

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO
DEPARTAMENTO DE COMPUTAÇÃO E ELETRÔNICA**

Aulas práticas - Laboratório de Redes

São Mateus, ES

2022

Sumário

Sumário	2
Aula prática I	3
Aula prática II	18
Aula prática III	24
Aula prática IV	32
Aula prática V	41

Aula prática I

Nessa aula iremos ver na prática uma comunicação TCP à nível de socket em uma arquitetura de redes do tipo cliente-servidor. Para tanto, utilizaremos a linguagem Python para construir uma aplicação simples através de script e faremos a sua execução via terminal no sistema operacional Windows 10. Além disso, para um maior detalhe de como ocorre essa comunicação usaremos o Wireshark, uma ferramenta conhecida como *sniffer*, isto é, um “farejador” capaz de fazer a escuta, digamos assim, de todos os pacotes que trafegam pela rede.

1. Socket servidor

1.1. Importação da biblioteca

É necessário realizar a importação da biblioteca em python responsável por permitir a realização da programação baseada em socket.

```
13| # IMPORTANDO A BIBLIOTECA SOCKET
14| import socket
```

1.2. Constantes

É preciso definir algumas constantes ou parâmetros de conexão para que a comunicação entre os processos cliente e servidor possa ser estabelecida. Perceba que a configuração da porta possui uma observação referente a sua numeração. É necessário que a porta de comunicação seja configurada acima de 1024 devido a alguns serviços conhecidos já utilizarem portas de valores mais baixo como a porta 80 referente ao serviço web ou a porta 25 referente ao serviço FTP e etc.

```
16| # CONSTANTES DO SOCKET
17| HOST = 'localhost' # NOME DO SERVIDOR. PODERIA SER INFORMADO O IP
18| PORT = 60000       # NÚMERO DA PORTA DE CONEXÃO. DEVE SER UM NÚMERO ELEVADO ACIMA DE 1024
```

1.3. Criando o socket

A criação de um ponto de comunicação no processo servidor se dá mediante a diretiva **socket()**, ela é encarregada de receber as informações referentes à família de protocolos utilizada e o tipo do protocolo, de tal modo que **IF_INET** representa a família de protocolos IPV4 e **SOCK_STREAM** especifica que será utilizado fluxo de bytes na comunicação e portanto o tipo de protocolo será o TCP.

```
20| # CRIANDO UM OBJETO SOCKET
21| # INFORMAMOS A FAMÍLIA DE PROTOTOCOLO E O TIPO DE PROTOCOLO
22| sckt_tcp = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # IF_INET -> FAMÍLIA DE PROTOCOLO = IPV4, SOCK_STREAM -> TIPO DE PROTOCOLO = TCP
```

1.4. Associando os parâmetros de conexão ao socket

A diretiva ***bind()***, informa ao SO a qual IP e porta estará associado aquele socket. Portanto, o IP=127.0.0.1 e a porta 60000 ficam atreladas aquele socket que será gerenciado pelo sistema operacional.

```
24 # INFORMAMOS PARA O SOCKET QUEM É O HOST E A PORTA DE CONEXÃO COM O SERVIDOR
25 sckt_tcp.bind((HOST, PORT))
```

1.5. O modo de escuta

A diretiva ***listen()*** indica que o socket ficará operando em modo de escuta, ou seja, que estará aguardando por conexão de algum cliente. Em outras palavras, o SO, através do socket, fica monitorando a porta configurada para aquele socket, aguardando conexão e requisições de algum cliente ao servidor.

```
27 # COLOCANDO O SOCKET NO MODO DE ESCUTA
28 sckt_tcp.listen(1) # O SERVIDOR FICA "ESCRUTANDO" NA PORTA ESPECIFICADA
```

1.6. Stand-by

Enquanto nenhum cliente se conecta ao servidor. É exibida uma mensagem de espera por conexão.

```
29 print("Aguardando a conexão de um cliente")
```

1.7. Aceitando conexões

A diretiva ***accept()*** fica responsável por informar aquele aceitar a(s) conexão(ões) feita por cliente(s) ao servidor. Logo, o ***connect()*** realizado por um cliente é aceito pelo socket servidor mediante essa diretiva. Como resultado desse ***accept()*** o socket nos retorna a conexão e o seu endereço (IP e porta) estabelecidas com o cliente, sendo essas informações salvas nas variáveis ***conn*** e ***ender***. A variável ***conn*** é a variável do SO associada aquela conexão, enquanto que a variável ***ender***, armazena o IP e a porta do cliente conectado.

```
31 # ACEITANDO A CONEXÃO
32 conn, ender = sckt_tcp.accept() # O MÉTODO ACCEPT NOS RETORNA A CONEXÃO (conn) E O ENDEREÇO (ender) DA MÁQUINA CONECTADA À ELE, OU SEJA DO CLIENTE
```

1.8. Recebendo mensagens

Iniciamos um loop para recebimento de mensagens do cliente. Enquanto enquanto houver mensagens sendo enviadas, elas são recebidas.

```
37 | # INICIANDO A TROCA DE MENSAGENS
38 | while True:
```

A diretiva **recv()** é responsável por receber as mensagens enviadas pelo cliente com um tamanho máximo de 1024 *bytes*.

```
39 |     # SALVANDO OS DADOS RECEBIDOS NA VARIÁVEL dados
40 |     dados = conn.recv(1024) # AQUI NÓS ESPECIFICAMOS NO MÉTODO recv O TAMANHO MÁXIMO DOS DADOS COMO ATÉ 1024 bytes
```

A saída do loop é feita quando não há mais dados sendo enviados. Assim, verificamos se não há mais dados paramos o *status* de recebimento de mensagens através do fechamento da conexão estabelecida com o cliente.

```
41 |     # VERIFICANDO SE NÃO HÁ MAIS DADOS
42 |     if not dados:
43 |         print("Fechando a conexão")
44 |         conn.close()    # FECHANDO A CONEXÃO COM O CLIENTE
45 |         break
```

Ao receber uma mensagem, o servidor retorna para o cliente a mensagem recebida, mas agora em caixa alta.

```
46 |     # CASO AINDA HAJA DADOS SENDO RECEBIDOS
47 |     conn.sendall(dados.upper()) # ENVIAMOS DE VOLTA PARA O CLIENTE OS DADOS RECEBIDOS
```

1.9. Finalizando o socket

Após todo esse processo o socket é encerrado através da diretiva **close()**.

```
48 | sckt_tcp.close()      # ENCERRA O SOCKET SERVIDOR
```

2. Socket cliente

2.1. Importação da biblioteca

Assim como no socket do servidor, é necessário realizar a importação da biblioteca em python responsável por permitir a realização da programação baseada em socket.

```
13 | # IMPORTANDO A BIBLIOTECA SOCKET
14 | import socket
```

2.2. Constantes

De modo semelhante ao que foi feito no socket do servidor, aqui nós também definimos constantes de conexão necessárias ao socket. Para o socket cliente nós definimos o endereço IP do servidor e a porta na qual queremos nos comunicar com ele.

```
16 # CONSTANTES DO SOCKET
17 HOST = '127.0.0.1' # IP DO SERVIDOR
18 PORT = 60000 # NÚMERO DA PORTA DE CONEXÃO COM O SERVIDOR.
```

2.3. Criando o socket

A criação do socket cliente segue o mesmo padrão do que foi realizado durante a criação do socket do servidor. Informamos a família de protocolos e o tipo de protocolo, ou seja, família IPV4 e protocolo TCP.

```
20 # CRIANDO UM OBJETO SOCKET
21 # INFORMAMOS A FAMÍLIA DE PROTOTOCOLO E O TIPO DE PROTOTOCOLO
22 sckt = socket.socket(socket.AF_INET, socket.SOCK_STREAM) # IF_INET -> FAMÍLIA DE PROTOTOCOLO = IPV4, SOCK_STREAM -> TIPO DE PROTOTOCOLO = TCP
```

2.4. Solicitação de conexão com o servidor

A diretiva **connect()** é responsável por solicitar uma conexão. Assim, o cliente, utilizando o seu socket e invocando a diretiva acima, informa qual o endereço IP do servidor e em qual ele deseja se conectar.

```
24 # SOLICITANDO A CONEXÃO COM SERVIDOR
25 sckt.connect((HOST, PORT)) # INFORMAMOS O HOST E A PORTA DE CONEXÃO COM O SERVIDOR
```

2.5. Informação a ser enviada ao servidor

A aplicação desenvolvida consiste na troca de mensagens entre cliente e servidor, de tal modo que o cliente envia uma mensagem qualquer para o servidor, e este retorne a mesma mensagem, mas em caixa alta. Sendo assim, na linha 28 é solicitado ao cliente uma mensagem de envio.

```
27 # SOLICITANDO A MENSAGEM AO CLIENTE PARA PODER SER TRANSFORMADA EM CAIXA ALTA
28 mensagem = input("Entre com a mensagem: ");
```

2.6. O envio da mensagem

A diretiva **sendall()** fica responsável por fazer o envio de todas as mensagens informadas pelo cliente. Porém, antes de ser enviada, nós a codificamos através do método **encode()** em uma *string* para que não ocorra nenhum tipo de “problema” com a mensagem, visto que esta pode conter caracteres especiais

```
30 # ENVIANDO UMA MENSAGEM PARA O SERVIDOR
31 sckt.sendall(str.encode(mensagem)) # CODIFICAMOS A MENSAGEM EM STRING ANTES DE SER ENVIADA AO SERVIDOR, VISTO QUE PODE HAVER CARACTERES ESPECIAIS
```

2.7. Recebimento de mensagens do servidor

Através da diretiva **recv()** o socket cliente recebe os dados ecoado pelo servidor nele é especificado o tamanho máximo em bytes, no caso o tamanho máximo definido foi 1024 bytes.

```
33 # SALVANDO A RESPOSTA DO SERVIDOR NA VARIÁVEL DADOS
34 dados = sckt.recv(1024) # DE MODO SEMELHANTE AO QUE FOI FEITO NO SOCKET DO SERVIDOR, AQUI NÓS ESPECIFICAMOS NO MÉTODO recv O TAMANHO MÁXIMO DOS DADOS COMO ATÉ 1024
```

2.8. Decodificando a mensagem recebida

Ao receber a mensagem ecoada pelo servidor, nós a decodificamos

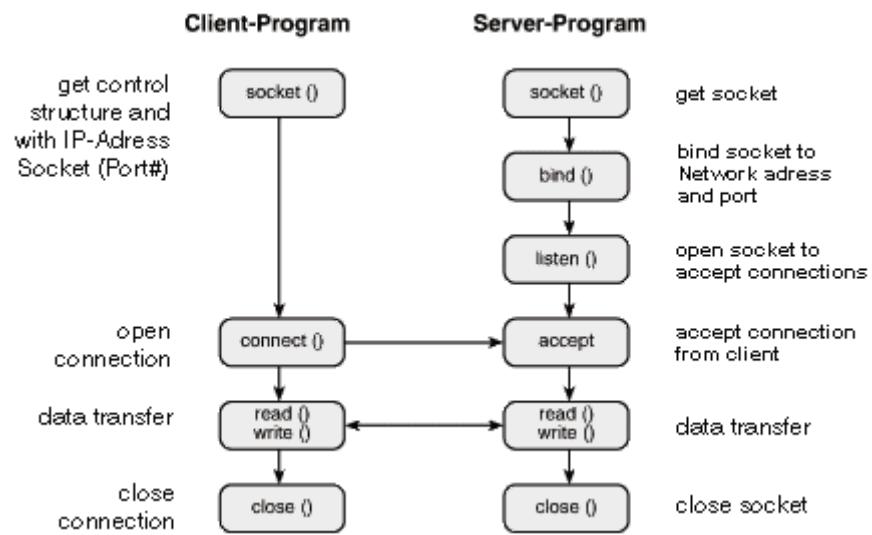
```
35 # EXIBINDO NA TELA A MENSAGEM RECEBIDA PELO RECEBIDA PELO SERVIDOR
36 print("Mensagem recebida: ", dados.decode()) # DECODIFICAMOS A MENSAGEM RECEBIDA ANTES DE A EXIBIRMOS
```

2.9. Finalizando o socket

Após o socket servidor ter encerrado a conexão e ter sido finalizado, o socket cliente também é encerrado.

```
|37| sckt.close()      # ENCERRA O SOCKET CLIENTE
```

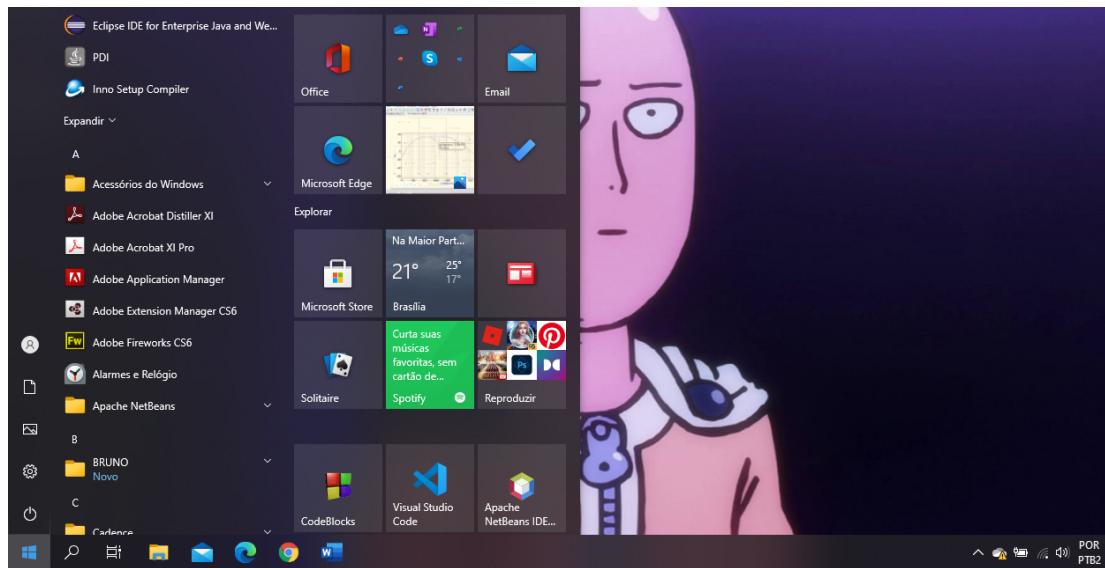
3. Estrutura envolvida na aplicação cliente x servidor baseada em sockets TCP.



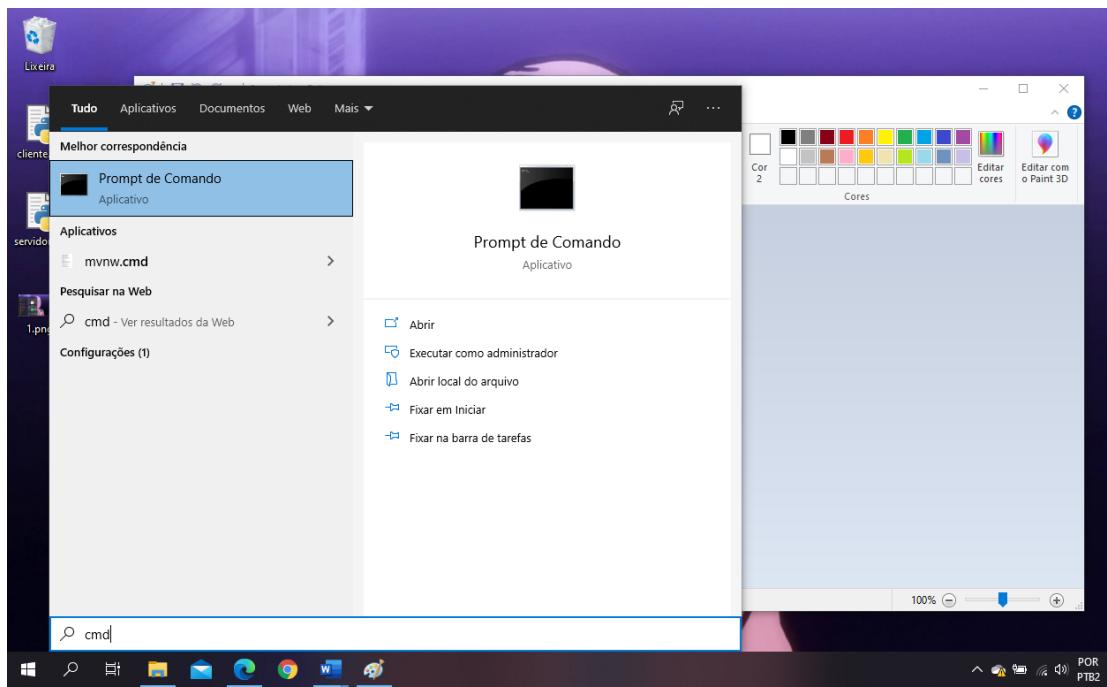
4. Captura de pacotes via wireshark durante a execução da aplicação.

- Executando aplicação servidor:

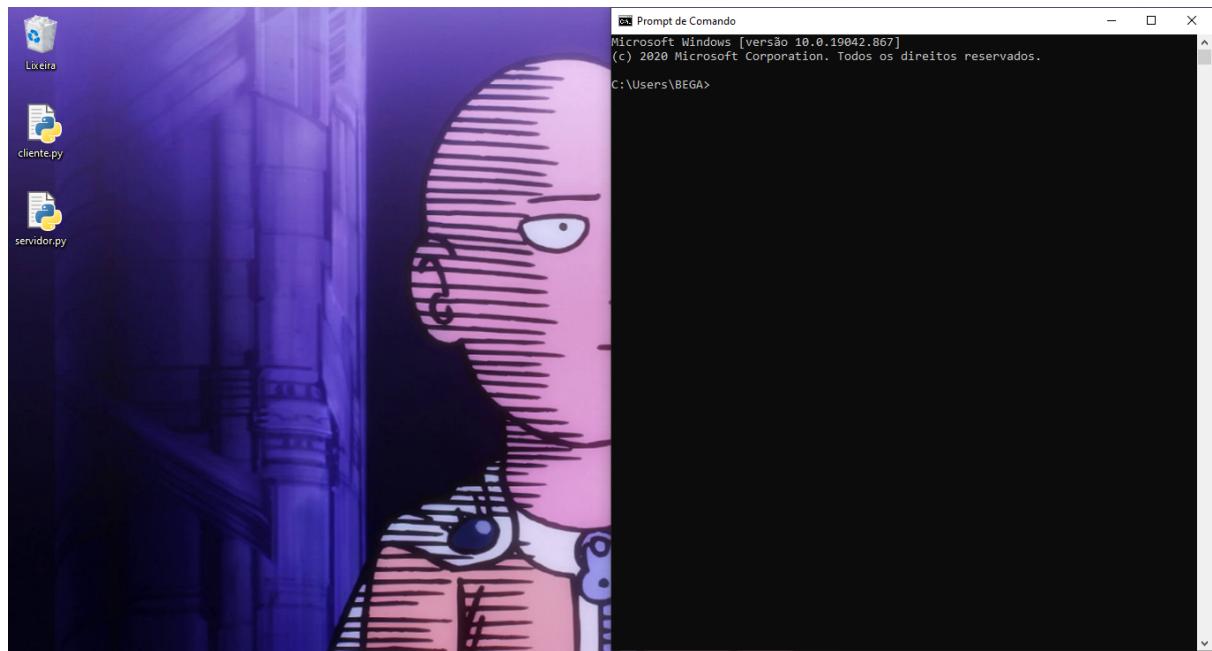
Acesse o menu iniciar do Windows.



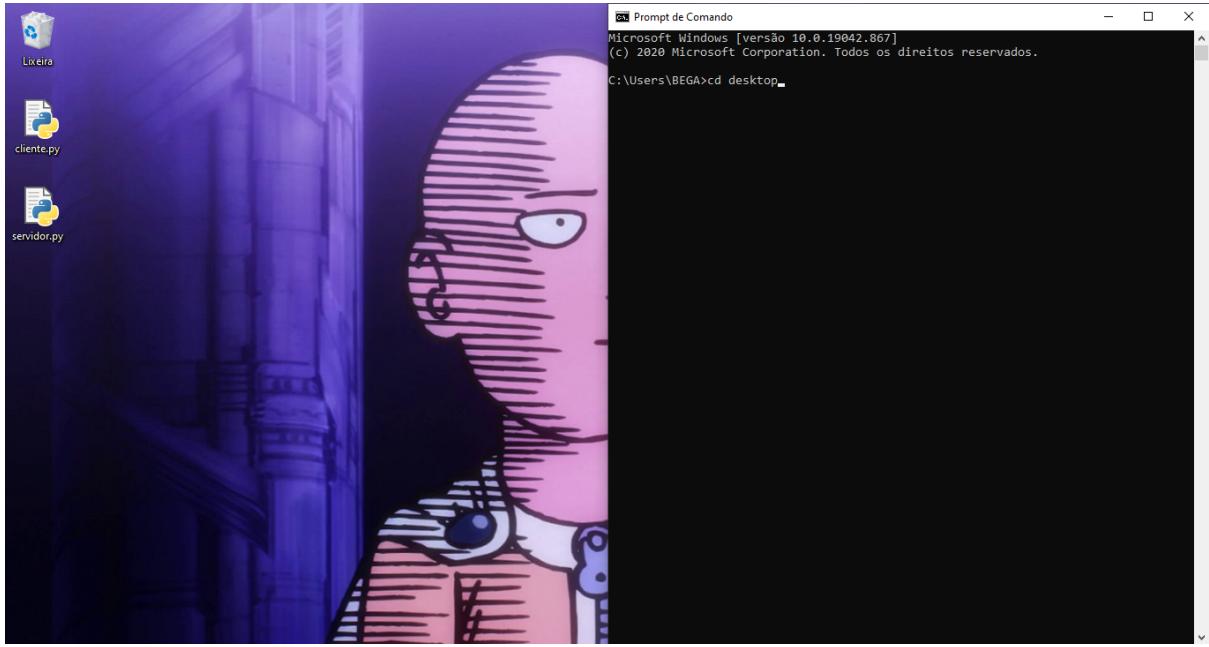
- Digite cmd e pressione Enter.



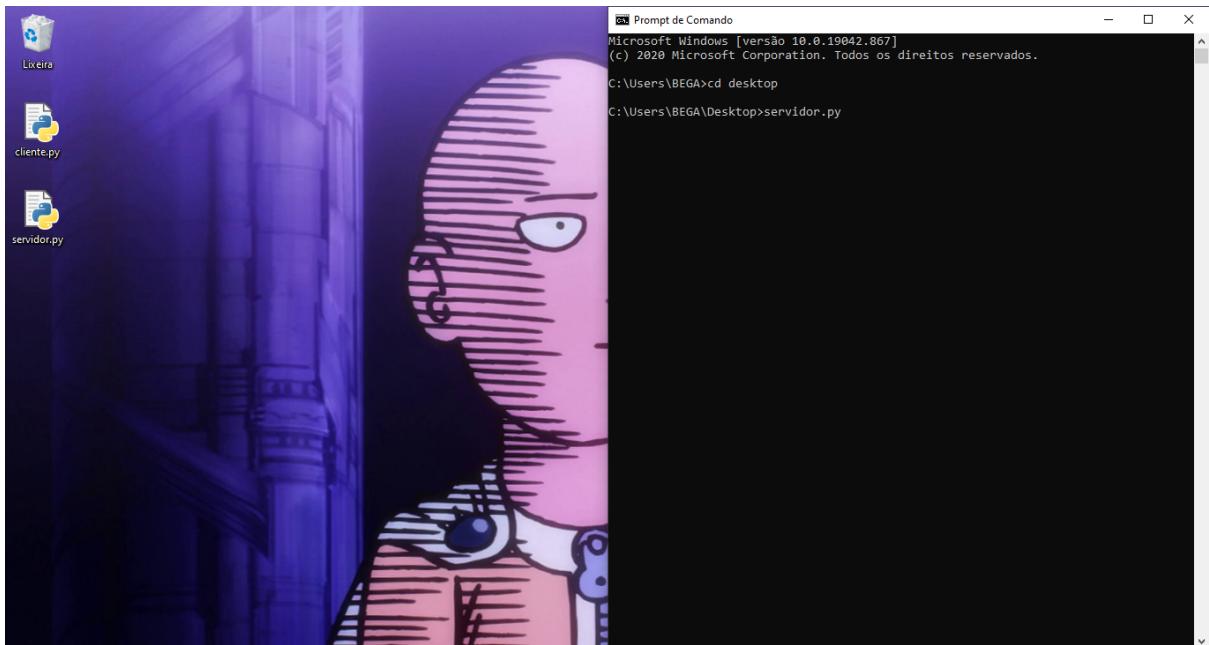
- Pressione "Enter". Será exibido o prompt de comando do Windows.



- Navegue até o diretório onde estão localizados os scripts da aplicação. No meu caso, eles estão na área de trabalho. Portanto, digite: cd desktop e pressione Enter.



- Em seguida digite o nome do script responsável por executar o socket no lado do servidor: `servidor.py` em seguida pressione Enter.

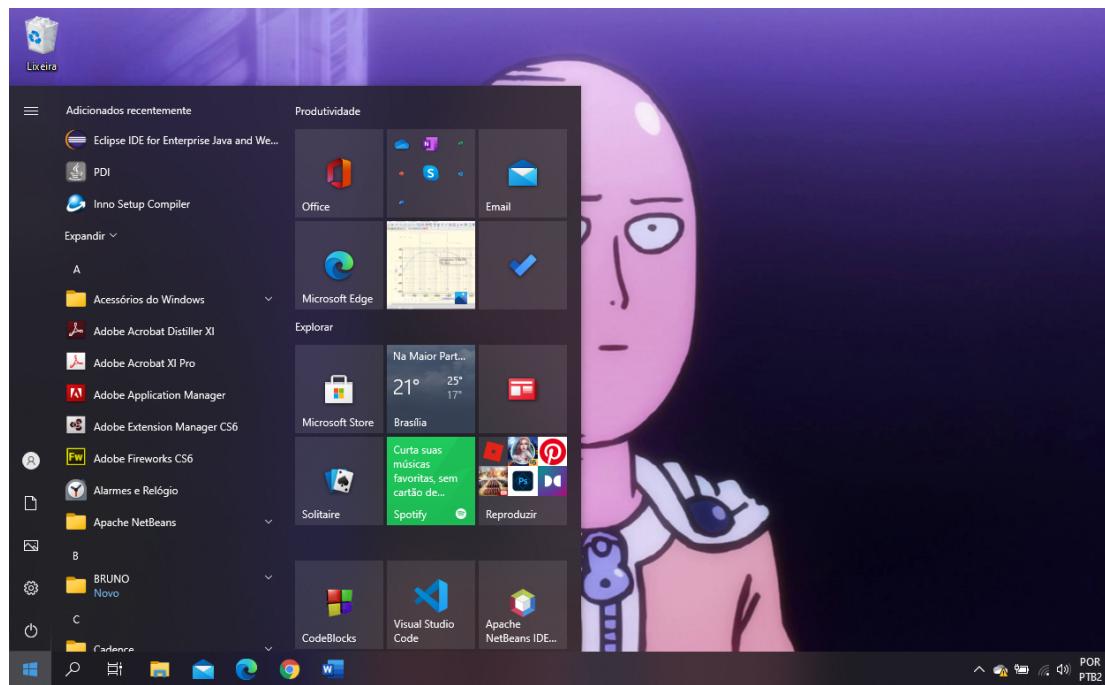


- O socket no lado servidor será iniciado. Ele aguarda pela conexão de um cliente.

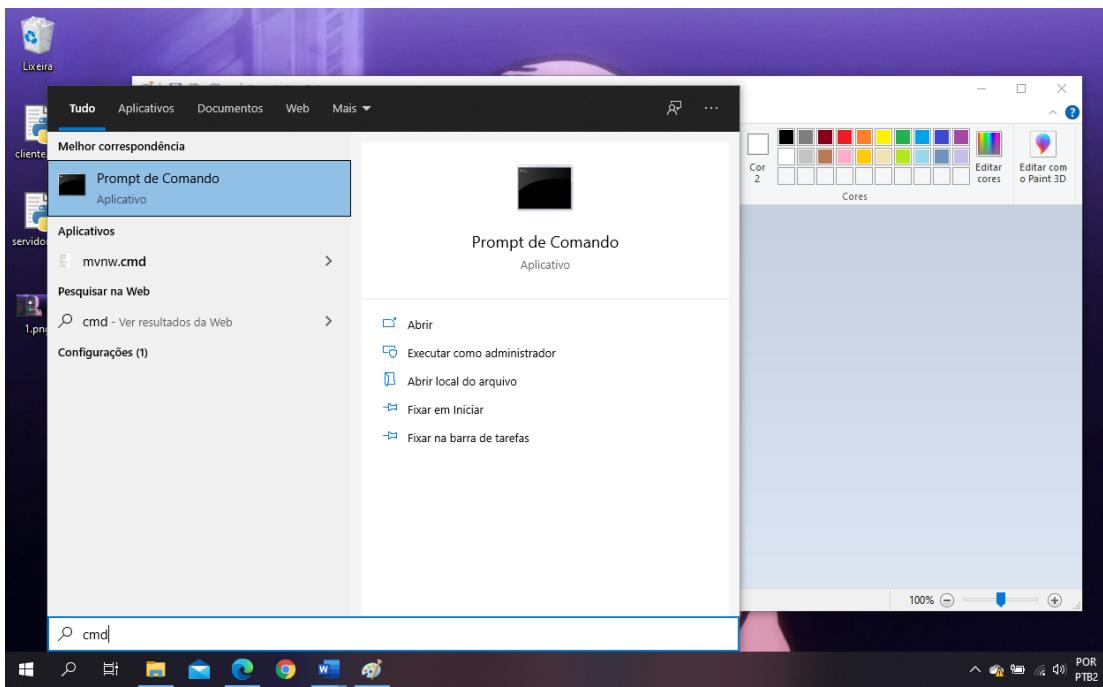


5. Executando o script cliente

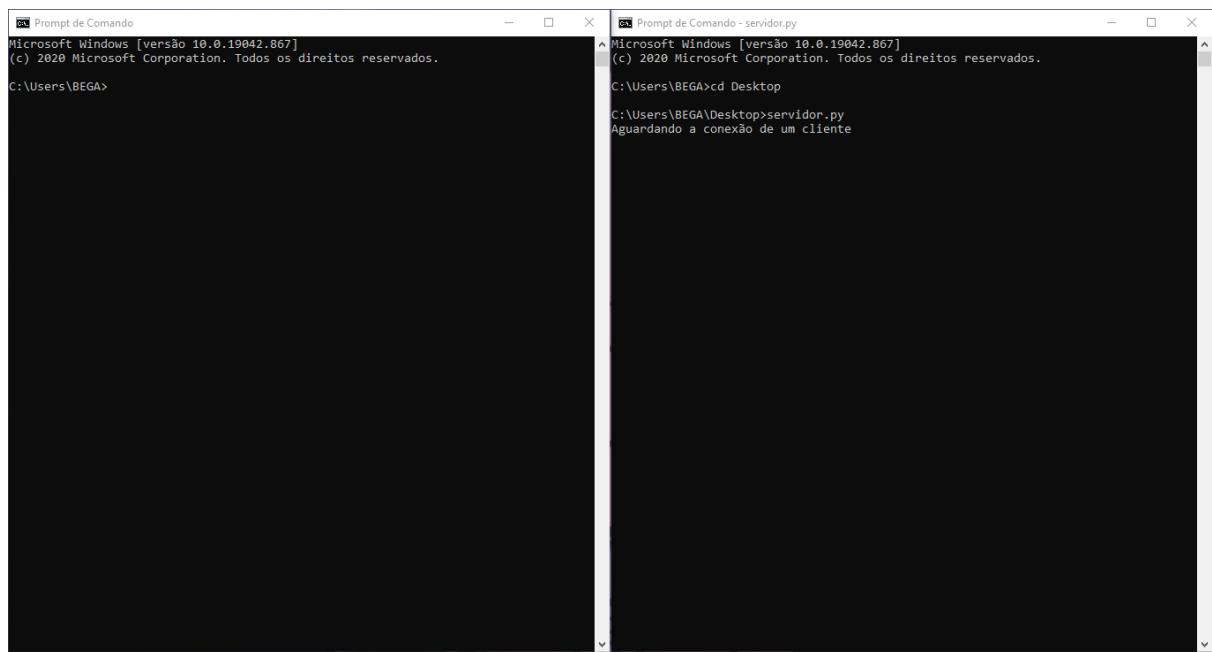
- Acesse o menu iniciar do Windows.



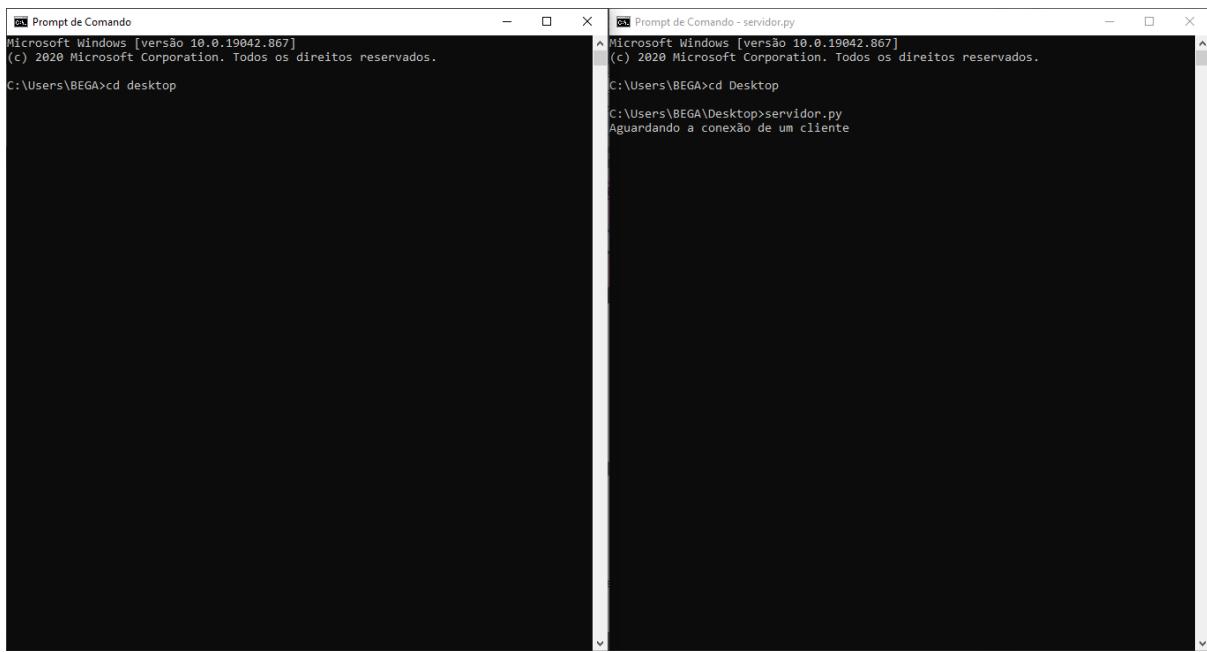
- Digite: cmd e pressione Enter.



- Será exibido o prompt de comando do Windows.



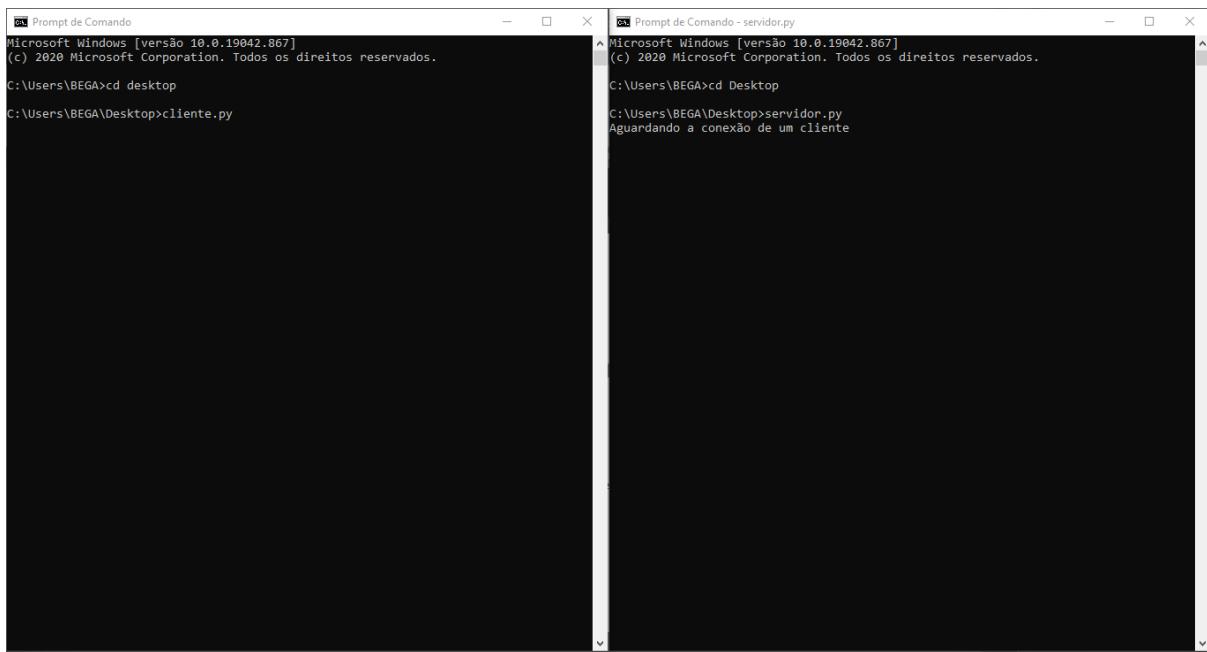
- Navegue até o diretório onde está o script do cliente. No meu caso, ele também está na área de trabalho. Portanto digite: cd desktop e pressione Enter.



The image shows two separate Windows Command Prompt windows side-by-side. The left window, titled 'Prompt de Comando', has the command 'cd desktop' entered and executed. The right window, titled 'Prompt de Comando - servidor.py', also has 'cd Desktop' entered and executed, followed by the message 'Aguardando a conexão de um cliente' (Waiting for a client connection).

```
C:\Users\BEGA>cd desktop
C:\Users\BEGA\Desktop>servidor.py
Aguardando a conexão de um cliente
```

- Em seguida digite o nome do script responsável por executar o socket no lado do cliente: cliente.py em seguida pressione Enter.



The image shows two separate Windows Command Prompt windows side-by-side. The left window, titled 'Prompt de Comando', has 'cd desktop' and 'cliente.py' entered and executed. The right window, titled 'Prompt de Comando - servidor.py', has 'cd Desktop' entered and executed, followed by the message 'Aguardando a conexão de um cliente' (Waiting for a client connection).

```
C:\Users\BEGA>cd desktop
C:\Users\BEGA\Desktop>cliente.py
C:\Users\BEGA>cd Desktop
C:\Users\BEGA\Desktop>servidor.py
Aguardando a conexão de um cliente
```

- O socket cliente é criado e é estabelecido a conexão com o servidor. Ao mesmo tempo, o socket a aplicação no lado cliente aguarda a entrada de um texto a ser informado pelo cliente.

The image shows two separate windows titled "Prompt de Comando".
The left window (cliente.py) shows:
Microsoft Windows [versão 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.
C:\Users\BEGA>cd Desktop
C:\Users\BEGA\Desktop>cliente.py
Entre com a mensagem: -
This is a blank window.
The right window (servidor.py) shows:
Microsoft Windows [versão 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.
C:\Users\BEGA>cd Desktop
C:\Users\BEGA\Desktop>servidor.py
Aguardando a conexão de um cliente
Conectado em: ('127.0.0.1', 49891)
This is a blank window.

- Digite uma mensagem qualquer e pressione Enter.

The image shows two separate windows titled "Prompt de Comando".
The left window (cliente.py) shows:
Microsoft Windows [versão 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.
C:\Users\BEGA>cd Desktop
C:\Users\BEGA\Desktop>cliente.py
Entre com a mensagem: bom dia, bruno!
Mesagem recebida: BOM DIA, BRUNO!
C:\Users\BEGA\Desktop>-
This is a blank window.
The right window (servidor.py) shows:
Microsoft Windows [versão 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.
C:\Users\BEGA>cd Desktop
C:\Users\BEGA\Desktop>servidor.py
Aguardando a conexão de um cliente
Conectado em: ('127.0.0.1', 49891)
Fechando a conexão
C:\Users\BEGA\Desktop>-
This is a blank window.

- A aplicação é finalizada e os sockets são encerrados

5. Captura de pacotes via wireshark durante a execução da aplicação.

O estabelecimento da conexão, como bem sabemos, por ser tratar de uma conexão TCP, isto é, orientada à conexão, ela é realizada em três etapas. Conhecida como **3-Way-Handshake**. Nela, o cliente envia um pacote TCP com a flag ACK ao servidor de destino informando, que deseja estabelecer uma comunicação. O servidor por sua vez responde com um pacote contendo as flags SYM, ACK, basicamente informando

ao cliente que “Tudo bem, pode se comunicar.” O cliente então envia outro pacote TCP com a flag ACK dizendo “Ok. Vou começar a me comunicar”. Todo esse processo pode ser visto na captura de pacotes abaixo, em destaque.

3-Way-Handshake

Porta de origem (Cliente): 51685
Porta de destino (Servidor): 60000

Flag ACK (Acknowledge).

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	51685 → 60000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
2	0.000057	127.0.0.1	127.0.0.1	TCP	56	60000 → 51685 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
3	0.000094	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	4.093925	127.0.0.1	127.0.0.1	TCP	49	51685 → 60000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=5
5	4.093961	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
6	4.094037	127.0.0.1	127.0.0.1	TCP	49	60000 → 51685 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=5
7	4.094057	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=6 Ack=6 Win=2619648 Len=0
8	4.094253	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [FIN, ACK] Seq=6 Ack=6 Win=2619648 Len=0
9	4.094271	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=6 Ack=7 Win=2619648 Len=0
10	4.094441	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [FIN, ACK] Seq=6 Ack=7 Win=2619648 Len=0
11	4.094472	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=7 Ack=7 Win=2619648 Len=0

Após essa etapa o cliente envia através do seu socket um pacote TCP de 5 bytes para o servidor. Esses dados são uma mensagem com a *string* teste conforme pode ser visto abaixo.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	51685 → 60000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
2	0.000057	127.0.0.1	127.0.0.1	TCP	56	60000 → 51685 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
3	0.000094	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	4.093925	127.0.0.1	127.0.0.1	TCP	49	51685 → 60000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=5
5	4.093961	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
6	4.094037	127.0.0.1	127.0.0.1	TCP	49	60000 → 51685 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=5
7	4.094057	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=6 Ack=6 Win=2619648 Len=0
8	4.094253	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [FIN, ACK] Seq=6 Ack=6 Win=2619648 Len=0
9	4.094271	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=6 Ack=7 Win=2619648 Len=0
10	4.094441	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [FIN, ACK] Seq=6 Ack=7 Win=2619648 Len=0
11	4.094472	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=7 Ack=7 Win=2619648 Len=0

Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 2148819016
0101 = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window size value: 10233
[Calculated window size: 2619648]
[Window size scaling factor: 256]
Checksum: 0x2ce4 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
> [SEQ/ACK analysis]
> [Timestamps]
TCP payload (5 bytes)
Data (5 bytes)
Data: 7465737465
[Length: 5]

No.	Time	Source	Destination	Protocol	Length	Info
0000	02.00.00.00 45.00.00.0d	2a 1c 40 00 80 00 00E.... * @.....			
0010	7f 00 00 01 7f 00 00 01	c9 e5 ea 60 e2 ad 98 bc			
0020	80 14 60 48 50 18 27 f9	2c e4 00 00 74 65 73 74	e HP- ,...test			
0030	65					

O servidor por sua vez, através do seu socket envia um pacote TCP para o cliente com a flag ACK, confirmando o recebimento da mensagem.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	51685 → 60000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
2	0.000057	127.0.0.1	127.0.0.1	TCP	56	60000 → 51685 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
3	0.000094	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	4.093925	127.0.0.1	127.0.0.1	TCP	49	51685 → 60000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=5
5	4.093961	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
6	4.094037	127.0.0.1	127.0.0.1	TCP	49	60000 → 51685 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=5
7	4.094057	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=6 Ack=6 Win=2619648 Len=0
8	4.094253	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [FIN, ACK] Seq=6 Ack=6 Win=2619648 Len=0
9	4.094271	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=6 Ack=7 Win=2619648 Len=0
10	4.094441	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [FIN, ACK] Seq=6 Ack=7 Win=2619648 Len=0
11	4.094472	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=7 Ack=7 Win=2619648 Len=0

> Frame 5: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface \Device\NPF_Loopback, id 0
 > Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > Transmission Control Protocol, Src Port: 60000, Dst Port: 51685, Seq: 1, Ack: 6, Len: 0
 Source Port: 60000
 Destination Port: 51685
 [Stream index: 0]
 [TCP Segment Len: 0]
 Sequence number: 1 (relative sequence number)
 Sequence number (raw): 2148819016
 [Next sequence number: 1 (relative sequence number)]
 Acknowledgment number: 6 (relative ack number)
 Acknowledgment number (raw): 3803027649
 0101 = Header Length: 20 bytes (5)
 > Flags: 0x010 (ACK)
 Window size value: 10233

```
0000 02 00 00 00 45 00 00 28 2a 1d 40 00 80 06 00 00  ....E.. (*@....  

0010 7f 00 00 01 7f 00 00 01 ea 60 c9 e5 80 14 60 48  .....H  

0020 e2 ad 98 c1 50 10 27 f9 79 c6 00 00  ....P... y....
```

Em seguida o servidor envia uma mensagem de volta para o cliente com 5 bytes de tamanho.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	51685 → 60000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
2	0.000057	127.0.0.1	127.0.0.1	TCP	56	60000 → 51685 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
3	0.000094	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	4.093925	127.0.0.1	127.0.0.1	TCP	49	51685 → 60000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=5
5	4.093961	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
6	4.094037	127.0.0.1	127.0.0.1	TCP	49	60000 → 51685 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=5
7	4.094057	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=6 Ack=6 Win=2619648 Len=0
8	4.094253	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [FIN, ACK] Seq=6 Ack=6 Win=2619648 Len=0
9	4.094271	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=6 Ack=7 Win=2619648 Len=0
10	4.094441	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [FIN, ACK] Seq=6 Ack=7 Win=2619648 Len=0
11	4.094472	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=7 Ack=7 Win=2619648 Len=0

Acknowledgment number: 6 (relative ack number)
 Acknowledgment number (raw): 3803027649
 0101 = Header Length: 20 bytes (5)
 > Flags: 0x018 (PSH, ACK)
 Window size value: 10233
 [Calculated window size: 2619648]
 [Window size scaling factor: 256]
 Checksum: 0x8df [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 > [SEQ/ACK analysis]
 > [Timestamps]
 TCP payload (5 bytes)
 > Data (5 bytes)
 Data: 5445535445
 [Length: 5]

```
0000 02 00 00 00 45 00 00 2a 1e 40 00 80 06 00 00  ....E.. (*@....  

0010 7f 00 00 01 7f 00 00 01 ea 60 c9 e5 80 14 60 48  .....H  

0020 e2 ad 98 c1 50 10 27 f9 8d 1f 00 00 54 45 53 54  ....P... TEST  

0030 45 E
```

O cliente então responde confirmando o recebimento da mensagem através do envio de um pacote com a flag ACK.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	51685 → 60000 [SYN] Seq=0 Win=65535 MSS=65495 WS=256 SACK_PERM=1
2	0.000057	127.0.0.1	127.0.0.1	TCP	56	60000 → 51685 [SYN, ACK] Seq=0 Ack=1 Win=65535 MSS=65495 WS=256 SACK_PERM=1
3	0.000094	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	4.093925	127.0.0.1	127.0.0.1	TCP	49	51685 → 60000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=5
5	4.093961	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
6	4.094037	127.0.0.1	127.0.0.1	TCP	49	60000 → 51685 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=5
7	4.094057	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=6 Ack=6 Win=2619648 Len=0
8	4.094253	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [FIN, ACK] Seq=6 Ack=6 Win=2619648 Len=0
9	4.094271	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=6 Ack=7 Win=2619648 Len=0
10	4.094441	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [FIN, ACK] Seq=6 Ack=7 Win=2619648 Len=0
11	4.094472	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=7 Ack=7 Win=2619648 Len=0

```
> Frame 7: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 51685, Dst Port: 60000, Seq: 6, Ack: 6, Len: 0
```

```
0000 02 00 00 00 45 00 00 28 2a 1f 40 00 80 06 00 00  ...E-(*:@...
0010 7f 00 00 01 7f 00 00 01 c9 e5 ea 60 e2 ad 98 c1  ...
0020 80 14 60 4d 50 10 27 f9 79 c1 00 00 00 00 00 00 00  ...MP-`y...
```

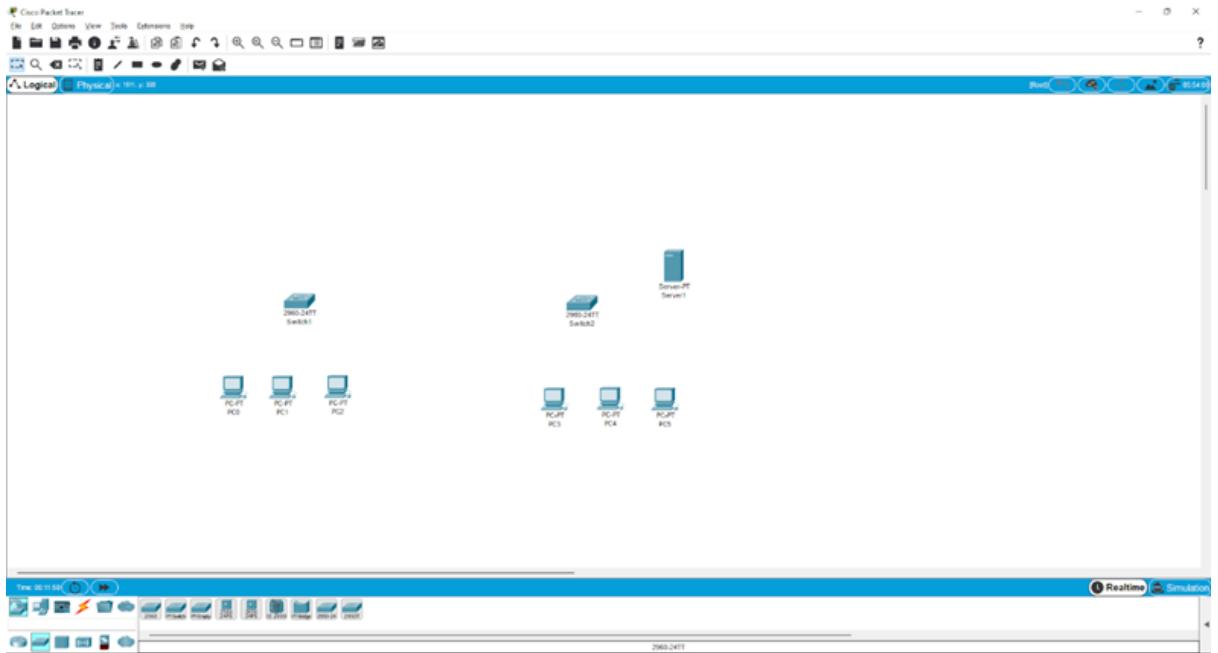
O processo de encerramento da conexão se dá nas etapas seguintes através do envio de pacotes com as flags FYN,ACK e pacotes com a flag ACK, trocados entre cliente e servidor, conforme pode ser visto abaixo:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	51685 → 60000 [SYN] Seq=0 Win=65535 MSS=65495 WS=256 SACK_PERM=1
2	0.000057	127.0.0.1	127.0.0.1	TCP	56	60000 → 51685 [SYN, ACK] Seq=0 Ack=1 Win=65535 MSS=65495 WS=256 SACK_PERM=1
3	0.000094	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	4.093925	127.0.0.1	127.0.0.1	TCP	49	51685 → 60000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=5
5	4.093961	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=1 Ack=6 Win=2619648 Len=0
6	4.094037	127.0.0.1	127.0.0.1	TCP	49	60000 → 51685 [PSH, ACK] Seq=1 Ack=6 Win=2619648 Len=5
7	4.094057	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=6 Ack=6 Win=2619648 Len=0
8	4.094253	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [FIN, ACK] Seq=6 Ack=6 Win=2619648 Len=0
9	4.094271	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [ACK] Seq=6 Ack=7 Win=2619648 Len=0
10	4.094441	127.0.0.1	127.0.0.1	TCP	44	60000 → 51685 [FIN, ACK] Seq=6 Ack=7 Win=2619648 Len=0
11	4.094472	127.0.0.1	127.0.0.1	TCP	44	51685 → 60000 [ACK] Seq=7 Ack=7 Win=2619648 Len=0

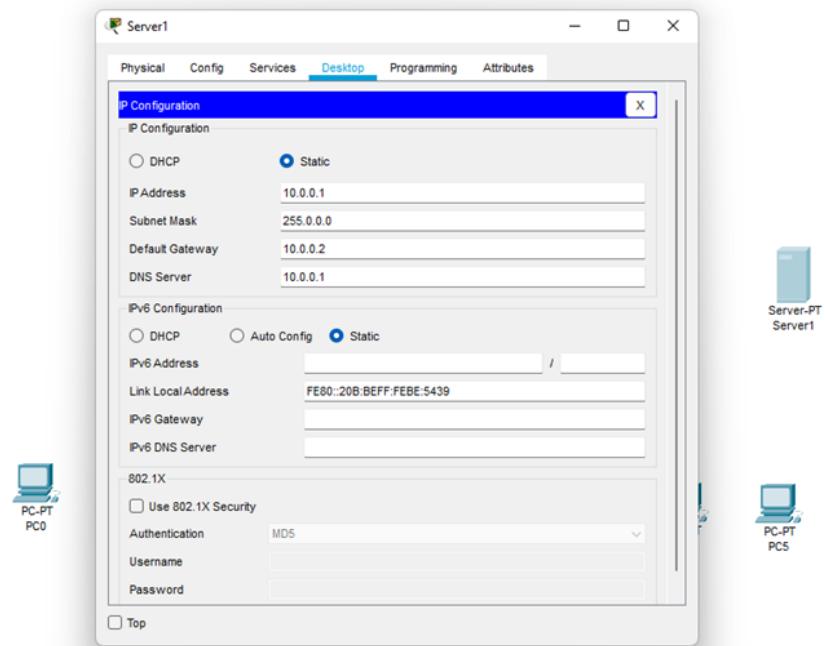
Aula prática II

Nessa aula, vamos configurar para a criação do Cascatamento de Switch e o Server - DHCP

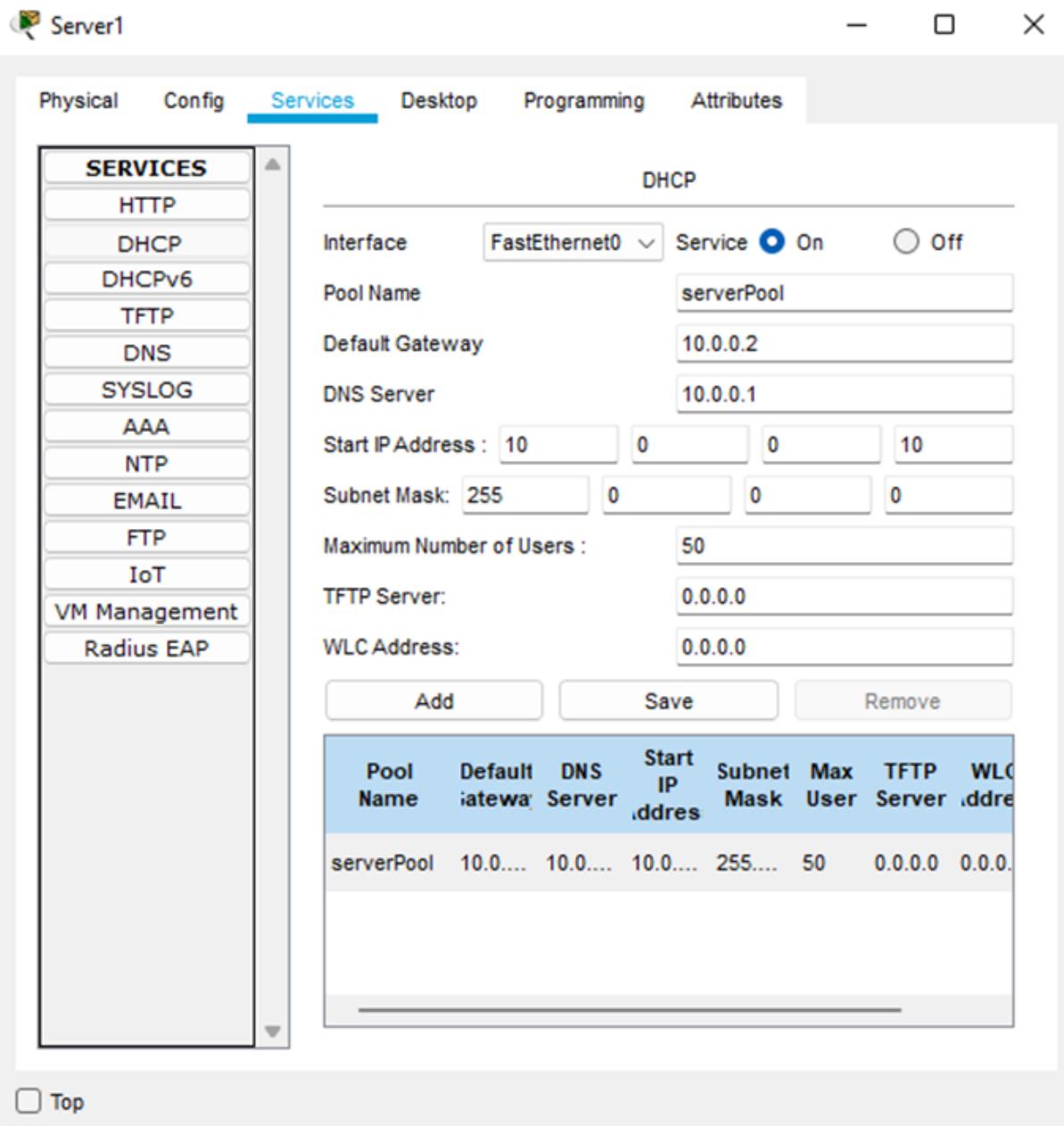
1. Faça a organização do seu ambiente de trabalho de acordo com a figura abaixo



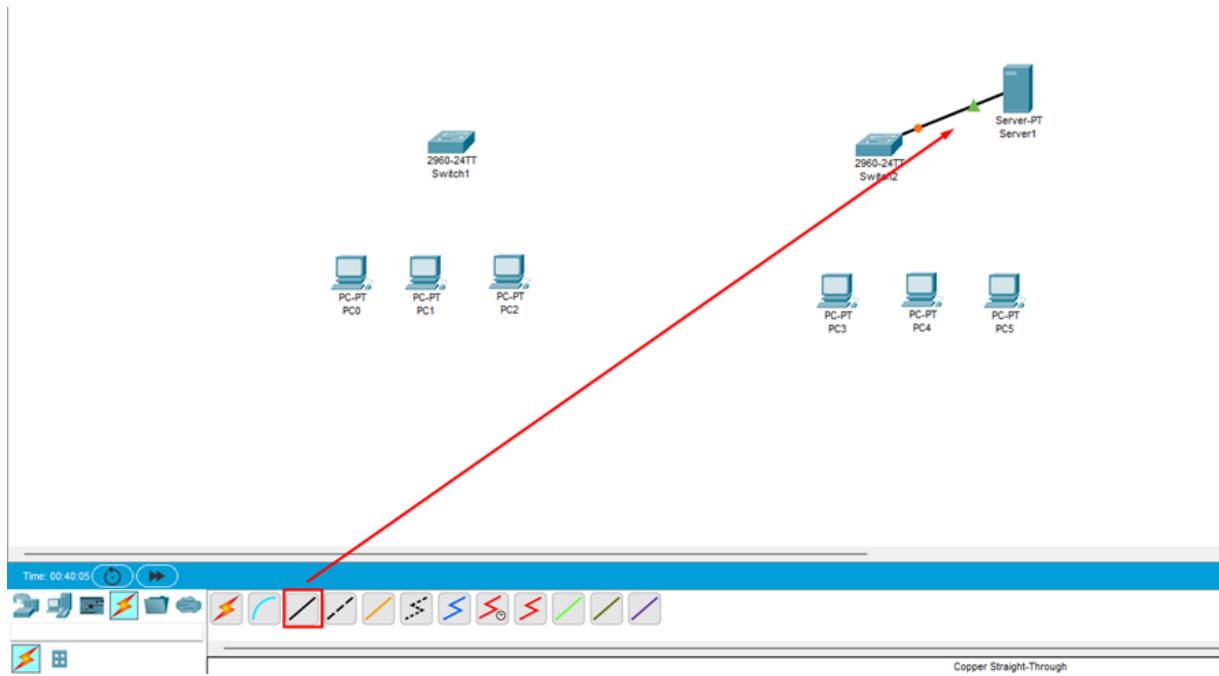
2. Clique duas vezes em Server 1, faça o acesso do Desktop e IP Configuration pois precisamos configurar o IP para que seja estático



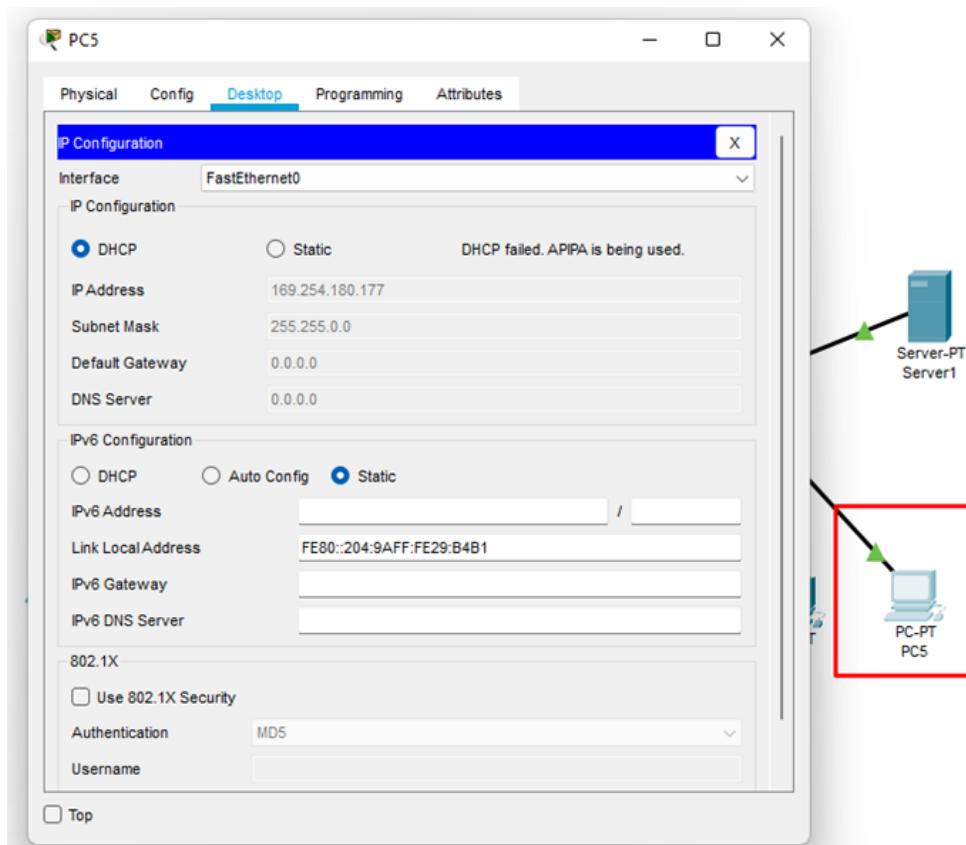
3. Faça o acesso de Services > DHCP, deixe as configurações igual da imagem e clique em “Save” para salvar as configurações



4. Selecione o cabo mostrado na imagem e ligue o Server no Switch(porta 23)



- Faça as conexões das estações de trabalho (PC) no Switch utilizando o mesmo cabo que foi utilizado no item anterior. Acessar cada um dos PCs e entrar Desktop > IP Configuration e mudar para DHCP



6. Faça a abertura do Command Prompt do PC3, veja o IP correto e utilize o PING para o PC5

The screenshot shows a Cisco Packet Tracer interface. At the top, there's a menu bar with tabs: Physical, Config, Desktop (which is selected), Programming, and Attributes. Below the menu is a toolbar with icons for copy, paste, and other functions. The main area is a terminal window titled "Command Prompt". The terminal output is as follows:

```
Packet Tracer PC Command Line 1.0
C:\>PING 10.0.0.10

Pinging 10.0.0.10 with 32 bytes of data:

Reply from 10.0.0.10: bytes=32 time=2ms TTL=128
Reply from 10.0.0.10: bytes=32 time=1ms TTL=128
Reply from 10.0.0.10: bytes=32 time<1ms TTL=128
Reply from 10.0.0.10: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 0ms

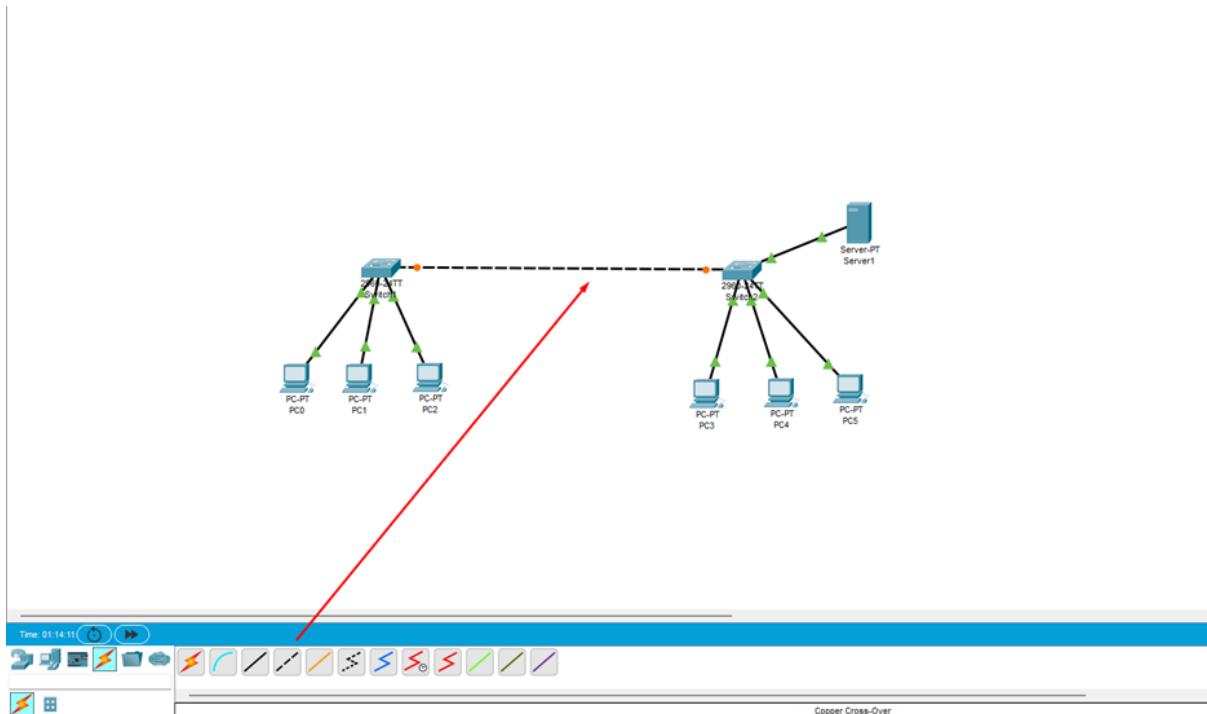
C:\>
```

At the bottom left of the terminal window, there's a checkbox labeled "Top".

7. Conecte no Switch da esquerda.



8. Faça a conexão dos dois switchs na porta 24(cascatamento), depois que a conexão for feita (pontos laranja se tornarem verde), faça a configuração do DHCP no PC0, PC1 e PC2.



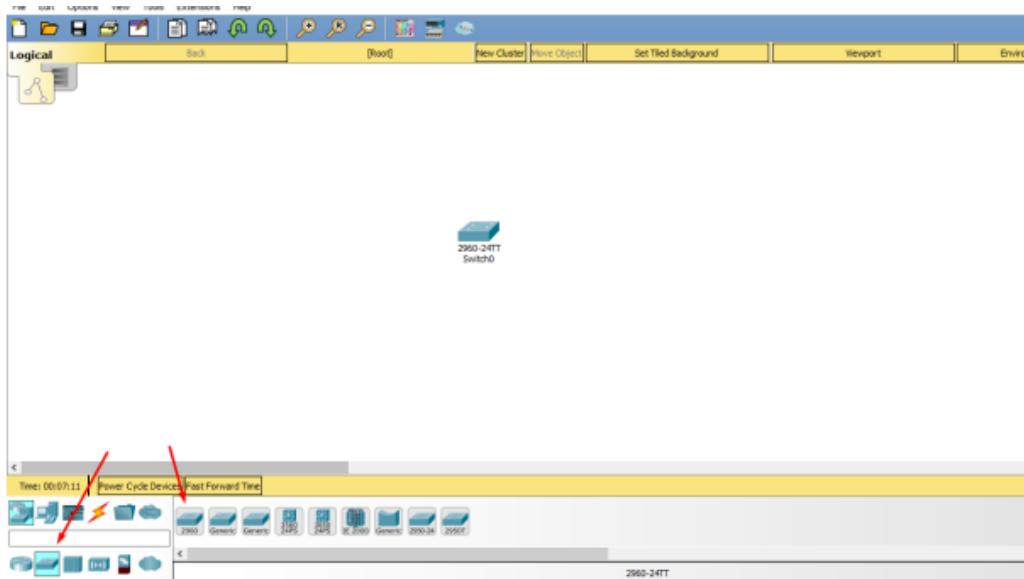
9. Finalize testando a rede utilizando o ping de qualquer PC conectado no Switch da esquerda em algum PC conectado na Switch da direita.

Aula prática III

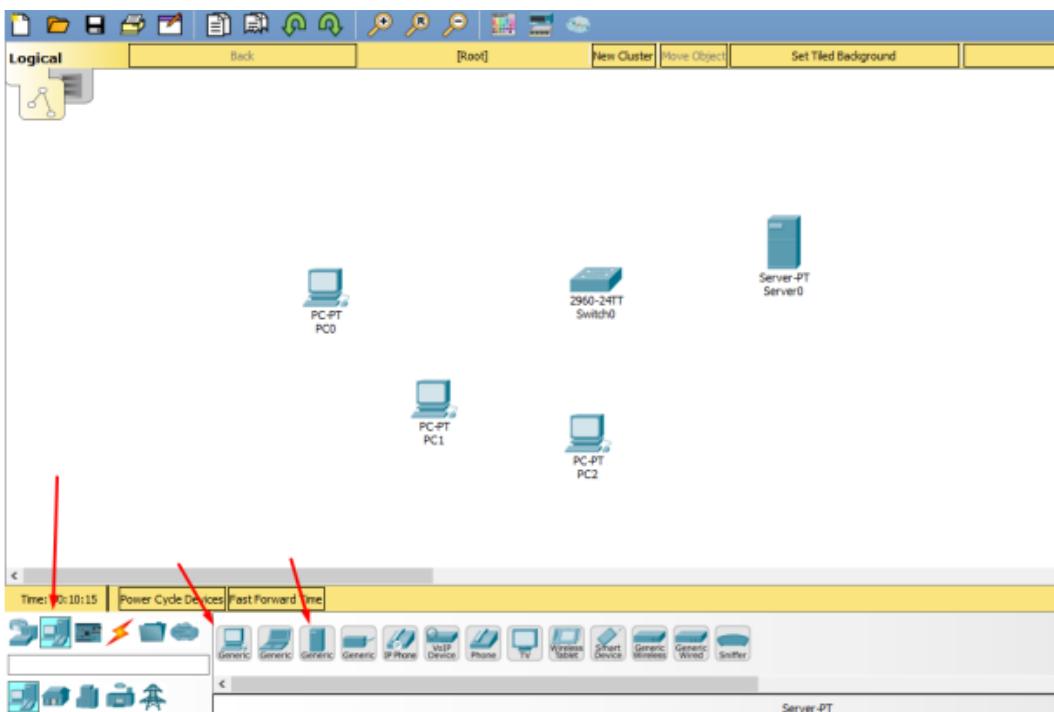
Nessa aula, iremos utilizar o Cisco Packet Tracer para montar uma rede simples para nos familiarizar com os componentes básicos de uma rede e suas configurações.

Essa rede possuirá 3 computadores, 1 switch e 1 servidor que vai servir unicamente como um servidor web, que vai servir uma página web para os computadores da rede:

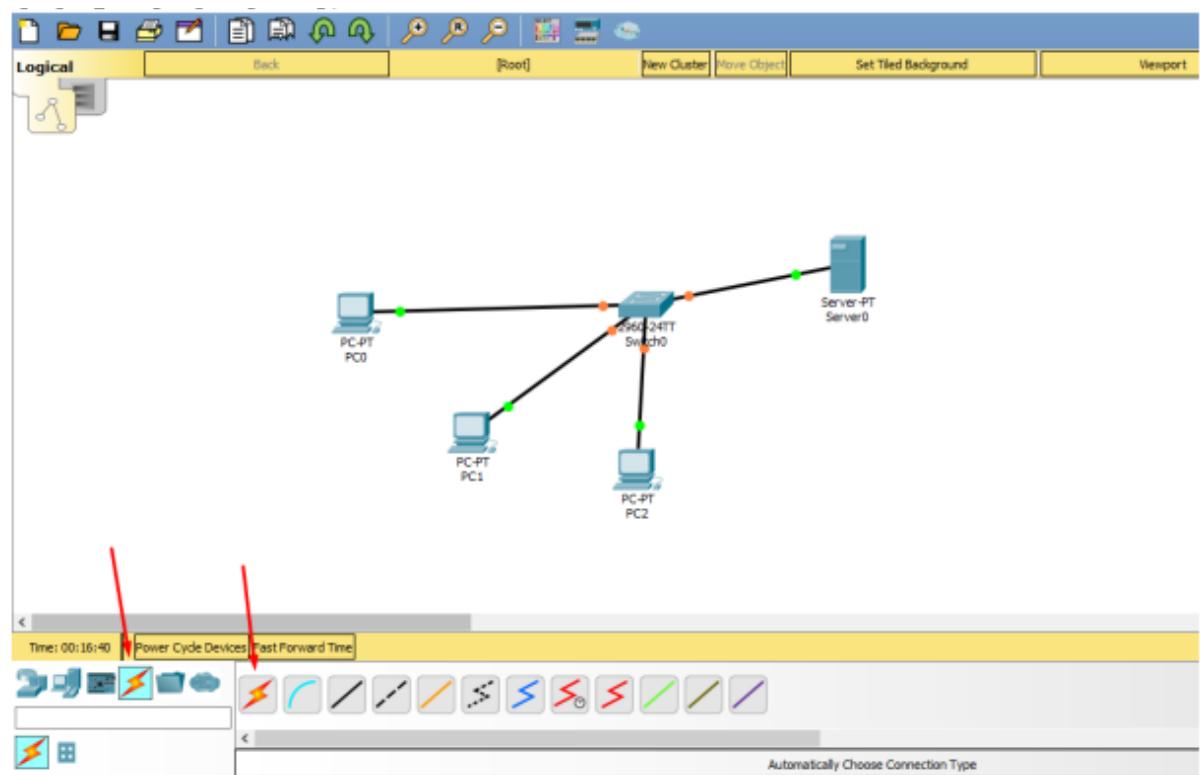
1. Primeiramente, selecione no canto inferior esquerdo a opção de “switches”, e selecione o 2960.



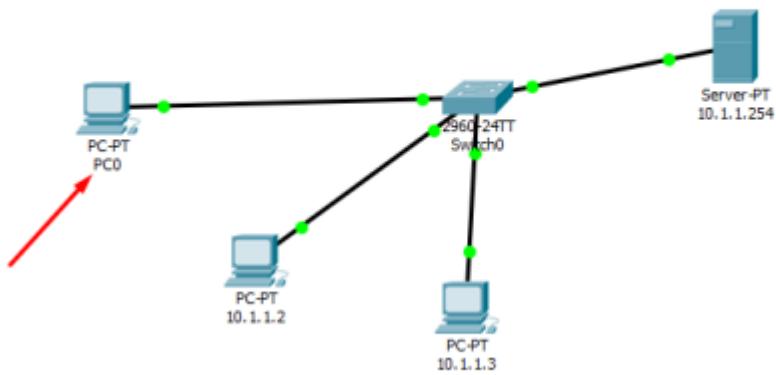
2. Após isso, clique na opção de “end devices”, e adicione os 3 computadores e o servidor.



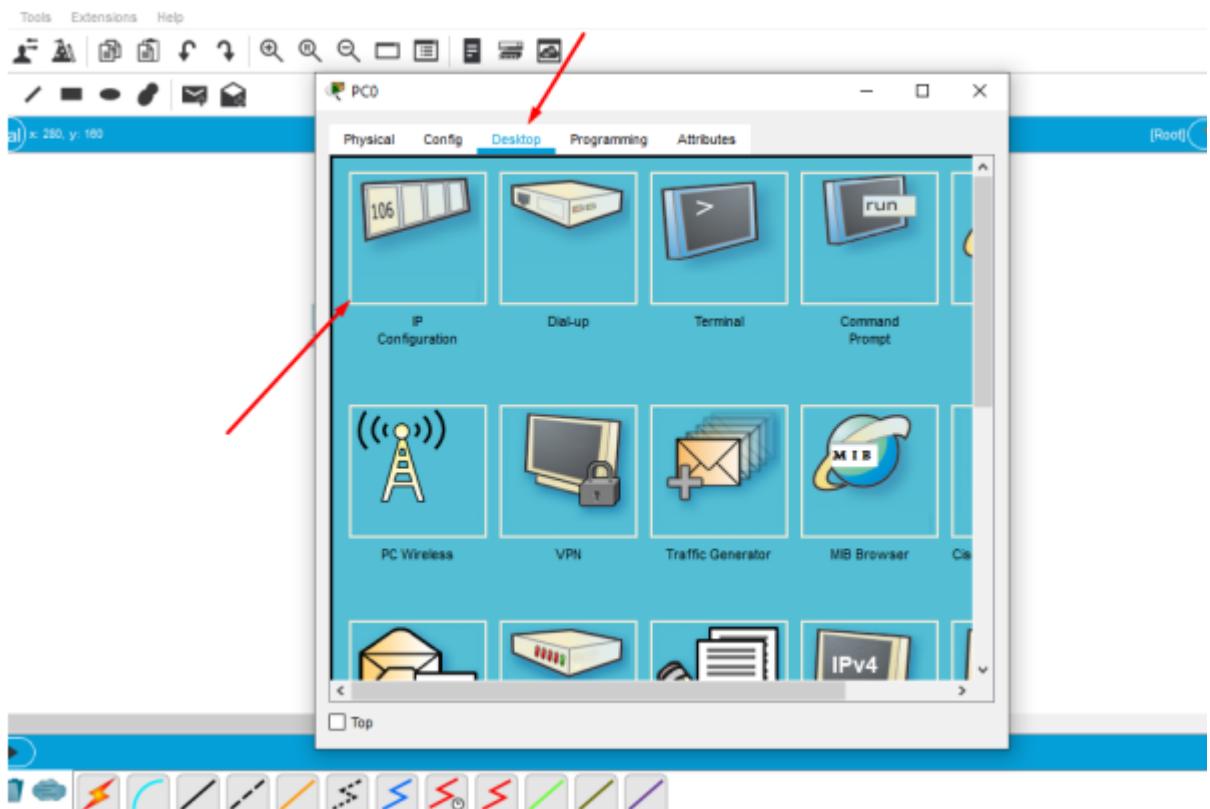
3. Em seguida, clique na opção de connections e conecte todos os “end devices” ao “switch”, como visto abaixo.



4. Nessa nomeamos os “end devices”, neste caso daremos como nome o valor do endereço de IP que atribuímos a cada um deles mais a frente para facilitar a visualização. Basta clicar no texto abaixo da figura do device que abrirá uma caixa de texto para que o nome seja alterado. A esse primeiro computador, daremos o endereço IP de 10.0.0.1, então sugerimos que esse seja também o nome dado a ele.

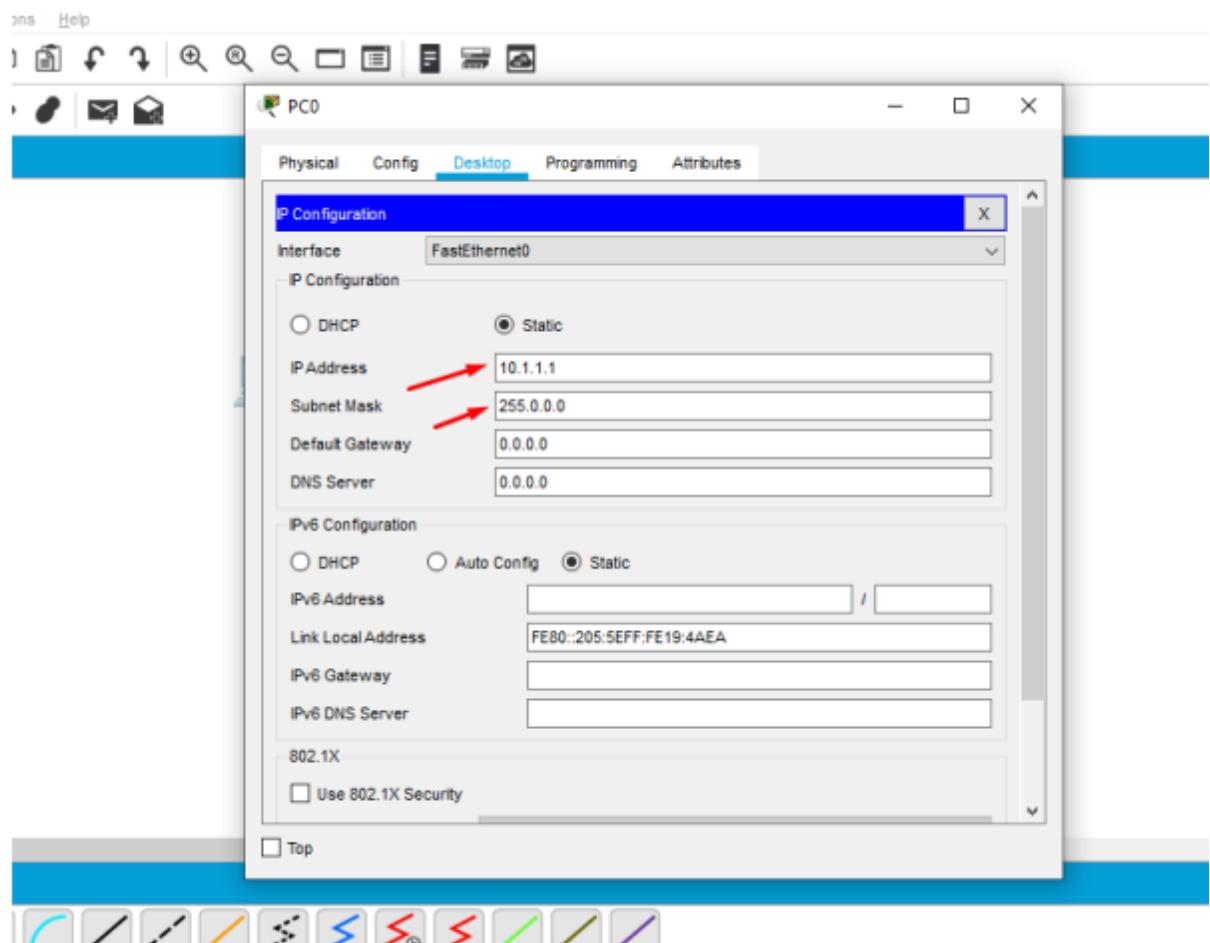


5. Agora, para que esses devices sejam encontrados na rede, é necessário que todos tenham um endereço de IP. Nesta etapa vamos mostrar como atribuir os endereços de IP para os computadores e o servidor Primeiro, vamos clicar no primeiro computador e selecionar a aba de “Desktop” e clicar em “IP Configuration”.

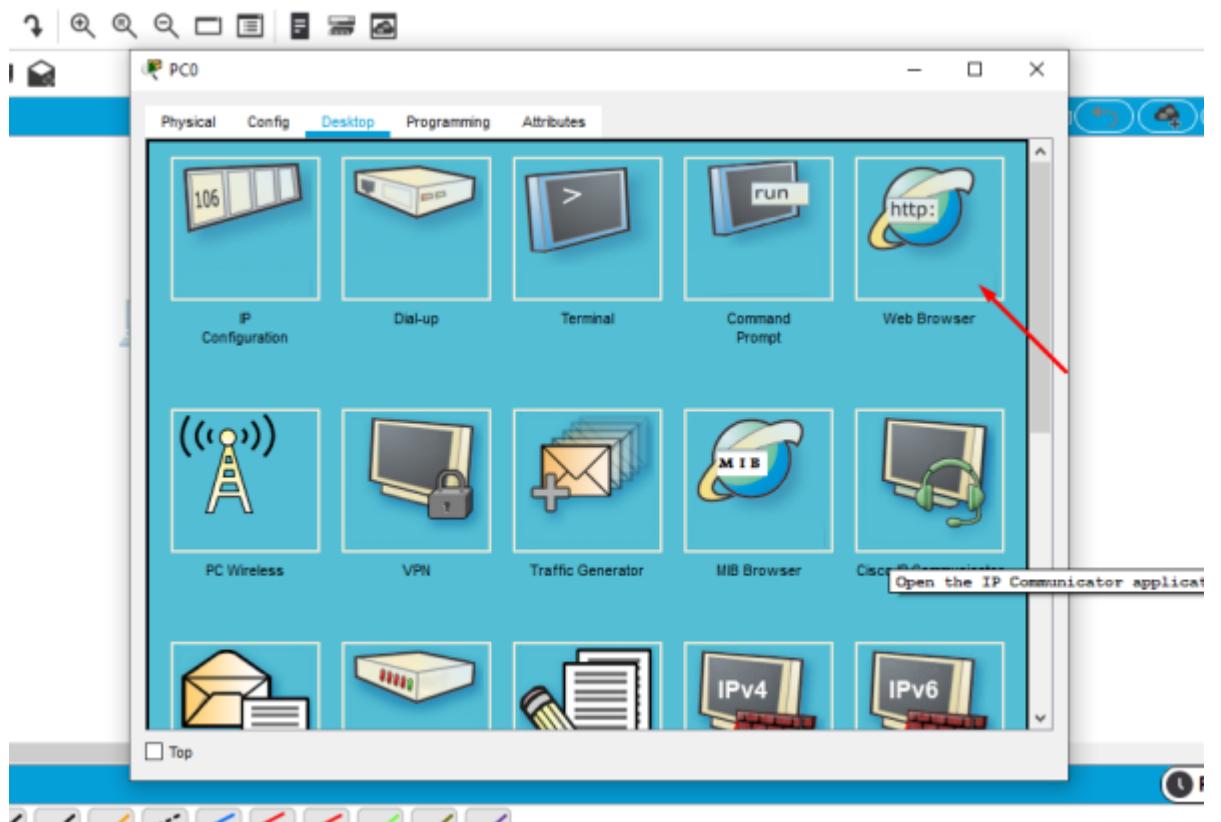


6. Nessa etapa, adicionamos o endereço de IP no campo de IP Address, lembre-se que estamos utilizando o mesmo nome do computador para o endereço de IP. Após preencher o campo de “IP Address”, basta clicar no

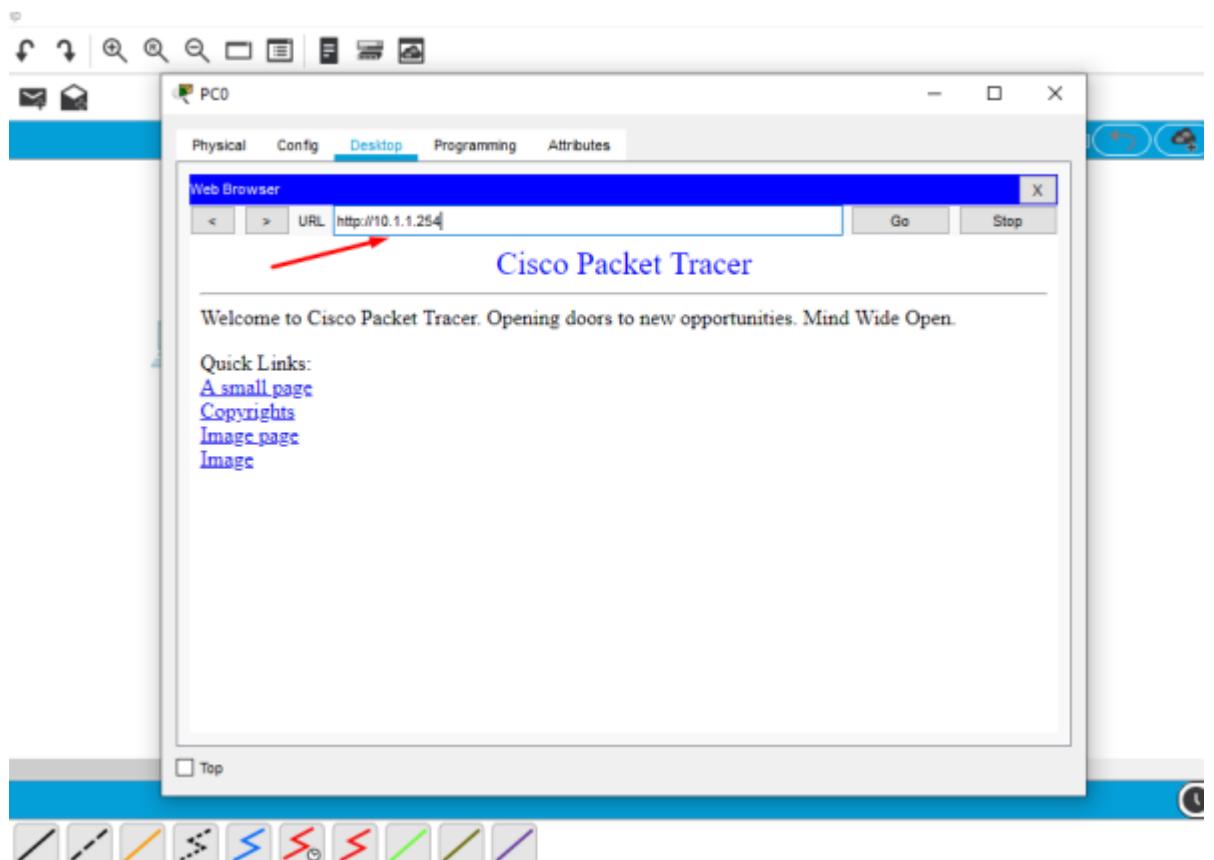
campo de “Subnet Mask” que ele será preenchido automaticamente com o valor da imagem.



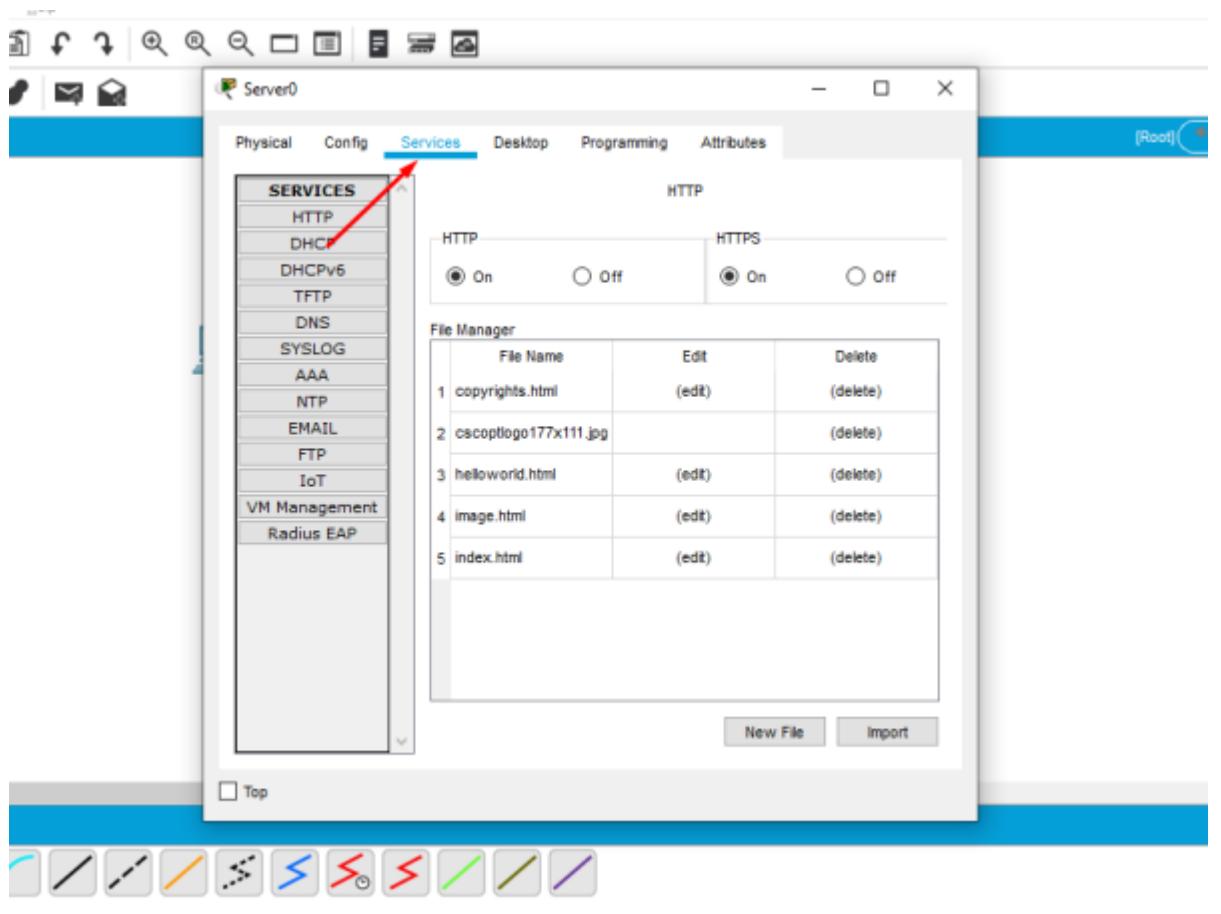
7. Repita nos outros 2 computadores, e com o servidor, preenchendo com seus respectivos endereços de IP. Lembrando que o endereço de IP do servidor é o 10.1.1.254.
8. Feito isso, vamos agora testar que o nosso servidor web funciona no software, clique em algum dos computadores e selecione a opção web browser.



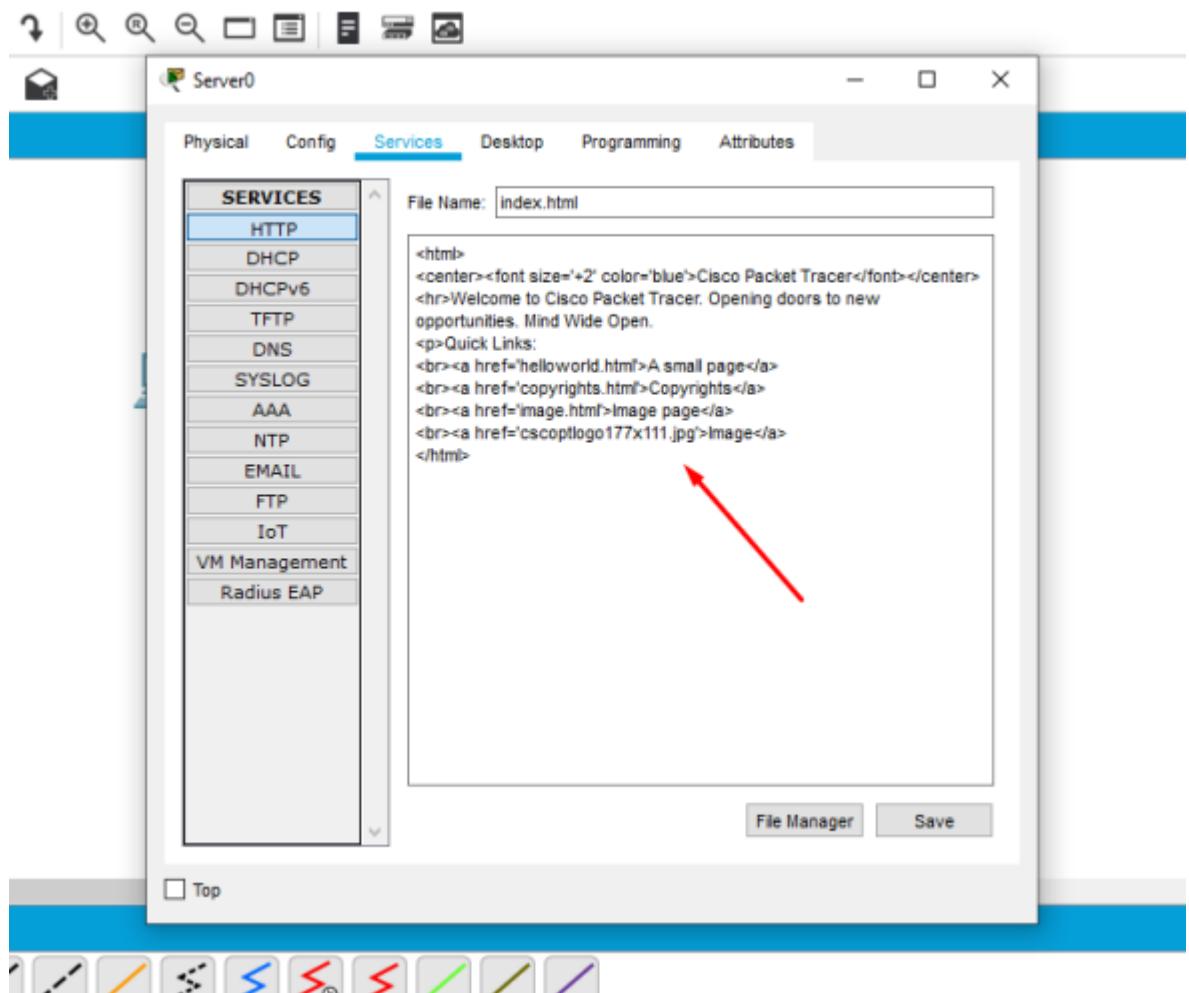
9. Agora digitamos o endereço de IP do servidor na URL e pressionamos enter. Veja que um site é exibido na tela, esse site está definido no nosso servidor.



Vamos então agora fechar essa janela relacionada ao computador e vamos clicar no nosso servidor, e clicar na aba de “Services”.



11. Repare que nas opções temos alguns arquivos definidos. O arquivo que abre quando digitamos o endereço de IP do servidor na URL é o “index.html”. Basta então clicar no botão (editar) do “index.html” e veremos um código em HTML, que é justamente o código que apareceu para nós quando acessamos do browser e o que ele interpreta.

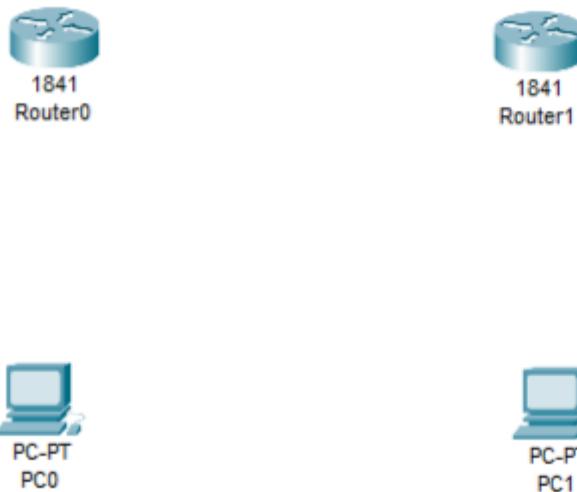


12. Faça mudanças nesse arquivo HTML e volte ao browser de um dos computadores e acesse novamente o endereço do servidor para ver as mudanças na tela.

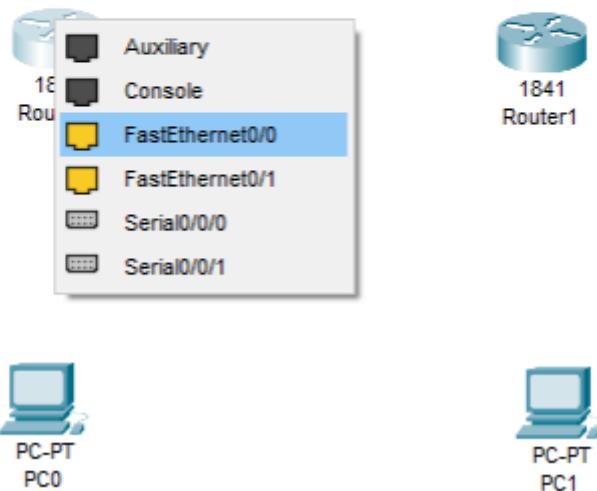
Aula prática IV

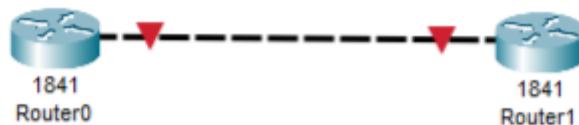
Nesta aula, vamos configurar o roteamento estático usando IPv6. Usaremos dois computadores e dois roteadores.

1. Primeiro, selecionaremos os dispositivos, dois roteadores 1841 e dois computadores PCs.

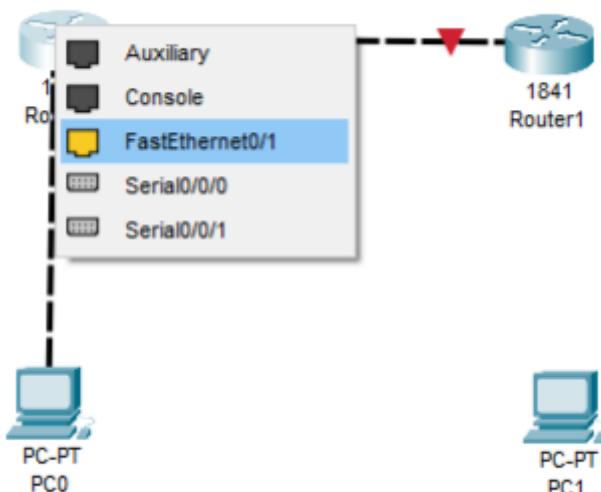
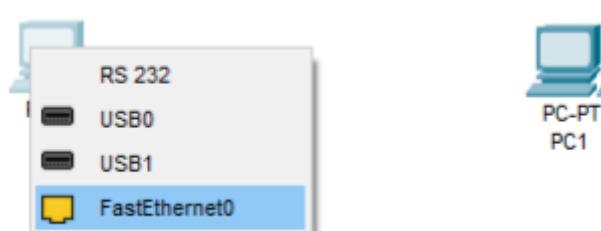
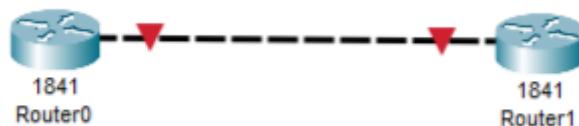


2. Em seguida, conectamos os dois roteadores usando um cabo crossover na opção FastEthernet0/0.

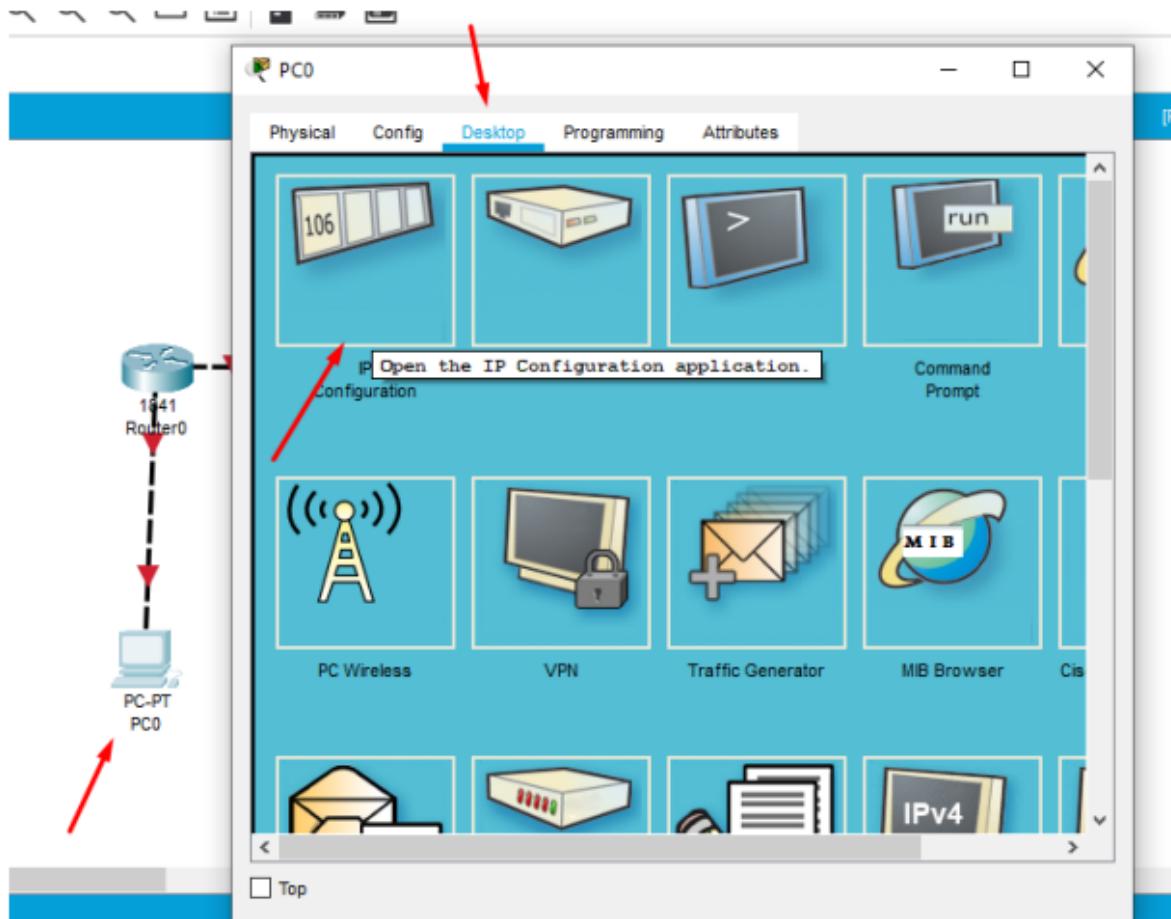




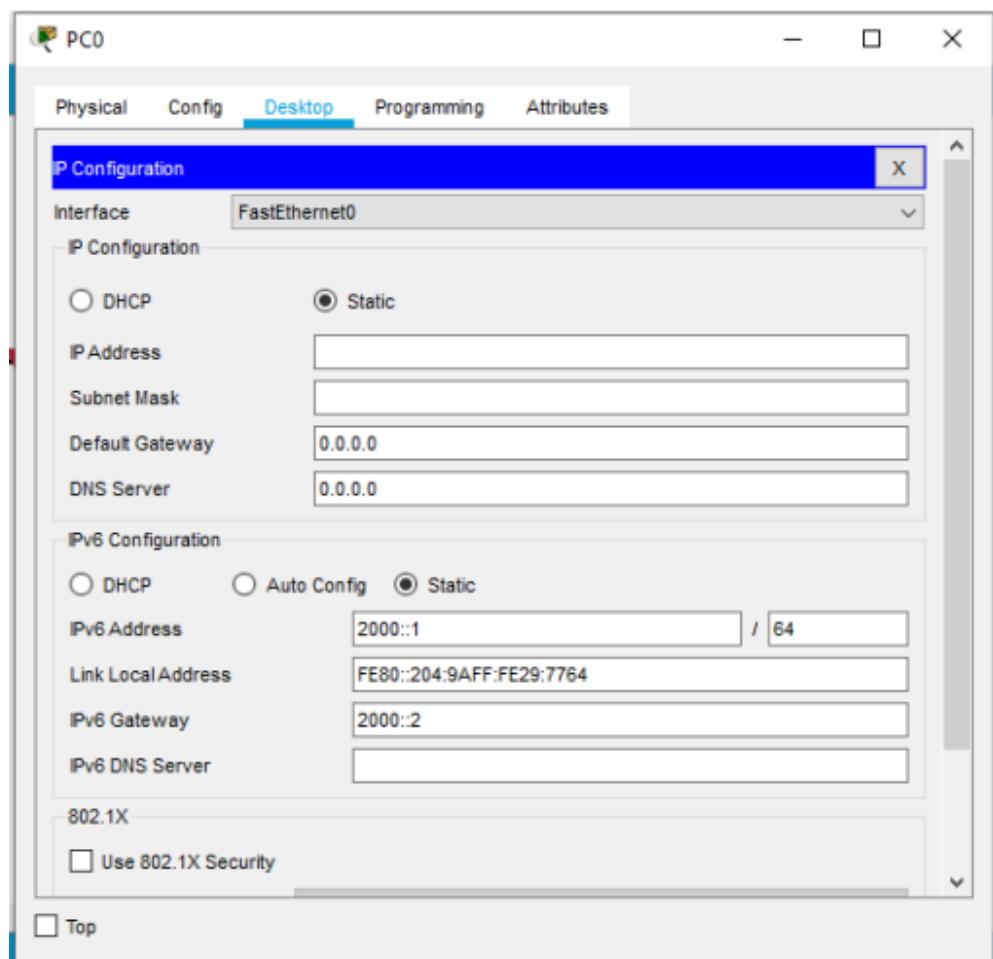
3. Então, será necessário conectar os PCs aos roteadores usando o cabo crossover e a opção FastEthernet, da mesma forma como conectamos os roteadores.



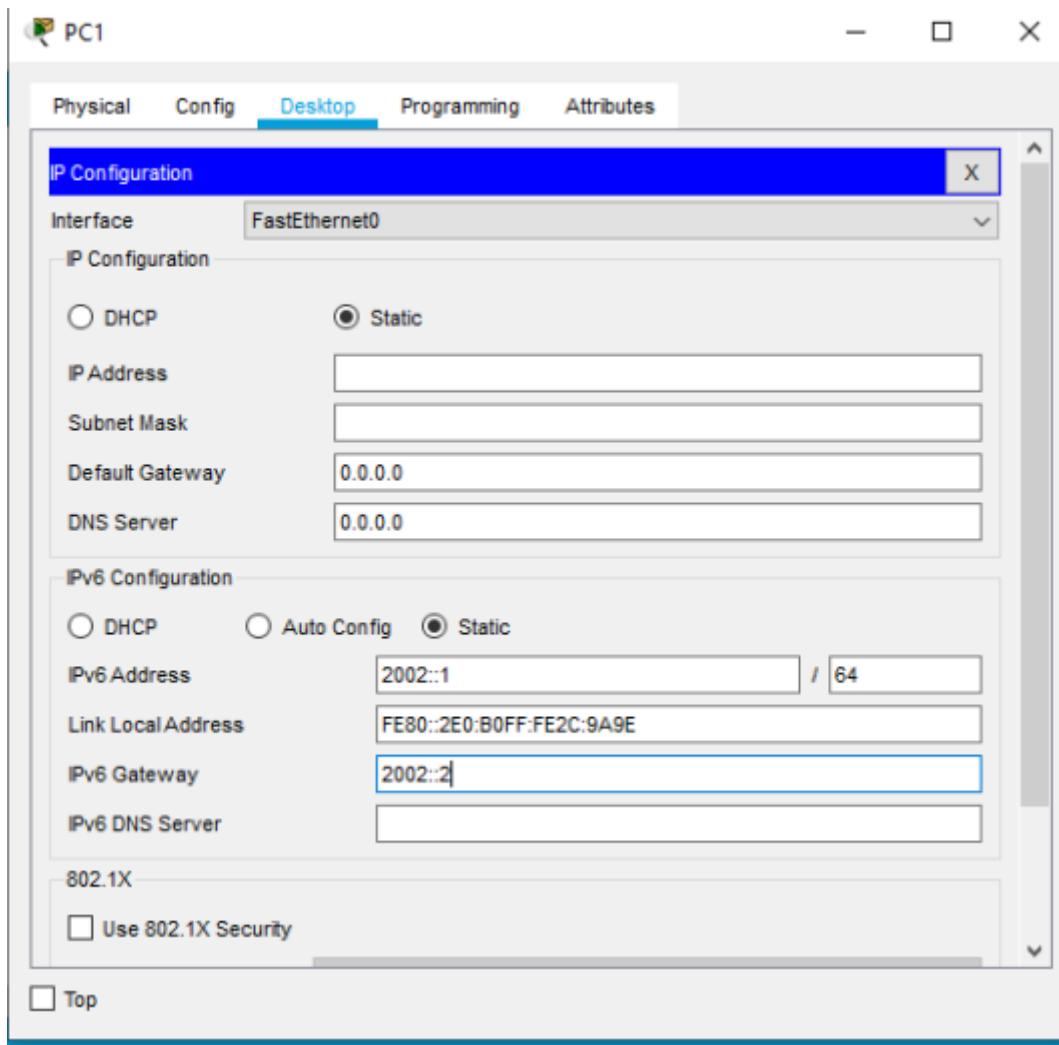
4. Nesta etapa, realizaremos a configuração do IPv6 para o PC0. Para isso, selecione o PC0, escolha a sessão desktop e clique na opção IP configuration.



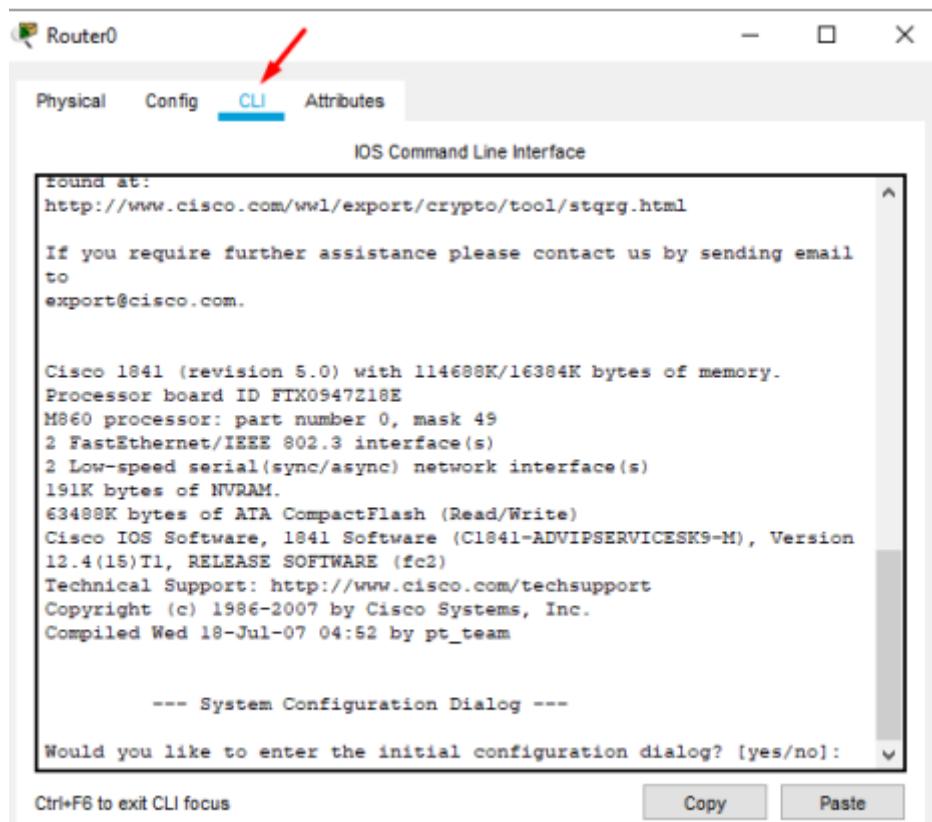
5. Na opção de *IPv6 address*, usaremos o endereço 2000::1. Os dois caracteres de dois pontos representam que preenchemos todo o meio do endereço IPv6 com zeros. O endereço completo seria 2000:0000:0000:0001. Após a barra, preenchemos com 64. No caso do *IPv6 gateway*, preenchemos com 2000::2.



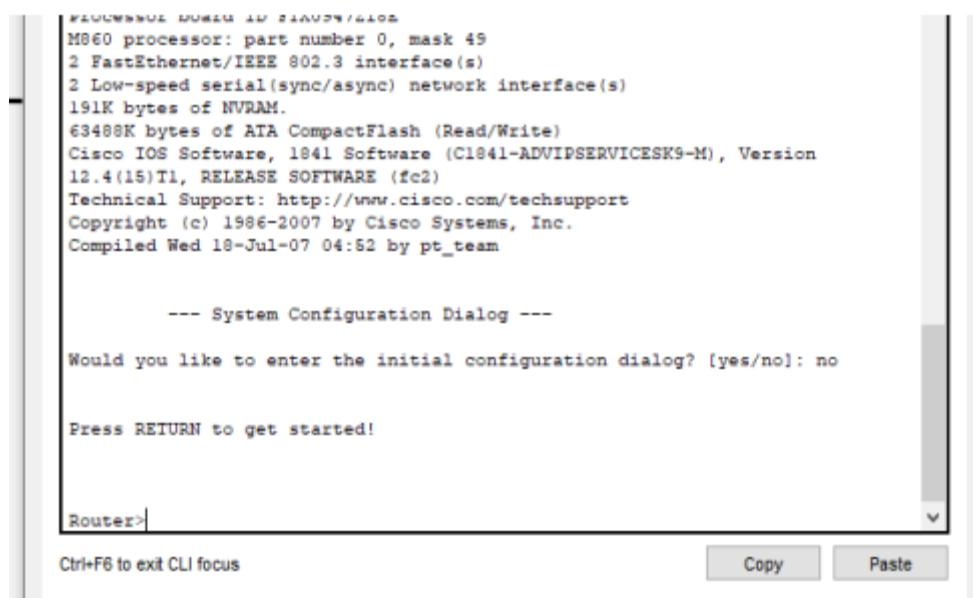
6. No PC1, faremos a mesma coisa, alterando apenas o valor 2000 para 2002.



7. Agora, podemos configurar os roteadores. Selecione o Router0 primeiro e escolha a sessão CLI. Então, executaremos alguns comandos no terminal. Executaremos a seguinte sequência de comandos:

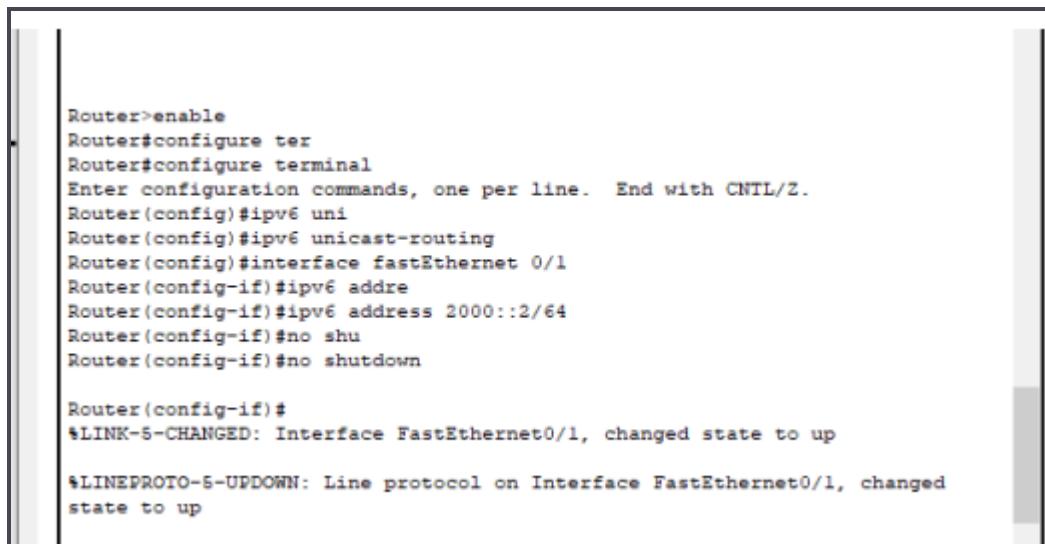


Agora, é necessário escrever *no* e pressionar enter duas vezes.



Agora, executamos a seguinte sequência de comandos:

```
> enable  
> configure terminal  
> ipv6 unicast-routing  
> interface fastEthernet 0/1  
> ipv6 address 2000::2/64  
> no shutdown
```



```
Router>enable  
Router#configure ter  
Router#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#ipv6 uni  
Router(config)#ipv6 unicast-routing  
Router(config)#interface fastEthernet 0/1  
Router(config-if)#ipv6 addre  
Router(config-if)#ipv6 address 2000::2/64  
Router(config-if)#no shu  
Router(config-if)#no shutdown  
  
Router(config-if)#  
*LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up  
  
*LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed  
state to up
```

Pressione enter e continuaremos com a configuração:

```
> exit  
> interface fastEthernet 0/0  
> ipv6 address 2001::1/64  
> no shutdown  
> exit  
> ipv6 route 2002::/64 2001::2
```

```
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up

Router(config-if)#exit
Router(config)#interface fastEthernet 0/0
Router(config-if)#ipv6 address 2001::1/64
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#ipv6 route 2002::/64 2001::2
Router(config)#

```

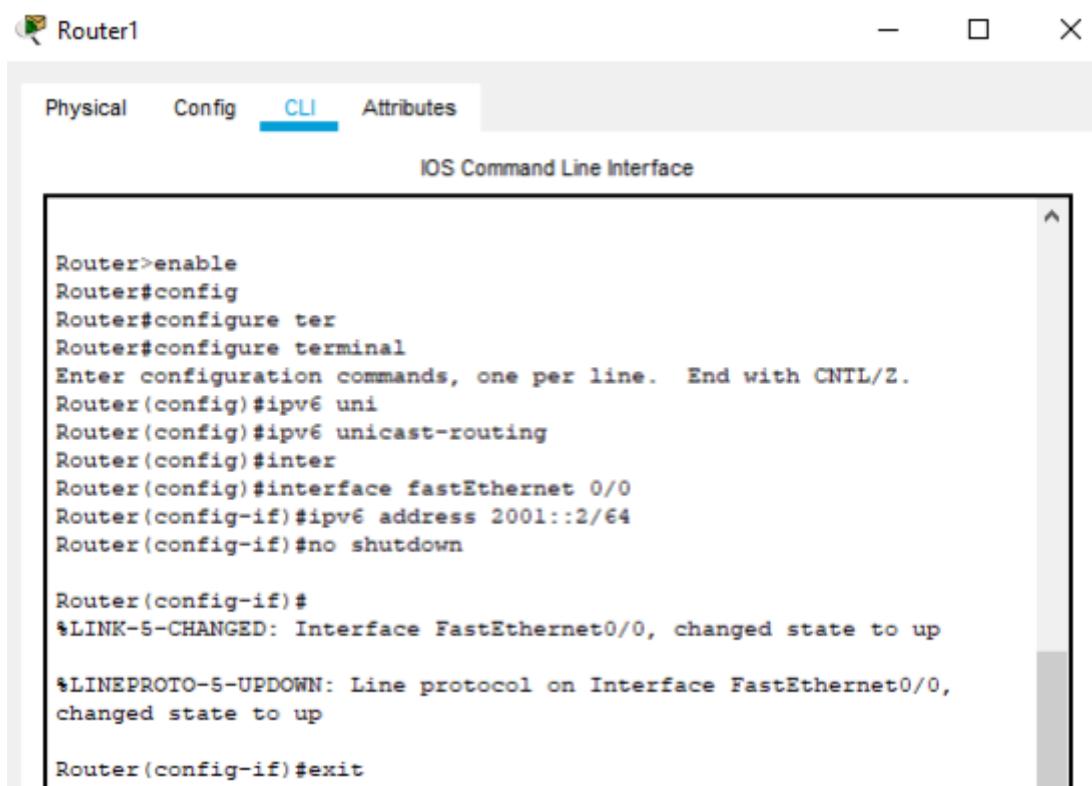
Ctrl+F6 to exit CLI focus

Copy

Paste

Após isso, podemos fechar essa janela

8. Agora, clicamos no router1, e realizaremos a mesma configuração, trocando apenas os valores dos comandos de *ipv6 address* e *ipv6 route*



Nesse caso, o *ipv6 address* será 2001::2/64

```

Router(config)#interface fastEthernet 0/1
Router(config-if)#ipv6 address 2002::2/64
Router(config-if)#no shutdown

Router(config-if)#
LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

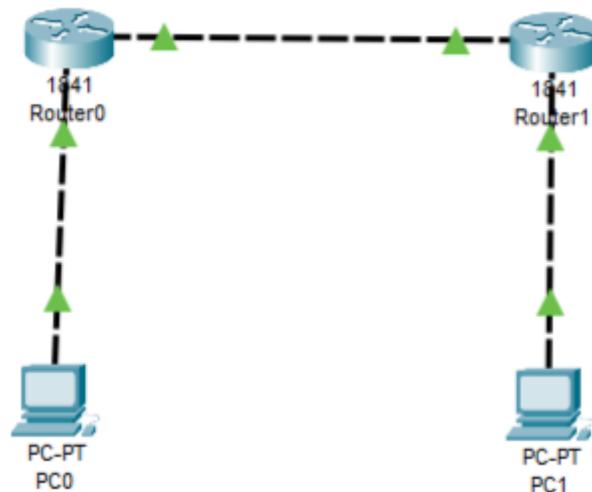
LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to up

Router(config-if)#exit
Router(config)#ipv6 route 2000::/64 2001::1
Router(config)#

```

e o ipv6 route será 2000::/64 2001::1

Após isso podemos fechar a janela, e neste momento já devemos ver que os triângulos nos cabos estão verdes, o que significa que foi tudo conectado de forma correta



9. A partir desse momento, já é possível realizar a comunicação entre os PCs, você pode fazer isso acessando o terminal de um PC e realizando um *ping* para o outro, como já fizemos em aulas anteriores.

Aula prática V

Nesta aula, iremos realizar a construção de uma rede mais complexa envolvendo servidor de DNS, servidor DHCP, servidores Web e configuração de roteadores e subredes. Os itens requisitos para a definição dessa rede são:

- Quatro sub-redes (A, B, C e D) cada uma contendo 4 hosts;
 - É obrigatório a consulta ao servidor DNS (um dos hosts) da sub-rede A;
 - A sub-rede B possui como um de seus hosts um servidor web, que fornece o domínio: www.host.com.br. Este servidor deve ser acessível a todos os outros hosts de toda a rede;
 - A sub-rede D também possui um servidor web que fornece o domínio www.host2.com.br;
 - Cada sub-rede possui o seu próprio roteador, responsável por fornecer um serviço de DHCP, para que os hosts possuam IPs dinâmicos, com exceção dos servidores que e roteadores que deverão possuir IPs estáticos;
 - A sub-rede A possui um enlace para a sub-rede B e D;
 - A sub-rede D possui um enlace para a sub-rede C e B;
 - O protocolo de roteamento para a rede deve ser o protocolo RIP;
1. Definição das sub-redes e máscara de sub-rede.

Primeiramente, vamos utilizar uma classe de IP do tipo C para cada sub-rede, ou seja, IPs que começem com: 192.168.x.x. Para o cenário pedido a rede terá como endereço o endereço: 192.168.0.0. Considerando que devemos ter 4 sub-redes, podemos utilizar a seguinte máscara de sub-rede em binário: 1.1.1.1 1.1.1.1 1.1.1.1 1.1.0.0 que equivale ao seguinte endereço em decimal: 255.255.255.192.

Portanto a configuração de cada sub-rede será:

- Sub-rede A: 192.168.0.0/26
- Sub-rede B: 192.168.0.64/26
- Sub-rede C: 192.168.0.128/26
- Sub-rede D: 192.168.0.192/26

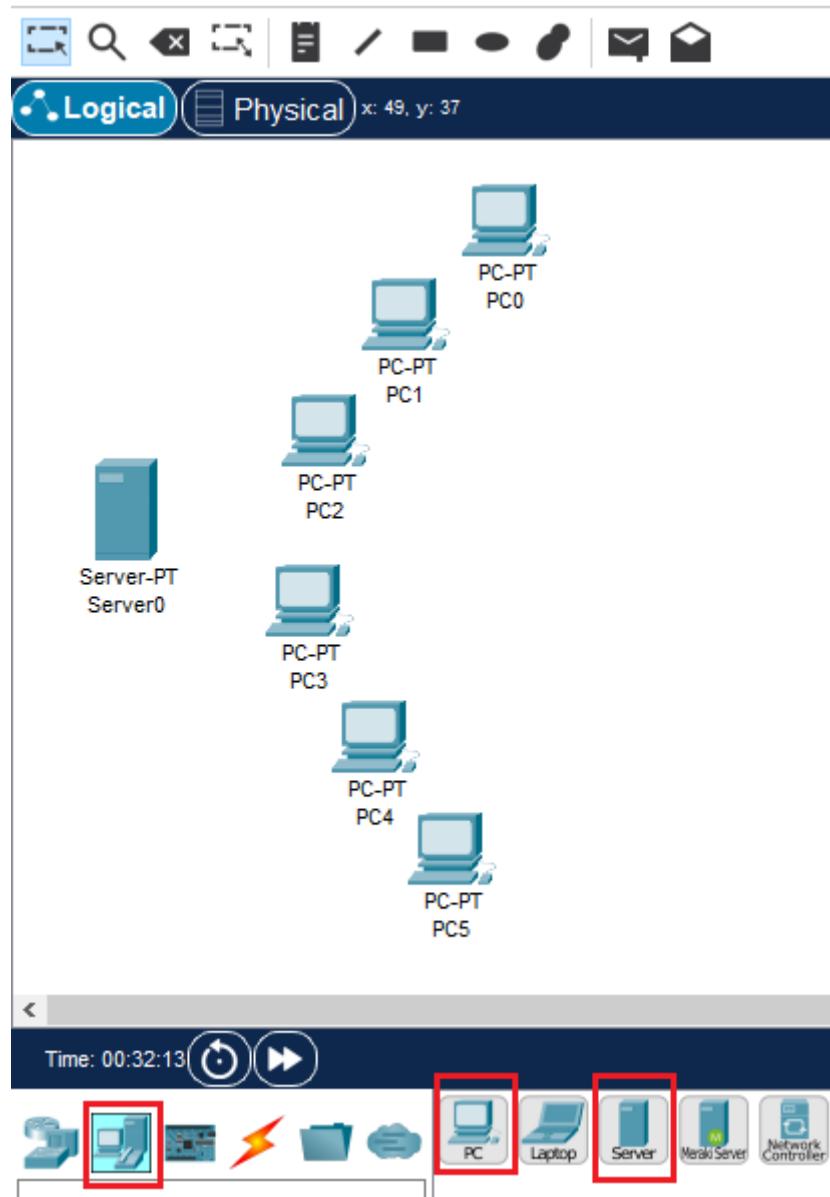
Note que cada sub-rede comporta 64 endereços IPs diferentes, sendo dois deles reservados ao endereço da sub-rede e o endereço de *broadcast*. Os demais são destinados aos hosts, ou seja 62 endereços, o que atende com uma folga considerável o requisito de 7 hosts por subrede.

2. Construção do cenário:

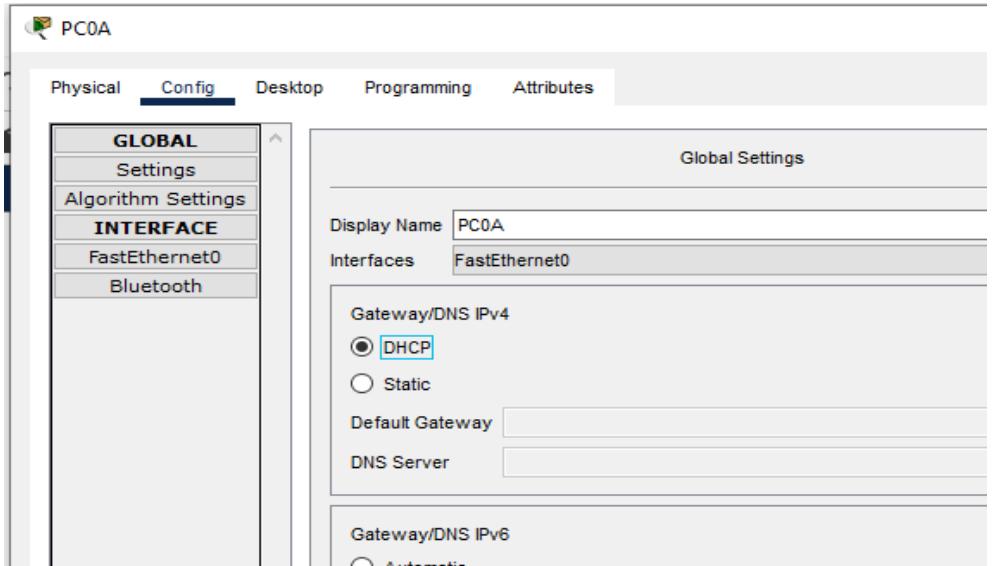
Iremos agora montar o cenário proposto. Para que fique de uma forma didática, iremos montar uma sub-rede de cada por vez. Porém a configuração dos roteadores de cada sub-rede será realizada no final.

2.1. Sub-rede A:

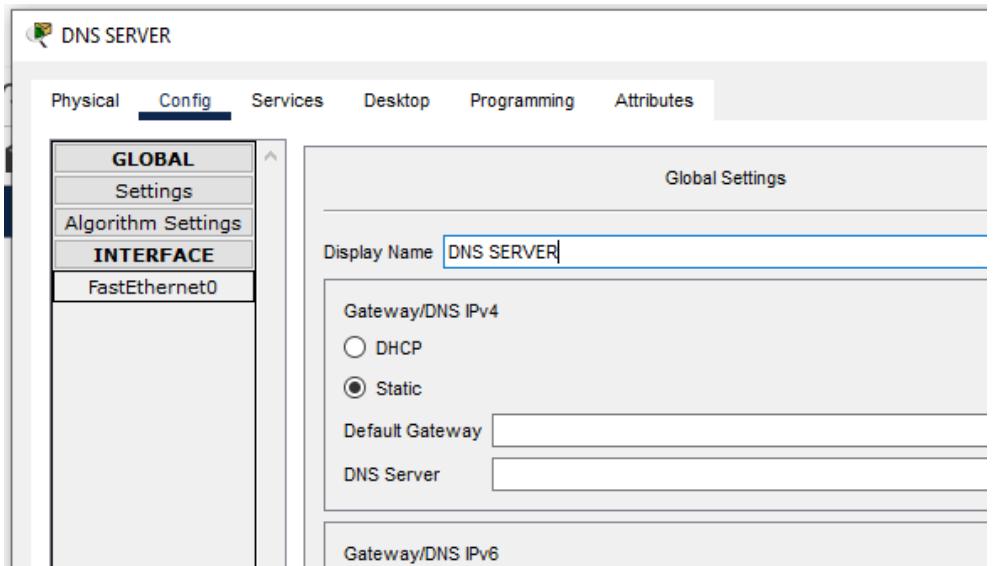
- 2.1.1. Iniciando pela sub-rede A. Abrimos o Cisco Packet Tracer e selecionamos os hosts indo em **End divices**. Lembrando que um deles será o nosso servidor de DNS.



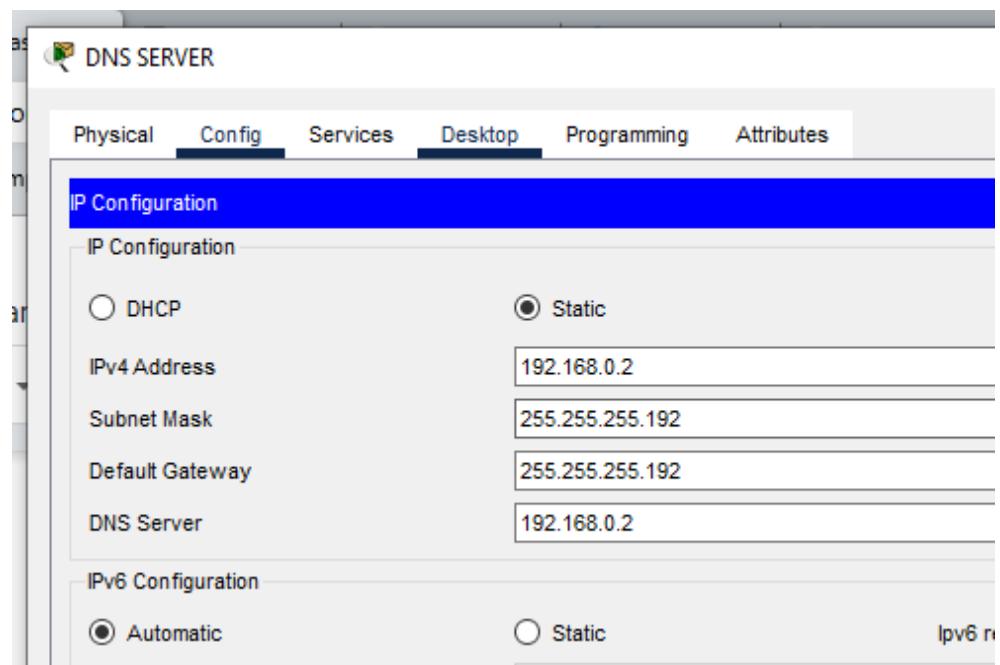
- 2.1.2. Para os PCs iremos renomeá-los adicionando o sufixo **A**. Para isso. Basta clicar duas vezes sobre eles e acessar a aba **Config -> Display Name**. Além disso, ativamos a opção DHCP para que cada um dos hosts recebam as configurações de IP, máscara de sub-rede e DNS de forma automática. Para tanto, vamos novamente na aba **Config -> Gateway/DNS IPv4** e marcamos a opção **DHCP**.



- 2.1.3. Para o servidor de DNS vamos também renomeá-lo. Para isso, clicamos duas vezes sobre ele e vamos na aba **Config** -> **Display Name** e nomeamos para DNS SERVER.



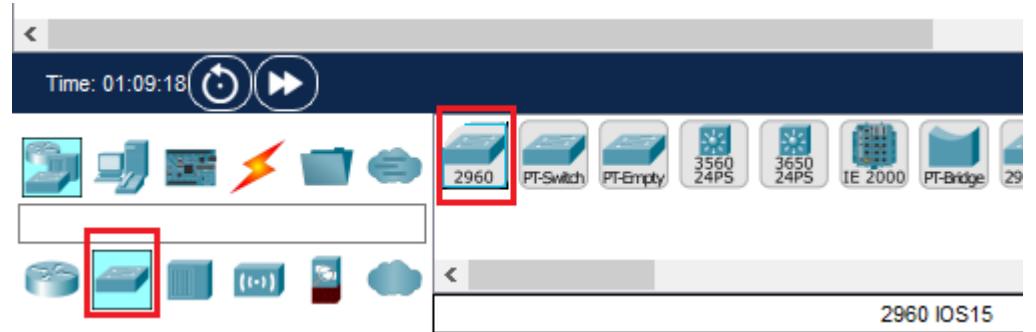
- 2.1.4. Além disso, vamos definir as suas configurações de endereço IP como estáticas. Para isso, vamos agora na aba **Desktop** -> **IP Configuration** e marcamos a opção Static. Em seguida preenchemos conforme abaixo:



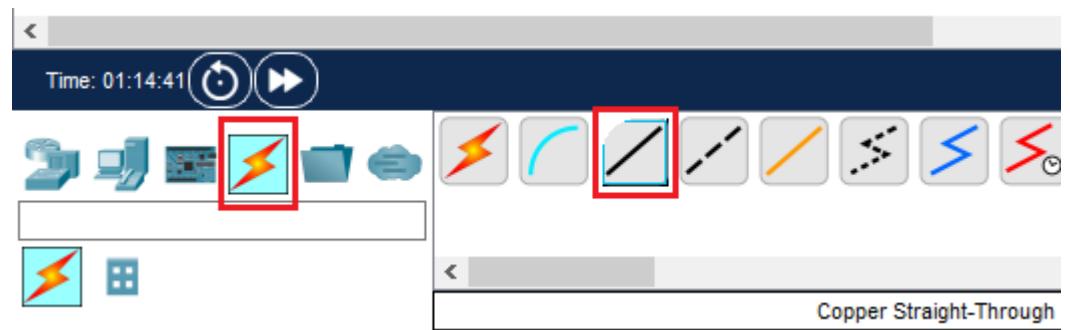
- 2.1.5. Vamos também registrar os servidores web para que o servidor DNS possa fazer o serviço de resolução de nomes. Para isso, adicionamos o domínio e o IP do servidor web que irá prover aquela resposta por aquela requisição.

No.	Name	Type	Detail
0	www.host.com.br	A Record	192.168.0.66
1	www.host2.com.br	A Record	192.168.0.194

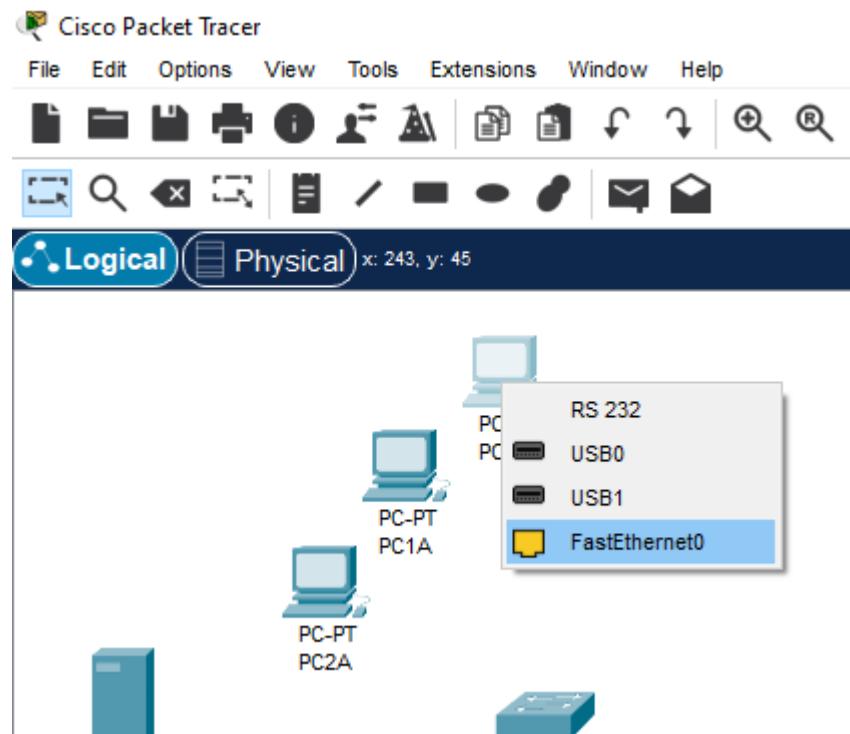
- 2.1.6. Vamos agora selecionar um switch para fazer a conexão dos hosts. Para isso, vamos em **Switches** e selecionamos o *switch 2960*.



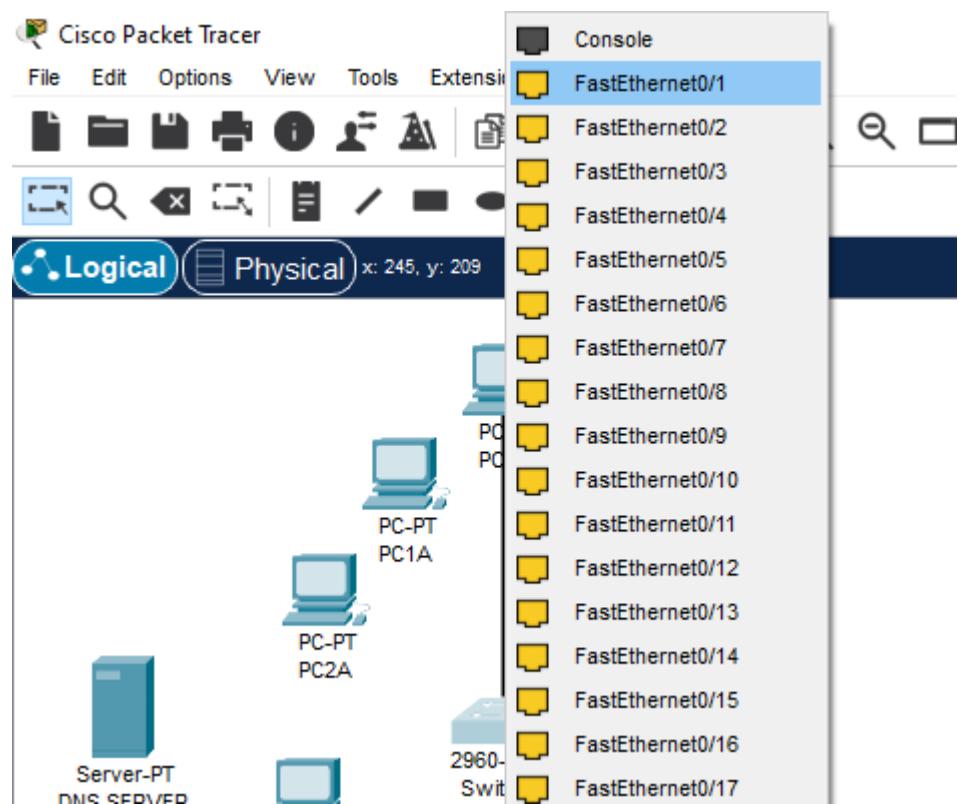
- 2.1.7. Agora iremos fazer a conexão dos hosts. Para isso vamos em **Connections** e selecionamos o cabo de cobre para conexão direta. Ele representará o cabo par-trançado padrão usado para conexões de rede.



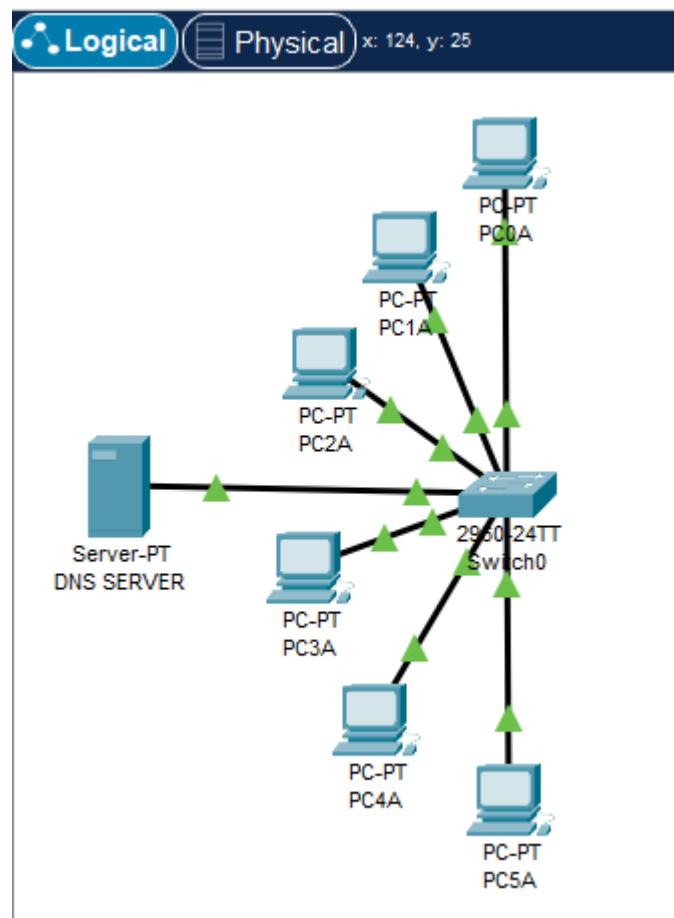
- 2.1.8. Para fazer as conexões clicamos em cada um dos hosts, selecionando a porta *FastEthernet*.



Em seguida clicamos no *switch* e selecionamos uma das portas *FastEthernet* disponíveis.

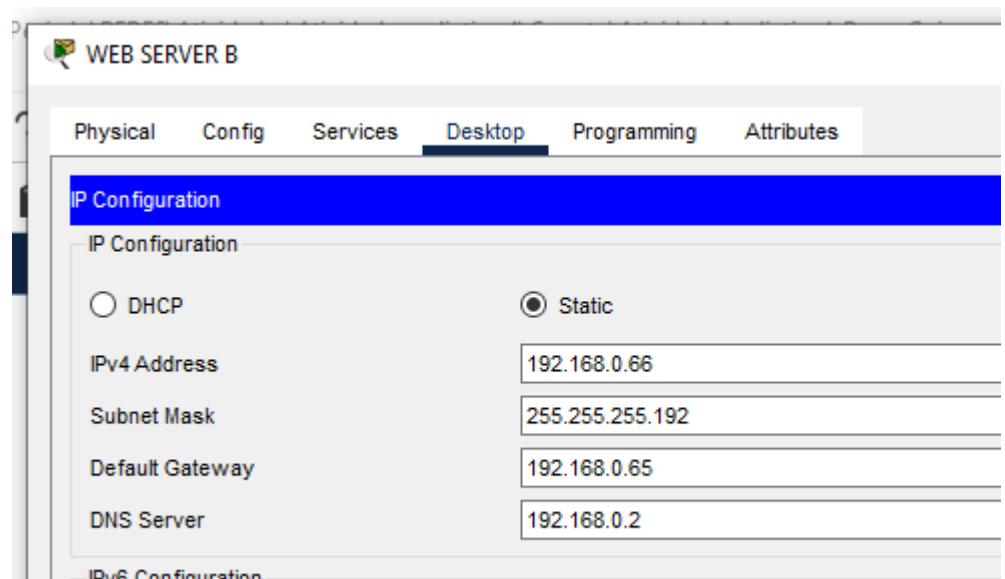


No final teremos a seguinte configuração:

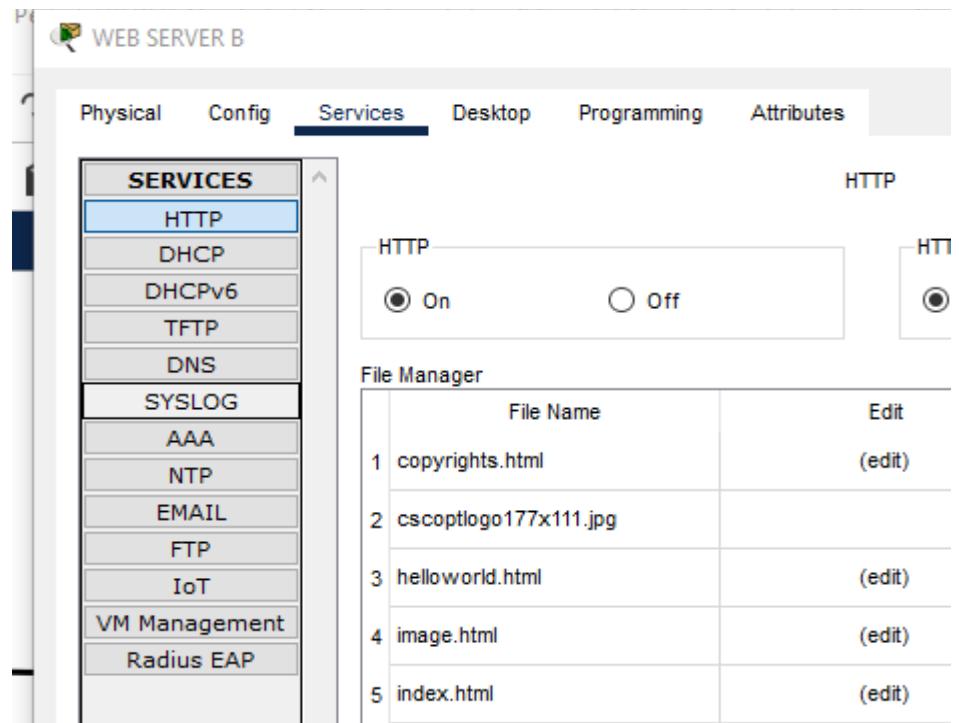


2.2. Sub-rede B:

2.2.1. Realizamos os mesmos procedimentos da seção anterior, modificando apenas o que foi feito nas seções 2.1.3 e 2.1.4. Isso porque, o servidor dessa sub-rede será um servidor web. Portanto, renomeamos o servidor para WEB SERVER B e aplicamos as seguintes configurações de endereço IP e máscara de sub-rede:



- 2.2.2. Configuração do serviço web. Como este servidor será um servidor web, devemos habilitar este serviço. Para isso vamos na aba **Services** -> **HTTP** e habilitamos para On.

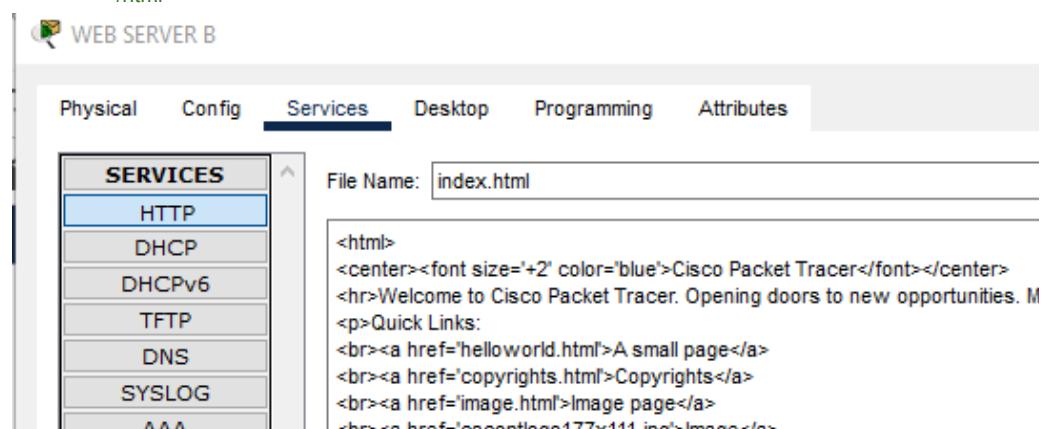


- 2.2.3. Configurando uma página web. Vamos também configurar uma página web simples que será fornecida por este servidor para todos os hosts da rede. Assim, clicamos no arquivo index.html e selecionamos a opção editar e colamos o seguinte código html abaixo e salvamos.

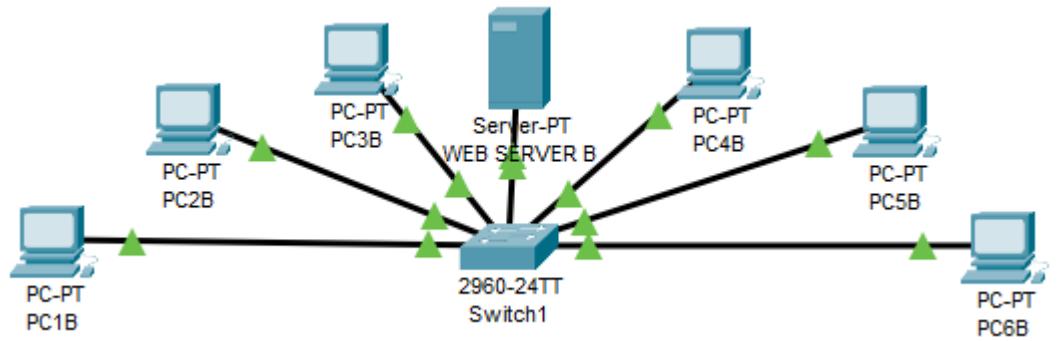
```

<html>
<center><font size='+2' color='blue'>Cisco Packet Tracer</font></center>
<hr>Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide
Open.
<p>Quick Links:
<br><a href='helloworld.html'>A small page</a>
<br><a href='copyrights.html'>Copyrights</a>
<br><a href='image.html'>Image page</a>
<br><a href='cscoptlogo177x111.jpg'>Image</a>
<h2>www.host.com.br -> Armazenado no Web Server da sub-rede B</h2>
</html>

```

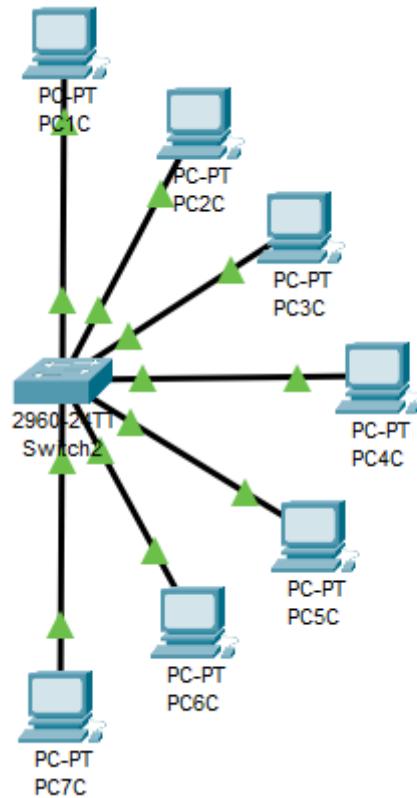


2.2.4. Ao final a sub-rede B tem a seguinte configuração:



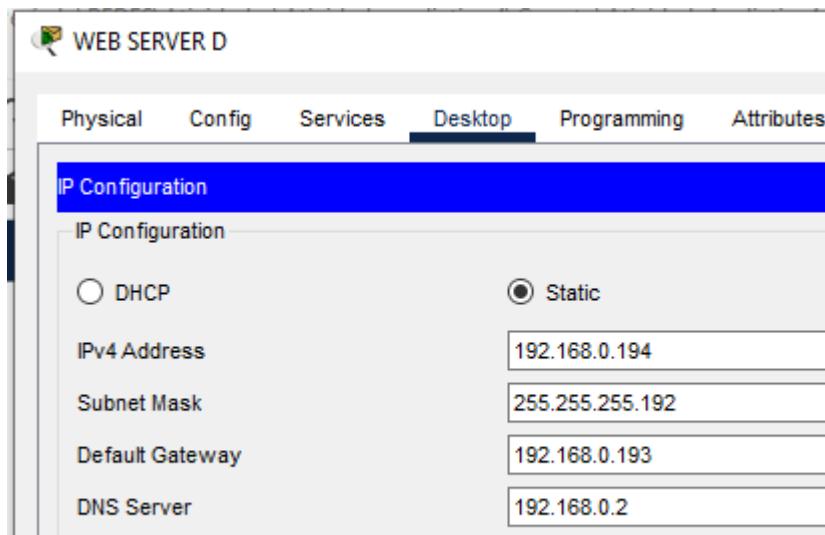
2.3. Sub-rede C:

2.3.1. Realizamos os mesmos procedimentos da seção 2.1 exceto os passos 2.1.3 e 2.1.4 pois, nesta sub-rede não teremos um servidor. Em seu lugar adicionamos mais um PC. A sua configuração pode ser vista abaixo:



2.4. Sub-rede D:

2.4.1. Realizamos os mesmos procedimentos feitos na sub-rede B, modificando apenas as configurações do servidor web. Para isso, alteramos o seu nome para SERVIDOR WEB D e adicionamos as seguintes configurações de endereço IP e máscara de sub-rede:



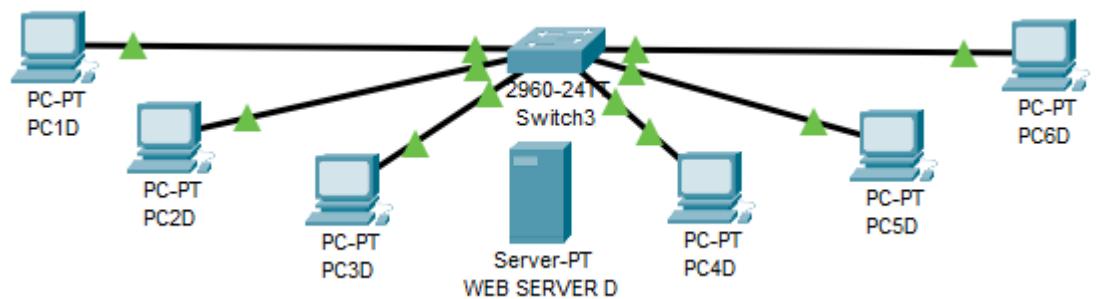
Além disso, na etapa 2.2.3 alteramos o código para:

```

<html>
<center><font size='2' color='blue'>Cisco Packet Tracer</font></center>
<hr>Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide
Open.
<p>Quick Links:
<br><a href='helloworld.html'>A small page</a>
<br><a href='copyrights.html'>Copyrights</a>
<br><a href='image.html'>Image page</a>
<br><a href='cscoptlogo177x111.jpg'>Image</a>
<h2>www.host2.com.br -> Armazenado no Web Server da sub-rede D</h2>
</html>

```

Ao final, a sub-rede D possui a seguinte configuração:

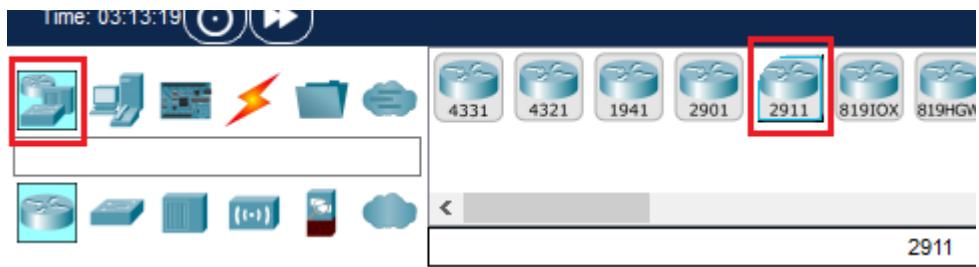


2.5. Configuração do roteador

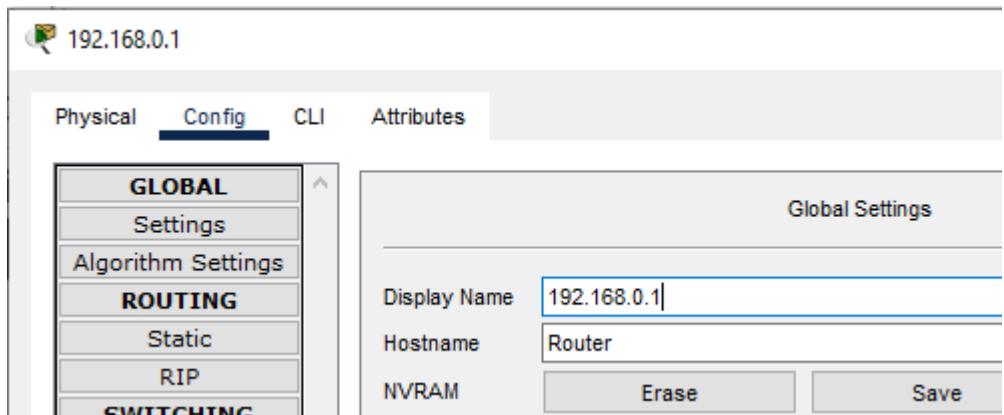
Nessa etapa iremos configurar os roteadores, tendo em mente que o protocolo de roteamento a ser utilizado será o protocolo RIP.

2.5.1. Sub-rede A:

2.5.2. Nesta etapa vamos iniciar a configuração do roteador da sub-rede A. Para isso, selecionamos o roteador em **Network Devices** e escolhemos o roteador 2911.



- 2.5.3. Primeiramente vamos mudar o nome do roteador. Por ser o roteador da sub-rede A, iremos identificá-lo com o primeiro IP da sub-rede, ou seja: 192.168.0.1. Para isso, clicamos duas vezes sobre ele e na aba **Config -> Display Name**, setamos o nome do roteador



- 2.5.4. Agora iremos executar um script via linha de comando. Os roteadores CISCO possuem uma interface na qual podemos acessar através via console. No Cisco Packet Tracer clicamos duas vezes no roteador e vamos na aba **CLI** e adicionamos o seguinte comando:

```

en
conf t
int gi0/0
ip add 192.168.0.1 255.255.255.192
no shut
ip dhcp pool SUBNET_A
default-router 192.168.0.1
network 192.168.0.0 255.255.255.192
dns-server 192.168.0.2
end
write

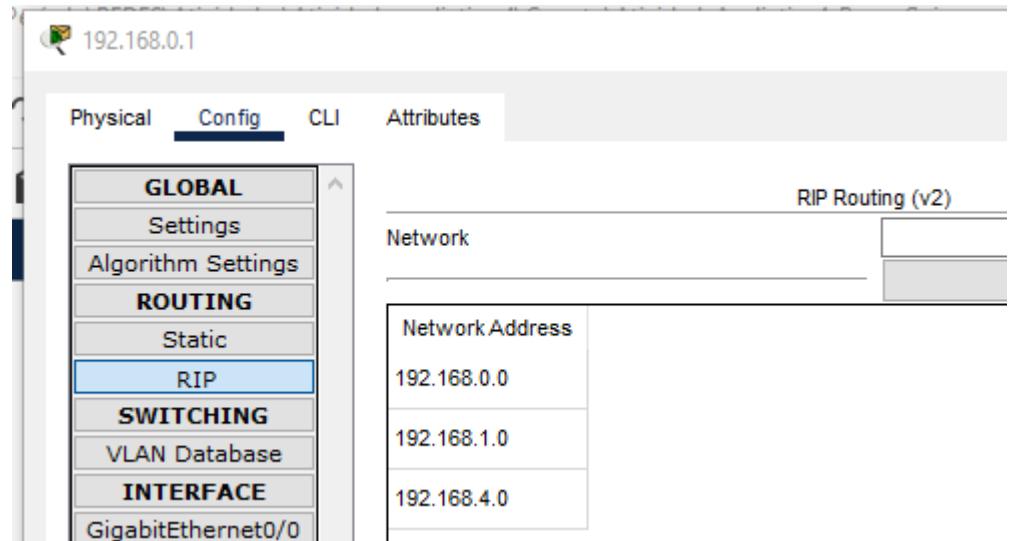
```

Este comando irá basicamente configurar o endereço do roteador, sua máscara de sub-rede e IP do servidor DNS da rede.

- 2.5.5. Para que as sub-redes possam se comunicar entre si, iremos configurar o roteamento considerando o protocolo RIP. A sub-rede A deve se comunicar diretamente com as sub-redes B

e D conforme especificado no início. Com isso, definimos então na aba **Config -> Routing -> RIP** os seguintes endereços.

- 192.168.0.0 IP da rede
- 192.168.1.0 IP do enlace de comunicação com a sub-rede B;
- 192.168.4.0 IP do enlace de comunicação com a sub-rede D;



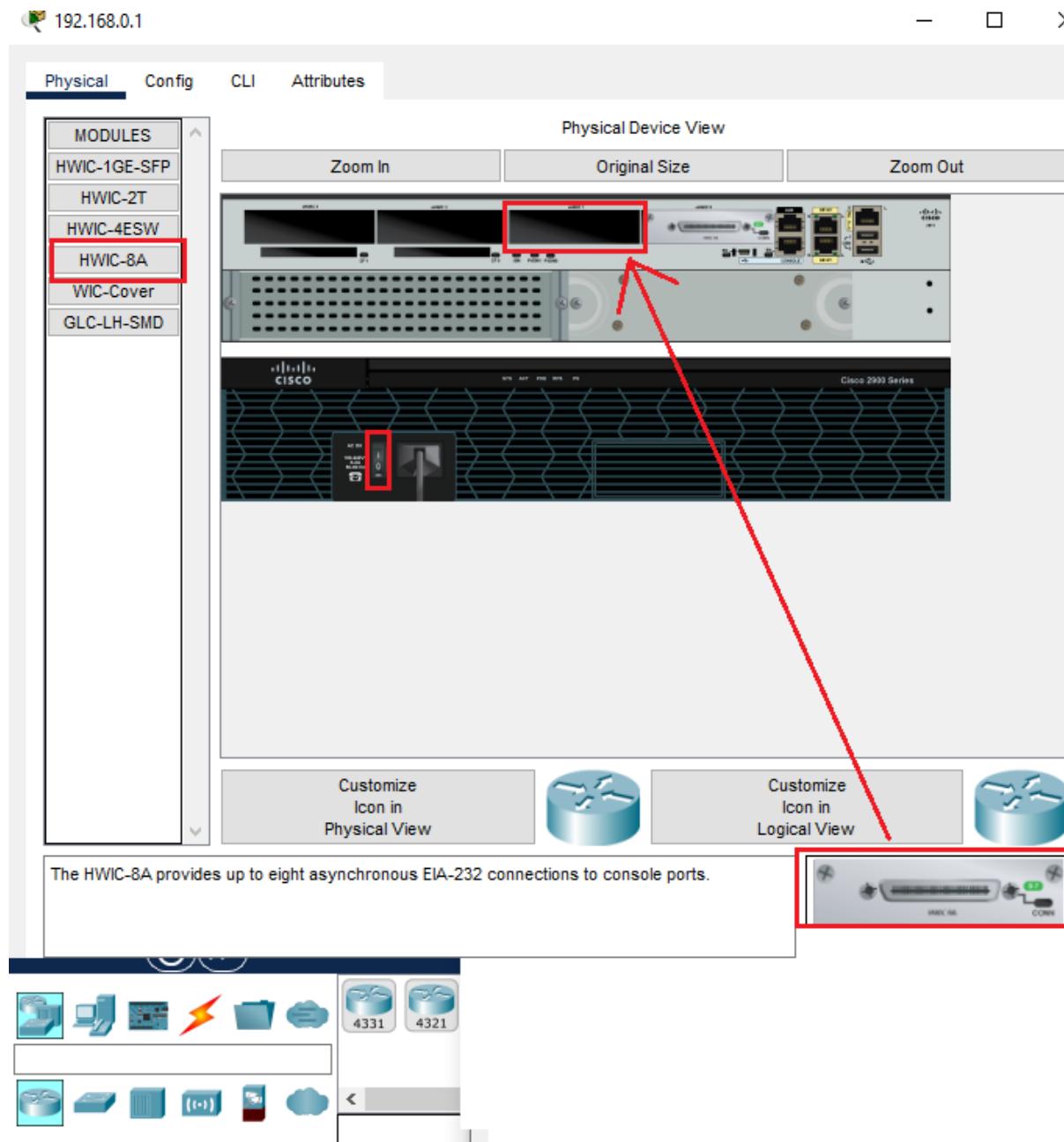
2.5.6. Além disso, configuramos adicionalmente a versão do protocolo rip para a versão 2 e desabilitamos a função *auto-summary*, digitado o comando abaixo no terminal acessando a aba **CLI**.

```
router rip  
version 2  
no auto-summary
```

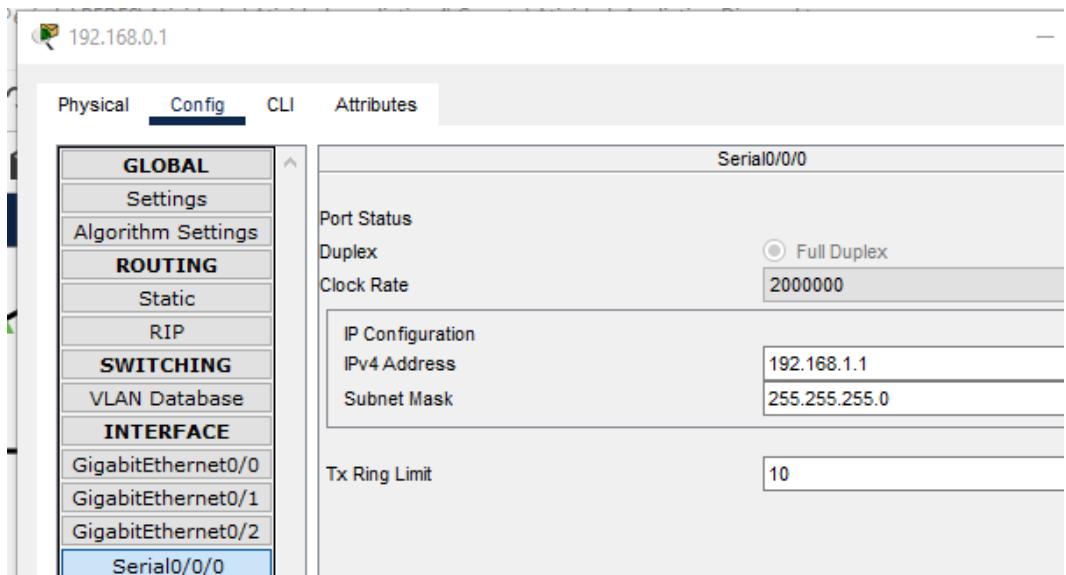
The screenshot shows a Cisco IOS CLI interface. At the top, there are tabs: Physical, Config, **CLI**, and Attributes. Below the tabs, it says "IOS Command Line Interface". The URL "http://www.cisco.com/wwl/export/crypto/tool/stqrg.html" is displayed. The main text area contains the following information:

```
If you require further assistance please contact us by sending email to  
export@cisco.com.  
  
Cisco CISCO2911/K9 (revision 1.0) with 491520K/32768K bytes of memory.  
Processor board ID FTX152400KS  
3 Gigabit Ethernet interfaces  
DRAM configuration is 64 bits wide with parity disabled.  
255K bytes of non-volatile configuration memory.  
249856K bytes of ATA System CompactFlash 0 (Read/Write)  
  
--- System Configuration Dialog ---  
  
Would you like to enter the initial configuration dialog? [yes/no]: n  
  
Press RETURN to get started!  
  
Router>enable  
Router#  
Router#configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Router(config)#router rip  
Router(config-router)#network 192.168.0.0  
Router(config-router)#network 192.168.1.0  
Router(config-router)#network 192.168.4.0  
Router(config-router)#router rip  
Router(config-router)#version 2  
Router(config-router)#no a  
Router(config-router)#no auto-summary  
Router(config-router)#[
```

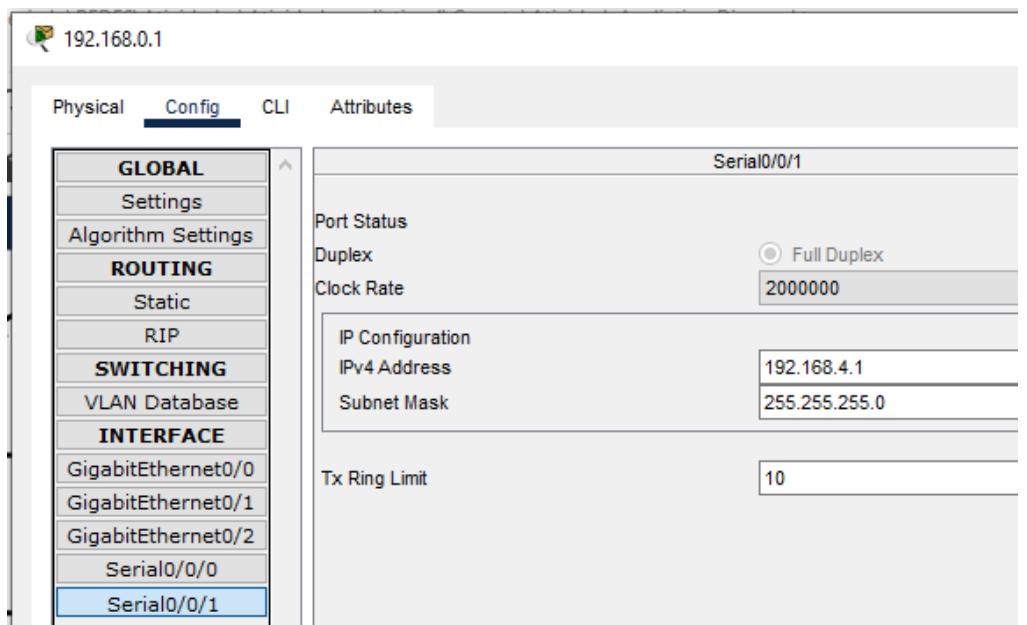
- 2.5.7. Por fim, adicionamos duas portas seriais para a comunicação entre os roteadores. Para fazer isso nós desligamos o roteador clicando na chave *on/off*. Em seguida selecionamos aba **Physical -> Modules** e adicionamos dois módulos HWIC-8A e então clicamos e arrastamos a porta serial na parte inferior do roteador e soltamos nos slots disponíveis na parte superior, conforme a imagem abaixo. Após isso, ligamos novamente o roteador.



2.5.8. Vamos agora configurar as portas seriais. Clicando no roteador vamo na aba **Config** -> **Serial0/0/0** -> **IP Configuration** e adicionamos as informações abaixo.



Para a porta serial 0/0/1 adicionamos a configuração abaixo:



2.5.9. Sub-rede B:

Executamos os mesmo procedimentos alterando os seguintes passos:

2.5.3. o nome do roteador é 192.168.0.65

2.5.4 o script é:

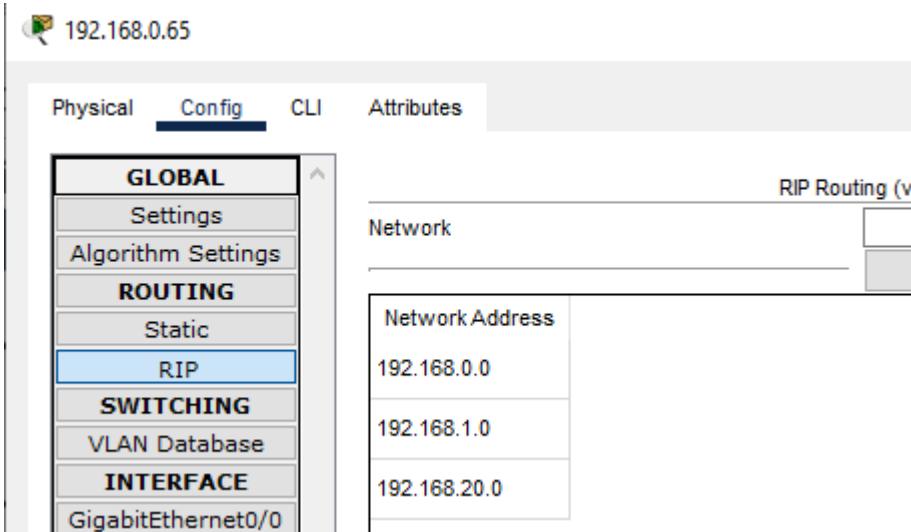
```

en
conf t
int gi0/0
ip add 192.168.0.65 255.255.255.192
no shut
ip dhcp pool SUBNET_B
default-router 192.168.0.65
network 192.168.0.64 255.255.255.192

```

```
dns-server 192.168.0.2  
end  
write
```

2.5.5 os IPs do protocolo de roteamento são:



2.5.8 os IPs das portas seriais são:

- Porta Serial0/0/0:
 IPv4 Address: 192.168.1.2
 Subnet Mask: 255.255.255.0
- Porta Serial0/0/1:
 IPv4 Address: 192.168.20.1
 Subnet Mask: 255.255.255.0

2.5.10. Sub-rede C:

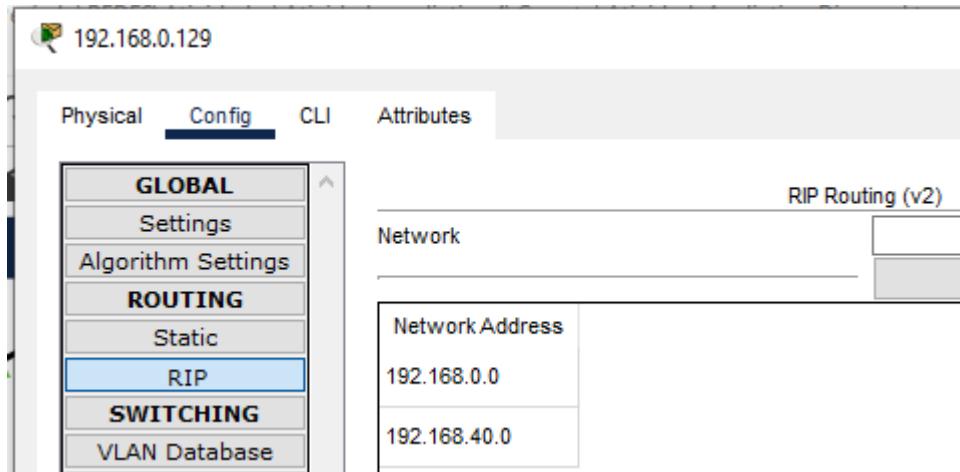
Executamos os mesmo procedimentos alterando os seguintes passos:

2.5.3. o nome do roteador é 192.168.0.129

2.5.4 o script é:

```
en  
conf t  
int gi0/0  
ip add 192.168.0.129 255.255.255.192  
no shut  
ip dhcp pool SUBNET_C  
default-router 192.168.0.129  
network 192.168.0.128 255.255.255.192  
dns-server 192.168.0.2  
end  
write
```

2.5.5 os IPs do protocolo de roteamento são:



2.5.8 Só há uma porta serial e o seu IP é:

- Porta Serial0/0/0:
IPv4 Address: 192.168.40.2
Subnet Mask: 255.255.255.0

2.5.11. Sub-rede D:

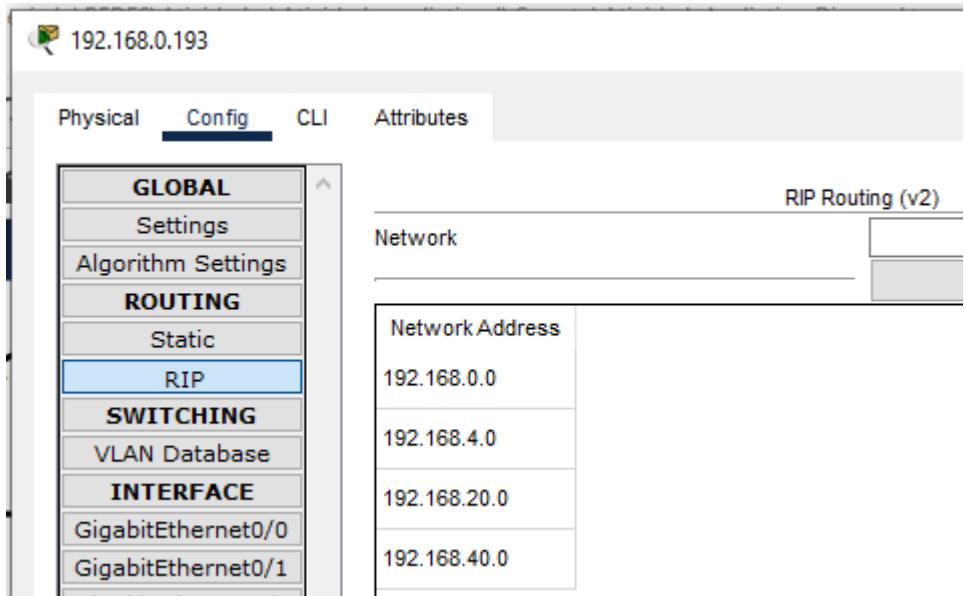
Executamos os mesmo procedimentos alterando os seguintes passos:

2.5.3. o nome do roteador é 192.168.0.193

2.5.4 o script é:

```
en
conf t
int gi0/0
ip add 192.168.0.193 255.255.255.192
no shut
ip dhcp pool SUBNET_D
default-router 192.168.0.193
network 192.168.0.192 255.255.255.192
dns-server 192.168.0.2
end
write
```

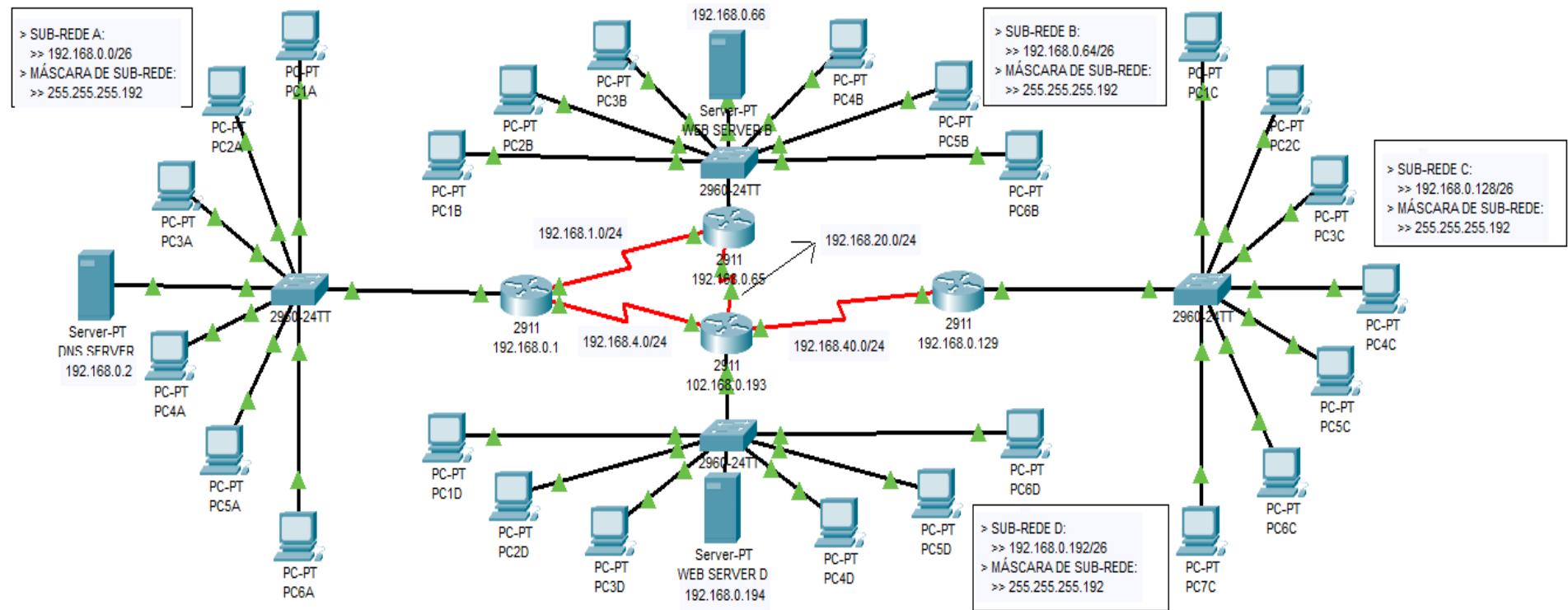
2.5.5 os IPs do protocolo de roteamento são:



2.5.8 A sub-rede D tem três portas seriais e os seus IPs são:

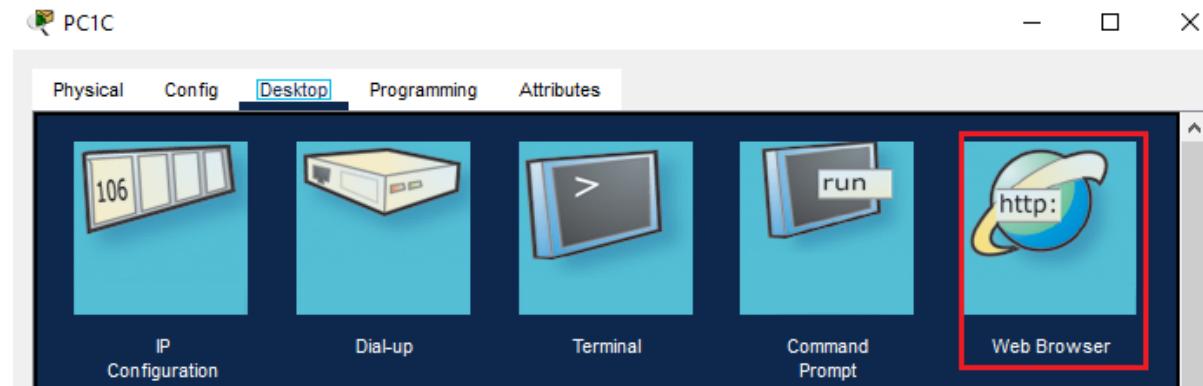
- Porta Serial0/0/0:
IPv4 Address: 192.168.4.2
Subnet Mask: 255.255.255.0
- Porta Serial0/0/1:
IPv4 Address: 192.168.20.2
Subnet Mask: 255.255.255.0
- Porta Serial0/0/1:
IPv4 Address: 192.168.40.1
Subnet Mask: 255.255.255.0

A topologia da rede fica então da seguinte forma:



3. Testes

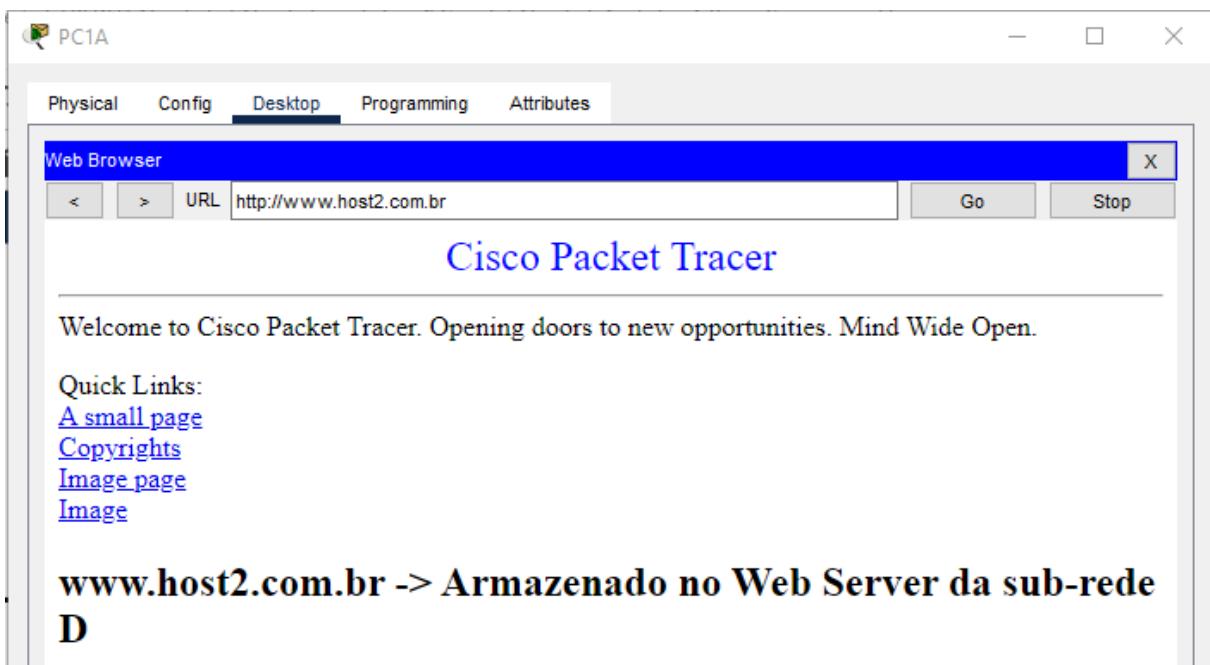
- 3.1. Acessando a página do servidor web B a partir de um host da sub-rede C
 - 3.1.1. Clicamos duas vezes em um dos hosts da sub-rede C e vamos na aba **Desktop -> Web Browser**.



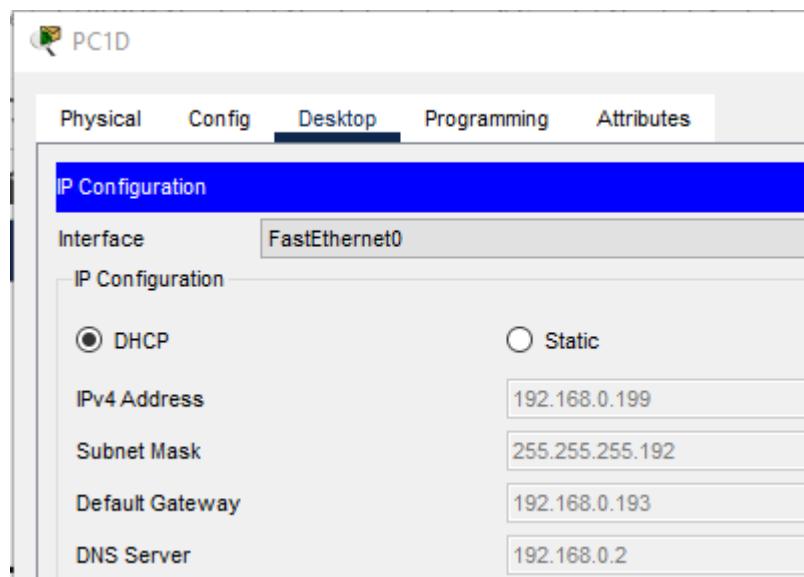
e digitamos o endereço que cadastramos no início: www.host.com.br



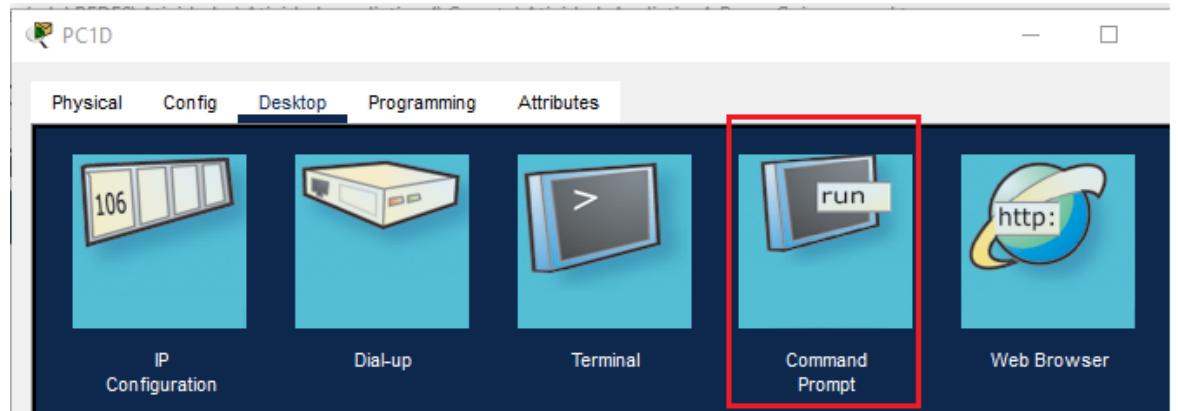
- 3.2. Acessando a página do servidor web D a partir de um host da sub-rede A



- 3.3. Dando um ping em um host da sub-rede D a partir da sub-rede A:
A sub-rede D tem endereços IPs no range: 192.168.0.192 - 192.168.0.254. Vamos executar um comando ping para o endereço 192.168.0.199 que, conforme podemos ver abaixo, corresponde a um de seus hosts.



Para tanto, clicamos duas vezes em um host da sub-rede A e vamos na aba **Desktop -> Command Prompt**



e digitamos **ping 192.168.0.199**

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.0.199

Pinging 192.168.0.199 with 32 bytes of data:

Reply from 192.168.0.199: bytes=32 time=1ms TTL=126
Reply from 192.168.0.199: bytes=32 time=2ms TTL=126
Reply from 192.168.0.199: bytes=32 time=1ms TTL=126
Reply from 192.168.0.199: bytes=32 time=36ms TTL=126

Ping statistics for 192.168.0.199:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 36ms, Average = 10ms

C:\>
```

A screenshot of the Cisco Packet Tracer Command Prompt window. The title bar says 'PC6A'. The window has a blue header bar with the text 'Command Prompt'. The main area of the window shows the output of a 'ping' command. It starts with 'Cisco Packet Tracer PC Command Line 1.0' and 'C:\>ping 192.168.0.199'. Then it shows four replies from the target IP address. Finally, it provides ping statistics: 'Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)', 'Approximate round trip times in milli-seconds', and 'Minimum = 1ms, Maximum = 36ms, Average = 10ms'. The prompt 'C:\>' is visible at the bottom.

Com isso, a rede está operacional e funcionando conforme especificado.