

Conectividade em Sistemas Ciberfísicos

ID	Requisitos
RF-1	O sistema deve imprimir na tela de todos os usuários conectados o nome de cada usuário que entrar no servidor imediatamente.
RF-2	O sistema deve permitir mensagens em tempo real para todos os usuários.
RF-3	O sistema deve permitir mensagens privadas entre usuários.
RF-4	O sistema deve imprimir na tela do servidor o IP e a porta do cliente que conectou
RF-5	O sistema deve permitir que o usuário encerre a sua sessão com o comando '/SAIR' no terminal.
RF-6	O sistema deve validar corretamente os dados de conexão.
RF-7	O sistema deve exibir para todos a saída do usuário do chat.
RF-8	O sistema deve mostrar que houve problemas ao encontrar erro no socket.
RF-9	O sistema utiliza TCP e trata a ocorrência para que haja broadcast
RF-10	O sistema aceita a conexão entre computadores diferentes

Os sockets foram utilizados para realizar a conexão entre as portas e relacioná-las ao seu IP correspondente. Para o broadcast, determinamos que a mensagem seria enviada para todos os clientes conectados, menos o próprio.

```
# Configurações do servidor
HOST = '192.168.194.25' # IP do servidor
PORTA = 50001 # Porta do servidor
special_keys = ["<USERNAME>"]
username = ""
# Criar o socket e conectar ao servidor
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((HOST, PORTA))
```

As threads foram utilizadas nos clientes e no servidor para que cada um conseguisse rodar simultaneamente, permitindo a conexão e comunicação entre eles em tempo real. Para cada cliente, uma thread é inicializada para receber as mensagens. Já no servidor, ao ser inicializado sua thread é criada.

```
# Iniciar a thread para receber mensagens
threading.Thread(target=receber_mensagens).start()
get_username()
# Loop para enviar mensagens
while True:
    mensagem = input(f"{username}> ")
    if "/SAIR" in mensagem:
        sock.sendall(mensagem.encode())
        break
    sock.sendall(mensagem.encode())
```

```
def iniciar_servidor():  
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
    sock.bind((HOST, PORTA))  
    sock.listen()  
    print("Servidor em execução, aguardando conexões...")  
  
    while True:  
        conn, addr = sock.accept()  
        users[conn] = ""  
        threading.Thread(target=gerenciar_cliente, args=(conn, addr), daemon=True).start()
```