UNIP – UNIVERSIDADE PAULISTA

Curso de Ciência da Computação

ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

DESENVOLVIMENTO DE UMA FERRAMENTA PARA COMUNICAÇÃO EM REDE

André Ademir de Sousa Oliveira - N629630

Daniel Chrispim Domingos - F263614

Gabriel de Paula Seki - N6853H6

Luiz Guilherme Davies Lencioni - N667125

Marcos Vinicius Restani Avanzini - G13HJB9

Vinícius Amorim - N641JC5

São José dos Campos, 13 de setembro de 2022

ATIVIDADES PRÁTICAS SUPERVISIONADAS - APS

DESENVOLVIMENTO DE UMA FERRAMENTA PARA COMUNICAÇÃO EM REDE

Atividades Práticas Supervisionadas do 5° Semestre do Curso de Ciência da Computação da **Universidade Paulista UNIP.**

Coordenador: Prof. Fernando A. Gotti
Prof. Responsável: André Yoshimi Kusumoto

São José dos Campos, 13 de setembro de 2022

RESUMO

Este trabalho foi desenvolvido com o intuito de demonstrar a funcionalidade de uma aplicação de conversa através de uma rede, aplicando conceitos de comunicação em rede. Para este objetivo, foi apresentado uma introdução a diversos conceitos fundamentais ao entendimento da aplicação e transferência de dados através de uma rede. No decorrer deste documento, será detalhado as definições e padrões da Internet, os diferentes protocolos de comunicação e também os diferentes tipos de topologia de redes de comunicação.

Palavras-chaves: Internet. Protocolos. Padrões da Internet. Redes. Topologia de Redes.

ABSTRACT

This document was developed within the intention of demonstrating the functionality of a local network chat application, applying concepts of network communication. To this objective, it was presented an introduction to various fundamental concepts to the understanding of the application and transfer of data through a network. Through this work, it will be detailed the definitions and standarts of the internet, the different protocols of communication aswell as many different kinds of communication network topologies.

Keywords: Internet. Protocols. Internet Standarts. Networks. Network Topologies.

SUMÁRIO

1 INTRODUÇÃO

As redes de comunicações são intrínsecas à comunicação humana moderna. O conceito delas se manifesta de diversas formas no dia a dia, dos aplicativos de conversas, dos serviços de correio eletrônico à *internet* em si. Até mesmo serviços essenciais à infraestrutura de cidades dependem dessas redes para realizarem suas funções. Por isso, o compreendimento do funcionamento dessas conexões, seus tipos e características individuais se torna cada vez mais importante conforme os avanços tecnológicos nesse meio se aceleram.

O conceito da comunicação entre dois ou mais computadores não é limitado apenas às *chat rooms*, aos fóruns *online* ou serviços de redes sociais. Toda transferência de dados é um tipo de comunicação e assim é aplicável a qualquer tipo de interação *online* ou em uma rede física. Portanto, a importância do entendimento das definições e tecnologias que compõem essas interligações não pode ser reduzida.

Como uma conversa harmônica entre dois indivíduos, a comunicação entre computadores também segue uma série de regras que podem ser definidas como protocolos. Eles garantem que as informações enviadas e recebidas pelo remetente e destinatário sejam compreensíveis para o outro.

1.1 Objetivos Gerais

Este trabalho busca apresentar os conceitos de redes de comunicação e comparar os diferentes métodos de implementação de uma rede física e o funcionamento de um programa que possibilita o envio de mensagens de texto através de uma conexão com interface gráfica.

Acredita-se que este documento possa demonstrar conhecimentos de diferentes métodos de desenvolvimento de *software* de comunicação, além dos métodos existentes para se realizar as conexões necessárias para o funcionamento correto dos mesmos.

1.2 Objetivos Específicos

Com o conhecimento apresentado durantes as aulas, estudos sobre comunicação entre computadores e a linguagem de programação *Python*, foi desenvolvido um programa que utiliza uma conexão para transferir dados entre duas ou mais interfaces.

Além disso, foi disponibilizado exemplos de funções executadas no código e imagens detalhando os diferentes tipos de topologias, redes e protocolos potencialmente utilizados.

1.3 Justificativa

Observando o funcionamento da aplicação criada para este trabalho, é perceptível o uso dos conceitos estudados durante o semestre, empenhando uma interface gráfica para simplificar o entendimento para um indivíduo leigo, sem conhecimento aprofundado de interfaces de linha de comando.

2 FUNDAMENTOS DA COMUNICAÇÃO DE DADOS EM REDE

2.1 Definição

Segundo (NOBREGA FILHO, 2016) “Rede é uma maneira de conectar computadores para que eles tenham consciência do outro e possam unir e compartilhar seus recursos”. Assim com a evolução da tecnologia e a grande necessidade de integrar um meio de comunicação entre os computadores e assim foi criado a comunicação de dados, essa criação permitiu que as máquinas conversem umas com as outras e assim facilitando diversas atividades.

Sua forma de funcionamento é através de um meio de transmissão físico, como cabos ou mais recentemente no ar *Wi-Fi*, a informação transmitida é no formato binário *bits* sendo a mais adequada já que as máquinas manipulam as informações em forma binária. (Porto Editora – redes de comunicação de dados na Infopédia, 2022).

2.2 Internet

A sua origem foi em 1969 após uma necessidade de conectar os computadores dos departamentos da *Advanced Research and Projects Agency* (*ARPA*), assim gerou a *ARPANET* que interligava quatro instituições: Universidade da Califórnia, LA e Santa Bárbara; Instituto de Pesquisa de Stanford e Universidade de Utah. (ESCOLA, Equipe Brasil. *Internet*; Brasil Escola, 2022).

De acordo com o dicionário de Oxford *Languages*, a *Internet* é uma rede de computadores dispersos por todo o planeta que trocam dados e mensagens utilizando um protocolo comum, unindo usuários particulares, entidades de pesquisa, órgãos culturais, institutos militares, bibliotecas e empresas de toda envergadura".

Com sua criação também surgiu a facilidade de transmitir dados para qualquer lugar do planeta e devido a recursos cada vez mais "pesados", a velocidade de transmissão está cada vez mais necessária.

2.3 Protocolos

Protocolo de rede é um conjunto de normas que permite às máquinas conectadas à internet se comuniquem entre si com uma linguagem universal, de forma que qualquer computador que seja produzido consiga interpretar o dado recebido ou enviado. (TEBALDI PEDRO, 2019)

Uma das funções dos protocolos de rede é coletar os dados transmitidos e dividir em pequenos pedaços que são denominados como pacotes, o pacote tem em si a informação do endereçamento de destino e origem do dado. (LONGEN ANDREI, 2018)

2.3.1 IP

O protocolo *IP Internet Protocol* é um dos mais importantes na *web*, com ele é possível fazer a elaboração e transmissão dos pacotes de dados, porém, a entrega dos pacotes não é garantida.

Com o endereço de IP (endereço da máquina) é possível determinar o destino da mensagem, então com a máscara de sub rede irá definir o endereço que se refere a rede de destino, e o campo de *gateway* estreita por padrão saberá o computador específico. (LONGEN ANDREI, 2018)

2.3.2 TCP/IP

TCP/IP é o acrônimo da combinação de dois protocolos: o *TCP* (*Transmission Control Protocol*) e *IP* (*Internet Protocol*), essa combinação é responsável pelo envio e recebimento de dados por toda a internet.

A sua origem foi devido a diversas aplicações da *ARPANET* referente à pesquisas militares em 1969, sua ideia é oferecer rápida troca de dados entre computadores conectados a uma rede. (TEBALDI PEDRO, 2019)

2.3.3 HTTP/HTTPS

O protocolo *HTTP* (*Hypertext Transfer Protocol*) é utilizado para fazer a conexão entre cliente (*browser*) e servidor (*site*), ele é responsável em garantir a sua navegação em sites da internet. (TEBALDI PEDRO, 2019)

Seu funcionamento é baseado em um navegador que faz um pedido de acesso a uma página da web, e o servidor do site decide a permissão de acesso, nesse momento também é enviado os arquivos da página que o usuário deseja acessar. (TEBALDI PEDRO, 2019)

Já o *HTTPS* (*Hyper Text Transfer Secure*) é uma camada extra de proteção que indica para o usuário quais sites são seguros para o acesso, para o site utilizar desse protocolo é necessário possuir um certificado *SSL* que cria uma proteção entre o cliente e o servidor. Essa proteção é feita com uma criptografia que impede atividades maliciosas de coletar dados do usuário.

2.3.4 FTP/SFTP

FTP (*File Transfer Protocol*) é um protocolo de transferência de dados mais simples entre duas máquinas pela rede.

Seu funcionamento é da seguinte forma. Um computador (cliente) faz um pedido de conexão com o servidor para pegar um documento ou arquivo, então o servidor recebe seu pedido de conexão e fornece ou não o arquivo para o cliente. (TEBALDI PEDRO, 2019)

Já o *SFTP* (*Simples File Transfer Protocol*) utiliza uma camada de proteção conhecida como *SSH* (*Secure Shell*) para autenticar e proteger a conexão entre cliente e servidor. No *SFTP* a transferência é feita por pacotes *SSH*, assim o cliente pode definir a quantidade de arquivos que quer transferir e simultaneamente o sistema faz a criação das senhas para reforçar a segurança do processo. (TEBALDI PEDRO, 2019)

2.3.5 Outros protocolos

Segue abaixo uma rápida abordagem de alguns outros tipos de protocolos e suas funções:

Protocolo *POP3* (*Post Office Protocol 3*): é um protocolo usado como mensagem eletrônica;

Protocolo *SSH* (*Secure Shell*): protocolo responsável pela proteção da troca de arquivos entre cliente e servidor;

Protocolo *SMTP* (*Simple Mail Transfer Protocol*): é conhecido por fazer o envio de mensagens eletrônicas, também conhecido como e-mails;

Protocolo *IMAP* (*Internet Message Access Protocol*): voltado para receber e enviar *e-mails*, a sua diferença dos anteriores é a permissão do gerenciamento de arquivos e mensagens pelo usuário diretamente do próprio servidor.

2.4 **OSI**

O modelo *OSI* (*Open System Interconnection*) foi lançado em 1984 e seu objetivo foi servir como modelo padrão para protocolos de comunicação entre diversos sistemas, sua arquitetura é dividida em 7 redes de computadores e o seu padrão permite que diferentes tecnologias sejam utilizadas em conjunto em um ambiente heterogêneo. (Nunes, 2020)

As camadas que compõem o modelo *OSI* são as seguintes: aplicação, apresentação, sessão, transporte, rede, dados e física. (Nunes, 2020)

Figura 1 – Representação do Modelo OSI

Fonte: DataRain, 2020.

2.5 Redes

Em 1969 nos Estados Unidos, dava-se início a origem da internet, na época, era chamada de *Arpanet* e tinha como únicos objetivos conectar universidades americanas e uso militar. Hoje, mais de cinco décadas depois, a Internet e as Redes de Computadores evoluíram explosivamente, tornando-se uma parte essencial da comunicação humana e da infraestrutura como sociedade. Hoje, segundo a ONU, 63% da população mundial tem acesso à internet.

No passado, com a introdução dos microcomputadores no cenário da informática, havia uma estrutura distribuída, na qual, diversos equipamentos e computadores processavam informações de formas isoladas, o que costumava acarretando em uma série de problemas, como a duplicação desnecessária de recursos de hardware e de software. Foi neste cenário onde surgiu a criação das redes de computadores, introduzindo um sistema de comunicação entre as máquinas, permitindo o compartilhamento de recursos. Uma rede pode ser definida como um grupo de pelo menos dois computadores que estão ligados entre si, comunicando-se uns com os outros, compartilhando recursos e informações, com velocidade e praticidade.

2.5.1 Taxonomia de Redes

As redes podem ser classificadas e divididas referentes a capacidade de alcance. São classificadas em:

***PAN – Personal Area Network*:** Comunicação direta entre dispositivos pessoais próximos, usados em residências e pequenos escritórios. Como exemplo *smartwatches*, fones de ouvido sem fio, impressoras, caixas de som *bluetooth*, entre outros dispositivos.

***LAN – Local Area Network*:** Redes privadas localizadas num mesmo prédio, sala ou campus, abrangendo alguns quilômetros de extensão. São utilizadas principalmente para o compartilhamento de recursos e troca de informações.

***MAN – Metropolitan Area Network*:** Semelhantes à *LAN*, porém consegue abranger maiores áreas, como uma cidade inteira.

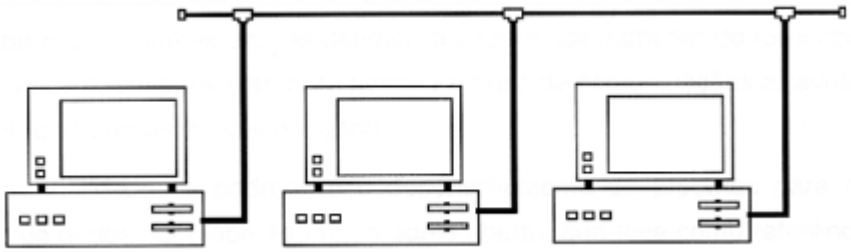
***WAN – Wide Area Network*:** Abrangem grandes áreas, como um país ou um continente inteiro.

2.5.2 Topologias de Redes

Dá-se o nome de Topologia de Redes a disposição física que os computadores estão organizados, suas conexões lógicas. Tem-se as seguintes topologias:

Topologia de Barramento: Topologia onde todos os computadores estão interligados em um único cabo, com uma peça chamada de Terminador de Barramento em cada ponta. Cada nó na barra tem acesso às informações transmitidas. Esta característica facilita as aplicações com mensagens para múltiplas estações.

Figura 2 – Topologia Barramento



Fonte: Souza, 2009.

Vantagens:

• Facilidade na instalação;

• Baixo custo;

• Baixa necessidade de cabos.

Desvantagens:

• Dificuldade em descobrir a causa de problemas de transmissão;

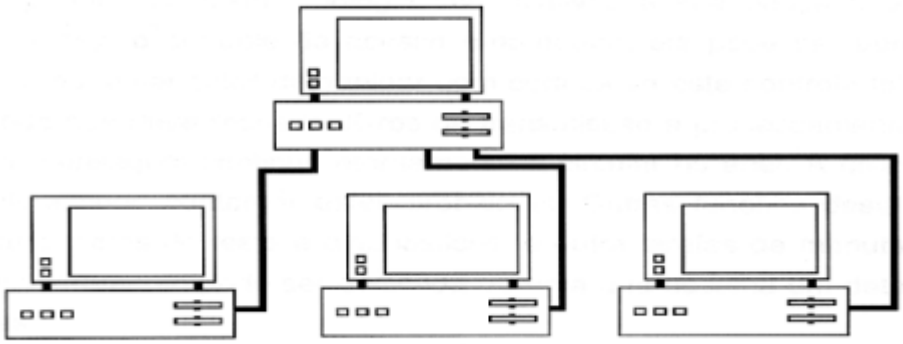
• Se o cabo se danificar a rede deixará de funcionar;

• A rede se torna mais lenta com uso intenso;

• Excesso de colisões.

Topologia Estrela: Nesta topologia, cada nó é interligado em um nó central (chamado de mestre), que é o centro de controle, comumente sendo um *HUB* ou um concentrador. Topologia Estrela é comum em ambientes de rede de grande porte.

Figura 3 – Topologia Estrela



Fonte: Souza, 2009.

Vantagens:

• Gerenciamento Centralizado;

• A adição de estações pode ser feita conectando-se às portas de comunicação que estejam livres;

• A análise de problemas na rede é mais simples;

• Uma máquina ou cabo defeituoso não afeta o funcionamento da rede.

Desvantagens:

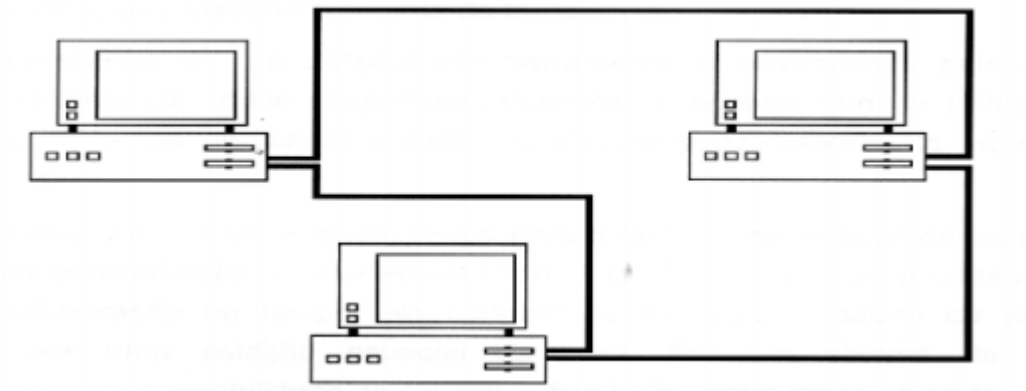
• O número de estações fica limitado ao número de portas do *HUB* ou do *Switch*;

• Maior quantidade de cabos;

• Custo elevado.

Topologia Anel: Nesta topologia, os computadores são todos conectados em forma de *loop* fechado. Um cabo conecta o primeiro computador ao segundo, o segundo ao terceiro e assim por diante, até que o último computador se conecte ao primeiro, formando um círculo (logicamente, fisicamente não há necessidade de formar um círculo). Na Topologia Anel a Informação circula unidirecionalmente até voltar ao remetente.

Figura 4 – Topologia Anel



Fonte: Souza, 2009.

Vantagens:

• Facilidade na adição e remoção de estações;

• Pouca tolerância a falhas.

Desvantagens:

• Mais cara;

• Complexidade de instalação;

• Se um nó sair do ar, todo o sistema fica indisponível.

2.6 Sinais e Transmissão

A camada física tem como função possibilitar a transmissão de uma sequência de *Bits* pela rede, função esta que se dá na conversão destes *Bits* em sinais eletromagnéticos. Em resumo, a conversão de sinais para a transmissão consiste em modificar o sinal contendo a informação e transmiti-la com a menor distorção possível, em uma faixa de frequência limitada, com facilidade de recuperação da informação na recepção e com o menor custo possível.

Na camada física, vários problemas podem acontecer afetando a transmissão do sinal, entre eles:

Atenuação: Acontece quando um sinal viaja por um meio físico, perdendo parte da sua energia para superar a resistência deste meio físico. É assim que um fio transportando sinais elétricos ganha um aumento de temperatura (ficando morno ou até mesmo quente), já que uma parte da energia elétrica do sinal é convertida em calor. Para resolver este problema, é necessário a utilização de amplificadores para aumentar o sinal.

Distorção: Cada componente do sinal apresenta uma velocidade de propagação em um meio e um certo atraso até chegar ao destino. Estas diferenças no atraso podem criar uma diferença de fase se o valor do atraso não for igual à duração do período, e a este problema é dado o nome de distorção de sinais.

Ruído: Ruídos são fatores externos que acabam corrompendo os sinais, diversos tipos de ruídos podem ser prejudiciais, como:

• **Ruído Térmico** - consiste no movimento aleatório de elétrons em um fio, o qual gera um sinal extra não enviado originalmente pelo emissor;

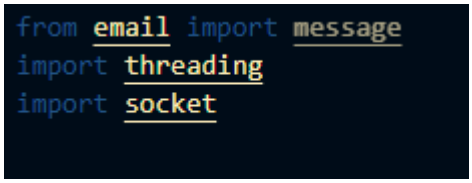
• **Ruído Indutivo** - provém de fontes como motores e eletrodomésticos;

• **Diafonia** - corresponde ao efeito de um fio sobre o outro;

• **Ruído Impulsivo** - um sinal com energia elevada e duração muito curta proveniente de linhas de transmissão de energia.

3 PLANO DE DESENVOLVIMENTO DA APLICAÇÃO

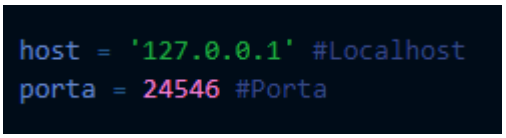
Figura 5 – Importação de Modulos no Server.py



Fonte: Autoria Própria, 2022.

No começo do código foi feita a importação dos módulos de *message*, *threading* e *socket* que serão utilizados.

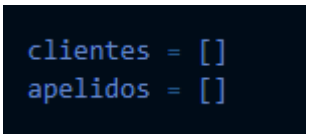
Figura 6 – Definição de Host e Porta no Server.py



Fonte: Autoria Própria, 2022.

No projeto foi utilizado o *IP* do *Local Host*, mas também é possível a utilização de um *IP* de um servidor privado, a porta foi definida aleatoriamente do numero 5000 para cima, portas estas que o *Windows* não usa por padrão.

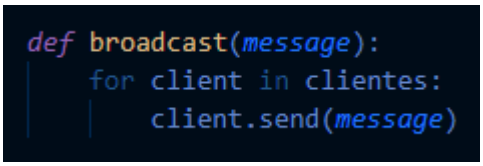
Figura 7 – Listas de Clientes e Apelidos no Server.py



Fonte: Autoria Própria, 2022.

Foram usadas listas vazias para, posteriormente, acrescentar via código os nomes e os Clientes que serão definidos.

Figura 8 – Definindo método Broadcast no Server.py

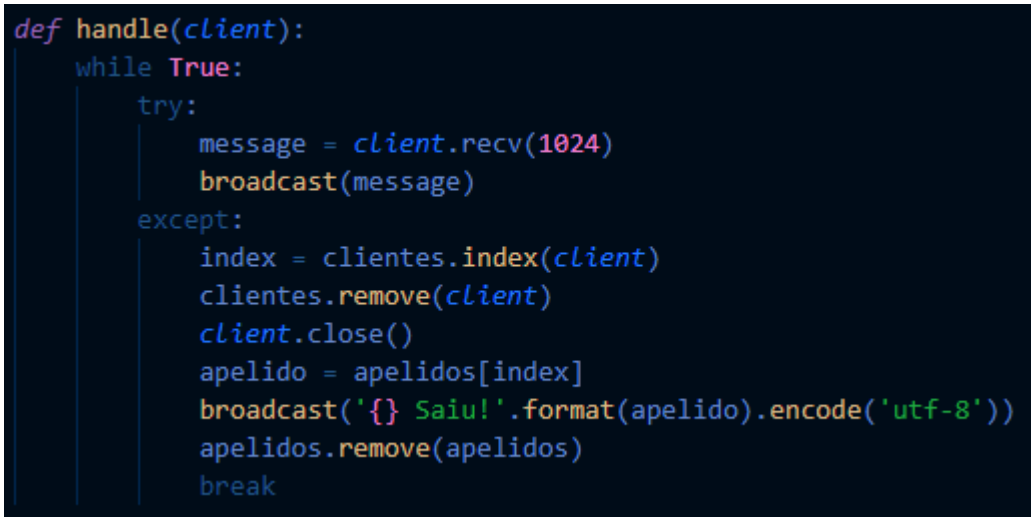


```
def broadcast(message):
    for client in clientes:
        client.send(message)
```

Fonte: Autoria Própria, 2022.

Esse método faz com que toda mensagem enviada por qualquer usuário seja replicada para todos os usuários.

Figura 9 – Definindo método handle no Server.py



```
def handle(client):
    while True:
        try:
            message = client.recv(1024)
            broadcast(message)
        except:
            index = clientes.index(client)
            clientes.remove(client)
            client.close()
            apelido = apelidos[index]
            broadcast('{} Saiu!'.format(apelido).encode('utf-8'))
            apelidos.remove(apelidos)
            break
```

Fonte: Autoria Própria, 2022.

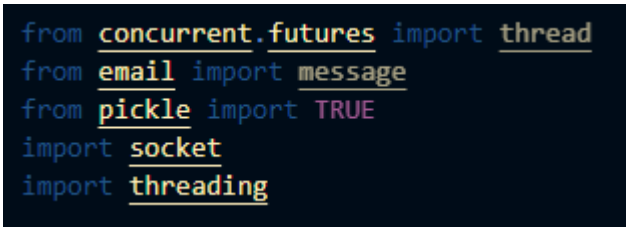
O método *handle* mantém o servidor aberto para receber novas mensagens e, caso algum usuário saia, ele mostra qual usuário é este.

Figura 10 – Definindo método Receber no Server.py

Fonte: Autoria Própria, 2022.

No método Receber o servidor vai receber o cliente e o endereço de *IP* e aceitar ele como um novo usuário, depois faz com que ele defina o seu apelido e depois mostrar aos outros usuários quem entrou na sala de chat.

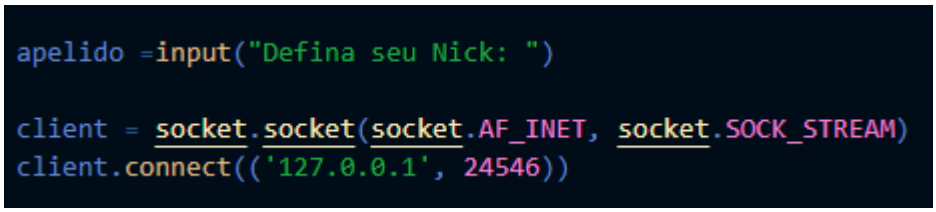
Figura 11 – Importação de módulos no client.py



Fonte: Autoria Própria, 2022.

No começo do código foi feita a importação dos módulos que serão usados mais tarde.

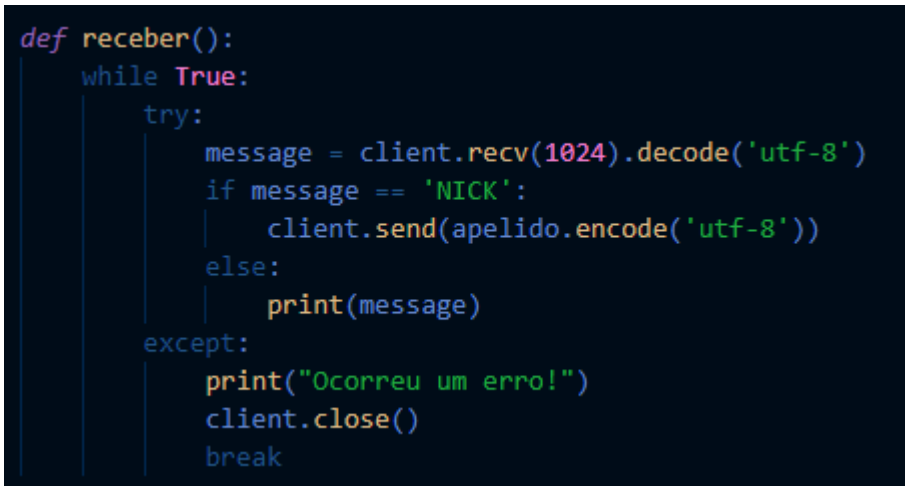
Figura 12 – Definição do apelido e qual o IP e porta vai conectar no client.py



Fonte: Autoria Própria, 2022.

Nesta parte do código é onde ocorre a definição do apelido do usuário e qual o *IP* e Porta o Cliente ira conectar.

Figura 13 – Definindo o método receber no client.py



Fonte: Autoria Própria, 2022.

Este método é para o usuário receber a receber a mensagem de outros usuários que estão conectados no servidor, pode-se ver que se o Cliente receber a palavra ’NICK’ irá mandar o apelido para o servidor, isso ocorre pois no método Receber do *Server.py* ele envia uma mensagem escrita ’NICK’ para chamar esse modulo, assim atribuindo o apelido do usuário a lista vazia.

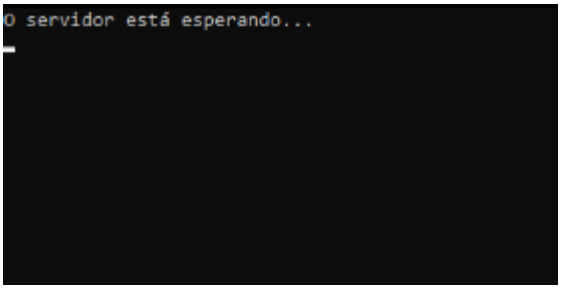
Figura 14 – Definindo o método escrever no client.py

```
def escrever():
    while True:
        message = '{}: {}'.format(apelido, input(''))
        client.send(message.encode('utf-8'))
```

Fonte: Autoria Própria, 2022.

Este método é usado para pegar o *Input* do usuário e manda-lo para o servidor que depois irá replicar essa mensagem para todos pelo método *Broadcast*.

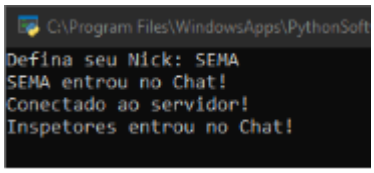
Figura 15 – Servidor funcionando



Fonte: Autoria Própria, 2022.

Nesse passo o servidor esta esperando algum cliente se conectar com ele.

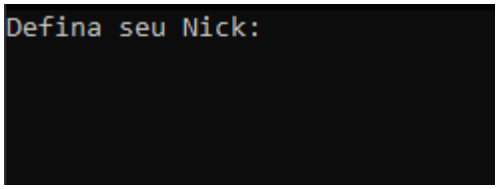
Figura 16 – Servidor com Clientes conectados



Fonte: Autoria Própria, 2022.

Neste passo é onde já ocorreu a conexão dos usuários ao servidor pelo Cliente.

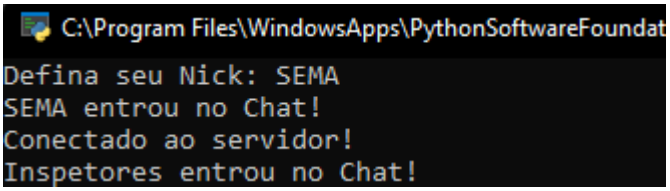
Figura 17 – Cliente definindo o apelido do usuário



Fonte: Autoria Própria, 2022.

Neste passo define-se o nome do usuário para identificação do servidor.

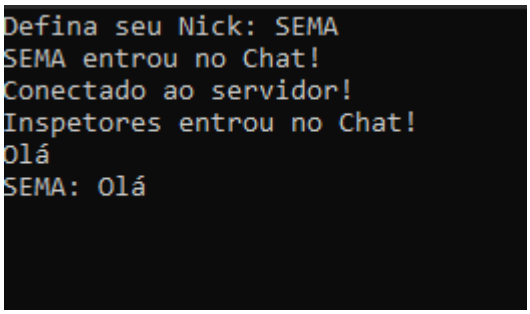
Figura 18 – Cliente



Fonte: Autoria Própria, 2022.

Agora o cliente mostra quem acabou de entrar no seu chat e mostra que você esta conectado no servidor.

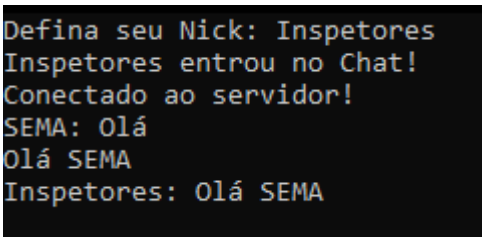
Figura 19 – Mandar a mensagem



Fonte: Autoria Própria, 2022.

Neste passo o usuário esta escrevendo sua mensagem que sera replicada pelo servidor para o outro usuário

Figura 20 – Receber a Mensagem



Fonte: Autoria Própria, 2022.

Agora o usuário Inspetores recebeu a mensagem da SEMA e consegue responder a pergunta.

4 RELATÓRIO COM AS LINHAS DE CÓDIGO

Arquivo 4.1 – Server.py

```
1  from email import message
2  import threading
3  import socket
4
5
6  host = '127.0.0.1' #Localhost
7  porta = 24546 #Porta
8
9  server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10 server.bind((host, porta))
11 server.listen()
12
13 clientes = []
14 apelidos = []
15
16 def broadcast(message):
17     for client in clientes:
18         client.send(message)
19
20 def handle(client):
21     while True:
22         try:
23             message = client.recv(1024)
24             broadcast(message)
25         except:
26             index = clientes.index(client)
27             clientes.remove(client)
28             client.close()
29             apelido = apelidos[index]
30             broadcast('{} Saiu!'.format(apelido).encode('utf-8'))
31             apelidos.remove(apelidos)
32             break
33
34 def receber():
35     while True:
36
37         client, endereço = server.accept()
38         print("conectado com {}".format(endereço))
39
40         client.send('NICK'.encode('utf-8'))
```

```
41          apelido = client.recv(1024).decode('utf-8')
42          apelidos.append(apelido)
43          clientes.append(client)
44
45
46          print("O Apelido é {}!".format(apelido))
47          broadcast( '{} entrou no Chat!'.format(apelido).encode('utf-8'))
48          client.send("Conectado ao servidor!".encode('utf-8'))
49
50          thread = threading.Thread(target=handle, args=(client,))
51          thread.start()
52
53  print("O servidor está esperando...")
54  receber()
```

Arquivo 4.2 – Client.py

```
1  from concurrent.futures import thread
2  from email import message
3  from pickle import TRUE
4  import socket
5  import threading
6
7  apelido =input("Defina seu Nick: ")
8
9  client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10 client.connect(('127.0.0.1', 24546))
11
12 def receber():
13     while True:
14         try:
15             message = client.recv(1024).decode('utf-8')
16             if message == 'NICK':
17                 client.send(apelido.encode('utf-8'))
18             else:
19                 print(message)
20         except:
21             print("Ocorreu um erro!")
22             client.close()
23             break
24
25 def escrever():
26     while True:
27         message = '{}: {}'.format(apelido, input(''))
28         client.send(message.encode('utf-8'))
29
30
31 receber_thread = threading.Thread(target=receber)
32 receber_thread.start()
33
34 escrever_thread = threading.Thread(target=escrever)
35 escrever_thread.start()
```

REFERÊNCIAS

CARLOS, J. ***Redes de Comunicação de Dados | Principais Conceitos***. [S.l.], 2021. Disponível em: <https://www.crtrj.gov.br/redes-de-comunicacao-de-dados-principais-conceitos/>. Acesso em: 29 mar. 2022.

CARMONA, R. A. H. T. ***UNIVERSIDADE REDES: Torne-se um especialista em redes de computador***. [S.l.]: Digerati Books, 2005.

DATARAIN. ***O que é o modelo OSI?*** [S.l.], 2020. Disponível em: <https://www.datarain.com.br/blog/tecnologia-e-inovacao/o-que-e-o-modelo-osi/>. Acesso em: 29 mar. 2022.

ESCOLA, E. B. ***Internet***. [S.l.], 2022. Disponível em: <https://brasilescola.uol.com.br/informatica/internet.htm>. Acesso em: 29 mar. 2022.

FIGUEIREDO, I. L. ***História das Redes de Computadores***. [S.l.], 2013. Disponível em: <https://www.oficinadanet.com.br/post/10123-historia-das-redes-de-computadores>. Acesso em: 29 mar. 2022.

KUROSE, K. W. R. J. F. ***Redes de Computadores e a Internet: uma abordagem top-down***. [S.l.]: Person Addison Wesley, 2006.

LONGEN, A. ***Protocolos de Rede: O Que São, Como Funcionam e Tipos de Protocolos de Internet***. [S.l.], 2020. Disponível em: <https://www.weblink.com.br/blog/tecnologia/conheca-os-principais-protocolos-de-internet/>. Acesso em: 29 mar. 2022.

OURENCO, A. ***O que é internet e como surgiu***. [S.l.], 2020. Disponível em: <https://www.opservices.com.br/protocolos-de-rede/>. Acesso em: 29 mar. 2022.

PINTO, P. ***Redes – Sabe o Que São Sockets de Comunicação? (Parte I)***. [S.l.], 2012. Disponível em: <https://pplware.sapo.pt/tutoriais/networking/redes-sabe-o-que-sao-sockets-de-comunicacao-parte-i/>. Acesso em: 29 mar. 2022.

PISA, P. ***O Que é IP?*** [S.l.], 2012. Disponível em: <https://www.techtudo.com.br/noticias/2012/05/o-que-e-ip.ghtml>. Acesso em: 29 mar. 2022.

SILVA, I. N. da. ***UNIDADE 4 – Protocolos Usando Modelos ISO/OSI***. [S.l.], 2020. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/5769950/mod_resource/content/1/REDES_Aula07.pdf>. Acesso em: 29 mar. 2022.

SOUZA, A. C. ***Redes de Computadores***. [S.l.], 2009. Disponível em: <http://www.ifba.edu.br/professores/antoniocarlos/index_arquivos/redesdecomputadores.pdf>. Acesso em: 29 mar. 2022.

TANENBAUM, A. S. ***Redes de Computadores***. [S.l.]: Editora Campus, 2003.

TEBALDI, P. C. ***Conheça os principais protocolos de rede e seus usos!*** [S.l.], 2019. Disponível em: <https://www.opservices.com.br/protocolos-de-rede/>. Acesso em: 29 mar. 2022.

TORRES, G. ***Redes de Computadores - Curso Completo***. [S.l.]: Axcel Books do Brasil Editora, 2001.