Atividade de Redes

Por - Gabriel Pereira Ho e Guilherme Mendonça

Código do servidor Python funcionando

```
# Importa o módulo socket
from socket import *
import sys  # Necessário para encerrar o programa

# Cria o socket TCP (orientado à conexão)
serverSocket = socket(AF_INET, SOCK_STREAM)

# Prepara o socket do servidor
serverPort = 6789  # Escolha a porta que deseja usar
serverSocket.bind(('0.0.0.0', serverPort))  # Bind do socket ao
endereço e porta
serverSocket.listen(1)  # Escuta apenas uma conexão por vez

print(f"Servidor web rodando na porta {serverPort}...")
print('Prontinho')
while True:
    # Estabelece a conexão

    connectionSocket, addr = serverSocket.accept()  # Aceita conexão do
cliente
    try:
        # Recebe a mensagem do cliente (requisição HTTP)
        message = connectionSocket.recv(1024).decode()  # Recebe até
1024 bytes
        filename = message.split()[1]  # Extrai o nome do arquivo
requisitado

        # Abre o arquivo requisitado
        f = open(filename[1:])  # Remove a barra inicial "/"
        outputdata = f.read()
        f.close()

        # Envia a linha de status do cabeçalho HTTP
        connectionSocket.send("HTTP/1.1 200 OK\r\n\r\n".encode())
        print(f"200 OK")
        # Envia o conteúdo do arquivo ao cliente
```

```
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i].encode())
        connectionSocket.send("\r\n".encode())

        # Fecha a conexão com o cliente
        connectionSocket.close()

    except IOError:
        # Envia mensagem de erro 404 se o arquivo não for encontrado
        connectionSocket.send("HTTP/1.1 404 Not
Found\r\n\r\n".encode())
        connectionSocket.send("<html><body><h1>404 Not
Found</h1></body></html>\r\n".encode())
        print(f"404 Not Found({filename})\n")
        # Fecha o socket do cliente
        connectionSocket.close()

# Fecha o socket do servidor (nunca será alcançado neste loop infinito)
serverSocket.close()
sys.exit()  # Encerra o programa
```

Código HTML

```
<html>
    <body>
        <h1>Hello, World!</h1>
    </body>
</html>
```

http://192.168.0.239:6789/HelloWorld.html

Navegador Web

Hello, World!

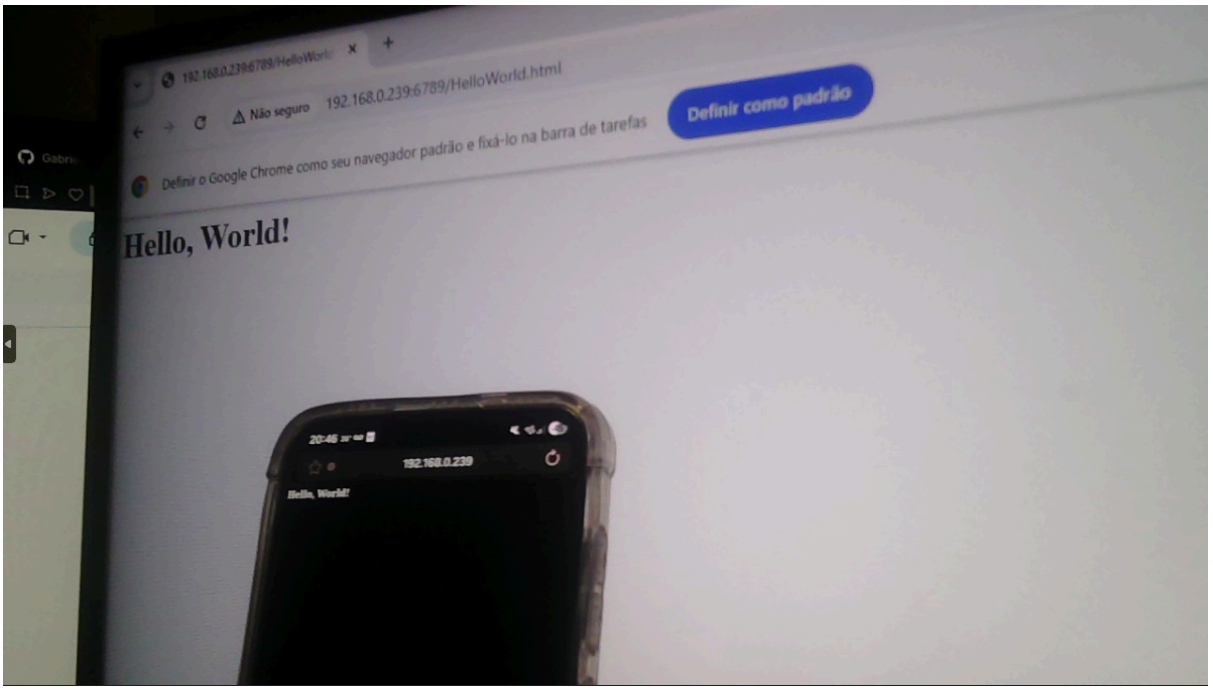
404

404 Not Found

Mensagens do terminal

```
PS C:\Raulzadas>  c:; cd 'c:\Raulzadas'; & 'c:\U
2-x64\bundled\libs\debugpy\launcher' '54007' '--
Servidor web rodando na porta 6789...
Prontinho
200 OK
404 Not Found(/lili.html)
```

Funcionando em outro dispositivo



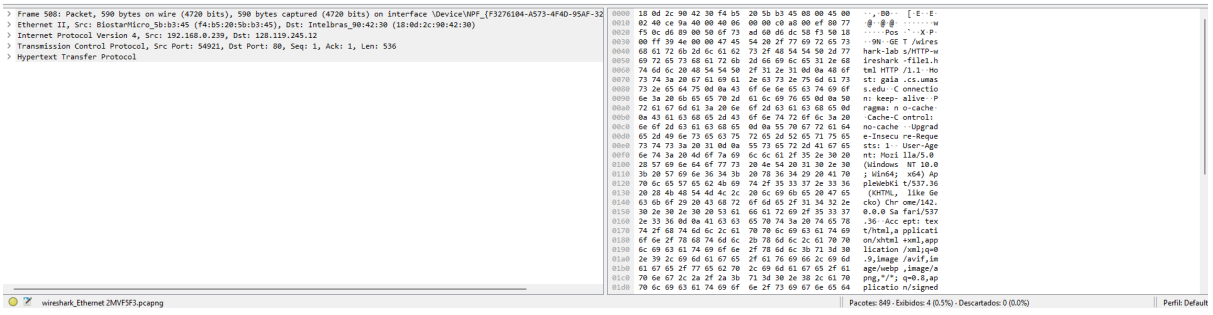
Servidor Funcionando

2) Wireshark e seus processos

2.1)

Link HTTP

Wireshark

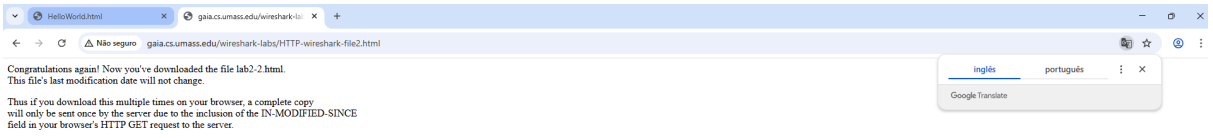


Respostas da Atividade

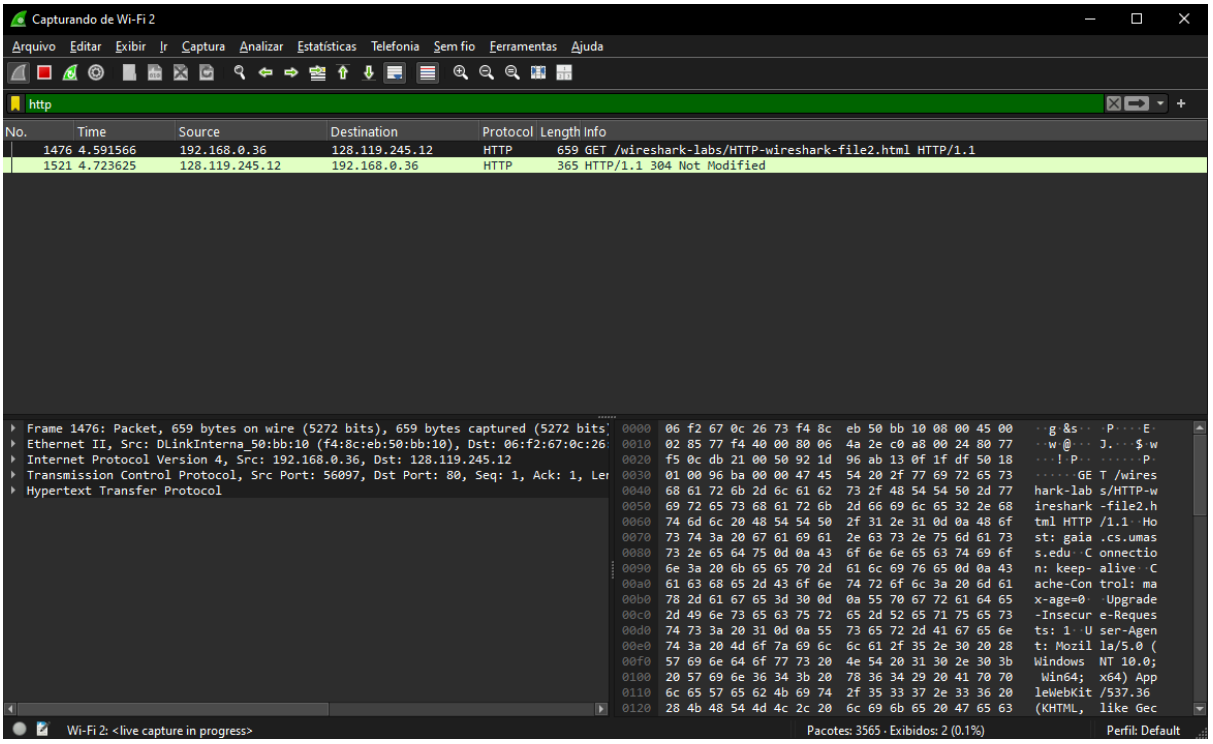
a)1.1,1.1
b) Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7\r\n\r\n
c)192.168.0.239 128.119.245.12

d)200 OK
e)Last-Modified: Tue, 28 Oct 2025 05:59:01 GMT\r\n
f)Frame 1508: Packet, 547 bytes on wire (4376 bits), 547 bytes captured (4376 bits) on interface \Device\NPF_{F3276104-A573-4F4D-95AF-326820F35AA3}, id 0

Navegador



Wireshark-304



Respostas

g)Não, aparece apenas nas próximas requisições

h)Sim, o HTML está dentro do próprio pacote HTTP caso inspecionado

i)If-Modified-Since: Tue, 28 Oct 2025 05:59:01 GMT\r\n

j)304 Not Modified, foi comparado com a primeira instância e foi relatado que não houve mudança, assim comprovando o uso do cache