

Gabriel Cecon Carlsen

## **Atividade 1 – Socket**

Criar uma aplicação na qual tenha o lado do cliente e posteriormente o lado do servidor. Sendo que o cliente inicie a comunicação com o servidor.

O servidor deve ficar esperando a conexão, de forma que responda automaticamente ao cliente no nível de rede e a nível de aplicação.

**Linguagem:** Python 3.8.5

**Biblioteca:** Socket

**Introdução:** O objetivo dessa aplicação é mostrar o funcionamento concreto do protocolo de transporte TCP/IP por meio de um Socket que habilita a comunicação em tempo real entre cliente/servidor. O Python implementa a interface de rede utilizando os fundamentos da API de Socket. Após fornecer o HOST e a PORTA, instanciamos o socket TCP/IPv4 em que o servidor aguardará por dados (até 1024 bytes) de apenas um cliente por vez, o processo Three-way Handshake acontece entre servidor/cliente e a aplicação/conexão é finalizada.

Abaixo disponibilizo o código comentado, exemplo de execução e registro linha a linha da conexão.

## Single Threading

### servidor.py

```
import socket

HOST = '127.0.0.1' # Localhost
PORT = 5050 # Porta

S = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # Padrão
'''
Objeto socket da biblioteca socket;
socket.AF_INET: representa a família do protocolo (IPv4);
socket.SOCK_STREAM: representa o tipo de protocolo de transporte (TCP);
'''

S.bind((HOST,PORT)) # Faz a ligação do socket com o endereço desejado
S.listen() # Habilita o servidor a aceitar conexões

print("Aguardando conexão de um cliente ...")

conn, ender = S.accept()
'''
Aceita a conexão;
Retornando socket para enviar e receber dados (conn);
Endereço ligado ao socket (ender);
'''

print("Conectado em", ender)

while True: # Aceitando novas conexões
    data = conn.recv(1024) # Recebe dados pelo socket (1024 bytes)
    if not data:
        print("Conexão fechada :)")
        conn.close()
        break
    conn.sendall(data) # Ecoando os dados de volta para o cliente
```

### cliente.py

```
import socket

HOST = '127.0.0.1' # Localhost
PORT = 5050 # Porta

C = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # IPv4 e TCP

C.connect((HOST,PORT)) # Conectando no HOST na porta PORT

C.sendall(str.encode('Olá Servidor, meu nome é Cliente e sou o único aqui.')) # Mensagem enviada para o Servidor

data = C.recv(1024) # Dados ecoados do Servidor

print("Mensagem ecoada:", data.decode())
```

## Execução (Servidor/Cliente)

servidor.py

```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.19041.1110]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>cd C:\Users\Cecon\Desktop\REDE COMP 1\ATIVIDADES\1 Socket\single-threading

C:\Users\Cecon\Desktop\REDE COMP 1\ATIVIDADES\1 Socket\single-threading>python servidor.py
Aguardando conexão de um cliente ...
Conectado em ('127.0.0.1', 49595)
Conexão fechada :)

C:\Users\Cecon\Desktop\REDE COMP 1\ATIVIDADES\1 Socket\single-threading>
```

cliente.py

```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.19041.1110]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\WINDOWS\system32>cd C:\Users\Cecon\Desktop\REDE COMP 1\ATIVIDADES\1 Socket\single-threading

C:\Users\Cecon\Desktop\REDE COMP 1\ATIVIDADES\1 Socket\single-threading>python cliente.py
Mensagem ecoada: Olá Servidor, meu nome é Cliente e sou o único aqui.
```

## Processo de registro de eventos (Log Wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
10.0000...	127.0.0.1	127.0.0.1	TCP	48	57128 → 5050	[SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1
20.0146...	127.0.0.1	127.0.0.1	TCP	48	5050 → 57128	[SYN, ACK] Seq=0 Ack=1 Win=65495 Len=0 MSS=65495 SACK_PERM=1
30.0146...	127.0.0.1	127.0.0.1	TCP	40	57128 → 5050	[ACK] Seq=1 Ack=1 Win=65495 Len=0
40.0146...	127.0.0.1	127.0.0.1	TCP	95	57128 → 5050	[PSH, ACK] Seq=1 Ack=1 Win=65495 Len=55
50.0148...	127.0.0.1	127.0.0.1	TCP	40	5050 → 57128	[ACK] Seq=1 Ack=56 Win=65440 Len=0
60.0148...	127.0.0.1	127.0.0.1	TCP	95	5050 → 57128	[PSH, ACK] Seq=1 Ack=56 Win=65440 Len=55
70.0149...	127.0.0.1	127.0.0.1	TCP	40	57128 → 5050	[ACK] Seq=56 Ack=56 Win=65440 Len=0
80.0149...	127.0.0.1	127.0.0.1	TCP	40	57128 → 5050	[FIN, ACK] Seq=56 Ack=56 Win=65440 Len=0
90.0149...	127.0.0.1	127.0.0.1	TCP	40	5050 → 57128	[ACK] Seq=56 Ack=57 Win=65440 Len=0
100.0149...	127.0.0.1	127.0.0.1	TCP	40	5050 → 57128	[FIN, ACK] Seq=56 Ack=57 Win=65440 Len=0
110.0149...	127.0.0.1	127.0.0.1	TCP	40	57128 → 5050	[ACK] Seq=57 Ack=57 Win=65440 Len=0

[1, 2, 3]: Three-way HandShake (SYN, SYN-ACK, ACK);

[4, 5, 6, 7]: PUSH, ACK (cliente), ACK (servidor), PUSH, ACK (servidor ecoando), ACK (cliente);

[8, 9, 10, 11]: Encerramento da conexão em quatro vias.