## **Tareas a realizar en la FE de BBDD**

**PARTE 1 :** Planteamiento del problema

Hoy en día, muchos gimnasios siguen gestionando sus actividades de forma manual o con sistemas poco eficientes. Esto puede provocar errores, retrasos o incluso pérdida de información importante, como inscripciones, pagos o asistencia a clases.

El objetivo de este proyecto es diseñar una base de datos sencilla pero funcional que permita a un gimnasio llevar un control básico de sus socios, las clases que se imparten, los monitores, los pagos y las inscripciones. La idea es que esta base de datos pueda ser usada como base para una futura aplicación web.

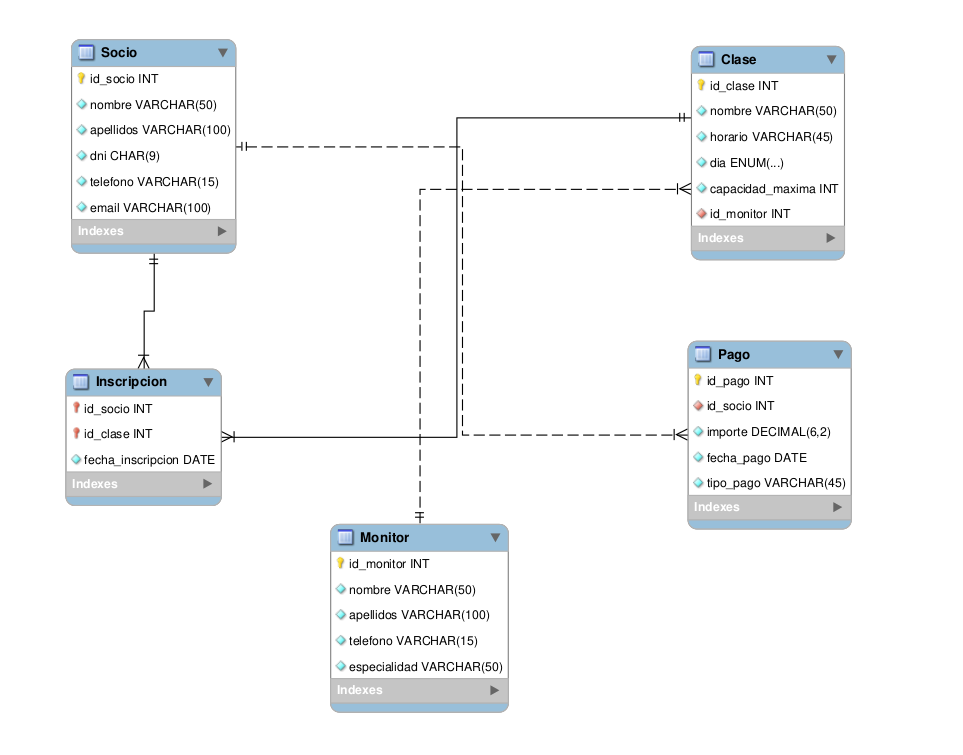
Este trabajo sirve para poner en práctica lo aprendido durante el curso, desde el análisis del problema hasta la creación e implementación de la base de datos. Además, puede ser útil para entender mejor cómo funciona el trabajo real de un analista de datos, y practicar con herramientas como **MySQL Workbench**

**PARTE 1 :** Descripción del problema con sus requerimientos.

La base de datos debe cubrir las necesidades básicas de un gimnasio pequeño. En concreto, se tienen en cuenta los siguientes puntos:

* **Socios:** Hay que guardar los datos personales de los socios: nombre, apellidos, DNI, teléfono, correo electrónico, etc.
* **Clases**: El gimnasio ofrece distintas clases (como yoga, zumba, pilates, etc.). Se debe registrar el nombre de la clase, el horario, la capacidad máxima y el monitor que la imparte.
* **Inscripciones**: Los socios se pueden apuntar a varias clases. Es necesario guardar quién se apunta a qué clase y en qué fecha.
* **Pagos**: Cada socio puede pagar de forma mensual o por clase. Hay que guardar el tipo de pago, el importe y la fecha.
* **Monitores**: Cada clase tiene un monitor asignado. Se deben guardar sus datos básicos y su especialidad.

**PARTE 2:** Diagrama completo E/R del problema utilizando MySQL Workbench



**PARTE 2 :** Descripción de los atributos de cada entidad con sus dominios

### **1. SOCIO**

* **id\_socio**: INT - **PK**, **NN**, **AI**
  + Clave primaria única para cada socio. Es un campo auto-incremental, lo que significa que MySQL lo asigna automáticamente cuando se crea un nuevo socio.
* **nombre**: VARCHAR(50) - **NN**
  + El nombre del socio. No puede ser nulo.
* **apellidos**: VARCHAR(100) - **NN**
  + Los apellidos del socio. Tampoco puede ser nulo.
* **dni**: CHAR(9) - **NN**, **UQ**
  + El DNI del socio. Debe ser único (no se puede repetir) y no puede ser nulo.
* **telefono**: VARCHAR(15) - **NN**
  + Número de teléfono del socio. Necesario para contacto, no puede ser nulo.
* **email**: VARCHAR(100) - **NN**, **UQ**
  + El correo electrónico del socio. Es único para cada socio y no puede ser nulo.

### **2. CLASE**

* **id\_clase**: INT - **PK**, **NN**, **AI**
  + Identificador único de la clase. Se genera automáticamente, no puede ser nulo y es la clave primaria.
* **nombre**: VARCHAR(50) - **NN**
  + El nombre de la clase, como "Spinning" o "Yoga". No puede ser nulo.
* **horario**: TIME - **NN**
  + El horario en el que se imparte la clase. No puede faltar.
* **dia**: ENUM('Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado') - **NN**
  + Día en el que se ofrece la clase. Solo puede ser uno de los valores predefinidos y es obligatorio.
* **capacidad\_maxima**: INT - **NN**
  + Número máximo de personas que pueden asistir a la clase. Necesario para controlar el aforo.
* **id\_monitor**: INT - **NN**, **FK** → Monitor(id\_monitor)  
  + Este es un campo de clave foránea que se refiere al monitor que imparte la clase. Es obligatorio y debe estar relacionado con un monitor existente.

### **3. INSCRIPCIÓN (Relación muchos a muchos entre Socios y Clases)**

* **id\_socio**: INT - **NN**, **PK**, **FK** → Socio(id\_socio)  
  + Hace referencia al socio que se inscribe. Es la clave foránea que relaciona la inscripción con el socio correspondiente.
* **id\_clase**: INT - **NN**, **PK**, **FK** → Clase(id\_clase)  
  + Hace referencia a la clase en la que el socio se inscribe. Es la clave foránea que conecta la inscripción con la clase específica.
* **fecha\_inscripcion**: DATE - **NN**
  + Fecha en la que el socio se inscribe en la clase. Es obligatorio.

**Nota**: La relación de inscripción es muchos a muchos, lo que significa que un socio puede estar inscrito en varias clases y una clase puede tener muchos socios. Esto se resuelve con una tabla intermedia (Inscripción) que tiene una clave primaria compuesta por **id\_socio** e **id\_clase**.

### **4. PAGO**

* **id\_pago**: INT - **PK**, **NN**, **AI**
  + Identificador único para cada pago. Se auto-incrementa y es la clave primaria.
* **id\_socio**: INT - **NN**, **FK** → Socio(id\_socio)  
  + Hace referencia al socio que realiza el pago. Es obligatorio.
* **importe**: DECIMAL(6,2) - **NN**
  + Monto del pago. No puede faltar y tiene dos decimales para manejar los centavos.
* **fecha\_pago**: DATE - **NN**
  + La fecha en la que se realiza el pago. Es obligatorio.
* **tipo\_pago**: ENUM('Mensual', 'Por clase') - **NN**
  + Tipo de pago: puede ser mensual o por clase. Es obligatorio.

### **5. MONITOR**

* **id\_monitor**: INT - **PK**, **NN**, **AI**
  + Identificador único para cada monitor. Es la clave primaria y se genera automáticamente.
* **nombre**: VARCHAR(50) - **NN**
  + El nombre del monitor. No puede ser nulo.
* **apellidos**: VARCHAR(100) - **NN**
  + Los apellidos del monitor. No puede faltar.
* **telefono**: VARCHAR(15) - **NN**
  + El teléfono del monitor. Necesario para contacto.
* **especialidad**: VARCHAR(50) - **NN**
  + La especialidad del monitor (por ejemplo, "Yoga", "Spinning"). Es obligatorio.

RESTRICCIONES USADAS:

**PK (Clave primaria)**: Asegura que el atributo sea único y no se repita en la tabla.

**NN (No Nulo)**: Indica que el atributo no puede tener un valor nulo (siempre tiene que haber un dato).

**AI (Auto Incremental)**: El valor del atributo se genera automáticamente, como en el caso de las claves primarias.

**UQ (Único)**: Garantiza que el valor del atributo sea único en la tabla (por ejemplo, DNI o correo electrónico).

**PARTE 3 :** Modelo Relacional

**PARTE 3 :** Modelo Relacional

#### **2. Tabla:** Socio

| **Atributo** | **Tipo** | **Restricciones** |
| --- | --- | --- |
| id\_socio | INT | PK, NN, AI |
| nombre | VARCHAR(50) | NN |
| apellidos | VARCHAR(100) | NN |
| dni | CHAR(9) | NN, UQ |
| telefono | VARCHAR(15) | NN |
| email | VARCHAR(100) | NN, UQ |

#### **2. Tabla:** Monitor

| **Atributo** | **Tipo** | **Restricciones** |
| --- | --- | --- |
| id\_monitor | INT | PK, NN, AI |
| nombre | VARCHAR(50) | NN |
| apellidos | VARCHAR(100) | NN |
| telefono | VARCHAR(15) | NN |
| especialidad | VARCHAR(50) | NN |

#### **3. Tabla:** Clase

| **Atributo** | **Tipo** | **Restricciones** |
| --- | --- | --- |
| id\_clase | INT | PK, NN, AI |
| nombre | VARCHAR(50) | NN |
| horario | TIME | NN |
| dia | ENUM('Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado') | NN |
| capacidad\_maxima | INT | NN |
| id\_monitor | INT | NN, FK → Monitor(id\_monitor) |

#### **4. Tabla:** Inscripción

| **Atributo** | **Tipo** | **Restricciones** |
| --- | --- | --- |
| id\_socio | INT | NN, PK, FK → Socio(id\_socio) |
| id\_clase | INT | NN, PK, FK → Clase(id\_clase) |
| fecha\_inscripcion | DATE | NN |

#### **5. Tabla:** Pago

| **Atributo** | **Tipo** | **Restricciones** |
| --- | --- | --- |
| id\_pago | INT | PK, NN, AI |
| id\_socio | INT | NN, FK → Socio(id\_socio) |
| importe | DECIMAL(6,2) | NN |
| fecha\_pago | DATE | NN |
| tipo\_pago | ENUM('Mensual','Por clase') | NN |