Ejercicio: Sistema de Gestión de Contactos

Descripción

Desarrolla un sistema de gestión de contactos utilizando Python y MySQL. El sistema debe permitir a los usuarios realizar las siguientes acciones a través de un menú interactivo:

1. **Agregar un contacto**: El usuario debe poder ingresar el nombre, teléfono y correo electrónico del contacto, y almacenar esta información en una base de datos MySQL.
2. **Listar todos los contactos**: El sistema debe mostrar una lista de todos los contactos almacenados en la base de datos.
3. **Eliminar un contacto**: El usuario debe poder eliminar un contacto de la base de datos proporcionando el ID del contacto.
4. **Salir del programa**: El usuario puede elegir salir del menú y terminar el programa.

Requisitos

1. **Base de Datos MySQL**:
    - Crea una base de datos llamada ContactosDB.
    - Dentro de esta base de datos, crea una tabla llamada contactos con los siguientes campos:
        - id (INT, AUTO_INCREMENT, PRIMARY KEY)
        - nombre (VARCHAR(100))
        - telefono (VARCHAR(20))
        - correo (VARCHAR(100))

```sql
CREATE DATABASE ContactosDB;

USE ContactosDB;

CREATE TABLE contactos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    telefono VARCHAR(20),
    correo VARCHAR(100)
);
```

2. **Código en Python**:
    - Utiliza el módulo mysql.connector para conectar Python con MySQL.
    - Implementa una clase Database para manejar la conexión a la base de datos y ejecutar consultas SQL.
    - Implementa una clase Contacto con métodos para agregar, listar y eliminar contactos.
    - Desarrolla una función menu que muestre un menú interactivo en la consola, permitiendo al usuario seleccionar opciones para agregar, listar, eliminar contactos o salir del programa.

Ejemplo de Uso

Cuando el usuario ejecute el programa, verá un menú como el siguiente:

```
--- Menú de Gestión de Contactos ---
1. Agregar contacto
2. Listar contactos
3. Eliminar contacto
4. Salir
```

- Al seleccionar la opción 1, el programa solicitará el nombre, teléfono y correo del contacto, y lo añadirá a la base de datos.
- Al seleccionar la opción 2, el programa mostrará todos los contactos almacenados.
- Al seleccionar la opción 3, el programa pedirá el ID del contacto a eliminar y lo eliminará de la base de datos.
- Al seleccionar la opción 4, el programa finalizará.

Ejercicio: Gestión de Productos en una Tienda

Descripción

Desarrolla un sistema de gestión de productos para una tienda utilizando Python y MySQL. El sistema debe permitir a los usuarios realizar las siguientes acciones a través de un menú interactivo:

1. **Agregar un producto**: El usuario debe ingresar el nombre del producto, su precio y la cantidad disponible, y almacenar esta información en una base de datos MySQL.
2. **Listar todos los productos**: El sistema debe mostrar una lista de todos los productos almacenados en la base de datos.
3. **Actualizar la cantidad de un producto**: El usuario debe poder actualizar la cantidad disponible de un producto proporcionando el ID del producto y la nueva cantidad.
4. **Eliminar un producto**: El usuario debe poder eliminar un producto de la base de datos proporcionando el ID del producto.
5. **Salir del programa**: El usuario puede elegir salir del menú y terminar el programa.

Requisitos

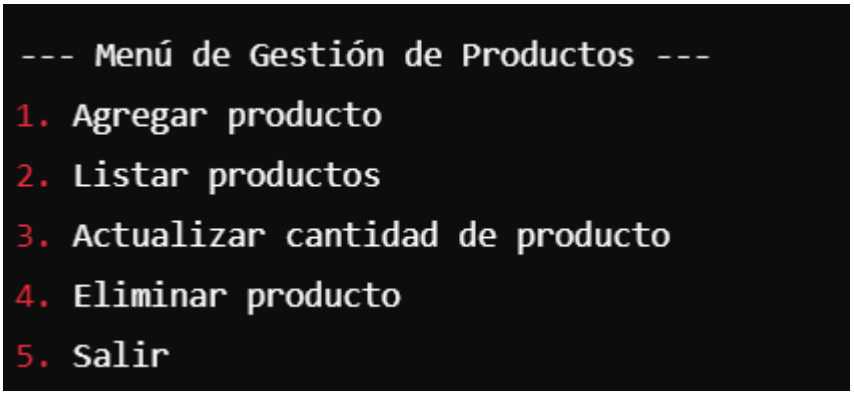
1. **Base de Datos MySQL**:
    - Crea una base de datos llamada TiendaDB.
    - Dentro de esta base de datos, crea una tabla llamada productos con los siguientes campos:
        - id (INT, AUTO_INCREMENT, PRIMARY KEY)
        - nombre (VARCHAR(100))
        - precio (DECIMAL(10, 2))
        - cantidad (INT)

```sql
CREATE DATABASE TiendaDB;

USE TiendaDB;

CREATE TABLE productos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    precio DECIMAL(10, 2),
    cantidad INT
);
```

2. **Código en Python**:
    - Utiliza el módulo mysql.connector para conectar Python con MySQL.
    - Implementa una clase Database para manejar la conexión a la base de datos y ejecutar consultas SQL.
    - Implementa una clase Producto con métodos para agregar, listar, actualizar y eliminar productos.
    - Desarrolla una función menu que muestre un menú interactivo en la consola, permitiendo al usuario seleccionar opciones para agregar, listar, actualizar, eliminar productos o salir del programa.

Ejemplo de Uso

Cuando el usuario ejecute el programa, verá un menú como el siguiente:



Ejercicio: Sistema de Gestión de Empleados

Descripción

Desarrolla un sistema de gestión de empleados para una empresa utilizando Python y MySQL. El sistema debe permitir a los usuarios realizar las siguientes acciones a través de un menú interactivo:

1. **Agregar un empleado**: El usuario debe ingresar el nombre del empleado, su puesto y el salario, y almacenar esta información en una base de datos MySQL.
2. **Listar todos los empleados**: El sistema debe mostrar una lista de todos los empleados almacenados en la base de datos.
3. **Actualizar el salario de un empleado**: El usuario debe poder actualizar el salario de un empleado proporcionando el ID del empleado y el nuevo salario.
4. **Eliminar un empleado**: El usuario debe poder eliminar un empleado de la base de datos proporcionando el ID del empleado.
5. **Salir del programa**: El usuario puede elegir salir del menú y terminar el programa.

Requisitos

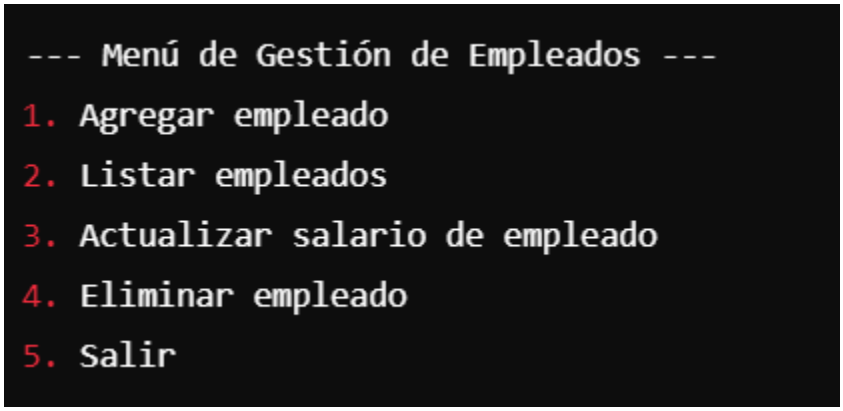
1. **Base de Datos MySQL**:
    - Crea una base de datos llamada EmpresaDB.
    - Dentro de esta base de datos, crea una tabla llamada empleados con los siguientes campos:
        - id (INT, AUTO_INCREMENT, PRIMARY KEY)
        - nombre (VARCHAR(100))
        - puesto (VARCHAR(100))
        - salario (DECIMAL(10, 2))

```sql
CREATE DATABASE EmpresaDB;

USE EmpresaDB;

CREATE TABLE empleados (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    puesto VARCHAR(100),
    salario DECIMAL(10, 2)
);
```

2. **Código en Python**:
    - Utiliza el módulo mysql.connector para conectar Python con MySQL.
    - Implementa una clase Database para manejar la conexión a la base de datos y ejecutar consultas SQL.
    - Implementa una clase Empleado con métodos para agregar, listar, actualizar y eliminar empleados.
    - Desarrolla una función menu que muestre un menú interactivo en la consola, permitiendo al usuario seleccionar opciones para agregar, listar, actualizar, eliminar empleados o salir del programa.

Ejemplo de Uso

Cuando el usuario ejecute el programa, verá un menú como el siguiente:



Ejercicio: Sistema de Gestión de Reservas en un Parque Natural

Descripción

Desarrolla un sistema que gestione las reservas de visitas guiadas en un parque natural. El sistema debe permitir a los usuarios realizar las siguientes acciones a través de un menú interactivo:

1. **Registrar una reserva**: El usuario debe ingresar su nombre, la fecha de la visita, el número de personas y el tipo de tour (por ejemplo, "Aventura", "Familiar", "Ecológico"). Esta información debe almacenarse en una base de datos MySQL.
2. **Listar todas las reservas**: El sistema debe mostrar una lista de todas las reservas registradas.
3. **Actualizar una reserva**: El usuario debe poder actualizar una reserva proporcionando el ID de la reserva y los nuevos detalles.
4. **Eliminar una reserva**: El usuario debe poder eliminar una reserva proporcionando el ID.
5. **Salir del programa**: El usuario puede finalizar el programa.

Requisitos

1. **Base de Datos MySQL**:
    - Crea una base de datos llamada ParqueDB.
    - Dentro de esta base de datos, crea una tabla llamada reservas con los siguientes campos:
        - id (INT, AUTO_INCREMENT, PRIMARY KEY)
        - nombre (VARCHAR(100))
        - fecha (DATE)
        - personas (INT)
        - tipo_tour (VARCHAR(50))

```sql
CREATE DATABASE ParqueDB;

USE ParqueDB;

CREATE TABLE reservas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100),
    fecha DATE,
    personas INT,
    tipo_tour VARCHAR(50)
);
```

2. **Código en Python**:
    - Utiliza mysql.connector para conectar con la base de datos.
    - Implementa una clase Reserva que permita gestionar las reservas, con métodos para agregar, listar, actualizar y eliminar.
    - Implementa un menú interactivo que permita al usuario gestionar las reservas.

Ejemplo de Uso

Al ejecutar el programa, el menú interactivo debe verse así:

```
--- Menú de Gestión de Reservas ---
1. Registrar nueva reserva
2. Listar todas las reservas
3. Actualizar una reserva
4. Eliminar una reserva
5. Salir
```