

# REPORTE DE PROYECTO

Base de Datos Distribuida para la gestión de un Gimnasio

ALUMNO:

Castro Paredes José Ángel  
Garcia Melo Ariana  
Morales Vázquez Rubén  
Pérez Romero Armando Oswaldo



# 1. Introducción

## Descripción general del proyecto

El presente proyecto tiene como objetivo el desarrollo de una plataforma para la gestión integral de gimnasios, basada en una arquitectura de Bases de Datos Distribuidas. Esta solución permitirá a los administradores gestionar de manera eficiente clientes, membresías, clases, entrenadores, pagos y otros aspectos fundamentales del negocio.

La adopción de una base de datos distribuida garantiza la disponibilidad de la información en múltiples ubicaciones físicas, lo que optimiza el rendimiento, facilita la escalabilidad del sistema y mejora la tolerancia a fallos.

## Dimensiones del Proyecto

### 1. Almacenamiento Distribuido

La información se almacenará en múltiples nodos distribuidos estratégicamente, lo que permitirá la gestión eficiente de grandes volúmenes de datos. Cada gimnasio contará con su propio nodo de almacenamiento, mientras que la información crítica, como pagos y membresías, será replicada en otros nodos con el fin de garantizar la disponibilidad y la consistencia de los datos.

### 2. Optimización del Rendimiento

Al distribuir la carga de trabajo entre varios servidores, se minimizarán los cuellos de botella y se mejorará el tiempo de respuesta del sistema. Esto es especialmente relevante durante los horarios de mayor demanda, como las horas de entrenamiento matutinas y vespertinas.

### 3. Escalabilidad Horizontal

El diseño del sistema permitirá la expansión progresiva de la plataforma mediante la incorporación de nuevos gimnasios o la ampliación de los existentes sin afectar el rendimiento global. Esta escalabilidad se logrará a través de la adición dinámica de nuevos nodos dentro de la red distribuida.

### 4. Seguridad y Consistencia de los Datos

Para garantizar la integridad y actualización de la información en todos los nodos, se implementarán mecanismos avanzados de replicación y sincronización de datos. Asimismo, se emplearán protocolos de seguridad robustos para la protección de la información sensible de los clientes, evitando accesos no autorizados o posibles vulnerabilidades.

### 5. Enfoque de Diseño Top-Down

La arquitectura del sistema se desarrollará siguiendo un enfoque Top-Down, iniciando con la definición de los objetivos estratégicos del negocio y desglosándolos en subsistemas específicos. Este proceso incluirá la elaboración de esquemas globales, la fragmentación lógica de los datos y la asignación eficiente de los fragmentos a los nodos correspondientes.

## Desafíos Potenciales y Estrategias de Mitigación

### 1. Consistencia de los Datos

En un entorno distribuido, mantener la coherencia de los datos es un desafío clave. Para abordar este problema, se implementarán protocolos de consenso, como Paxos o Raft, junto con técnicas avanzadas de replicación para garantizar la integridad de la información.

### 2. Rendimiento en Escenarios de Alta Demanda

Durante picos de uso, el sistema deberá procesar un gran volumen de transacciones concurrentes sin degradar su desempeño. Para mitigar este riesgo, se emplearán estrategias de balanceo de carga, mecanismos de caching y técnicas de sharding que optimicen el acceso a los datos y reducen la latencia.

### 3. Escalabilidad sin Interrupciones

A medida que la red de gimnasios crezca, el sistema deberá permitir la incorporación de nuevos nodos sin afectar su operatividad. La arquitectura distribuida facilitará esta expansión de manera natural, permitiendo una integración fluida y sin interrupciones del servicio.

## 2. Marco teorico

### Metodología de diseño SCRUM

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán Sprints. Contando con 5 fases:

1. Concepto: Se define de forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.
2. Especulación: en esta fase se hacen disposiciones con la información obtenida y se establecen los límites que marcarán el desarrollo del producto, tales como costes y agendas.
3. Exploración: Se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.
4. Revisión: El equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.
5. Cierre: Se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios, denominados "Mantenimiento".

### Objetivo General

Desarrollar una plataforma integral de gestión de gimnasios basada en una arquitectura de bases de datos distribuidas, que permita a los administradores gestionar de manera eficiente clientes, membresías, clases, entrenadores y pagos, garantizando alta disponibilidad, escalabilidad y seguridad de la información.

### Objetivos Especificos

1. Implementar una arquitectura de bases de datos distribuidas.
2. Optimizar el rendimiento del sistema.
3. Garantizar la escalabilidad del sistema.
4. Asegurar la integridad y seguridad de los datos.
5. Facilitar la gestión integral de gimnasios.
6. Mejorar la experiencia del usuario.
7. Garantizar la alta disponibilidad del sistema.
8. Cumplir con normativas y estándares.

## Requerimientos Funcionales y No Funcionales

### Funcionales

#### 1. Reservas de clases y equipos

Descripción: Implementar un sistema de reservas para clases grupales, entrenadores personales o equipos específicos (como bicicletas estáticas o máquinas de pesas).

##### Requerimientos funcionales:

- El sistema debe permitir a los usuarios reservar clases o equipos en horarios específicos.
- El sistema debe enviar recordatorios automáticos (por correo o notificaciones push).
- El sistema debe permitir a los administradores gestionar la disponibilidad de clases y equipos.

#### 2. Planes de entrenamiento personalizados

Descripción: Ofrecer a los usuarios planes de entrenamiento personalizados basados en sus objetivos (perder peso, ganar masa muscular, mejorar resistencia, etc.).

##### Requerimientos funcionales:

- El sistema debe permitir a los usuarios definir sus objetivos y preferencias.
- El sistema debe generar planes de entrenamiento automáticamente o permitir a los entrenadores crearlos manualmente.
- El sistema debe permitir el seguimiento del progreso del usuario en relación con su plan.

#### 3. Integración con pasarelas de pago y facturación electrónica

Descripción: Facilitar el proceso de pagos y facturación mediante integración con pasarelas de pago (como PayPal, Stripe o MercadoPago) y emisión automática de facturas electrónicas.

##### Requerimientos funcionales:

- El sistema debe permitir pagos en línea con diferentes métodos (tarjeta, PayPal, etc.).
- El sistema debe generar facturas electrónicas automáticamente después de cada pago.
- El sistema debe enviar comprobantes de pago por correo electrónico.

### No Funcionales

#### 1. Seguridad

Descripción: Implementar medidas adicionales de seguridad, mediante contraseñas para usuarios y entrenadores.

##### Requerimientos no funcionales:

- El sistema debe requerir autenticación a usuarios y entrenadores.
- El sistema debe registrar y auditar todos los accesos a la plataforma.

#### 2. Tiempo de respuesta

Descripción: Garantizar que el sistema responda rápidamente a las solicitudes de los usuarios, incluso durante picos de demanda.

##### Requerimientos no funcionales:

- El sistema debe responder en menos de 2 segundos para el noventa y cinco por ciento de las solicitudes.

### 3. Capacidad de usuarios concurrentes

Descripción: Asegurar que el sistema pueda manejar un gran número de usuarios activos al mismo tiempo sin degradar su rendimiento.

#### Requerimientos no funcionales:

- El sistema debe soportar 1,000 usuarios concurrentes sin degradación del rendimiento.

## Alcances y limitaciones

### Alcances

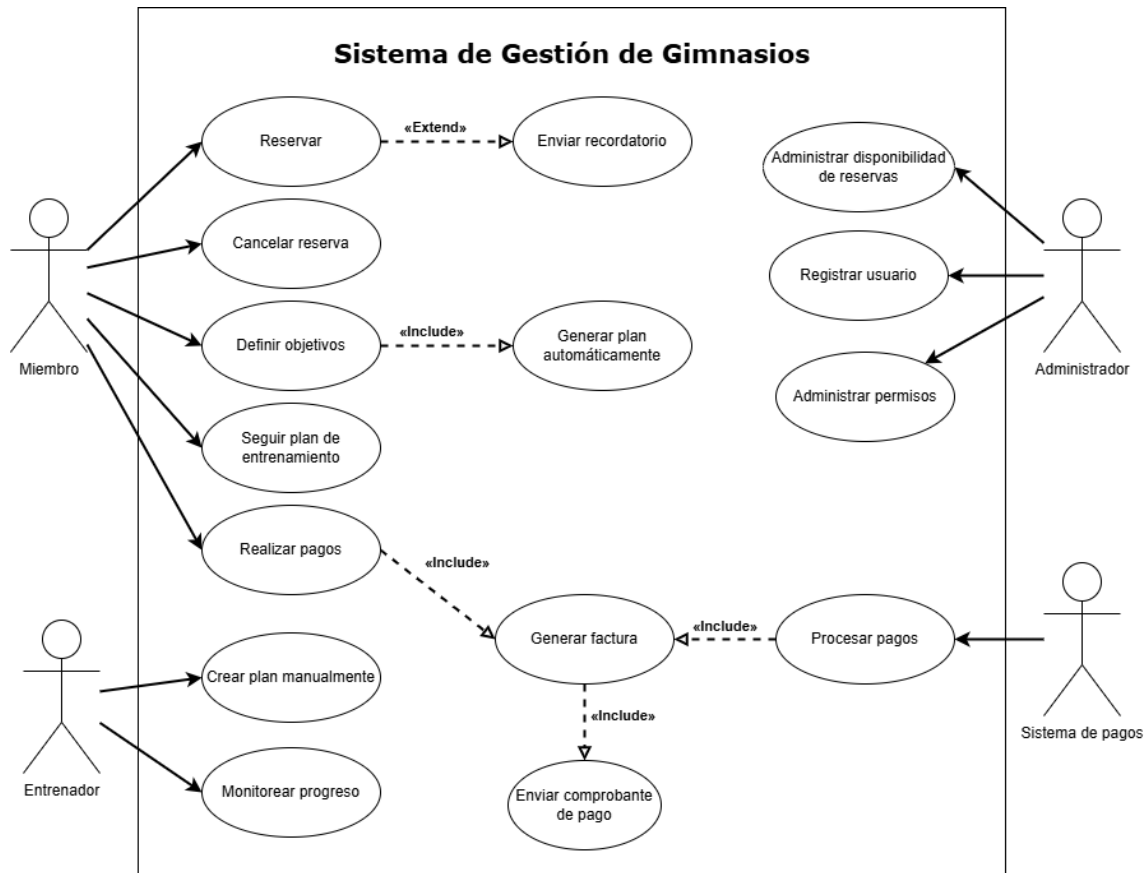
- La plataforma permitirá gestionar clientes, membresías y entrenadores.
- Se implementará una arquitectura distribuida básica con tres o cuatro nodos para demostrar el concepto de replicación y consistencia de datos.
- Se desarrollará una interfaz web básica con funcionalidades esenciales (registro de clientes, asignación de membresías, etc.).
- Se implementará autenticación de usuarios con contraseñas encriptadas.

### Limitaciones

- El proyecto se desarrollará con recursos limitados (hardware, software y tiempo), por lo que no se implementarán funcionalidades avanzadas como integración con wearables o apps de fitness.
- La arquitectura distribuida se implementará a pequeña escala (4 nodos) debido a limitaciones de infraestructura.
- El proyecto se desarrollará en un plazo limitado de un semestre, por lo que no se podrán implementar todas las funcionalidades deseadas.

### 3. Casos de Usuarios

#### 3.0.1 Casos de Uso



### 3.0.2 Secuencial

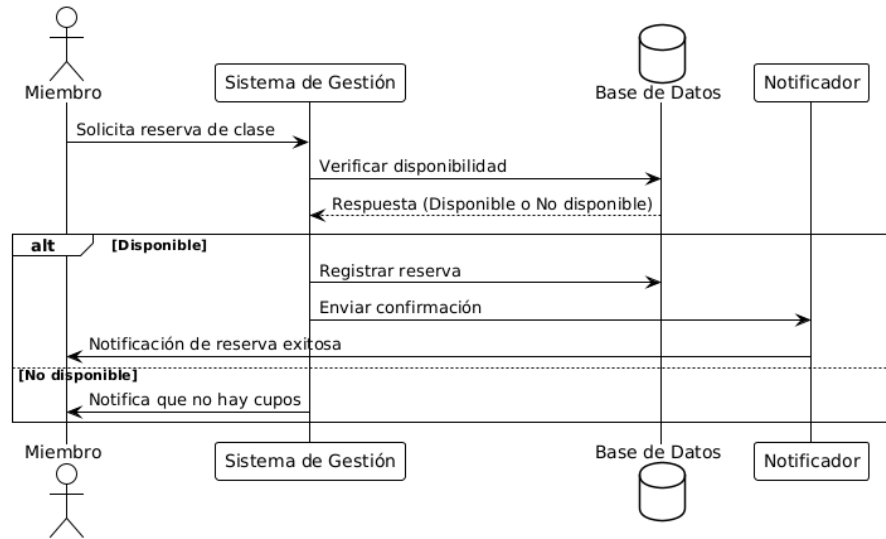


Figure 1: Reservar una Clase

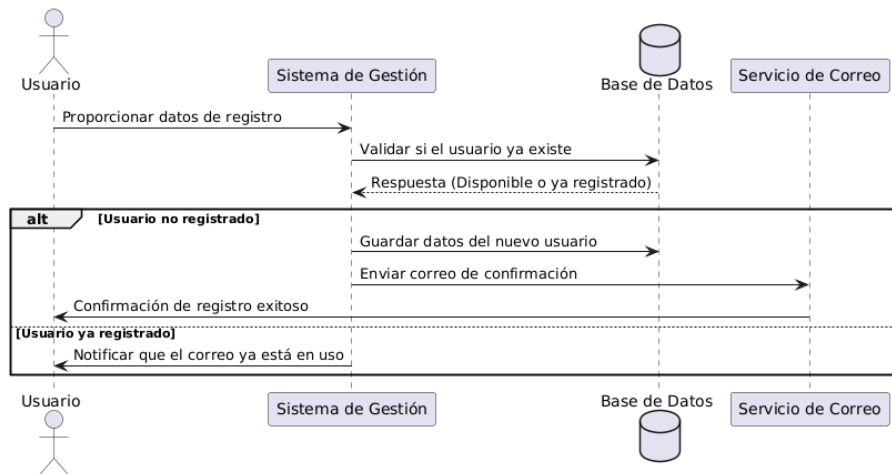


Figure 2: Registrar un Nuevo Usuario

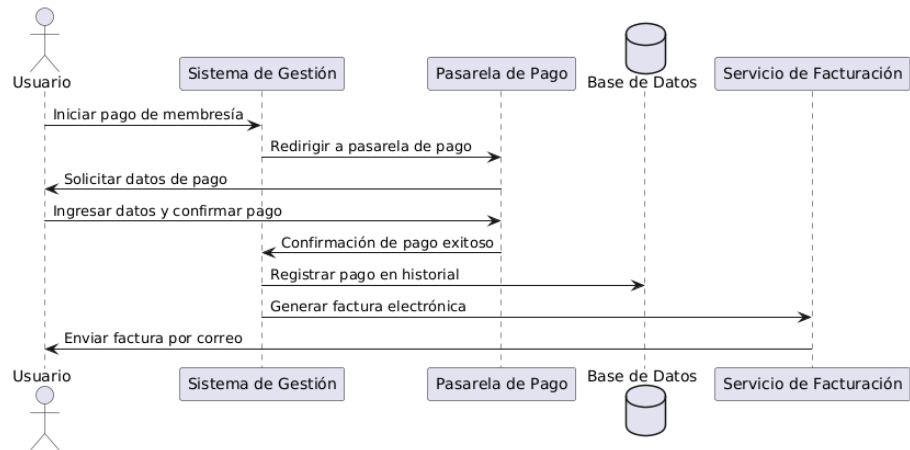


Figure 3: Procesar un Pago de Membresía

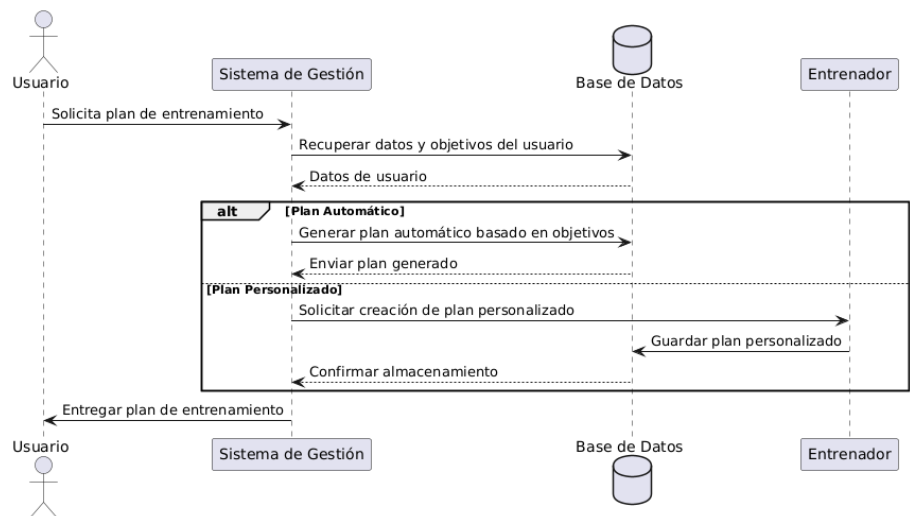


Figure 4: Generar Plan de Entrenamiento Personalizado



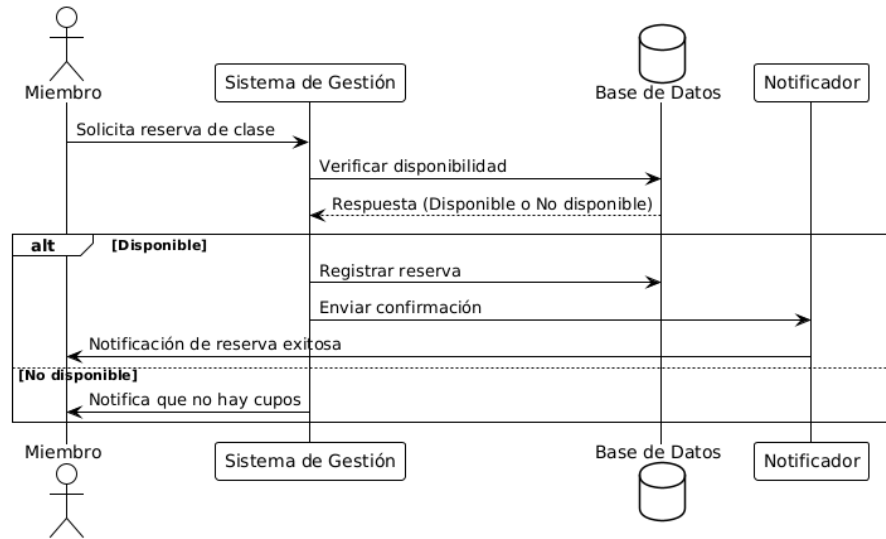


Figure 5: Registrar Asistencia a una Clase

## 4. Desarrollo

## 5. Conclusiones

## Referencias Bibliográficas

## References

- [1] Grabowska, S.; Saniuk, S. (**2022**). Business models in the industry 4.0 environment—results of web of science bibliometric analysis. *J. Open Innov. Technol. Mark. Complex*, 8(1), 19.