**server.py**

|  |
| --- |
| import socket import threading import time  SERVER\_IP = "127.0.0.1" PORT = 5050 ADDR = (SERVER\_IP, PORT) FORMATO = 'utf-8'  server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) server.bind(ADDR)  conexoes = [] mensagens = []   def enviar\_mensagem\_individual(conexao):  print(f"[ENVIANDO] Enviando mensagens para {conexao['addr']}")  for i in range(conexao['last'], len(mensagens)):  mensagem\_de\_envio = "msg=" + mensagens[i]  conexao['conn'].send(mensagem\_de\_envio.encode())  conexao['last'] = i + 1  time.sleep(0.2)   def enviar\_mensagem\_todos():  for conexao in conexoes:  enviar\_mensagem\_individual(conexao)   def handle\_clientes(conn, addr):  print(f"[NOVA CONEXAO] Um novo usuario se conectou pelo endereço {addr}")  global conexoes  global mensagens  nome = False   while True:  msg = conn.recv(1024).decode(FORMATO)  if msg:  if msg.startswith("nome="):  mensagem\_separada = msg.split("=")  nome = mensagem\_separada[1]  conexoes.append({'conn': conn, 'addr': addr, 'nome': nome, 'last': 0})  enviar\_mensagem\_individual({'conn': conn, 'addr': addr, 'nome': nome, 'last': 0})  elif msg.startswith("msg="):  mensagem\_separada = msg.split("=")  mensagem = nome + "=" + mensagem\_separada[1]  mensagens.append(mensagem)  enviar\_mensagem\_todos()   def start():  print("[INICIANDO] Iniciando Socket")  server.listen()  while True:  conn, addr = server.accept()  thread = threading.Thread(target=handle\_clientes, args=(conn, addr))  thread.start()   if \_\_name\_\_ == '\_\_main\_\_':  start() |

Esse é o código do servidor do chat e funciona da seguinte maneira:

O servidor cria a sala de bate-papo e server como uma forma de “carteiro”, ele aceita as conexões dos clientes, recebe seus nomes e suas mensagens e re-transmite para todos os cliente, criando uma forma de chat de whatsapp, com as mensagens chegando em ordem do primeiro cliente ao último.

Toda vez que um novo cliente envia uma mensagem ao servidor, o mesmo salva essa mensagem juntos com as mensagens antigas e também a envia para os outros clientes já conectados, dessa forma ele guarda um registro das mensagens antigas no chat para enviá-las para todos os novos clientes através da função “enviar\_mensagem\_individual()”, dessa forma eles conseguem ler as mensagens antigas.

Toda vez que um cliente manda uma mensagem, o servidor a lê e a envia para todo mundo usando a função “enviar\_mensagem\_todos()”.

**client.py**

|  |
| --- |
| import socket import threading import time  PORT = 5050 FORMATO = 'utf-8' SERVER = "127.0.0.1" ADDR = (SERVER, PORT)  client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) client.connect(ADDR)   def handle\_mensagens():  while True:  msg = client.recv(1024).decode()  mensagem\_separada = msg.split("=")  print(mensagem\_separada[1] + ": " + mensagem\_separada[2])   def enviar(mensagem):  client.send(mensagem.encode(FORMATO))   def enviar\_mensagem():  mensagem = input()  enviar("msg=" + mensagem)   def enviar\_nome():  nome = input('Digite seu nome: ')  enviar("nome=" + nome)   def iniciar\_envio():  enviar\_nome()  enviar\_mensagem()   def iniciar():  thread1 = threading.Thread(target=handle\_mensagens)  thread2 = threading.Thread(target=iniciar\_envio)  thread1.start()  thread2.start()   if \_\_name\_\_ == '\_\_main\_\_':  iniciar() |

O código cliente representa a pessoa que envia a “carta” para o “carteiro” que é o servidor, o código tem duas funções importantes, a “handle\_mensagens()” que é responsável por mostrar na tela do novo cliente as mensagens antigas do chat, e a “iniciar\_envio()”, que é responsável por enviar o nome e a mensagem desse novo cliente para o servidor.

O problema desse código é que o cliente só pode enviar uma mensagem, depois o código fica bloqueado, logo, para enviar mais de uma mensagem precisamos iniciar novos clientes para cada mensagem.