

# Proyecto Final: GimnasioDB - Figueredo

## ## Introducción

Este proyecto busca desarrollar una base de datos relacional para un sistema de gestión de gimnasios. La solución propuesta abarca la administración integral de socios, entrenadores, clases, membresías y pagos, incluyendo reportes analíticos que permiten extraer información clave para la toma de decisiones.

La implementación técnica incluye la creación de un esquema, tablas relacionales con claves primarias y foráneas, vistas, funciones, procedimientos almacenados, triggers y un proceso de carga de datos. Adicionalmente, se desarrollan consultas para generar informes a partir de los datos almacenados.

---

## ## Objetivo

El objetivo principal es crear una base de datos robusta y escalable que permita al gimnasio:

- Gestionar eficientemente a sus socios, entrenadores y clases.
- Automatizar procesos como la renovación de membresías y el registro de asistencias.
- Generar informes que faciliten el análisis de métricas clave, como niveles de ocupación de clases y estado de membresías.

---

## ## Situación problemática

El gimnasio actualmente gestiona sus operaciones a través de sistemas dispersos, lo que dificulta el acceso a información actualizada y precisa. Por ejemplo, no existe un control unificado para el registro de asistencias o la renovación de membresías, lo que genera errores y retrabajos.

Con la implementación de la base de datos relacional, se busca centralizar la información y reducir estas brechas operativas.

---

## ## Modelo de negocio

El modelo de negocio del gimnasio se centra en la oferta de clases y servicios personalizados a través de entrenadores calificados. Los ingresos provienen principalmente de membresías y pagos por servicios adicionales. La base de datos debe reflejar esta estructura, conectando socios, entrenadores, clases, asistencias y pagos de manera eficiente.

---

## ## Diagrama E-R

El siguiente diagrama representa el modelo entidad-relación (E-R) del sistema propuesto, incluyendo todas las entidades, atributos y relaciones definidas:

---

## ## Listado de tablas con descripción de su estructura

### ### 1. Tabla: Socios

- **\*\*Descripción:\*\*** Contiene información básica de los socios registrados.
- **\*\*Estructura:\*\***
  - IDSocio (INT, PK, AUTO\_INCREMENT): Identificador único del socio.
  - Nombre (VARCHAR(50)): Nombre del socio.
  - Apellido (VARCHAR(50)): Apellido del socio.
  - DNI (VARCHAR(10), UNIQUE): Documento de identidad del socio.
  - FechaNacimiento (DATE): Fecha de nacimiento del socio.

### ### 2. Tabla: Membresías

- **\*\*Descripción:\*\*** Registra las membresías adquiridas por los socios.
- **\*\*Estructura:\*\***
  - IDMembresia (INT, PK, AUTO\_INCREMENT): Identificador único de la membresía.
  - IDSocio (INT, FK): Identificador del socio relacionado.
  - TipoMembresia (VARCHAR(50)): Tipo de membresía adquirida.
  - Precio (DECIMAL(10, 2)): Costo de la membresía.
  - FechaVencimiento (DATE): Fecha de vencimiento de la membresía.

### ### 3. Tabla: Entrenadores

- **\*\*Descripción:\*\*** Almacena información de los entrenadores del gimnasio.
- **\*\*Estructura:\*\***
  - IDEntrenador (INT, PK, AUTO\_INCREMENT): Identificador único del entrenador.

- Nombre (VARCHAR(50)): Nombre del entrenador.
- Apellido (VARCHAR(50)): Apellido del entrenador.
- DNI (VARCHAR(10), UNIQUE): Documento de identidad del entrenador.

#### ### 4. Tabla: Clases

- **\*\*Descripción:\*\*** Registra las clases disponibles en el gimnasio.
- **\*\*Estructura:\*\***
  - IDClase (INT, PK, AUTO\_INCREMENT): Identificador único de la clase.
  - NombreClase (VARCHAR(50)): Nombre de la clase.
  - IDEntrenador (INT, FK): Identificador del entrenador asignado.
  - Horario (DATETIME): Fecha y hora de la clase.

#### ### 5. Tabla: Asistencias

- **\*\*Descripción:\*\*** Registra las asistencias de los socios a las clases.
- **\*\*Estructura:\*\***
  - IDAsistencia (INT, PK, AUTO\_INCREMENT): Identificador único de la asistencia.
  - IDClase (INT, FK): Identificador de la clase asistida.
  - IDSocio (INT, FK): Identificador del socio.
  - Fecha (DATETIME): Fecha y hora de la asistencia.

---

#### ## Scripts de creación de cada objeto de la base de datos

Se incluyen en un archivo .sql los scripts de creación de:

- Tablas
- Vistas
- Funciones
- Procedimientos almacenados
- Triggers

**\*\*Enlace al archivo SQL en GitHub:\*\***

[<https://github.com/Sofia-Figueroa/Coder---SQL>](<https://github.com/Sofia-Figueroa/Coder---SQL>)

---

## ## Scripts de inserción de datos

El archivo .sql también contiene scripts para la carga inicial de datos de prueba en cada tabla. Estos datos han sido diseñados para validar las relaciones y funcionalidad del sistema.

---

## ## Informes generados

### ### Informe 1: Socios activos

- **\*\*Consulta:\*\***

```sql

SELECT \* FROM V\_SociosActivos;

```

- **\*\*Descripción:\*\*** Proporciona una lista de socios con membresías vigentes.

### ### Informe 2: Clases con más asistentes

- **\*\*Consulta:\*\***

```sql

```
SELECT NombreClase, CantidadAsistentes FROM V_ClasesPorEntrenador ORDER BY  
CantidadAsistentes DESC;
```

```

- **\*\*Descripción:\*\*** Identifica las clases más concurridas.

---

## ## Herramientas y tecnologías utilizadas

- **\*\*MySQL:\*\*** Motor de base de datos.
- **\*\*DBeaver:\*\*** Interfaz gráfica para diseño y pruebas.
- **\*\*Lucidchart:\*\*** Creación del diagrama E-R.
- **\*\*GitHub:\*\*** Gestión de versiones y almacenamiento del proyecto.

---

## ## Conclusión

El proyecto final entrega una solución completa para la gestión de gimnasios, centralizando y optimizando procesos clave. La base de datos ha sido diseñada para ser escalable y adaptable a futuras necesidades del negocio.