

# **Documentación del Sistema de Gestión de Gimnasio**

Este documento proporciona una descripción detallada del sistema de gestión para un gimnasio implementado en Python. El programa permite manejar clientes, aparatos, reservas y pagos, utilizando una base de datos MySQL para almacenar toda la información.

El sistema de gestión de gimnasio es una aplicación de consola que ofrece una interfaz de usuario basada en menús para interactuar con la base de datos. Permite realizar diversas operaciones como agregar clientes, registrar aparatos, hacer reservas y gestionar pagos.

## **Estructura de la Base de Datos**

El programa utiliza una base de datos MySQL con las siguientes tablas:

1. **\*\*Clientes\*\***: Almacena información de los clientes del gimnasio.
2. **\*\*Aparatos\*\***: Contiene detalles de los equipos disponibles en el gimnasio.
3. **\*\*Pagos\*\***: Registra los pagos realizados por los clientes.
4. **\*\*Reservas\*\***: Guarda las reservas de aparatos hechas por los clientes.

## **Funciones Principales**

### **1. Crear conexion()**

Esta función establece una conexión con la base de datos MySQL.

- Retorno: Un objeto de conexión si es exitosa, None si falla.
- Implementación: Utiliza la biblioteca mysql.connector para establecer la conexión.
- Manejo de errores: Captura y muestra cualquier error de conexión.

### **2. Agregar\_cliente()**

Añade un nuevo cliente a la base de datos y solicita el pago del primer mes.

#### **- Parámetros:**

- conexion: Objeto de conexión a la base de datos.

#### **- Funcionalidad:**

- Solicita al usuario ingresar la información del cliente (nombre, apellidos, DNI, edad, dirección, teléfono).
- Verifica si el DNI ya existe en la base de datos para evitar duplicados.
- Si es un cliente nuevo, inserta la información en la tabla de clientes.
- Solicita y registra el pago del primer mes.

#### **- Validaciones:**

- Comprueba la unicidad del DNI.
- Asegura que todos los campos requeridos estén completos.

### **3. Agregar aparato()**

Añade un nuevo aparato a la base de datos.

- Parámetros:

- conexion: Objeto de conexión a la base de datos.

- Funcionalidad:

- Solicita al usuario ingresar la información del aparato (tipo, marca, modelo).
- Inserta el nuevo aparato en la tabla correspondiente de la base de datos.

- Validaciones:

- Asegura que todos los campos necesarios estén completos.

### **4. Agregar pago()**

Registra un nuevo pago para un cliente existente.

- Parámetros:

- conexion: Objeto de conexión a la base de datos.
- dni: DNI del cliente (opcional).

- Funcionalidad:

- Si no se proporciona DNI, lo solicita al usuario.
- Verifica la existencia del cliente en la base de datos.
- Solicita el mes para el cual se realiza el pago.
- Registra el pago en la tabla de pagos.

- Validaciones:

- Comprueba que el cliente exista.
- Evita pagos duplicados para el mismo mes.

## **5. Agregar reserva()**

Crea una nueva reserva de un aparato para un cliente.

### **- Parámetros:**

- conexion: Objeto de conexión a la base de datos.

### **- Funcionalidad:**

- Solicita el DNI del cliente, el ID del aparato y la hora de inicio de la reserva.
- Calcula automáticamente la hora de fin (30 minutos después del inicio).
- Verifica la disponibilidad del aparato en el horario especificado.
- Si está disponible, registra la reserva en la base de datos.

### **- Validaciones:**

- Comprueba la existencia del cliente y del aparato.
- Verifica que no haya reservas superpuestas para el mismo aparato.

## **6. Ver pagos cliente()**

Muestra los pagos realizados por un cliente específico.

### **- Parámetros:**

- conexion: Objeto de conexión a la base de datos.

### **- Funcionalidad:**

- Solicita el DNI del cliente.
- Recupera y muestra todos los pagos registrados para ese cliente.

## **7. Ver clientes()**

Muestra una lista de todos los clientes registrados en el gimnasio.

### **- Parámetros:**

- conexion: Objeto de conexión a la base de datos.

### **- Funcionalidad:**

- Recupera la información de todos los clientes de la base de datos.
- Muestra la información de manera organizada.

## **8. Ver aparatos()**

Muestra una lista de todos los aparatos registrados en el gimnasio.

### **- Parámetros:**

- conexion: Objeto de conexión a la base de datos.

### **- Funcionalidad:**

- Recupera la información de todos los aparatos de la base de datos.
- Muestra la información de manera organizada.

### **- Presentación:**

- Presenta los datos en un formato tabular, incluyendo ID, tipo, marca y modelo.

## **9. Ver\_reservas\_aparato()**

Muestra las reservas para un aparato específico.

### **- Parámetros:**

- Conexión: Objeto de conexión a la base de datos.

### **- Funcionalidad:**

- Solicita el ID del aparato.
- Recupera y muestra todas las reservas para ese aparato.
- Incluye detalles del cliente y horarios de las reservas.

### **- Presentación:**

- Organiza la información cronológicamente.
- Muestra claramente los horarios de inicio y fin de cada reserva.

## 10. Main()

Función principal que ejecuta el programa y maneja la interacción con el usuario.

- Funcionalidad:

- Crea la conexión inicial a la base de datos.
- Presenta un menú de opciones al usuario.
- Maneja la selección de opciones y llama a las funciones correspondientes.
- Mantiene el programa en ejecución hasta que el usuario decide salir.

- Estructura del menú:

1. Agregar cliente
2. Agregar aparato
3. Agregar pago
4. Agregar reserva
5. Ver pagos de un cliente
6. Ver todos los clientes
7. Ver todos los aparatos
8. Ver reservas de un aparato
9. Salir

## Manejo de Errores y Excepciones

El programa implementa un manejo robusto de excepciones para tratar errores de conexión a la base de datos y operaciones fallidas.

Se proporcionan mensajes de error claros y específicos para ayudar en la resolución de problemas.

## Conclusión

Este sistema de gestión de gimnasio proporciona una solución completa para manejar las operaciones diarias de un gimnasio, incluyendo la gestión de clientes, equipos, reservas y pagos. Su diseño modular permite una fácil expansión y mantenimiento del código.