

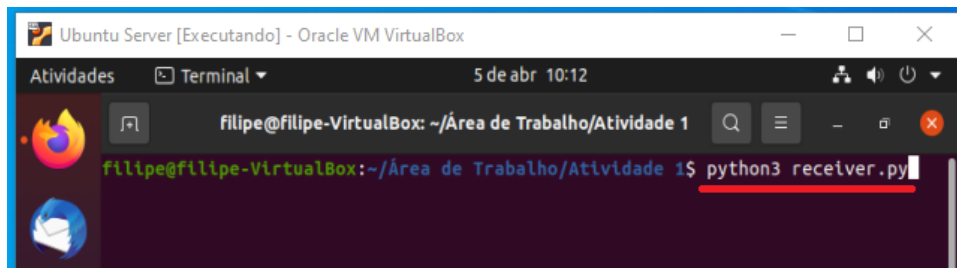
Descrição Projeto 1

Nome: Filipe Brener Ferreira Santos

Matricula: 5952

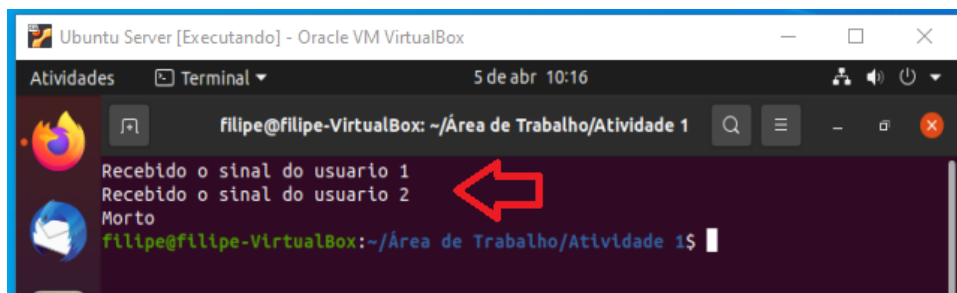
Etapa 1

Para executar a etapa 1 basta inicializar o receiver.py no terminal assim como no exemplo abaixo:

A screenshot of a terminal window titled 'Ubuntu Server [Executando] - Oracle VM VirtualBox'. The terminal shows the prompt 'filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 1' and the command 'python3 receiver.py' being entered. The command is underlined in red. The terminal window has a dark background and a light blue border.

O próprio receiver.py ficará responsável pela inicialização do processo sender.py.

Após toda a execução do sender.py a saída no terminal com o receiver.py ficará da seguinte forma:

A screenshot of a terminal window titled 'Ubuntu Server [Executando] - Oracle VM VirtualBox'. The terminal shows the output of the receiver.py script: 'Recebido o sinal do usuario 1', 'Recebido o sinal do usuario 2', and 'Morto'. A red arrow points to the 'Recebido o sinal do usuario 2' line. The terminal window has a dark background and a light blue border.

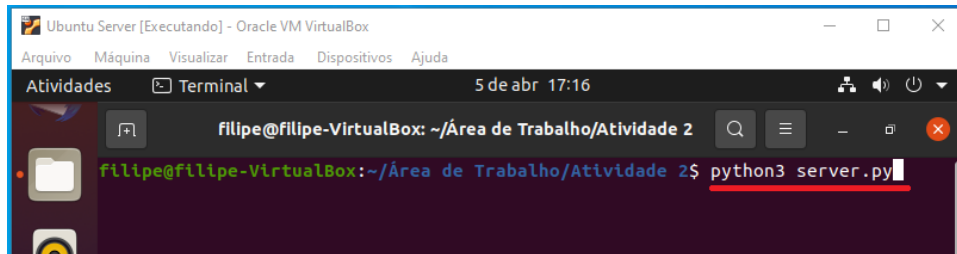
Foi utilizado para a etapa 1 uma máquina virtual com um ambiente Ubuntu 20.04 com Python 3.7.

Descrição da execução da etapa 1:

O receiver.py faz uma chamada de sistema inicializando o sender.py passando seu próprio PID como parâmetro e fica em espera bloqueante de um sinal, através do signal.pause(). O sender.py por sua vez envia 3 sinais para o receiver.py com um intervalo de 3 segundos cada. No tratamento de cada sinal no receiver.py é printado na tela o recebimento do mesmo e logo em seguida entra em espera bloqueante novamente, esse comportamento se repete até ele receber um SIGKILL, matando o processo.

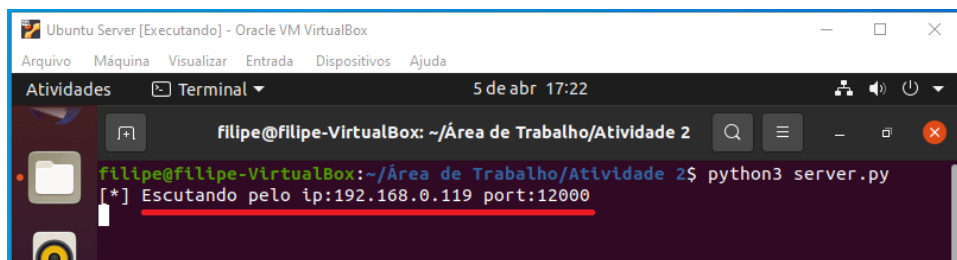
Etapa 2

A etapa 2 do projeto deve-se iniciar o server.py primeiro no terminal assim como no exemplo abaixo:



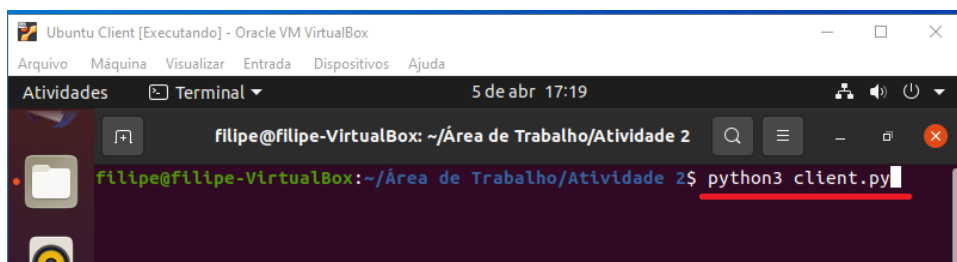
A screenshot of a terminal window titled 'Ubuntu Server [Executando] - Oracle VM VirtualBox'. The window shows the command `python3 server.py` being entered at the prompt `filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2`. The command is highlighted with a red underline.

Após o server ser inicializado aparecerá no terminal o endereço e a porta por qual o servidor se comunicará como no exemplo abaixo.



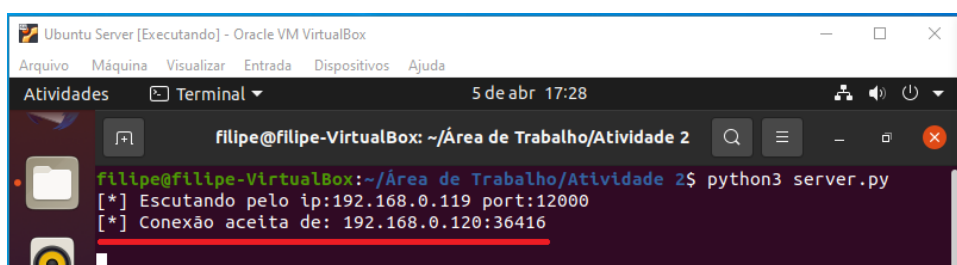
A screenshot of the same terminal window showing the output of the `python3 server.py` command. The output is `[*] Escutando pelo ip:192.168.0.119 port:12000`, which is highlighted with a red underline.

Nesse ponto o servidor estará aguardando uma requisição de conexão de algum cliente. Por sua vez em outra máquina o agora o client.py deve ser executado assim como no exemplo abaixo:



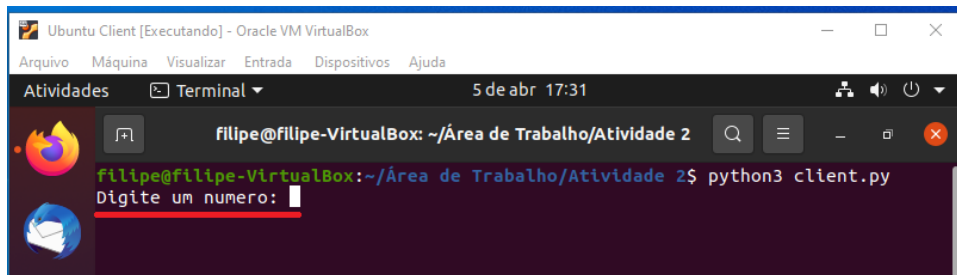
A screenshot of a terminal window titled 'Ubuntu Client [Executando] - Oracle VM VirtualBox'. The window shows the command `python3 client.py` being entered at the prompt `filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2`. The command is highlighted with a red underline.

Ao executar o client.py automaticamente o client irá se conectar com o server aparecendo no terminal do server a conexão sendo aceita como na imagem abaixo.



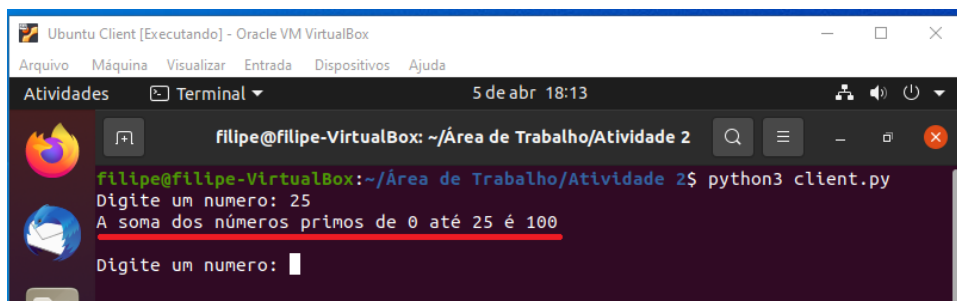
A screenshot of the terminal window from the previous step, showing the output of the `python3 server.py` command. The output now includes two lines: `[*] Escutando pelo ip:192.168.0.119 port:12000` and `[*] Conexão aceita de: 192.168.0.120:36416`. The second line is highlighted with a red underline.

Por sua vez o client irá pedir para que o usuário insira um número cujo será enviado para o server para ser tratado, como ilustrado na imagem abaixo.



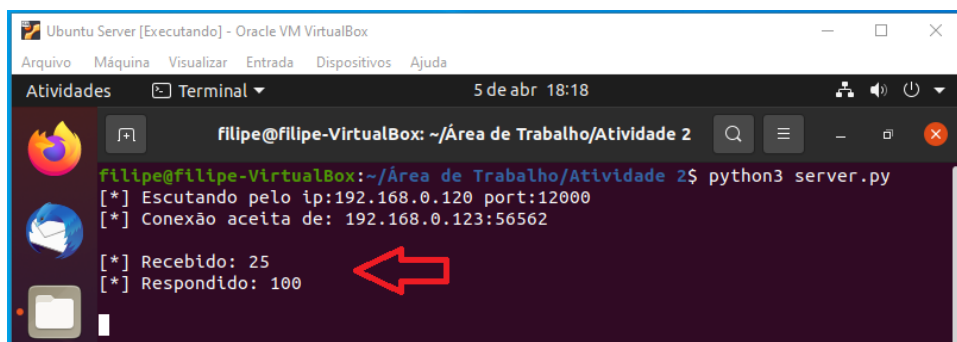
```
Ubuntu Client [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
Atividades  Terminal  5 de abr 17:31
filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$ python3 client.py
Digite um numero: 
```

Após digitar um número como por exemplo 25, esse número será enviado para o server que retornará 100 e logo em seguida o processo já pede novamente para que o usuário entre com um número como mostrado na imagem abaixo.



```
Ubuntu Client [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
Atividades  Terminal  5 de abr 18:13
filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$ python3 client.py
Digite um numero: 25
A soma dos números primos de 0 até 25 é 100
Digite um numero: 
```

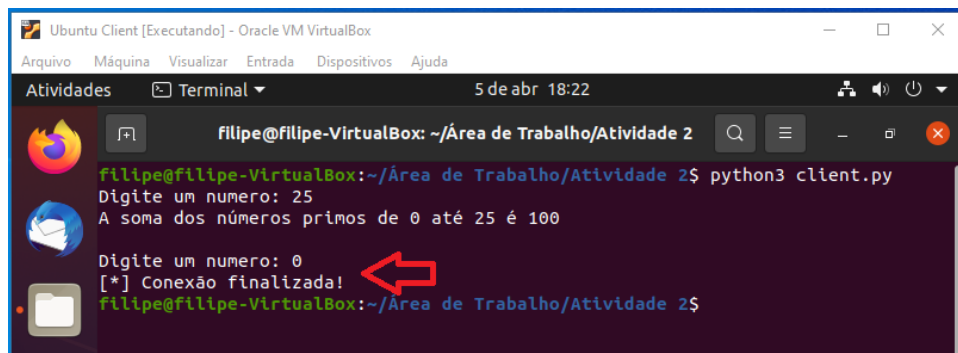
Dado o mesmo acontecimento o terminal do server aparecerá dois outputs, o primeiro do número recebido e o segundo do número enviado.



```
Ubuntu Server [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
Atividades  Terminal  5 de abr 18:18
filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$ python3 server.py
[*] Escutando pelo ip:192.168.0.120 port:12000
[*] Conexão aceita de: 192.168.0.123:56562
[*] Recebido: 25
[*] Respondido: 100
```

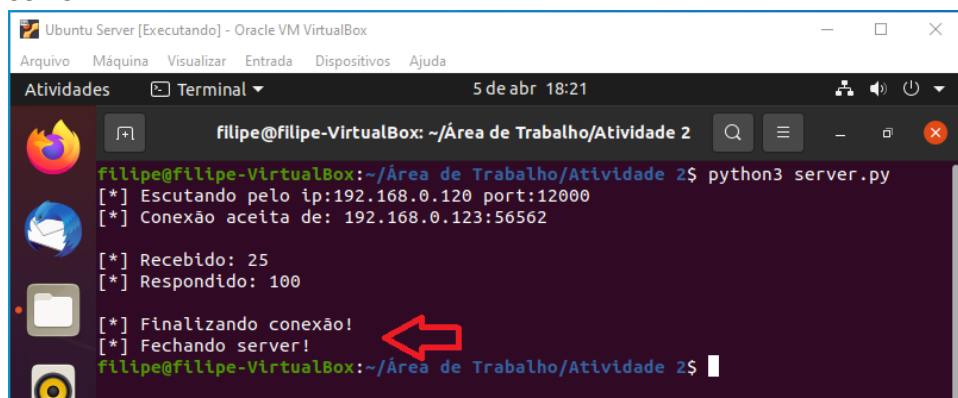
Os processos irão se repetir até que o usuário entre com o número 0 no lado do cliente, como nas imagens abaixo:

Client:



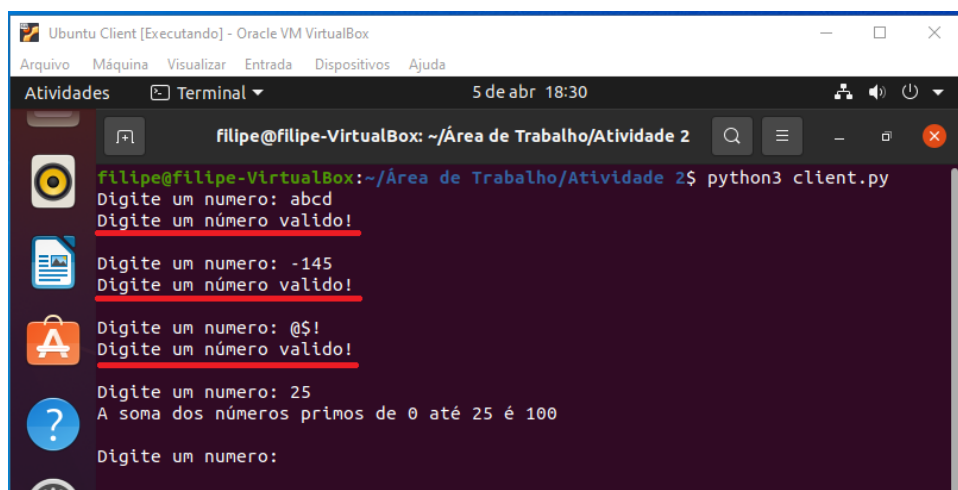
```
filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$ python3 client.py
Digite um numero: 25
A soma dos números primos de 0 até 25 é 100
Digite um numero: 0
[*] Conexão finalizada!
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$
```

Server:



```
filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$ python3 server.py
[*] Escutando pelo ip:192.168.0.120 port:12000
[*] Conexão aceita de: 192.168.0.123:56562
[*] Recebido: 25
[*] Respondido: 100
[*] Finalizando conexão!
[*] Fechando server!
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$
```

Foi utilizado para a etapa 2 duas máquinas virtuais com um ambiente Ubuntu 20.04 com Python 3.7. Caso o usuário tente entrar algo que não seja um número válido o client não envia o input para o server, até que o usuário entre com um dado válido, como ilustrado na imagem abaixo:



```
filipe@filipe-VirtualBox: ~/Área de Trabalho/Atividade 2
filipe@filipe-VirtualBox:~/Área de Trabalho/Atividade 2$ python3 client.py
Digite um numero: abcd
Digite um número válido!
Digite um numero: -145
Digite um número válido!
Digite um numero: @$!
Digite um número válido!
Digite um numero: 25
A soma dos números primos de 0 até 25 é 100
Digite um numero:
```