

Torres Abonce Luis Miguel

Se hizo un código en python donde se tiene el control de los clientes, el servidor muestra y da la posibilidad de desconectar a cualquier usuario.

Capturas Pantalla.

Mostrar los clientes conectados.

```
Cliente2.py
Ejercicio > Cliente2.py
10 TCPClientSocket.recv(buffer_size)#esperar desconexión

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
File "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py", line 10, in module
TCPClientSocket.recv(buffer_size)#esperar desconexión
ConnectionError: [WinError 10054] Se ha forzado la interrupción de una conexión existente por el host remoto
PS C:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1>
PS C:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1> & C:\Users\Luis\
/AppData/Local/Programs/Python/Python311/python.exe "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py"
Conectado al servidor
[]

Cliente2.py
Ejercicio >
1 PORT = 65432 # Puerto del servidor
2 buffer_size = 1024
3
4 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as TCPClientSocket:
5     TCPClientSocket.connect((HOST, PORT))
6     print("Conectado al servidor")
7
8
9
10 TCPClientSocket.recv(buffer_size)#esperar desconexión

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
File "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py", line 10, in module
TCPClientSocket.recv(buffer_size)#esperar desconexión
ConnectionError: [WinError 10054] Se ha forzado la interrupción de una conexión existente por el host remoto
PS C:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1> & C:\Users\Luis\
/AppData/Local/Programs/Python/Python311/python.exe "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py"
Conectado al servidor
[]

Servidor.py
Practical > Ejercicio > Servidor.py
1 import socket
2
3 HOST = "127.0.0.1" # Dirección de la interfaz de loopback estándar (localhost)
4 PORT = 65432 # Puerto que usa el cliente (los puertos sin privilegios son > 1023)
5 buffer_size = 1024
6
7 def desconexión(cliente):
8     cliente.close()
9     print("Cliente desconectado")
10    clientes = []#lista de clientes
11
12 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as TCPServerSocket:
13     TCPServerSocket.bind((HOST, PORT))
14     while True:
15         TCPServerSocket.listen()
16         print("El servidor TCP está disponible y en espera de solicitudes")
17         conn, addr = TCPServerSocket.accept()#acepta la conexión
18         clientes.append((conn, addr))#agregar al cliente a la lista
19         for i, (cliente_conn, cliente_addr) in enumerate(clientes):#imprime los clientes
20             print(f"Cliente {i + 1}: {cliente_addr}")
21         opcion = input("Elija una opción:\n1. Desconectar cliente\n2. Continuar servidor")
22         if opcion=="1":
23             desconectar = input("Cliente a desconectar (0 cancelar): ")
24             if desconectar == "0":
25                 print("cancelado")
26             else:
27                 try:
28                     num_cliente = int(desconectar)-1
29                     if 0 <= num_cliente < len(clientes):
30                         desconexión(clientes[num_cliente][0])
31                         del clientes[num_cliente]#borrar al cliente
32                     else:
33                         print("Número de cliente no válido.")
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Desconectar a un cliente que se elija.

```
app-client.py
Ejercicio > Cliente2.py
10 TCPClientSocket.recv(buffer_size)#esperar desconexión

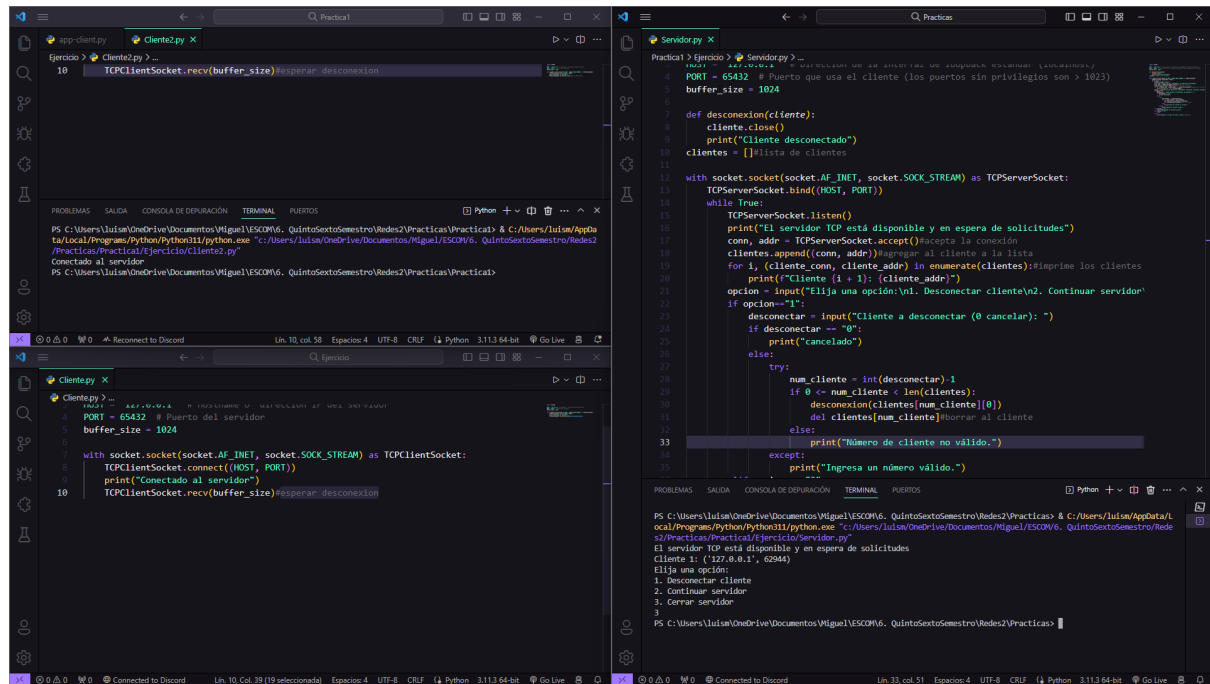
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
File "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py", line 10, in module
TCPClientSocket.recv(buffer_size)#esperar desconexión
ConnectionError: [WinError 10054] Se ha forzado la interrupción de una conexión existente por el host remoto
PS C:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1>
PS C:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1> & C:\Users\Luis\
/AppData/Local/Programs/Python/Python311/python.exe "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py"
Conectado al servidor
[]

Cliente2.py
Ejercicio >
1 HOST = 127.0.0.1 # Dirección de la interfaz de loopback estándar (localhost)
2 PORT = 65432 # Puerto del servidor
3 buffer_size = 1024
4
5 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as TCPClientSocket:
6     TCPClientSocket.connect((HOST, PORT))
7     print("Conectado al servidor")
8
9
10 TCPClientSocket.recv(buffer_size)#esperar desconexión

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
File "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py", line 10, in module
TCPClientSocket.recv(buffer_size)#esperar desconexión
ConnectionError: [WinError 10054] Se ha forzado la interrupción de una conexión existente por el host remoto
PS C:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1>
PS C:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1> & C:\Users\Luis\
/AppData/Local/Programs/Python/Python311/python.exe "c:\Users\Luis\OneDrive\Documents\Wlguel\ESCOM6. QuintoSemestre\Vedes2\Practicas\PracticalVejercicio1\iente2.py"
Conectado al servidor
[]

Servidor.py
Practical > Ejercicio > Servidor.py
1 import socket
2
3 HOST = "127.0.0.1" # Dirección de la interfaz de loopback estándar (localhost)
4 PORT = 65432 # Puerto que usa el cliente (los puertos sin privilegios son > 1023)
5 buffer_size = 1024
6
7 def desconexión(cliente):
8     cliente.close()
9     print("Cliente desconectado")
10    clientes = []#lista de clientes
11
12 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as TCPServerSocket:
13     TCPServerSocket.bind((HOST, PORT))
14     while True:
15         TCPServerSocket.listen()
16         print("El servidor TCP está disponible y en espera de solicitudes")
17         conn, addr = TCPServerSocket.accept()#acepta la conexión
18         clientes.append((conn, addr))#agregar al cliente a la lista
19         for i, (cliente_conn, cliente_addr) in enumerate(clientes):#imprime los clientes
20             print(f"Cliente {i + 1}: {cliente_addr}")
21         opcion = input("Elija una opción:\n1. Desconectar cliente\n2. Continuar servidor")
22         if opcion=="1":
23             desconectar = input("Cliente a desconectar (0 cancelar): ")
24             if desconectar == "0":
25                 print("cancelado")
26             else:
27                 try:
28                     num_cliente = int(desconectar)-1
29                     if 0 <= num_cliente < len(clientes):
30                         desconexión(clientes[num_cliente][0])
31                         del clientes[num_cliente]#borrar al cliente
32                     else:
33                         print("Número de cliente no válido.")
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

## Desconectar el servidor.



```
Practical1 > Ejercicio > Servidor.py
1 host = "127.0.0.1" # Dirección IP de la máquina local (servidor)
2 PORT = 65432 # Puerto que usa el cliente (los puertos sin privilegios son > 1023)
3 buffer_size = 1024
4
5 def desconexion(cliente):
6     cliente.close()
7     print("cliente desconectado")
8     clientes = [] # lista de clientes
9
10 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as TCPServerSocket:
11     TCPServerSocket.bind((HOST, PORT))
12     while True:
13         TCPServerSocket.listen()
14         print("El servidor TCP está disponible y en espera de solicitudes")
15         conn, addr = TCPServerSocket.accept() # acepta la conexión
16         clientes.append((conn, addr)) # agrega al cliente a la lista
17         for i, (cliente_conn, cliente_addr) in enumerate(clientes): # imprime los clientes
18             print(f"Cliente {i + 1}: {cliente_addr}")
19         opcion = input("Elija una opción:\n1. Desconectar cliente\n2. Continuar servidor\n")
20         if opcion == "1":
21             desconectar = input("Cliente a desconectar (0 cancelar): ")
22             if desconectar == "0":
23                 print("cancelado")
24             else:
25                 try:
26                     num_cliente = int(desconectar) - 1
27                     if 0 <= num_cliente < len(clientes):
28                         desconexion(clientes[num_cliente][0])
29                         del clientes[num_cliente] # borra al cliente
30                     else:
31                         print("Número de cliente no válido.")
32                 except:
33                     print("Ingresa un número válido.")
34
35 PS C:\Users\luisa\OneDrive\Documents\luisa\ESCOM\6. QuintoSemestre\Redes2\Practicas> & C:/Users/luisa/AppData/Local/Programs/Python/Python111/python.exe -c "C:/Users/luisa/OneDrive/Documents/luisa/ESCOM/6. QuintoSemestre/Redes2/Practicas/Practical1/Ejercicio/Cliente2.py"
Conectado al servidor
PS C:\Users\luisa\OneDrive\Documents\luisa\ESCOM\6. QuintoSemestre\Redes2\Practicas>
```

```
Practical1 > Ejercicio > Cliente2.py
1 PORT = 65432 # Puerto del servidor
2 buffer_size = 1024
3
4 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as TCPClientSocket:
5     TCPClientSocket.connect((HOST, PORT))
6     print("Conectado al servidor")
7     TCPClientSocket.recv(buffer_size) # esperar desconexión
8
9 PS C:\Users\luisa\OneDrive\Documents\luisa\ESCOM\6. QuintoSemestre\Redes2\Practicas>
```