



## Guía: Creación de la clase de Conexión a la Base de Datos en Python


### Introducción

En esta guía aprenderemos a crear una clase `Conexion` en Python, encargada de gestionar la conexión a una base de datos PostgreSQL. Definiremos atributos y métodos de clase para obtener una conexión y un cursor, encapsulando la lógica de conexión de manera profesional y reutilizable.

---

### Paso 1: Crear el archivo de la clase de conexión

 Primero, creamos un nuevo archivo Python llamado `conexion.py` dentro de la carpeta de trabajo:

 **Ruta del archivo:** `capa_datos_persona/conexion.py`

En este archivo escribiremos toda la lógica de la clase `Conexion`.

---

### Paso 2: Definir la clase `Conexion` y sus atributos

## Descripción

Definimos la clase `Conexion` con los atributos necesarios para conectarnos a la base de datos, incluyendo `database`, `username`, `password`, `port`, y `host`, así como las referencias `conexion` y `cursor`.

### Código trabajado (`capa_datos_persona/conexion.py`):

```
from logger_base import log
import psycopg

class Conexion:
    _DATABASE = 'test_db'
    _USERNAME = 'postgres'
    _PASSWORD = 'admin'
    _DB_PORT = '5432'
    _HOST = '127.0.0.1'
```

### Explicación:

- Usamos atributos privados de clase (con guión bajo).
- Importamos `psycopg` para gestionar la conexión y `log` para registrar error.

---

## Paso 3: Definir el método `obtenerConexion`

### Descripción

Creamos el método de clase `obtenerConexion` que devuelve la conexión a la base de datos, creándola solo si no existe.

### Código trabajado (`capa_datos_persona/conexion.py`):

```
@classmethod
def obtener_conexion(cls):
    return psycopg.connect(
        host=cls._HOST,
        user=cls._USERNAME,
        password=cls._PASSWORD,
        port=cls._DB_PORT,
        dbname=cls._DATABASE
    )
```

### Explicación:

- Creamos y retornamos un objeto de tipo conexión.



## Sección final: Código completo de los archivos trabajados



Código completo del archivo:



Ruta y nombre del archivo: `capa_datos_persona/conexion.py`

```
from logger_base import log

import psycopg

class Conexion:
    _DATABASE = 'test_db'
    _USERNAME = 'postgres'
    _PASSWORD = 'admin'
    _DB_PORT = '5432'
    _HOST = '127.0.0.1'

    @classmethod
    def obtener_conexion(cls):
        return psycopg.connect(
            host=cls._HOST,
            user=cls._USERNAME,
            password=cls._PASSWORD,
            port=cls._DB_PORT,
            dbname=cls._DATABASE
        )
```



## Conclusión



En esta guía hemos creado la clase `Conexion`, encapsulando la lógica de conexión en una base de datos PostgreSQL. Esta implementación permite reutilizar y centralizar la gestión de la conexión, evitando conexiones duplicadas y controlando errores de manera profesional.



¡Con esta clase, nuestra aplicación estará lista para interactuar con la base de datos de forma segura y eficiente!



Sigue adelante con tu aprendizaje 🚀, ¡el esfuerzo vale la pena!

¡Saludos! 🙌

**Ing. Marcela Gamiño e Ing. Ubaldo Acosta**

**Fundadores de [GlobalMentoring.com.mx](https://www.globalmentoring.com.mx)**