

Taller TCT1

Erwin Jonner Jaime Lamus, 1115720307

Kevin Lorenzo Madrid Neira, 1117132136

Luz Marina Santos Jaimes

Universidad de pamplona

Redes

Pamplona

2025

1. Verificar que Python este instalado en ambos equipos

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\jonne> python --version
Python 3.13.5
PS C:\Users\jonne>
```

2. Ip del servidor

```
Adaptador de LAN inalámbrica Wi-Fi:

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . . . : fe80::92ff:8cfe:e811:e0e7%6
Dirección IPv4. . . . . : 10.179.54.32
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 10.179.54.192
PS C:\Users\jonne>
```

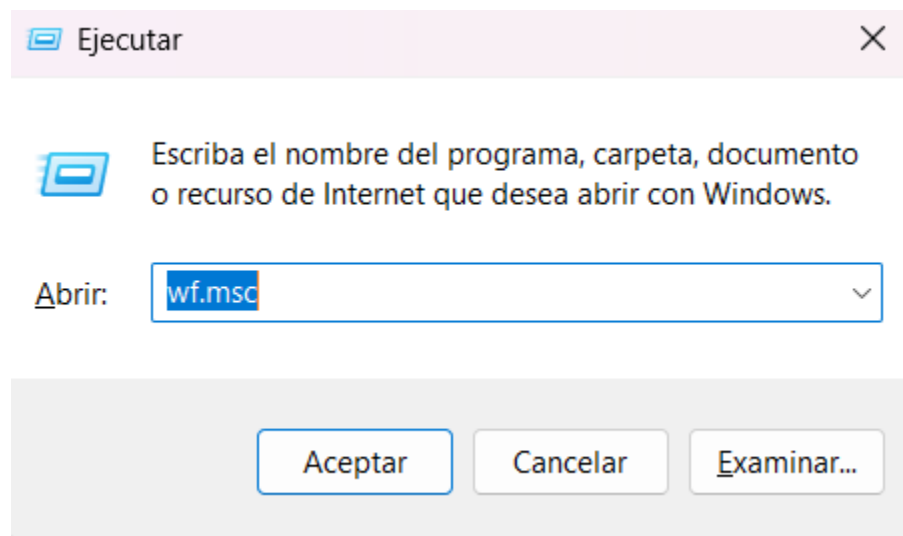
3. Ip del cliente

```
Adaptador de LAN inalámbrica Wi-Fi:

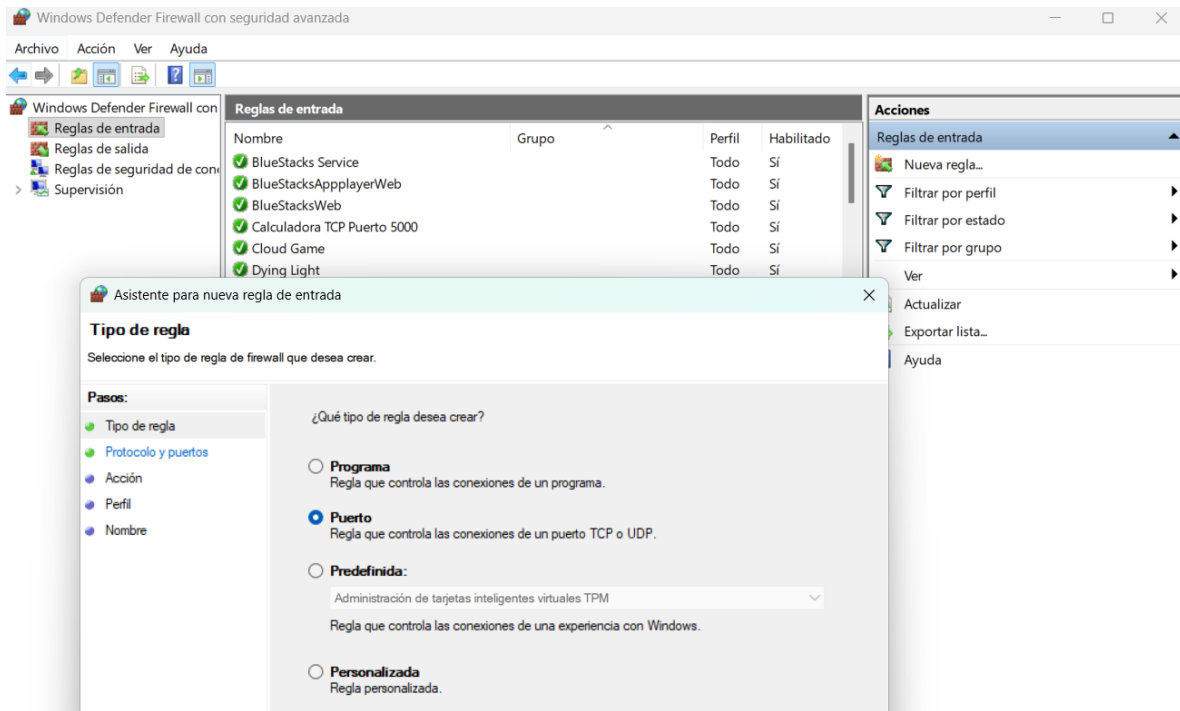
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . . . : fe80::c7fd:bbfd:904b:cb7b%18
Dirección IPv4. . . . . : 192.168.1.12
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

Habilitar el puerto 5000 para la practica

1. Presionamos la tecla Windows + R y escribimos
 - Wf.msc



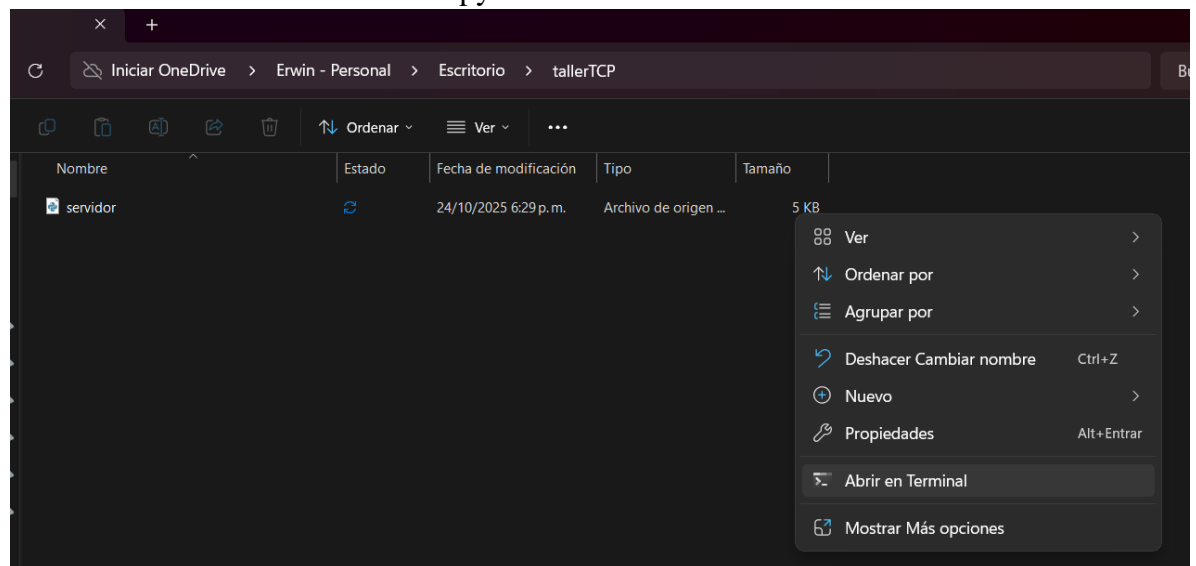
2. Creamos una nueva regla de entrada de tipo puerto TCP o UDP



3. Le damos en siguiente y le decimos que escuche por el puerto 5000 y presionamos siguiente.
4. Luego seleccionamos permitir la conexión.
5. Le damos siguiente hasta la ultima opción y le damos de nombre Calculadora TCP Puerto 5000

Parte del servidor

1. Crear una carpeta en escritorio llamada tallerTCP y abrimos visual studio code donde creamos el archivo servidor.py

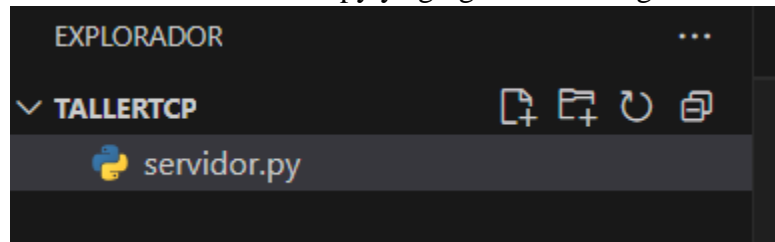


```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\jonne\OneDrive\Escritorio\tallerTCP> code .
```

2. El visual creamos el archivo servidor.py y agregamos el código



3. Repetimos los mismos pasos para el caso del cliente pero el archivo se llamara cliente.py

Para ejecutar el código parte del servidor

1. Abrimos una terminal en la ruta donde está el archivo servidor.py y ejecutamos el siguiente comando para iniciar el servidor
 - Python servidor.py

```
PS C:\Users\jonne\OneDrive\Escritorio\tallerTCP> python servidor.py
[SERVIDOR] Iniciado en 0.0.0.0:5000
[SERVIDOR] Esperando conexiones de clientes...
```

2. Cuando el cliente se conecta sale lo siguiente

```
PS C:\Users\jonne\OneDrive\Escritorio\tallerTCP> python servidor.py
[SERVIDOR] Iniciado en 0.0.0.0:5000
[SERVIDOR] Esperando conexiones de clientes...

[CONEXIÓN] Cliente conectado desde ('10.179.54.231', 54302)
[RECIBIDO] 15,*19
[RESULTADO] 285.0
[DESCONEXIÓN] Cliente ('10.179.54.231', 54302) desconectado
```

3. El servidor queda escuchando nuevamente para próximas solicitudes, para cerrarlo cerrar la terminal

Para ejecutar el código de parte del cliente

1. Abrimos una terminal en la ruta donde está el archivo cliente.py y ejecutamos el siguiente comando para iniciar el cliente
 - Python cliente.py
2. La calculadora primero pedirá el primer dato, luego el operador y por último el segundo dato.

```

PS C:\Users\User\Desktop\Taller redes> python cliente.py
=====
      CALCULADORA REMOTA TCP - CLIENTE
=====

Ingrese la IP del servidor: 10.179.54.32
Ingrese el primer número: 15
Ingrese el operador (+, -, *, /): *
Ingrese el segundo número: 19

[CLIENTE] Conectando al servidor 10.179.54.32:5000...
[CLIENTE] Conectado exitosamente
[CLIENTE] Enviando: 15 * 19

=====
RESULTADO: 15 * 19 = 285.0
=====

[CLIENTE] Conexión cerrada
PS C:\Users\User\Desktop\Taller redes>

```

3. Luego de cada conexión al servidor esta vuelve a cerrarse, se debe hacer todo el procedimiento de nuevo para una nueva solicitud

Datos capturados en wireshark

1. Para filtrar la información que nos importa podemos filtrar por la ip del cliente o por el puerto q habilitamos para escuchar el 5000

No.	Time	Source	Destination	Protocol	Length	Info
837	18.891812	10.179.54.231	10.179.54.32	TCP	66	64013 → 5000 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
841	19.527895	10.179.54.32	10.179.54.231	TCP	66	5000 → 64013 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
870	19.897149	10.179.54.32	10.179.54.231	TCP	66	[TCP Retransmission] 5000 → 64013 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
871	19.959983	10.179.54.231	10.179.54.32	TCP	54	64013 → 5000 [ACK] Seq=1 Ack=1 Win=65280 Len=0
872	19.960967	10.179.54.231	10.179.54.32	TCP	61	64013 → 5000 [PSH, ACK] Seq=1 Ack=1 Win=65280 Len=7
873	19.961395	10.179.54.32	10.179.54.231	TCP	59	5000 → 64013 [PSH, ACK] Seq=1 Ack=8 Win=65280 Len=5
874	19.962515	10.179.54.32	10.179.54.231	TCP	54	5000 → 64013 [FIN, ACK] Seq=0 Ack=8 Win=65280 Len=0
879	20.527272	10.179.54.231	10.179.54.32	TCP	66	[TCP Dup ACK 871#1] 64013 → 5000 [ACK] Seq=8 Ack=1 Win=65280 Len=0 SLE=0 SRE=1
880	20.529746	10.179.54.231	10.179.54.32	TCP	54	64013 → 5000 [ACK] Seq=8 Ack=7 Win=65280 Len=0
881	20.529746	10.179.54.231	10.179.54.32	TCP	54	64013 → 5000 [FIN, ACK] Seq=8 Ack=7 Win=65280 Len=0
882	20.529815	10.179.54.32	10.179.54.231	TCP	54	5000 → 64013 [ACK] Seq=7 Ack=9 Win=65280 Len=0