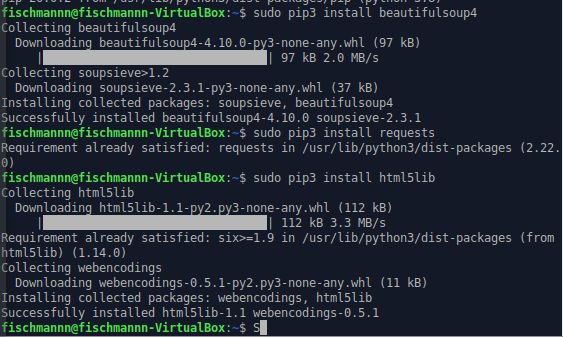
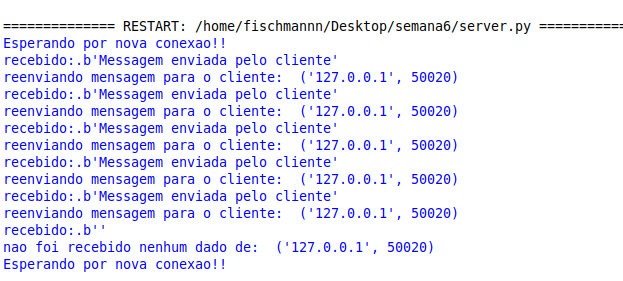
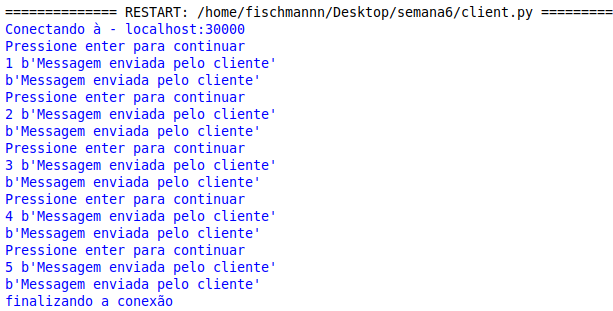
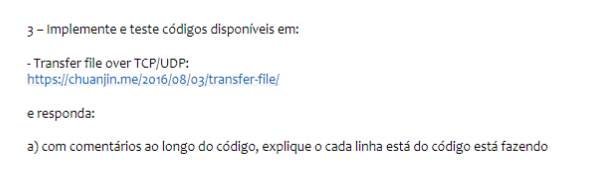


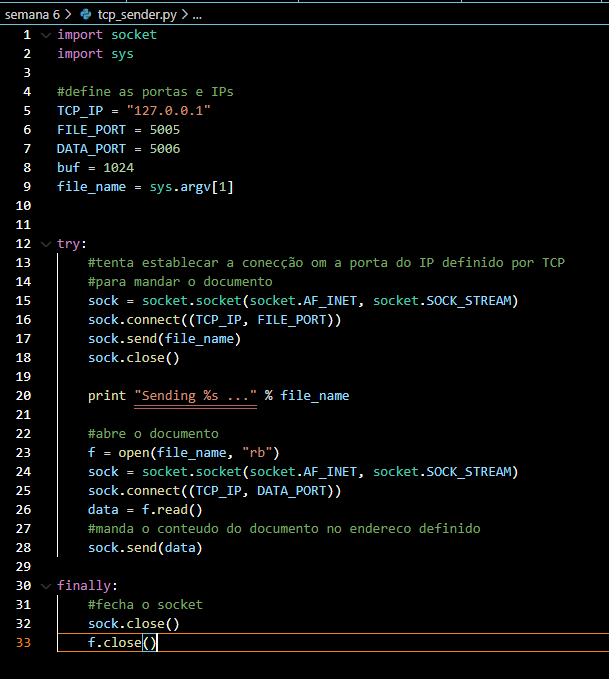
**Aula Pratica :**

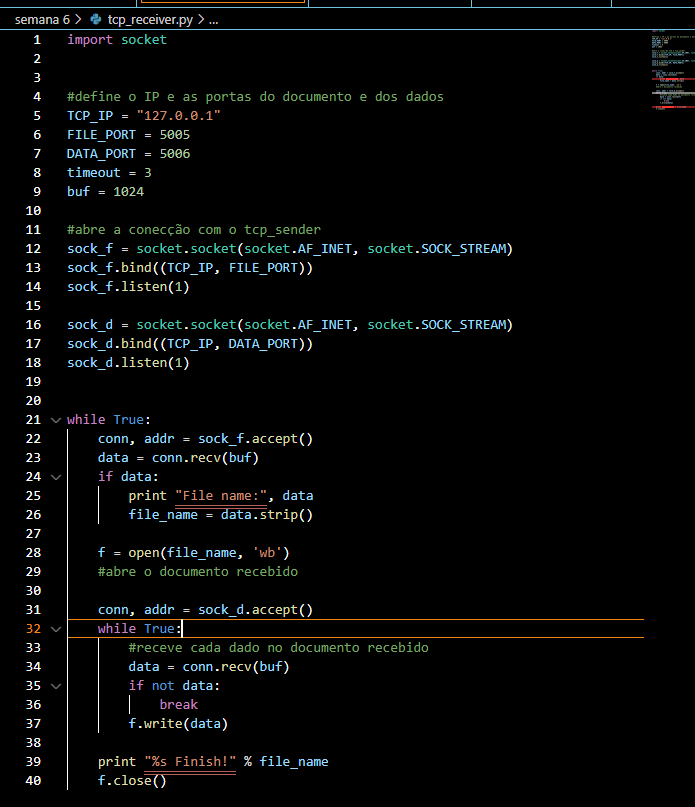


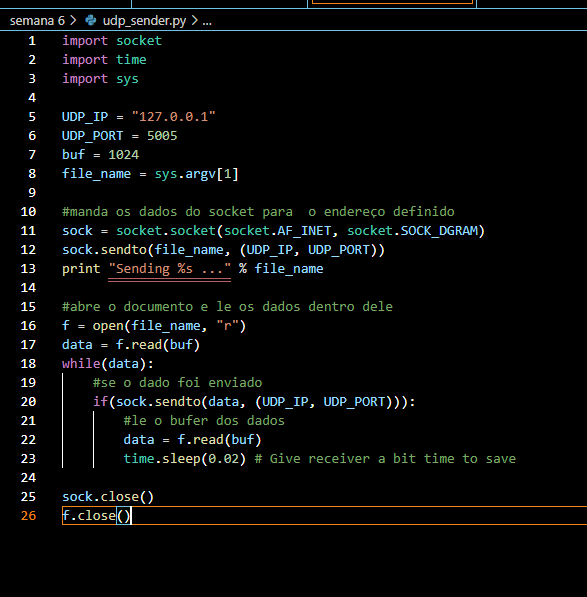


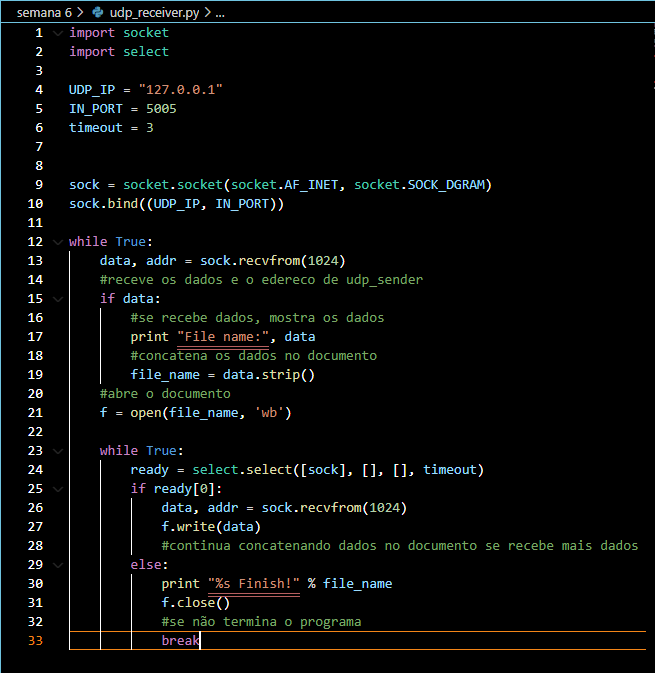


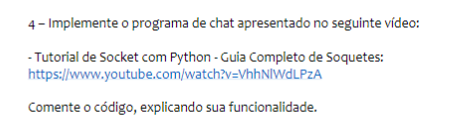












Server.py

import socket

import threading

import time

#define o ip do servidor coo endereco IP do HOST

SERVER\_IP = socket.gethostbyname(socket.gethostname())

PORT = 5050

ADDR = (SERVER\_IP, PORT)

#define o formato de comunicação como utf-8

FORMATO = 'utf-8'

server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server.bind(ADDR)

conexoes = []

mensagens = []

def enviar\_mensagem\_individual(conexao):

    #Essa função manda a mensagem i da lista mensagens para uma 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():

    #Essa função envia todas as mensagens da lista mensagens para

    #as conexoes da lista conexao

    global conexoes

    for conexao in conexoes:

        enviar\_mensagem\_individual(conexao)

"""

1 vez que o cliente entrar, vai mandar o nome:

nome=.....

E as mensagens vem:

msg=

"""

def handle\_clientes(conn, addr):

    #Essa função recebe o endereço, a conexão e o nome de um cliente

    #e envia uma mensagem individual para ele

    #ou envia todas as mensagens para os clientes

    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]

                mapa\_da\_conexao = {

                    "conn": conn,

                    "addr": addr,

                    "nome": nome,

                    "last": 0

                }

                conexoes.append(mapa\_da\_conexao)

                enviar\_mensagem\_individual(mapa\_da\_conexao)

            elif(msg.startswith("msg=")):

                mensagem\_separada = msg.split("=")

                mensagem = nome + "=" + mensagem\_separada[1]

                mensagens.append(mensagem)

                enviar\_mensagem\_todos()

def start():

    #abre o server e establece a conexão como thread

    print("[INICIANDO] Iniciando Socket")

    server.listen()

    while(True):

        conn, addr = server.accept()

        thread = threading.Thread(target=handle\_clientes, args=(conn, addr))

        thread.start()

start()

Client.py

import socket

import threading

import time

PORT = 5050

FORMATO = 'utf-8'

SERVER = "192.168.0.109"

ADDR = (SERVER, PORT)

#abre o socket e se conecta no endereço definido

client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client.connect(ADDR)

def handle\_mensagens():

    #esssa funcao trata a mensagem enviada separando ela pelo '='

    while(True):

        msg = client.recv(1024).decode()

        mensagem\_splitada = msg.split("=")

        print(mensagem\_splitada[1] + ": " + mensagem\_splitada[2])

def enviar(mensagem):

    #enssa função envia a mensagem no formato utf-8

    client.send(mensagem.encode(FORMATO))

def enviar\_mensagem():

    #essa função pede ao utilizador de escrever a mensagem a ser enviada

    mensagem = input()

    enviar("msg=" + mensagem)

def enviar\_nome():

    #essa função envia o nome digitado pelo utilizador

    nome = input('Digite seu nome: ')

    enviar("nome=" + nome)

def iniciar\_envio():

    #o utilizador digita o nome e a mensagem e são enviados

    enviar\_nome()

    enviar\_mensagem()

def iniciar():

    #O thread 1 mostra as mensagens enviadas

    thread1 = threading.Thread(target=handle\_mensagens)

    #O thread 2 pede o nome e a mensagem do utilizados para mandar os dois

    thread2 = threading.Thread(target=iniciar\_envio)

    thread1.start()

    thread2.start()

iniciar()