



Informe Desarrollo Móvil 1

Aplicación para un Studio Fitness

Alumnos:

Jessica Ávila – T32411145

José Martínez - T32311436

Índice

Contenido

Informe Desarrollo Móvil	3
Resumen gerencial	3
Propósito principal	3
Características principales.....	3
Beneficios para el gimnasio y o administrador	3
Beneficios para el usuario	4
Glosario.....	4
Requerimientos:	5
Metas de arquitectura:	5
Diagrama de proceso	6
Funcionalidades principales.....	7
Tecnología 8	
Desarrollo.....	8
Modelo Entidad Relación	10

Informe Desarrollo Móvil

Descripción de la solución

Resumen gerencial

El presente proyecto tiene como objetivo el desarrollo de una aplicación móvil integral que facilite la gestión de clases, horarios, eventos y promociones para un gimnasio, al mismo tiempo que organiza y almacena eficientemente la información de los clientes en una base de datos centralizada.

La plataforma cuenta con una interfaz de usuario intuitiva, diseñada para permitir una navegación fluida y fácil acceso a todas las funcionalidades. Los usuarios podrán consultar horarios y clases disponibles, visualizar eventos y promociones, y realizar reservas de clases de manera sencilla y rápida. Esta solución no solo mejorará la experiencia del cliente al hacer sus interacciones más ágiles, sino que también optimizará la administración del gimnasio, reduciendo la carga operativa y permitiendo una gestión más efectiva de los recursos.

Propósito principal

La aplicación tiene como objetivo mejorar la experiencia tanto de los usuarios como de los administradores del gimnasio, optimizando la administración de recursos, facilitando la comunicación, y promoviendo una interacción fluida y eficiente.

Características principales

- Gestión de usuarios
- Gestionar clases
- Gestión de reservas o matriculas de clases
- Gestionar eventos y promociones

Beneficios para el gimnasio y o administrador

- Automatización de reservas
- Mayor interacción con los clientes
- Aumento en la retención de clientes y fidelización mediante la aplicación

Beneficios para el usuario

- Comodidad para el cliente al momento de informarse sobre clases y horarios.
- Hacer reservas desde la aplicación con solo presionar un botón en la pantalla.
- Informarse sobre los eventos y promociones de manera rápida.

Glosario

Término	Definición
import	Permite traer módulos, funciones o componentes desde otros archivos o bibliotecas.
export	Se usa para hacer funciones, variables o componentes disponibles en otros archivos.
Login	Proceso de autenticación para acceder a una aplicación o sistema.
Models	Representan la estructura y lógica de los datos de la aplicación.
Pages	Pantallas o secciones de la aplicación.
Component	Elemento reutilizable de la interfaz en React y React Native.
Function:	Bloque de código que ejecuta una tarea específica.
Provider	Componente que gestiona y proporciona estado o datos a otros componentes.
View:	Contenedor en React Native para estructurar la interfaz de usuario.
Navigator	Sistema que permite moverse entre pantallas en la aplicación.
Context	Es una herramienta que permite compartir datos globales entre componentes sin necesidad de pasarlos manualmente como <i>props</i> .

Requerimientos:

Plataforma de Desarrollo:

- La aplicación esta desarrollada con tecnologías que permitan el acceso desde dispositivos móviles, como React Native o Expo para la parte del frontend.
- El backend debe estar construido con Node.js y Express.js, garantizando que la aplicación sea escalable y pueda manejar múltiples usuarios de forma eficiente.

Acceso a Internet:

- La aplicación depende de una conexión a internet para acceder a la base de datos y realizar acciones como reservas de clases, consultas de horarios, eventos y actualizaciones en tiempo real.
- Las funcionalidades como la visualización de clases y el historial de reservas deben estar disponibles siempre que el usuario tenga acceso a la red.

Metas de arquitectura:

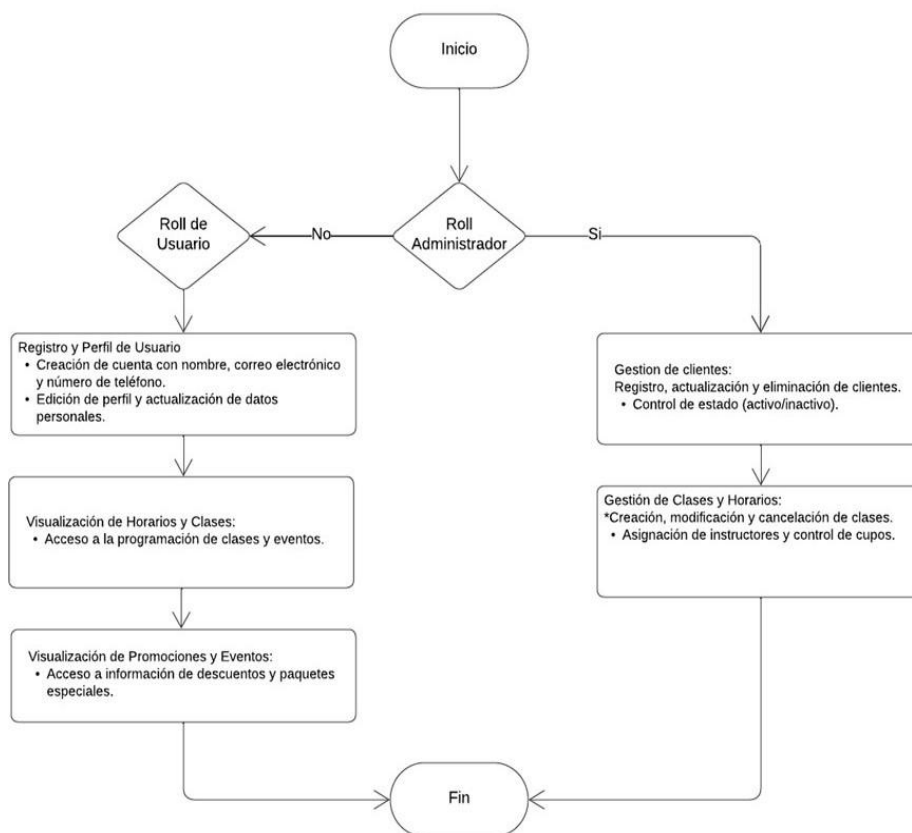
Escalabilidad:

- La arquitectura debe permitir agregar nuevas funcionalidades sin afectar el rendimiento. Por ejemplo, agregar nuevos tipos de clases o funcionalidades relacionadas con la gestión de promociones sin que sea necesario rehacer todo el sistema.
- Debe ser fácil integrar nuevas características como la integración con servicios de pago para procesar reservas o la adición de nuevos módulos como clases en vivo o streaming de eventos.

Modularidad:

- La arquitectura debe ser modular, permitiendo una fácil separación de elementos (frontend, backend, base de datos) y permitiendo una fácil actualización de partes individuales sin afectar otras áreas.
- Cada componente debe tener una responsabilidad clara (por ejemplo, el módulo de reservas debe manejar exclusivamente las reservas, sin involucrar la gestión de clases).

Diagrama de proceso



Funcionalidades principales

Nombre del uso	Descripción de la funcionalidad	Ruta de uso
Inicio de sesión	Permite al usuario iniciar sesión utilizando su correo electrónico y contraseña para acceder a las funciones del sistema.	- Al iniciar sesión, se debe verificar el rol del usuario (Administrador o Usuario) y redirigir a la página correspondiente.
Gestión de usuarios	Los administradores pueden crear, ver, actualizar, y eliminar usuarios.	- El administrador puede agregar nuevos usuarios y asignarles roles. - Se permite la actualización de datos de los usuarios y la eliminación de usuarios.
Gestión de clases	Permite a los administradores agregar, actualizar, ver y eliminar clases en el gimnasio. Incluye detalles como nombre, horario, instructor y número de cupos.	- El administrador debe ingresar los datos de la clase, como nombre, descripción, instructor, horario y cupos. - Se debe mostrar la lista de clases para que el administrador pueda realizar modificaciones o eliminarlas. - Los usuarios pueden ver las clases disponibles.
Gestión de eventos	Permite a los administradores crear, actualizar, eliminar y mostrar eventos informativos que estarán disponibles para los usuarios.	- El administrador agrega un evento con una descripción. - Los eventos deben mostrarse a los usuarios para mantenerlos informados de actividades importantes. - El administrador debe poder eliminar eventos previos.
Reservas de clases	Permite a los usuarios reservar clases de acuerdo con su disponibilidad de cupos.	- El usuario debe elegir una clase activa para realizar la reserva. - La reserva debe estar asociada con el ID del usuario y de la clase. - Se debe verificar si hay cupos disponibles antes de permitir la reserva. - Las reservas deben ser cancelables.
Visualización de eventos	Los usuarios pueden ver una lista de eventos informativos para mantenerse actualizados sobre las actividades del gimnasio.	- Los usuarios deben ver los eventos en una interfaz amigable, con una breve descripción. - Los eventos deben ser visibles hasta que el administrador decida eliminarlos.

Tecnología

Desarrollo

Node.js: Entorno de ejecución de JavaScript: en el servidor, basado en el motor V8 de Chrome. Permite desarrollar aplicaciones escalables y de alto rendimiento, especialmente para backend y APIs.

Express.js: Framework minimalista para Node.js que facilita la construcción de APIs RESTful. Usado para manejar las rutas y peticiones del servidor de manera eficiente.

React Native: Framework de código abierto desarrollado por Meta (Facebook) para crear aplicaciones móviles nativas en iOS y Android usando JavaScript y React.

MySQL: Sistema de gestión de bases de datos relacional (RDBMS) basado en SQL. Es ampliamente utilizado para almacenar y administrar datos en aplicaciones web y empresariales.

Sequelize: ORM (Object-Relational Mapping) para Node.js que facilita la interacción con la base de datos MySQL, permitiendo la definición de modelos y la ejecución de consultas de manera más sencilla.

React Navigation: Biblioteca para manejar la navegación en aplicaciones de React Native. Permite implementar navegación entre pantallas, como pilas (Stack), pestañas (Tabs) y cajones (Drawers), de manera fluida y personalizable. Es una de las soluciones más utilizadas para gestionar la navegación en apps móviles.

React Context API: Usada para manejar el estado global de la aplicación, como el usuario autenticado y sus reservas. Permite compartir datos entre componentes sin necesidad de pasar props manualmente.

Scrcpy: Herramienta de código abierto que permite controlar y visualizar la pantalla de dispositivos Android desde una computadora mediante USB o conexión inalámbrica, sin necesidad de root.

Postman: Herramienta utilizada para probar las APIs del backend y asegurarse de que las rutas y las respuestas sean correctas antes de ser implementadas en la aplicación.

Modelo Entidad Relación

