



MINISTERIO DEL TRABAJO

# Configurar Proyecto Python + MySQL

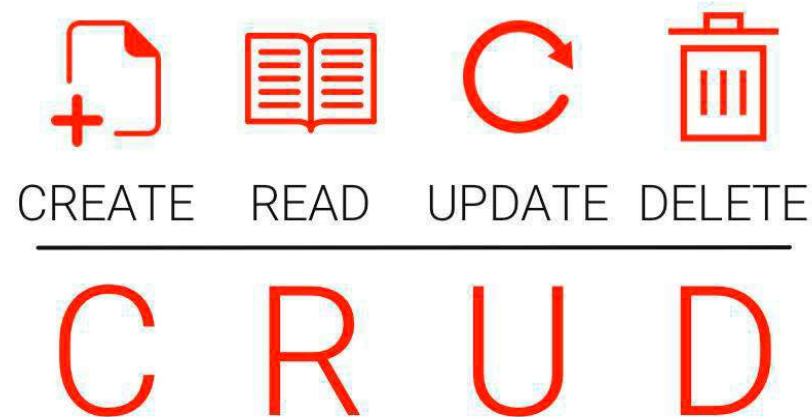


@SENAcomunica

[www.sena.edu.co](http://www.sena.edu.co)

# Que es CRUD

**CRUD** (Create, Read, Update, Delete) es un acrónimo para la manera en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente. CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos, pero también puede aplicar a funciones de un nivel superior de una aplicación como **soft deletes** donde la información no es realmente eliminada, sino marcada como eliminada a través de un estatus.





# Construcción de una Aplicación CRUD Básica:

Para la creación de nuestra aplicación **CRUD** básica con Python usaremos Mysql como nuestra base de datos y crearemos nuestras propias funciones para gestionar la información.

```
principal.py  conexion.py X
BD > conexion.py > ...
1 #Creating connection with MySQL workbench
2 import mysql.connector # 1. Lo primero que hacemos es importar el módulo que nos permite conectarnos con MySQL
3 def conectarBD(): # 2. Creamos la función conectarBD
4     """ 3. Del módulo importado llamamos a la función connect pasando la ubicación de nuestro servidor
5         que es 'localhost', el usuario que por defecto al instalar MySQL se creó el usuario 'root' y la contraseña
6         de ese usuario que tiene por defecto un string vacío; y la base de datos en mi caso 'farmacia' """
7     cnx=mysql.connector.connect(host='localhost',
8                                user='root',
9                                passwd='',
10                               database='farmacia')
11     return cnx #4. Retornamos la conexión a la Base de Datos
```

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  conexion.py - app_consola_crud - Visual Studio Code
principal.py  conexion.py X
BD > conexion.py > listarCategorias
22 def listarCategorias():
23     l=conectarBD()
24     c=l.cursor()
25     c.execute("select * from categoria")
26     for tabla in c: #6. Mediante un for podemos ver todas las categorías existentes de la base de datos 'farmacia'
27         print(tabla) #7. Imprime las tablas de farmacia
28     l.close() #8. Cerramos la conexión con el servidor de MySQL
29     return l #9. Retornamos la lista de categorías
30
31 #conectarBD() #10. Invocamos la función para mostrar el resultado final
32 listarCategorias() #10. Invocamos la función para mostrar el resultado final
33

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  JUPYTER
(basura) C:\Users\Acer\Desktop\app_consola_crud>C:/Users/Acer/anaconda3/envs/basura/python.exe c:/Users/Acer/Desktop/a
pp_consola_crud/BD/conexion.py
(1, 'Medicamentos Naturales')
(2, 'Jarabes')
(3, 'Pastillas')

(basura) C:\Users\Acer\Desktop\app_consola_crud>
```



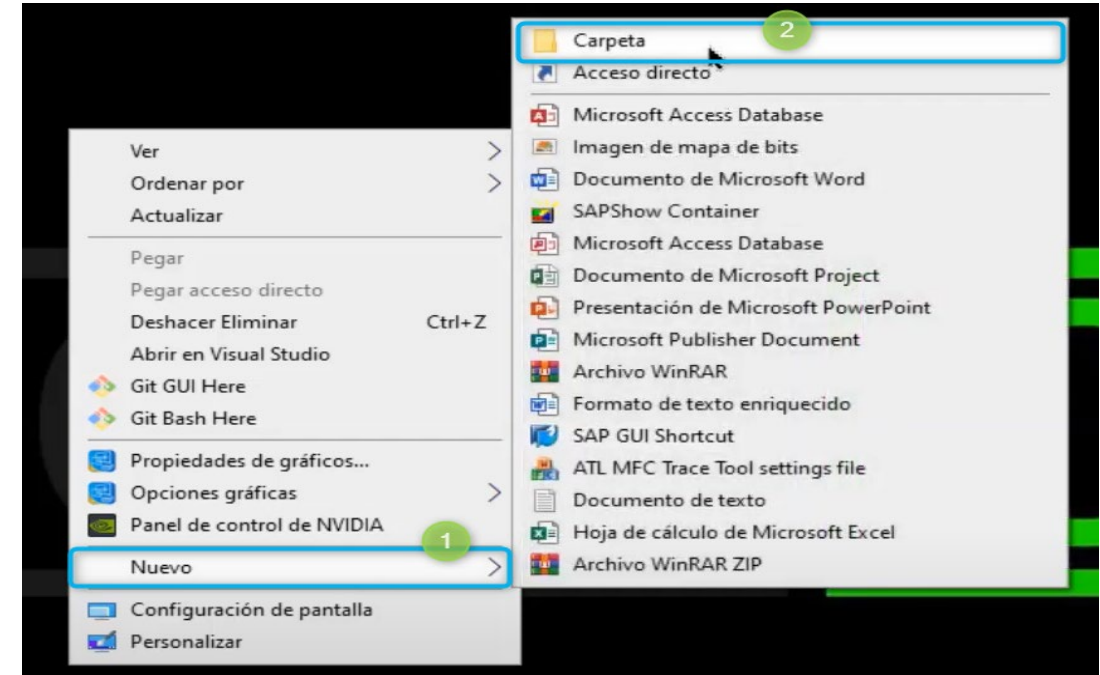
# Configurar el Directorio de Trabajo



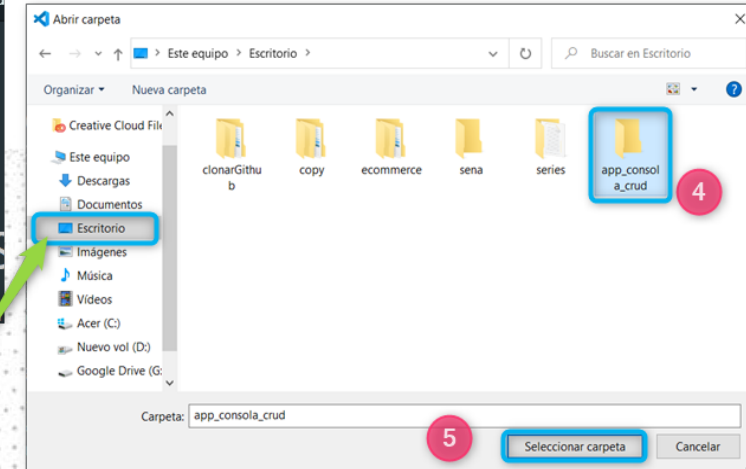
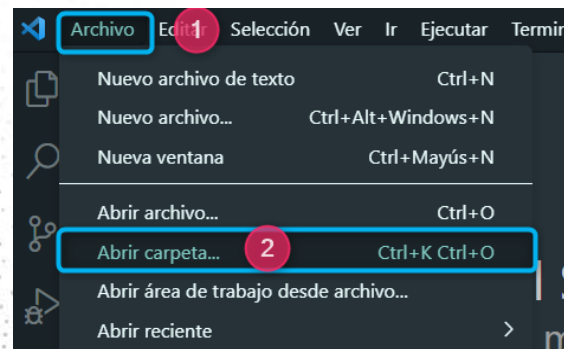
# Crear una Carpeta en el Escritorio

## Clic derecho nuevo + carpeta

- Nombre de la carpeta  
**app\_consola\_crud**

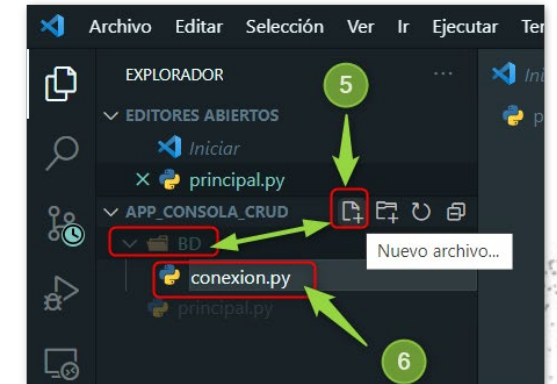
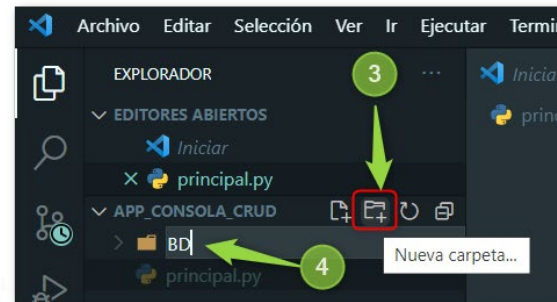
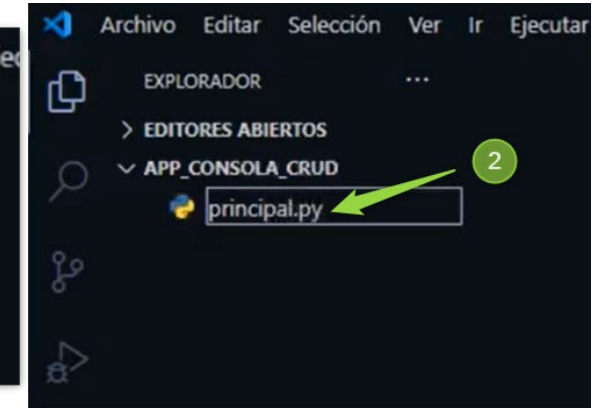
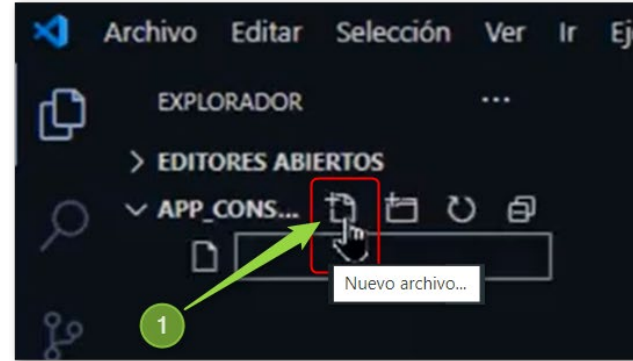


- Arrastrar la Carpeta a Visual Studio Code o  
Abrir la carpeta en el Editor de Visual



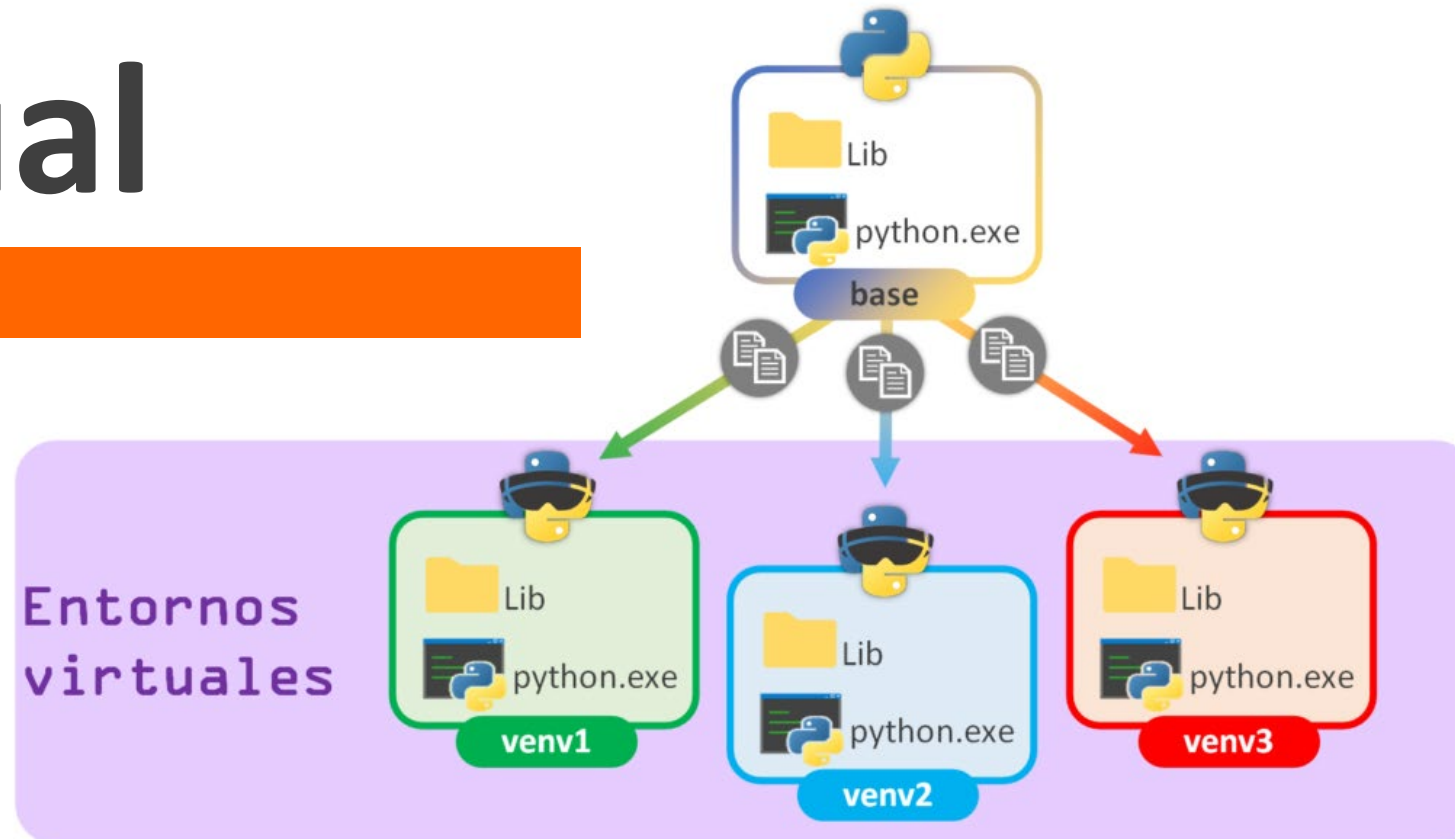
# Crear Archivos de Python de Visual

1. En el explorador de Visual Dar Clic en el botón Nuevo archivo
2. escribir: **principal.py**
3. En el explorador Dar clic en el botón Crear Carpeta
4. escribir: **BD**
5. Selecciona la Capeta Creada y Clic en Nuevo Archivo
6. Escribir : **conexión.py**





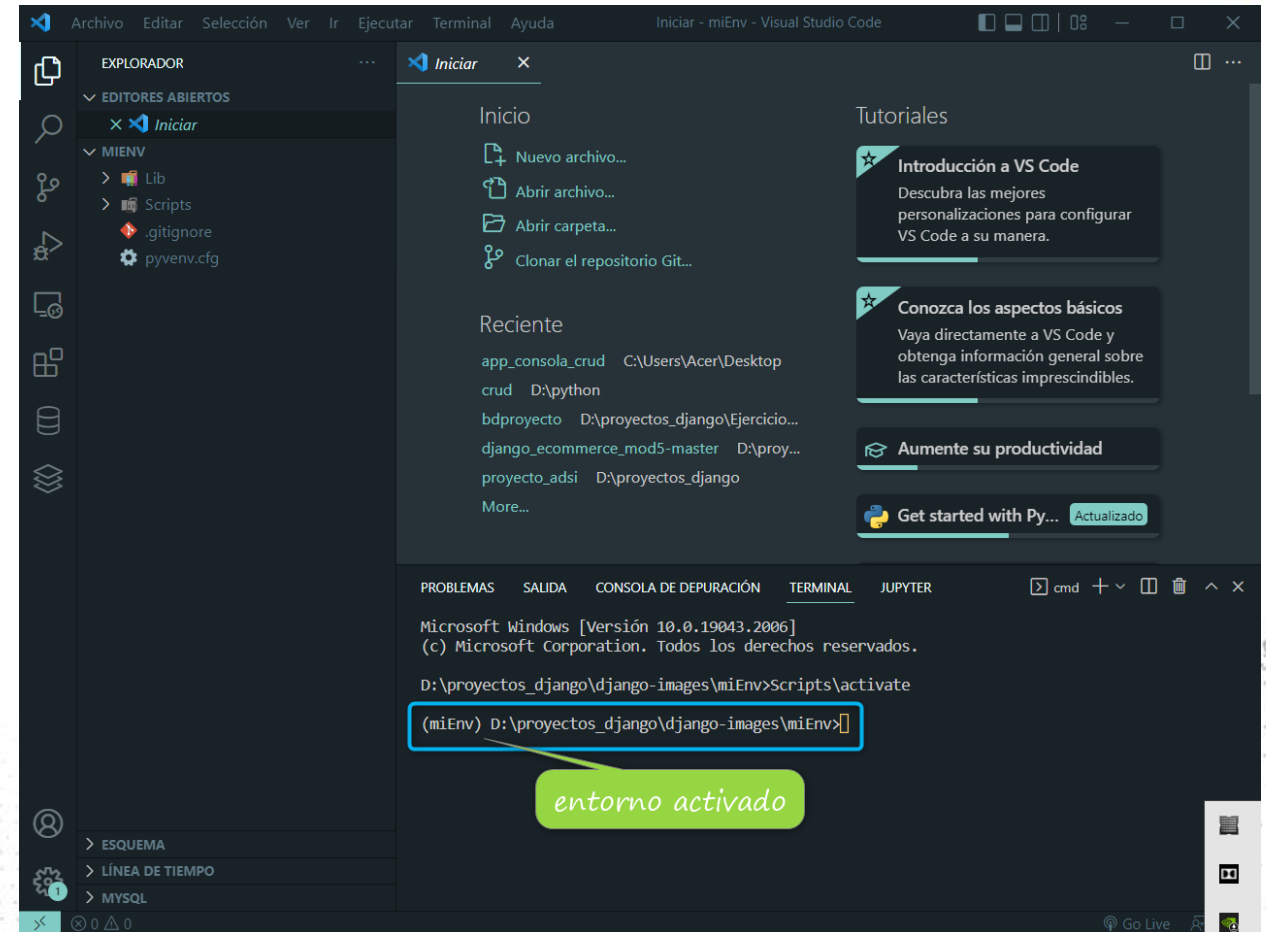
# Configurar el entorno virtual



# Configurar entorno virtual

**Ctrl + ñ:** abrir la terminal

- Crear entorno virtual por primera vez  
**python -m virtualenv miEnv**
- activar el entorno virtual:  
**mi\_env\Scripts\activate**



Nota: en caso de no tener instalado la herramienta de entornos virtuales, escriben en la terminal:

**pip install virtualenv**





# Paquete de Python necesario para conectarnos a MySQL.

Desde la línea de comandos ejecutamos el programa pip con el siguiente paquete a instalar:

-> `pip install mysql-connector-python`

Luego de ejecutar el programa pip podemos ver que nos informa de la instalación del paquete 'mysql-connector-python':

```
C:\Users\Acer\Desktop\app_consola_crud>pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-8.0.30-cp39-cp39-win_amd64.whl (7.8 MB)
    |████████████████████| 7.8 MB 598 kB/s
Requirement already satisfied: protobuf<=3.20.1,>=3.11.0 in c:\users\acer\anaconda3\lib\site-p
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-8.0.30
C:\Users\Acer\Desktop\app_consola_crud>
```



# Conexión al Servidor de Mysql



# Archivo conexion.py



```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  conexion.py - app_consola_crud - Visual Studio Code

principal.py  conexion.py X
BD > python conexion.py > ...
1  #Creating connection with MySQL workbench
2  import mysql.connector # 1. Lo primero que hacemos es importar el módulo que nos permite conectarnos con MySQL:
3  def conectarBD(): # 2. Creamos la función conectarBD
4      ''' 3. Del módulo importado llamamos a la función connect pasando la ubicación de nuestro servidor
5          que es 'localhost', el usuario que por defecto al instalar MySQL se creó el usuario 'root' y la clave
6          de ese usuario que tiene por defecto un string vacío: y la base de datos en mi caso 'farmacia' '''
7      cnx=mysql.connector.connect(host='localhost',
8                                  user='root',
9                                  passwd='',
10                                 database='farmacia')
11      c=cnx.cursor()# 4. Luego a partir del objeto 'cnx' que es de la clase 'MySQLConnection' llamamos al método 'cursor'
12      c.execute("show tables")# 5. A partir del objeto 'cursor=c' llamamos al método execute y le pasamos como parámetro
13                          # un comando SQL, en este caso 'show tables':
14      for tabla in c: #6. Mediante un for podemos ver todas las tablas existentes de la base de datos 'farmacia':
15          print(tabla)#7. Imprime las tablas de farmacia
16      cnx.close() #8. Cerramos la conexión con el Servidor de Mysql
17      return cnx #9. Retornamos la conexión a la Base de Datos
18  conectarBD()#10. Invocamos la función para mostrar el resultado final
```

# Resultado al Ejecutar el conexión.py



```
principal.py  conexion.py X
BD > conexion.py > ...
1 #Creating connection with MySQL workbench
2 import mysql.connector # 1. Lo primero que hacemos es importar el módulo que nos pe
3 def conectarBD(): # 2. Creamos la funcion conectarBD
4     ''' 3. Del módulo importado llamamos a la función connect pasando la ubicación de
5         que es 'localhost', el usuario que por defecto al instalar MySQL se creó el u
6         de ese usuario que tiene por defecto un string vacío: y la base de datos en m
7     cnx=mysql.connector.connect(host='localhost',
8                                 user='root',
9                                 passwd='',
                                
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER

( 'producto', )  
( 'termino', )

(basura) C:\Users\Acer\Desktop\app\_consola\_crud>C:/Users/Acer/anaconda3/envs/basura/python.exe c:/Users/Acer/Desktop/app\_consola\_crud/BD/conexion.py

( 'categoria', )  
( 'compania', )  
( 'factura', )  
( 'factura\_producto', )  
( 'persona', )  
( 'producto', )  
( 'termino', )

(basura) C:\Users\Acer\Desktop\app\_consola\_crud>

```
XAMPP for Windows - mysql -u root -p
Acer@DESKTOP-3K100D3 c:\xampp
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 23
Server version: 10.4.14-MariaDB mariadb.org binary distribut
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and
Type 'help;' or '\h' for help. Type '\c' to clear the curren

MariaDB [(none)]> use farmacia;
Database changed
MariaDB [farmacia]> show tables;
+-----+
| Tables_in_farmacia |
+-----+
| categoria           |
| compania            |
| factura             |
| factura_producto    |
| persona             |
| producto            |
| termino             |
+-----+
7 rows in set (0.000 sec)

MariaDB [farmacia]>
```

# Función Listar Categorías



The screenshot shows the Visual Studio Code interface. The editor window displays the file 'conexion.py' with the following Python code:

```
BD > conexion.py > listarCategorias
22 def listarCategorias():
23     l=conectarBD()
24     c=l.cursor()
25     c.execute("select * from categoria")
26     for tabla in c: #6. Mediante un for podemos ver todas las categorias existentes de la base de datos 'fa
27         print(tabla)#7. Imprime las tablas de farmacia
28     l.close() #8. Cerramos la conexión con el Servidor de Mysql
29     return l #9. Retornamos la lista de Categorías
30
31 #conectarBD()#10. Invocamos la función para mostrar el resultado final
32 listarCategorias()#10. Invocamos la función para mostrar el resultado final
33
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
(basura) C:\Users\Acer\Desktop\app_consola_crud>C:/Users/Acer/anaconda3/envs/basura/python.exe c:/Users/Acer/Desktop/a
pp_consola_crud/BD/conexion.py
(1, 'Medicamentos Naturales')
(2, 'Jarabes')
(3, 'Pastillas')

(basura) C:\Users\Acer\Desktop\app_consola_crud>
```

The status bar at the bottom indicates the file is 'master\*', the cursor is at 'Lín. 26, col. 71', and the Python version is '3.10.4 ('basura': conda)'.



## Función Agregar Categoría



Crear la función y agregarla en el archivo de conexion.py



## Función Actualizar Categoría



Crear la función y agregarla en el archivo de conexion.py

## Función Borrar Categoría



Crear la función y agregarla en el archivo de conexion.py



GRACIAS

[www.sena.edu.co](http://www.sena.edu.co)

**Si su presentación requiere incluir el logo de una marca externa, deberá ubicarlo cómo se indica en las siguientes diapositivas:**



# Título Presentación

Marca externa



**El tamaño del logo de la marca externa no debe superar el tamaño del logo SENA.**

**Marca externa**



# Título

---

Textos en Calibri, en gris oscuro. Puede reemplazar la imagen por una fotografía alusiva al tema que se está tratando. Al reemplazar la foto tenga cuidado de colocarlo debajo del logo del SENA, que debe ir en la esquina superior derecha en blanco.



Marca externa

**El tamaño del logo de la marca externa no debe superar el tamaño del logo SENA.**

**Marca externa**



Para las diapositivas con fondo de color naranja, es recomendable colocar la versión blanca del logo de la marca externa.

**Marca externa**