todo se administra en un solo repositorio.

h) Publicación del proyecto:

- La aplicación web (frontend + API Routes) se desplegará en Vercel, que ofrece integración directa con proyectos Next.js y despliegue continuo a partir del repositorio.
- La base de datos SQL se alojará en un servicio gestionado en la nube (por ejemplo, una instancia de PostgreSQL en un proveedor externo), al que el backend se conectará mediante variables de entorno seguras.

Esta arquitectura permite que FitAccess funcione como una aplicación web moderna, accesible desde cualquier navegador, con un backend ligero pero suficiente para las necesidades del sistema.

3. Persistencia de Datos (Base de Datos):

i) Elección entre SQL y NoSQL:

Para la persistencia de datos de FitAccess se optará por una base de datos SQL, concretamente PostgreSQL.

La elección se justifica porque la información del sistema es altamente estructurada y relacional:

- Socios que pertenecen a un gimnasio.
- Membresías asociadas a cada socio, con fechas de inicio y fin.

- Planes de membresía predefinidos.
- Rutinas de entrenamiento asignadas por instructores.
- Registros de progreso vinculados a un socio y, opcionalmente, a una rutina.

Un modelo relacional facilita:

- Permite definir llaves foráneas entre las distintas entidades.
- Garantiza integridad referencial (por ejemplo, que no exista una membresía sin socio o una rutina sin su socio asociado).
- Facilita consultas típicas como: “¿Qué socios tienen membresía activa?”, “¿quiénes están por vencer esta semana?”, “¿qué rutinas tiene asignadas un socio?”, “¿cómo ha sido el progreso de un socio en el último mes?”

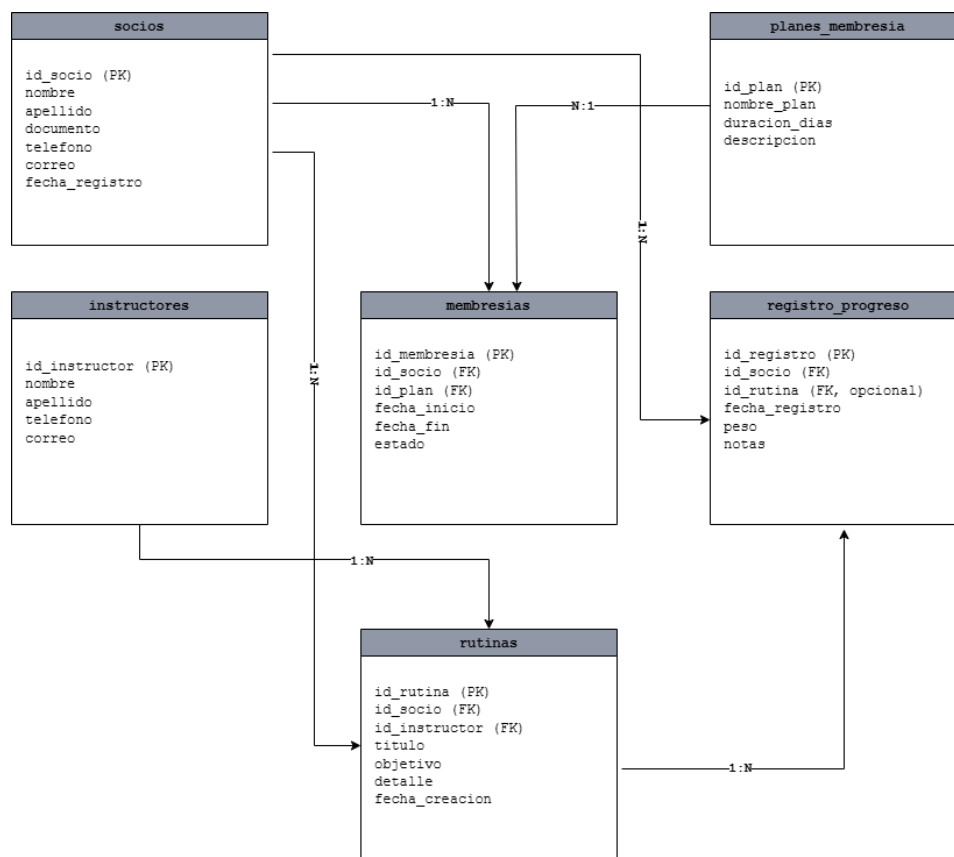
Por estas razones, una base de datos SQL resulta más natural y mantenible que una alternativa NoSQL para la primera versión de FitAccess.

j) Esquema de base de datos:

El modelo de datos de FitAccess se organiza en torno a cinco grupos principales de información:

- Socios que pertenecen a un gimnasio.
- Membresías asociadas a cada socio, con fechas de inicio y fin.
- Planes de membresía predefinidos.
- Rutinas de entrenamiento asignadas por instructores.
- Registros de progreso vinculados a un socio y, opcionalmente, a una rutina.

A continuación, se presenta el diagrama de entidad relación, con sus campos principales.

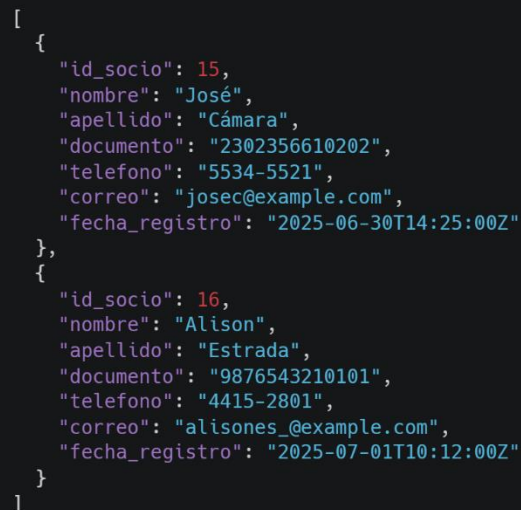


4. Diseño de interfaz de programación (API):

La API de FitAccess se diseña bajo el estilo REST, utilizando JSON como formato de intercambio de datos. Estos endpoints serán implementados como API Routes en Next.js y serán consumidos por el frontend de la aplicación.

Endpoint 1 – Obtener lista de socios:

- **Método:** GET
- **Ruta:** /api/v1/socios
- **Lenguaje:** JSON



```
[
  {
    "id_socio": 15,
    "nombre": "José",
    "apellido": "Cámara",
    "documento": "2302356610202",
    "telefono": "5534-5521",
    "correo": "josec@example.com",
    "fecha_registro": "2025-06-30T14:25:00Z"
  },
  {
    "id_socio": 16,
    "nombre": "Alison",
    "apellido": "Estrada",
    "documento": "9876543210101",
    "telefono": "4415-2801",
    "correo": "alisiones@example.com",
    "fecha_registro": "2025-07-01T10:12:00Z"
  }
]
```

Endpoint 2 – Crear nuevo socio:

- **Método:** POST
- **Ruta:** /api/v1/socios
- **Lenguaje:** JSON



```
{
  "id_socio": 17,
  "nombre": "Luis",
  "apellido": "Hernández",
  "documento": "2757503371801",
  "telefono": "4000-2718",
  "correo": "luisher1@example.com",
  "fecha_registro": "2025-07-20T14:25:00Z"
}
```

Respuesta 201 Created:



```
{
  "nombre": "Luis",
  "apellido": "Hernández",
  "documento": "2757503371801",
  "telefono": "4000-2718",
  "correo": "luisher1@example.com"
}
```

Endpoint 3 – Obtener detalle de un socio:

- **Método:** GET
- **Ruta:** /api/v1/socios/:id_socio/detalle
- **Lenguaje:** JSON

```
{
  "socio": {
    "id_socio": 17,
    "nombre": "Luis",
    "apellido": "Hernández",
    "documento": "2757503371801",
    "telefono": "4000-2718",
    "correo": "luisher1@example.com",
    "fecha_registro": "2025-07-20T14:25:00Z"
  },
  "membresias": [
    {
      "id_membresia": 30,
      "id_plan": 1,
      "nombre_plan": "Mensual",
      "fecha_inicio": "2025-07-20",
      "fecha_fin": "2025-08-17",
      "estado": "vencida"
    },
    {
      "id_membresia": 32,
      "id_plan": 2,
      "nombre_plan": "Mensual Premium",
      "fecha_inicio": "2025-09-01",
      "fecha_fin": "2025-12-12",
      "estado": "activa"
    }
  ],
  "rutinas": [
    {
      "id_rutina": 5,
      "titulo": "Rutina fuerza",
      "objetivo": "Ganar fuerza y masa muscular",
      "detalle": "Lunes: pecho y tríceps; Miércoles: espalda y bíceps; Viernes: pierna y hombro.",
      "fecha_creacion": "2025-01-10",
      "instructor": {
        "id_instructor": 2,
        "nombre": "Carla",
        "apellido": "López"
      }
    }
  ]
}
```

5. Planificación y costos:

k) Alcance considerado:

La planificación y los costos de FitAccess se basan en el trabajo necesario para desarrollar e implementar una primera versión funcional del sistema en un gimnasio pequeño (MVP).

Este alcance incluye:

- Analizar la situación actual del gimnasio en cuanto a control de socios, membresías y seguimiento básico de entrenamientos.
- Diseñar la solución a nivel técnico: arquitectura de la aplicación web, modelo de datos y API.
- Desarrollar una versión inicial de FitAccess que permita registrar socios, asignar planes de membresía y consultar información básica de rutinas y progreso.
- Configurar el sistema en un entorno de despliegue y dejarlo listo para ser utilizado en un gimnasio piloto.
- Preparar la documentación mínima y una breve sesión de capacitación para que el personal del gimnasio pueda usar la herramienta.

No se consideran en esta estimación fases posteriores como soporte continuo a largo plazo, nuevas funcionalidades avanzadas o despliegue en múltiples sucursales, sino únicamente el esfuerzo necesario para una primera versión utilizable de FitAccess.

I) Estimación de esfuerzo:

La estimación de horas de trabajo para el desarrollo del MVP de FitAccess se ha realizado pensando en una sola persona encargada del análisis, diseño, desarrollo y puesta en marcha inicial del sistema.

#	TAREA	HORAS ESTIMADAS
1	Levantamiento y análisis de requerimientos con enfoque en la realidad de un gimnasio pequeño	6 h
2	Diseño de la solución: arquitectura general, modelo de datos y definición de la API	8 h
3	Configuración del entorno de desarrollo y herramientas (proyecto Next.js, control de versiones)	4 h
4	Desarrollo del frontend: pantallas principales para gestión de socios y visualización de membresías	10 h
5	Desarrollo del backend y conexión con la base de datos (endpoints principales y lógica básica)	10 h
6	Pruebas funcionales básicas, corrección de errores y ajustes de interfaz	8 h
7	Despliegue del sistema en un entorno en la nube y configuración inicial para su uso en el gimnasio	6 h
8	Documentación básica del sistema y breve capacitación al personal del gimnasio piloto	6 h
Total estimado sin buffer		58 h

m) Buffer por incertidumbre (20 %):

Como en todo proyecto de software, pueden surgir imprevistos (retrabajo, pequeños cambios de alcance, problemas técnicos). Para evitar subestimar el esfuerzo, se añade un 20 % de colchón sobre las 58 horas estimadas.

- **Horas base:** 58 h
- **20 % de 58 h** \approx 12 h
- **Total con buffer:** 58 h + 12 h = 70 h

El esfuerzo total estimado para desarrollar e implementar la primera versión de FitAccess es de aproximadamente 70 horas de trabajo individual.

n) Presupuesto:

Para calcular el costo del proyecto, se asume una tarifa de Q 80.00 por hora de trabajo, correspondiente a un desarrollador junior que realiza tanto el análisis como el desarrollo y la puesta en marcha inicial del sistema.

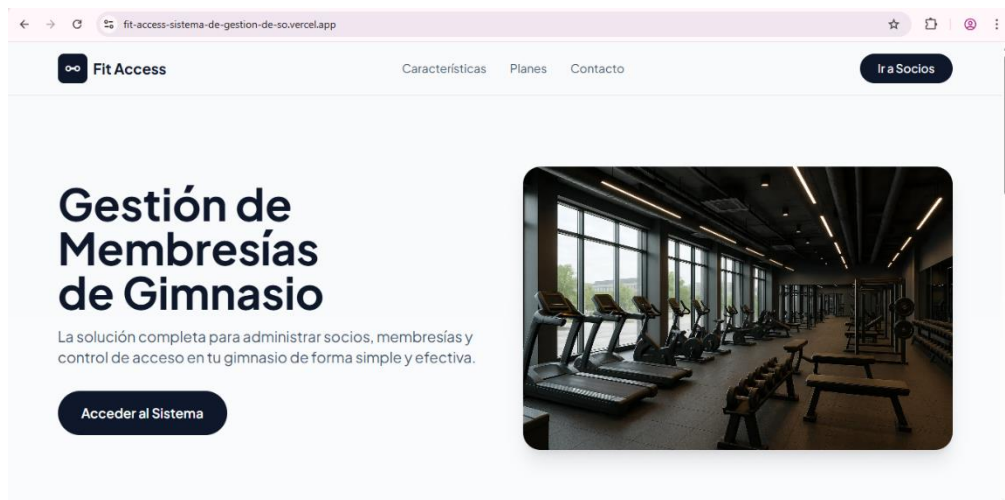
- **Costo por hora:** Q 80.00
- **Horas totales estimadas (con buffer):** 70 h

$$\text{Costo Total} = 70 \text{ h} \times \text{Q}80.00/\text{h} = \text{Q}5,600.00$$

Costo total estimado del proyecto FitAccess (MVP): Q 5,600.00

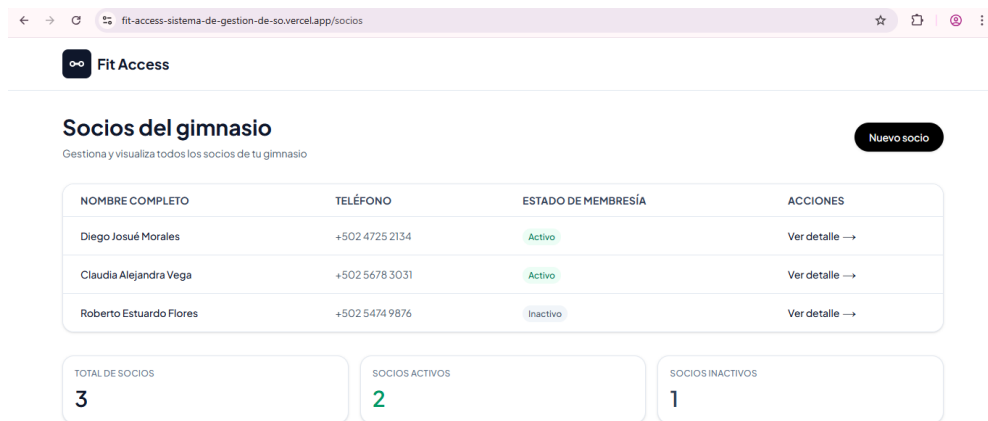
6. Prototipo de la interfaz de Fit Access:

El prototipo v0 de Fit Access se implementó como una aplicación web sencilla pero visualmente limpia y moderna, compuesta por tres vistas principales: la página de inicio, el módulo de socios y la vista de detalle de cada socio. Estas pantallas permiten demostrar el flujo básico del sistema y cómo se organizaría la información en una versión más completa.



La pantalla de inicio presenta la marca Fit Access en la parte superior, junto con un encabezado que incluye el logo, el nombre de la aplicación y un botón de acceso rápido al módulo de socios. En la sección principal se muestra un título destacado sobre la gestión de membresías de gimnasio, un texto explicativo y un botón de llamada a la acción. Más abajo se agregan tarjetas que resumen las características principales de la aplicación (gestión de socios, planes flexibles, reportes) y una sección de estadísticas de ejemplo, pensada para ilustrar el tipo de métricas que podría mostrar el sistema en el futuro.

Mientras tanto, la vista /socios muestra una tabla con el listado de socios registrados. Cada fila incluye el nombre completo del socio, su número de teléfono y el estado de su membresía. Este estado se representa mediante etiquetas de colores (verde para “Activo”, gris para “Inactivo”) que facilitan una lectura rápida. En la parte superior se muestra un título descriptivo (“Socios del gimnasio”) y un botón “Nuevo socio”, que en futuras versiones podría utilizarse para registrar nuevos socios desde la interfaz. Desde esta misma tabla se puede navegar al detalle de cada socio mediante el enlace “Ver detalle”.



NOMBRE COMPLETO	TELÉFONO	ESTADO DE MEMBRESÍA	ACCIONES
Diego Josué Morales	+502 4725 2134	Activo	Ver detalle →
Claudia Alejandra Vega	+502 5678 3031	Activo	Ver detalle →
Roberto Estuardo Flores	+502 5474 9876	Inactivo	Ver detalle →

TOTAL DE SOCIOS

3

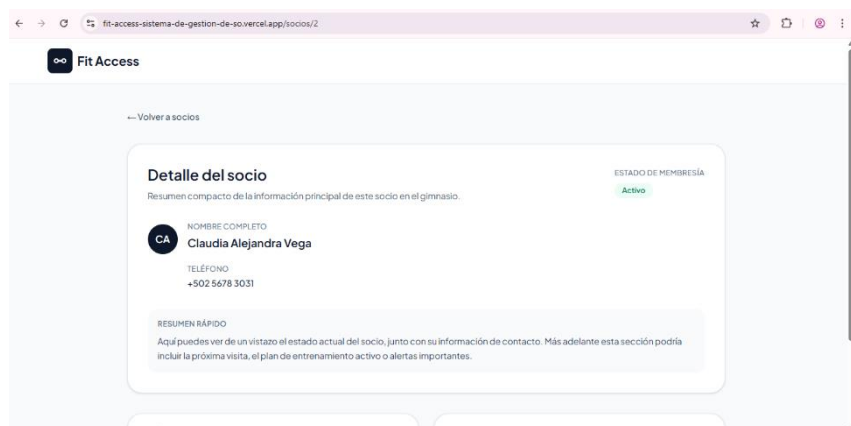
SOCIOS ACTIVOS

2

SOCIOS INACTIVOS

1

Por último, la ruta /socios/[id] presenta una ficha más detallada del socio seleccionado. En la parte superior se mantiene el encabezado con el logo y la marca Fit Access, así como un enlace para regresar al listado general de socios. En el contenido principal se muestra un



avatar con las iniciales del socio, su nombre completo, su teléfono y el

estado actual de su membresía, acompañado de una breve descripción que explica que se trata de un resumen compacto de su información. Asimismo, se incluyen tarjetas informativas como “Próximos pasos” y “Notas”, donde se describe el tipo de datos que podrían incorporarse en versiones futuras (por ejemplo, evaluaciones físicas, renovaciones próximas, historial de rutinas o comentarios del instructor).

Conclusión:

El desarrollo de Fit Access permitió recorrer de forma completa el ciclo de conceptualización y diseño de un proyecto de software profesional, desde la identificación del problema hasta la planificación del esfuerzo necesario para implementar un primer prototipo funcional. A partir de la necesidad detectada en gimnasios que no cuentan con un sistema formal de gestión de socios, se definió una solución que aporta valor al centralizar la información de membresías, datos de contacto y estado de los usuarios.

En el plano técnico, se seleccionó una arquitectura basada en una aplicación web con frontend en React/Next.js, un backend que expone una API REST y una base de datos relacional en SQL, cuyo modelo fue representado mediante un diagrama entidad-relación y un archivo schema.sql. Además, se documentaron endpoints clave para la interacción entre frontend y backend, y se estimaron horas de trabajo y costos considerando un margen de incertidumbre del 20%, como es recomendable en proyectos reales.

Finalmente, la construcción del prototipo v0 y su integración con la documentación técnica refuerzan la importancia de planificar antes de desarrollar, justificar cada decisión tecnológica y comunicar claramente el alcance del sistema. Aunque Fit Access se presenta como un MVP, su diseño deja abierta la posibilidad de crecer hacia funcionalidades mucho más avanzadas, como el registro de rutinas, evaluaciones físicas, historial de pagos y paneles de métricas para la administración del gimnasio.

Links:

Repositorio en GitHub: <https://github.com/lcamara15/Fit-Access-Sistema-de-Gestion-de-Socios-para-Gimnasios.git>

Vídeo de Youtube:

Presentación: https://youtu.be/S_GITjiiREI?si=N9P7EX-QjvhjU6to

Recorrido por la interfaz: <https://youtu.be/gcgZ2Atz0CE?si=hARELuXa1xiRqLJ->

Link del prototipo: <https://fit-access-sistema-de-gestion-de-so.vercel.app/>