El código proporcionado es un programa de administración de una clínica llamada "Clínica San Patrás". A continuación, explicaré el funcionamiento y la estructura del código, así como el uso de las sentencias SQL y las funciones implementadas.

1. Importación de bibliotecas:

- `psycopg2`: Es una biblioteca de Python que proporciona soporte para la conexión y manipulación de bases de datos PostgreSQL.

- `os`: Proporciona funciones para interactuar con el sistema operativo, en este caso, se utiliza para limpiar la consola.

2. Función `establecer\_conexion()`:

- Esta función se encarga de establecer la conexión con la base de datos PostgreSQL.

- Utiliza el módulo `psycopg2` para establecer la conexión y devuelve el objeto de conexión si la conexión es exitosa.

- Si ocurre algún error al conectar, muestra el mensaje de error correspondiente.

3. Función `crear\_tablas(cursor)`:

- Esta función se encarga de crear las tablas necesarias en la base de datos.

- Utiliza sentencias SQL para crear tres tablas: `Pacientes`, `Medicos` e `Ingresos`.

- Cada tabla tiene una estructura definida con columnas y tipos de datos.

- Si ocurre algún error al crear las tablas, muestra el mensaje de error correspondiente.

4. Función `borrar\_consola()`: (En su lugar se usa directamente el comando os.system(“cls”)

- Utiliza el módulo `os` para ejecutar el comando `cls` en la consola, lo cual permite limpiar la pantalla.

5. Funciones de inserción:

- `insertar\_paciente(cursor)`: Permite al usuario ingresar información de un paciente y la inserta en la tabla `Pacientes`.

- `insertar\_medico(cursor)`: Permite al usuario ingresar información de un médico y lo inserta en la tabla `Medicos`.

- `insertar\_ingreso(cursor)`: Permite al usuario ingresar información de un ingreso y lo inserta en la tabla `Ingresos`.

- Todas estas funciones utilizan sentencias SQL `INSERT INTO` para insertar los datos proporcionados por el usuario en las respectivas tablas.

6. Funciones de consulta:

- `mostrar\_pacientes(cursor)`: Recupera todos los registros de la tabla `Pacientes` y los muestra en la consola.

- `mostrar\_medicos(cursor)`: Recupera todos los registros de la tabla `Medicos` y los muestra en la consola.

- `mostrar\_ingresos(cursor)`: Recupera todos los registros de la tabla `Ingresos` y los muestra en la consola.

- Estas funciones utilizan sentencias SQL `SELECT` para recuperar los datos de las tablas y luego los muestran en la consola.

7. Funciones de búsqueda (continuación):

- `buscar\_paciente(cursor)`: Permite al usuario buscar un paciente por su código y muestra su información si se encuentra en la tabla `Pacientes`.

- `buscar\_medico(cursor)`: Permite al usuario buscar un médico por su código y muestra su información si se encuentra en la tabla `Medicos`.

- `buscar\_ingreso(cursor)`: Permite al usuario buscar un ingreso por su código y muestra su información si se encuentra en la tabla `Ingresos`.

Estas funciones utilizan sentencias SQL `SELECT` con cláusulas `WHERE` para buscar registros específicos en las tablas y luego muestran la información encontrada en la consola.

8. Funciones principales:

- `menu\_principal()`: Muestra el menú principal del programa, donde el usuario puede elegir entre diferentes opciones, como agregar un paciente, agregar un médico, realizar una búsqueda, mostrar todos los registros, etc.

- `main()`: Función principal del programa que se encarga de establecer la conexión con la base de datos, crear las tablas (si no existen) y mostrar el menú principal.

Estas funciones permiten al usuario interactuar con el programa y seleccionar las diferentes acciones que desee realizar.

La función `crear\_tablas(cursor)` se encarga de crear las tablas necesarias en la base de datos. A continuación, explicaré las relaciones entre las tablas, los tipos de datos utilizados y la estructura de cada una de ellas:

1. Tabla "Pacientes":

- Columnas:

- `codigo` (SERIAL PRIMARY KEY): Es un identificador único para cada paciente, generado automáticamente.

- `nombre` (VARCHAR(100)): Almacena el nombre del paciente.

- `apellidos` (VARCHAR(100)): Almacena los apellidos del paciente.

- `direccion` (VARCHAR(100)): Almacena la dirección del paciente.

- `poblacion` (VARCHAR(100)): Almacena la población del paciente.

- `provincia` (VARCHAR(100)): Almacena la provincia del paciente.

- `codigo\_postal` (VARCHAR(10)): Almacena el código postal del paciente.

- `telefono` (VARCHAR(20)): Almacena el número de teléfono del paciente.

- `fecha\_nacimiento` (DATE): Almacena la fecha de nacimiento del paciente.

2. Tabla "Medicos":

- Columnas:

- `codigo` (SERIAL PRIMARY KEY): Es un identificador único para cada médico, generado automáticamente.

- `nombre` (VARCHAR(100)): Almacena el nombre del médico.

- `apellidos` (VARCHAR(100)): Almacena los apellidos del médico.

- `telefono` (VARCHAR(20)): Almacena el número de teléfono del médico.

- `especialidad` (VARCHAR(100)): Almacena la especialidad del médico.

3. Tabla "Ingresos":

- Columnas:

- `codigo\_ingreso` (SERIAL PRIMARY KEY): Es un identificador único para cada ingreso, generado automáticamente.

- `codigo\_paciente` (INTEGER REFERENCES Pacientes(codigo)): Hace referencia al código del paciente relacionado con el ingreso en la tabla "Pacientes".

- `codigo\_medico` (INTEGER REFERENCES Medicos(codigo)): Hace referencia al código del médico relacionado con el ingreso en la tabla "Medicos".

- `numero\_habitacion` (INTEGER): Almacena el número de habitación del ingreso.

- `numero\_cama` (INTEGER): Almacena el número de cama del ingreso.

- `fecha\_ingreso` (DATE): Almacena la fecha de ingreso del paciente.

Las relaciones entre las tablas se establecen mediante las cláusulas `REFERENCES` en las columnas `codigo\_paciente` y `codigo\_medico` de la tabla "Ingresos", que hacen referencia a las columnas `codigo` de las tablas "Pacientes" y "Medicos" respectivamente.

En resumen, la tabla "Pacientes" almacena la información de los pacientes, la tabla "Medicos" almacena la información de los médicos y la tabla "Ingresos" almacena la información de los ingresos de los pacientes, relacionando los códigos de pacientes y médicos correspondientes.