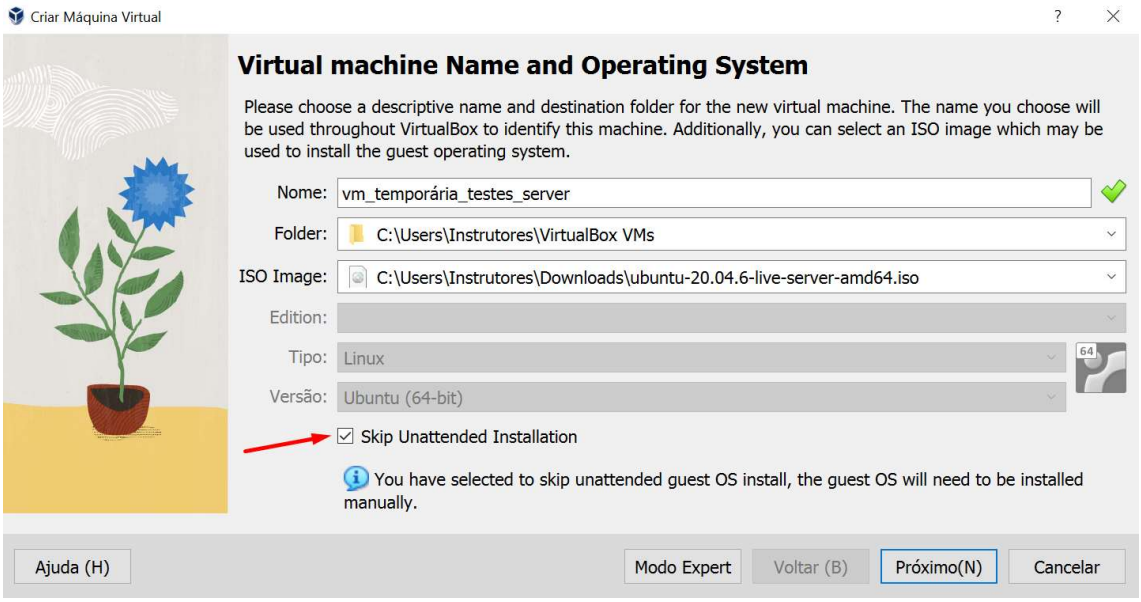
*Passo a passo para instalar a VM + PhpMyAdmin + NODE + React Native a fim de realizar o exercício "API-REACTNATIVE-0705" do Git do Prof. Lucas Calu - https://github.com/lucascalu/API-REACTNATIVE-0705*

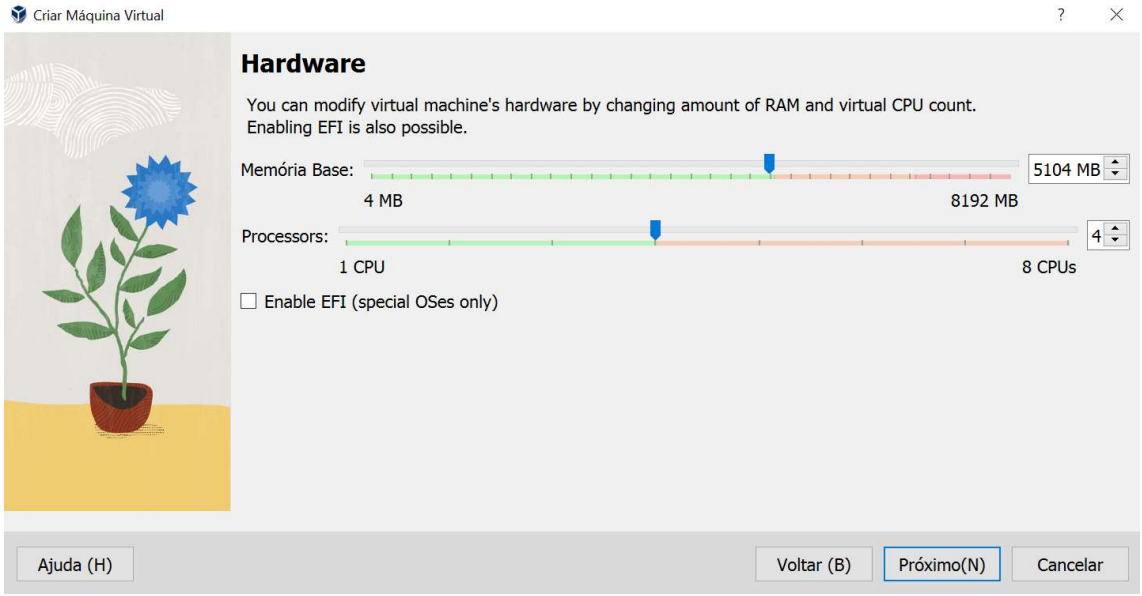
Versão 2

# *Parte I – Instalando a VM e preparando acesso ao phpmyadmin (e MySQL)*

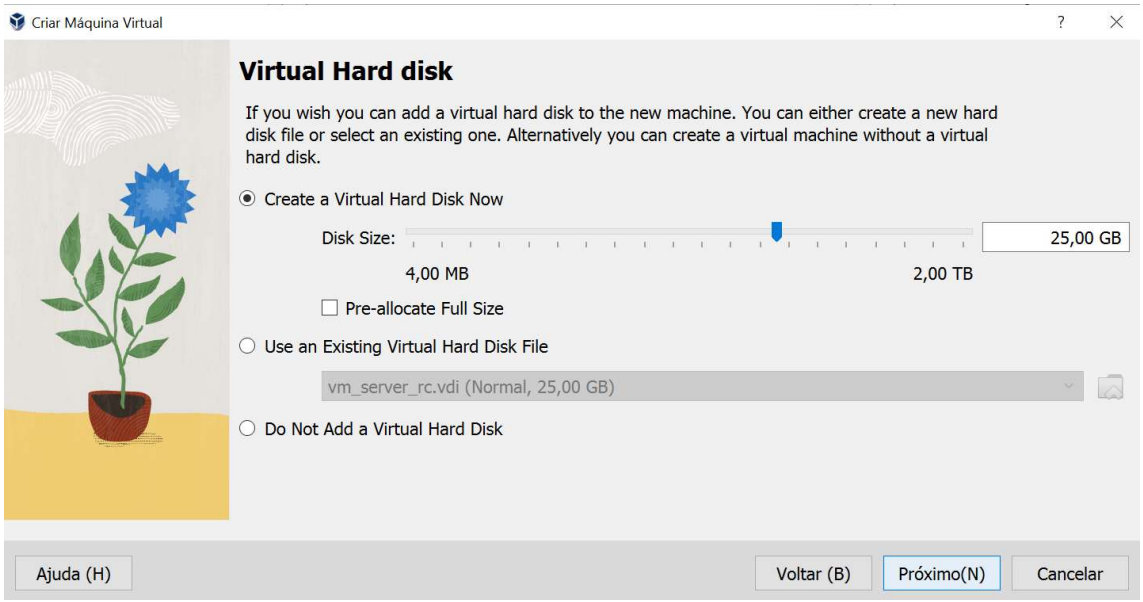
Partindo do suposto que nós já temos o arquivo .ISO salvo, proceda conforme abaixo:




Marque a opção "Skip Unattended Installation";



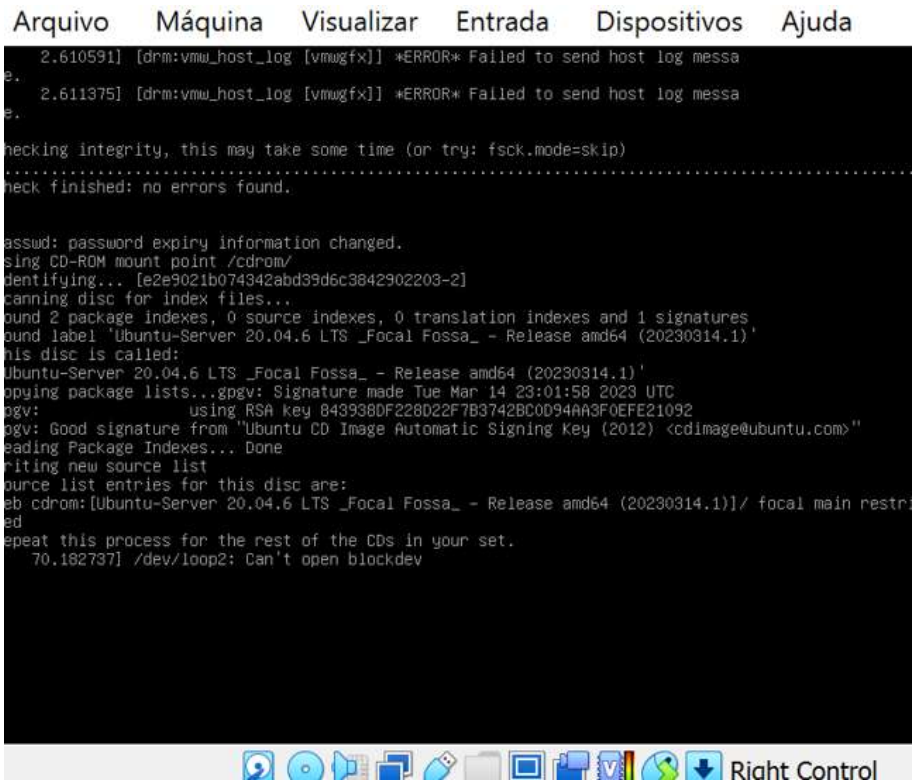

Poderá ampliar as configurações para o “máximo na cor verde”;




Deixe como padrão e avançar... depois clique em Finalizar.

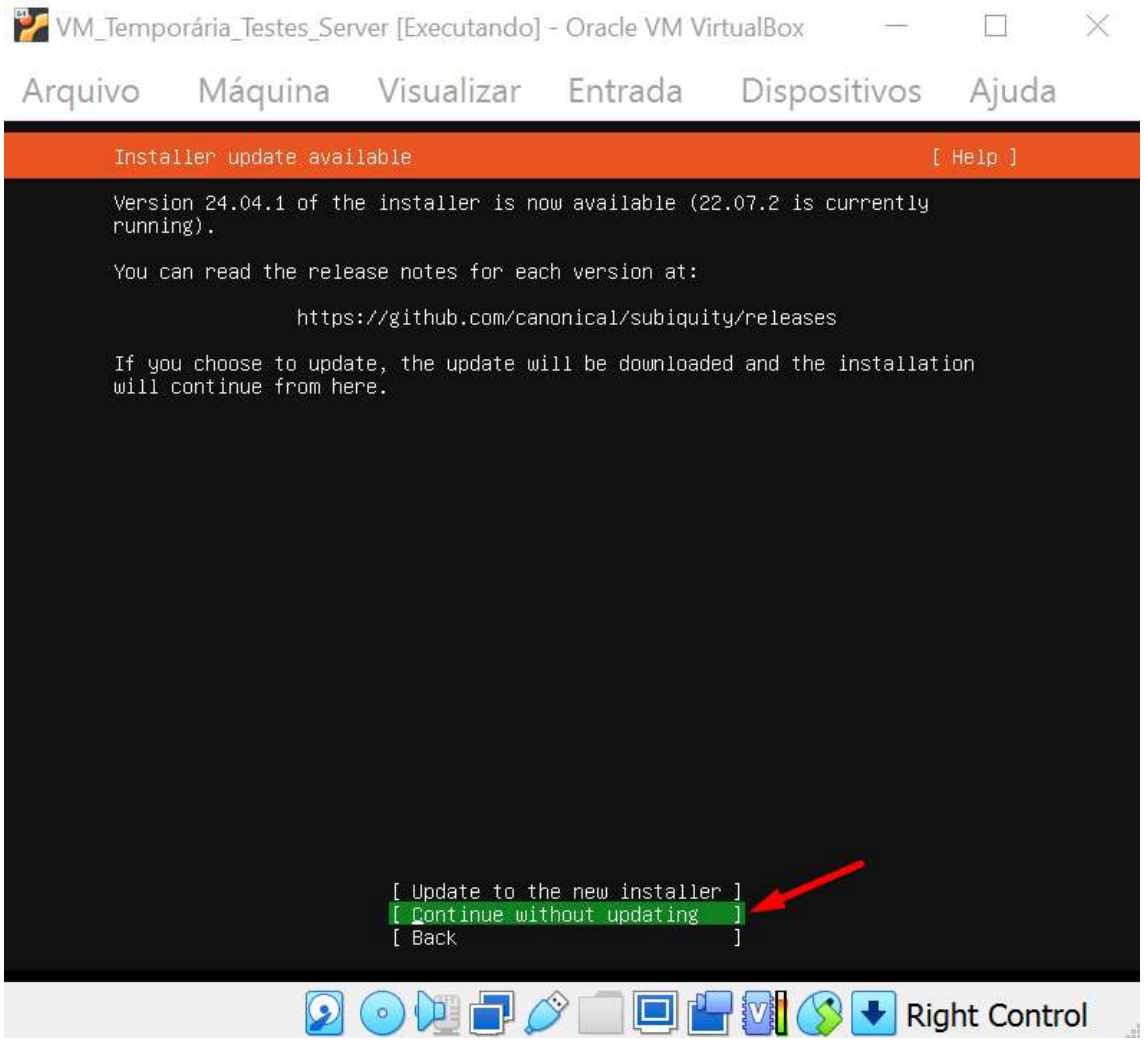
Após, selecione a Máquina Virtual e clique em INICIAR:

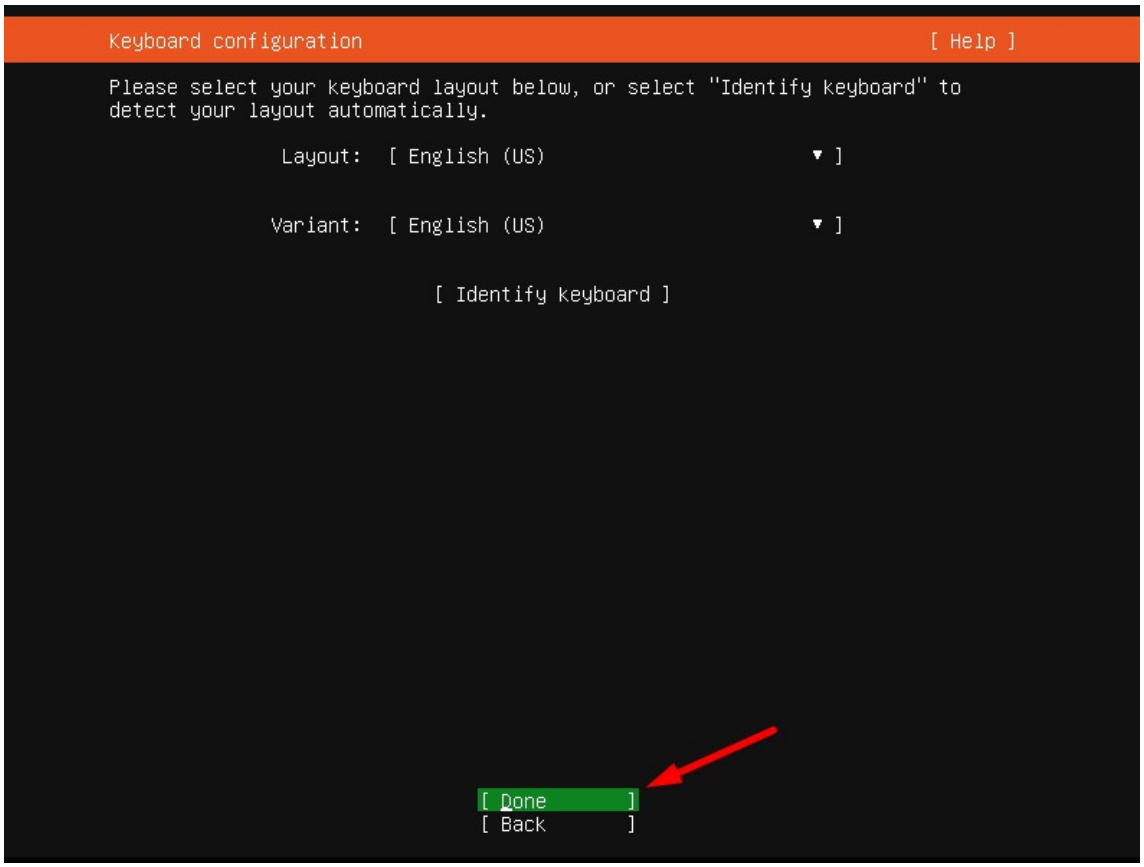

A instalação começará a ocorrer:

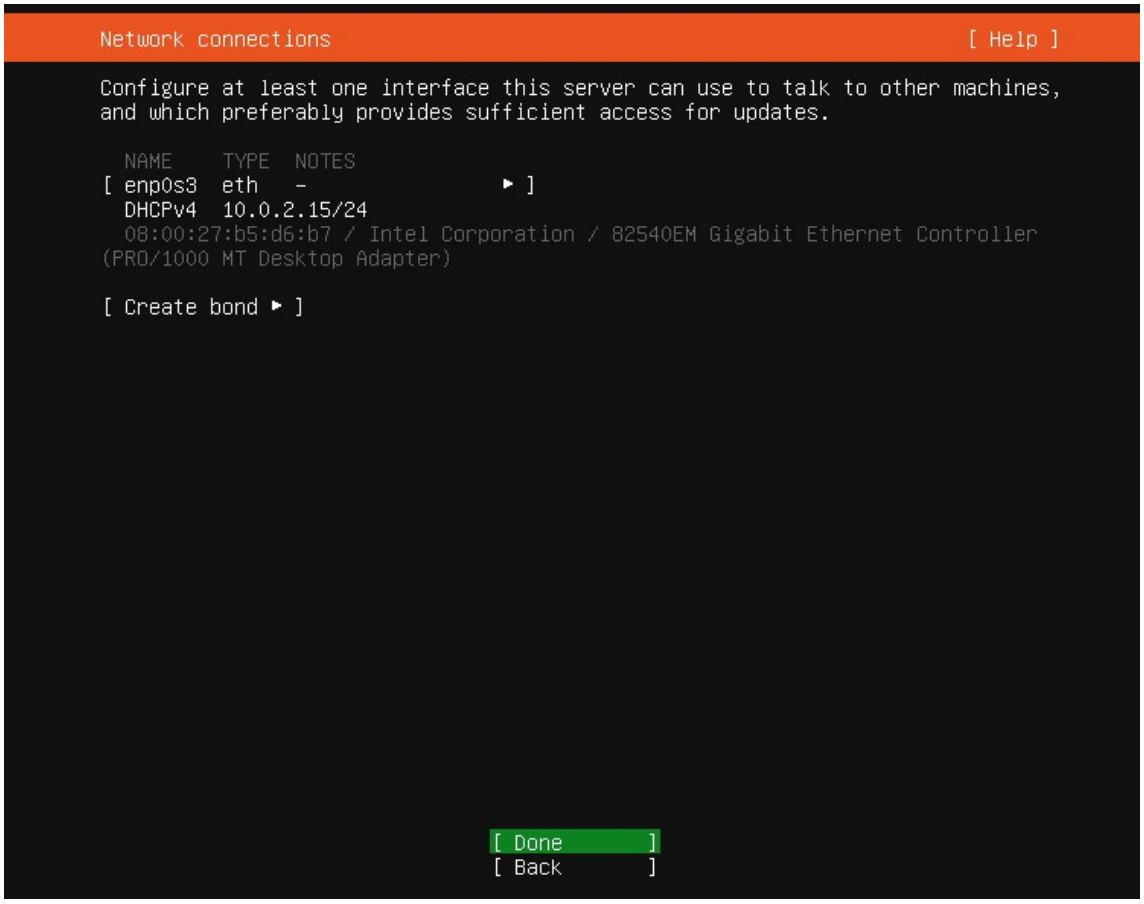


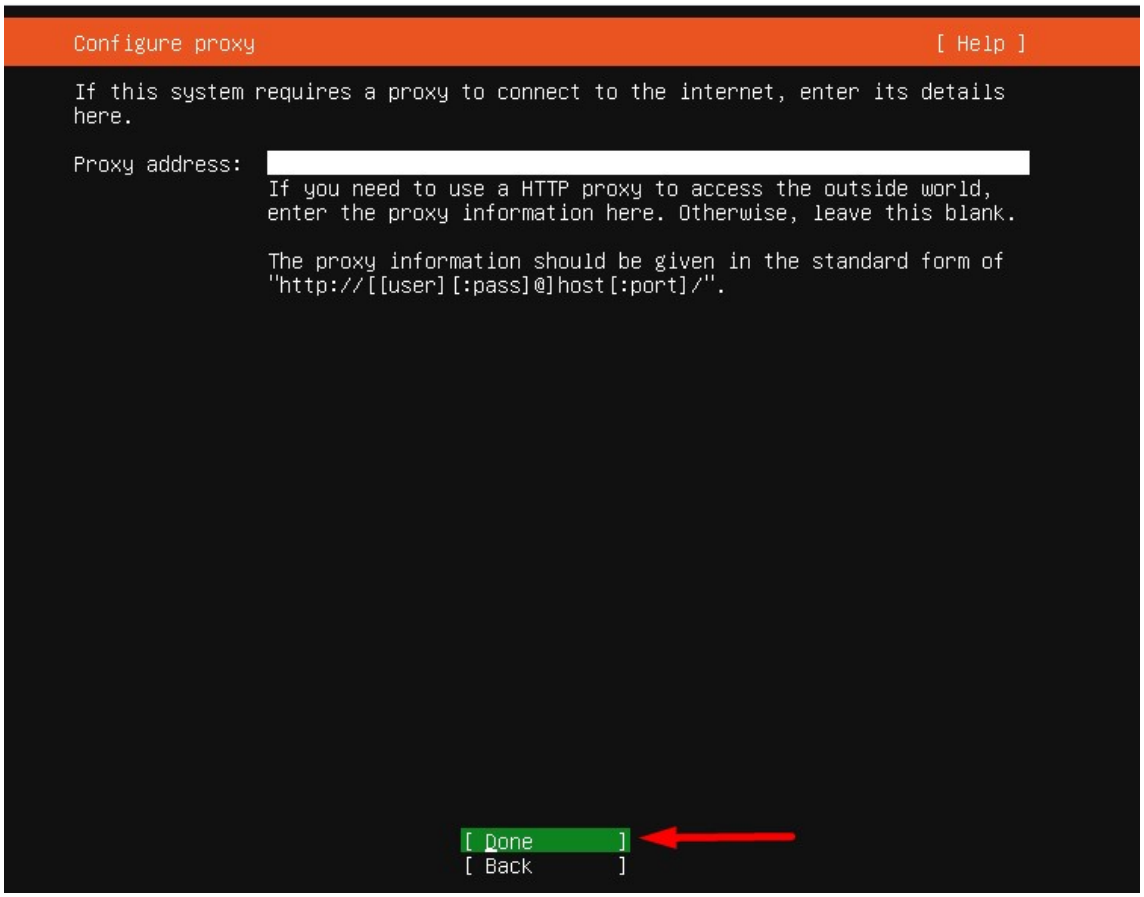


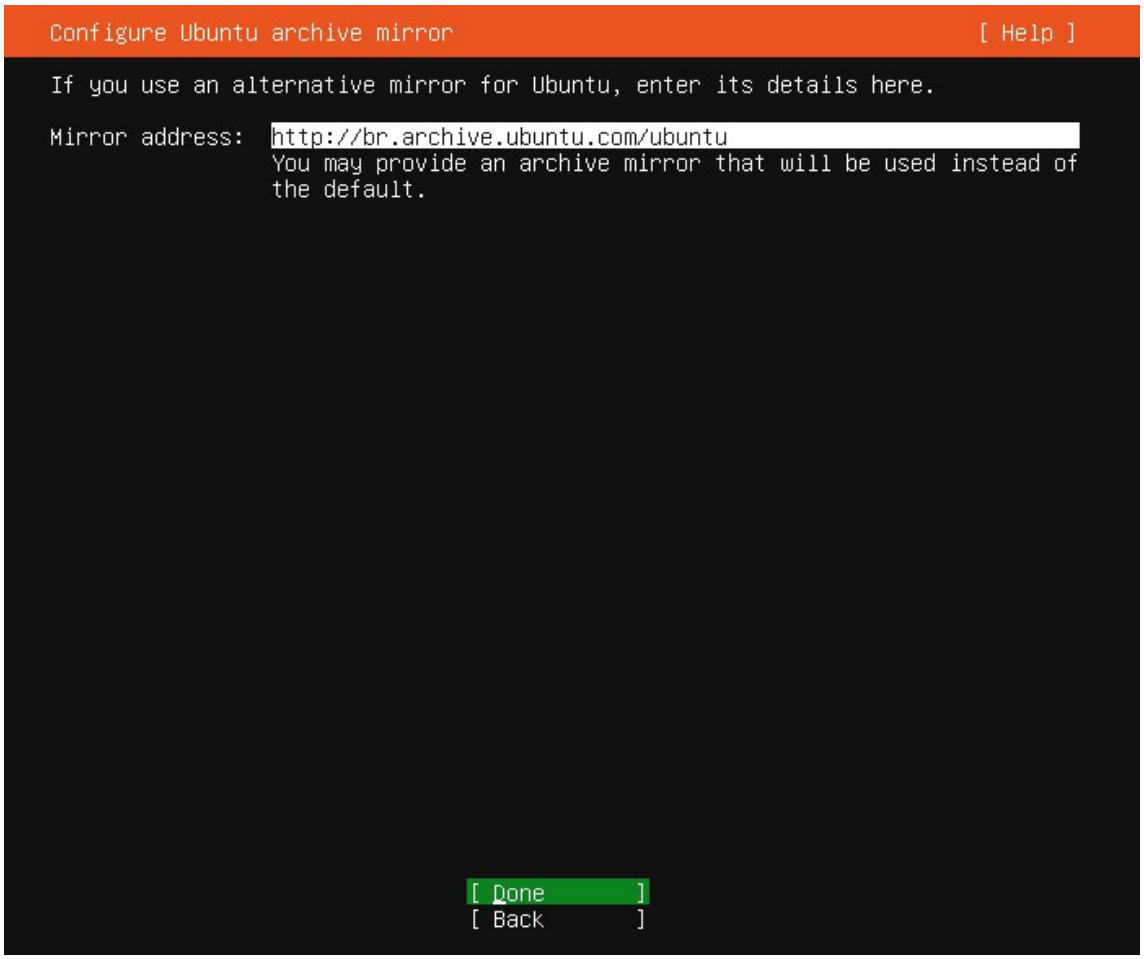
Pode se manter a configuração padrão...


VM_Temporária_Testes_Server [Executando] - Oracle VM VirtualBox
Arquivo Máquina Visualizar Entrada Dispositivos Ajuda
Installer update available [ Help ]
Version 24.04.1 of the installer is now available (22.07.2 is currently running).
You can read the release notes for each version at:
https://github.com/canonical/subiquity/releases
If you choose to update, the update will be downloaded and the installation will continue from here.
[ Update to the new installer ]
[ Continue without updating ]
[ Back ]
Right Control

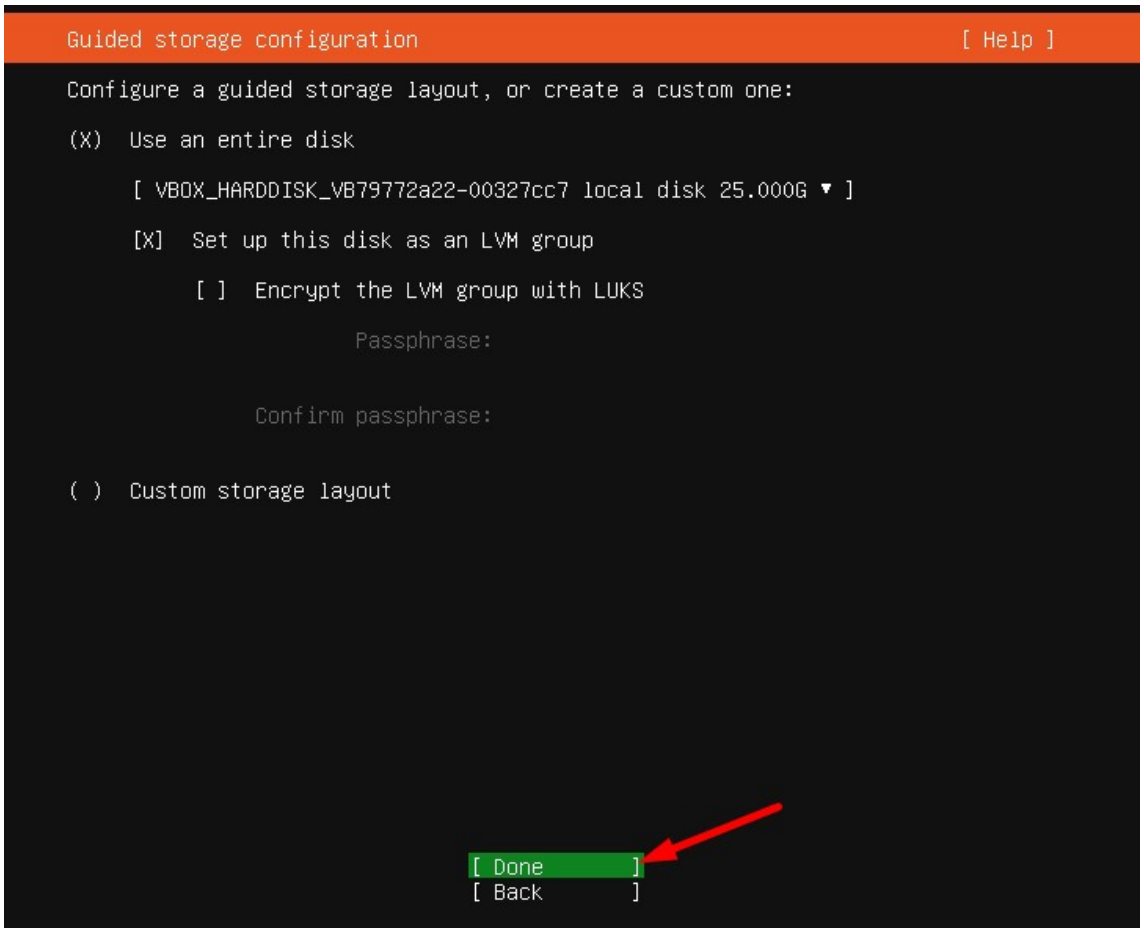

Keyboard configuration
[ Help ]
Please select your keyboard layout below, or select "Identify keyboard" to
detect your layout automatically.
Layout: [ English (US) ▾ ]
Variant: [ English (US) ▾ ]
[ Identify keyboard ]
[ Done ]
[ Back ]

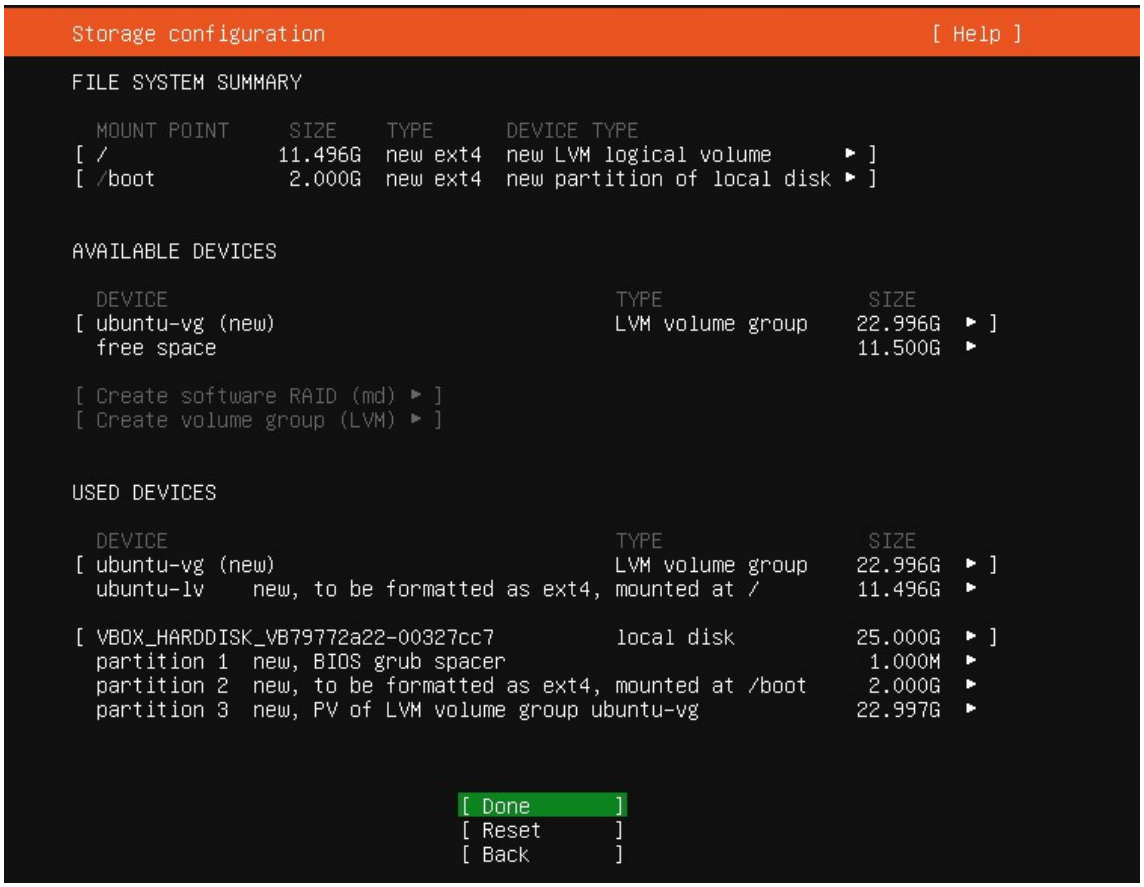

Network connections
[ Help ]
Configure at least one interface this server can use to talk to other machines,
and which preferably provides sufficient access for updates.
NAME TYPE NOTES
[ enp0s3 eth - ▸ ]
DHCPv4 10.0.2.15/24
08:00:27:b5:d6:b7 / Intel Corporation / 82540EM Gigabit Ethernet Controller
(PRO/1000 MT Desktop Adapter)
[ Create bond ▸ ]
[ Done ]
[ Back ]

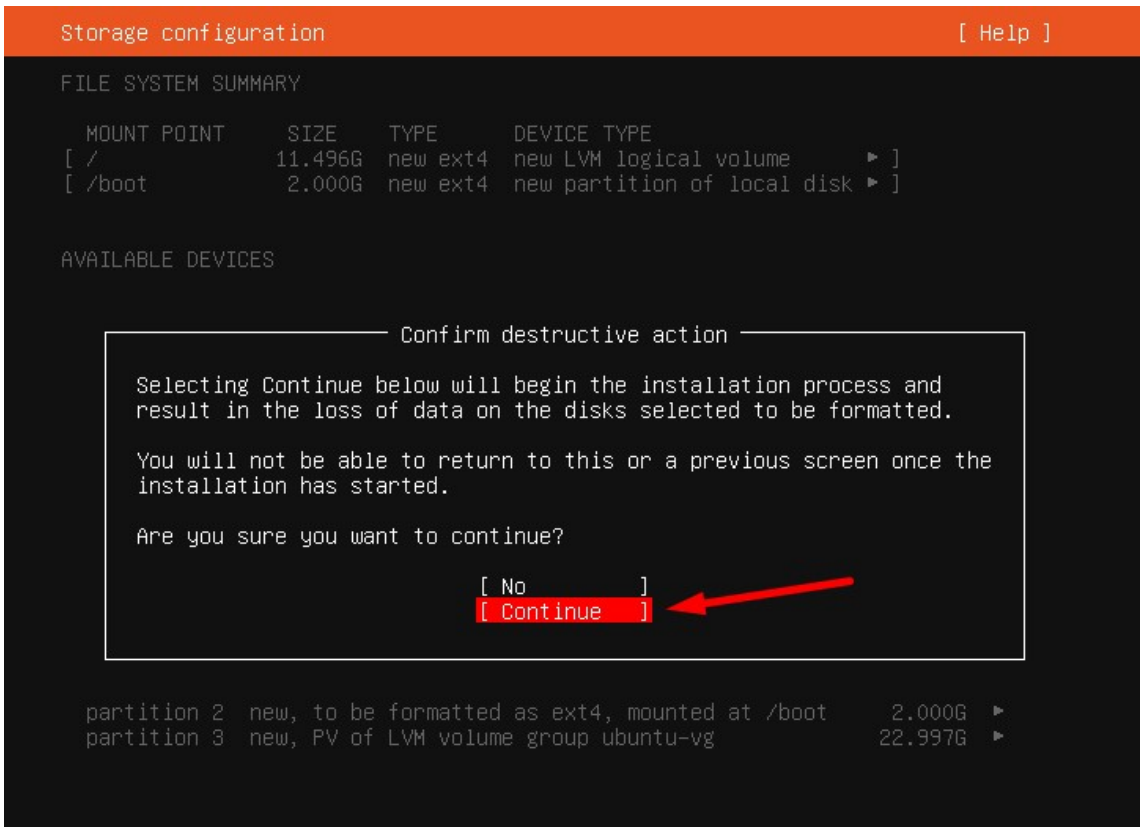

Configure proxy
[ Help ]
If this system requires a proxy to connect to the internet, enter its details here.
Proxy address:
If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.
The proxy information should be given in the standard form of "http://[[user][:pass]@]host[:port]/".
[ Done ]
[ Back ]


Configure Ubuntu archive mirror
[ Help ]
If you use an alternative mirror for Ubuntu, enter its details here.
Mirror address: http://br.archive.ubuntu.com/ubuntu
You may provide an archive mirror that will be used instead of the default.
[ Done ]
[ Back ]

Aperte o TAB até ficar na opção DONE:

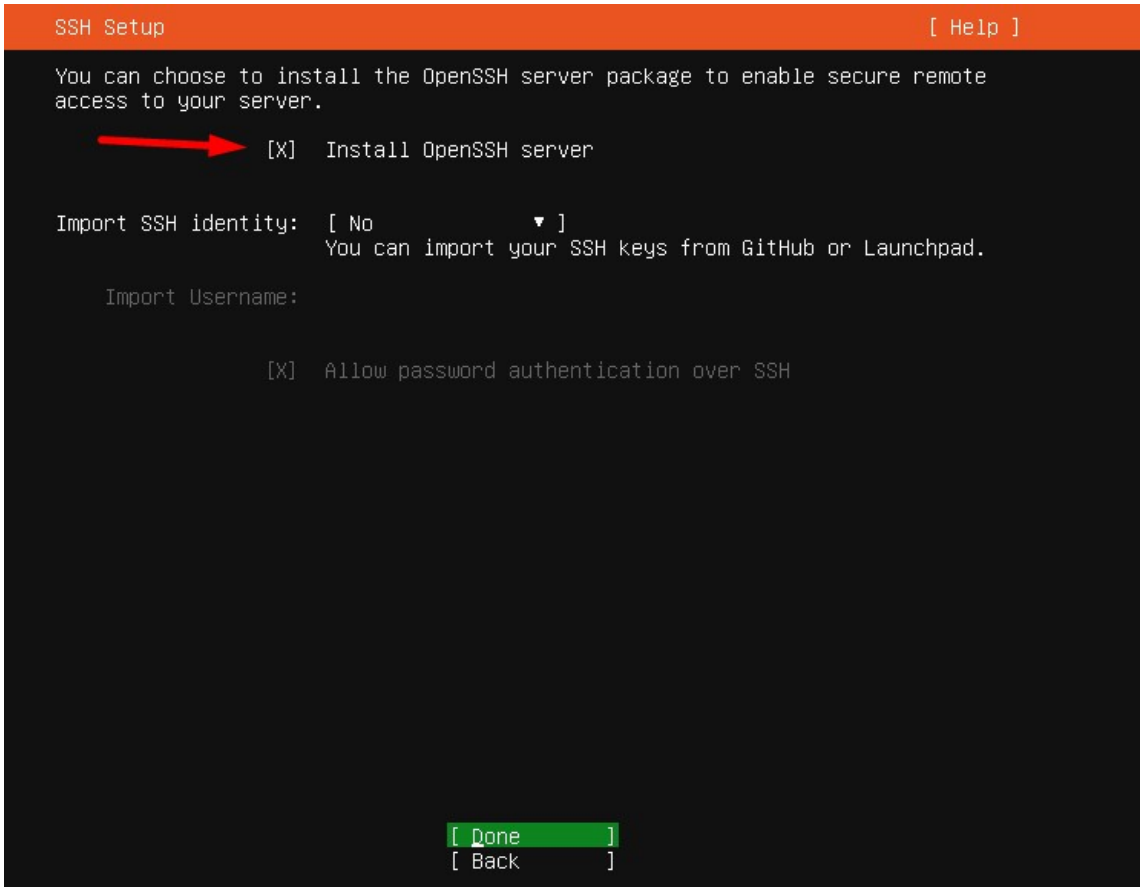



Storage configuration
[ Help ]
FILE SYSTEM SUMMARY
MOUNT POINT SIZE TYPE DEVICE TYPE
[ / 11.496G new ext4 new LVM logical volume ▸ ]
[ /boot 2.000G new ext4 new partition of local disk ▸ ]
AVAILABLE DEVICES
DEVICE TYPE SIZE
[ ubuntu-vg (new) LVM volume group 22.996G ▸ ]
free space 11.500G ▸
[ Create software RAID (md) ▸ ]
[ Create volume group (LVM) ▸ ]
USED DEVICES
DEVICE TYPE SIZE
[ ubuntu-vg (new) LVM volume group 22.996G ▸ ]
ubuntu-lv new, to be formatted as ext4, mounted at / 11.496G ▸
[ VBOX_HARDDISK_VB79772a22-00327cc7 local disk 25.000G ▸ ]
partition 1 new, BIOS grub spacer 1.000M ▸
partition 2 new, to be formatted as ext4, mounted at /boot 2.000G ▸
partition 3 new, PV of LVM volume group ubuntu-vg 22.997G ▸
[ Done ]
[ Reset ]
[ Back ]

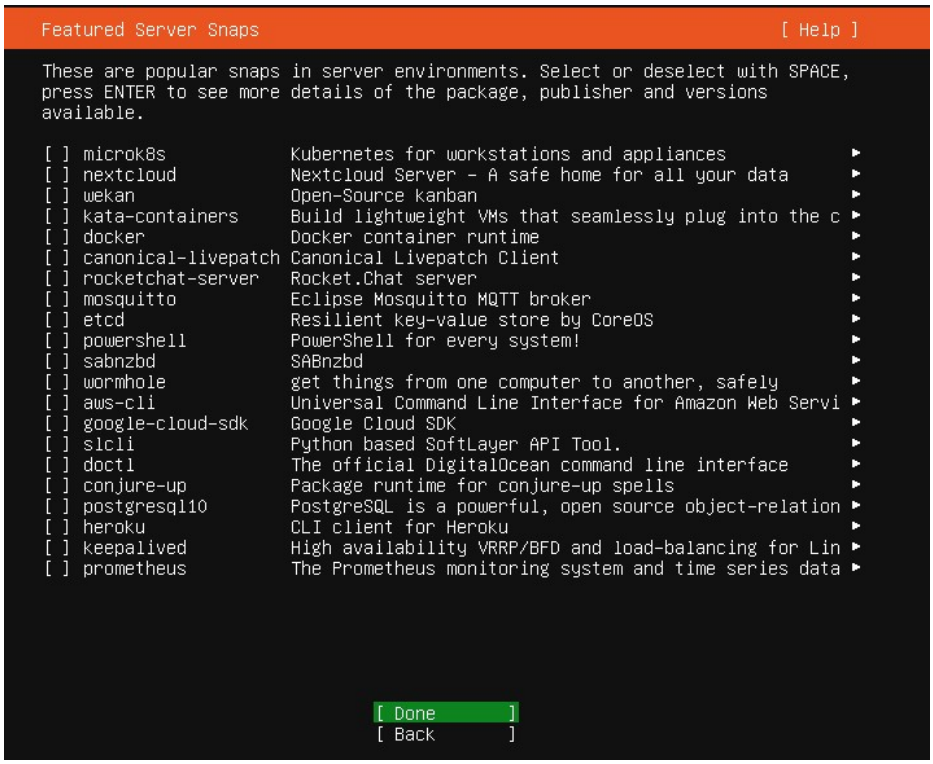

Storage configuration
[ Help ]
FILE SYSTEM SUMMARY
MOUNT POINT SIZE TYPE DEVICE TYPE
[ / 11.496G new ext4 new LVM logical volume ▸ ]
[ /boot 2.000G new ext4 new partition of local disk ▸ ]
AVAILABLE DEVICES
Confirm destructive action
Selecting Continue below will begin the installation process and result in the loss of data on the disks selected to be formatted.
You will not be able to return to this or a previous screen once the installation has started.
Are you sure you want to continue?
[ No ]
[ Continue ]
partition 2 new, to be formatted as ext4, mounted at /boot 2.000G ▸
partition 3 new, PV of LVM volume group ubuntu-vg 22.997G ▸

Informe suas credenciais e avance, semelhante a figura abaixo:

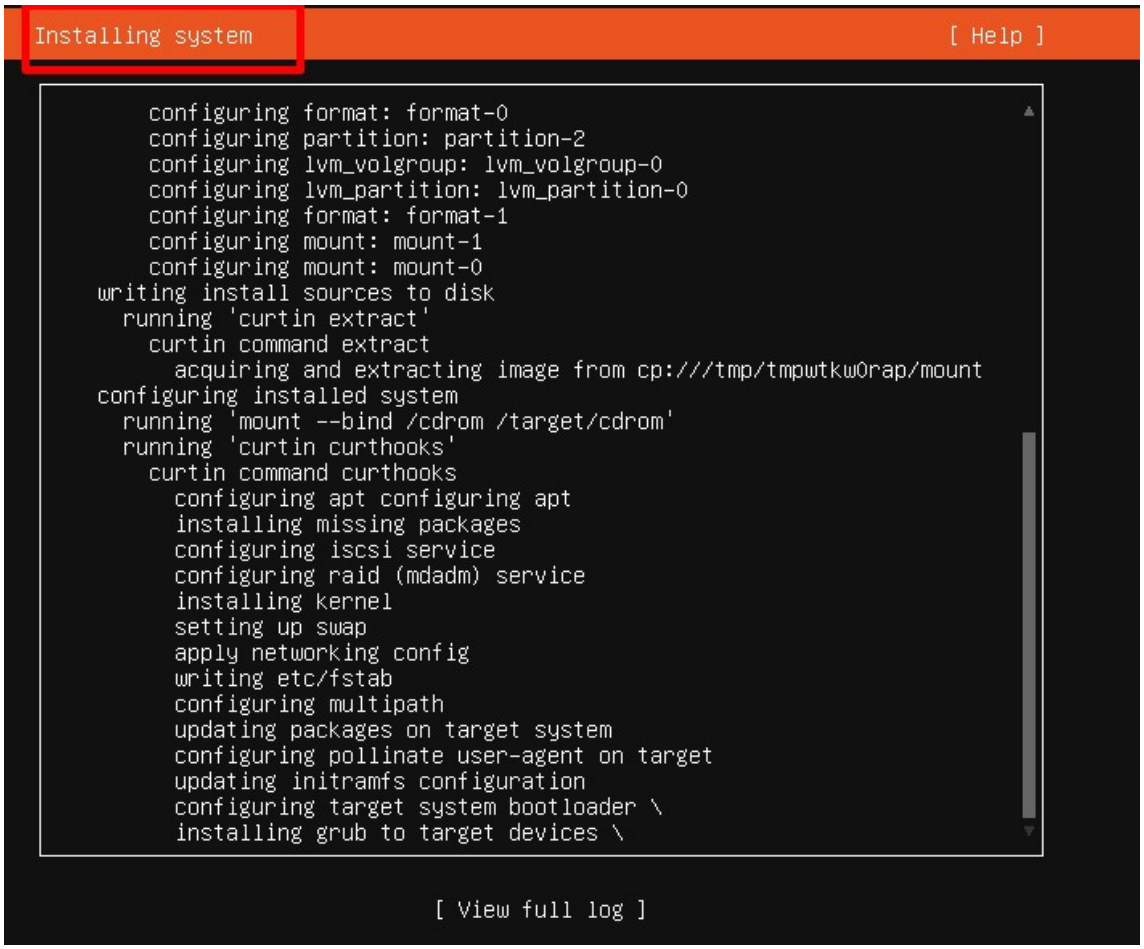

Você poderá selecionar a opção SSH antes de continuar:

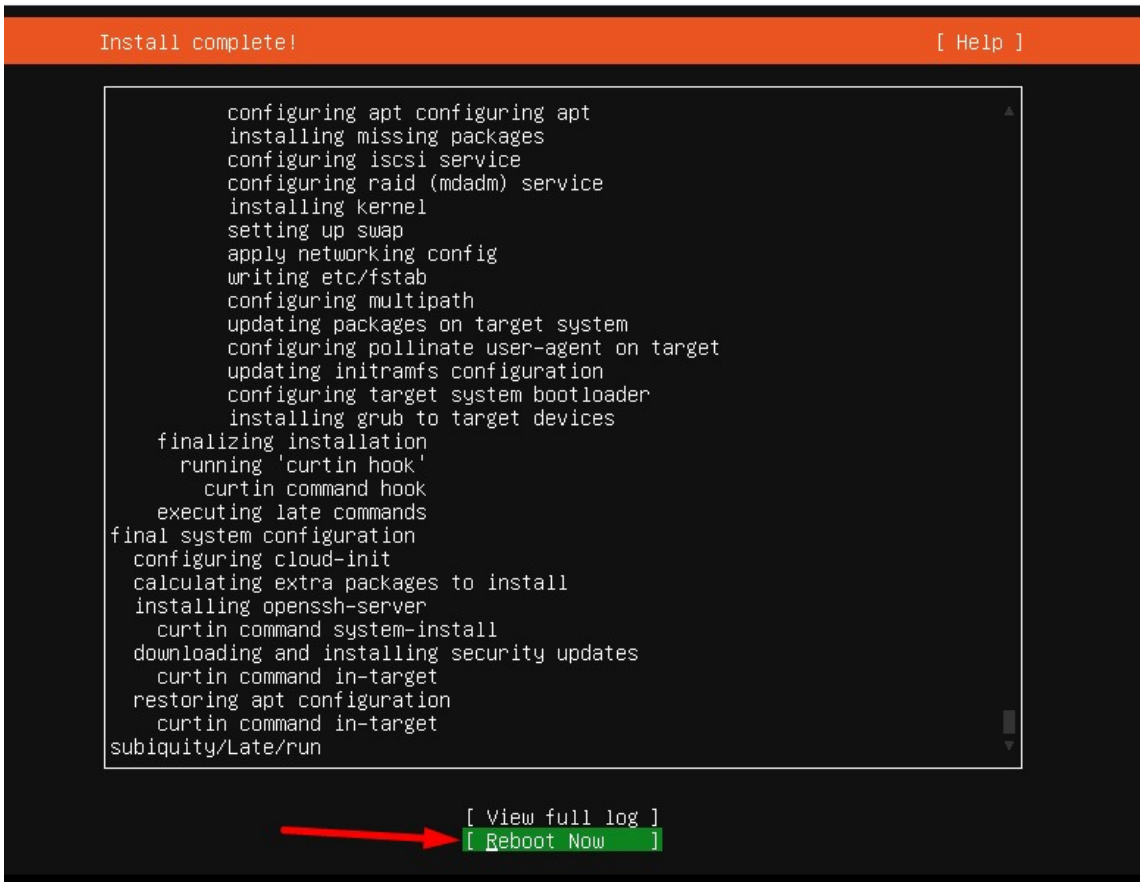


Poderá deixar como padrão e avançar



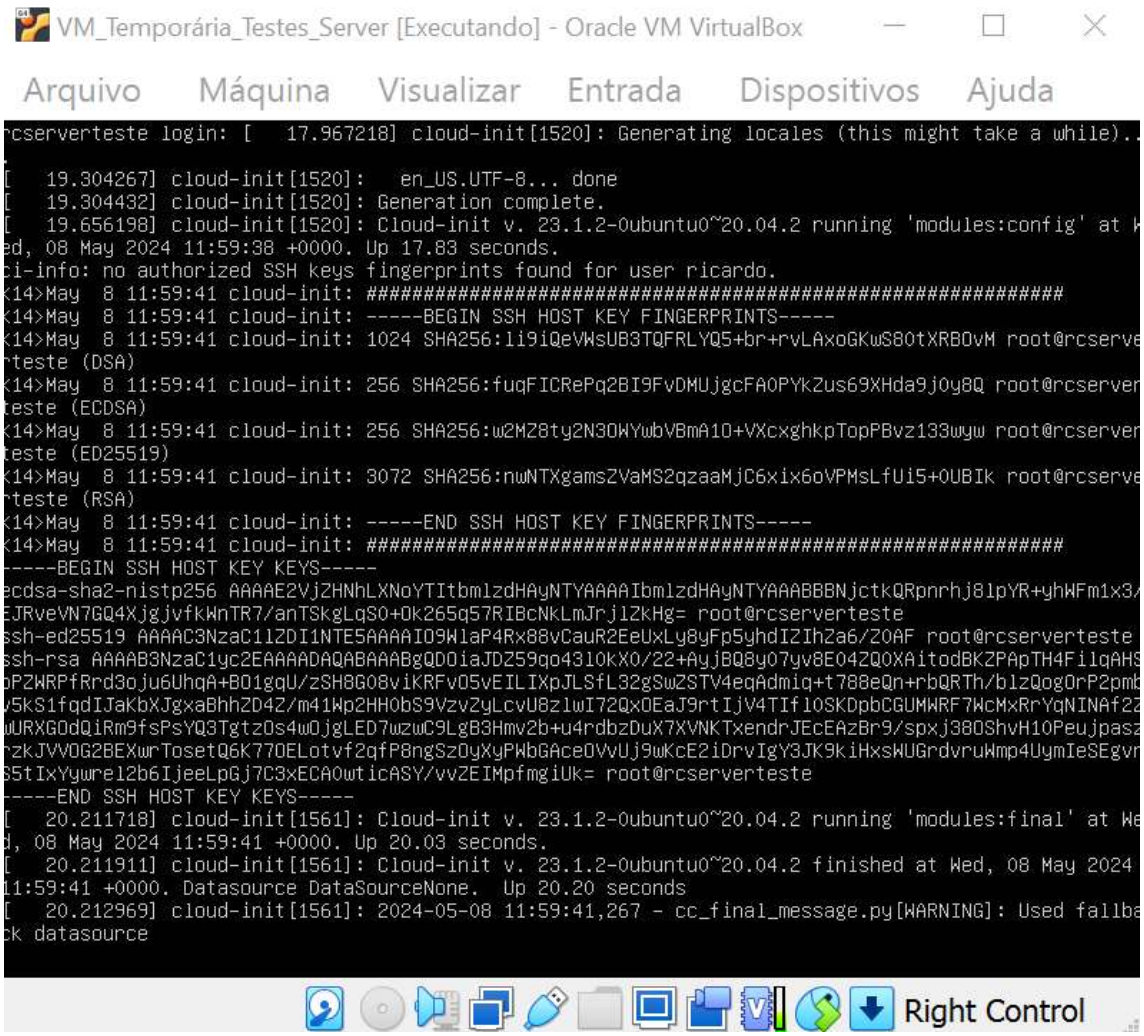

Começará a instalação:



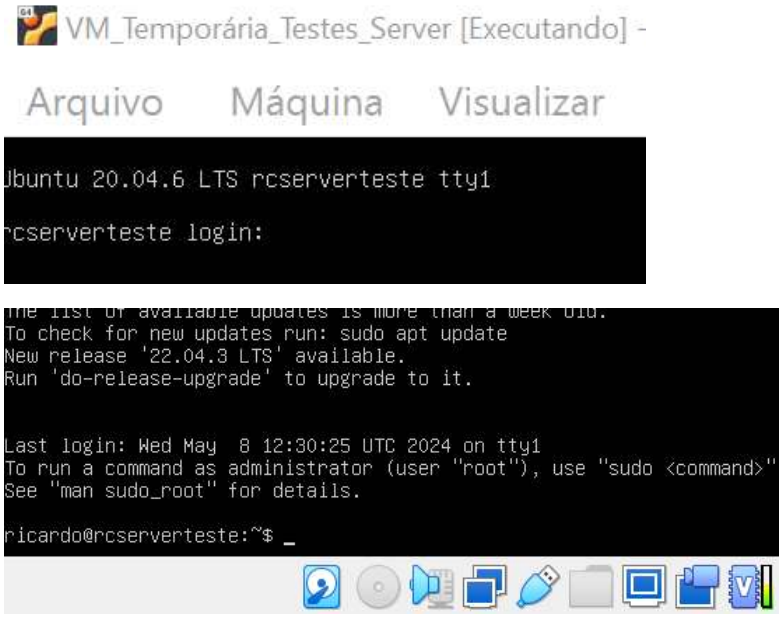



Se apresentar algum erro, aperte <ENTER> para continuar e “rebootar” o pc

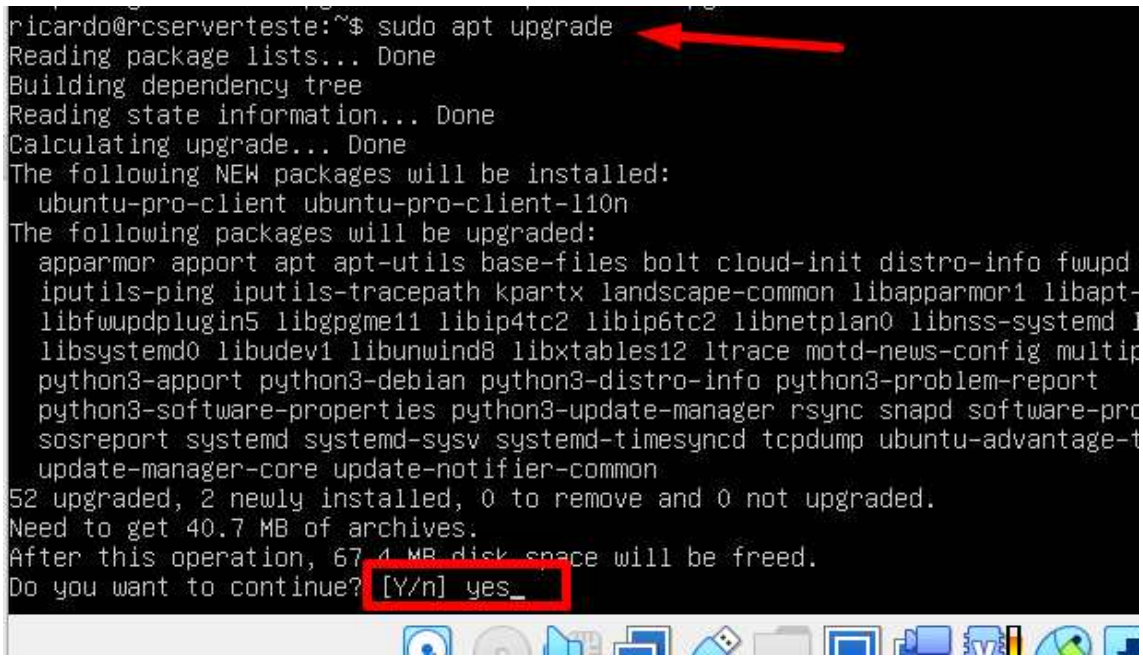
No final, aperte ENTER para logar no servidor com as credencias que você criou:

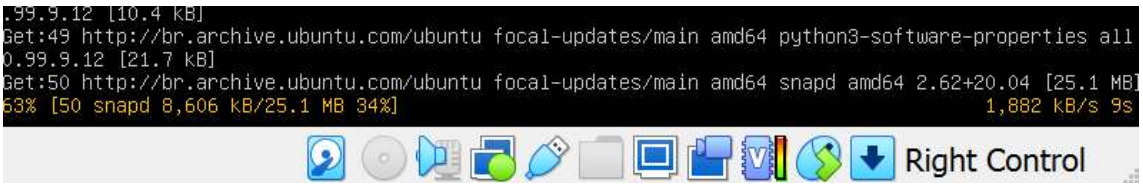



Agora, você poderá informar seu usuário e senha:



```
sudo apt update
sudo apt upgrade
```

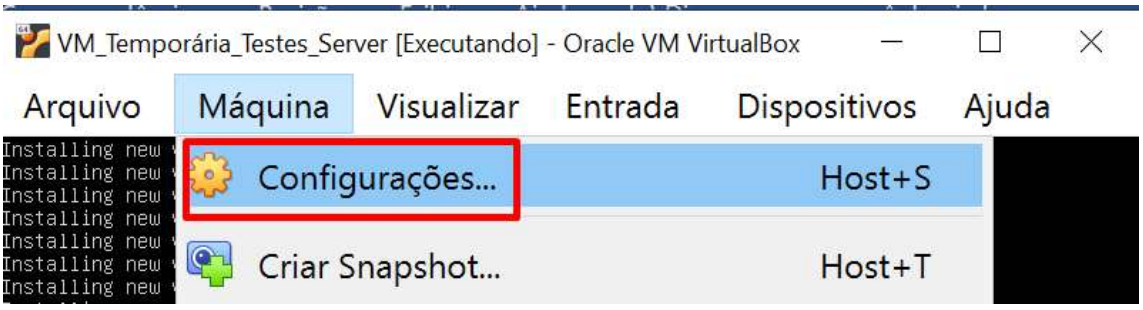


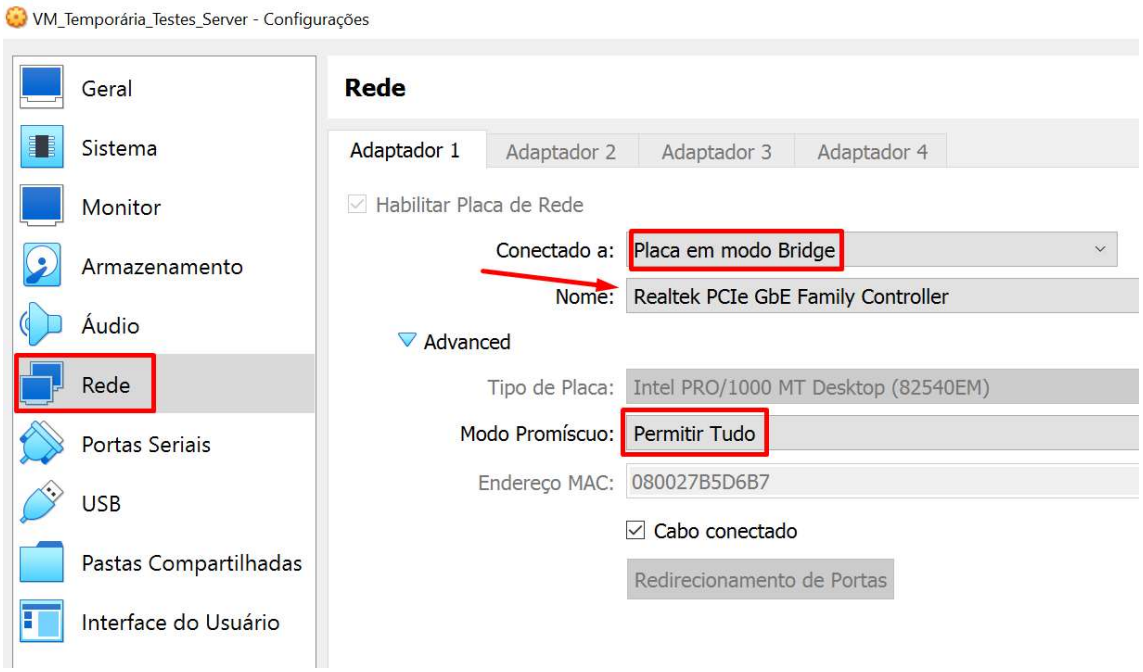




Uma alternativa para mexer na sua máquina virtual (Linux server) é acessar por SSH através do seu *prompt* de comando do Windows.

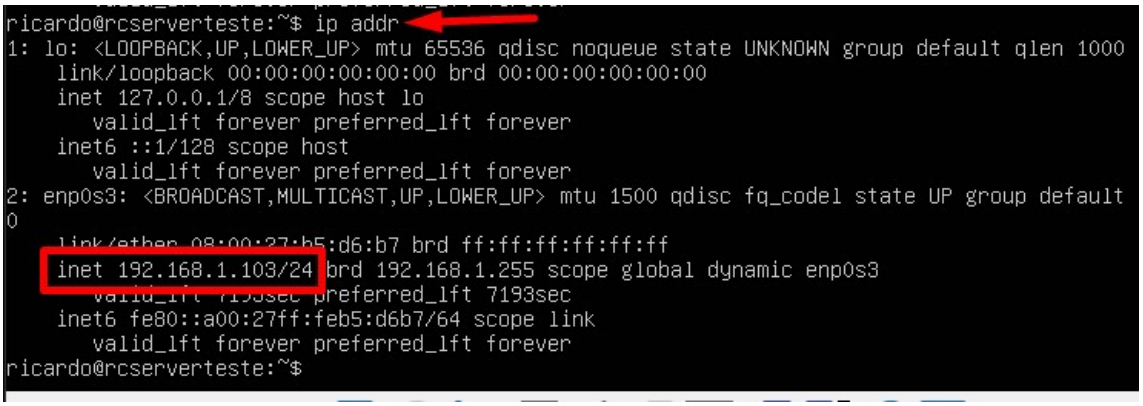
Vamos conferir, portanto, como está nossa configuração de REDE na máquina virtual.

Altere suas configurações, conforme exemplo abaixo:

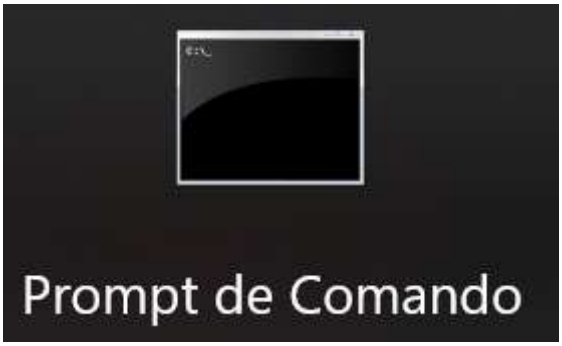





Antes de conectar, precisaremos qual é o IP do nosso Linux. Para isso execute o comando “ip addr”:

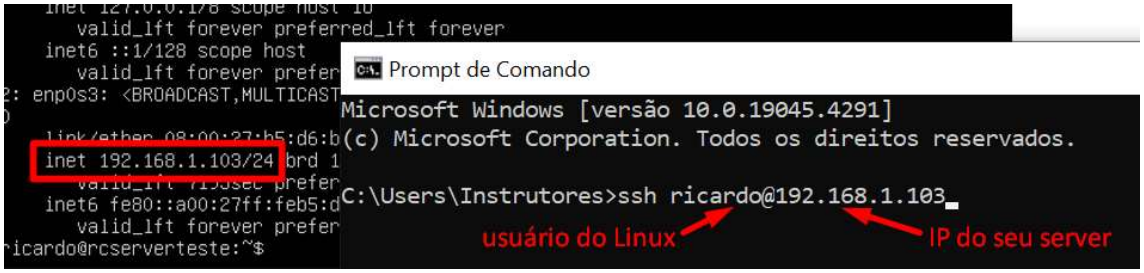



Para isso, basta acessar o prompt de comando e digitar o comando abaixo:

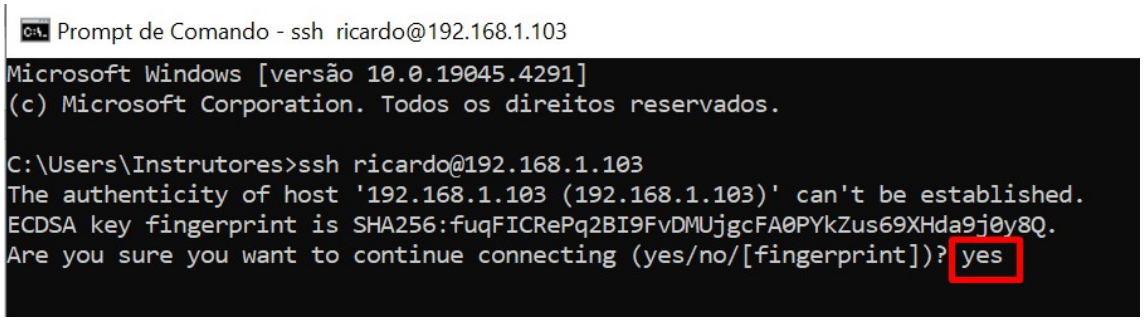



No Prompt, sigite ssh usuário_do_linux@numero_ip_do_seu_server

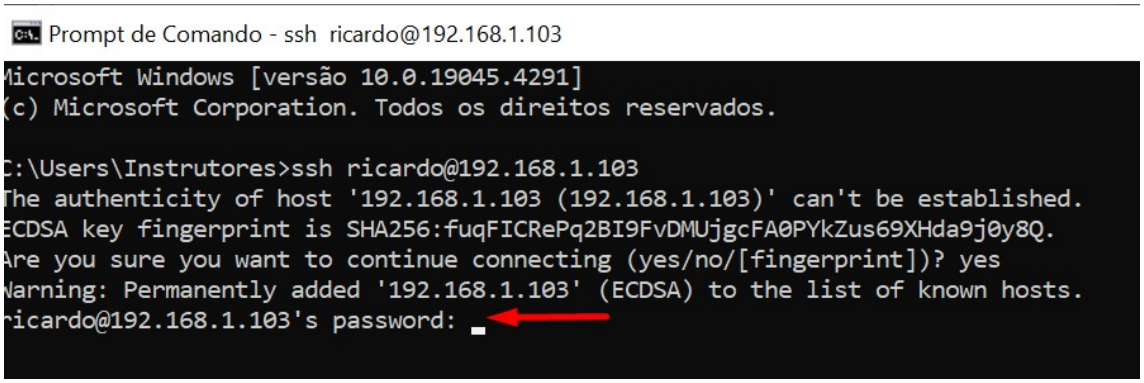
Por exemplo: ssh ricardo@192.168.1.103



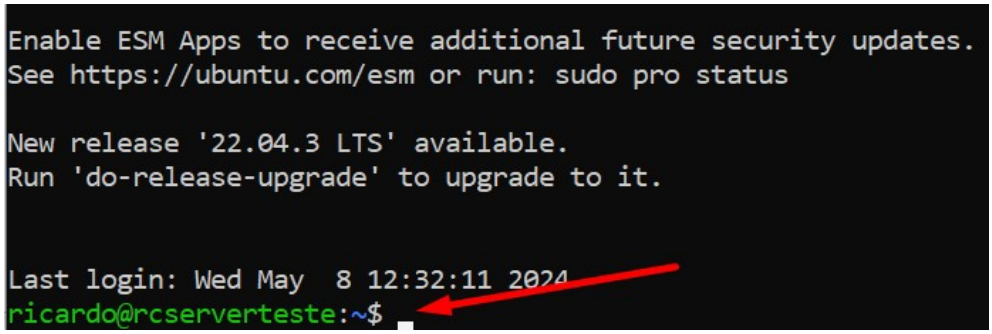

Informe “yes” e aperte <ENTER>:

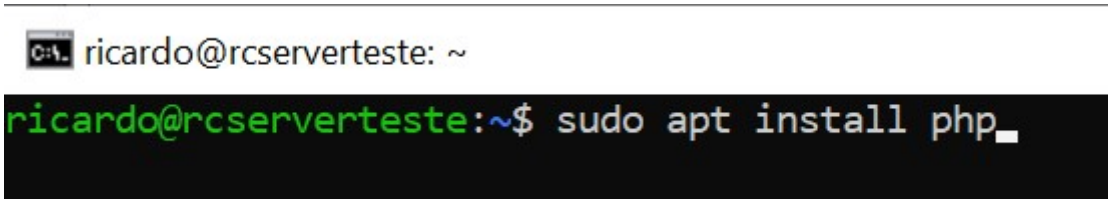



Digite sua senha do Linux:

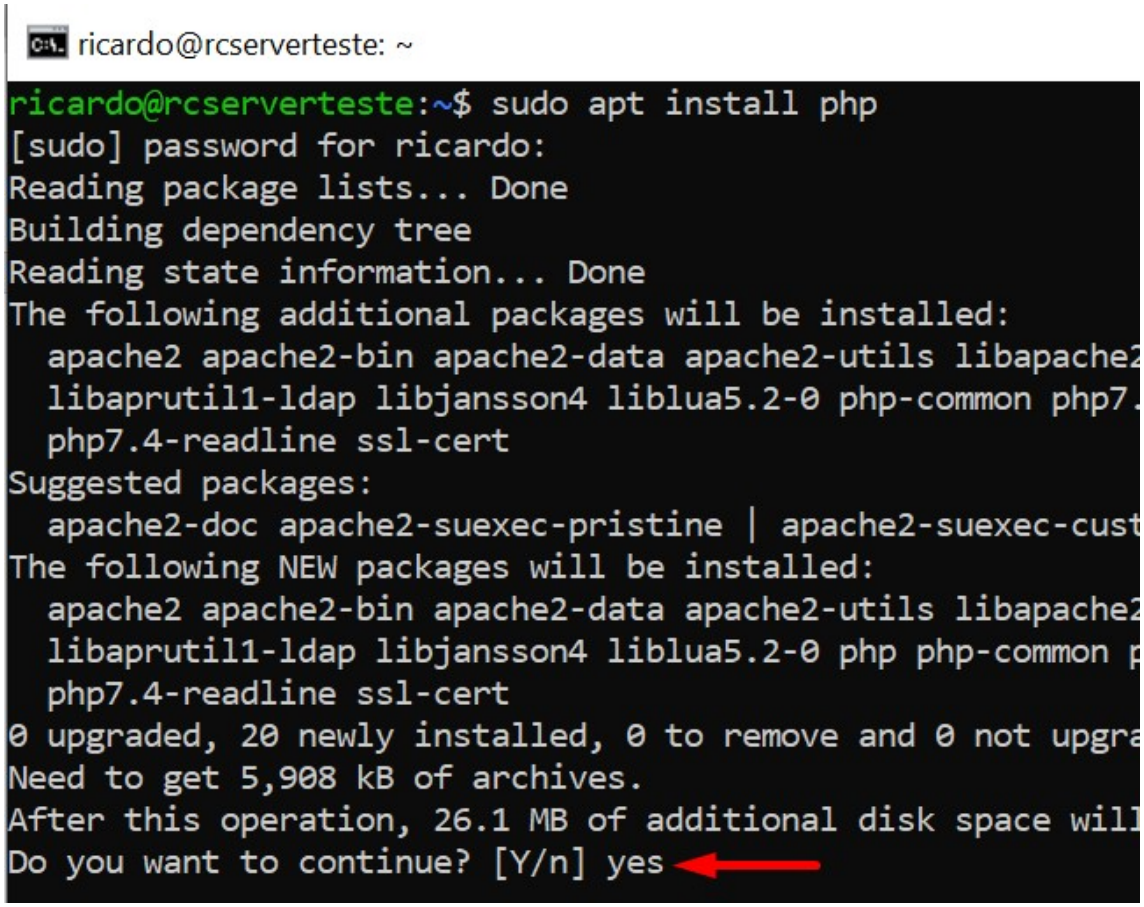



Se tudo der certo, aparecerá semelhante da tela abaixo:

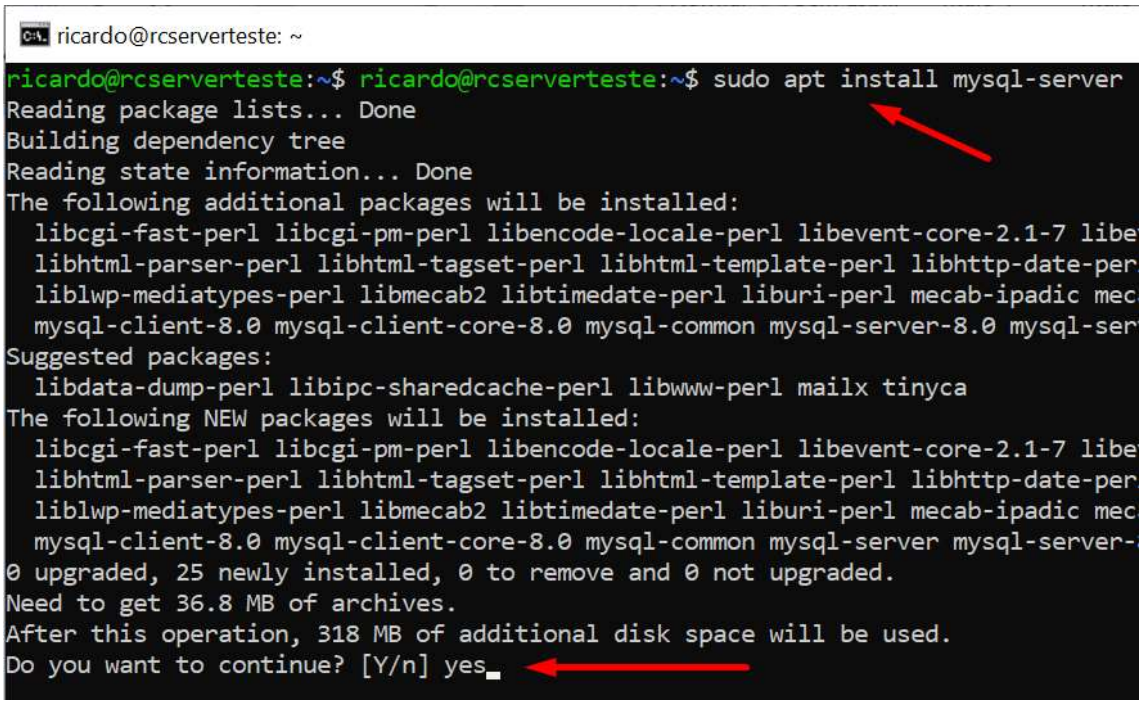




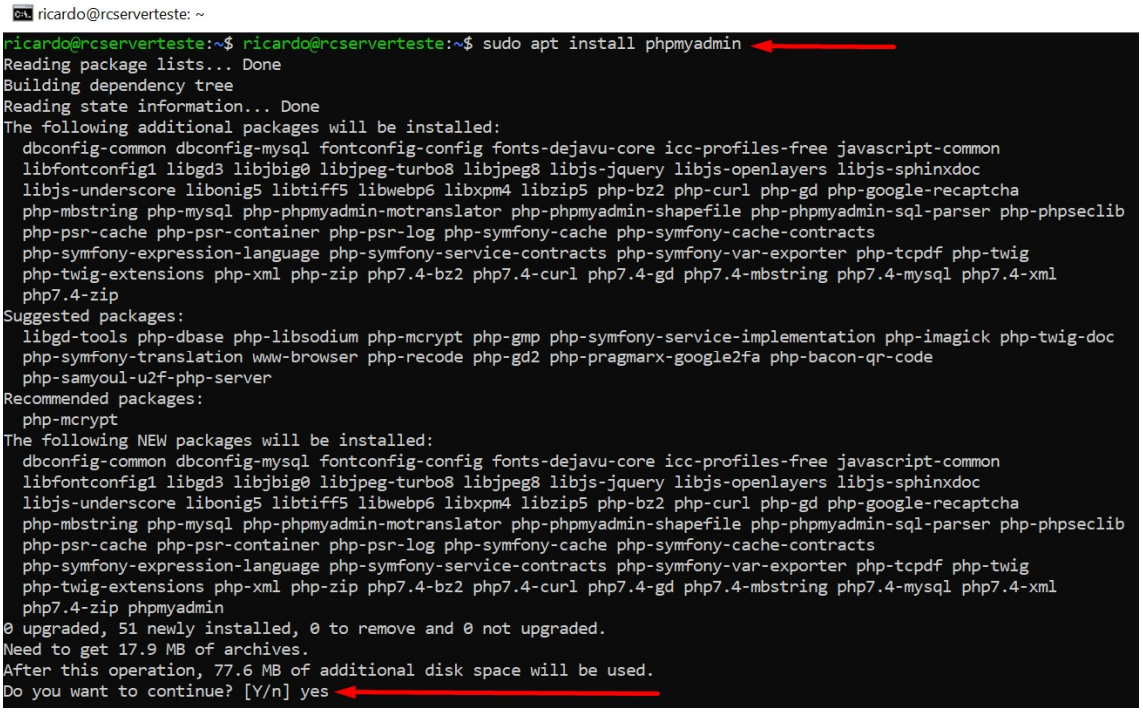

sudo apt install php



sudo apt install mysql-server

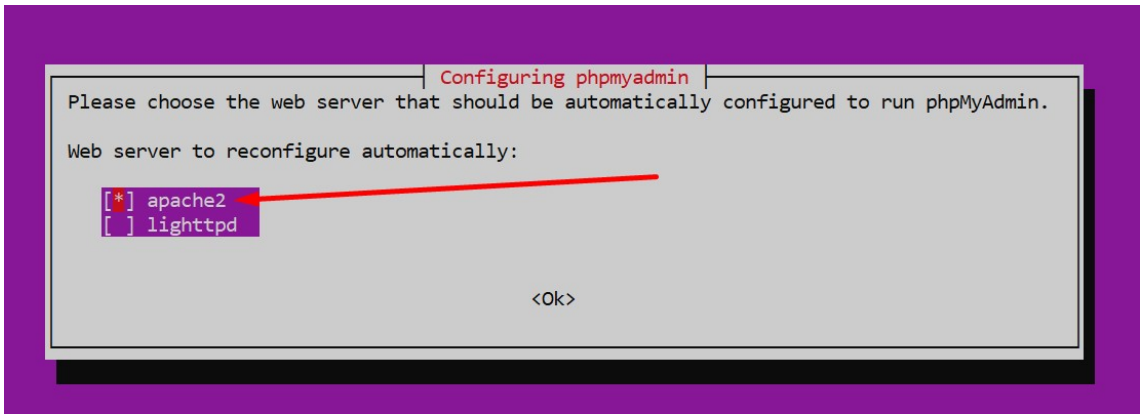


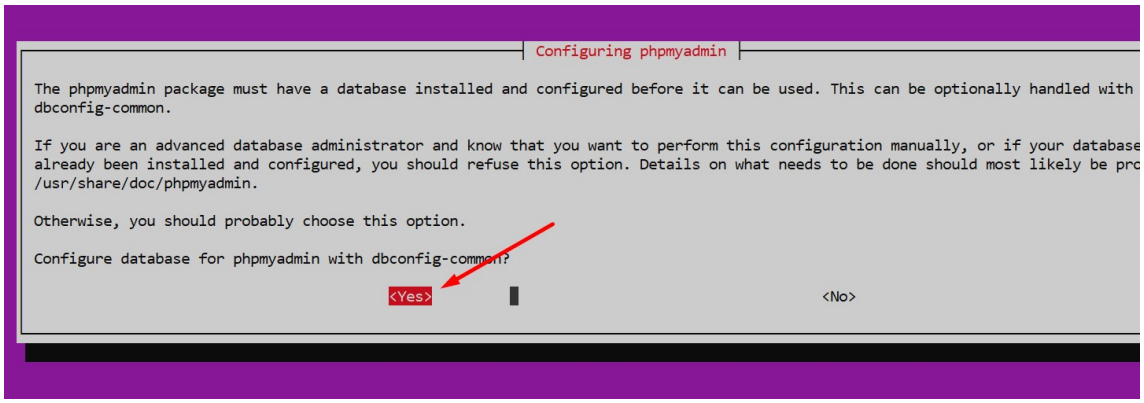

sudo apt install phpmyadminsudo



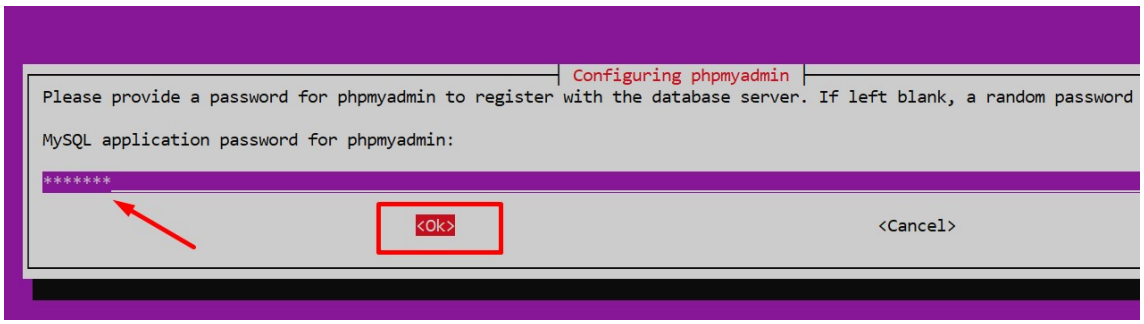

Se tudo der certo, veja se aparecerá a tela abaixo:

(selecione a opção apache2 – com a barra de espaço e com o TAB dê “OK”)

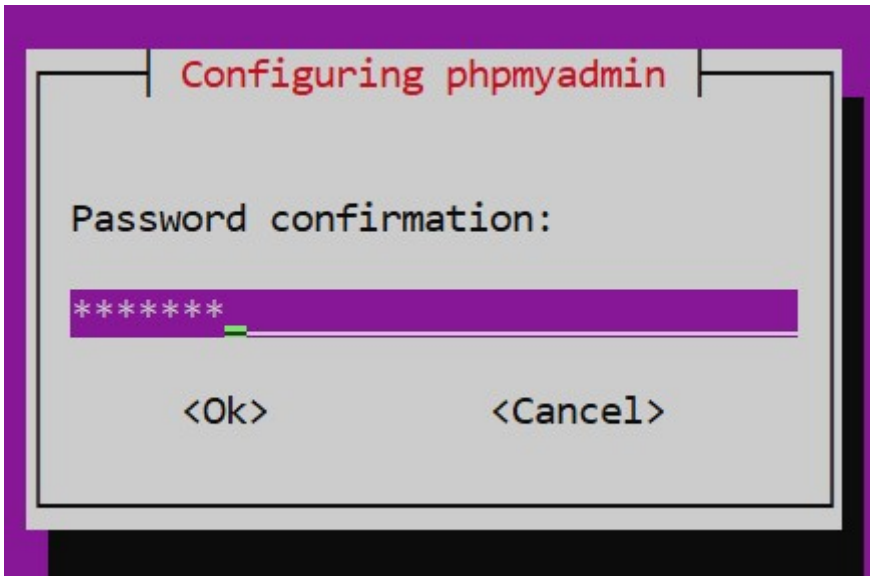

Configuring phpmyadmin
Please choose the web server that should be automatically configured to run phpMyAdmin.
Web server to reconfigure automatically:
[*] apache2
[ ] lighttpd
<Ok>




Informe a senha do seu banco mysql e depois clique em <OK>:

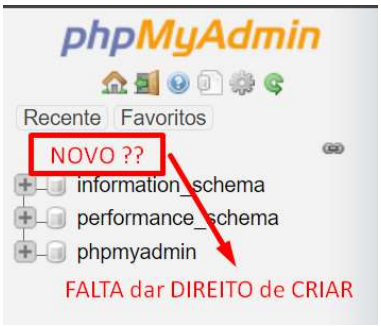



Repita a senha:




Agora, no seu navegador, teste o acesso ao phpmyadmin, informando o SEU_IP/phpmyadmin, semelhante à figura abaixo:

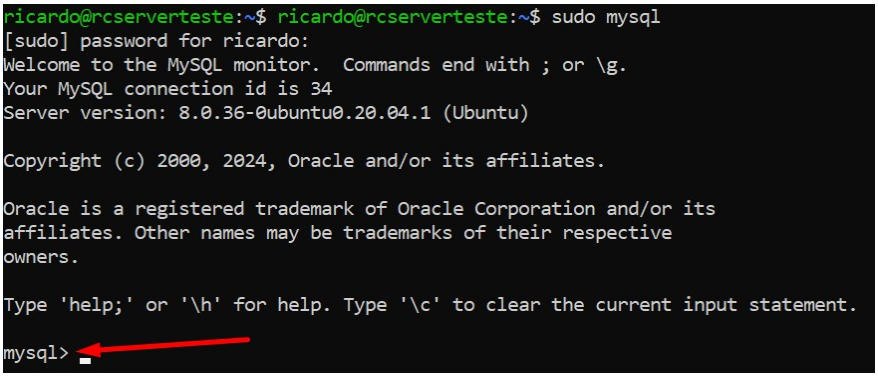

Porém quando logarmos, vamos perceber que você não terá direito de criar novos bancos de dados, será necessário dar esse direito.




Para dar esse direito, precisamos acessar o MYSQL via “terminal” e dar dois comandos:

sudo mysql


Ficará assim:



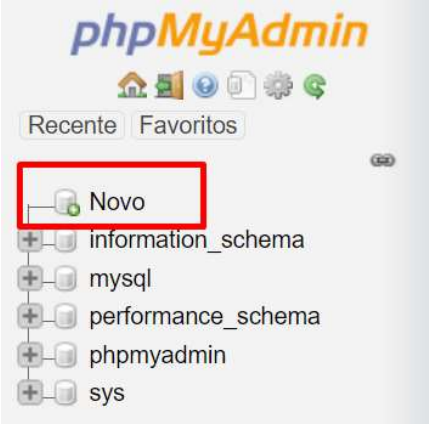
GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'localhost';

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'localhost';
Query OK, 0 rows affected (0.04 sec)

mysql>
```

FLUSH PRIVILEGES;

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Se tudo ocorreu bem, se você sair e entrar novamente no *phpmyadmin*, você observará que agora você poderá criar novos bancos:




Para sair do mysql, digite “exit”:

```
mysql> exit
Bye
ricardo@rcserverteste:~$
```

## *Até essa parte, preparamos nosso Linux para acessar o banco de dados, caso necessário.*

---

# *Parte II – Preparando nosso ambiente para NODEJS (roda na porta 3000)*

Para instalar o Node.js no Ubuntu Server, você pode seguir estes passos:

Atualize o Gerenciador de Pacotes: Antes de instalar o Node.js, é uma boa prática atualizar o gerenciador de pacotes apt para garantir que você tenha a versão mais recente dos pacotes disponíveis. Você pode fazer isso executando:

```
sudo apt update
```

Instale o Node.js: Você pode instalar o Node.js no Ubuntu de algumas maneiras diferentes. A maneira mais comum é usando o curl ou wget para baixar o script de instalação do Node.js e, em seguida, executá-lo. Usando o Curl:

Vamos realizar as instalações seguindo os passos abaixo:

```
wget -qO- https://deb.nodesource.com/setup_lts.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs
```

Este script adiciona o repositório do Node.js ao seu sistema e, em seguida, instala o Node.js e o npm (gerenciador de pacotes do Node.js).

Verifique a instalação: Depois de instalar o Node.js, você pode verificar se foi instalado corretamente e a versão instalada usando os seguintes comandos:

```
node -v
```

```
npm -v
```

```
ricardo@rcserverteste:~$ node -v
npm -v
v20.13.0
ricardo@rcserverteste:~$ npm -v
10.5.2
ricardo@rcserverteste:~$
```

Agora vamos criar um diretório e deixar nosso NODE RODANDO:

```
mkdir Aula0905-08h50
```

```
cd Aula0905-08h50/
```

```
npm init -y
```

```
ricardo@rcserverteste: ~/Aula0905-08h50
ricardo@rcserverteste:~$ node -v
v20.13.0
ricardo@rcserverteste:~$ npm -v
10.5.2
ricardo@rcserverteste:~$ mkdir Aula0905-08h50
ricardo@rcserverteste:~$ cd Aula0905-08h50/
ricardo@rcserverteste:~/Aula0905-08h50$ npm init -y
Wrote to /home/ricardo/Aula0905-08h50/package.json:

{
  "name": "aula0905-08h50",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}


ricardo@rcserverteste:~/Aula0905-08h50$
```

npm install express body-parse fs

```
ricardo@rcserverteste:~/Aula0905-08h50$ npm install express body-parse fs

added 66 packages, and audited 67 packages in 5s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ricardo@rcserverteste:~/Aula0905-08h50$
```

npm install cors

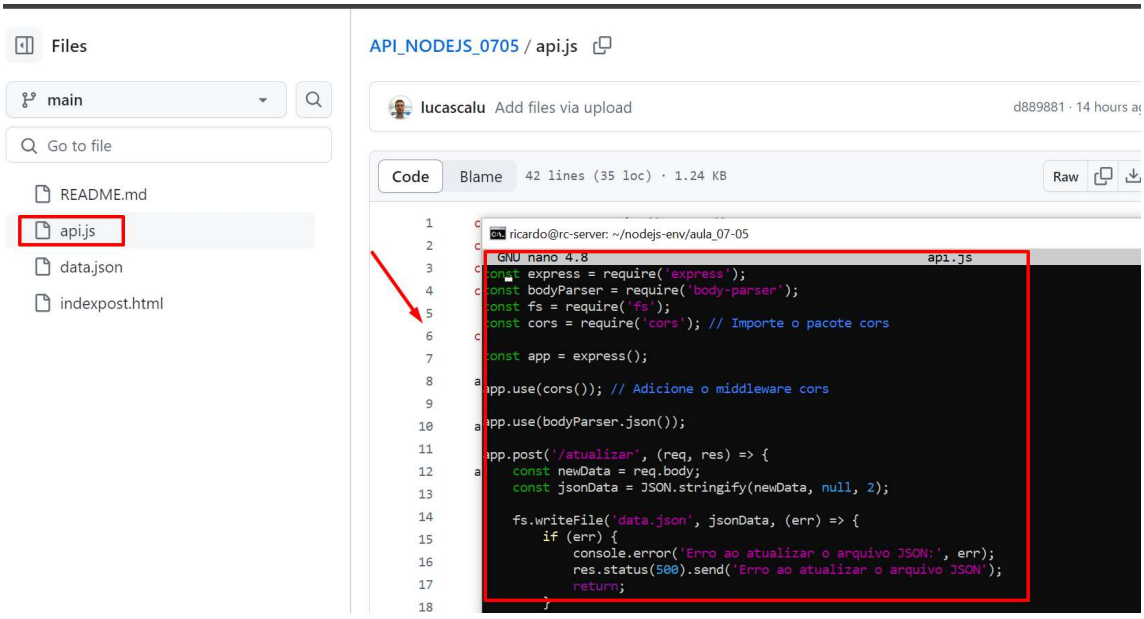
```
ricardo@rcserverteste:~/Aula0905-08h50$ npm install cors

added 2 packages, and audited 69 packages in 1s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
ricardo@rcserverteste:~/Aula0905-08h50$
```

Agora precisamos copiar o código App.js que está no GIT:

https://github.com/lucascalu/API_NODEJS_0705

No arquivo “api.js”, copie o código e **cole no nano**:

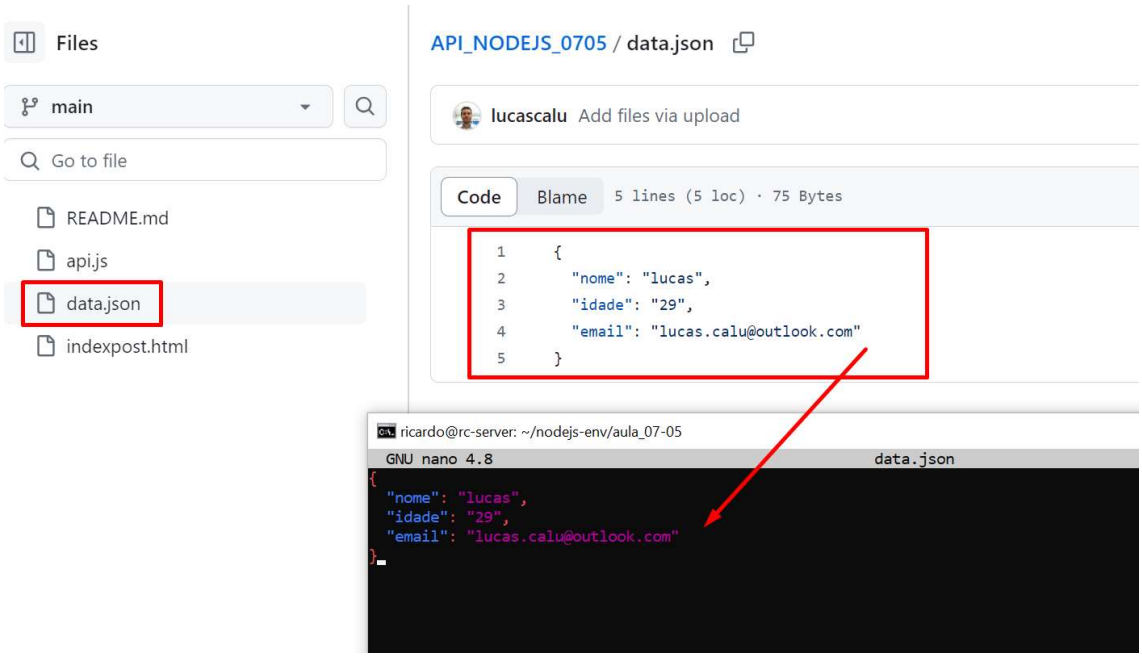
nano api.js

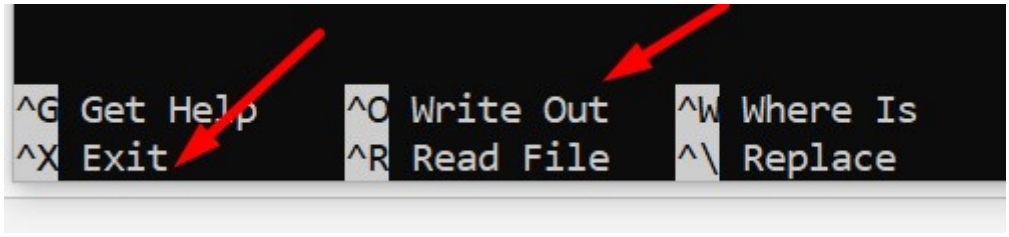



Aperte o CTRL+ O & CTRL + X para salvar e sair

Copie também o arquivo “data.json”:

nano data.json

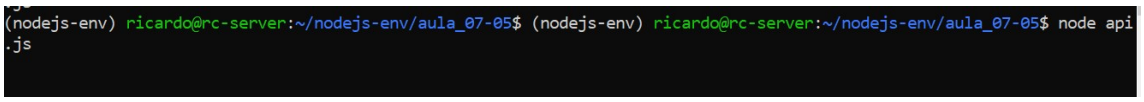


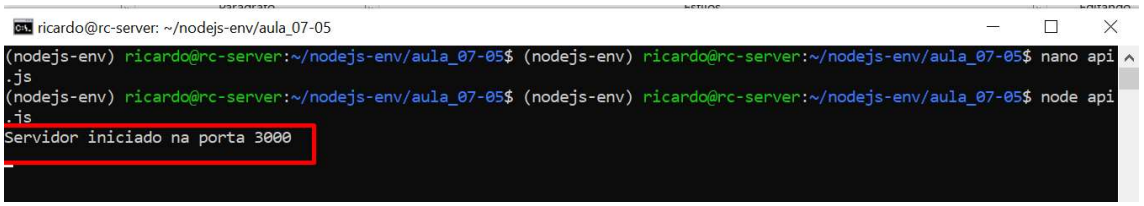

Salve e saia...

Para rodar a aplicação, faça:

node api.js




---




O texto “servidor iniciado na porta 3000” indica que está pronto, esperando que você abra o navegador e digite:


**(apenas para entender, instruções abaixo):**

Usamos o endereço do nosso servidor Ubuntu (o seu server) acrescentando :3000 (porta usada) e "/dados" pois está configurado no arquivo "api.js" conforme imagem abaixo:

API_NODEJS_0705 / api.js

Code | Blame | 42 lines (35 loc) · 1.24 KB

```
            console.error('Erro ao atualizar o arquivo JSON:', err);
            res.status(500).send('Erro ao atualizar o arquivo JSON');
            return;
        }
        console.log('Arquivo JSON atualizado com sucesso');
        res.send('Arquivo JSON atualizado com sucesso');
    });
});

app.get('/dados', (req, res) => {
    fs.readFile('data.json', 'utf8', (err, data) => {
        if (err) {
            console.error('Erro ao ler o arquivo JSON:', err);
            res.status(500).send('Erro ao ler o arquivo JSON');
            return;
        }
        const jsonData = JSON.parse(data);
        res.json(jsonData);
    });
});
```

(a imagem acima apenas demonstra o motivo pelo qual nós colocamos "/dados" no navegador).

## *Até essa parte, preparamos o NODEJS*

# *Parte III – Preparando nosso ambiente para REACT NATIVE (roda na porta 19006)*

=x=x=x=x=x=x=x=x=x=x=x=x=x=x

***Vamos entender o que fizemos até agora:***

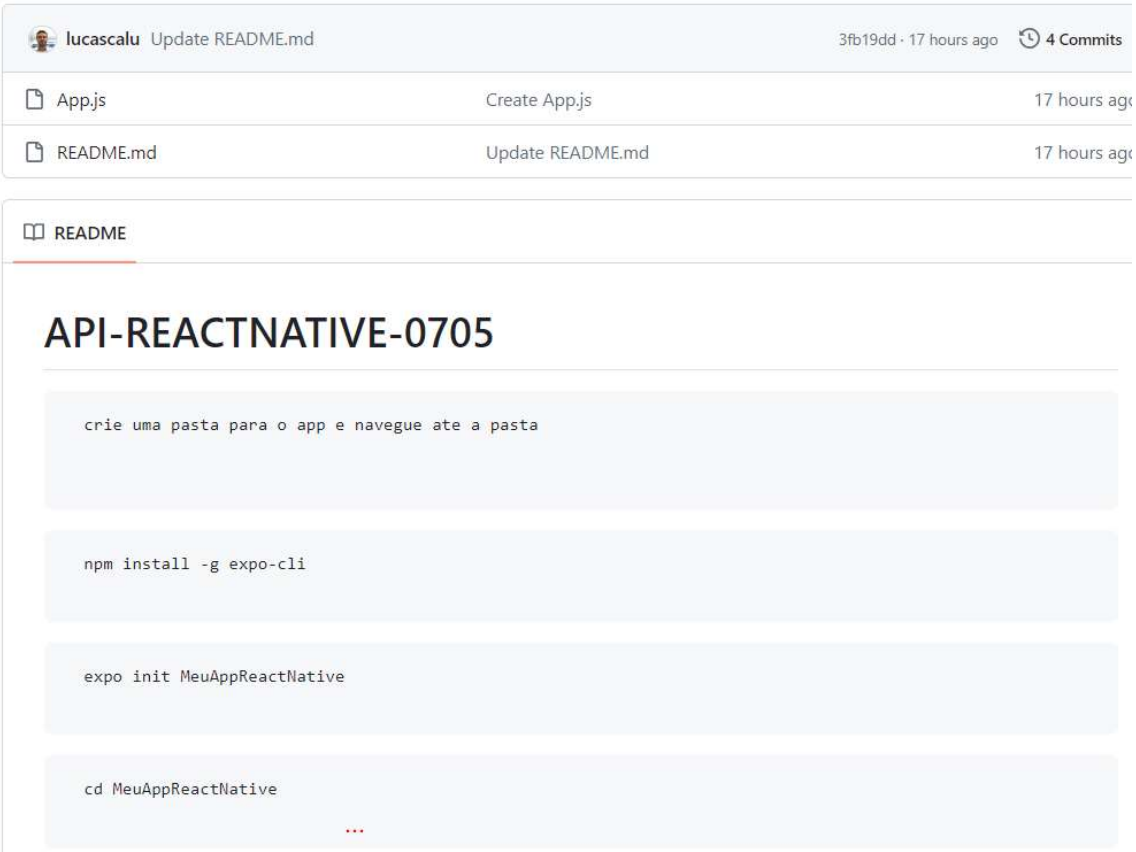
*- Preparamos a máquina virtual*

*- Preparamos e deixamos rodando o NODE (pela porta 3000) e deixamos aberto o prompt (terminal) e acessamos o "back" pelo endereço 192.168.1.**XXX**:3000/dados*

*Agora vamos fazer o "front" em React Native para interagir com essa página "de Back"*

=x=x=x=x=x=x=x=x=x=x=x=x=x=x

Relembramos que vamos seguir o passo a passo descrito no início da descrição desta etapa III (3) – Git do professor Lucas Calu:



Comecemos pelo item abaixo:

```
crie uma pasta para o app e navegue ate a pasta
```

mkdir Aula0905-10h14REACT

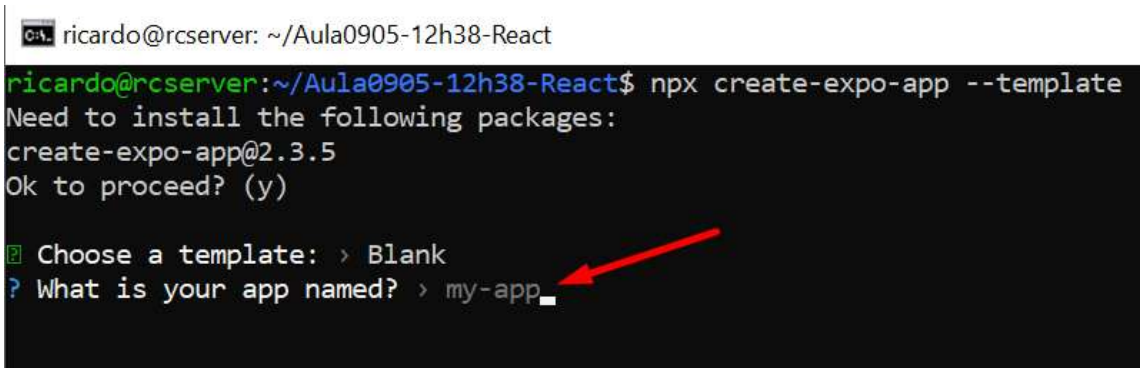
cd Aula0905-10h14REACT/

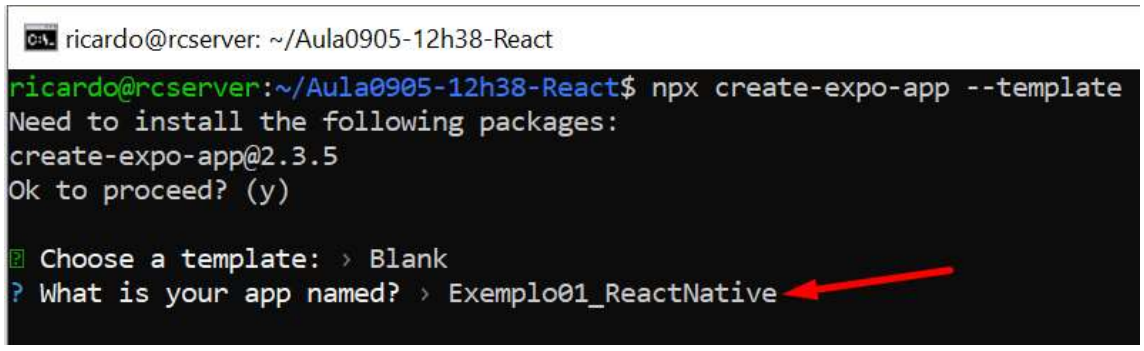
```
ricardo@rcserverteste:~$ ls
Aula0905-08h50  package.json
ricardo@rcserverteste:~$ mkdir Aula0905-10h14REACT
ricardo@rcserverteste:~$ cd Aula0905-10h14REACT/
ricardo@rcserverteste:~/Aula0905-10h14REACT$
```

## npx create-expo-app --template

```
aluno@alunosenaiserver: ~/Aula01_REACTNATIVE
aluno@alunosenaiserver:~$ ls
Aula01_NODE
aluno@alunosenaiserver:~$ mkdir Aula01_REACTNATIVE
aluno@alunosenaiserver:~$ cd Aula01_REACTNATIVE/
aluno@alunosenaiserver:~/Aula01_REACTNATIVE$ npx create-expo-app --template
Need to install the following packages:
create-expo-app@2.3.5
Ok to proceed? (y)
```

```
ricardo@rcserver: ~/Aula0905-12h38-React
ricardo@rcserver:~/Aula0905-12h38-React$ npx create-expo-app --template
Need to install the following packages:
create-expo-app@2.3.5
Ok to proceed? (y)

? Choose a template: › - Use arrow-keys. Return to submit.
    Default
    Blank - a minimal app as clean as an empty canvas
    Blank (TypeScript)
    Navigation (TypeScript)
    Blank (Bare)
```




```
cd Exemplo01_ReactNative/
```

(Entrando na pasta do novo projeto)

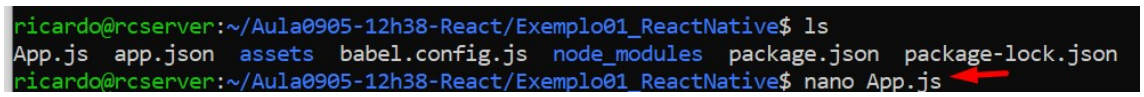
Instala as dependências necessárias para executar o aplicativo no navegador:

```
npx expo install react-native-web react-dom @expo/metro-runtime
```

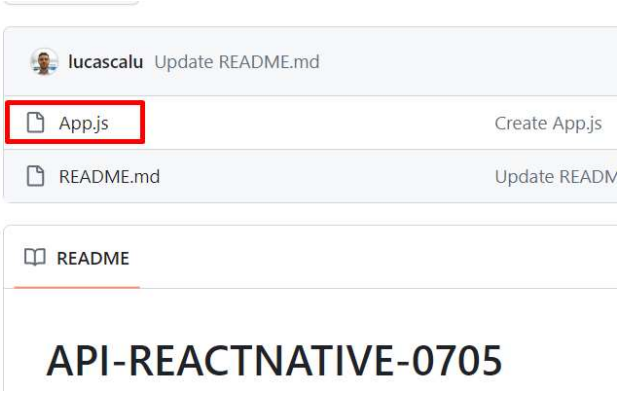
Instala a biblioteca Axios:

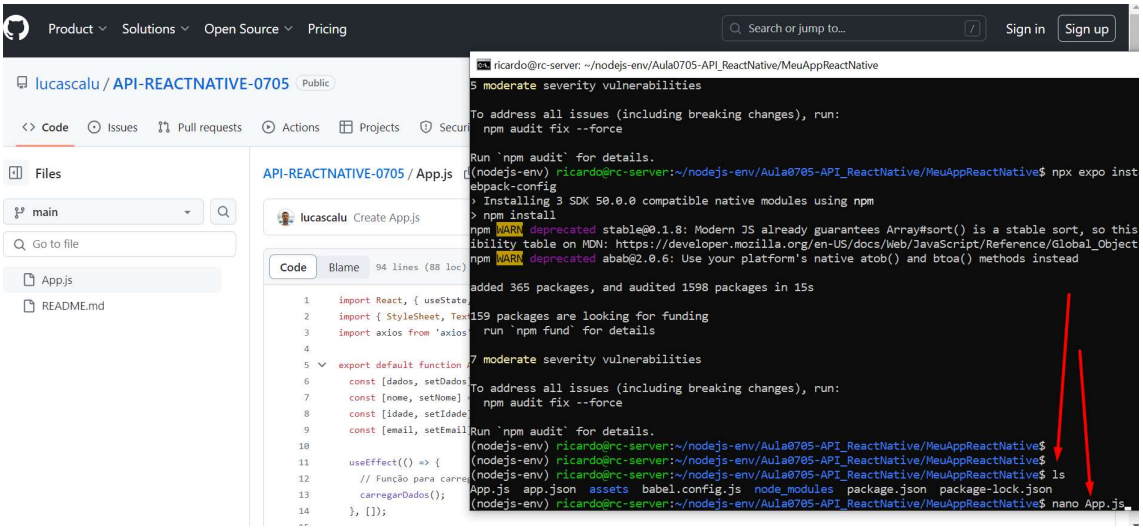
```
npm install axios
```

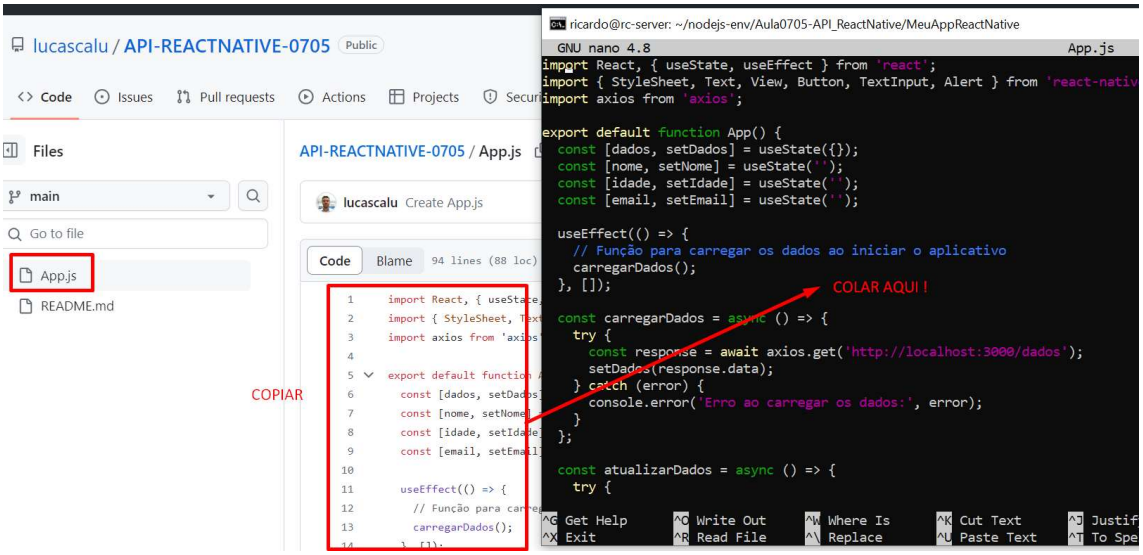
Agora vamos copiar o código do App.js




(Na parte de cima do GIT, possui um arquivo chamado “App.js”. É necessário copiar este código e substituir, alterando assim o código:




Onde acharmos o endereço LOCALHOST, substitua pelo seu ip:

```
useEffect(() => {
  // Função para carregar os dados ao iniciar o aplicativo
  carregarDados();
}, []);

const carregarDados = async () => {
  try {
    const response = await axios.get('http://192.168.1.174:3000/dados');
    setDados(response.data);
  } catch (error) {
    console.error('Erro ao carregar os dados:', error);
  }
};
```

```
const carregarDados = async () => {
  try {
    const response = await axios.get('http://192.168.1.174:3000/dados');
    setDados(response.data);
  } catch (error) {
    console.error('Erro ao carregar os dados:', error);
  }
};

const atualizarDados = async () => {
  try {
    await axios.post('http://localhost:3000/atualizar', {
      nome,
      idade,
      email
    });
    // Recarregar os dados após a atualização
    carregarDados();
    Alert.alert('Dados atualizados com sucesso');
  } catch (error) {
```

Após,

```
  const atualizarDados = async () => {
    try {

^G Get Help    ^O Write Out    ^W Where Is
^X Exit        ^R Read File    ^\ Replace
```

Salvar e sair...

# Para rodar:

sudo npx expo start –tunnel

“ou” npx expo start

```
aluno@alunosenaiserver:~/Aula01_REACTNATIVE/MeuAppExemplo01$ sudo npx expo start --t
unnel
[sudo] password for aluno:
Starting project at /home/aluno/Aula01_REACTNATIVE/MeuAppExemplo01
Starting Metro Bundler
? The package @expo/ngrok@^4.1.0 is required to use tunnels, would you like to insta
ll it globally? › (Y/n)
```

Y


Acesse o navegador pelo http://seu_ip:8081

http://**192.168.4.107**:8081/


# *Parte IV – Preparando nosso ambiente para NODE RED*

Vamos seguir o GIT do professor Lucas Calu a respeito do NODE RED:

https://github.com/lucascalu/DESAFIO_JSON_0705

Como deixamos ativado o NODEJS de forma Global (seguindo o passo a passo neste documento deste o início), vamos fazer os passos abaixo:

sudo npm install -g node-red

```
aluno@alunosenaiserver:~$ sudo npm install -g node-red
[sudo] password for aluno:

added 303 packages in 14s

46 packages are looking for funding
  run `npm fund` for details
aluno@alunosenaiserver:~$
```

node-red

```
aluno@alunosenaiserver:~$ node-red
10 May 16:46:59 - [info]

Welcome to Node-RED
===================

10 May 16:46:59 - [info] Node-RED version: v3.1.9
10 May 16:46:59 - [info] Node.js  version: v20.13.1
10 May 16:46:59 - [info] Linux 5.4.0-181-generic x64 LE
10 May 16:46:59 - [info] Loading palette nodes
10 May 16:46:59 - [info] Settings file  : /home/aluno/.node-red/settings.js
10 May 16:46:59 - [info] Context store  : 'default' [module=memory]
10 May 16:46:59 - [info] User directory : /home/aluno/.node-red
10 May 16:46:59 - [warn] Projects disabled : editorTheme.projects.enabled=false
10 May 16:46:59 - [info] Flows file     : /home/aluno/.node-red/flows.json
10 May 16:46:59 - [info] Creating new flow file
10 May 16:46:59 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

10 May 16:46:59 - [info] Server now running at http://127.0.0.1:1880/
10 May 16:46:59 - [warn] Encrypted credentials not found
10 May 16:46:59 - [info] Starting flows
10 May 16:46:59 - [info] Started flows
```

http://192.168.4.107:1880 (troque por seu ip)