

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



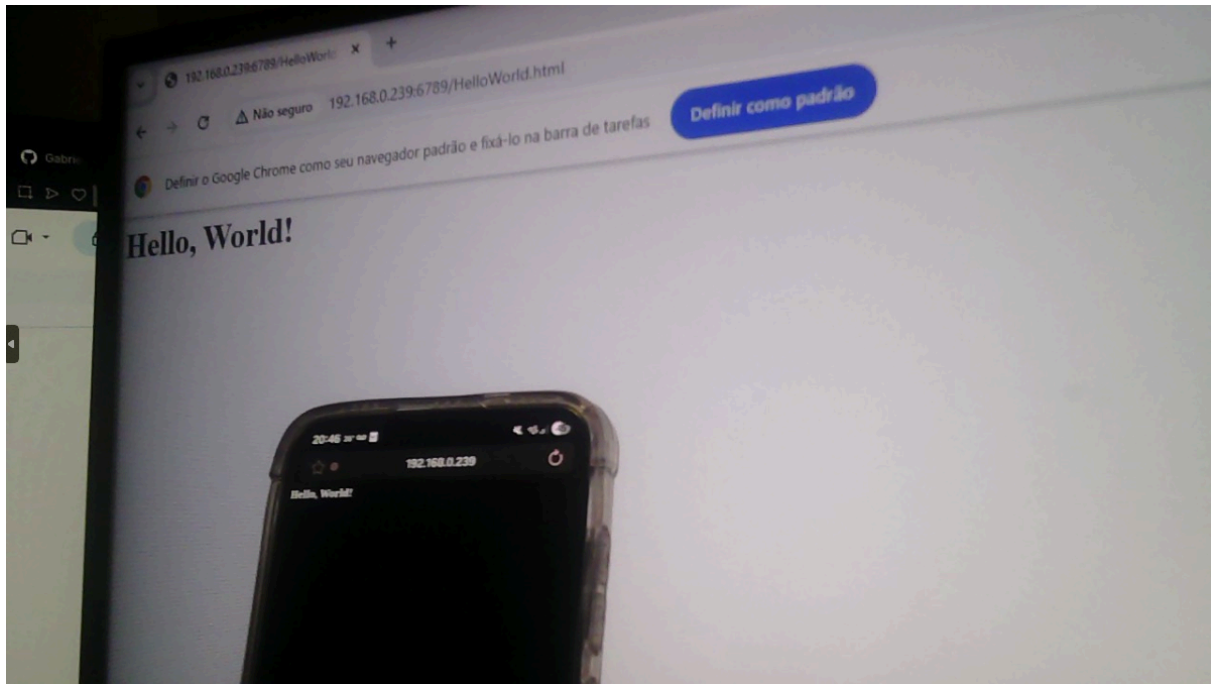
404



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

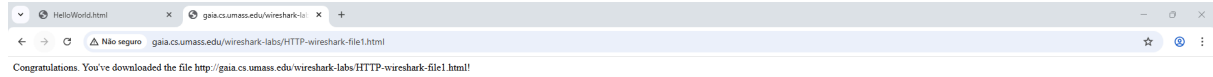


Servidor Funcionando

2) Wireshark e seus processos

2.1)

Link HTTP



Wireshark

http					
No.	Time	Source	Destination	Protocol	Length Info
508	6.81262	192.168.0.239	128.119.245.12	HTTP	508 GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
513	6.952829	128.119.245.12	192.168.0.239	HTTP	543 HTTP/1.1 200 OK (text/html)
514	6.974481	192.168.0.239	128.119.245.12	HTTP	536 GET /favicon.ico HTTP/1.1
521	7.110211	128.119.245.12	192.168.0.239	HTTP	636 HTTP/1.1 301 Moved Permanently (text/html)

Frame 508: Packet, 508 bytes on wire (4720 bits), 508 bytes captured (4720 bits) on Interface \Device\NPF... (f3276104-a573-4f40-95af-32...)	
Ethernet II, Src: BiostarMicro_Sb:b3:b3:45 (f4:b5:20:5b:b3:45), Dst: Intelbras_90:42:30 (18:0d:2c:90:42:30)	
Internet Protocol Version 4, Src: 192.168.0.239, Dst: 128.119.245.12	
Transmission Control Protocol, Src Port: 54921, Dst Port: 80, Seq: 1, Ack: 1, Len: 536	
Hypertext Transfer Protocol	
0000	18 0d 2c 90 42 30 f4 b5 20 5b b3 45 00 00 45 00 [E-E-]
0010	02 40 ce 9a 40 00 40 06 00 00 c0 a0 00 ef 00 77 ... @.@
0020	f5 0c 06 00 00 50 6f 73 ad 68 06 4c 50 f3 50 18Pos
0030	00 ff 39 4e 00 00 47 45 54 20 2f 77 69 72 65 73GET /wires
0040	68 61 72 69 2d 6c 61 62 73 2f 40 54 5a 50 2d 77hark-lab s/HTTP-w
0050	69 72 65 73 68 61 72 68 2d 68 69 6c 65 31 2e 68reshark _file1.h
0060	74 6d 6c 20 40 54 54 50 2f 31 2e 31 00 0a 48 6ftel HTTP /1.1 Mo
0070	73 74 3a 20 67 61 69 61 2e 63 73 2e 75 6d 61 73st: gaia.cs.umass
0080	73 2e 65 64 75 0d 0a 43 6f 6e 6e 65 63 74 69 6fs.edu: Connectio
0090	6e 3a 20 68 65 65 70 2d 61 6c 69 76 65 0d 0a 50n: keep-alive P
00a0	72 61 67 6d 61 3a 20 6a 6f 2d 63 61 63 60 65 0drname: n o-cache
00b0	0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20Cache-Control:
00c0	6e 6f 2d 63 61 63 68 65 0d 0a 55 78 67 72 61 64no-cache Upgrad
00d0	65 2d 49 6e 73 65 63 75 72 65 2d 52 65 71 75 65e:Insecu re-Reque
00e0	73 74 73 2a 20 31 0d 0a 55 73 65 72 2d 42 67 65sts: 1 - User-Age
00f0	6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20nt: NotI lla/5.0
0100	28 57 69 6e 64 6f 77 73 20 4e 54 20 31 30 2e 30(Windows NT 10.0
0110	30 20 57 69 6e 36 34 30 20 78 36 34 20 20 41 70; kind=; m=) Ap
0120	70 6c 65 57 65 62 40 69 74 2f 35 33 37 2e 33 36pleieki t/557.36
0130	20 28 40 40 54 4d 4c 2c 20 6c 69 6b 65 20 47 65(HTTP), like Ge
0140	63 6b 6f 29 20 43 68 72 6f 6d 65 2f 31 34 32 2echo) Chr ome/142.
0150	30 2e 36 2e 20 20 53 61 06 61 72 69 2f 35 33 370.0.0 Sa rari/557
0160	2e 33 36 0d 0a 41 63 63 65 70 74 3a 20 74 65 78;50 acc rpt: tex
0170	74 2f 68 74 6d 6c 2c 61 70 78 6c 69 63 61 74 69t/html,a policati
0180	6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 78on/html xml:app
0190	6c 69 63 61 74 69 6f 6e 2f 78 6d 6c 30 71 3d 30lication /xml;q=0
01a0	2e 39 2c 69 6d 61 67 65 2f 61 76 69 66 2c 69 6d;image /svg;im
01b0	61 67 65 2f 77 65 62 70 2c 69 6d 61 67 65 2f 61agr/ebp image/a
01c0	70 6e 67 2c 2a 2f 2a 39 71 3d 30 2e 30 2c 61 70png;*/; q=0.8,ap
01d0	70 6c 69 63 61 74 69 6f 6e 2f 73 69 6f 6e 65 64plicitio n/signed

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

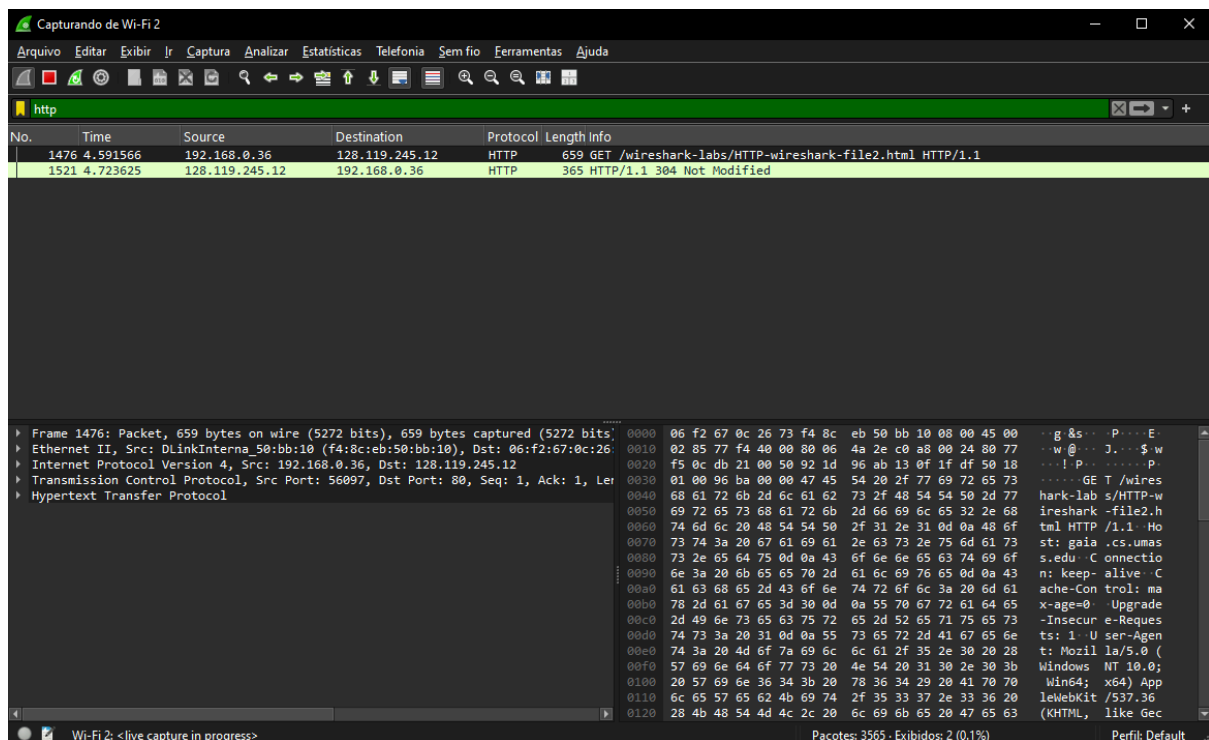
f) Frame 1508: Packet, 547 bytes on wire (4376 bits), 547 bytes captured (4376 bits) on interface

\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