**EXPLICACIÓN GENERAL DEL CÓDIGO**

El código proporcionado muestra un programa de consola para gestionar un concesionario de coches utilizando una base de datos PostgreSQL. A continuación, se explica su funcionamiento y lógica paso a paso:

1. Se importan los módulos necesarios: `psycopg2` para la conexión con la base de datos PostgreSQL y `os` para realizar operaciones relacionadas con el sistema operativo.

2. Se define una función `borrar\_consola()` que utiliza el comando del sistema operativo `cls` para borrar la consola. Esto se utiliza para limpiar la pantalla y mejorar la presentación del programa.

3. Se establece la conexión a la base de datos PostgreSQL utilizando los parámetros de conexión proporcionados (host, base de datos, usuario y contraseña).

4. Se define la función `crear\_tablas()` que se encarga de crear las tablas necesarias en la base de datos si no existen. Esta función utiliza el objeto `cursor` para ejecutar sentencias SQL que crean las tablas `coches`, `clientes`, `fichas`, `mecanicos` y `reparaciones`. Cada tabla tiene su propia estructura y restricciones de clave primaria y referencias a otras tablas. Después de ejecutar las sentencias, se confirma la transacción y se cierra el cursor.

5. A continuación, se definen varias funciones que realizan operaciones relacionadas con la base de datos:

- `insertar\_coche()`: inserta un nuevo coche en la tabla `coches`.

- `insertar\_cliente()`: inserta un nuevo cliente en la tabla `clientes`.

- `crear\_ficha\_compra()`: crea una ficha de compra en la tabla `fichas`.

- `insertar\_mecanico()`: inserta un nuevo mecánico en la tabla `mecanicos`.

- `registrar\_reparacion()`: registra una reparación en la tabla `reparaciones`.

- `mostrar\_datos()`: muestra los datos de todas las tablas en la base de datos.

- `borrar\_coche()`: borra un coche de la tabla `coches` según la matrícula proporcionada.

- `borrar\_cliente()`: borra un cliente de la tabla `clientes` según el DNI proporcionado.

- `borrar\_mecanico()`: borra un mecánico de la tabla `mecanicos` según el DNI proporcionado.

- `borrar\_todo()`: borra todas las tablas de la base de datos.

- `guardar\_cambios()`: guarda los cambios realizados en la base de datos.

6. Por último, se define la función `menu\_principal()` que muestra un menú de opciones para que el usuario interactúe con el programa. El usuario puede seleccionar una opción ingresando el número correspondiente. Dependiendo de la opción seleccionada, se llamará a una función específica para realizar la operación deseada. El ciclo `while True` permite que el menú se muestre continuamente hasta que el usuario seleccione la opción "0" para salir.

7. Antes de ejecutar el menú principal, se llama a la función `crear\_tablas()` para asegurarse de que las tablas estén creadas en la base de datos.

8. Finalmente, se ejecuta la función `menu\_principal()` para iniciar el programa.

El código proporcionado permite al usuario interactuar con la base de datos del concesionario de coches mediante el menú principal.

**EXPLICACIÓN DE LAS SENTENCIAS SQL**

Claro, aquí está una explicación de las sentencias SQL utilizadas en el programa:

1. CREATE TABLE: Esta sentencia se utiliza para crear una tabla en la base de datos. Se definen las columnas de la tabla junto con sus tipos de datos y restricciones. Por ejemplo:

CREATE TABLE coches (

matricula VARCHAR(10) PRIMARY KEY,

marca VARCHAR(50),

modelo VARCHAR(50),

anio INTEGER,

precio FLOAT

);

En este caso, se crea la tabla "coches" con las columnas "matricula" (clave primaria), "marca", "modelo", "anio" y "precio".

1. INSERT INTO: Esta sentencia se utiliza para insertar datos en una tabla. Se especifican los valores que se insertarán en las columnas correspondientes. Por ejemplo:

INSERT INTO coches (matricula, marca, modelo, anio, precio)

VALUES ('ABC123', 'Ford', 'Mustang', 2022, 50000.0);

En este caso, se inserta un nuevo coche en la tabla "coches" con los valores proporcionados.

1. SELECT: Esta sentencia se utiliza para recuperar datos de una o varias tablas. Puede especificar las columnas que desea seleccionar y también puede utilizar cláusulas WHERE para filtrar los resultados. Por ejemplo:

SELECT \* FROM coches;

Esta sentencia selecciona todas las filas y columnas de la tabla "coches".

1. DELETE FROM: Esta sentencia se utiliza para eliminar filas de una tabla. Puede utilizar una cláusula WHERE para especificar las filas que se eliminarán. Por ejemplo:

DELETE FROM coches WHERE matricula = 'ABC123';

En este caso, se elimina la fila de la tabla "coches" que tiene la matrícula "ABC123".

Estas son algunas de las sentencias SQL utilizadas en el programa para crear tablas, insertar datos, seleccionar datos, eliminar filas y modificar la estructura de las tablas. El programa utiliza estas sentencias junto con el módulo **psycopg2** para interactuar con la base de datos PostgreSQL y realizar operaciones de gestión de datos en el concesionario de coches.

.