

# GIMNASIO

## PROYECTO FINAL

BALERDI LEONARDO AUGUSTO

Profesor: Lo Presti Leonel

Tutor: Goldfeld Jennifer

Curso: SQL Flex

# Índice

Introducción	3
Objetivo	3
Situación Problemática	3
Tablas	3
Relación entre tablas	5
Diagrama Entidad-Relación (DER)	7
Vistas	7
Funciones	8
Stored Procedures	9

## Introducción:

En la actualidad, la gestión eficiente de la información es fundamental para el funcionamiento de cualquier organización. En el caso de los gimnasios, el manejo ordenado de datos sobre socios, instructores, ejercicios, abonos y servicios resulta esencial para optimizar los procesos internos y ofrecer una mejor atención a los clientes.

## Objetivo:

Diseñar y desarrollar una base de datos representativa de un gimnasio, permitiendo centralizar y organizar la información de manera segura, precisa y accesible. Mediante esta herramienta, se busca facilitar tareas administrativas, mejorar la toma de decisiones y garantizar un control más efectivo de las actividades diarias del establecimiento.

## Situación Problemática:

En la actualidad, los gimnasios enfrentan el desafío de administrar grandes volúmenes de información relacionados con socios, personal (instructores, empleados), pagos, rutinas y servicios. Muchas veces, esta gestión se realiza mediante planillas manuales, archivos dispersos o sistemas poco integrados, lo que provoca ineficiencias y errores. La implementación de una base de datos en un gimnasio permite centralizar y organizar la información de manera segura, reduciendo errores y tiempos administrativos. Esto contribuye a mejorar la eficiencia interna, optimizar la gestión del personal y ofrecer un servicio más personalizado a los socios.

## Listado de Tablas:

<b>Tabla</b>	<b>Clave Primaria</b>	<b>Atributos Destacados / Claves Foráneas</b>
<b>Socio</b>	Idsocio (PK)	Idsucursal (FK), Nombre, Apellido, DNI, Dirección, Edad, Teléfono, Correo, TeléfonoEmergencia
<b>Instructor</b>	Idinstructor (PK)	Idsucursal (FK), Nombre, Apellido, DNI, Dirección, Edad, Teléfono, Correo
<b>Empleado</b>	Idempleado (PK)	Idsucursal (FK), Nombre, Apellido, DNI, Dirección, Edad, Teléfono, Correo

<b>Tabla</b>	<b>Clave Primaria</b>	<b>Atributos Destacados / Claves Foráneas</b>
<b>Sucursal</b>	Idsucursal (PK)	Nombre, Dirección, Teléfono, Ciudad
<b>Clase</b>	Idclase (PK)	Idzona (FK), Idsucursal (FK), Descripción
<b>Zona</b>	Idzona (PK)	Descripción, Capacidad
<b>HorarioClase</b>	Idhorarioclase (PK)	Idclase (FK), DiaSemana, Horalnicio, HoraFin
<b>ReservaClase</b>	Idreserva (PK)	Idsocio (FK), Idclase (FK), Fechareserva, Cancelada
<b>PlanAbono</b>	Idplan (PK)	Descripción, DíaN, Precio
<b>PlanSocio</b>	Idplansocio (PK)	Idsocio (FK), Idplan (FK), FechaCompra, FechaLimite
<b>PlaneEntrenamiento</b>	Idplanentrenamiento (PK)	Idobjetivoentrenamiento (FK), días de la semana (Lunes a Domingo)
<b>ObjetivoEntrenamiento</b>	Idobjetivoentrenamiento (PK)	Descripción
<b>Ejercicio</b>	Idejercicio (PK)	Idplanentrenamiento (FK), Descripción, Observaciones
<b>EjercicioSocio</b>	Idejerciciosocio (PK)	Idsocio (FK), Idobjetivoentrenamiento (FK)
<b>FichaSocio</b>	Idficha (PK)	Idsocio (FK), Idinstructor (FK), Peso, Altura, IMC, medidas corporales
<b>Venta</b>	Idventa (PK)	Idsocio (FK), Fecha
<b>DetalleVenta</b>	Idventadetalle (PK)	Idventa (FK), Idproducto (FK), Idlocal (FK), Descripción, Precio, Cantidad
<b>Producto</b>	Idproducto (PK)	Descripción, Precio, Stock

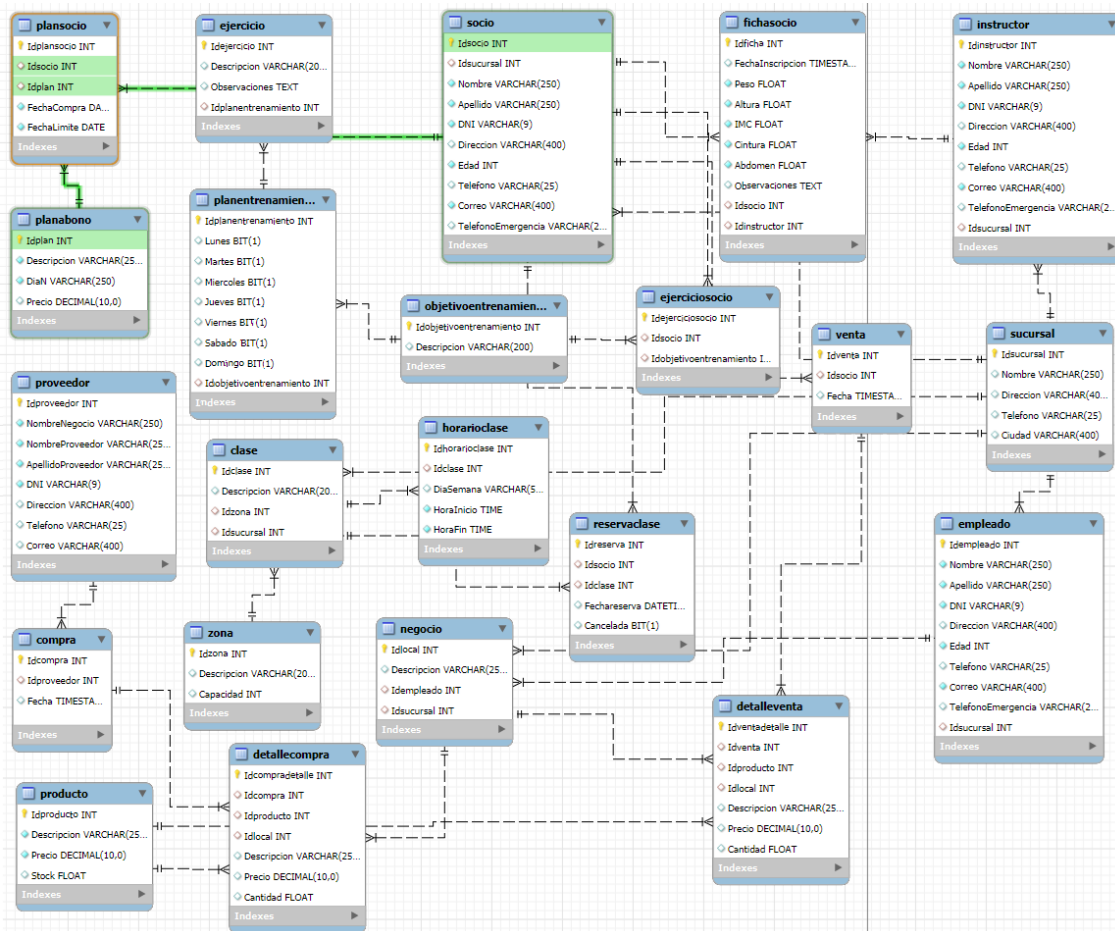
Tabla	Clave Primaria	Atributos Destacados / Claves Foráneas
<b>Compra</b>	Idcompra (PK)	Idproveedor (FK), Fecha
<b>DetalleCompra</b>	Idcompradetalle (PK)	Idcompra (FK), Idproducto (FK), Idlocal (FK), Descripción, Precio, Cantidad
<b>Proveedor</b>	Idproveedor (PK)	NombreNegocio, NombreProveedor, ApellidoProveedor, DNI, Dirección, Teléfono, Correo
<b>Negocio</b>	Idlocal (PK)	Idempleado (FK), Idsucursal (FK), Descripción

### Relaciones Entre Tablas (Claves Foráneas):

- **Socio – Sucursal (1:N)**: un socio puede ir a una sucursal, pero una sucursal puede tener varios socios.
- **Instructor – Sucursal (1:N)**: cada instructor está asignado a una sucursal, pero cada sucursal cuenta con varios instructores.
- **Empleado – Sucursal (1:N)**: cada empleado trabaja en una sucursal, cada sucursal tiene varios empleados.
- **Clase – Zona y Sucursal (1:N)**: cada clase se dicta en una zona específica y en una sucursal determinada, pero en cada zona se pueden impartir varias clases.
- **HorarioClase – Clase (1:N)**: cada clase se dicta en un horario específico, pero una clase se puede dar en varios horarios.
- **ReservaClase – Socio y Clase (1:N)**: Un socio solo puede registrar una reserva para una misma clase, pero esa misma clase puede ser reservada por uno o muchos socios.
- **PlanSocio – Socio y PlanAbono (1:N)**: Un socio solo va a adquirir un plan a la vez, pero el mismo plan puede ser adquirido por varios socios.
- **PlanEntrenamiento – ObjetivoEntrenamiento (1:N)**: cada plan se asocia a un objetivo de entrenamiento, pero un objetivo de entrenamiento puede tener varios planes.

- **Ejercicio – PlanEntrenamiento (N:N):** un ejercicio es asignado a muchos planes de entrenamiento y un plan de entrenamiento consta de varios ejercicios.
- **EjercicioSocio – Socio y ObjetivoEntrenamiento (N:N):** varios ejercicios son asignados a un socio según su objetivo de entrenamiento, pero varios socios pueden hacer los mismos ejercicios, ya sea con el mismo plan u otro.
- **FichaSocio – Socio e Instructor (1:N):** registro de evaluación física de un socio lo realiza un instructor, pero un instructor puede realizar varios registros físicos de uno o varios socios.
- **Venta – Socio (1:N):** cada venta está asociada a un socio y un socio puede tener una o muchas ventas asociadas.
- **DetalleVenta – Venta y Producto (y Negocio) (1:N):** cada venta tiene un único detalle de venta, en la cual puede incluir uno o varios productos y se realizad en un negocio, pero cada negocio puede tener muchas ventas que a su vez cada venta puede incluir uno o muchos productos vendidos.
- **Compra – Proveedor (1:N):** cada venta tiene un único detalle de venta y se realizad en un negocio, pero cada negocio puede tener muchas ventas que a su vez cada venta puede incluir uno o muchos productos vendidos.
- **DetalleCompra – Compra, Producto y Negocio (1:N):** cada compra tiene un único registro, en el cual puede incluir uno o varios productos y es realizado en cada local, pero cada local realiza compras, que a su vez puede incluir uno o muchos productos.
- **Negocio – Empleado y Sucursal (1:N):** cada local tiene un o varios empleado y pertenece a una sucursal y cada sucursal puede tener locales y cada local a su vez puede tener uno o muchos empleados.

## Diagrama Entidad – Relación:



## Vistas:

### Vista 1

Nombre: vista\_Objetivo\_EntrenamientoSocio

Descripción: Mostrar Apellido y Nombre de los socios y su objetivo en el gimnasio.

Objetivo: Este análisis sirve para determinar qué tipo de equipamiento (maquinarias, pesas, colchonetas, etc.) son más requeridos.

Tablas y Datos que lo componen: Socio, ObjetivoEntrenamiento, EjercicioSocio

```
CREATE VIEW vista_Objetivo_EntrenamientoSocio AS
SELECT s.Apellido, s.Nombre, o.Descripcion
FROM EjercicioSocio e
JOIN ObjetivoEntrenamiento o ON o.Idobjetivoentrenamiento = e.Idobjetivoentrenamiento
JOIN Socio s ON s.Idsocio = e.IdejercicioSocio
ORDER BY s.Apellido ASC;
```

### Vista 2

Nombre: vista\_ClasesxSucursal

Descripción: Mostrar todas las clases y en que gimnasio se imparten.

Objetivo: Brinda un panorama de las clases que se imparten en cada sucursal.

Tablas y Datos que lo componen: Clase y Sucursal

```
CREATE VIEW vista_ClasesxSucursal AS
SELECT c.Descripcion, suc.Nombre, suc.Direccion
      FROM Clase c
      LEFT JOIN Sucursal suc ON suc.Idsucursal = c.Idsucursal
      ORDER BY c.Descripcion;
```

### Vista 3

Nombre: vista\_EmpleadosxSucursal

Descripción: Muestra que el listado de empleados que trabaja en cada sucursal acomodado por alfabeto.

Objetivo: Saber cuantos empleados tiene cada Sucursal.

Tablas y Datos que lo componen: Empleado, Sucursal

```
CREATE VIEW vista_EmpleadosxSucursal AS
SELECT CONCAT(e.Apellido, " ",e.Nombre) AS Nombre_Completo, s.Nombre
      FROM Empleado e
      JOIN Sucursal s ON e.Idsucursal = s.Idsucursal
      ORDER BY Nombre_Completo ASC;
```

### Vista 4

Nombre: vista\_ClasesyCapacidad

Descripción: Mostrar las clases con el nombre del sector del gimnasio, que cupo de inscripción tiene cada clase y en que sucursal se dicta y ordenar por nombre de la clase.

Objetivo: Saber si los espacios asignados a cada clase son correctos o si conviene cambiarlos a un espacio con mayor capacidad o viceversa.

Tablas y Datos que lo componen: Clase, Sucursal, Zona

```
CREATE VIEW vista_ClasesyCapacidad AS
SELECT c.Descripcion AS Clase, z.Descripcion AS Espacio, z.Capacidad, suc.Nombre
      FROM Clase c
      JOIN Zona z ON c.Idzona = z.Idzona
      JOIN Sucursal suc ON c.Idsucursal = suc.Idsucursal
      ORDER BY Clase;
```

## Funciones:

### Función 1

Nombre: EstadoPlanAbono

Descripción: Muestra si el socio esta activo o si se termino su plan y debe abonar un nuevo plan.

Objetivo: Agilizar la carga de datos, solo se debe cargar el Id del socio y con la fecha del día te trae el resultado.

Tablas y Datos que lo componen Idsocio INT, FechaHoy DATE

Ejemplo:



```

DELIMITER //
CREATE FUNCTION EstadoPlanAbono(Id socio INT, FechaHoy DATE)
RETURNS Varchar (50)
DETERMINISTIC
BEGIN
    DECLARE estado VARCHAR(20);
    SELECT IF(MAX(FechaLimite) >= FechaHoy, "Activo", "Vencido")
    INTO estado
    FROM PlanSocio
    WHERE idsocio = idsocio;

    RETURN estado;
END //
DELIMITER ;

```

## Función 2

Nombre: CalcularIMC

Descripción: Realiza el cálculo del índice de Masa Corporal de los clientes.

Objetivo: Agilizar la carga de datos.

Tablas y Datos que lo componen: Peso FLOAT, Altura FLOAT

Ejemplo:

```

DELIMITER //
CREATE FUNCTION CalcularIMC(Peso FLOAT, Altura FLOAT)
RETURNS FLOAT
DETERMINISTIC
BEGIN
    RETURN Peso / (Altura * Altura);
END //
DELIMITER ;

```

## Stores Procedures:

### SP 1

Nombre: Cargar\_Socio

Descripción: Al ingresar los datos de los socios los carga automáticamente a la base de datos.

Objetivo: Agilizar la carga de socios

Tablas y Datos que lo componen: Nombre VARCHAR(250), Apellido VARCHAR(250), DNI VARCHAR (9), Dirección VARCHAR(400), Edad INT, Teléfono VARCHAR(25), Correo VARCHAR(400), TeléfonoEmergencia VARCHAR(25))

Ejemplo:

```

DELIMITER //
CREATE PROCEDURE Cargar_Socio(
    IN Nombre VARCHAR(250),
    IN Apellido VARCHAR(250),
    IN DNI VARCHAR (9),
    IN Direccion VARCHAR(400),
    IN Edad INT,
    IN Telefono VARCHAR(25),
    IN Correo VARCHAR(400),
    IN TelefonoEmergencia VARCHAR(25))
BEGIN
    INSERT INTO Socio (Nombre, Apellido, DNI, Direccion, Edad, Telefono, Correo, TelefonoEmergencia)
    VALUES (Nombre, Apellido, DNI, Direccion, Edad, Telefono, Correo, TelefonoEmergencia);
END //
DELIMITER ;

```

## SP 2

Nombre: Cargar\_Proveedor

Descripción: Al ingresar los datos de los proveedores los carga automáticamente a la base de datos.

Objetivo: Agilizar la carga de proveedores

Tablas y Datos que lo componen: NombreProveedor VARCHAR(250), ApellidoProveedor VARCHAR(250), DNI VARCHAR(9), Dirección VARCHAR(400), Teléfono VARCHAR(25), Correo VARCHAR(400))

Ejemplo:

```

DELIMITER //
CREATE PROCEDURE Cargar_Proveedor(
    IN NombreNegocio VARCHAR(250),
    IN NombreProveedor VARCHAR(250),
    IN ApellidoProveedor VARCHAR(250),
    IN DNI VARCHAR(9),
    IN Direccion VARCHAR(400),
    IN Telefono VARCHAR(25),
    IN Correo VARCHAR(400))
BEGIN
    INSERT INTO Proveedor (NombreNegocio, NombreProveedor, ApellidoProveedor, DNI, Direccion, Telefono, Correo)
    VALUES (NombreNegocio, NombreProveedor, ApellidoProveedor, DNI, Direccion, Telefono, Correo);
END //
DELIMITER ;

```