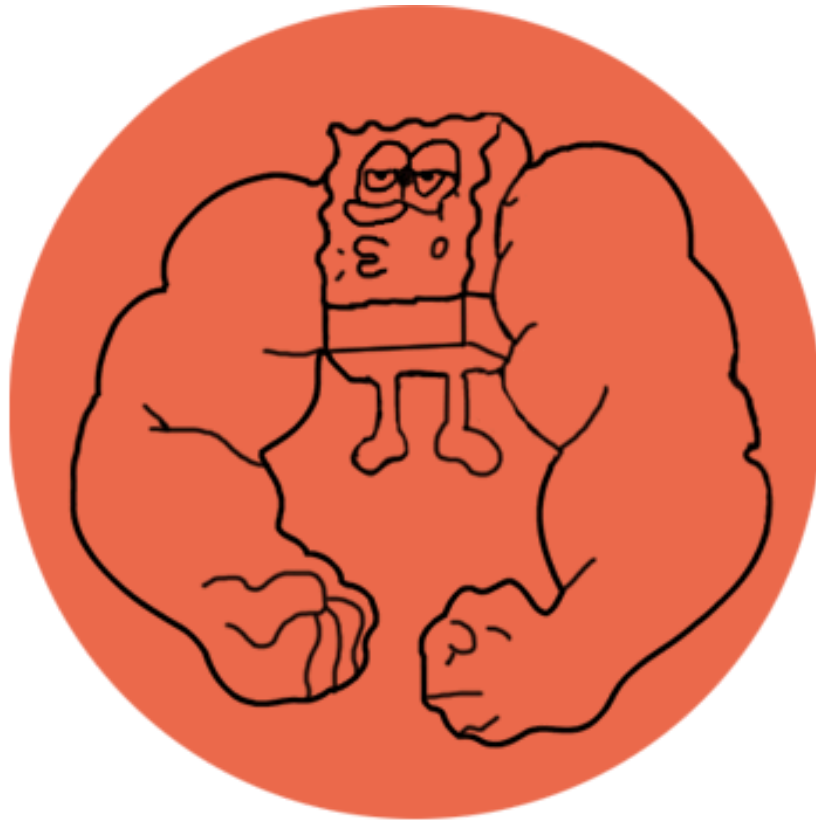


# Proyecto de Entornos de Desarrollo.



# MAZAOGYM

**Profesor:** Luis Pilo Aceituno

# ÍNDICE

Introducción al proyecto

- Equipo y herramientas de desarrollo
- Programación
- Desarrollo Web
- Base de Datos
- Estructuras

# INTRODUCCIÓN AL PROYECTO

El proyecto del sitio web "**MazaoGym**" se centra en proporcionar una plataforma interactiva donde los usuarios puedan acceder a información relevante sobre el gimnasio, como horarios de clases, detalles de entrenamientos, y datos importantes. Además, permitirá a los usuarios inscribirse en clases y entrenamientos de manera conveniente y eficiente.

## Objetivos del Proyecto:

El principal objetivo de "**MazaoGym**" es proporcionar una plataforma centrada en el usuario que permita a los clientes y al personal del gimnasio acceder y administrar información de manera eficiente. Algunos objetivos específicos incluyen:

1. **Facilitar la inscripción:** Permitir a los usuarios registrarse en clases y entrenamientos a través de la plataforma en línea.
2. **Proporcionar información detallada:** Ofrecer horarios de clases, programas de entrenamiento, y perfiles de los empleados del gimnasio.
3. **Gestión de perfiles de usuarios:** Permitir a los clientes gestionar sus perfiles personales y actualizar su información según sea necesario.
4. **Optimización de la gestión interna:** Facilitar a los empleados del gimnasio el acceso y la gestión de la información de los clientes, así como la administración de las clases y entrenamientos.

# EQUIPO Y HERRAMIENTAS DE DESARROLLO

## **Equipo del proyecto:**

Jefe de Proyecto: Ismael Pavón

Analistas de Sistemas: Kevin González y David Roncero

Diseñador de Interfaz de Usuario (UI/UX): Yago Catalano

Desarrollador Frontend: Yago Catalano

Desarrollador Backend: Ismael Pavón

Gestor de Base de Datos: Alejandro Gonzalez

Tester: Abel Jimenez

Reunión de Planificación Inicial: El equipo se reúne para discutir los requisitos del cliente y definir los objetivos del proyecto.

## **Roles y Responsabilidades:**

Jefe de Proyecto: Coordina el equipo, gestiona el alcance y el calendario del proyecto, y se comunica con el cliente.

Desarrollo de Prototipos: El diseñador de UI/UX crea prototipos de la aplicación para su revisión por parte del equipo. Mediante herramientas gratuitas como Figma, el diseñador ha realizado el diseño al completo de las páginas.

Pruebas y Correcciones: se realizan pruebas de la aplicación para identificar errores y problemas de usabilidad. El equipo corrige y mejora la aplicación en base a los resultados de las pruebas.

Analistas de Sistemas: Analizan los requisitos del cliente y elabora la documentación técnica.

Diseñador de Interfaz de Usuario: Diseña la interfaz de usuario de la aplicación y crea prototipos.

Desarrollador Frontend: Implementa la interfaz de usuario de la aplicación utilizando tecnologías web.

Desarrollador Backend: Desarrolla la lógica de negocio y la funcionalidad del servidor.

Testers: Realizan pruebas de la aplicación y reportan errores para su corrección.

### **Herramientas utilizadas en el proyecto:**

Frontend: HTML5, CSS3, JavaScript.

Backend: MySQL, phpMyAdmin.

Herramientas de Desarrollo: GitHub, VS Code.

# PROGRAMACIÓN

## USUARIO

### - **Formulario.php**

Este archivo PHP se encarga de gestionar las operaciones relacionadas con los clientes del gimnasio en la base de datos. Utiliza los datos recibidos desde el formulario para realizar acciones como insertar, actualizar, borrar y consultar información de clientes, así como también consultar las clases y entrenamientos en los que están inscritos. Además, genera respuestas HTML dinámicas para mostrar los resultados de estas operaciones en la interfaz de usuario.

## ADMINISTRADOR

- **registrarEmpleados.php:** Gestiona el proceso de registro de nuevos empleados en el sistema.
- **consultarEmpleados.php:** Permite consultar información sobre los empleados registrados en el sistema.
- **consultarClientes.php:** Proporciona la capacidad de consultar información sobre los clientes registrados en el sistema.
- **consultarClasesYEntrenamientos.php:** Permite consultar información sobre las clases y entrenamientos disponibles en el sistema.
- **borrarEmpleados.php:** Gestiona el proceso de eliminación de empleados del sistema.

## TODOS

### - **Registro.php**

Este archivo PHP maneja el inicio de sesión de los usuarios. Verifica las credenciales ingresadas en el formulario de inicio de sesión, buscando coincidencias en las tablas de clientes y empleados. Dependiendo de las coincidencias encontradas, redirige al usuario a diferentes páginas del sitio web.

# DESARROLLO WEB

El desarrollo web del proyecto "**MazaoGym**" se enfoca en crear una interfaz de usuario **atractiva y funcional** que permita a los usuarios interactuar con la plataforma de manera intuitiva y eficiente. Se utiliza una combinación de HTML, CSS y JavaScript para diseñar y desarrollar las diferentes páginas y funcionalidades del sitio web.

## Páginas:

1. **Página de inicio (index.html):** Esta página proporciona una introducción al gimnasio y destaca sus características principales. Incluye secciones como información sobre el gimnasio, horarios de apertura, testimonios de clientes, y una llamada a la acción para registrarse o iniciar sesión.
2. **Página de registro (registro.html):** Aquí los usuarios pueden registrarse como nuevos clientes del gimnasio. El formulario de registro solicita información personal como nombre, apellidos, dirección, información de contacto, y crea un nombre de usuario único para acceder al sistema.
3. **Página de inicio de sesión (login.html):** Permite a los usuarios iniciar sesión en sus cuentas. Se solicita el nombre de usuario y la contraseña para acceder a las funciones personalizadas del sitio web.
4. **Página de perfil (perfil.html):** Después de iniciar sesión, los usuarios pueden acceder a su perfil personal donde pueden ver y actualizar su información, ver las clases y entrenamientos en los que están inscritos, y administrar su cuenta.
5. **Página de clases (clases.html):** Muestra una lista de todas las clases disponibles en el gimnasio, junto con detalles como el nombre de la clase, el instructor, el horario y la descripción. Los usuarios pueden inscribirse en las clases desde esta página.
6. **Página de entrenamientos (entrenamientos.html):** Similar a la página de clases, pero para los programas de entrenamiento ofrecidos por el gimnasio. Proporciona

información sobre los diferentes programas de entrenamiento y permite a los usuarios inscribirse en ellos.

7. **Página de contacto (contacto.html):** Contiene información de contacto del gimnasio, como dirección, número de teléfono y dirección de correo electrónico. También puede incluir un formulario de contacto para que los usuarios envíen consultas directamente desde el sitio web.

## **Funcionalidades:**

1. **Interfaz interactiva:** Se utiliza CSS y JavaScript para crear una interfaz de usuario interactiva y atractiva. Se pueden incluir efectos de animación y elementos desplegados para mejorar la experiencia del usuario.
2. **Validación de formularios:** Se implementa validación de formularios tanto en el lado del cliente como en el servidor para garantizar que se ingresen datos válidos.
3. **Integración con la base de datos:** Se establecen conexiones con la base de datos utilizando lenguajes como PHP para recuperar y almacenar información de manera segura y eficiente.



# BASE DE DATOS

La base de datos del proyecto "**MazaoGym**" está diseñada para gestionar eficientemente la información relacionada con los clientes, las clases, los entrenamientos y los empleados de un gimnasio. A continuación, se detalla la estructura de las tablas y su función en el sistema:

## **Tabla clientes:**

- Esta tabla almacena la información personal de los clientes del gimnasio, como su nombre, apellidos, DNI, dirección, información de contacto, pagos y foto de perfil.
- Se utiliza una restricción de clave primaria en el campo DNI para garantizar la unicidad de cada cliente en la base de datos.
- El campo NOMBRE\_USUARIO se declara como único para facilitar el acceso del cliente al sistema mediante un nombre de usuario único.
- La contraseña se almacena de forma segura utilizando una longitud de cadena extendida para aumentar la seguridad.

## **Tabla clases:**

- Esta tabla contiene información sobre las clases ofrecidas por el gimnasio, como el nombre de la clase y una descripción detallada de la misma.
- Se utiliza un identificador único (ID) como clave primaria para cada registro de clase, con incremento automático para facilitar la gestión de los registros.

## **Tabla entrenamiento:**

- Similar a la tabla de clases, esta tabla almacena información sobre los programas de entrenamiento disponibles en el gimnasio, incluyendo el nombre del entrenamiento y una descripción detallada.
- Se utiliza un identificador único (ID) como clave primaria para cada registro de entrenamiento, con incremento automático para garantizar la integridad de los datos.

**Tabla empleados:**

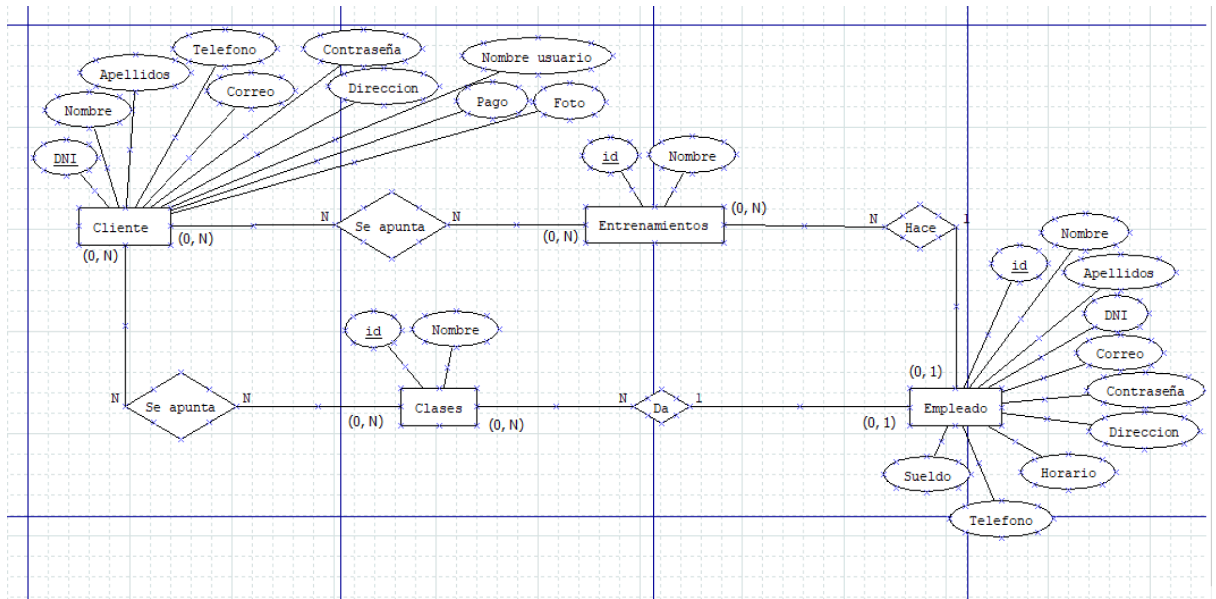
- Esta tabla mantiene registros de los empleados del gimnasio, incluyendo su nombre, apellidos, DNI, información de contacto, ubicación, sueldo, horario de trabajo y contraseña.
- Un identificador único (ID\_EMPLEADO) se utiliza como clave primaria para cada empleado, con incremento automático para facilitar la gestión de los registros.

**Tablas de relaciones:**

- Las tablas **relacion\_clientes\_clases\_entrenamiento** y **relacion\_empleados\_clases\_entrenamiento** se utilizan para establecer relaciones entre clientes y las clases/entrenamientos en los que están inscritos, así como entre empleados y las clases/entrenamientos que imparten.
- Se utilizan claves externas para garantizar la integridad referencial entre las tablas y permitir operaciones como la eliminación en cascada para mantener la consistencia de los datos.

# ESTRUCTURAS

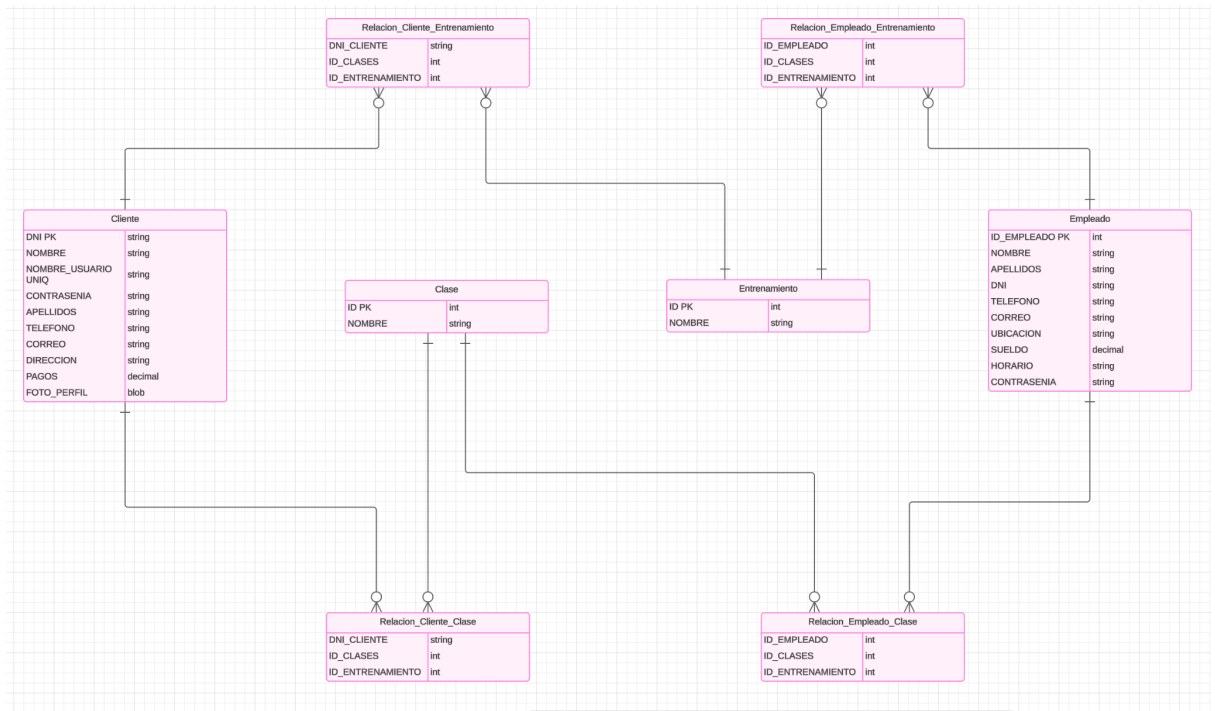
## - Modelo Entidad / Relación:



Este diagrama de Entidad Relación muestra claramente las entidades involucradas en el sistema y las relaciones entre ellas, proporcionando una visión integral de cómo se gestionan los datos de clientes, empleados, clases y entrenamientos en el gimnasio. Este diseño permite una gestión eficiente y organizada de las operaciones diarias y la interacción entre los diferentes componentes del sistema.

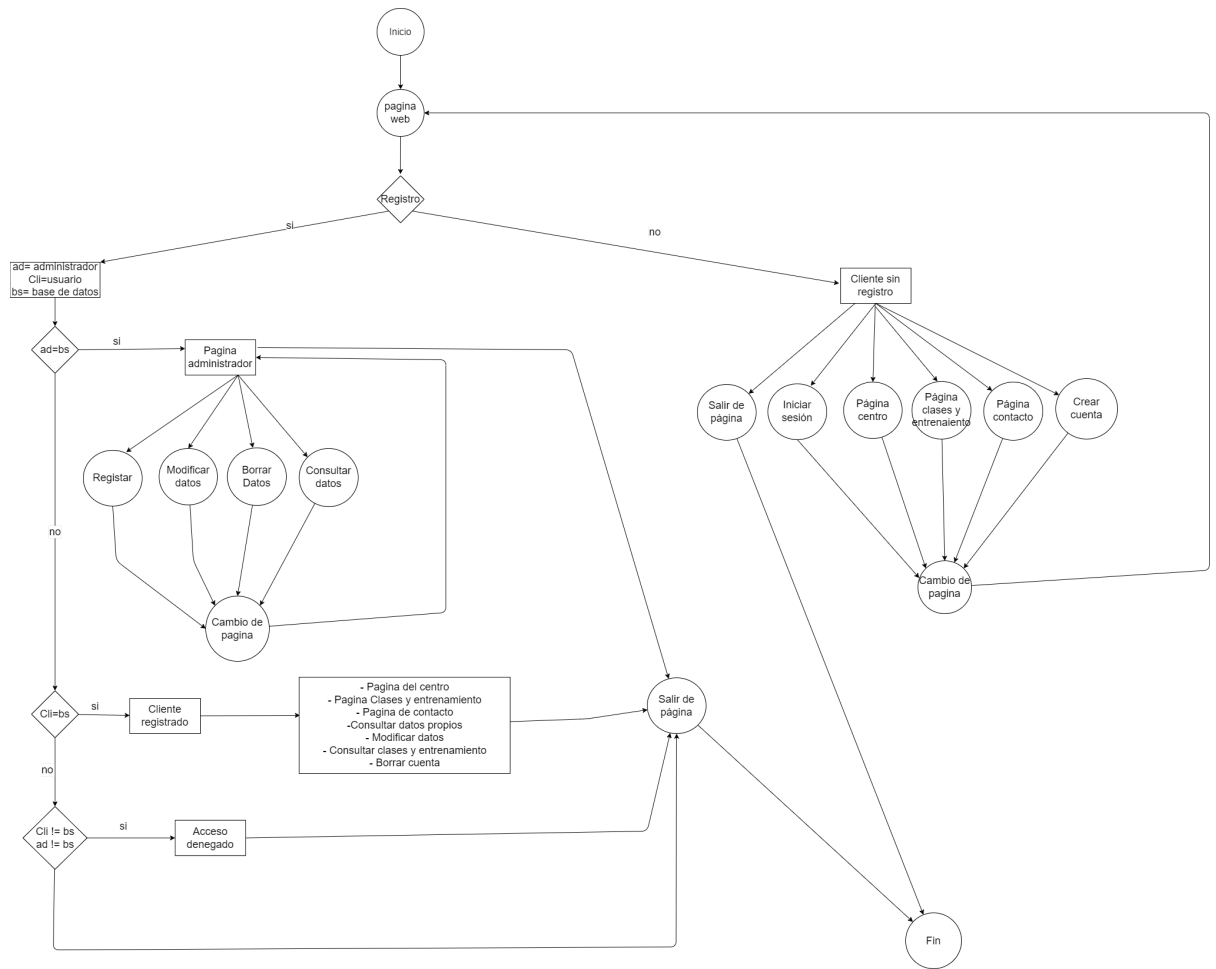
Con este diagrama, estamos preparados para implementar una base de datos que soporte las necesidades del gimnasio, asegurando una operación fluida y una experiencia óptima para los usuarios.

- **Modelo relacional:**



El modelo relacional presentado garantiza una estructura de base de datos normalizada y eficiente para gestionar las operaciones del gimnasio. Las tablas Cliente, Empleado, Clases y Entrenamientos, lo que hace que este diseño relacional facilite la implementación de un sistema robusto y escalable que soportará las necesidades operativas del gimnasio y mejorará la experiencia de los usuarios al asegurar la integridad y consistencia de los datos.

- **Diagrama de casos de prueba:**



- Diagrama de casos de uso:



## **USUARIO**

### **Algoritmo SistemaBasico**

**Definir nombre, correo, password, correoIngresado, passwordIngresado Como  
Caracter**

**Definir usuarioRegistrado Como Logico**

**usuarioRegistrado = Falso**

**// Menú principal**

**Repetir**

**Escribir "Sistema de Gestión"**

**Escribir "1. Registrar usuario"**

**Escribir "2. Iniciar sesión"**

**Escribir "3. Salir"**

**Leer opcion**

**Segun opcion Hacer**

**1:**

**Escribir "Registro de nuevo usuario"**

**Escribir "Nombre: "**

**Leer nombre**

**Escribir "Correo: "**

**Leer correo**

**Escribir "Password: "**

**Leer password**

**usuarioRegistrado <- Verdadero**

**Escribir "Registro exitoso"**

**2:**

**Escribir "Inicio de sesión"**

**Escribir "Correo: "**

**Leer correoIngresado**

**Escribir "Password: "**

**Leer passwordIngresado**

**Si usuarioRegistrado Y correo = correoIngresado Y password = passwordIngresado Entonces**

**Escribir "Bienvenido ", nombre**

**Sino**

**Escribir "Correo o password incorrectos"**

**FinSi**

**3:**

**Escribir "Saliendo del sistema..."**

**De Otro Modo:**

**Escribir "Opción no válida"**

**FinSegun**

**Hasta Que opcion = 3**

**Finalgoritmo**

## **ADMINISTRADOR**

### **Algoritmo SistemaDeGestion**

**// Definición de variables**

**Definir opcion Como Entero**

**Definir usuario, password Como Caracter**

**Definir esAdmin Como Booleano**

**Definir consulta Como Entero**

**// Menú principal**

**Repetir**

**Escribir "Sistema de Gestión"**

**Escribir "1. Iniciar sesión"**

**Escribir "2. Salir"**

**Leer opcion**

**Segun opcion Hacer**



**1:**

**Escribir "Inicio de sesión"**

**Escribir "Usuario: "**

**Leer usuario**

**Escribir "Password: "**

**Leer password**

**esAdmin <- Verdadero**

**Si esAdmin Entonces**

**Repetir**

**Escribir "Menú de Consultas"**

**Escribir "1. Consulta 1"**

**Escribir "2. Consulta 2"**

**Escribir "3. Consulta 3"**

**Escribir "4. Consulta 4"**

**Escribir "5. Salir"**

**Leer consulta**

**Segun consulta Hacer**

**1:**

**Escribir "Consulta 1: empleados"**

**2:**

**Escribir "Consulta 2: clientes"**

**3:**

**Escribir "Consulta 3: clases"**

**4:**

**Escribir "Crear empleado"**

**5:**

**Escribir "Saliendo del sistema de consultas..."**

**FinSegun**

**Hasta Que consulta = 5**

**FinSi**

**2:**

**Escribir "Saliendo del sistema..."**

**De Otro Modo:**

**Escribir "Opción no válida"**

**FinSegun**

**Hasta Que opcion = 2**

**Finalgoritmo**