Aceptar una solicitud de conexión

El servidor se ejecuta en un bucle infinito y escucha las solicitudes de clientes para aceptar la conexión. Una vez que se encuentra una solicitud de cliente, el servidor acepta la solicitud utilizando el método `accept()`.
El método accept() devuelve una tupla de dirección (cliente, dirección) donde cliente representa un nuevo objeto socket que usamos para enviar y recibir mensajes, dirección es donde está vinculado.

Comunicación con el cliente (Send() y recv())

Después de aceptar la conexión, el servidor puede comunicarse con el cliente usamos el método C para enviar un mensaje al cliente. El método C se invoca sobre el cliente devuelto por el mensaje de aceptar y usamos el método recv para recibir los mensajes.
método recv cuando se invoca el cliente, acepta un número que representa el número máximo de bits que pueden leer la conexión. después de la ejecución devuelve los datos leídos de la conexión. Una vez completadas todas las operaciones, debemos cerrar la conexión. Para ello invocamos el método close() sobre el objeto client devuelto por el método accept().
Después de crear, ver todos los métodos para el servidor.

Ahora creamos un proceso del cliente que se comunica:

```python
import socket
# Direccion y puerto del servidor
HOST= '127.0.0.1'   # Direccion local (localhost)
PORT= 65432         # Puerto a escuchar
# Crear socket TCP/IP
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.blind((HOST, PORT))   # Vinular el socket a direccion y puerto
    s.listen()        # Escuchar conexiones entrantes
    print()
    print('Servidor esperando...')
    # Aceptar conexion
    conn, addr = s.accept()
    with conn:
        print()
        print('Servidor conectado a', addr)
        while True:
            data = conn.recv(1024)
            if not data:
                break
            print()
            print('El Servidor recibe', repr(data))
            conn.sendall('Hola desde el servidor')
```

Ahora creamos un proceso de cliente que se comunicará con el servidor

```python
import socket

# Dirección y puerto del servidor
HOST = '127.0.0.1'  # Dirección local (localhost)
PORT = 65432        # Puerto al que conectarse

# Crear un socket TCP/IP
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT)) # Conectarse al servidor
    mensaje = "Hola desde el cliente!"
    s.sendall(mensaje.encode()) # Enviar un mensaje al servidor
    data = s.recv(1024) # Recibir la respuesta del servidor

print()
print(f'Cliente recibe: {data.decode()}')
```

COMO CREAR UN CLIENTE EN PROGRAMACIÓN SOC...

Para crear un cliente primero necesitamos crear un socket con el método socket como lo hicimos al crear el servidor. Recuerden que los protocolos definidos para el socket del cliente deben ser los mismos que los del socket del servidor, de lo contrario el programa no funcionará correctamente.

Después de crear el socket necesitamos conectar el servidor usando connect()

Conectarse al servidor (Connect) - 1 - (-1

Sintaxis del método:

connect ((Host, Port))

Aquí el parametro Host denota la dirección del servidor. El parametro port indica el número de puerto en el que se creo el socket del servidor. Debe dar los mismos valores como entrada al Host y el parametro del puerto que proporciono el método bin al crear el servidor.

Comunicación con el servidor

Después de conectarse al servidor, puede comunicarse con el servidor utilizando los métodos send() y recv(). Finalmente procedería a cerrar la conexión desde el lado del cliente utilizando close().