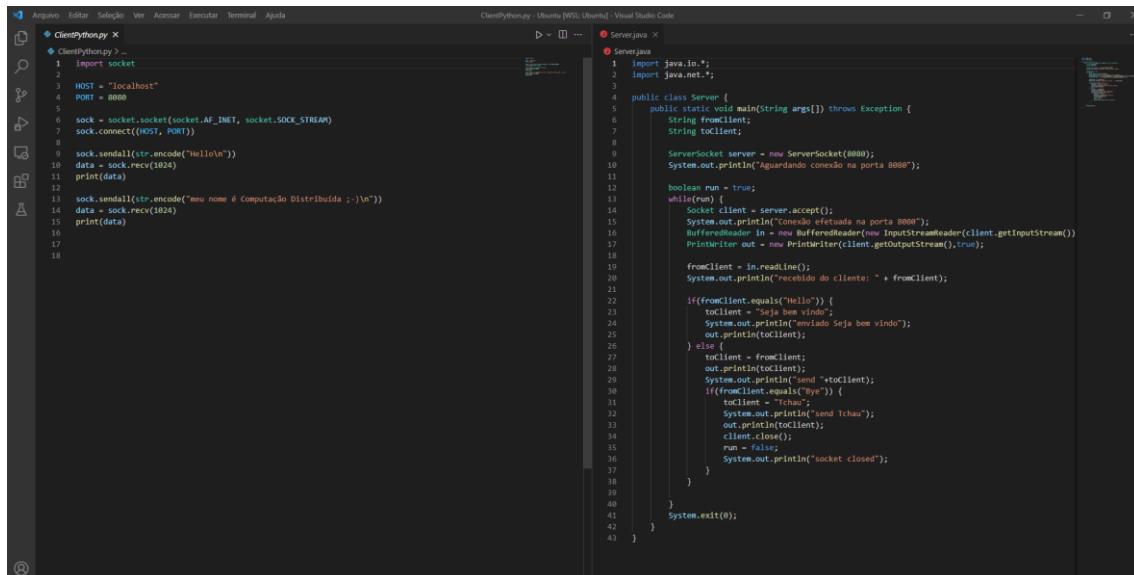


NOME: Pedro Unello Neto

TIA: 41929713

P.1.

Códigos (como passados):



```
ClientPython.py > ...
1 import socket
2
3 HOST = "localhost"
4 PORT = 8080
5
6 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 sock.connect((HOST, PORT))
8
9 sock.sendall(str.encode("Hello\n"))
10 data = sock.recv(1024)
11 print(data)
12
13 sock.sendall(str.encode("meu nome é Computação Distribuída ;-\n"))
14 data = sock.recv(1024)
15 print(data)
16
17
18
```

```
Server.java
1 import java.io.*;
2 import java.net.*;
3
4 public class Server {
5     public static void main(String args[]) throws Exception {
6         String fromClient;
7         String toClient;
8
9         ServerSocket server = new ServerSocket(8080);
10        System.out.println("Aguardando conexão na porta 8080");
11
12        boolean run = true;
13        while(run) {
14            Socket client = server.accept();
15            System.out.println("Conexão efetuada na porta 8080");
16            BufferedReader in = new BufferedReader(new InputStreamReader(client.getInputStream()));
17            PrintWriter out = new PrintWriter(client.getOutputStream(), true);
18
19            fromClient = in.readLine();
20            System.out.println("recebido do cliente: " + fromClient);
21
22            if(fromClient.equals("Hello")) {
23                toClient = "Seja bem vindo";
24                System.out.println("enviado Seja bem vindo");
25                out.println(toClient);
26            } else {
27                toClient = fromClient;
28                out.println(toClient);
29                System.out.println("send " + toClient);
30                if(fromClient.equals("bye")) {
31                    toClient = "tchau";
32                    System.out.println("send tchau");
33                    out.println(toClient);
34                    client.close();
35                    run = false;
36                    System.out.println("socket closed");
37                }
38            }
39        }
40        System.exit(0);
41    }
42}
43
```

```
ClientPython.py > ...
1 import socket
2
3 HOST = "localhost"
4 PORT = 8080
5
6 sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7 sock.connect((HOST, PORT))
8
9 sock.sendall(str.encode("Hello\n"))
10 data = sock.recv(1024)
11 print(data)
12
13 sock.sendall(str.encode("meu nome é Computação Distribuída ;-\n"))
14 data = sock.recv(1024)
15 print(data)
16
17
18
```

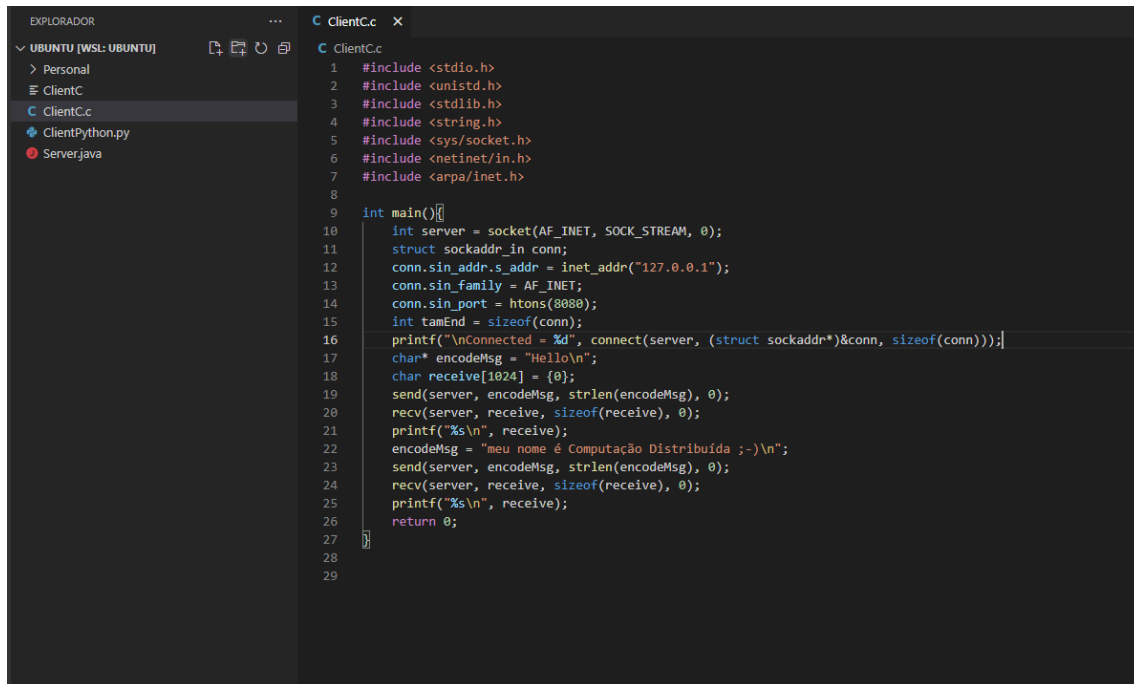
```
Server.java X
Server.java
12     boolean run = true;
13     while(run) {
14         Socket client = server.accept();
15         System.out.println("Conexão efetuada na porta 8080");
16         BufferedReader in = new BufferedReader(new InputStreamReader(client.getInputStream()));
17         PrintWriter out = new PrintWriter(client.getOutputStream(),true);
18
19         fromClient = in.readLine();
20         System.out.println("recebido do cliente: " + fromClient);
21
22         if(fromClient.equals("Hello")) {
23             toClient = "Seja bem vindo";
24             System.out.println("enviado Seja bem vindo");
25             out.println(toClient);
26         } else {
27             toClient = fromClient;
28             out.println(toClient);
29             System.out.println("send "+toClient);
30             if(fromClient.equals("Bye")) {
31                 toClient = "Tchau";
32                 System.out.println("send Tchau");
33                 out.println(toClient);
34                 client.close();
35                 run = false;
36                 System.out.println("socket closed");
37             }
38         }
39     }
40 }
41 System.exit(0);
42 }
43 }
```

Execução (WSL-ubuntu):

```
pedrounello@DESKTOP-OH05V  ×  +  ∨  
pedrounello@DESKTOP-OH05V3U: /mnt/c/Users/pedro/Desktop/Ubuntu$ java Server.java  
Aguardando conexão na porta 8080  
Conexão efetuada na porta 8080  
recebido do cliente: Hello  
enviado Seja bem vindo  
  
pedrounello@DESKTOP-OH05V3U: /mnt/c/Users/pedro/Desktop/Ubuntu$ python3 ClientPython.py  
b'Seja bem vindo\n'  
|
```

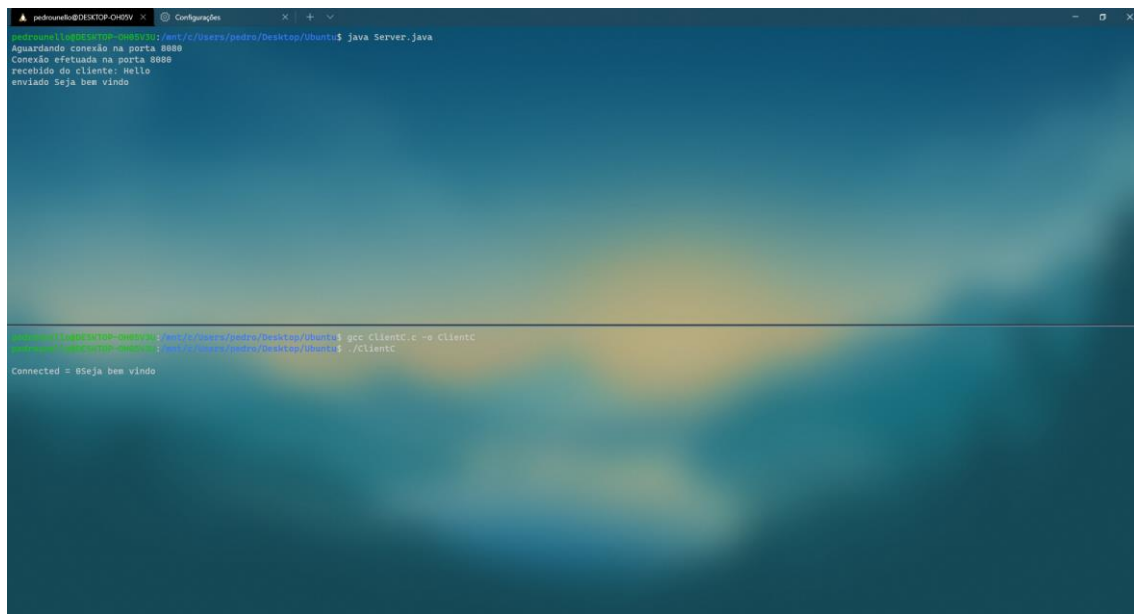
P.2.

Código (em C):



```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <stdlib.h>
4 #include <string.h>
5 #include <sys/socket.h>
6 #include <netinet/in.h>
7 #include <arpa/inet.h>
8
9 int main()
10 {
11     int server = socket(AF_INET, SOCK_STREAM, 0);
12     struct sockaddr_in conn;
13     conn.sin_addr.s_addr = inet_addr("127.0.0.1");
14     conn.sin_family = AF_INET;
15     conn.sin_port = htons(8080);
16     int tamEnd = sizeof(conn);
17     printf("\nConnected = %d", connect(server, (struct sockaddr*)&conn, sizeof(conn)));
18     char* encodeMsg = "Hello\n";
19     char receive[1024] = {0};
20     send(server, encodeMsg, strlen(encodeMsg), 0);
21     recv(server, receive, sizeof(receive), 0);
22     printf("%s\n", receive);
23     encodeMsg = "meu nome é Computação Distribuída ;-)\n";
24     send(server, encodeMsg, strlen(encodeMsg), 0);
25     recv(server, receive, sizeof(receive), 0);
26     printf("%s\n", receive);
27     return 0;
28 }
29
```

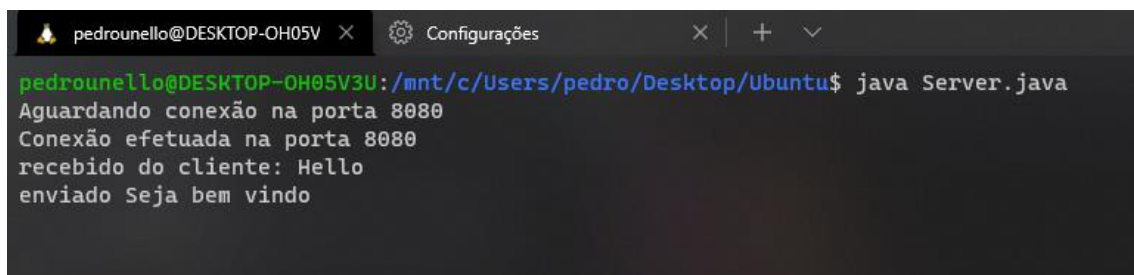
Execução (WSL):



```
pedrounello@DESKTOP-OH05V: ~$ java Server.java
Aguardando conexão na porta 8080
Conexão efetuada na porta 8080
recebido do cliente: Hello
enviado Seja bem vindo

pedrounello@DESKTOP-OH05V: ~$ gcc ClientC.c -o ClientC
pedrounello@DESKTOP-OH05V: ~$ ./ClientC
Connected = 0Seja bem vindo
```

Resposta no java.



```
pedrounello@DESKTOP-OH05V3U: /mnt/c/Users/pedro/Desktop/Ubuntu$ java Server.java
Aguardando conexão na porta 8080
Conexão efetuada na porta 8080
recebido do cliente: Hello
enviado Seja bem vindo
```

Console do C (connected = 0 é confirmação de conexão).

```
pedrounello@DESKTOP-OH05V3U:/mnt/c/Users/pedro/Desktop/Ubuntu$ gcc ClientC.c -o ClientC
pedrounello@DESKTOP-OH05V3U:/mnt/c/Users/pedro/Desktop/Ubuntu$ ./ClientC

Connected = 0Seja bem vindo
```