

1 Verificación funcional del módulo

- Lo primero es verificar que la librería *psycopg2*, se encuentre instalada, basta con escribir en el símbolo del sistema *psycopg2*. En caso de que no este instalada escribir *pip install psycopg2*.
- Asegurarse que la ruta de instalación este en el **PATH**, por medio de la edición de variables de entorno.
- Al momento de realizar la conexión asegurarse que el puerto de la conexión sea correcto

```
1 connection = psycopg2.connect(  
2     host = 'localhost',  
3     user = 'postgres',  
4     password = '123456789',  
5     database = 'Video_Games_Sales',  
6     port = '5433'  
7 )
```

Listing 1: Conexión a la base de datos

- Usar estructuras que permiten el manejo de excepción de errores, como el bloque *try* en python

```
1 try:  
2     # Código que se intentara ejecutar  
3     # Puede contener operaciones que podrían generar una  
4     # excepción  
5 except ExceptionTipo1 as nombre_variable1:  
6     # Manejo de la excepción del tipo ExceptionTipo1  
7 except ExceptionTipo2 as nombre_variable2:  
8     # Manejo de la excepción del tipo ExceptionTipo2  
9 else:  
10    # Código a ejecutar si no se levanta ninguna excepción  
11    # en el bloque try  
12 finally:  
13    # Código que se ejecutara siempre, independientemente  
14    # de si se levanto una excepción o no en el bloque try
```

Listing 2: Estructura de manejo de errores

- Procurar cerrar la conexión con la base de datos.

```
1
2 # Inicializacion de la conexion con la base de datos
3
4 import psycopg2
5 try:
6     connection = psycopg2.connect(
7         host = 'Localhost',
8         user = 'postgres',
9         password = '123456789',
10        database = 'Video_Games_Sales',
11        port = '5433'
12    )
13
14
15 #-----#
16
17 # Consulta a las tablas
18
19 .
20 .
21 .
22 .
23
24 except Exception as ex:
25     print(ex)
26 finally:
27     connection.close()
28     print("Conexion finalizada")
```

Listing 3: Verificación final