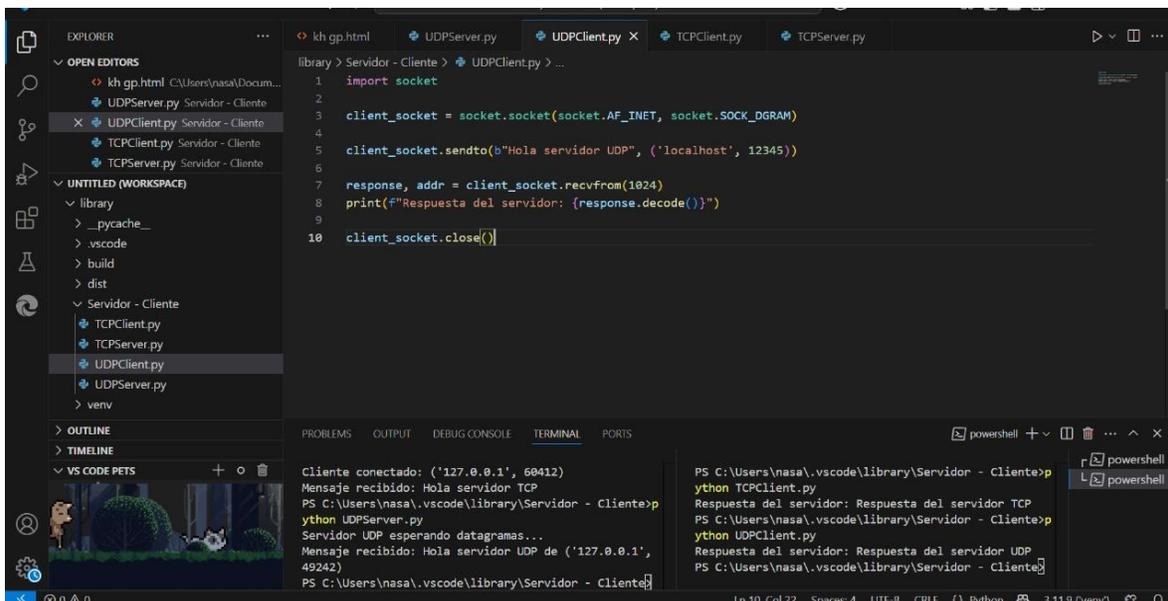


UNIVERSIDAD DEL SURESTE
CAMPUS COMITAN
FRANCISCO ARGUETA
MATERIA: COMUNICACIONES

IMPLEMENTAR UNA COMUNICACIÓN CLIENTE-SERVIDOR TCP

Durante la implementación de nuestro cliente servidor utilizamos una aplicación de la cual intervino otra persona ya que en nuestro código estaba un poco no convencional a lo que se tenía planeado y en estas fotos reflejamos paso a paso como nos salió y del paso a paso de nuestro código.



```
library > Servidor - Cliente > UDPClient.py > ...
1 import socket
2
3 client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
4
5 client_socket.sendto(b"Hola servidor UDP", ('localhost', 12345))
6
7 response, addr = client_socket.recvfrom(1024)
8 print(f"Respuesta del servidor: {response.decode()}")
9
10 client_socket.close()
```

```
Cliente conectado: ('127.0.0.1', 60412)
Mensaje recibido: Hola servidor TCP
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
y python UDPServer.py
Servidor UDP esperando datagramas...
Mensaje recibido: Hola servidor UDP de ('127.0.0.1',
49242)
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
```

```
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
y python TCPClient.py
Respuesta del servidor: Respuesta del servidor TCP
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
y python UDPClient.py
Respuesta del servidor: Respuesta del servidor UDP
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
```

En esta imagen vemos que nuestro servidor estaba esperando respuesta para continuar.

```

library > Servidor - Cliente > UDPServer.py > ...
1 import socket
2
3 server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
4 server_socket.bind(('localhost', 12345))
5 print("Servidor UDP esperando datagramas...")
6
7 data, addr = server_socket.recvfrom(1024)
8 print(f"Mensaje recibido: {data.decode()} de {addr}")
9
10 server_socket.sendto(b"Respuesta del servidor UDP", addr)
11 server_socket.close()

```

```

PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
yhton TCPClient.py
Respuesta del servidor: Respuesta del servidor TCP
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
yhton UDPClient.py
Respuesta del servidor: Respuesta del servidor UDP
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>

```

Acá en esta captura observamos que ya tenia conexión con el servidor y el cliente

```

library > Servidor - Cliente > UDPServer.py > ...
1 import socket
2
3 server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
4 server_socket.bind(('localhost', 12345))
5 print("Servidor UDP esperando datagramas...")
6
7 data, addr = server_socket.recvfrom(1024)
8 print(f"Mensaje recibido: {data.decode()} de {addr}")
9
10 server_socket.sendto(b"Respuesta del servidor UDP", addr)
11 server_socket.close()

```

```

PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
yhton TCPClient.py
Respuesta del servidor: Respuesta del servidor TCP
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>
yhton UDPClient.py
Respuesta del servidor: Respuesta del servidor UDP
PS C:\Users\nasa\.vscode\library\Servidor - Cliente>

```

Y es esta imagen ya observamos como se conectan y logramos el en nuestra estructura de nuestra conexión si sirve para cualquier servidor y con la que pueda surgir en las conexiones, se utilizo varias formas de códigos ya que durante el proceso se eliminara y fue a error y prueba.

Se utilizo ayuda de las conexiones tcp.