

## Enunciado del Ejercicio: Gestión de Libros de una Librería

**Objetivo:** Desarrollar una aplicación en Python que se conecte a una base de datos MySQL y permita gestionar la información de los libros de una librería. La aplicación debe ser capaz de realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre los registros de la tabla libros.

### Descripción del problema:

Tienes que desarrollar un sistema para una librería que permita:

1. **Agregar** nuevos libros a la base de datos, almacenando el título, el autor y el precio.
2. **Visualizar** la lista completa de libros almacenados en la base de datos.
3. **Actualizar** el precio de un libro existente.
4. **Eliminar** libros que ya no estén disponibles en la librería.

### Requisitos:

- La base de datos debe ser MySQL.
- Los libros deben estar almacenados en una tabla llamada libros, que contendrá las siguientes columnas:
  - id: un identificador único (PRIMARY KEY, AUTO\_INCREMENT).
  - titulo: el título del libro (VARCHAR).
  - autor: el nombre del autor del libro (VARCHAR).
  - precio: el precio del libro (DECIMAL).

### Pasos a seguir:

1. **Crear la base de datos:**
  - Crear una base de datos llamada libreria.
  - Crear una tabla llamada libros con los campos mencionados anteriormente.

```
CREATE DATABASE libreria;

USE libreria;

CREATE TABLE libros (
    id INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(255) NOT NULL,
    autor VARCHAR(255),
    precio DECIMAL(10, 2)
);
```

## 2. Desarrollar el programa en Python:

- El programa debe conectarse a la base de datos MySQL.
- Debe ofrecer un menú con las siguientes opciones:
  1. Agregar un nuevo libro.
  2. Ver todos los libros almacenados.
  3. Actualizar el precio de un libro.
  4. Eliminar un libro.
  5. Salir del programa.

## 3. Operaciones CRUD:

- Implementar funciones en Python para:
  - Insertar un nuevo libro en la tabla.
  - Consultar y mostrar todos los libros almacenados.
  - Actualizar el precio de un libro basado en su id.
  - Eliminar un libro de la tabla usando su id.

## Ejemplo del Menú de la Aplicación:

## Gestión de Librería

1. Agregar libro
2. Ver libros
3. Actualizar libro
4. Eliminar libro
5. Salir

Selecciona una opción:

## Enunciado del Ejercicio: Gestión de Empleados de una Empresa

**Objetivo:** Desarrollar una aplicación en **Python** que se conecte a una base de datos **MySQL** y permita gestionar la información de los empleados de una empresa. La aplicación debe ser capaz de realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre los registros de la tabla empleados.

### Descripción del problema:

La empresa necesita un sistema para gestionar su lista de empleados. Este sistema debe permitir:

1. **Registrar** nuevos empleados con su nombre, puesto de trabajo y salario.
2. **Visualizar** la lista completa de empleados almacenados en la base de datos.
3. **Actualizar** el salario o el puesto de un empleado existente.
4. **Eliminar** empleados que ya no trabajen en la empresa.

### Requisitos:

- La base de datos debe ser MySQL.
- Los empleados deben estar almacenados en una tabla llamada empleados, que contendrá las siguientes columnas:
  - id: un identificador único (PRIMARY KEY, AUTO\_INCREMENT).
  - nombre: el nombre del empleado (VARCHAR).
  - puesto: el cargo o puesto del empleado (VARCHAR).
  - salario: el salario del empleado (DECIMAL).

### Pasos a seguir:

1. **Crear la base de datos:**
  - Crear una base de datos llamada empresa.
  - Crear una tabla llamada empleados con los campos mencionados anteriormente.

```
CREATE DATABASE empresa;

USE empresa;

CREATE TABLE empleados (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    puesto VARCHAR(255),
    salario DECIMAL(10, 2)
);
```

## 2. Desarrollar el programa en Python:

- El programa debe conectarse a la base de datos MySQL.
- Debe ofrecer un menú con las siguientes opciones:
  1. Agregar un nuevo empleado.
  2. Ver todos los empleados.
  3. Actualizar el puesto o salario de un empleado.
  4. Eliminar un empleado.
  5. Salir del programa.

## 3. Operaciones CRUD:

- Implementar funciones en Python para:
  - Insertar un nuevo empleado en la tabla.
  - Consultar y mostrar todos los empleados almacenados.
  - Actualizar el puesto o el salario de un empleado basado en su id.
  - Eliminar un empleado de la tabla usando su id.

### Ejemplo del Menú de la Aplicación:

Gestión de Empleados

1. Agregar empleado
2. Ver empleados
3. Actualizar puesto o salario
4. Eliminar empleado
5. Salir

Selecciona una opción:

## Enunciado del Ejercicio: Gestión de Inventario de una Tienda

**Objetivo:** Desarrollar una aplicación en **Python** que se conecte a una base de datos **MySQL** y permita gestionar el inventario de una tienda. La aplicación debe ser capaz de realizar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) sobre los productos en la base de datos.

### Descripción del problema:

Se requiere un sistema para administrar el inventario de productos de una tienda. Este sistema debe permitir:

1. **Registrar** nuevos productos con su nombre, categoría y cantidad en inventario.
2. **Visualizar** la lista completa de productos almacenados.
3. **Actualizar** la cantidad o categoría de un producto existente.
4. **Eliminar** productos que ya no estén disponibles en la tienda.

### Requisitos:

- La base de datos debe ser MySQL.
- Los productos deben estar almacenados en una tabla llamada productos, que contendrá las siguientes columnas:
  - id: un identificador único (PRIMARY KEY, AUTO\_INCREMENT).
  - nombre: el nombre del producto (VARCHAR).
  - categoria: la categoría del producto (VARCHAR).
  - cantidad: la cantidad disponible en inventario (INT).

### Pasos a seguir:

1. **Crear la base de datos:**
  - Crear una base de datos llamada tienda.
  - Crear una tabla llamada productos con los campos mencionados anteriormente.

```
CREATE DATABASE tienda;

USE tienda;

CREATE TABLE productos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    categoria VARCHAR(255),
    cantidad INT
);
```

### 1. Desarrollar el programa en Python:

- El programa debe conectarse a la base de datos MySQL.
- Debe ofrecer un menú con las siguientes opciones:
  1. Agregar un nuevo producto.
  2. Ver todos los productos.
  3. Actualizar la cantidad o categoría de un producto.
  4. Eliminar un producto.
  5. Salir del programa.

### 2. Operaciones CRUD:

- Implementar funciones en Python para:
  - Insertar un nuevo producto en la tabla.
  - Consultar y mostrar todos los productos almacenados.
  - Actualizar la cantidad o categoría de un producto basado en su id.
  - Eliminar un producto de la tabla usando su id.



### Ejemplo del Menú de la Aplicación:

Gestión de Inventario de la Tienda

1. Agregar producto
2. Ver productos
3. Actualizar cantidad o categoría
4. Eliminar producto
5. Salir

Selecciona una opción: