Atividade sobre HTTP

Redes de Computadores

Sabrina Araújo Cardoso - 118210114

Lab. 1: Web Server Lab

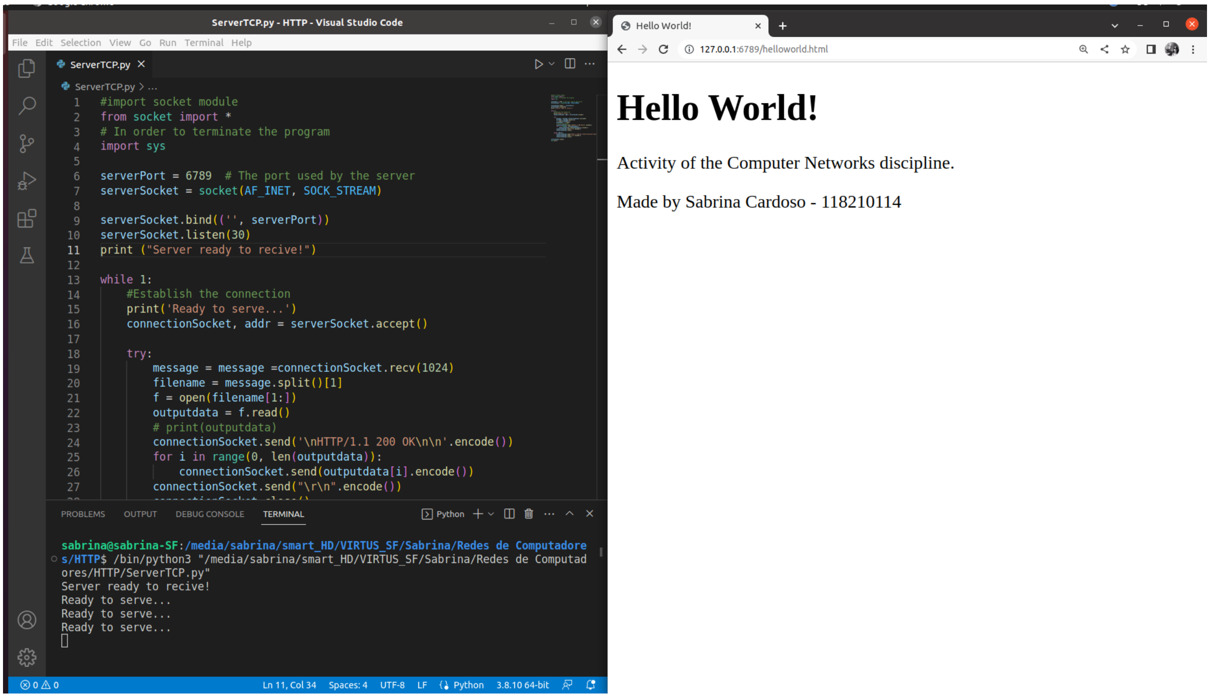
- Código para a criação do servidor TCP, chamado ServerTCP.py:

```
#import socket module
from socket import *
# In order to terminate the program
import sys

serverPort = 6789  # The port used by the server
serverSocket = socket(AF_INET, SOCK_STREAM)

serverSocket.bind(('', serverPort))
serverSocket.listen(30)
print ("Server ready to recive!")

while 1:
    #Establish the connection
    print('Ready to serve...')
    connectionSocket, addr = serverSocket.accept()

    try:
        message = message =connectionSocket.recv(1024)
        filename = message.split()[1]
        f = open(filename[1:])
        outputdata = f.read()
        print(outputdata)
        connectionSocket.send('\nHTTP/1.1 200 OK\n\n'.encode())
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i].encode())
        connectionSocket.send("\r\n".encode())
        connectionSocket.close()

    except IOError:
        connectionSocket.send("HTTP/1.1 404 Not Found\r\nContent-Type: text/plain\r\n\r\n404 Not Found".encode())
        connectionSocket.send("\r\n".encode())
        connectionSocket.close()

serverSocket.close()
sys.exit()
```

- O código html, chamado helloworld.html, que aparecerá no browser ao utilizar o protocolo HTTP:

```
<!DOCTYPE html>
<html>
<head>
  <title>Hello World!</title>
</head>
<body>
  <h1>Hello World!</h1>
  <p>Activity of the Computer Networks discipline.</p>
  <p>Made by Sabrina Cardoso - 118210114</p>
</body>
</html>
```

- Resultado:



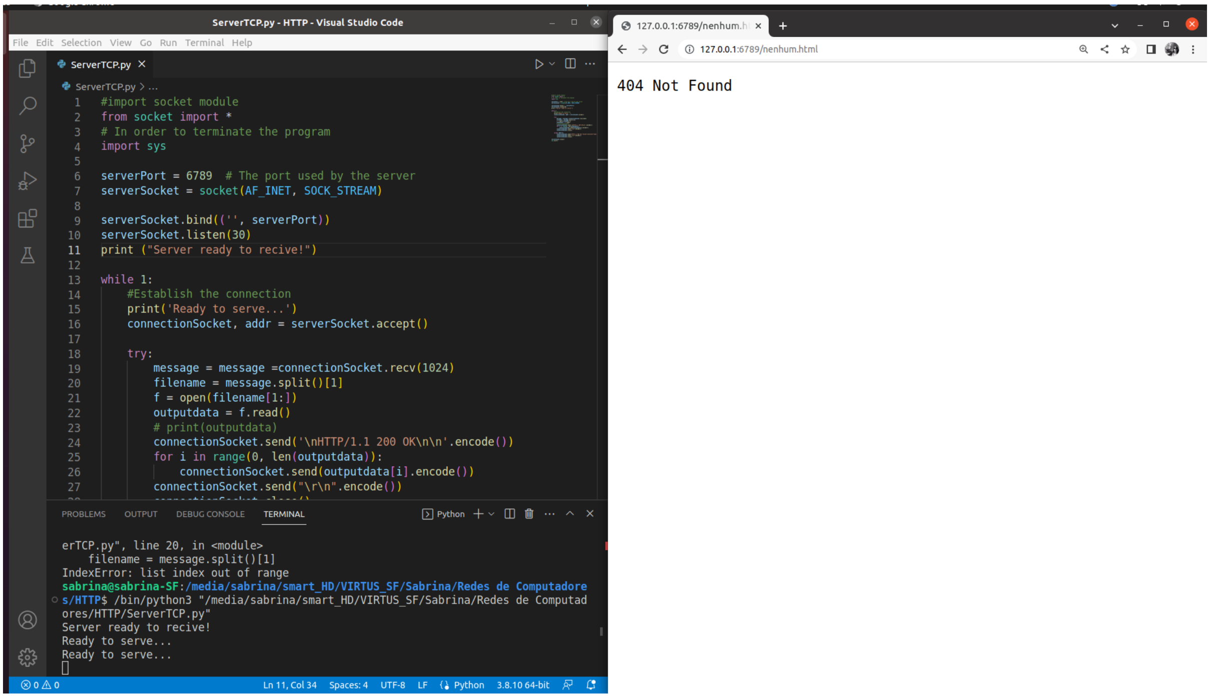
```python
#import socket module
from socket import *
# In order to terminate the program
import sys

serverPort = 6789  # The port used by the server
serverSocket = socket(AF_INET, SOCK_STREAM)

serverSocket.bind(('', serverPort))
serverSocket.listen(30)
print ("Server ready to recive!")

while 1:
    #Establish the connection
    print('Ready to serve...')
    connectionSocket, addr = serverSocket.accept()

    try:
        message = message =connectionSocket.recv(1024)
        filename = message.split()[1]
        f = open(filename[1:])
        outputdata = f.read()
        # print(outputdata)
        connectionSocket.send('\nHTTP/1.1 200 OK\n\n'.encode())
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i].encode())
        connectionSocket.send("\r\n".encode())
```

```
sabrina@sabrina-SF:/media/sabrina/smart_HD/VIRTUS_SF/Sabrina/Redes de Computadore
s/HTTP$ /bin/python3 "/media/sabrina/smart_HD/VIRTUS_SF/Sabrina/Redes de Computad
ores/HTTP/ServerTCP.py"
Server ready to recive!
Ready to serve...
Ready to serve...
Ready to serve...
```

Hello World!

Activity of the Computer Networks discipline.

Made by Sabrina Cardoso - 118210114



ServerTCP.py - HTTP - Visual Studio Code
File Edit Selection View Go Run Terminal Help
ServerTCP.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Ln 11, Col 34 Spaces: 4 UTF-8 LF Python 3.8.10 64-bit
127.0.0.1:6789/nenhum.html

```python
#import socket module
from socket import *
# In order to terminate the program
import sys

serverPort = 6789  # The port used by the server
serverSocket = socket(AF_INET, SOCK_STREAM)

serverSocket.bind(('', serverPort))
serverSocket.listen(30)
print ("Server ready to recive!")

while 1:
    #Establish the connection
    print('Ready to serve...')
    connectionSocket, addr = serverSocket.accept()

    try:
        message = message =connectionSocket.recv(1024)
        filename = message.split()[1]
        f = open(filename[1:])
        outputdata = f.read()
        # print(outputdata)
        connectionSocket.send('\nHTTP/1.1 200 OK\n\n'.encode())
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i].encode())
        connectionSocket.send("\r\n".encode())
```

```
erTCP.py", line 20, in <module>
    filename = message.split()[1]
IndexError: list index out of range
sabrina@sabrina-SF:/media/sabrina/smart_HD/VIRTUS_SF/Sabrina/Redes de Computadore
s/HTTP$ /bin/python3 "/media/sabrina/smart_HD/VIRTUS_SF/Sabrina/Redes de Computad
ores/HTTP/ServerTCP.py"
Server ready to recive!
Ready to serve...
Ready to serve...
```

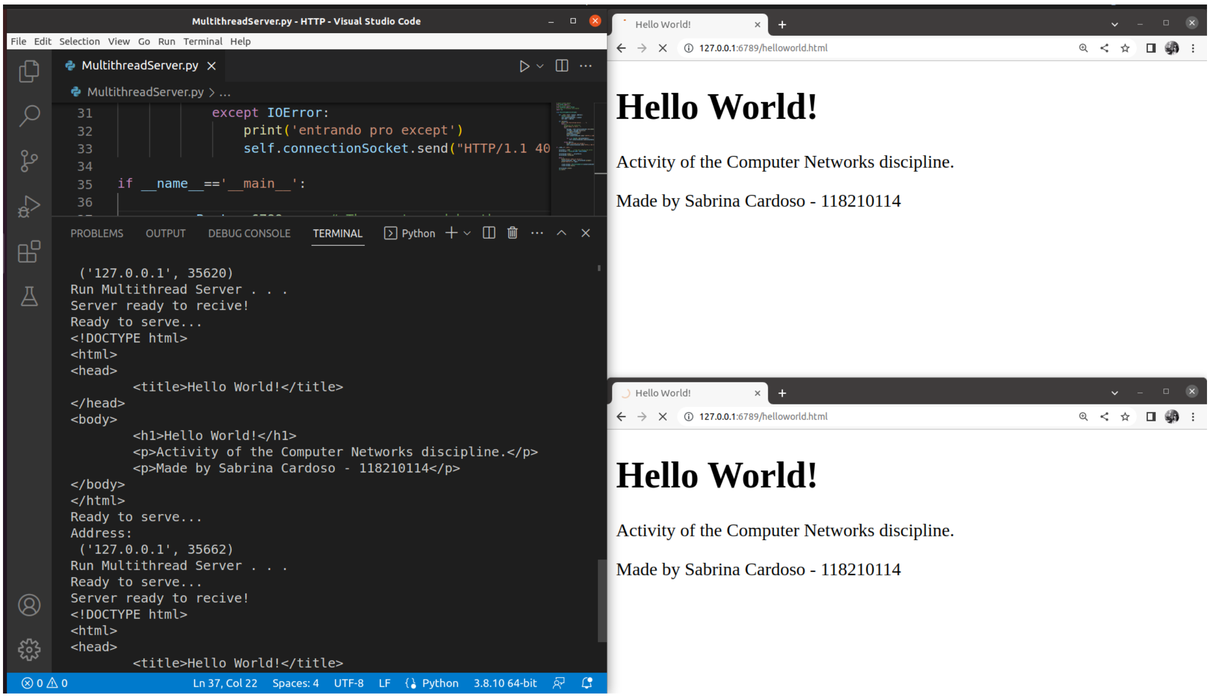
```
404 Not Found
```

Questão 1

- Código do MultiThread:

```python
# Import socket module
from socket import *
# Import thread module
from threading import Thread
# In order to terminate the program
import sys

class MultiThreadServer(Thread):

    def __init__(self, connect, address):
        Thread.__init__(self)
        self.connectionSocket = connect
        self.addr = address

    def run(self):
        print ("Run Multithread Server . . . ")
        while 1:
            #Establish the connection
            print('Ready to serve...')
            try:
                message = self.connectionSocket.recv(1024)
                filename = message.split()[1]
                f = open(filename[1:])
                outputdata = f.read()
                print(outputdata)
                self.connectionSocket.send('\nHTTP/1.1 200 OK\n\n'.encode())

                for i in range(0, len(outputdata)):
                    self.connectionSocket.send(outputdata[i].encode())

            except IOError:
                print('entrando pro except')
                self.connectionSocket.send("HTTP/1.1 404 Not Found\r\nContent-Type: te
xt/plain\r\n\r\n404 Not Found".encode())

if __name__=='__main__':

    serverPort = 6789 # The port used by the server
    serverSocket = socket(AF_INET, SOCK_STREAM)

    serverSocket.bind(('', serverPort))
    serverSocket.listen(30)

    while 1:
        print("Server ready to recive!")
        connectionSocket, addr = serverSocket.accept()
        print("Address:\n", addr)

        client_thread = MultiThreadServer(connectionSocket, addr)
        client_thread.start()

    serverSocket.close()
    sys.exit()
```

- Resultado


MultithreadServer.py - HTTP - Visual Studio Code
File Edit Selection View Go Run Terminal Help
MultithreadServer.py
MultithreadServer.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Python
Ln 37, Col 22
Spaces: 4
UTF-8
LF
Python
3.8.10 64-bit
Hello World!
127.0.0.1:6789/helloworld.html
Hello World!
127.0.0.1:6789/helloworld.html

```
            except IOError:
                print('entrando pro except')
                self.connectionSocket.send("HTTP/1.1 40

if __name__=='__main__':
```

```
 ('127.0.0.1', 35620)
Run Multithread Server . . .
Server ready to recive!
Ready to serve...
<!DOCTYPE html>
<html>
<head>
        <title>Hello World!</title>
</head>
<body>
        <h1>Hello World!</h1>
        <p>Activity of the Computer Networks discipline.</p>
        <p>Made by Sabrina Cardoso - 118210114</p>
</body>
</html>
Ready to serve...
Address:
 ('127.0.0.1', 35662)
Run Multithread Server . . .
Ready to serve...
Server ready to recive!
<!DOCTYPE html>
<html>
<head>
        <title>Hello World!</title>
```

Hello World!

Activity of the Computer Networks discipline.

Made by Sabrina Cardoso - 118210114

Hello World!

Activity of the Computer Networks discipline.

Made by Sabrina Cardoso - 118210114

Questão 2

- Código do cliente:

```
import socket
import sys

SERVER_HOST = sys.argv[1]
SERVER_PORT = int(sys.argv[2])
FILE_PATH = sys.argv[3]

# create a TCP socket and connect to the server
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((SERVER_HOST, SERVER_PORT))

# send HTTP GET request to the server
http_request = (f"GET /{FILE_PATH} HTTP/1.1\r\nHost: {SERVER_HOST}\r\n\r\n")
client_socket.sendall(http_request.encode())

# receive and display the server response
server_response = client_socket.recv(1024).decode()
print(server_response)

# close the socket
client_socket.close()
```