

## Actividad 1 de clase

Haz la misma actividad anterior, pero con otro conector, denominado “PyMySQL”. Debes abrir la conexión a base de datos, cerrarla en caso de éxito y gestionar los errores. Además, en caso de error, también debe cerrar la conexión con la base de datos.

```
usuario@usuario-B250M-DS3H: ~/actividad1_ad
usuario@usuario-B250M-DS3H:~$ mkdir actividad1_ad
usuario@usuario-B250M-DS3H:~$ cd actividad1_ad/
usuario@usuario-B250M-DS3H:~/actividad1_ad$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 90ms
  creator CPython3Posix(dest=/home/usuario/actividad1_ad/venv, clear=False, no_v
cs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/us
uario/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
hellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/actividad1_ad$ ls
venv
usuario@usuario-B250M-DS3H:~/actividad1_ad$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/actividad1_ad$ pip install PyMySQL
Collecting PyMySQL
  Downloading PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
  Downloading PyMySQL-1.1.1-py3-none-any.whl (44 kB)
    45.0/45.0 kB 1.4 MB/s eta 0:00:00
Installing collected packages: PyMySQL
Successfully installed PyMySQL-1.1.1
(venv) usuario@usuario-B250M-DS3H:~/actividad1_ad$
```

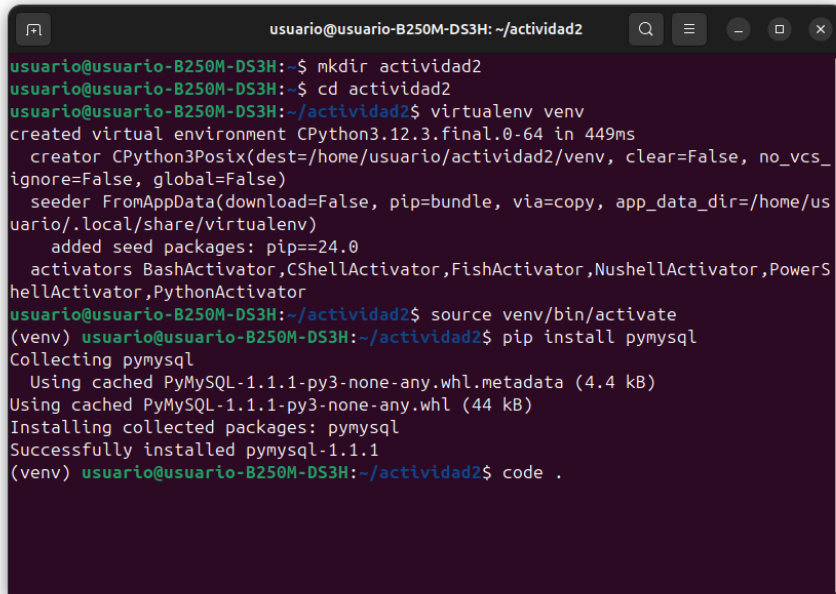
```
Welcome
actividad1.py x
actividad1.py > ...
1 import pymysql
2
3 # Parámetros de conexión
4 host = 'localhost'
5 user = 'usuario'
6 password = 'usuario'
7 database = '2DAM'
8
9 # Inicializamos la conexión
10 connection = None
11
12 try:
13     # Abrir conexión a la base de datos
14     connection = pymysql.connect(
15         host=host,
16         user=user,
17         password=password,
18         database=database
19     )
20     print("Conexión exitosa a la base de datos.")

```

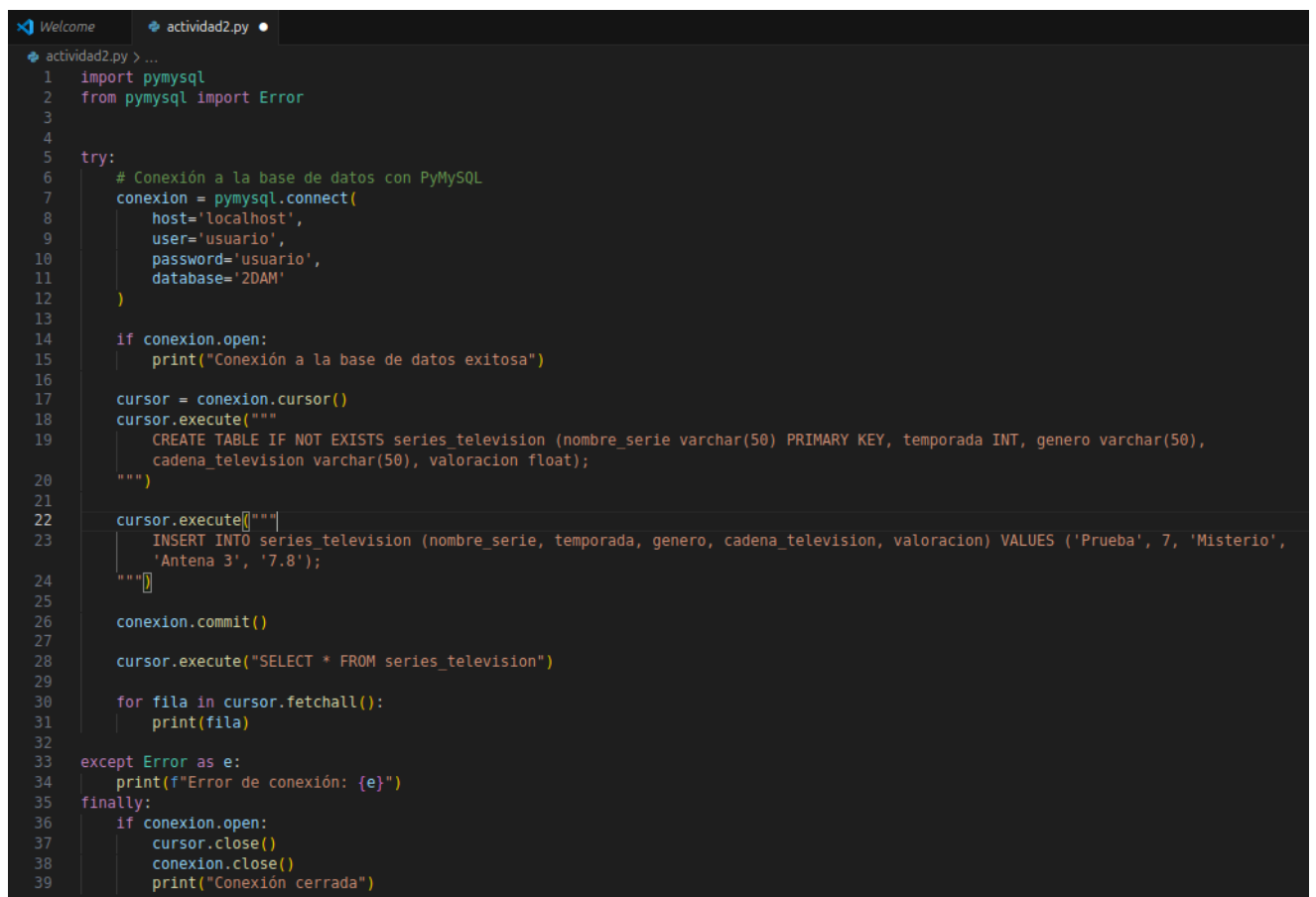
```
venvusuario@usuario-B250M-DS3H:~/actividad1_ad$ /home/usuario/actividad1_ad/venv/bin/python /home/usuario/actividad1_ad/actividad1
.py
Conexión exitosa a la base de datos.
Versión de MySQL: 8.0.39-0ubuntu0.24.04.2
Conexión cerrada.
venvusuario@usuario-B250M-DS3H:~/actividad1_ad$
```

## Actividad 2 de clase

Haz el mismo script de MySQL, pero usando el conector PyMySQL en lugar de mysql connector. El programa ha de acceder a la tabla de objetos que te haya tocado en el reparto.

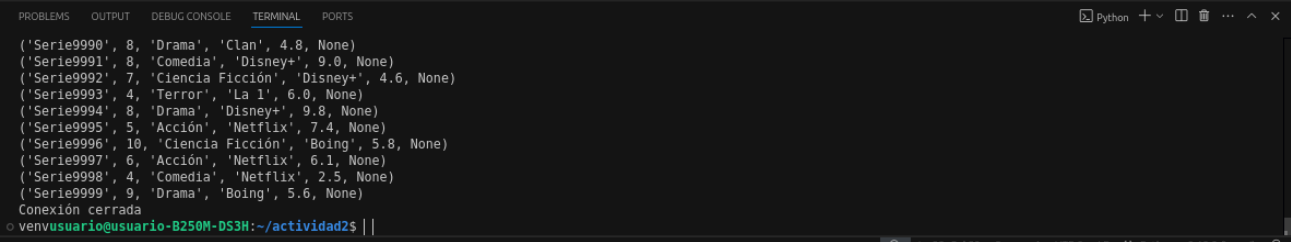


```
usuario@usuario-B250M-DS3H: ~/actividad2
usuario@usuario-B250M-DS3H:~$ mkdir actividad2
usuario@usuario-B250M-DS3H:~$ cd actividad2
usuario@usuario-B250M-DS3H:~/actividad2$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 449ms
  creator CPython3Posix(dest=/home/usuario/actividad2/venv, clear=False, no_vcs_
  ignore=False, global=False)
  seeder FromAppData(download=False, pip=24.0, via=copy, app_data_dir=/home/us
  uario/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
  hellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/actividad2$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/actividad2$ pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
(venv) usuario@usuario-B250M-DS3H:~/actividad2$ code .
```



```
Welcome actividad2.py
actividad2.py > ...
1  import pymysql
2  from pymysql import Error
3
4
5  try:
6      # Conexión a la base de datos con PyMySQL
7      conexion = pymysql.connect(
8          host='localhost',
9          user='usuario',
10         password='usuario',
11         database='2DAM'
12     )
13
14     if conexion.open:
15         print("Conexión a la base de datos exitosa")
16
17         cursor = conexion.cursor()
18         cursor.execute("""
19             CREATE TABLE IF NOT EXISTS series_television (nombre_serie varchar(50) PRIMARY KEY, temporada INT, genero varchar(50),
20             cadena_television varchar(50), valoracion float);
21         """)
22
23         cursor.execute("""
24             INSERT INTO series_television (nombre_serie, temporada, genero, cadena_television, valoracion) VALUES ('Prueba', 7, 'Misterio',
25             'Antena 3', '7.8');
26         """)
27
28         conexion.commit()
29
30         cursor.execute("SELECT * FROM series_television")
31
32         for fila in cursor.fetchall():
33             print(fila)
34
35     except Error as e:
36         print(f"Error de conexión: {e}")
37     finally:
38         if conexion.open:
39             cursor.close()
40             conexion.close()
41             print("Conexión cerrada")
```

Me salen muchos registros porque realice antes la actividad 3, por lo que hay muchos registros insertados en la tabla



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The terminal output displays a list of TV series records, each as a tuple containing a series ID, a rating, a genre, a platform, and a score. The records are as follows:

Serie	Rating	Genre	Platform	Score
Serie9990	8	Drama	Clan	4.8
Serie9991	8	Comedia	Disney+	9.0
Serie9992	7	Ciencia Ficción	Disney+	4.6
Serie9993	4	Terror	La 1	6.0
Serie9994	8	Drama	Disney+	9.8
Serie9995	5	Acción	Netflix	7.4
Serie9996	10	Ciencia Ficción	Boing	5.8
Serie9997	6	Acción	Netflix	6.1
Serie9998	4	Comedia	Netflix	2.5
Serie9999	9	Drama	Boing	5.6

Below the list, the text 'Conexión cerrada' is displayed. The terminal prompt shows the user 'venvusuario' at the machine 'usuario-B250M-D53H' in the directory '~/actividad2\$'.

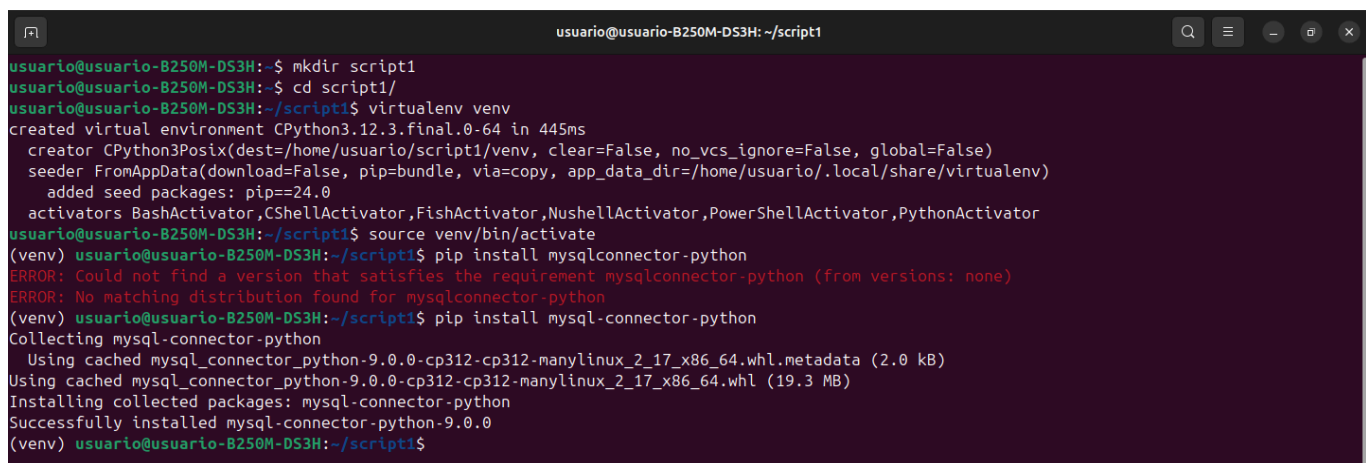
## Actividad 3 de clase

Lo primero de todo, iniciamos mysql para poder conectarnos a la base de datos para las inserciones y las consultas



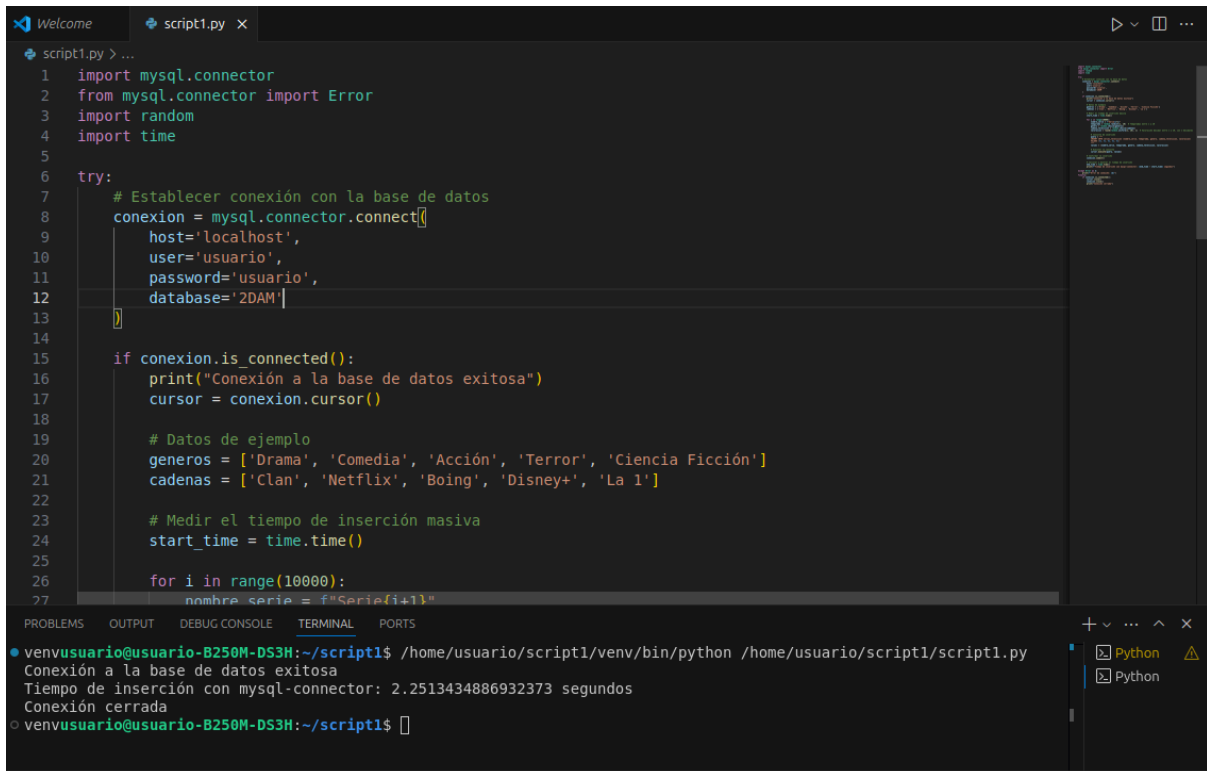
```
usuario@usuario-B250M-DS3H: ~  
usuario@usuario-B250M-DS3H:~$ mysql -u usuario -pusuario  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 8  
Server version: 8.0.39-0ubuntu0.24.04.2 (Ubuntu)  
  
Copyright (c) 2000, 2024, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql>
```

(A) Utilizando mysql-connector, midiendo el tiempo que toma cada operación.



```
usuario@usuario-B250M-DS3H: ~/script1  
usuario@usuario-B250M-DS3H:~$ mkdir script1  
usuario@usuario-B250M-DS3H:~$ cd script1/  
usuario@usuario-B250M-DS3H:~/script1$ virtualenv venv  
created virtual environment CPython3.12.3.final.0-64 in 445ms  
  creator CPython3Posix(dest=/home/usuario/script1/venv, clear=False, no_vcs_ignore=False, global=False)  
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/usuario/.local/share/virtualenv)  
  added seed packages: pip==24.0  
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator  
usuario@usuario-B250M-DS3H:~/script1$ source venv/bin/activate  
(venv) usuario@usuario-B250M-DS3H:~/script1$ pip install mysqlconnector-python  
ERROR: Could not find a version that satisfies the requirement mysqlconnector-python (from versions: none)  
ERROR: No matching distribution found for mysqlconnector-python  
(venv) usuario@usuario-B250M-DS3H:~/script1$ pip install mysql-connector-python  
Collecting mysql-connector-python  
  Using cached mysql_connector_python-9.0.0-cp312-cp312-manylinux_2_17_x86_64.whl.metadata (2.0 kB)  
Using cached mysql_connector_python-9.0.0-cp312-cp312-manylinux_2_17_x86_64.whl (19.3 MB)  
Installing collected packages: mysql-connector-python  
Successfully installed mysql-connector-python-9.0.0  
(venv) usuario@usuario-B250M-DS3H:~/script1$
```

## Ejecución 1

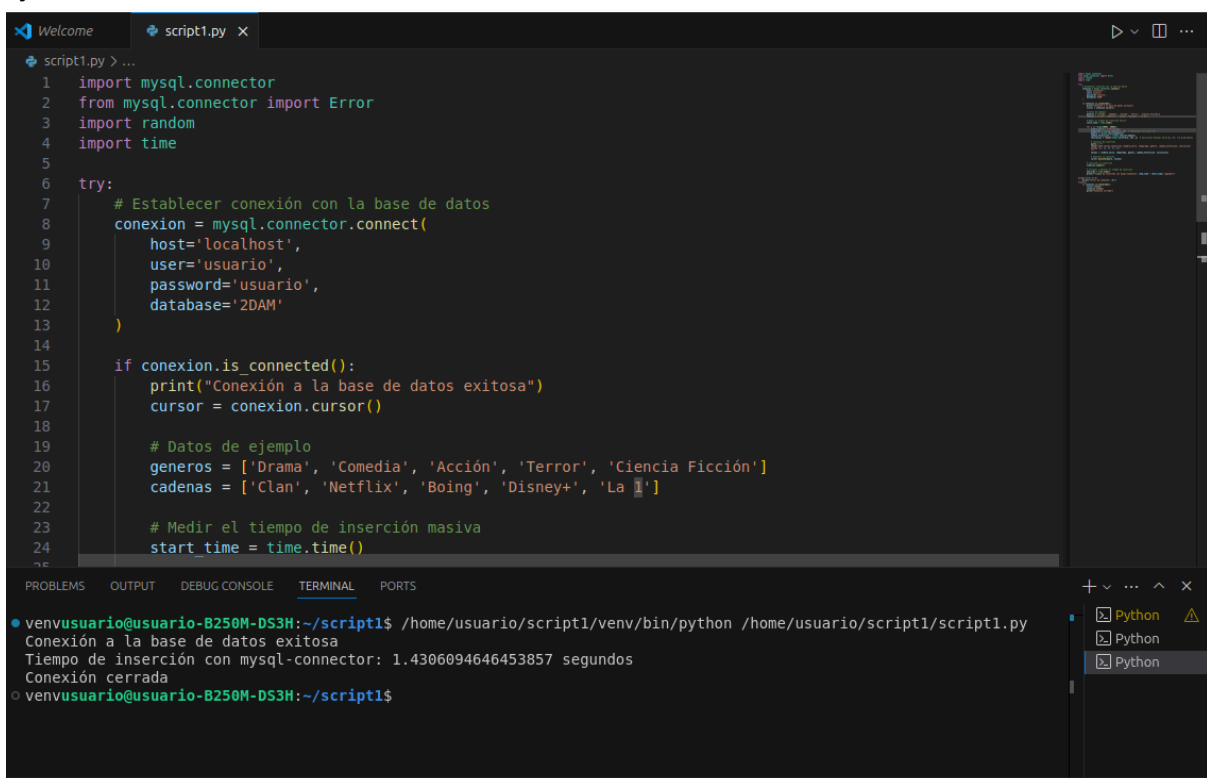


The screenshot shows a code editor with a Python script named `script1.py`. The script imports `mysql.connector`, `random`, and `time`. It attempts to connect to a MySQL database named '2DAM' on localhost using the username 'usuario' and password 'usuario'. If the connection is successful, it prints a message and creates a cursor. It then defines a list of genres and a list of streaming services. A loop is intended to insert data, but the code is partially obscured. The terminal output shows the command to run the script and the resulting messages: 'Conexión a la base de datos exitosa', 'Tiempo de inserción con mysql-connector: 2.2513434886932373 segundos', and 'Conexión cerrada'.

```
1 import mysql.connector
2 from mysql.connector import Error
3 import random
4 import time
5
6 try:
7     # Establecer conexión con la base de datos
8     conexion = mysql.connector.connect(
9         host='localhost',
10        user='usuario',
11        password='usuario',
12        database='2DAM'
13    )
14
15    if conexion.is_connected():
16        print("Conexión a la base de datos exitosa")
17        cursor = conexion.cursor()
18
19        # Datos de ejemplo
20        generos = ['Drama', 'Comedia', 'Acción', 'Terror', 'Ciencia Ficción']
21        cadenas = ['Clan', 'Netflix', 'Boing', 'Disney+', 'La 1']
22
23        # Medir el tiempo de inserción masiva
24        start_time = time.time()
25
26        for i in range(10000):
27            nombre_serie = f"Serie{i+1}"
```

venvusuario@usuario-B250M-DS3H:~/script1\$ /home/usuario/script1/venv/bin/python /home/usuario/script1/script1.py  
Conexión a la base de datos exitosa  
Tiempo de inserción con mysql-connector: 2.2513434886932373 segundos  
Conexión cerrada  
venvusuario@usuario-B250M-DS3H:~/script1\$

## Ejecución 2



This screenshot shows the same Python script as in Ejecución 1. The terminal output shows the command to run the script and the resulting messages: 'Conexión a la base de datos exitosa', 'Tiempo de inserción con mysql-connector: 1.4306094646453857 segundos', and 'Conexión cerrada'. The execution time is significantly lower than in Ejecución 1.

```
1 import mysql.connector
2 from mysql.connector import Error
3 import random
4 import time
5
6 try:
7     # Establecer conexión con la base de datos
8     conexion = mysql.connector.connect(
9         host='localhost',
10        user='usuario',
11        password='usuario',
12        database='2DAM'
13    )
14
15    if conexion.is_connected():
16        print("Conexión a la base de datos exitosa")
17        cursor = conexion.cursor()
18
19        # Datos de ejemplo
20        generos = ['Drama', 'Comedia', 'Acción', 'Terror', 'Ciencia Ficción']
21        cadenas = ['Clan', 'Netflix', 'Boing', 'Disney+', 'La 1']
22
23        # Medir el tiempo de inserción masiva
24        start_time = time.time()
25
26        for i in range(10000):
27            nombre_serie = f"Serie{i+1}"
```

venvusuario@usuario-B250M-DS3H:~/script1\$ /home/usuario/script1/venv/bin/python /home/usuario/script1/script1.py  
Conexión a la base de datos exitosa  
Tiempo de inserción con mysql-connector: 1.4306094646453857 segundos  
Conexión cerrada  
venvusuario@usuario-B250M-DS3H:~/script1\$

(A) Utilizando pymysql, midiendo el tiempo que toma cada operación.

```
usuario@usuario-B250M-DS3H: ~/script2
usuario@usuario-B250M-DS3H:~$ mkdir script2
usuario@usuario-B250M-DS3H:~$ sudo apt install virtualenv
[sudo] contraseña para usuario:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
virtualenv ya está en su versión más reciente (20.25.0+ds-2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
usuario@usuario-B250M-DS3H:~$ cd script2
usuario@usuario-B250M-DS3H:~/script2$ sudo apt install virtualenv
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
virtualenv ya está en su versión más reciente (20.25.0+ds-2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
usuario@usuario-B250M-DS3H:~/script2$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 90ms
  creator CPython3Posix(dest=/home/usuario/script2/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/usuario/.local/share/virtualenv)
    added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/script2$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/script2$ pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
(venv) usuario@usuario-B250M-DS3H:~/script2$
```

## Ejecución 1 y 2

```
Welcome script2.py x
script2.py > ...
6 # Establecer conexión con la base de datos usando pymysql
7 conexion = pymysql.connect(
8     host='localhost',
9     user='usuario',
10    password='usuario',
11    database='2DAM'
12 )
13
14 print("Conexión a la base de datos exitosa")
15 cursor = conexion.cursor()
16
17 # Datos de ejemplo
18 generos = ['Drama', 'Comedia', 'Acción', 'Terror', 'Ciencia Ficción']
19 cadenas = ['Clan', 'Netflix', 'Boing', 'Disney+', 'La 1']
20
21 # Medir el tiempo de inserción masiva
22 start_time = time.time()
23
24 for i in range(20001, 30001):
25     nombre_serie = f"Serie{i}"
26     temporada = random.randint(1, 10) # Temporadas entre 1 y 10
27     genero = random.choice(generos)
28     cadena_television = random.choice(cadenas)
29     valoracion = round(random.uniform(1, 10), 1) # Valoración decimal entre 1 y 10, con 1 decimal
30
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - [ ] ... ^ x
• venvusuario@usuario-B250M-DS3H:~/script2$ /home/usuario/script2/venv/bin/python /home/usuario/script2/script2.py
Conexión a la base de datos exitosa
Tiempo de inserción con pymysql: 1.2208423614501953 segundos
Conexión cerrada
• venvusuario@usuario-B250M-DS3H:~/script2$ /home/usuario/script2/venv/bin/python /home/usuario/script2/script2.py
Conexión a la base de datos exitosa
Tiempo de inserción con pymysql: 1.141266107559204 segundos
Conexión cerrada
• venvusuario@usuario-B250M-DS3H:~/script2$
```

(B) Utilizando mysql-connector, midiendo el tiempo que toma cada operación.

```
usuario@usuario-B250M-DS3H: ~/script3
Descargas      java_error_in_studio_.hprof  script1
usuario@usuario-B250M-DS3H:~$ mkdir script3
usuario@usuario-B250M-DS3H:~$ cd script3
usuario@usuario-B250M-DS3H:~/script3$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 414ms
  creator CPython3Posix(dest=/home/usuario/script3/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/usuario/.local/share/virtualenv)
    added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/script3$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/script3$ pip install mysql-connector-python
Collecting mysql-connector-python
  Using cached mysql_connector_python-9.0.0-cp312-cp312-manylinux_2_17_x86_64.whl.metadata (2.0 kB)
Using cached mysql_connector_python-9.0.0-cp312-cp312-manylinux_2_17_x86_64.whl (19.3 MB)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.0.0
(venv) usuario@usuario-B250M-DS3H:~/script3$ code .
```

## Ejecución 1 y 2

```
script3.py > ...
1  import mysql.connector
2  from mysql.connector import Error
3  import time
4
5  try:
6      # Establecer conexión con la base de datos
7      conexion = mysql.connector.connect(
8          host='localhost',
9          user='usuario',
10         password='usuario',
11         database='2DAM'
12     )
13
14     # Verifico si la conexión fue exitosa
15     if conexion.is_connected():
16         print("Conexión a la base de datos exitosa")
17         cursor = conexion.cursor()
18
19         start_time = time.time()
20
21         query = "SELECT * FROM series_television;"
22         cursor.execute(query) # Ejecuto la consulta
23
24         # Recupero todos los registros de la base de datos usando fetchall

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v

```
• venvusuario@usuario-B250M-DS3H:~/script3$ /home/usuario/script3/venv/bin/python /home/usuario/script3/script3.py
Conexión a la base de datos exitosa
Se recuperaron 40005 registros.
Tiempo de lectura con mysql-connector: 0.08075165748596191 segundos
Conexión cerrada
• venvusuario@usuario-B250M-DS3H:~/script3$ /home/usuario/script3/venv/bin/python /home/usuario/script3/script3.py
Conexión a la base de datos exitosa
Se recuperaron 40005 registros.
Tiempo de lectura con mysql-connector: 0.07003474235534668 segundos
Conexión cerrada
```



(B) Utilizando mysql-connector, midiendo el tiempo que toma cada operación.

```

usuario@usuario-B250M-DS3H: ~/script4
usuario@usuario-B250M-DS3H:~$ mkdir script4
usuario@usuario-B250M-DS3H:~$ cd script4
usuario@usuario-B250M-DS3H:~/script4$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 93ms
  creator CPython3Posix(dest=/home/usuario/script4/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/usuario/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/script4$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/script4$ pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
  Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
(venv) usuario@usuario-B250M-DS3H:~/script4$ code .

```

Ejecución 1 y 2

```

Welcome  script4.py x
script4.py > ...
1  import pymysql
2  import time
3
4  try:
5      # Establecer conexión con la base de datos
6      conexion = pymysql.connect(
7          host='localhost',
8          user='usuario',
9          password='usuario',
10         database='2DAM'
11     )
12
13     # Verifico si la conexión fue exitosa
14     if conexion.open:
15         print("Conexión a la base de datos exitosa")
16         cursor = conexion.cursor()
17
18         # Inicio el cronómetro para medir cuánto tarda en recuperar los datos
19         start_time = time.time()
20
21         # Consulta SQL para seleccionar todos los registros de la tabla 'Comidas'
22         query = "SELECT * FROM series_television;"
23         cursor.execute(query) # Ejecuto la consulta
24
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
Python + v  [ ]  [x]
• venvusuario@usuario-B250M-DS3H:~/script4$ /home/usuario/script4/venv/bin/python /home/usuario/script4/script4.py
Conexión a la base de datos exitosa
Tiempo de lectura con pymysql: 0.26132965087890625 segundos
Conexión cerrada
• venvusuario@usuario-B250M-DS3H:~/script4$ /home/usuario/script4/venv/bin/python /home/usuario/script4/script4.py
Conexión a la base de datos exitosa
Tiempo de lectura con pymysql: 0.2508730888366699 segundos
Conexión cerrada
• venvusuario@usuario-B250M-DS3H:~/script4$

```



## Actividad 4 de clase

```
usuario@usuario-B250M-DS3H: ~/actividad4
usuario@usuario-B250M-DS3H:~$ mkdir actividad4
usuario@usuario-B250M-DS3H:~$ cd actividad4
usuario@usuario-B250M-DS3H:~/actividad4$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 91ms
  creator CPython3Posix(dest=/home/usuario/actividad4/venv, clear=False, no_vcs_
  ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/us
  uario/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
  hellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/actividad4$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/actividad4$ pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
(venv) usuario@usuario-B250M-DS3H:~/actividad4$ code .
(venv) usuario@usuario-B250M-DS3H:~/actividad4$
```

```
Welcome  actividad4.py x
actividad4.py > ...
1  import pymysql
2  import time
3
4  # Conexión a la base de datos
5  connection = pymysql.connect(
6      host='localhost',
7      user='usuario',
8      password='usuario',
9      database='20AM'
10 )
11
12 try:
13     with connection.cursor() as cursor:
14         # Paso 1: Crear la tabla Actores
15         sql_crear_actores = """
16         CREATE TABLE IF NOT EXISTS Actores (
17             id_actor INT AUTO INCREMENT PRIMARY KEY,
18             nombre_actor VARCHAR(100) NOT NULL,
19             edad INT,
20             nacionalidad VARCHAR(50)
21         );
22         """
23         cursor.execute(sql_crear_actores)
24         print("Tabla 'Actores' creada.")
25
26         # Paso 2: Modificar la tabla Series_Television para añadir la columna actor_id
27         # como clave foránea que se relaciona con la tabla actores
28         sql_alter_series = """
29         ALTER TABLE series_television
30         ADD COLUMN actor_id INT,
31         ADD CONSTRAINT fk_actor
32         FOREIGN KEY (actor_id) REFERENCES Actores(id_actor);
33         """
34         cursor.execute(sql_alter_series)
35         print(f"Columna 'actor_id' añadida a 'series_television' y relación entre las tablas creada.")
36
37         # Confirmar cambios
38         connection.commit()
39         print("Cambios confirmados en la base de datos.")
40
41 except pymysql.MySQLError as e:
42     print(f"Ocurrió un error al ejecutar las instrucciones SQL: {e}")
43
44 finally:
45     connection.close()
46     print("Conexión cerrada.")
```

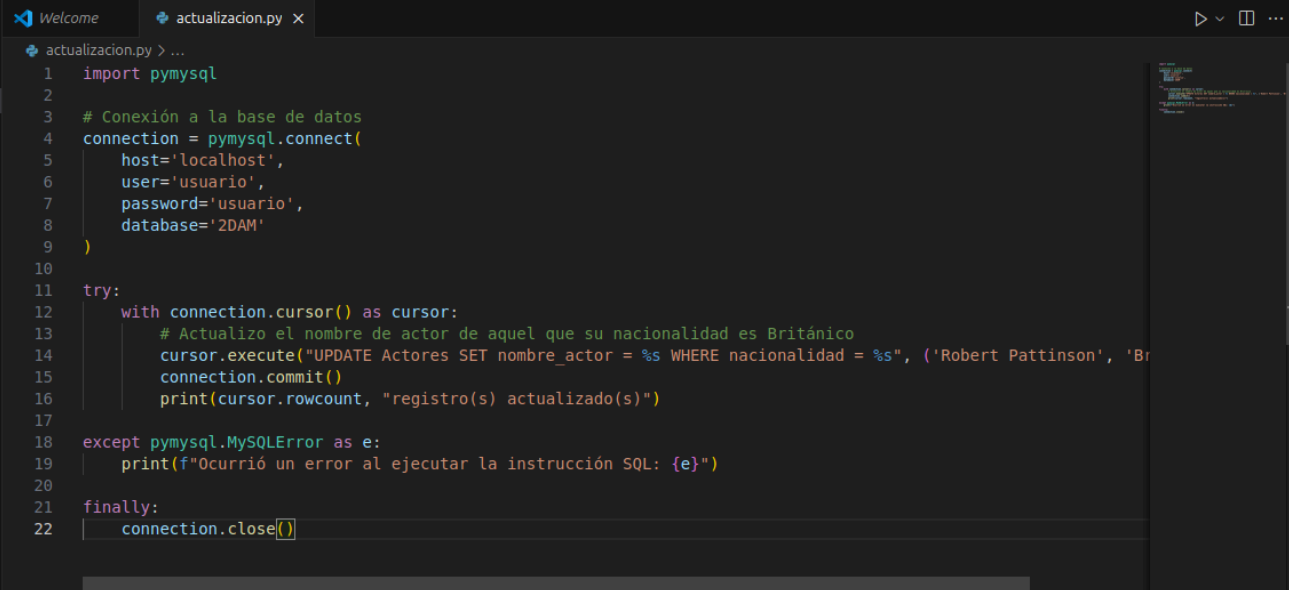
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
venvusuario@usuario-B250M-DS3H:~/actividad4$ /home/usuario/actividad4/venv/bin/python /home/usuario/actividad4/actividad4.py
Tabla 'Actores' creada.
Columna 'actor_id' añadida a 'series television' y relación entre las tablas creada.
Cambios confirmados en la base de datos.
Conexión cerrada.
venvusuario@usuario-B250M-DS3H:~/actividad4$
```

## Actividad 5 de clase

En esta actividad utilizaré la nueva tabla Actores creada anteriormente a la que le inserto 5 registros para poder realizar esta actividad y que podamos observar los cambios que se realizan

```
usuario@usuario-B250M-DS3H: ~/actividad5
usuario@usuario-B250M-DS3H:~$ mkdir actividad5
usuario@usuario-B250M-DS3H:~$ cd a
actividad1_ad/ actividad4/ actividad5/ alumnos/
usuario@usuario-B250M-DS3H:~$ cd actividad5
usuario@usuario-B250M-DS3H:~/actividad5$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 91ms
  creator CPython3Posix(dest=/home/usuario/actividad5/venv, clear=False, no_vcs_
ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/us
uario/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
hellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/actividad5$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/actividad5$ pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
(venv) usuario@usuario-B250M-DS3H:~/actividad5$ code .
```

## ACTUALIZACIÓN



```
1 import pymysql
2
3 # Conexión a la base de datos
4 connection = pymysql.connect(
5     host='localhost',
6     user='usuario',
7     password='usuario',
8     database='2DAM'
9 )
10
11 try:
12     with connection.cursor() as cursor:
13         # Actualizo el nombre de actor de aquel que su nacionalidad es Británico
14         cursor.execute("UPDATE Actores SET nombre_actor = %s WHERE nacionalidad = %s", ('Robert Pattinson', 'Br
15         connection.commit()
16         print(cursor.rowcount, "registro(s) actualizado(s)")
17
18 except pymysql.MySQLError as e:
19     print(f"Ocurrió un error al ejecutar la instrucción SQL: {e}")
20
21 finally:
22     connection.close()
```

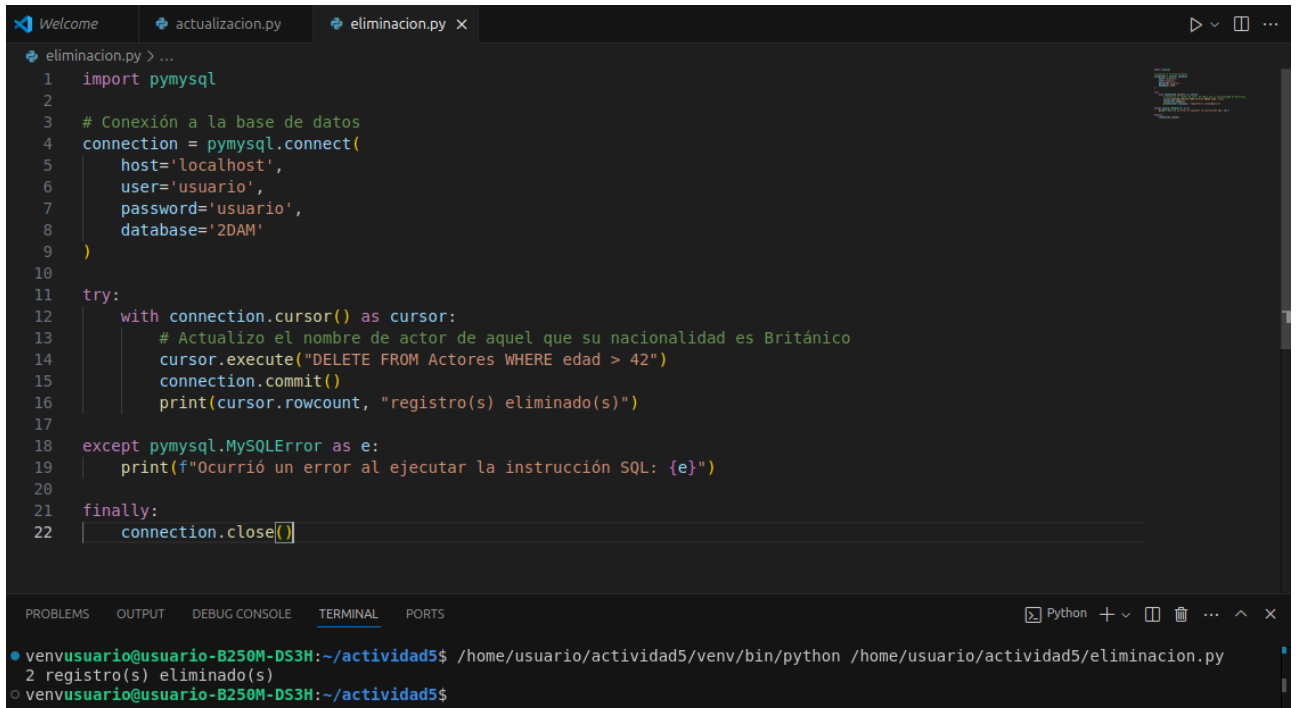
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v ... ^ x

```
● venvusuario@usuario-B250M-D53H:~/actividad5$ /home/usuario/actividad5/venv/bin/python /home/usuario/actividad5/actualizacion.py
1 registro(s) actualizado(s)
○ venvusuario@usuario-B250M-D53H:~/actividad5$
```

```
mysql> SELECT * FROM Actores;
+-----+-----+-----+-----+
| id_actor | nombre_actor | edad | nacionalidad |
+-----+-----+-----+-----+
| 1 | Robert Downey Jr. | 58 | Estadounidense |
| 2 | Scarlett Johansson | 39 | Estadounidense |
| 3 | Chris Hemsworth | 41 | Australiano |
| 4 | Tom Hiddleston | 43 | Británico |
| 5 | Zendaya | 28 | Estadounidense |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)
```

```
mysql> SELECT * FROM Actores;
+-----+-----+-----+-----+
| id_actor | nombre_actor | edad | nacionalidad |
+-----+-----+-----+-----+
| 1 | Robert Downey Jr. | 58 | Estadounidense |
| 2 | Scarlett Johansson | 39 | Estadounidense |
| 3 | Chris Hemsworth | 41 | Australiano |
| 4 | Robert Pattinson | 43 | Británico |
| 5 | Zendaya | 28 | Estadounidense |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)
```

## ELIMINACIÓN



```
1 import pymysql
2
3 # Conexión a la base de datos
4 connection = pymysql.connect(
5     host='localhost',
6     user='usuario',
7     password='usuario',
8     database='2DAM'
9 )
10
11 try:
12     with connection.cursor() as cursor:
13         # Actualizo el nombre de actor de aquel que su nacionalidad es Británico
14         cursor.execute("DELETE FROM Actores WHERE edad > 42")
15         connection.commit()
16         print(cursor.rowcount, "registro(s) eliminado(s)")
17 except pymysql.MySQLError as e:
18     print(f"Ocurrió un error al ejecutar la instrucción SQL: {e}")
19
20 finally:
21     connection.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - - - - -

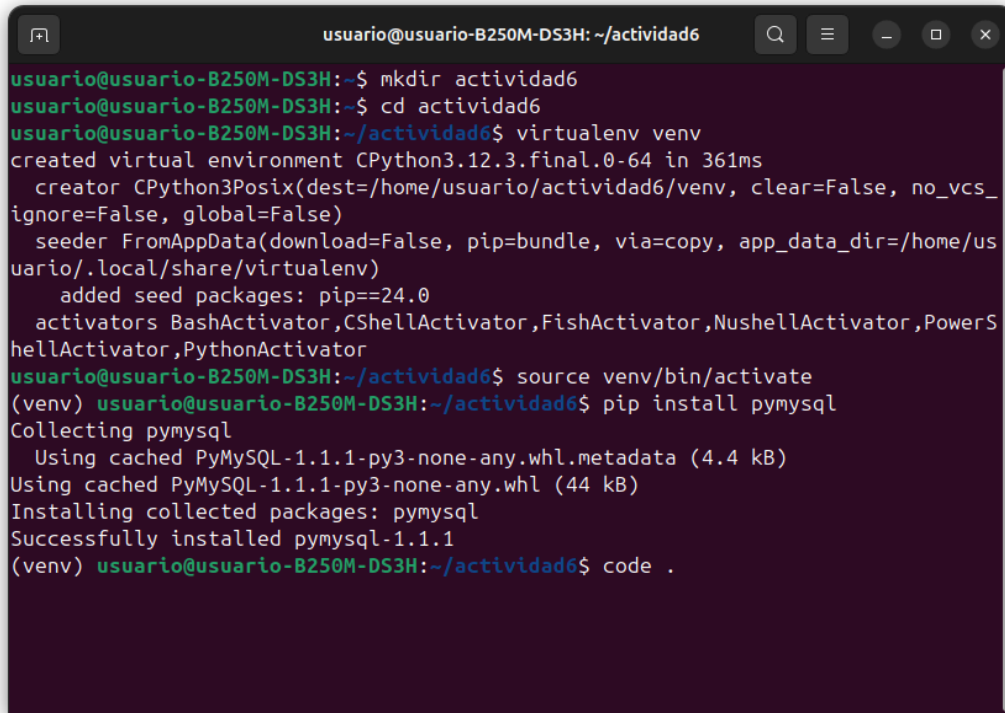
```
• venvusuario@usuario-B250M-DS3H:~/actividad5$ /home/usuario/actividad5/venv/bin/python /home/usuario/actividad5/eliminacion.py
2 registro(s) eliminado(s)
• venvusuario@usuario-B250M-DS3H:~/actividad5$
```

```
mysql> SELECT * FROM Actores;
+-----+-----+-----+-----+
| id_actor | nombre_actor | edad | nacionalidad |
+-----+-----+-----+-----+
| 1 | Robert Downey Jr. | 58 | Estadounidense |
| 2 | Scarlett Johansson | 39 | Estadounidense |
| 3 | Chris Hemsworth | 41 | Australiano |
| 4 | Robert Pattinson | 43 | Británico |
| 5 | Zendaya | 28 | Estadounidense |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)

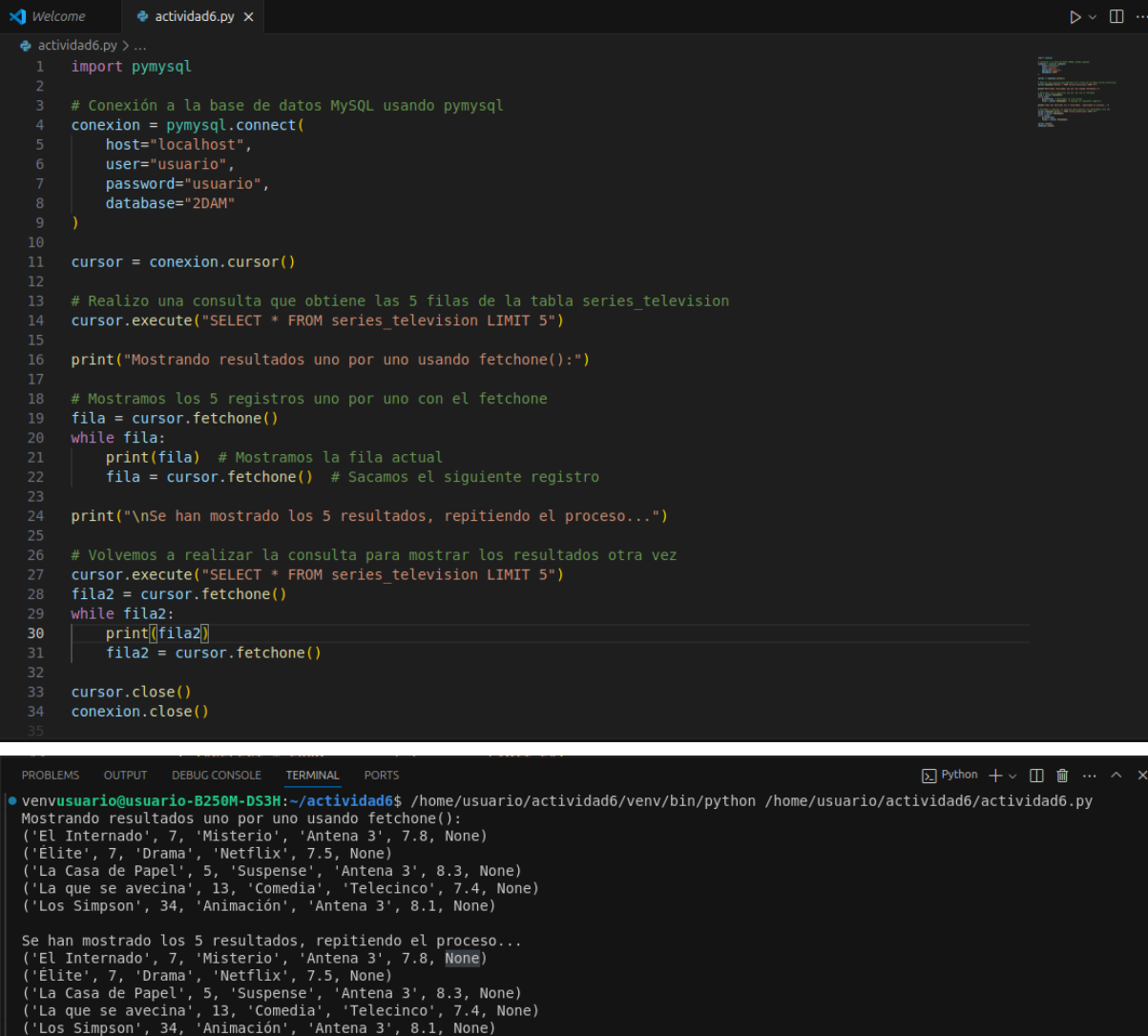
mysql> SELECT * FROM Actores;
+-----+-----+-----+-----+
| id_actor | nombre_actor | edad | nacionalidad |
+-----+-----+-----+-----+
| 2 | Scarlett Johansson | 39 | Estadounidense |
| 3 | Chris Hemsworth | 41 | Australiano |
| 5 | Zendaya | 28 | Estadounidense |
+-----+-----+-----+-----+
3 rows in set (0,01 sec)
```

## Actividad 6 de clase

Cada alumno@ partirá de la tabla que se le haya asignado (restaurantes, películas, etc.) con sólo 5 filas. Ahora cambiamos el conector a pymysql. Mostraremos los resultados uno a uno y cuando se terminen de mostrar los 5 resultados, mostraremos de nuevo dichos 5 resultados. El programa sólo debe usar fetchone() y además no debe almacenar en ninguna estructura de datos los datos concernientes a más de una fila.



```
usuario@usuario-B250M-DS3H: ~/actividad6
usuario@usuario-B250M-DS3H:~$ mkdir actividad6
usuario@usuario-B250M-DS3H:~$ cd actividad6
usuario@usuario-B250M-DS3H:~/actividad6$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 361ms
  creator CPython3Posix(dest=/home/usuario/actividad6/venv, clear=False, no_vcs_
ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/us
uario/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerS
hellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/actividad6$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/actividad6$ pip install pymysql
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
(venv) usuario@usuario-B250M-DS3H:~/actividad6$ code .
```



The image shows a Visual Studio Code editor window with a file named `actividad6.py`. The script connects to a MySQL database using `pymysql`, executes a query to select 5 rows from the `series_television` table, and prints the results using `fetchone()`. The script is then executed in a terminal window, which shows the output of the program.

```
1 import pymysql
2
3 # Conexión a la base de datos MySQL usando pymysql
4 conexion = pymysql.connect(
5     host="localhost",
6     user="usuario",
7     password="usuario",
8     database="2DAM"
9 )
10
11 cursor = conexion.cursor()
12
13 # Realizo una consulta que obtiene las 5 filas de la tabla series_television
14 cursor.execute("SELECT * FROM series_television LIMIT 5")
15
16 print("Mostrando resultados uno por uno usando fetchone():")
17
18 # Mostramos los 5 registros uno por uno con el fetchone
19 fila = cursor.fetchone()
20 while fila:
21     print(fila) # Mostramos la fila actual
22     fila = cursor.fetchone() # Sacamos el siguiente registro
23
24 print("\nSe han mostrado los 5 resultados, repitiendo el proceso...")
25
26 # Volvemos a realizar la consulta para mostrar los resultados otra vez
27 cursor.execute("SELECT * FROM series_television LIMIT 5")
28 fila2 = cursor.fetchone()
29 while fila2:
30     print(fila2)
31     fila2 = cursor.fetchone()
32
33 cursor.close()
34 conexion.close()
35
```

The terminal output shows the execution of the script. It prints the results of the query, which are 5 rows of data from the `series_television` table. The output is repeated twice, as indicated by the print statement in the script.

```
● venvusuario@usuario-B250M-DS3H:~/actividad6$ /home/usuario/actividad6/venv/bin/python /home/usuario/actividad6/actividad6.py
Mostrando resultados uno por uno usando fetchone():
('El Internado', 7, 'Misterio', 'Antena 3', 7.8, None)
('Elite', 7, 'Drama', 'Netflix', 7.5, None)
('La Casa de Papel', 5, 'Suspense', 'Antena 3', 8.3, None)
('La que se avecina', 13, 'Comedia', 'Telecinco', 7.4, None)
('Los Simpson', 34, 'Animación', 'Antena 3', 8.1, None)

Se han mostrado los 5 resultados, repitiendo el proceso...
('El Internado', 7, 'Misterio', 'Antena 3', 7.8, None)
('Elite', 7, 'Drama', 'Netflix', 7.5, None)
('La Casa de Papel', 5, 'Suspense', 'Antena 3', 8.3, None)
('La que se avecina', 13, 'Comedia', 'Telecinco', 7.4, None)
('Los Simpson', 34, 'Animación', 'Antena 3', 8.1, None)
```





## Actividad 8 de clase

```
usuario@usuario-B250M-DS3H: ~/actividad8
usuario@usuario-B250M-DS3H:~$ mkdir actividad8
usuario@usuario-B250M-DS3H:~$ cd actividad8
usuario@usuario-B250M-DS3H:~/actividad8$ virtualenv venv
created virtual environment CPython3.12.3.final.0-64 in 92ms
  creator CPython3Posix(dest=/home/usuario/actividad8/venv, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/home/usuario/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
usuario@usuario-B250M-DS3H:~/actividad8$ source venv/bin/activate
(venv) usuario@usuario-B250M-DS3H:~/actividad8$ pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.1.0-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (6.0 kB)
  Downloading mysql_connector_python-9.1.0-cp312-cp312-manylinux_2_28_x86_64.whl (34.4 MB)
34.4/34.4 MB 9.5 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.1.0
(venv) usuario@usuario-B250M-DS3H:~/actividad8$ code .
```

Creamos el proceso en la consola de mysql

```
mysql> DELIMITER //
mysql>
mysql> CREATE PROCEDURE contar_actores_por_nacionalidad(IN nacionalidad_actor VARCHAR(50))
-> BEGIN
->     SELECT COUNT(*) FROM Actores WHERE nacionalidad = nacionalidad_actor;
-> END //
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> DELIMITER ;
```

```
Welcome  actividad8.py x
actividad8.py > ...
1  import mysql.connector
2
3
4  conexion = mysql.connector.connect(
5      host="localhost",
6      user="usuario",
7      password="usuario",
8      database="2DAM"
9  )
10
11 if conexion.is_connected():
12     print("Conexión exitosa a la base de datos")
13
14 cursor = conexion.cursor()
15
16 # Definimos un parámetro para el procedimiento almacenado, en este caso "Estadounidense"
17 # Este parámetro representa la nacionalidad de los actores que queremos contar
18 parametro = "Estadounidense"
19
20 # Llamamos al procedimiento almacenado 'contar_actores_por_nacionalidad', pasando el parámetro de entrada
21 cursor.callproc("contar_actores_por_nacionalidad", (parametro,))
22
23 # Recorremos los resultados devueltos por el procedimiento almacenado
24 for resultado in cursor.stored_results():
25     datos = resultado.fetchone() # Obtenemos una fila del resultado
26     # Imprimimos el número de actores con la nacionalidad indicada
27     print(f"Número de actores de nacionalidad {parametro}: {datos[0]}")
28
29 cursor.close()
30
31 conexion.close()
32 print("Conexión cerrada")
33

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● venvusuario@usuario-B250M-DS3H:~/actividad8$ /home/usuario/actividad8/venv/bin/python /home/usuario/actividad8/actividad8.py
Conexión exitosa a la base de datos
Número de actores de nacionalidad Estadounidense: 4
Conexión cerrada
○ venvusuario@usuario-B250M-DS3H:~/actividad8$ ||
```