UNIVERSIDAD CATÓLICA BOLIVIANA
COCHABAMBA

Departamento de Ingeniería y Ciencias Exactas

Carrera de Ingeniería de Sistemas

Documentación Proyecto Backend Gimnasio

Estudiante(s):

Christian Alejandro Ferrufino Morales

Docente:

Richar Canoa Canchis

Cochabamba – Bolivia

Octubre de 2025

ÍNDICE

1.- Descripción del proyecto

El Sistema de Gestión de Gimnasio es una aplicación backend desarrollada en .NET Core que permite administrar las operaciones principales de un gimnasio, incluyendo gestión de usuarios, membresías, clases, horarios y control de asistencias.

2.- Casos de uso identificados

2.1- Gestión de usuarios

CU01 Registrar nuevo usuario

CU02 Actualizar información de usuario

CU03 Inactivar/Activar usuario

CU04 Consultar usuarios

2.2- Gestión de membresía

CU05 Asignar membresía a usuario

CU06 Crear membresía

CU07 Consultar membresías

CU08 Consultar membresía de usuario

2.3- Gestión de clases y horarios

CU09 Crear clase

CU10 Consultar clase

CU11 Asignar horario

2.4- Gestión de asistencias

CU12 Registrar asistencia

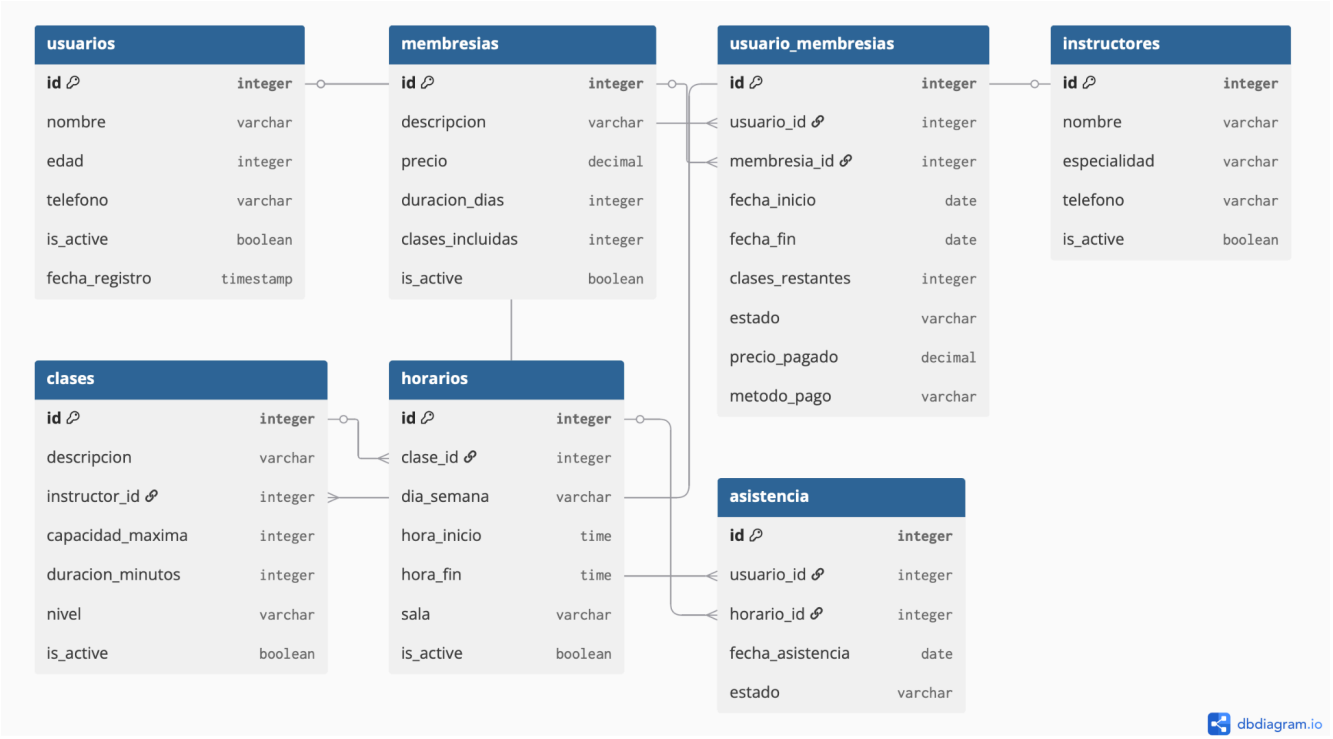
CU13 Consultar historial de asistencias

CU14 Verificar capacidad de clases

CU15 Verificar cantidad de clases restantes en membresía

CU16 Validar duplicidad de asistencias

3.- Entidad relación de la base de datos



4.- Priorización de procesos

Implementado (Alta prioridad) :

CU01,CU04 - CU16

Planeado (Baja prioridad) :

CU03 - CU04

5.- Explicación detallada casos de uso

- **CU01:** Registrar nuevo usuario:

Service: UsuarioService

Permite registrar a un usuario a la base de datos con su información basica. El usuario queda activo por defecto.

- **CU02:** Actualizar información de usuario:

Service: UsuarioService

Permite modificar los datos personales de un usuario ya registrado.

- **CU03:** Inactivar / Eliminar Usuario

Service: UsuarioService

Permite eliminar o marcar como inactivo a un usuario mediante el metodo DeleteUsuarioAsync() que por el momento es el que se encarga de deshabilitar el IsActive en las propiedades de usuarios.

- **CU04:** Consultar usuarios

Service: UsuarioService

Permite obtener todos los usuarios registrados o consultar algun usuario en especifico.

- **CU05:** Asignar membresia a Usuario

Service: UsuarioMembresiaService

Asocia una membresía existente a un usuario que exista. El sistema se encarga de validar que ambos no sean nulos.

- **CU06:** Crear membresía

Service: UsuarioMembresiaService

Crea un nuevo tipo de membresía con parámetros configurables como cantidad de clases y fecha de inicio y fin.

- **CU07:** Consultar membresía

Service: MembresiaService

Permite consultar los tipos de membresía existentes, facilitando selección y administración de estos.

- **CU08:** Consultar membresía de usuario

Service: UsuarioMembresiaService

Permite obtener todas las membresías asociadas a los usuarios o consultar la membresía de algún usuario en especifico**.**

- **CU09:** Crear clase

Service: ClaseService

Registrar una nueva clase en el sistema, especificando su descripción, capacidad maxima e instructor.

- **CU10:** Consultar clases

Service: ClaseService

Permite consultar todas las clases registradas, buscar por id o filtrar por instructor (GetClaseByInstructorId).

- **CU11:** Asignar horario

Service: HorarioService

Permite obtener los horarios disponibles por id o todos, facilitando al usuario elegir sesiones dependiendo su disponibilidad y ocupación de la clase.

- **CU12:** Registrar Asistencia

Service: AsistenciaService
Registra la asistencia de un usuario a una clase en una fecha determinada. El sistema valida que el usuario existe y este activo (IsActive = 1), que el horario corresponda a una clase valida, que no exista un registro duplicado para el mismo, que la clase no haya alcanzado su capacidad maxima para esa fecha, que el usuario tenga una membresía activa con clases restantes disponibles (ClasesRestantes > 0) y que la membresía aun este en vigencia comparando la fecha de finalización de la membresía con la fecha puesta dentro de la asistencia a registrar. Si todas las condiciones anteriormente mencionadas son cumplidas, la asistencia se guarda y se decrementa ClasesRestantes de la membresía.

- **CU13:** Consultar historial de asistencias

Service: AsistenciaService

Consulta todas las asistencias dependiendo si se quiere buscar por id de la asistencia, todas o todas las asistencias mandando un Id de usuario para visualizar todas las asistencias de ese usuario en concreto.

- **CU14:** Verificar capacidad de clases

Service: AsistenciaService

Antes de registrar una asistencia, el sistema valida que la clase no haya alcanzado su capacidad maxima de alumnos establecida.

- **CU15:** Verificar cantidad de clases restantes en membresía

Service: AsistenciaService

Controla que el usuario tenga clases disponibles en su membresía activa antes de permitir registrar la asistencia.

- **CU16:** Validar duplicidad de asistencias

Service: AsistenciaService

Evita que el usuario se registre mas de una vez en la misma clase o horario.

6.- Criterios de aceptación y reglas de negocio

6.1-Criterios de aceptación

CU01 :
El usuario debe contener campos todos los campos obligatorios(id, nombre, edad, telefono)

El usuario se registra con IsActive = 1

No se permiten usuarios con edad igual o menor a 10 años

CU02:

El sistema debe permitir modificar los datos personales del usuario existente

CU03:

El sistema debe cambiar el esta IsActive a 0 al inactivar en el (DeleteUsuarioAsync())

Por el momento no se permite eliminar fisicamente usuarios en la base de datos

Usuarios inactivos no pueden registrar nuevas asistencias.

CU04:

Debe permitir buscar usuarios registrados por Id o ver todos los registros

CU05:

El sistema debe validar que el usuario como la membresía existan

No se permite asignaciones a usuarios o membresías inexistentes

CU06:

El sistema debe crear un nuevo tipo de membresía con todos los parámetros requeridos

Debe validar que la cantidad de clases sea positivo

Debe asegurar que las fechas inicio y fin sean consistentes

CU07:

El sistema debe retornar todos los tipos de membresías disponibles

Debe permitir buscar membresías específicas

CU08:

El sistema debe retornar todas las membresías asociadas a un usuario específico

Debe mostrar todos sus atributos

CU09:

El sistema debe registrar una nueva clase con descripción, capacidad e instructor

Debe validar que el instructor exista en la base de datos

CU10:

El sistema debe retornar todas las clases registradas o las específicas por id

Debe permitir filtrar clases por instructor

CU11:

El sistema debe retornar todos los horarios disponibles o las específicas por id

Cada horario debe estar asociado a una clase válida

CU12:

El sistema debe verificar que el usuario existe y este activo (IsActive =1)

Debe verificar que el horario corresponda a una clase válida

No debe permitir registros duplicados

Debe validar que la clase no hay alcanzado su capacidad máxima

Debe confirmar que el usuario tenga clases restantes en su membresía y que esta esté activa

Debe verificar que la fecha de asistencias esté dentro del periodo de vigencia de la membresía

Al registrar se debe reducir en 1 las clases disponibles de la membresía

No se puede registrar asistencias en fechas futuras

CU13:

El sistema debe retornar todas las asistencia registradas o las específicas mediante id

CU14:

El sistema debe validar que la clase no hay alcanzado su capacidad máxima establecida

CU15:

El sistema debe verificar que el usuario tenga clases disponibles

Debe validar que la membresía esté vigente

CU16:

El sistema debe evitar que un usuario se registre más de una vez a la misma clase

6.2-Reglas del negocio

6.2.1-Respecto a los usuarios:

- Los usuarios que no están activos no pueden registrar asistencias (IsActive = 0).

Marcar asistencia a clases está reservado solamente para los usuarios que tienen su estado activo (IsActive = 1).

- Necesidad de una edad mínima

Los usuarios deben tener más de 10 años para poder registrarse en el sistema.

- Reactivación por suscripción

El estado de un usuario se convierte automáticamente en activo cuando se le asigna una membresía.

6.2.2-Respecto a las asistencias:

- Control de duplicaciones

Un usuario no tiene la posibilidad de marcar asistencia más de una vez por clase en un mismo día.

- Verificación de capacidad

No se pueden inscribir más alumnos de los que el límite máximo por clase permite.

- Verificación de las clases existentes

El usuario solo puede registrar asistencia si tiene clases disponibles en su membresía vigente.

- Actualización del saldo

Cuando se registra una asistencia, disminuye el número de clases restantes en la membresía del usuario

6.2.3-Respecto a horarios:

- Relación obligatoria con clases

Todos los horarios del sistema deben estar asociados a una clase específica