



Universidade Federal de Uberlândia  
Faculdade de Engenharia Elétrica - Campus Patos de Minas  
Engenharia Eletrônica e de Telecomunicações

**Natanael Magalhães Lima de Sousa – 41911ETE001**  
**Guilherme Spinola Ramôa – 42011ETE006**

### **Manual do Usuário**



Patos de Minas

2023

## SUMÁRIO

<b>1. ORIENTAÇÕES .....</b>	<b>3</b>
<b>2. DEFINIÇÕES .....</b>	<b>3</b>
<b>I. RASPBERRY PI.....</b>	<b>3</b>
<b>II. ARDUINO NANO .....</b>	<b>4</b>
<b>III. ROS.....</b>	<b>6</b>
<b>IV. PONTE H - L298N .....</b>	<b>6</b>
<b>V. MOTORES DC .....</b>	<b>8</b>
<b>VI. GITHUB .....</b>	<b>9</b>
<b>VII. LINUX E UBUNTU.....</b>	<b>10</b>
<b>VIII. MÁQUINA VIRTUAL.....</b>	<b>10</b>
<b>3. MATERIAIS UTILIZADOS.....</b>	<b>11</b>
<b>4. MONTAGEM .....</b>	<b>11</b>
<b>5. VIRTUALBOX E MÁQUINA VIRTUAL.....</b>	<b>14</b>
<b>6. INICIANDO UBUNTU 22.04 PELA PRIMEIRA VEZ .....</b>	<b>26</b>
<b>7. VIRTUAL BOX CD .....</b>	<b>32</b>
<b>8. INSTALANDO ROS 2 HUMBLE HAWKSBILL .....</b>	<b>35</b>
<b>9. FORMATAR CARTÃO SD COM O SD CARD FORMATTER .....</b>	<b>38</b>
<b>10. INSTALAR UBUNTU 22.04 NO RASPBERRY 3B+ .....</b>	<b>43</b>
<b>11. CRIAÇÃO CLONES DE UM CARTÃO SD COM O WIN32DISK.....</b>	<b>47</b>
<b>12. CONEXÃO COM A INTERNET .....</b>	<b>55</b>
<b>13. CONEXÃO WIFI UFU.....</b>	<b>57</b>
<b>14. CONEXÃO SSH.....</b>	<b>58</b>
<b>15. “WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!” .....</b>	<b>59</b>
<b>16. INSTALANDO ROS2 HUMBLE .....</b>	<b>59</b>
<b>17. “Could not get lock /var/lib/dpkg/lock” .....</b>	<b>60</b>
<b>18. INTERFACE GRÁFICA.....</b>	<b>61</b>
<b>19. INSTALAÇÃO DA IDE ARDUINO .....</b>	<b>63</b>
<b>20. INSTALAÇÃO DO VS CODE.....</b>	<b>72</b>
<b>21. CRIAÇÃO DE PACOTES ROS 2 .....</b>	<b>77</b>
<b>22. CÓDIGOS SERIAL_MOTOR NO ARDUINO .....</b>	<b>84</b>
<b>23. CÓDIGOS SERIAL_MOTOR NO RASPBERRY .....</b>	<b>85</b>
<b>24. CIRCUITO PARA PRÓXIMAS VERSÕES .....</b>	<b>87</b>

## 1. ORIENTAÇÕES

O principal objetivo desse projeto foi a elaboração de um protótipo de robô que utiliza um Raspberry Pi 3B+ com um sistema operacional Ubuntu Server e com ROS2 instalado, e que deveria ser capaz de se comunicar com um Arduino. Essa comunicação deveria se dar de forma serial e controlar o par de motores de forma independente, uma vez que a uROS havia sido descontinuada para a versão ROS 2 Foxy Fitzroy. Além disso, optou-se por elaborar um manual bem descritivo e ilustrado, ensinando os passos mais importantes de cada etapa para um entendimento mais didático.

## 2. DEFINIÇÕES

### I. RASPBERRY PI

O Raspberry Pi é um computador de baixo custo e tamanho reduzido, desenvolvido com o objetivo de promover a educação em computação e fornecer uma plataforma acessível para projetos eletrônicos e de software. Lançado em 2012, o Raspberry Pi é amplamente utilizado por entusiastas, educadores e profissionais para uma variedade de aplicações, desde a automação residencial até robótica e servidores web.

Suas Principais características são:

- Tamanho compacto: Componentes integrados em uma única placa (placa-mãe).
- Hardware básico: Tipicamente possui um processador ARM, uma quantidade variável de RAM, portas USB, HDMI, conectividade de rede (Ethernet/Wi-Fi), além de pinos GPIO (General Purpose Input/Output) para controle de hardware externo.
- Sistema Operacional: Funciona principalmente com distribuições Linux, como o Raspberry Pi OS (anteriormente Raspbian), mas também suporta outros sistemas operacionais como Ubuntu, Windows 10 IoT e até Android.
- Preço acessível: Projetado para ser financeiramente acessível, tornando-o popular para fins educacionais e experimentação.
- Aplicações variadas: amplamente utilizado em projetos faça você mesmo, em áreas de automação, prototipagem e até como uma ferramenta de aprendizado em programação e eletrônica.

Tabela 1 - Características de alguns modelos de Raspberry Pi.

Características	Raspberry Pi 5B	Raspberry Pi 4B	Raspberry Pi 3B+	Raspberry Pi Zero W
Clock CPU [GHz]	2,4	1.5	1.4	1

RAM [GB]	4/8 SDRAM	1-8 LPDDR4	1 DDR2	0.512
<b>USB 3.0</b>	2	2	-	-
<b>USB 2.0</b>	2	2	4	-
<b>Vídeo</b>	2 micro HDMI	2 micro HDMI	HDMI	mini HDMI
<b>Áudio</b>	3.5 mm jack	3.5 mm jack	3.5 mm jack	-
<b>Wi-Fi</b>	5	GEN 1,3,4,5	GEN 1,3,4,5	GEN 4
<b>Bluetooth</b>	5.0 e BLE	5.0	4.2 BLE	4.1
<b>Alimentação</b>	5 A/5 V	1.25 A/5 V	1.13 A/5 V	180 mA/5 V
<b>Custo [R\$]</b>	915,00	557,70	409,50	259,94

Dito isso, optou-se pelo emprego da versão 3B+, uma vez que apresenta o mínimo de processamento necessário e um custo relativamente baixo ao comparado com as versões 4 e 5, além de ter certa disponibilidade no laboratório de pesquisa.

Figura 1 - Raspberry PI 3B+.



## II. ARDUINO NANO

O Arduino é uma plataforma de hardware e software open-source usada principalmente para criar projetos de eletrônica e prototipagem. Ele foi desenvolvido para facilitar o uso de microcontroladores em uma ampla gama de aplicações, como automação, robótica e sistemas de controle. As principais vantagens presentes em um Arduino são:

- Grande variedade de modelos de placas: A placa Arduino consiste em um microcontrolador (geralmente da família AVR da Atmel), que é um chip programável responsável por executar as instruções fornecidas. Cada placa tem pinos de entrada e saída (I/O) que podem ser usados para ler sensores e controlar atuadores, como LEDs, motores e displays.
- IDE própria: o Arduino IDE (Integrated Development Environment) é o software onde os usuários escrevem, testam e carregam o código (chamado de \*sketch\*) para a placa.

A linguagem de programação usada no Arduino é baseada em \*\*C/C++\*\*, com simplificações para torná-la mais acessível aos iniciantes.

- Usos variados: é utilizado em uma grande variedade de projetos de automação, arte interativa, educação e até prototipagem industrial. Pode-se citar controle de iluminação automatizada, robôs autônomos monitoramento de temperatura e umidade e sistemas de segurança.
- Comunidade: o Arduino é open-source, o que significa que tanto o hardware quanto o software estão disponíveis para qualquer pessoa modificar, criar suas próprias versões e contribuir com a comunidade global. E graças a sua simplicidade, o Arduino se tornou uma ferramenta fundamental para iniciantes e profissionais interessados em eletrônica, programação e automação.

Tabela 2 - Característica de algumas placas Arduino.

Características	Arduino Due	Arduino Nano	Arduino Uno Rev3	Arduino Mega
<b>Tensão de operação [V]</b>	3.3	5	5	5
<b>Pinos I/O digital</b>	54 (12 PWM)	22 (6 PWM)	14 (6 PWM)	54 (15 PWM)
<b>Pinos analógicos</b>	12	8	6	16
<b>SRAM [kB]</b>	96	2	2	8
<b>Flash [kB]</b>	512	32	32	256
<b>Velocidade [MHz]</b>	84	16	16	16
<b>Custo [R\$]</b>	306,99	24,99	34,99	89,25

Para o desenvolvimento do projeto optou-se pelo emprego da versão Nano, uma vez que apresenta o menor custo, um número de portas com suporte PWM razoáveis para o que é necessário aqui e alta disponibilidade no laboratório de pesquisa.

Figura 2 - Arduino Nano.

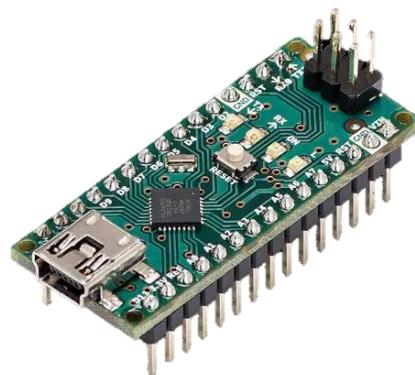
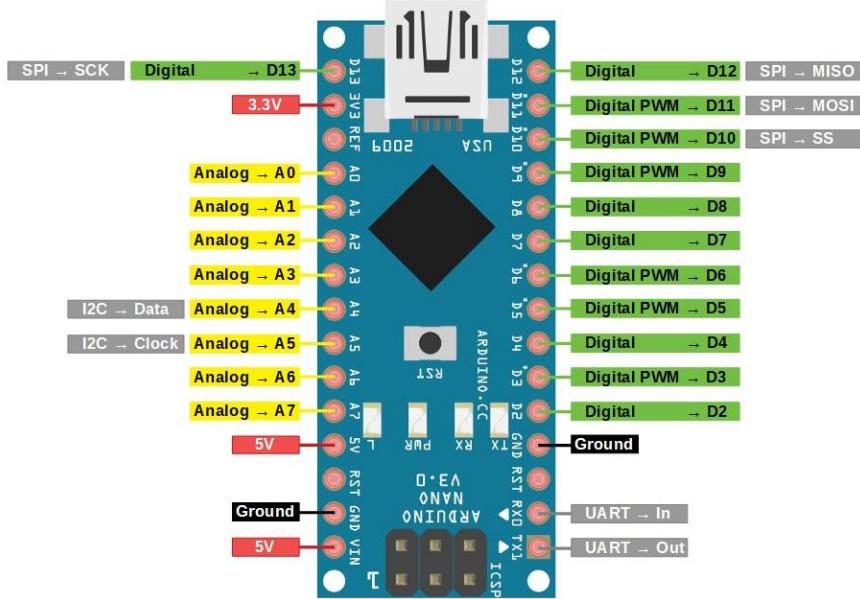


Figura 3 – Pinagem do Arduino Nano.



### III. ROS

O ROS (Robot Operating System) é um conjunto de bibliotecas open-source que fornece ferramentas e convenções para o desenvolvimento de sistemas robóticos. Apesar do nome, o ROS não é um sistema operacional completo, mas sim um middleware (camada oculta de tradução, permitindo a comunicação e o gerenciamento de dados entre o sistema operacional e os aplicativos nele executados) que facilita a comunicação entre diferentes partes de um sistema robótico, como sensores, atuadores e algoritmos de controle. Suas principais características são: arquitetura modular, comunicação por mensagens, bibliotecas e ferramentas robustas, tem suporte a simuladores e a múltiplas linguagens além de uma grande comunidade.

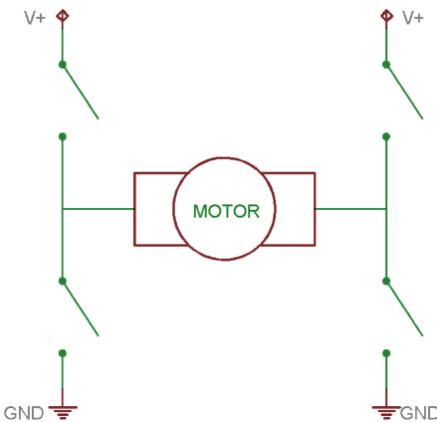
Já o ROS 2 é a evolução do ROS original, desenvolvido para superar limitações da primeira versão e atender melhor às necessidades dos sistemas robóticos modernos. Embora o ROS 1 tenha sido amplamente utilizado e popular, ele apresentava problemas de escalabilidade, segurança, tempo real e suporte a arquiteturas distribuídas. O ROS 2 foi projetado para corrigir essas limitações, incorporando melhorias significativas.

### IV. PONTE H - L298N

A ponte H é um circuito feito para controlar motores DC a partir de sinais gerados por um microcontrolador. Devido ao modo que seus componentes são alocados, é fácil selecionar o sentido da rotação de um motor, invertendo apenas a polaridade sobre seus terminais. Ela pode ser criada com o intuito de controlar o sentido da rotação de um motor utilizando chaves simples, relés ou transistores. Uma ponte H básica é composta por 4 chaves posicionadas

formando a letra “H”, sendo que cada uma localiza-se num extremo e o motor é posicionado no meio. Isso pode ser visto na figura abaixo:

Figura 4 – Esquemático de Ponte H.



Para funcionamento, basta acionar um par de chaves diagonalmente opostas, assim a corrente flui do polo positivo para o negativo atravessando o motor e fazendo-o girar. Para uma rotação inversa, acionamos o outro par de chaves, o que faz com que a corrente siga na direção oposta e, consequentemente, o sentido da rotação do motor é alterada.

Se as duas chaves superiores ou inferiores forem acionadas, você produzirá um eficiente mecanismo de freio. Isso acontece porque quando os terminais do motor são conectados no mesmo polo, um “curto-círcuito” é produzido, e a tensão gerada força-o a girar na direção oposta, que faz com que ele pare instantaneamente. Mas se todas as chaves forem fechadas, o circuito será desligado e o motor parará suavemente. Caso acionar as chaves de um mesmo lado do “H” simultaneamente, isso faz com que o fluxo de corrente vá direto do polo positivo para o negativo. Isso causa um curto-círcito fatal para a fonte de alimentação e para os componentes eletrônicos do circuito. Um driver de ponte H muito empregado por sua versatilidade e por ser muito completo é o modelo L298N.

Figura 5 – Ponte H modelo L298N.

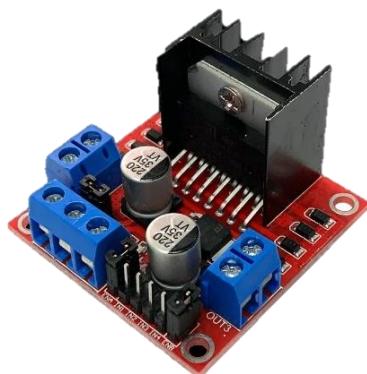
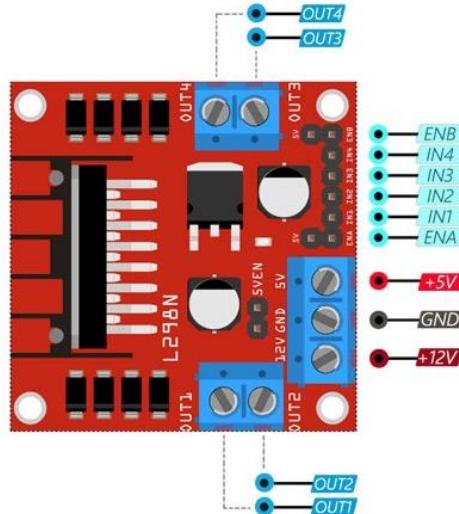
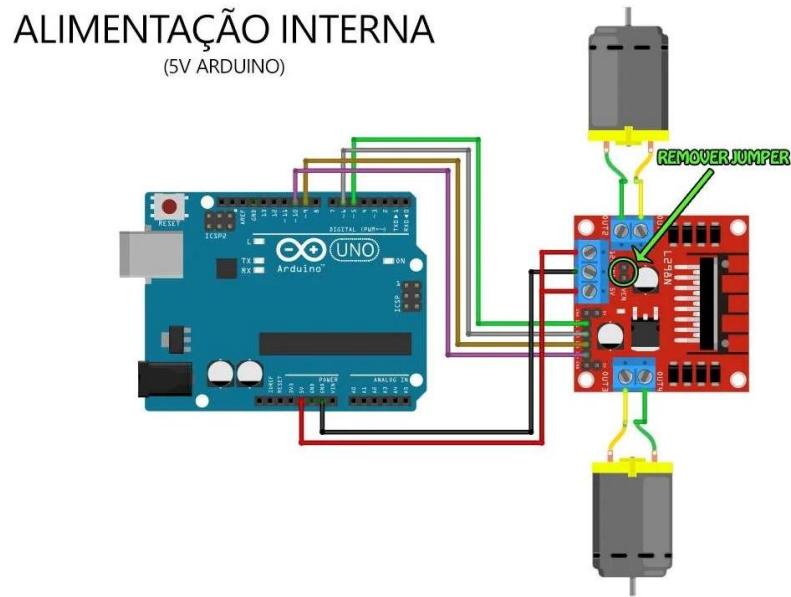


Figura 6 – Pinagem da ponte H modelo L298N.



Abaixo segue uma sugestão de alimentação do modulo usando os 5V do Arduino, cuidado com o jumper do regulador de tensão (deve ser removido).

Figura 7 – Sugestão de circuito usando o módulo de ponte H modelo L298N.



## V. MOTORES DC

São motores mais simples, eles recebem uma tensão em corrente contínua que é gerada em um campo magnético nas espiras do motor, acarretando assim uma repulsão e atração entre o campo magnético criado artificialmente pela tensão DC e o ímã real que existe dentro do motor. Existem mecanismos internos no motor DC que mudam a polaridade do ímã artificial sempre que a rotação se inicia, fazendo com que o motor nunca encontre uma posição de repouso e que os campos magnéticos sempre estejam se repelindo. Dessa forma, o movimento ocorre, convertendo a energia elétrica em mecânica.

Figura 8 – Motor DC.

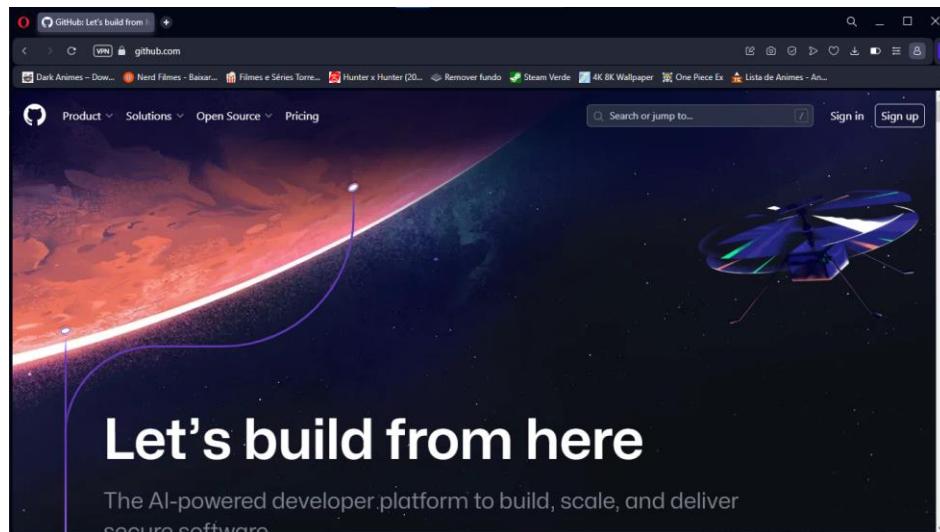


## VI. GITHUB

O GitHub é uma plataforma online que hospeda repositórios de código-fonte usando o sistema de controle de versão Git. Ele facilita a colaboração entre desenvolvedores e equipes de software, permitindo o versionamento, a revisão de código e o trabalho conjunto em projetos de software, seja de forma pública ou privada. Seu uso se deve pelas principais vantagens:

- Hosteragem de Re却itórios Git: sistema de controle de versão distribuído que permite que os desenvolvedores acompanhem as mudanças no código, revertam a versões anteriores, e colaborem de forma eficiente;
- Uso de Branches: podem criar ramificações para trabalhar em novas funcionalidades ou correções sem afetar o código principal;
- Issues e Gestão de Projetos: ele permite aos desenvolvedores rastrear bugs, solicitar novas funcionalidades, ou discutir melhorias;
- Open-source: Projetos públicos podem ser visualizados e clonados por qualquer pessoa, e os desenvolvedores podem contribuir através de pull requests.

Figura 9 – Página inicial do GitHub.



## VII. LINUX E UBUNTU

Linux é um sistema operacional de código aberto, baseado no núcleo (kernel) desenvolvido por Linus Torvalds em 1991. O termo "Linux" refere-se, tecnicamente, ao núcleo (kernel) do sistema, mas, na prática, também é usado para designar todo o conjunto de sistemas operacionais que usam esse kernel, como Ubuntu, Fedora, Debian, entre outros. Esses sistemas são conhecidos como distribuições Linux. Seu uso se deve principalmente por ser open-source podendo ser usado, modificado e distribuído por qualquer pessoa. Isso permitiu a criação de inúmeras distribuições personalizadas e a colaboração de uma vasta comunidade global de desenvolvedores.

Já o Ubuntu é uma das distribuições mais populares do sistema operacional Linux, desenvolvida pela empresa Canonical Ltd. e lançada em 2004. É conhecido por sua facilidade de uso, estabilidade e suporte comunitário, o que o torna uma das principais escolhas para quem está começando no mundo do Linux, tanto para uso pessoal quanto empresarial. Sua escolha se deve principalmente pelo seu gerenciamento de pacotes, facilidade de instalação, personalização, flexibilidade e a existência de uma comunidade ativa e global com muitos fóruns, tutoriais e documentação disponíveis.

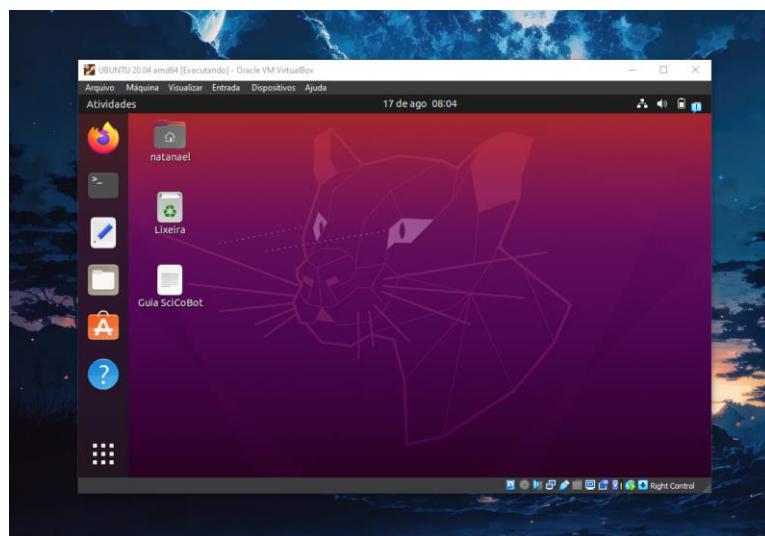
## VIII. MÁQUINA VIRTUAL

Uma máquina virtual é um programa que simula um ambiente computacional, e pode executar sistemas operacionais e aplicativos como se fosse uma máquina física. A máquina virtual opera de forma completamente independente e isolada, podendo inclusive rodar sistemas operacionais que normalmente não seriam compatíveis com sua arquitetura. Um bom exemplo de camada de virtualização é o Parallels Desktop, software pago que permite criar máquinas virtuais do Windows ou de distribuições Linux dentro do macOS. Outro exemplo é o VirtualBox, gratuito, de código aberto e compatível com Windows, macOS e Linux, que permite rodar quase qualquer sistema operacional.

A máquina virtual é uma excelente forma de executar um sistema operacional novo antes de instalar definitivamente no computador, antecipando erros ou problemas graves. Cada máquina virtual pode rodar um sistema específico, com mais ou menos memória, mais ou menos espaço dedicado e aplicações diferentes, para funções diversas ou algumas iguais a outras, como redundâncias, para garantir a segurança dos dados. Dependendo da configuração, servidores com virtualização diminuem riscos como perda de dados ou indisponibilidade do sistema, otimizando uso do hardware.

Porém, dependendo do uso, uma máquina virtual pode apresentar limitações. Por exemplo caso o computador tenha um processador antigo ou pouca RAM, a máquina virtual ficará lenta e dependendo do uso, pode não atender às expectativas. Outra é que para o uso do Windows ou distribuições pagas, de modo legal, é preciso adquirir a licença para cada máquina virtual.

Figura 10 – Máquina virtual Oracle com o sistema Ubuntu 20.04 sendo utilizado em windows.



### 3. MATERIAIS UTILIZADOS

- Raspberry 3B+;
- Arduino Nano;
- Chassi (com rodas e motores);
- Jumpers;
- Driver de motor Ponte H dupla (L298N);
- Driver de motor Ponte H dupla (L298N);
- Velcro;
- Cartão SD classe 10;
- Adaptador cartão SD para USB.

### 4. MONTAGEM

Inicialmente, dividiu-se o projeto em duas partes independentes: uma de criação e montagem do Arduino com uma ponte H, que deveria controlar dois motores à partir de uma entrada serial usando uma função “Serial.read()”, e a segunda fazendo um raspberry ler um comando escrito no teclado e enviá-lo serialmente para o Arduino ligado em alguma de suas portas usando ROS 2.

Para a montagem do circuito do projeto focou-se na ligação do Arduino com a ponte H, modelo L298N. Optou-se pelo uso do Arduino Nano por este ser mais compacto, com baixo custo e por suprir as portas necessárias para o funcionamento do circuito, mas qualquer outra placa com 4 portas PWM pode ser empregada.

Tabela 3 – Pinagem dos componentes;

MICROCONTROLADOR ARDUINO	DRIVER PONTE H L298N	MOTOR
5V	5V	NENHUM
5V	12V	NENHUM
GND	GND	NENHUM
D3	IN2	1
D4	IN1	1
D5	ENB	2
D6	ENA	1
D7	1N4	2
D8	IN3	2

OBS: REMOVER JUMPER DO VLOGIC!

Figura 11 – Esquemático com a montagem realizada.

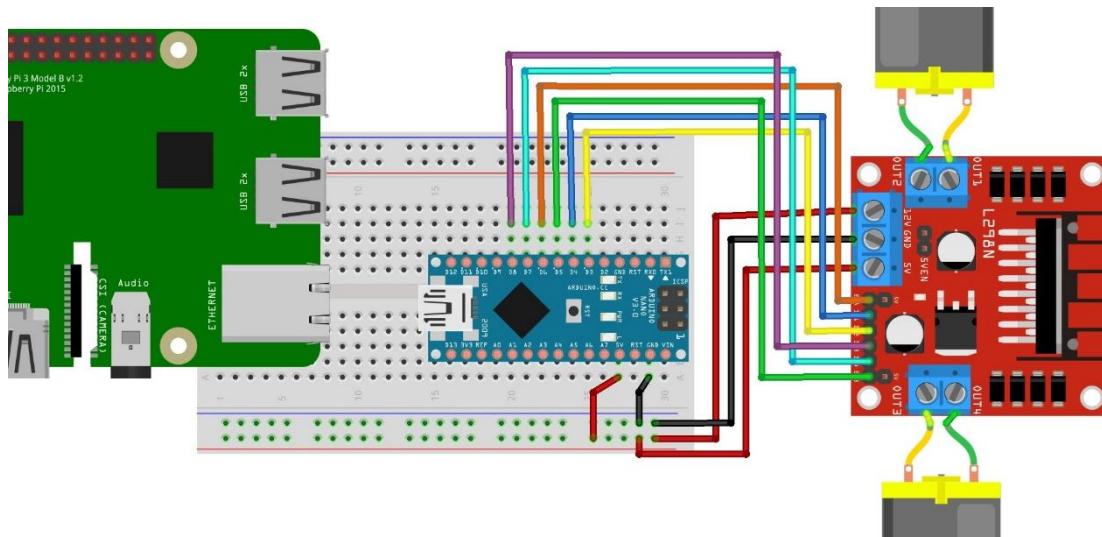


Figura 12 – Pinagem do driver de ponte H modelo L298N.

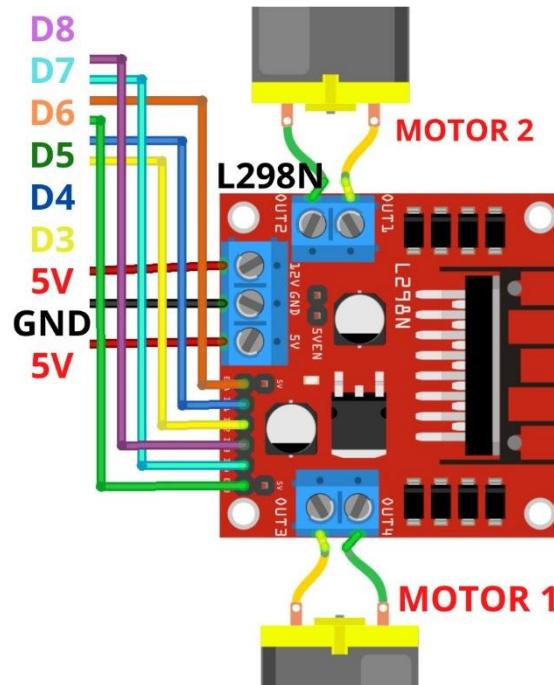


Figura 13 – Pinagens das ligações feitas no Arduino Nano comentadas.

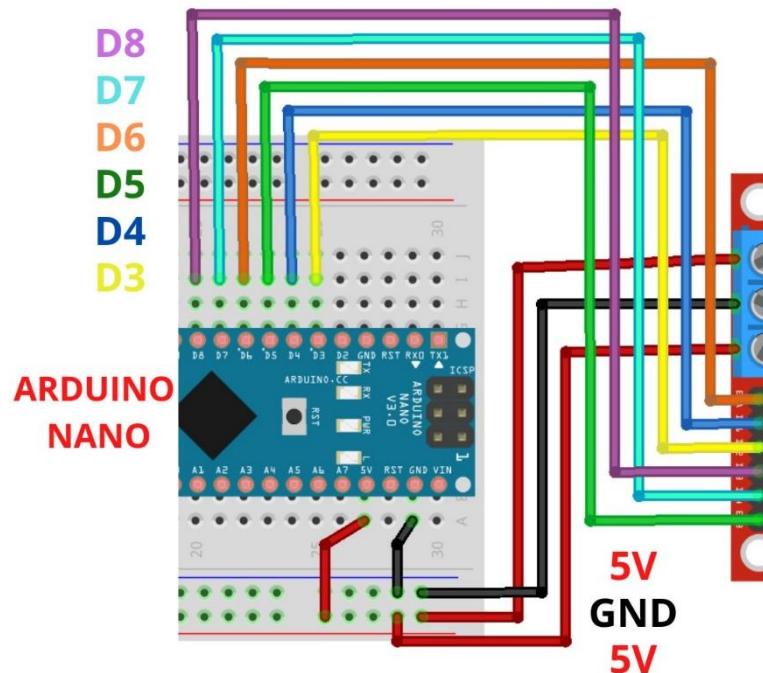
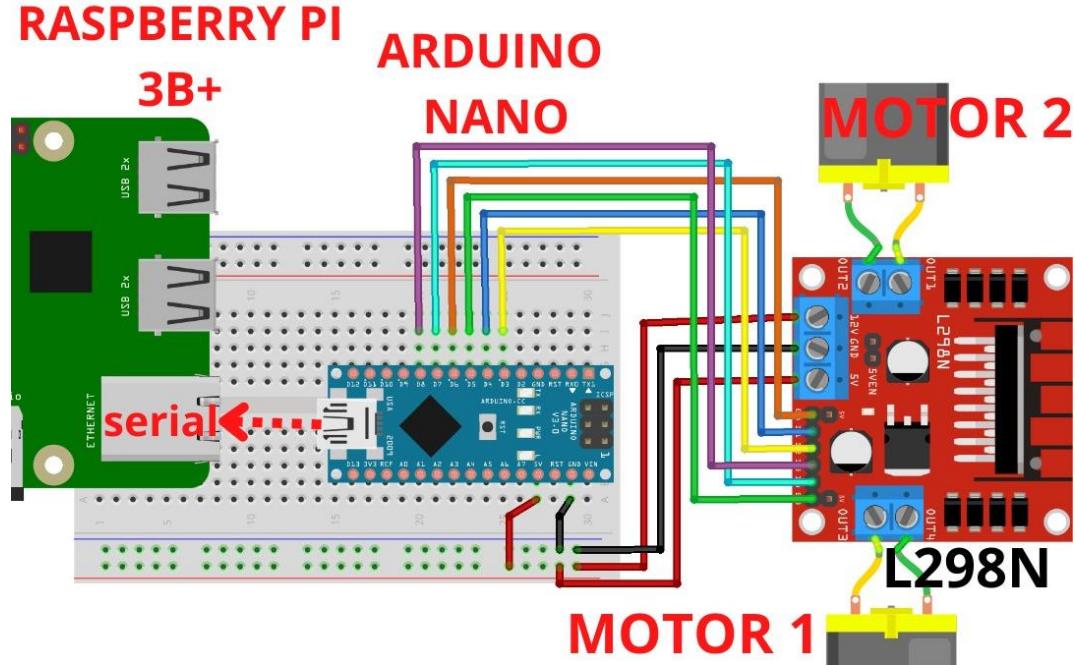


Figura 14 – Esquemático com a montagem realizada comentada.



## 5. VIRTUALBOX E MÁQUINA VIRTUAL

Para esse guia optou-se pelo software VirtualBox por ser gratuito e fácil de se utilizar, explicar e entender todos os passos necessários para a criação de uma máquina virtual voltada ao uso do Scicobot. Antes de tudo acesse o site oficial do VirtualBox [<https://www.virtualbox.org/>]:

Figura 15 – Site VirtualBox.



Pressione o botão de Download e na página seguinte escolha a opção “Windows hosts”, dessa forma o aplicativo executável será baixado de seu navegador.

Figura 16 – Opções de download VirtualBox.

## Download VirtualBox

Here you will find links to VirtualBox binaries and

### VirtualBox binaries

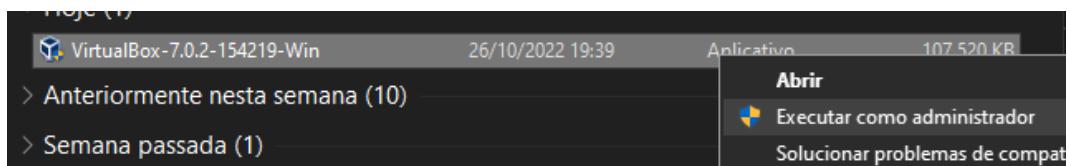
By downloading, you agree to the terms and co

If you're looking for the latest VirtualBox 6.1 pac

### VirtualBox 7.0.2 platform packages

- ↗ Windows hosts
- ↗ macOS / Intel hosts

Figura 17 – Download executável VirtualBox.



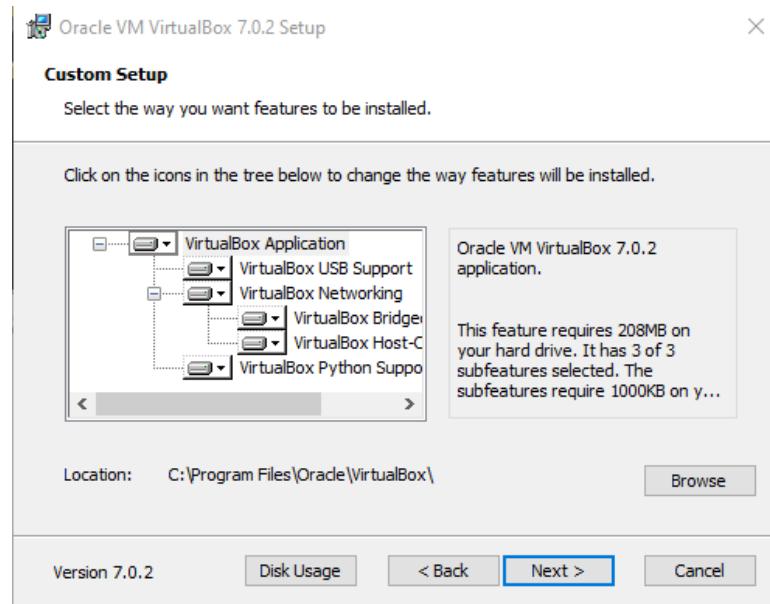
Aperte com o botão direito do mouse e execute como administrador, desse modo o download se iniciará e uma nova janela se abrirá.

Figura 18 – Passo do instalador VirtualBox.



Pressione “**Next >**” e uma janela de customização aparecerá. Aqui você pode editar as ferramentas a serem instaladas no programa, mas é recomendável que para nossa finalidade elas não se alterem e se mantenham como padrão. Além disso, aqui você pode escolher onde o programa será instalado dentre suas pastas de arquivos.

Figura 19 – Passo do instalador VirtualBox.



Em seguida aparecerão algumas abas de confirmação antes de instalar o VirtualBox propriamente dito. Confirme as janelas que aparecerem e em seguida aperte “Install”.

Figura 20 – Passo do instalador VirtualBox.

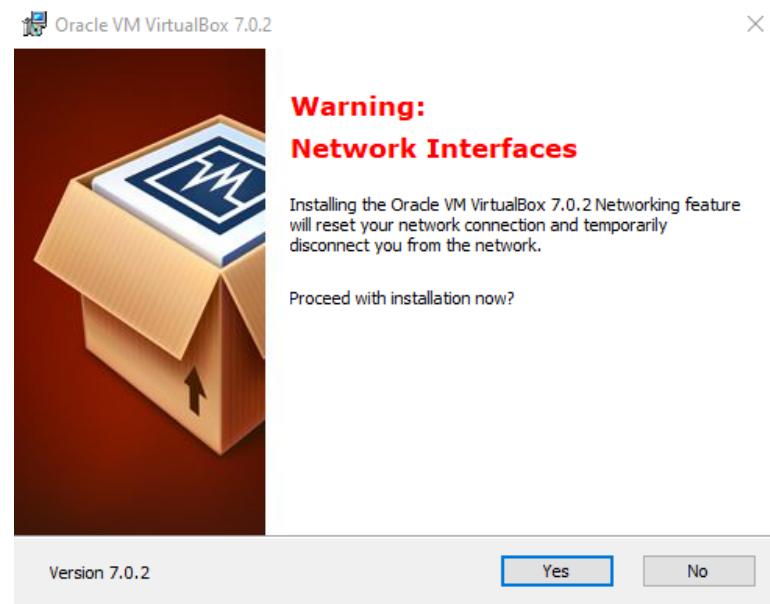


Figura 21 – Passo do instalador VirtualBox.

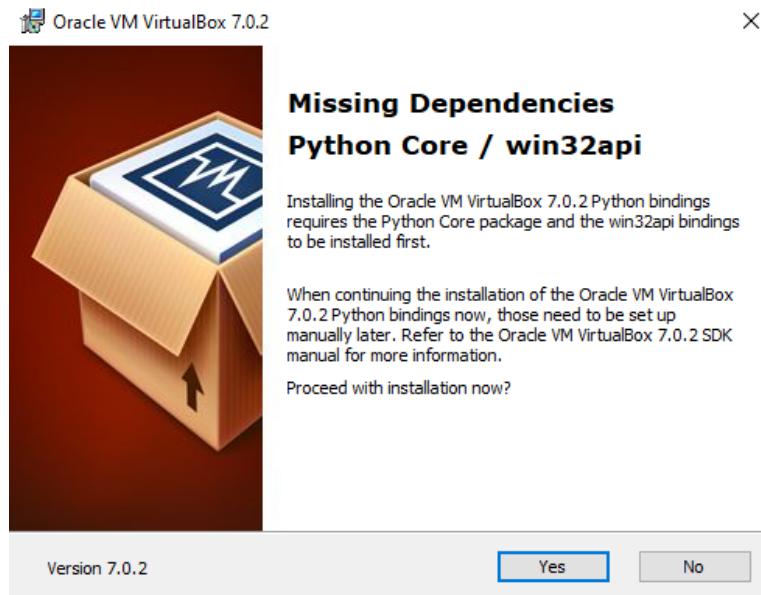


Figura 22 – Passo do instalador VirtualBox.

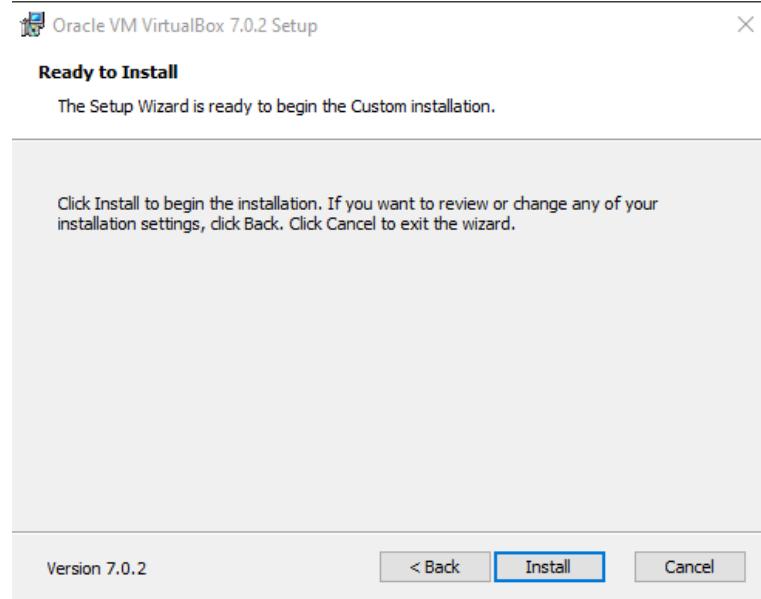
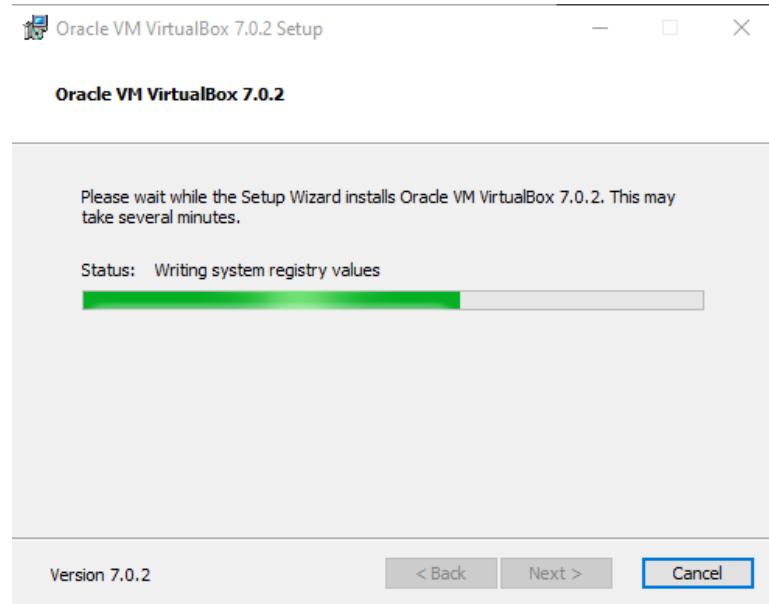


Figura 23 – Passo do instalador VirtualBox.



Depois que toda barra se carregar e o download terminar o software estará pronto para ser executado. Aperte “Finish” e siga para a criação de sua máquina Linux Ubuntu 22.04 Humble Hawksbill amd64bits.

Figura 24 – Passo do instalador VirtualBox.



Figura 25 – Ícone Oracle VM VirtualBox.



Abrindo o programa Oracle VM VirtualBox você se deparará com uma janela com várias opções. Para começar a criação de sua máquina vá em “Novo”, ou em “Máquina” e em seguida em “Novo” ou até mesmo [CTRL]+[N].

Figura 26 – Interface VirtualBox.

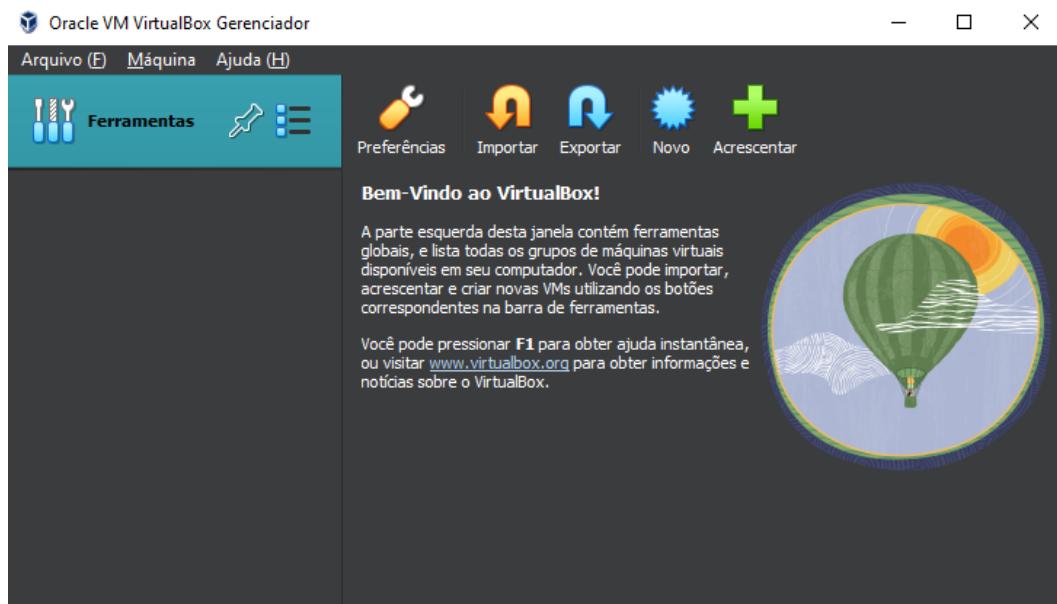


Figura 27 – Criando uma máquina na Interface VirtualBox parte 1.



Assim se abrirá uma janela para preenchimento de dados. Nesse guia ensinaremos a criar uma máquina virtual com o sistema operacional Linux Ubuntu 22.04 Jammy Jellyfish amd64bits, mas caso queira criar alguma com outro sistema para testes sinta-se livre, o procedimento é similar. Aqui escreva o nome de sua máquina, o tipo e a versão (conforme escreve o programa ajusta o tipo e a versão automaticamente, porém confirme manualmente

antes de prosseguir). Além disso, caso seja preciso aperte o botão “Ajuda” ou [H] para uma explicação mais detalhada sobre as etapas de criação do Virtual Box.

Figura 28 – Criando uma máquina na interface VirtualBox parte 2.

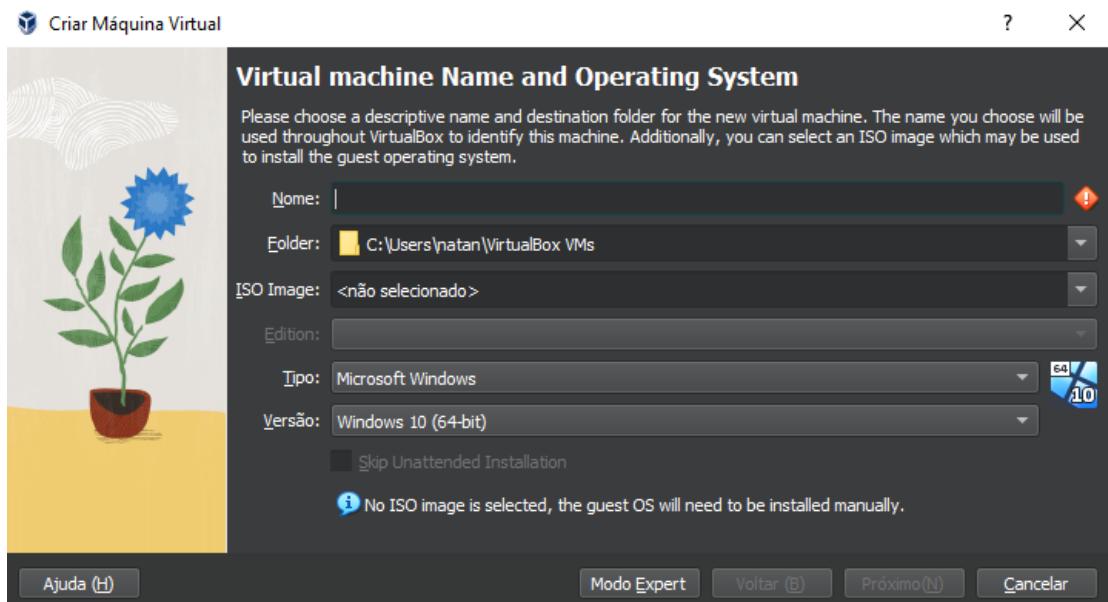


Figura 29 – Criando uma máquina na interface VirtualBox parte 3.

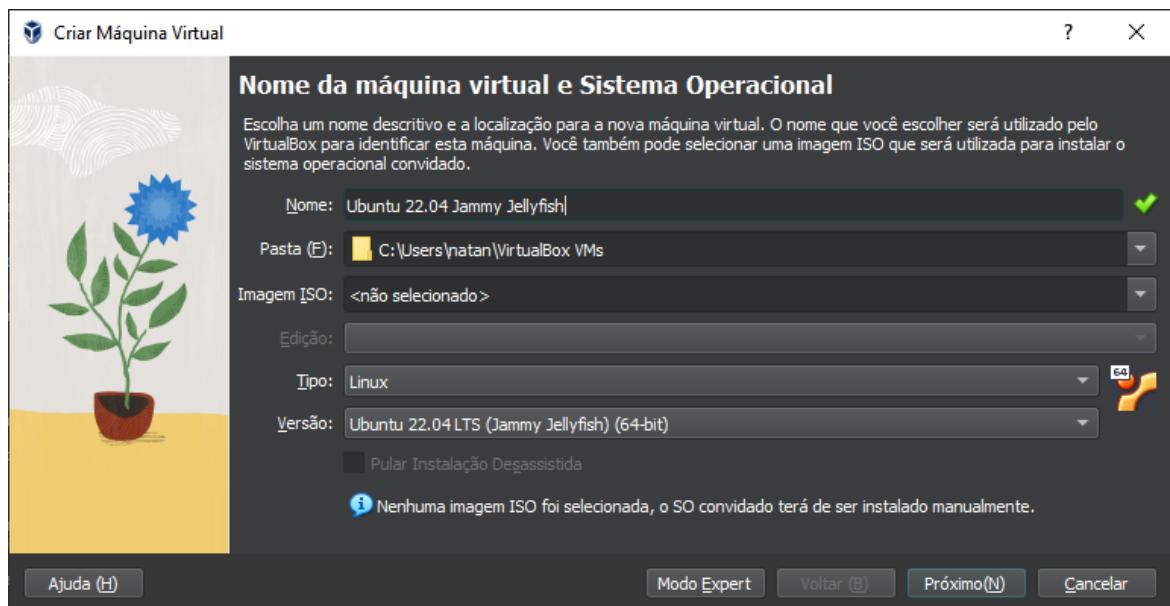


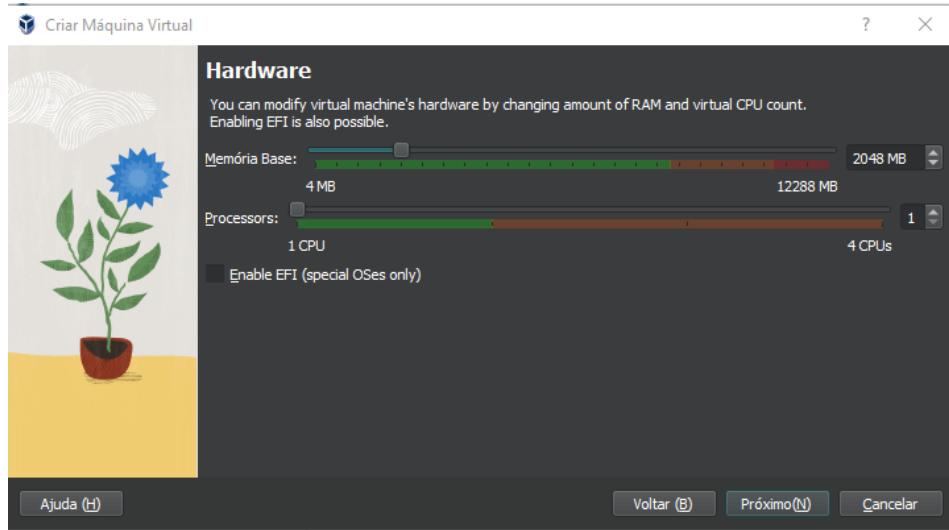
Figura 30 – Criando uma máquina na Interface VirtualBox parte 4.



Em seguida você definirá a quantidade de hardware de sua máquina que poderá ser aproveitada pela máquina virtual, juntamente com a quantidade de processadores. O próprio

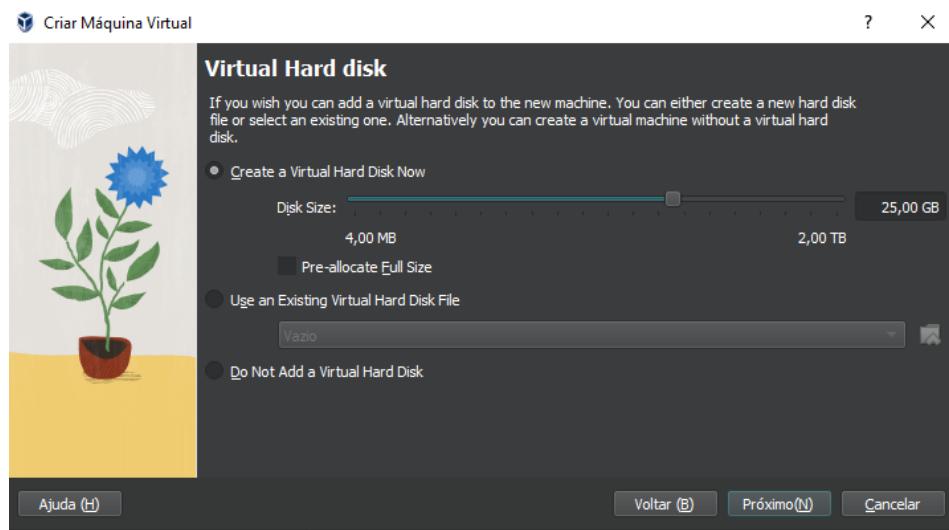
software te ajuda nessa parte: ele aponta em verde os requisitos recomendados! A área destacada em verde não acarretará malefícios para sua máquina, enquanto disponibilizar muitos recursos (mais que o recomendado) deixará seu computador mais lento.

Figura 31 – Criando uma máquina na Interface VirtualBox parte 5.



Nessa etapa você criará um disco virtual para sua máquina. Defina como fez anteriormente seu tamanho de acordo com a capacidade de sua máquina. Para a aplicação que faremos nesse guia não é necessária uma capacidade extraordinária de recursos, mas pense sempre em uma margem de segurança para que não ocorram falhas futuras e seja necessário criar uma máquina do zero (também é possível ajustar esses valores nas configurações mais tarde, mas desde que a máquina se encontre desligada). Caso já tenha um disco virtual do mesmo tipo previamente criado, ele pode ser reutilizado clicando em “[Use na Existing Virtual Hard Disk File](#)”.

Figura 32 – Criando uma máquina na Interface VirtualBox parte 6.



Por fim um sumário irá se abrir com todas as especificações determinadas por você. Dê uma lida e observe se tudo está como você escolheu ou se algo deve ser corrigido com um “Voltar” ou [B].

Figura 33 – Criando uma máquina na Interface VirtualBox parte 7.

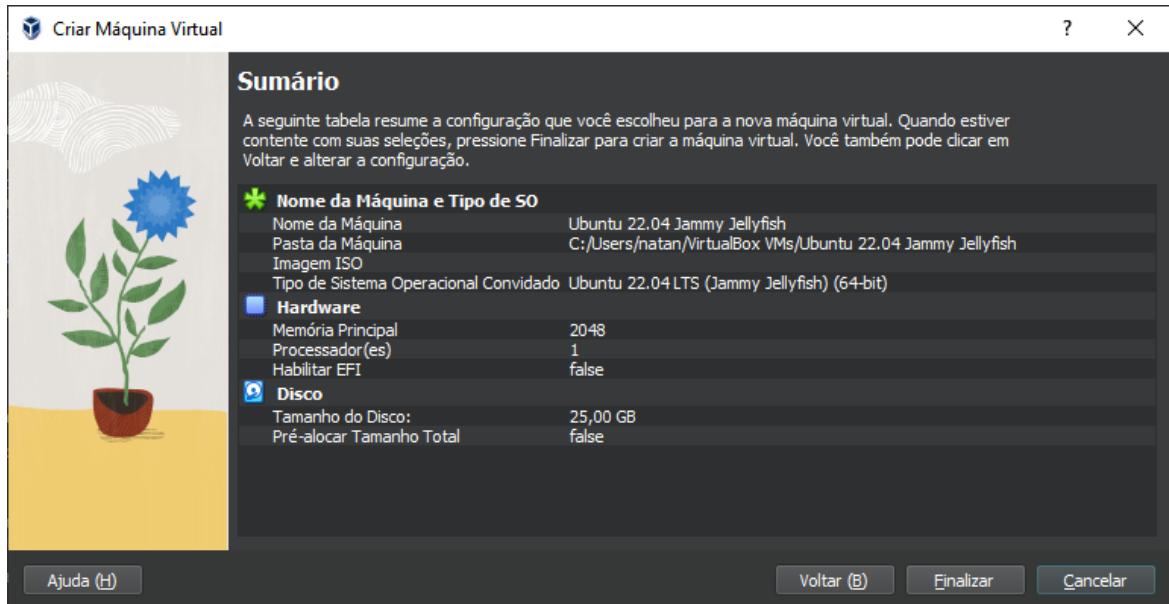
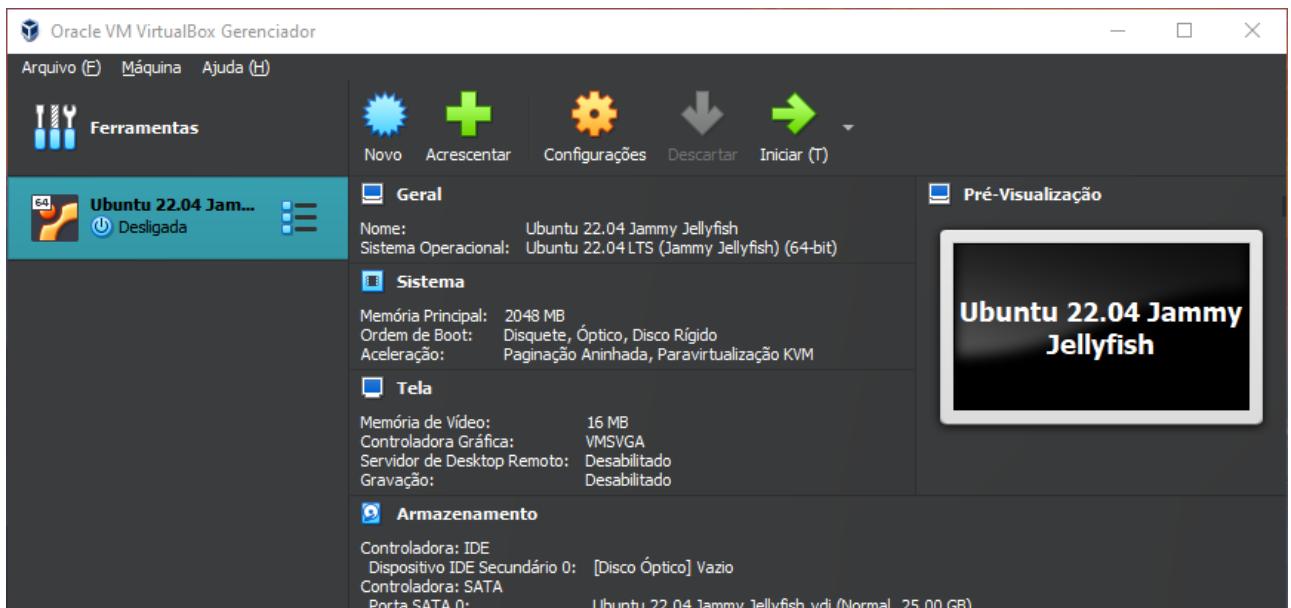
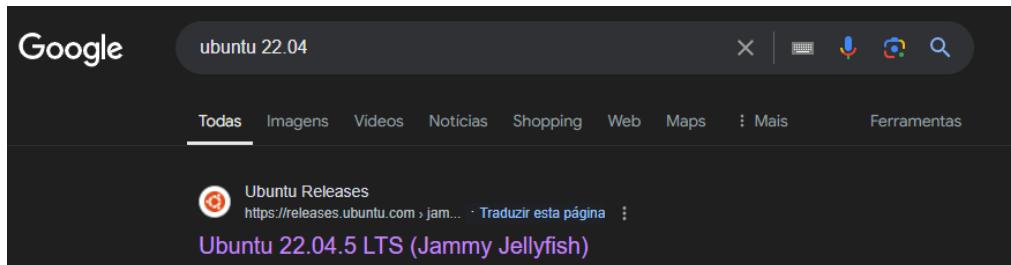


Figura 34 –Máquina criada na Interface VirtualBox.



Com a máquina criada você precisará baixar o sistema operacional. Pesquise no Google por “ubuntu 22.04” ou abra o link [<https://releases.ubuntu.com/jammy/>]:

Figura 35 – Busca pelo sistema operacional.



Role a página até encontrar a seção “[Select an image](#)”. Vá até a opção “[Desktop image](#)” e faça o download do arquivo de imagem do sistema operacional.

Figura 36 – Download da imagem do sistema operacional.

### Select an image

Ubuntu is distributed on three types of images described below.

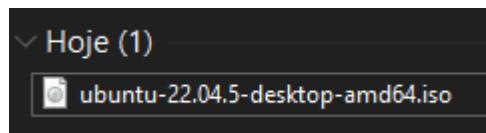
#### Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MiB of RAM to install from this image.

#### 64-bit PC (AMD64) desktop image

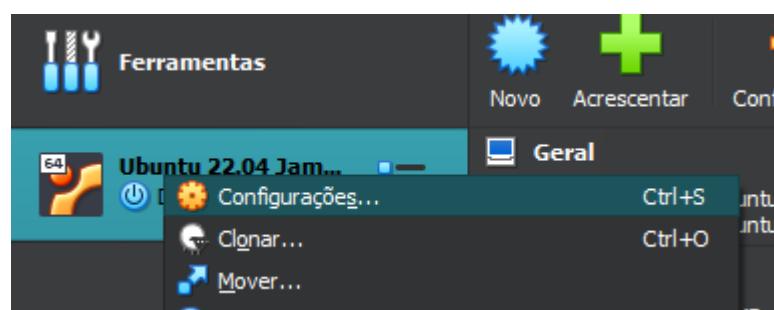
Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

Figura 37 – Imagem do sistema operacional.



Com a imagem em mãos, volte à sua máquina virtual criada. Clique com o botão direito e vá em configurações. Aqui é possível alterar alguns parâmetros pré-estabelecidos durante a criação da máquina.

Figura 38 – Configurando a máquina virtual na VirtualBox parte 1.



Vá na aba “[Armazenamento](#)”, na sessão “[Dispositivos de Armazenamento](#)” e na aba “[Controladora: IDE](#)”. Observe que por padrão o disco está vazio. Clique com o botão esquerdo sobre ele e vá até o a imagem de um disco no canto direito da tela, clique sobre ele com o botão esquerdo e escolha a opção “[Escolher uma imagem de disco...](#)”. Aí é só ir até

onde seu navegador aloca seus downloads e escolher a imagem do “ubuntu-22.04.5-desktop-amd64”.

Figura 39 – Configurando a máquina virtual na VirtualBox parte 2.

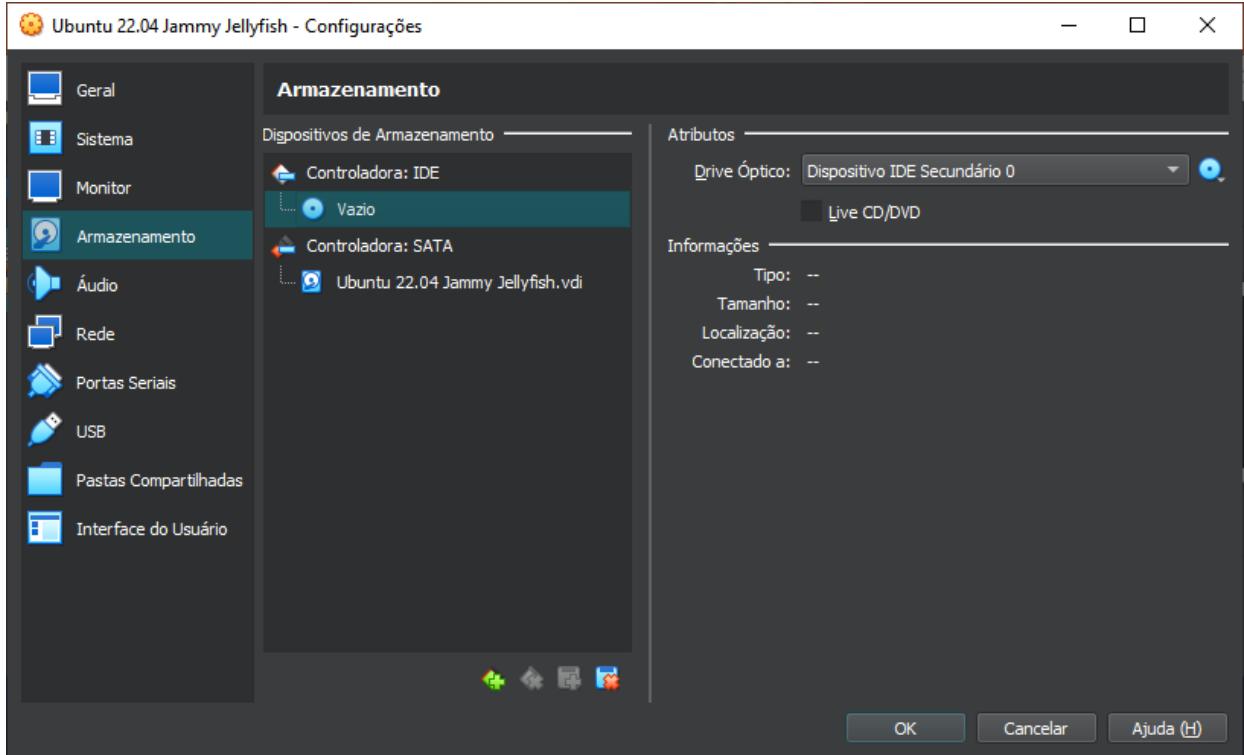


Figura 40 – Configurando a máquina virtual na VirtualBox parte 3.

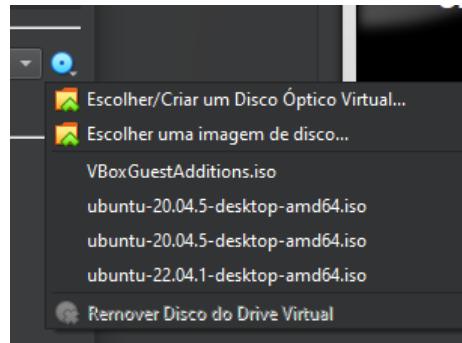
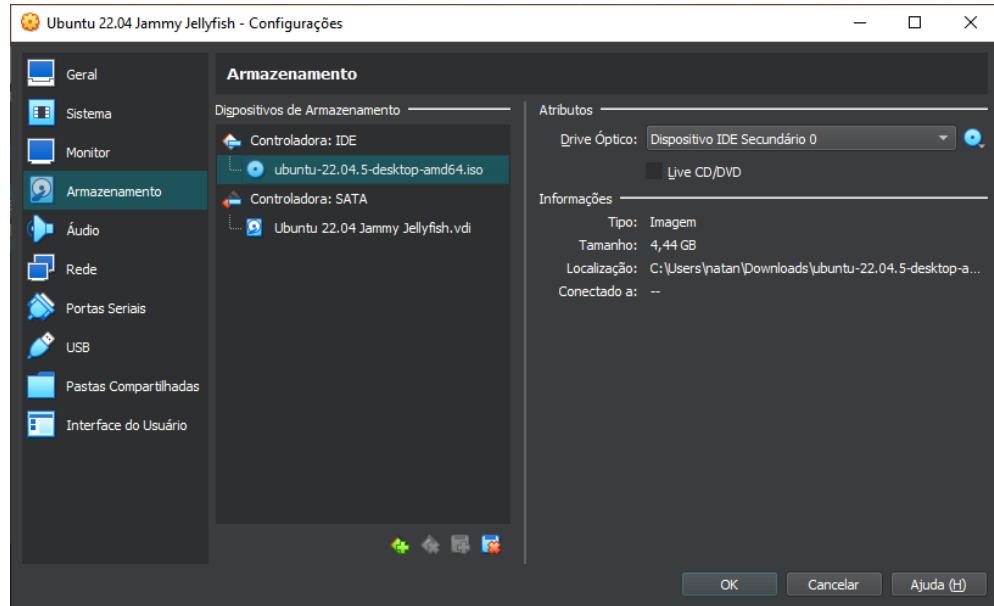


Figura 41 – Configurando a máquina virtual na VirtualBox parte 4.



Próximo passo é configurar as redes da máquina, passo crucial pois sem conexão com a rede não é possível instalar o ROS2 e nem atualizar os pacotes necessários para esse objetivo. Além disso conexão com a rede é necessária para a conexão SSH com a raspberry. Para isso recomenda-se criar duas redes, uma para conexão wifi e outra para uso de cabos de rede.

- Em “**Adaptador 1**” criaremos a rede cabeada: habilite a placa de rede e a conecte à “**Placa em modo Bridge**” e defina seu nome para “**Realtek PCIe GbE Family Controller**”.
- Em “**Adaptador 2**” criaremos a rede wireless: habilite a placa de rede e a conecte à “**Placa em modo Bridge**” e defina seu nome para “**Intel(R) Dual Band Wireless-AC 3168**”.

Figura 42 – Configurando a rede da máquina virtual na VirtualBox parte 1.

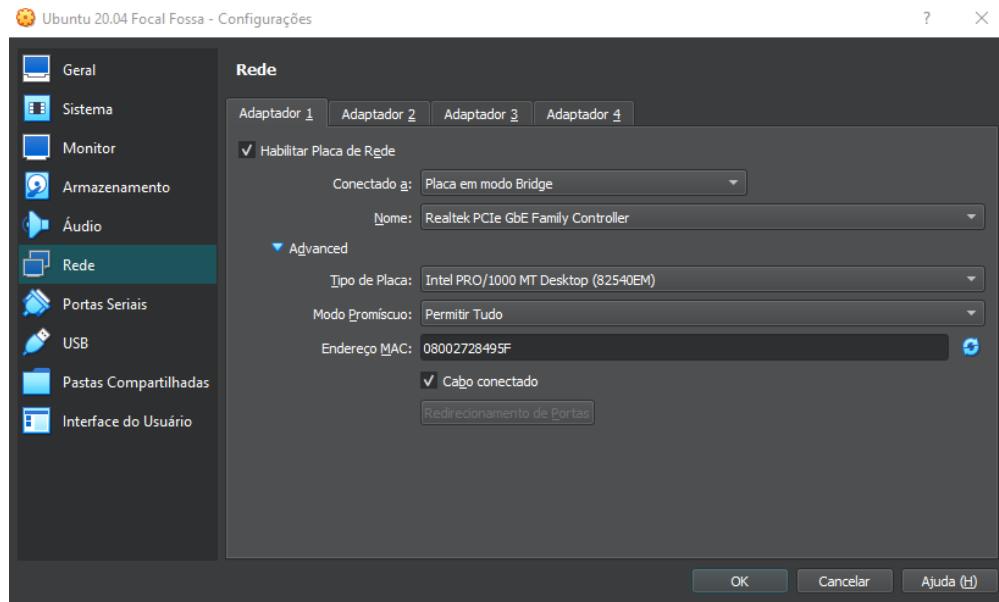
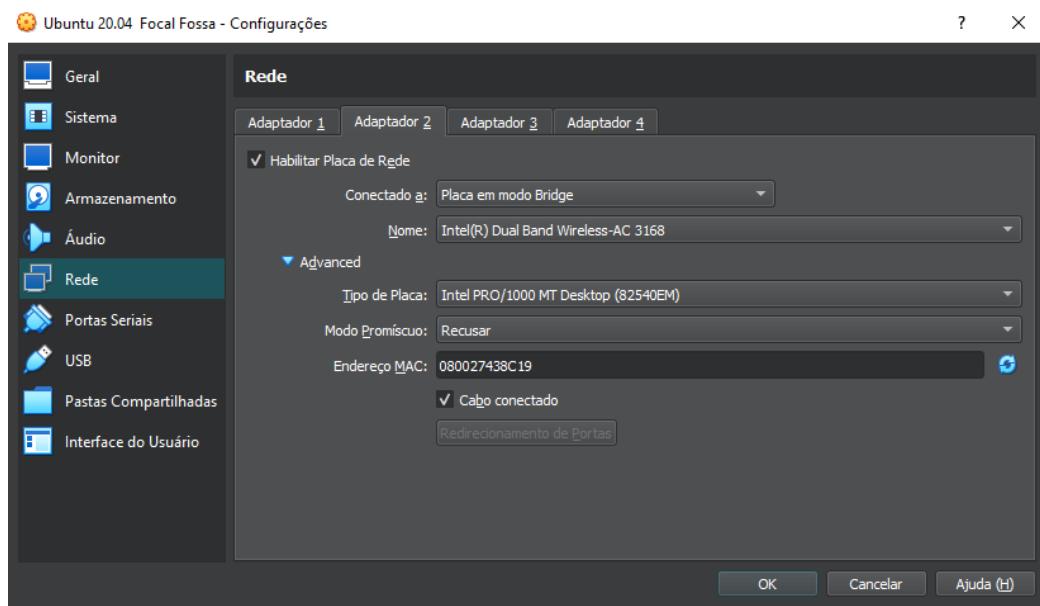


Figura 43 – Configurando a rede da máquina virtual na VirtualBox parte 2.

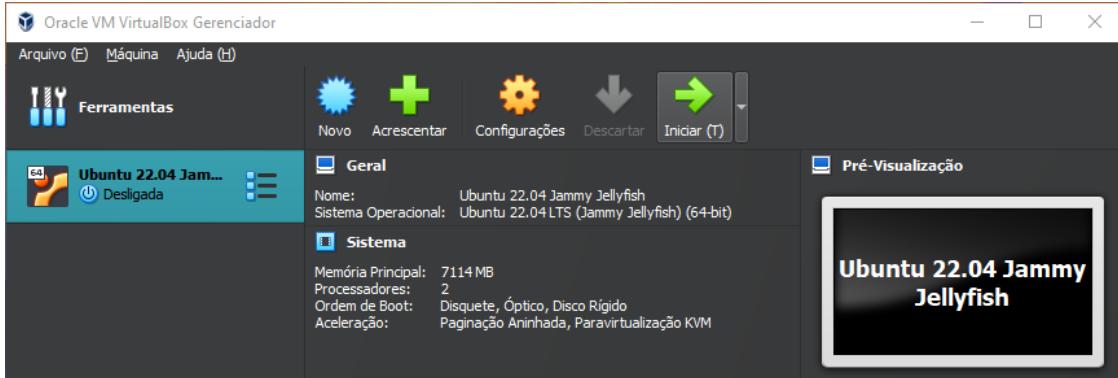


Para nossa finalidade está tudo pronto, caso queira analisar outras opções como inserir dispositivos seriais ou criar pastas compartilhadas entre máquinas sinta-se livre a se aventurar. Com essas configurações o que resta é apenas executar a máquina e fazer suas configurações iniciais.

## 6. INICIANDO UBUNTU 22.04 PELA PRIMEIRA VEZ

Se todos os passos tivessem sido executados como foram apresentados aqui basta selecionar a máquina no programa Oracle VM VirtualBox e clicar com o botão direito no botão “**Iniciar**” ou [T] ou ainda com o botão esquerdo na máquina e depois na opção “**Iniciar**”.

Figura 44 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 1.



A tela se iniciará com o sistema checando o disco e logo em seguida abrirá uma tela para determinar o idioma do tutorial além de opções de instalar o Ubuntu ou apenas de testar. Para a finalidade deste guia recomenda-se que o sistema seja instalado, uma vez que a versão de testes não salva as mudanças caso a máquina seja desligada. Um ponto a se destacar é que a tela inicialmente aparecerá com tamanho fixo em uma moldura quadrada, então as vezes se tornará necessário ajustar a janela da tela para encontrar algumas opções ou informações.

Figura 45 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 2.

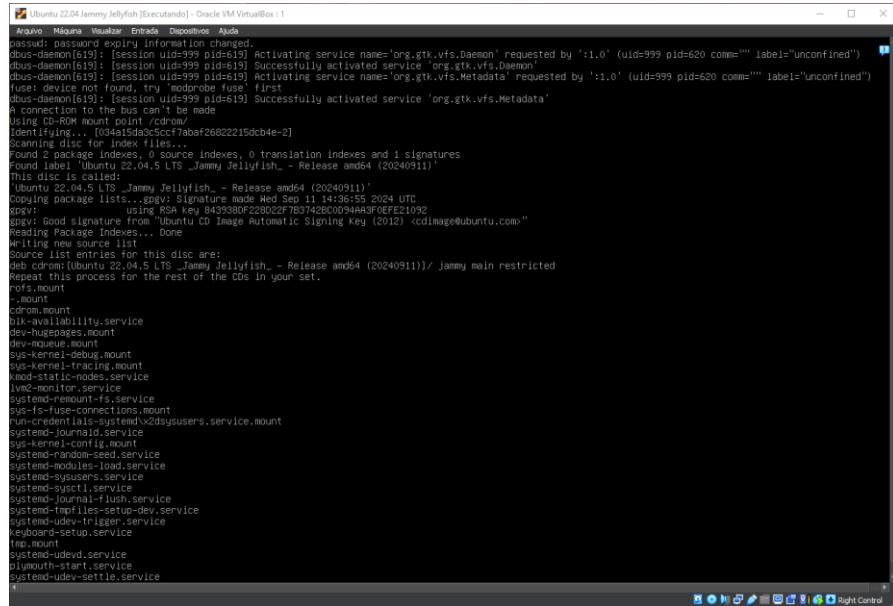
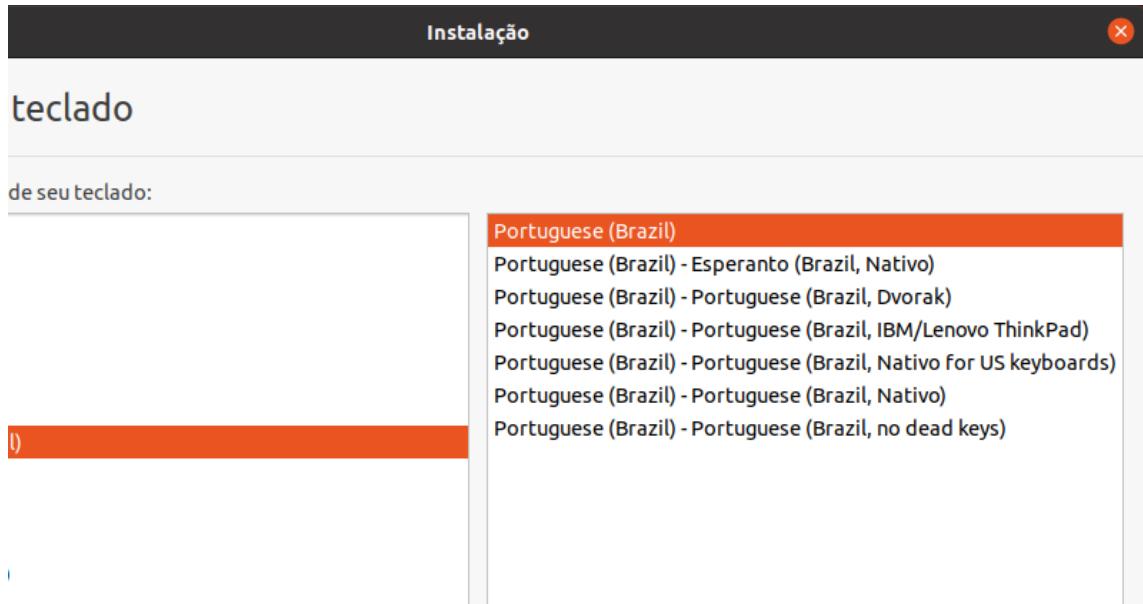


Figura 46 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 2.



Em seguida deve-se escolher o idioma do teclado (escreva uma palavra de teste como atenção – que leve “ç” ou “~”). Depois ajuste a janela na tela para encontrar o botão de continuar e prossiga com a instalação.

Figura 47 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 3.



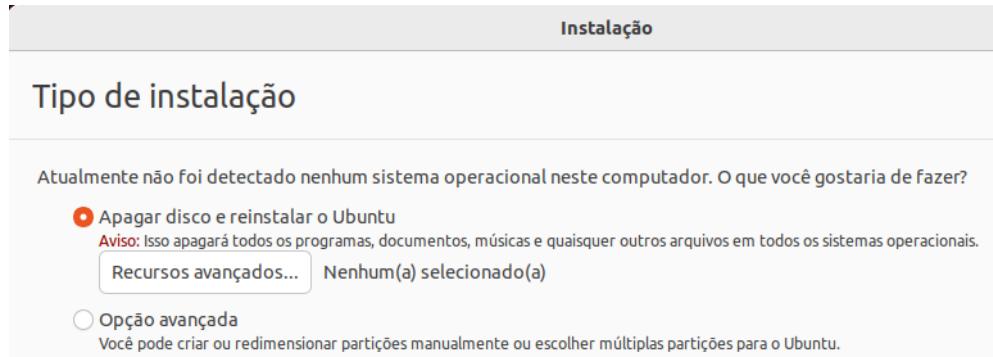
Para nosso fim, uma instalação mínima basta, mas caso queira uma instalação normal só demandará mais tempo. Além disso, para economia de tempo recomenda-se não marcar as duas “Outras opções” que são apresentadas nessa tela.

Figura 48 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 4.



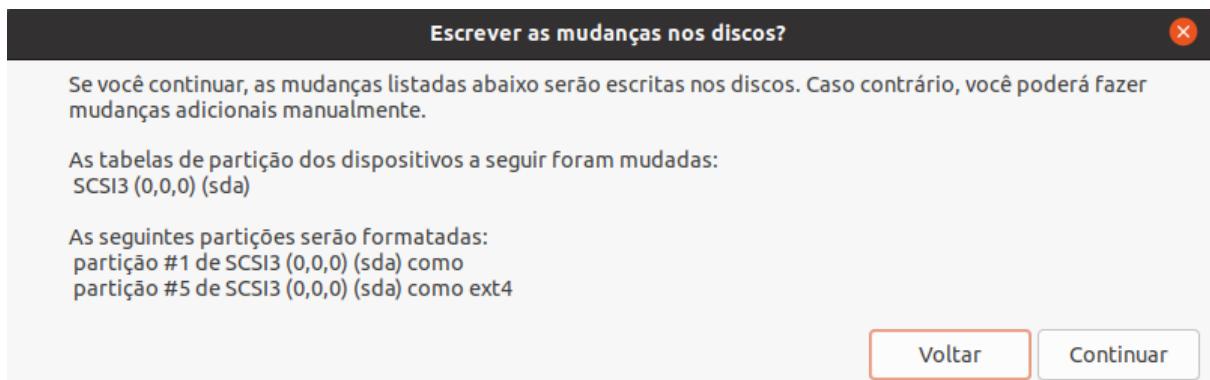
Como não existe nenhum outro sistema na máquina recomenda-se “[Apagar o disco e reinstalar o Ubuntu](#)”, sem necessidade de funcionalidades avançadas.

Figura 49 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 5.



Em seguida aparecerá uma mensagem de confirmação pressione continuar para continuar o processo.

Figura 50 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 6.



O próximo passo é determinar a localização do computador, escolha de acordo com o lugar em que esteja fazendo esse procedimento para que o fuso horário se adeque corretamente.

Figura 51 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 7.



Posteriormente você deve determinar seu nome, o nome do seu computador consequentemente será determinado por isso, além de seu usuário e senha de acesso à máquina. Outras opções como “Iniciar sessão automaticamente” ou “Solicitar minha senha para entrar” estão disponíveis para escolha.

Figura 52 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 8.

Seu nome:	<input type="text"/>
Nome do seu computador:	<input type="text"/>
O nome usado quando conversa com outros computadores.	
Escolha um nome de usuário:	<input type="text"/>
Escolha uma senha:	<input type="password"/> <small>(@)</small>
Confirme sua senha:	<input type="password"/>
<input type="radio"/> Iniciar sessão automaticamente <input checked="" type="radio"/> Solicitar minha senha para entrar <input type="checkbox"/> Usar o Active Directory <small>Você entrará com o domínio e outros detalhes no próximo passo.</small>	

Por fim o sistema irá se instalar e a máquina se reiniciará. Esse processo pode demandar um bom tempo, não se preocupe caso não pareça estar progredindo.

Figura 53 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 9.

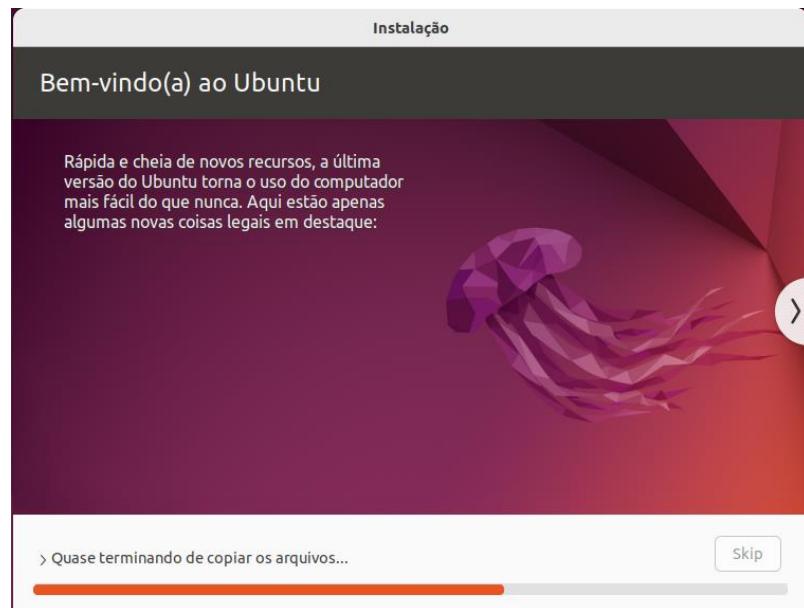
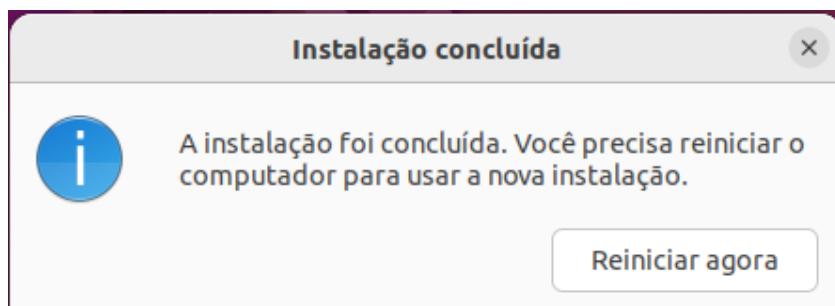
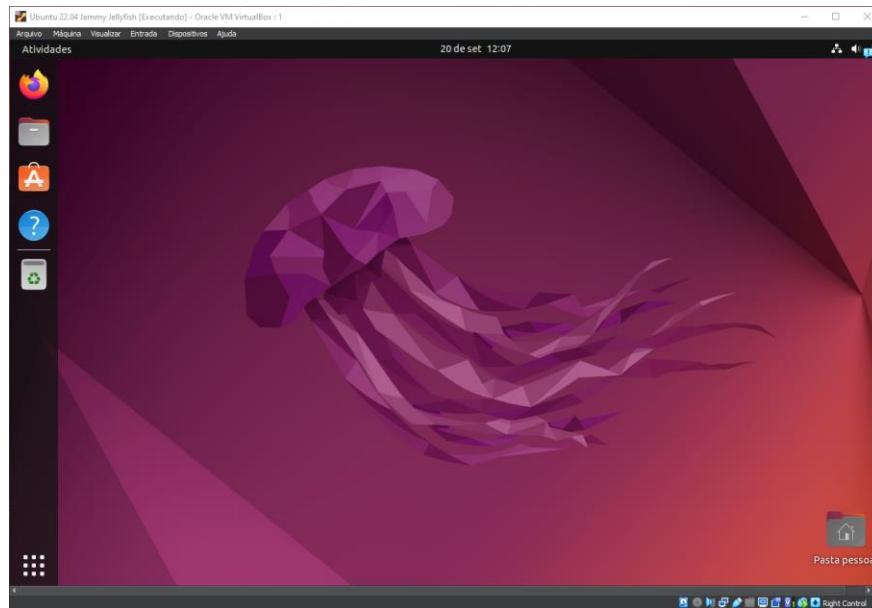


Figura 54 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 10.



Por fim o sistema irá pedir para se reiniciar e após isso pedirá um “ENTER”. Nesse processo ele fará as instalações e logo em seguida você se irá se deparar com opções para fazer login em sua conta, termos de privacidade, relatórios para ajuda aos desenvolvedores. Depois de tudo isso finalizado, o sistema Linux Ubuntu 22.04 Jammy Jellyfish estará pronto para ser utilizado.

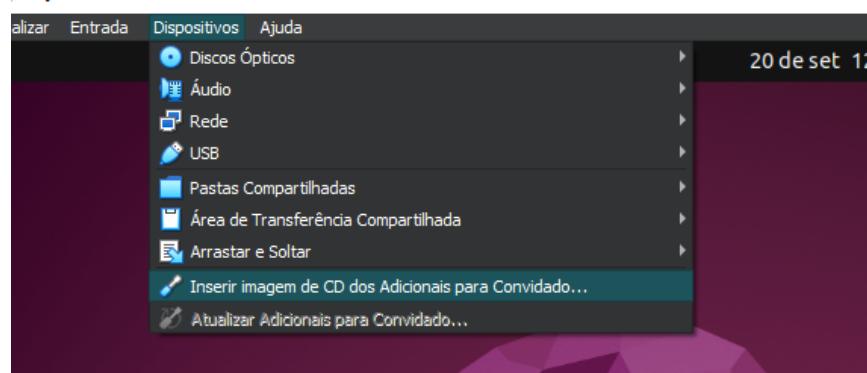
Figura 55 – Iniciando a máquina virtual na VirtualBox pela primeira vez parte 12.



## 7. VIRTUAL BOX CD

Essa etapa é realizada para ajuste do tamanho da tela, além de instalar outras funções do Oracle VM VirtualBox que não vem nativas com a criação da máquina. Para isso, vá até a aba “**Dispositivos**” na janela de seu VirtualBox e escolha a opção “**Inserir imagem de CD dos Adicionais para Convidado**”.

Figura 56 – Inserindo CD de adicionais na VirtualBox parte 1.



O CD para essa versão do Ubuntu não tem início automático, logo é preciso executá-lo manualmente. Clique com o botão direito sobre o ícone do CD e selecione “**Nova janela**”. Dentro da pasta do CD, clique com o botão direito novamente e selecione “**Abrir no terminal**”.

Figura 57 – Inserindo CD de adicionais na VirtualBox parte 2.

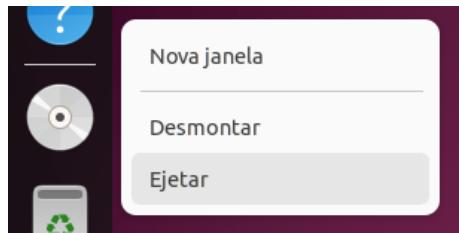
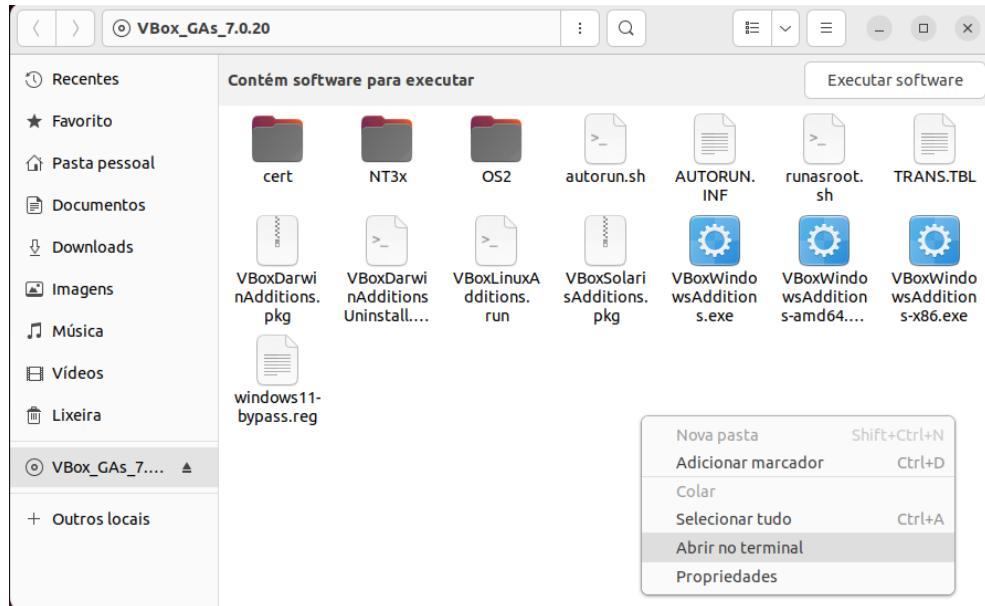


Figura 58 – Inserindo CD de adicionais na VirtualBox parte 3.



Agora, com o terminal já aberto escreva o comando:

```
ls
```

O sistema irá listar todo conteúdo não oculto dentro do CD. Observe que há uma opção chamada “**VBoxWindowsAdditons.exe**”, esse é o arquivo executável dos conteúdos adicionais. Para executá-lo escreva:

```
sudo ./VBoxWindowsAdditons.run
```

Dessa forma o conteúdo será instalado normalmente e a tela se ajustará em seu computador.

Figura 59 – Inserindo CD de adicionais na VirtualBox parte 4.

```

scicobot@scicobot-VirtualBox: /media/scicobot/VBox_GAs_7.0.20
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

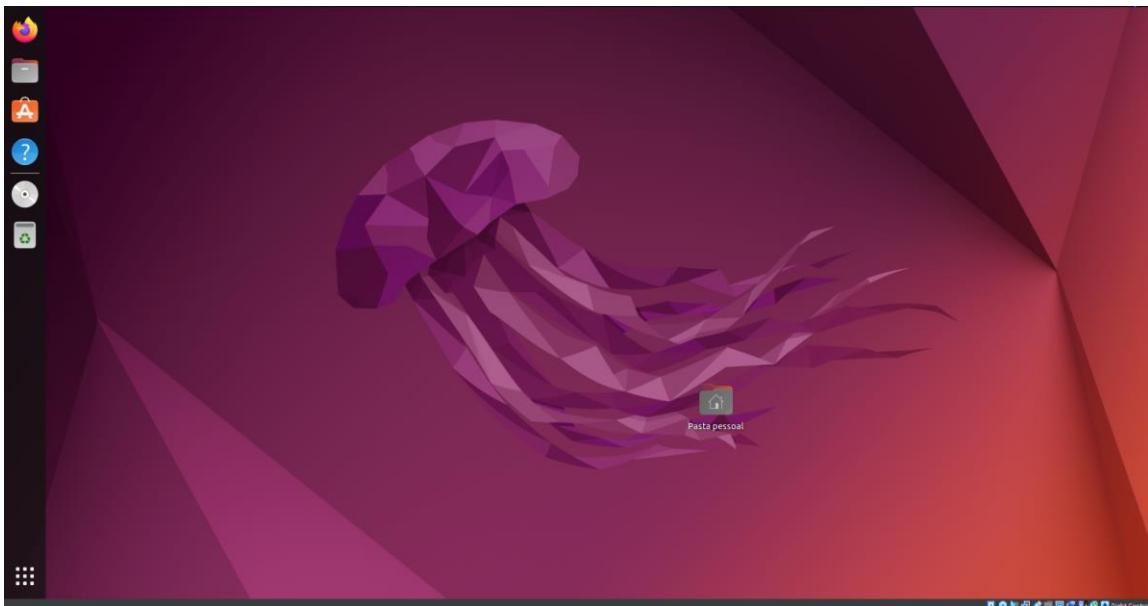
scicobot@scicobot-VirtualBox: /media/scicobot/VBox_GAs_7.0.20$ ls
AUTORUN.INF  runasroot.sh          VBoxSolarisAdditions.pkg
autorun.sh    TRANS.TBL           VBoxWindowsAdditions-amd64.exe
cert         VBoxDarwinAdditions.pkg  VBoxWindowsAdditions.exe
scicobot@scicobot-VirtualBox: /media/scicobot/VBox_GAs_7.0.20$ sudo ./VBoxLinuxAdditions.run
[sudo] senha para scicobot:
Sinto muito, tente novamente.
[sudo] senha para scicobot:
Sinto muito, tente novamente.
[sudo] senha para scicobot:
Sinto muito, tente novamente.
[sudo] senha para scicobot:
Verifying archive integrity... 100% MD5 checksums are OK. All good.
Uncompressing VirtualBox 7.0.20 Guest Additions for Linux 100%
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Starting.
VirtualBox Guest Additions: Setting up modules
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel
modules. This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions:   /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: or
VirtualBox Guest Additions:   /sbin/rcvboxadd quicksetup all
VirtualBox Guest Additions: Building the modules for kernel 6.8.0-45-generic.

This system is currently not set up to build kernel modules.
Please install the gcc make perl packages from your distribution.
VirtualBox Guest Additions: Running kernel modules will not be replaced until
the system is restarted or 'rcvboxadd reload' triggered
VirtualBox Guest Additions: reloading kernel modules and services
VirtualBox Guest Additions: kernel modules were not reloaded
VirtualBox Guest Additions: kernel modules and services were not reloaded
The log file /var/log/vboxadd-setup.log may contain further information.

scicobot@scicobot-VirtualBox: /media/scicobot/VBox_GAs_7.0.20$
```

Por fim vá ao botão de desligar o computador em sua interface Ubuntu e reinicie sua máquina, assim ela se reiniciará com tela cheia o que facilitará seu uso.

Figura 60 – Máquina virtual totalmente completa.



Também é possível habilitar funções de copiar e colar ou arrastar algum conteúdo de um computador ao outro, seja ele máquina > PC ou PC > máquina. Siga até “[Dispositivos](#)” na

sua interface da máquina virtual e vá em “Área de Transferência Compartilhada” ou “Arrastar e Soltar” e selecione a opção “Bi-direcional”:

Figura 61 – Funcionalidades opcionais da VirtualBox parte 1.

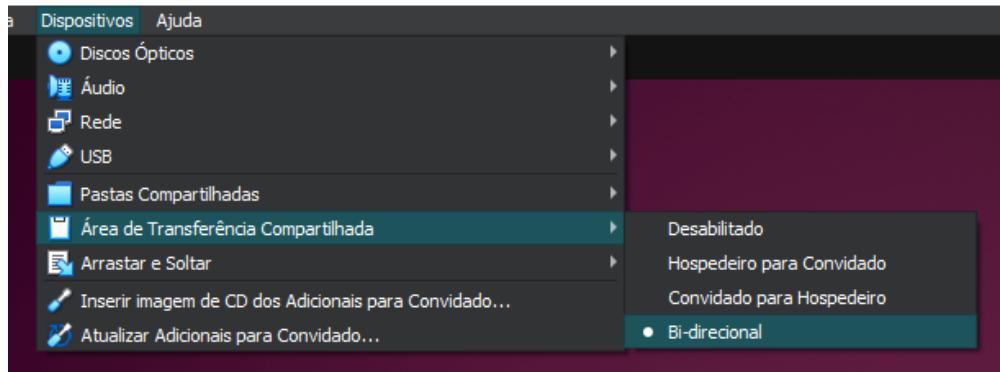
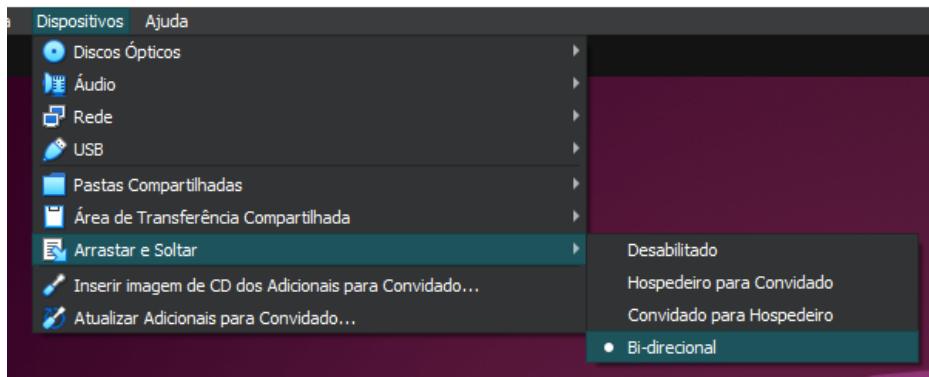


Figura 62 – Funcionalidades opcionais da VirtualBox parte 2.



Dessa forma, a troca de dados entre computador, máquina e Raspberry se torna muito mais rápida e eficiente.

## 8. INSTALANDO ROS 2 HUMBLE HAWKSBILL

Essa parte do material é focada especificamente no ensino da instalação de ROS2 versão Humble Hawksbill (humble). Caso se interesse por outras versões aconselha-se que verifique o site oficial da ROS para que possa usar em seu projeto um sistema operacional compatível à ela.

Para começar pesquise por “ROS2 humble debian” ou acesse o link [<https://docs.ros.org/en/humble/Installation/Ubuntu-Install-Debians.html>] em sua máquina. Aqui você encontrará todas as instruções para instalar o ROS2 Humble no Ubuntu 22.04 (ou no Raspberry Pi, mas para isso recomenda-se o uso de conexão SSH).

Figura 63 – Site oficial da ROS2 Humble Hawksbill.

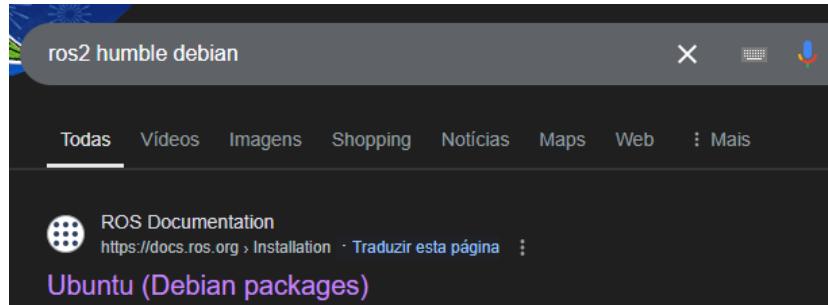


Figura 64 – Guia passo a passo de como instalar ROS2 Humble Hawksbill.

Em primeiro instante digite “`locale`” em seu terminal. Isso é feito para verificar se o sistema apresenta uma localização que suporta UTF-8. Caso não seja seu caso, é necessário executar os seguintes comandos:

```
sudo apt update && sudo apt install locales
```

```
sudo locale-gen en_US en_US.UTF-8
```

```
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
```

```
export LANG=en_US.UTF-8
```

Com isso, verifique novamente a presença do UTF-8:

```
locale
```

Uma observação que o uso dos comandos anteriores resultará na instalação do idioma inglês americano na máquina. Caso queira alterar o idioma posteriormente pode-se acessar as configurações do computador ou então trocar via terminal, o que é ensinado posteriormente nesse tutorial.

Em seguida devemos adicionar o repositório ROS apt ao sistema. Para isso devemos ver se o Repertório Universal Ubuntu está ativado. Escreva em seu terminal:

```
sudo apt install software-properties-common
```

```
sudo add-apt-repository universe
```

Em seguida adicione a chave ROS2 GPG com apt. Escreva:

```
sudo apt update && sudo apt install curl -y
```

```
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o
/usr/share/keyrings/ros-archive-keyring.gpg
```

E adicione o repositório à sua lista de fontes. Escreva:

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-
archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo
$UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

Agora se iniciará a instalação dos pacotes ROS propriamente dita. Comece procurando por atualizações no sistema:

```
sudo apt update
```

Em seguida atualize o sistema (essa etapa tende a demorar, não se preocupe com o tempo gasto):

```
sudo apt upgrade
```

Depois instale o ROS-Base, com bibliotecas de comunicação, pacotes de mensagens e ferramentas de linhas de comando:

```
sudo apt install ros-humble-ros-base
```

Esse passo é opcional, recomenda-se fortemente que o execute para um melhor entendimento dos exemplos ROS 2. Aqui será instalado o ROS Foxy Desktop (essa etapa também tende a demorar bastante):

```
sudo apt install ros-humble-desktop
```

Para finalizar, configure seu ambiente com o seguinte arquivo:

```
sudo apt install ros-dev-tools
```

Se o ROS Foxy Desktop for instalado, recomenda-se que utilize o seguinte comando de teste:

```
source /opt/ros/humble/setup.bash
ros2 run demo_nodes_cpp talker
```

Em seguida abra um novo terminal em seu computador e escreva o seguinte código:

```
source /opt/ros/humble/setup.bash
ros2 run demo_nodes_py listener
```

Você deve ver algo parecido como a imagem abaixo:

Figura 65 – Execução do exemplo Talker – Listener.

```
[Atividades] Terminal natanael@natanael-VirtualBox: ~ [9 de jan 19:31]
[INFO] [1673303430.764698561] [talker]: Publishing: 'Hello World: 2'
[INFO] [1673303431.769778622] [talker]: Publishing: 'Hello World: 3'
[INFO] [1673303432.764429004] [talker]: Publishing: 'Hello World: 4'
[INFO] [1673303433.764628902] [talker]: Publishing: 'Hello World: 5'
[INFO] [1673303434.764969947] [talker]: Publishing: 'Hello World: 6'
[INFO] [1673303435.766044087] [talker]: Publishing: 'Hello World: 7'
[INFO] [1673303436.765281212] [talker]: Publishing: 'Hello World: 8'
[INFO] [1673303437.765792987] [talker]: Publishing: 'Hello World: 9'
[INFO] [1673303438.765618998] [talker]: Publishing: 'Hello World: 10'
[INFO] [1673303439.771895864] [talker]: Publishing: 'Hello World: 11'
[INFO] [1673303440.766177938] [talker]: Publishing: 'Hello World: 12'
[INFO] [1673303441.766581830] [talker]: Publishing: 'Hello World: 13'
[INFO] [1673303442.766981830] [talker]: Publishing: 'Hello World: 14'
[INFO] [1673303443.767916293] [talker]: Publishing: 'Hello World: 15'
[INFO] [1673303444.767339611] [talker]: Publishing: 'Hello World: 16'
[INFO] [1673303445.768441909] [talker]: Publishing: 'Hello World: 17'
[INFO] [1673303446.813213225] [talker]: Publishing: 'Hello World: 18'
[INFO] [1673303447.768690776] [talker]: Publishing: 'Hello World: 19'
[INFO] [1673303448.778502012] [talker]: Publishing: 'Hello World: 20'
[INFO] [1673303449.770025366] [talker]: Publishing: 'Hello World: 21'
[INFO] [1673303450.770447700] [talker]: Publishing: 'Hello World: 22'
[INFO] [1673303451.769815172] [talker]: Publishing: 'Hello World: 23'
[INFO] [1673303452.771030645] [talker]: Publishing: 'Hello World: 24'
[INFO] [1673303453.770715047] [talker]: Publishing: 'Hello World: 25'
[INFO] [1673303454.773136351] [talker]: Publishing: 'Hello World: 26'
[INFO] [1673303455.773711643] [talker]: Publishing: 'Hello World: 27'
[INFO] [1673303456.778944788] [talker]: Publishing: 'Hello World: 28'
[INFO] [1673303457.772700127] [talker]: Publishing: 'Hello World: 29'
[INFO] [1673303458.774969963] [talker]: Publishing: 'Hello World: 30'
[INFO] [1673303459.773732379] [talker]: Publishing: 'Hello World: 31'
[INFO] [1673303460.773176956] [talker]: Publishing: 'Hello World: 32'
[INFO] [1673303461.773533983] [talker]: Publishing: 'Hello World: 33'
[INFO] [1673303462.774131103] [talker]: Publishing: 'Hello World: 34'

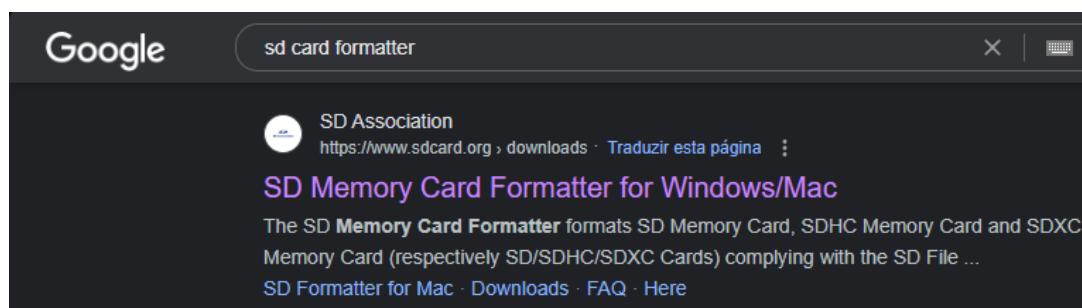
natanael@natanael-VirtualBox: ~ [9 de jan 19:31]
[natanael@natanael-VirtualBox: ~] S source /opt/ros/foxy/setup.bash
[natanael@natanael-VirtualBox: ~] S ros2 run demo_nodes_py listener
[INFO] [1673303454.830414011] [listener]: I heard: [Hello World: 26]
[INFO] [1673303455.778328787] [listener]: I heard: [Hello World: 27]
[INFO] [1673303456.773835434] [listener]: I heard: [Hello World: 28]
[INFO] [1673303457.777299572] [listener]: I heard: [Hello World: 29]
[INFO] [1673303458.779427694] [listener]: I heard: [Hello World: 30]
[INFO] [1673303459.778717169] [listener]: I heard: [Hello World: 31]
[INFO] [1673303460.780099916] [listener]: I heard: [Hello World: 32]
[INFO] [1673303461.778333611] [listener]: I heard: [Hello World: 33]
[INFO] [1673303462.779003005] [listener]: I heard: [Hello World: 34]
```

OBS: Em alguns momentos pode ser requisitado a senha do computador ou confirmações de “(Y/n)” ou “(S/n)”. Para prosseguir com o passo, informe a sua senha quando for pedido e escreva “Y” ou “S” em maiúsculo quando requerido.

## 9. FORMATAR CARTÃO SD COM O SD CARD FORMATTER

Essa etapa não é fundamental para a execução do projeto, mas ajuda consideravelmente a manter a organização no decorrer do tempo. Aqui será descrito como instalar o SD Card Formatter, um programa que formata cartões SD e pendrives. Para isso, pesquise no Google por “SD Card Formatter” e selecione o link apresentado abaixo:

Figura 66 – Pesquisa por “sd card formatter” na web.



Role o cursor até a parte inferior onde é possível encontrar as opções de download do programa. Selecione a opção “For Windows” e em seguida concorde com os termos apresentados pela empresa:

Figura 67 – Opções de download do programa ofertadas pelo site.

### SD Memory Card Formatter Download for Windows/Mac



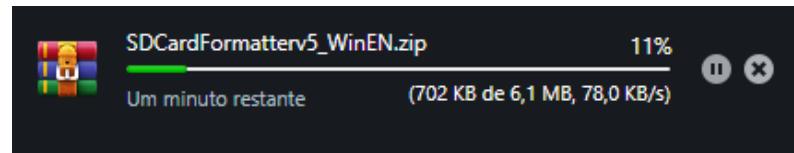
Figura 68 – Termos da empresa.

YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE READ THIS AGREEMENT AND INTEND TO BE BOUND AS IF YOU HAD SIGNED THIS AGREEMENT IN WRITING. IF YOU ARE ACTING ON BEHALF OF AN ENTITY, YOU WARRANT THAT YOU HAVE THE AUTHORITY TO ENTER INTO THIS AGREEMENT ON BEHALF OF SUCH ENTITY AND BIND SUCH ENTITY TO THE TERMS OF THIS AGREEMENT.



Com essa etapa concluída o download de um arquivo .zip se iniciará contendo o instalador do programa. Por conta disso, pode ser necessário que instale um programa de descompactação, como o WinRAR por exemplo.

Figura 69 – Arquivo .zip do instalador.



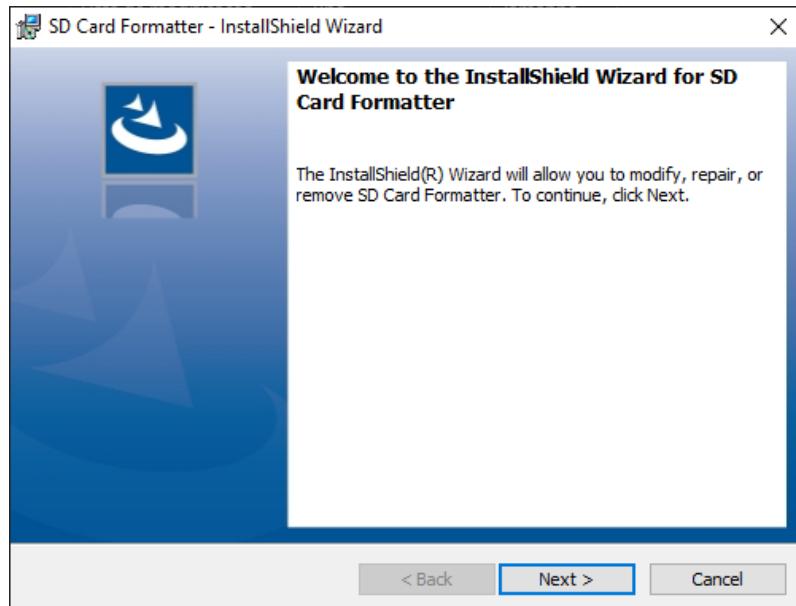
Após o download e a extração do arquivo, basta clicar duas vezes com o botão esquerdo sobre o arquivo executável.

Figura 70 – Arquivo instalador extraído do arquivo .zip.



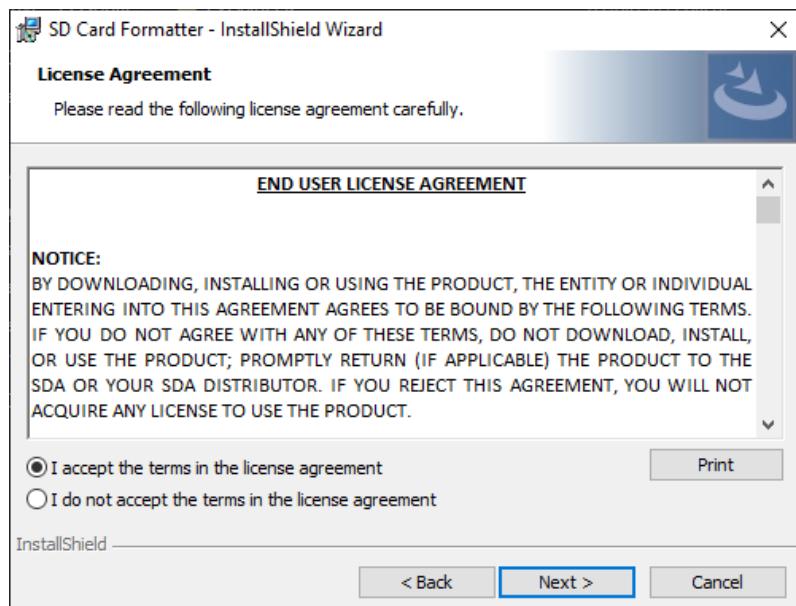
Na página inicial do instalador clique em “Next”.

Figura 71 – Primeira etapa de instalação do SD Card Formatter.



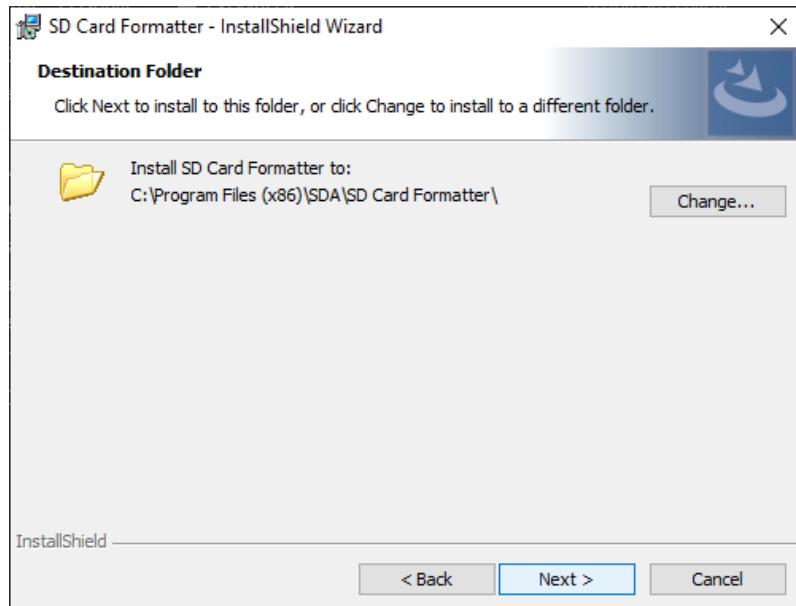
Aceite novamente os termos de instalação com a opção “I accept the terms in the license agreement” e clique em “Next”.

Figura 72 – Segunda etapa de instalação do SD Card Formatter.



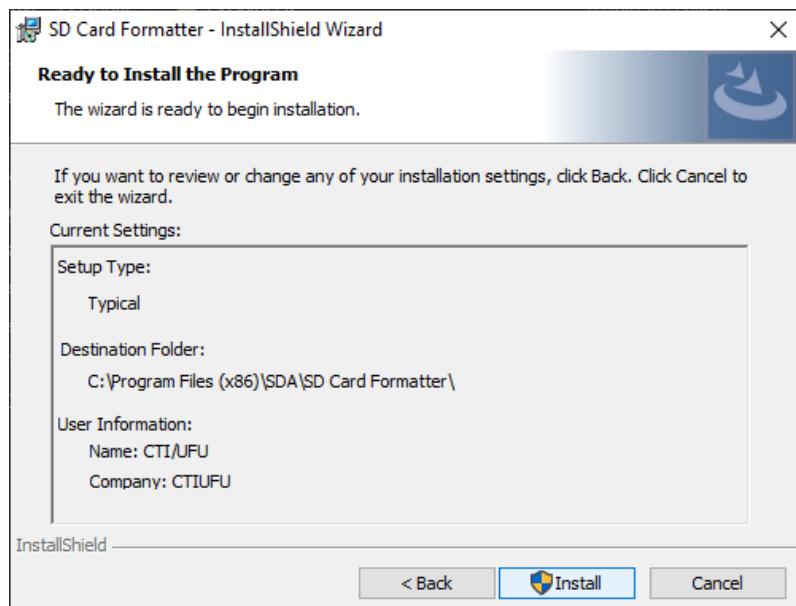
Escolha o local onde os arquivos serão instalados e clique em “Next”.

Figura 73 – Terceira etapa de instalação do SD Card Formatter.



Por fim certifique-se que tudo foi configurado como esperado e clique em “**Install**”.

Figura 74 – Quarta etapa de instalação do SD Card Formatter.



Após a instalação clique em “**Finish**”. Além disso, pode-se verificar a criação de um ícone na área de trabalho.

Figura 75 – Última etapa de instalação do SD Card Formatter.

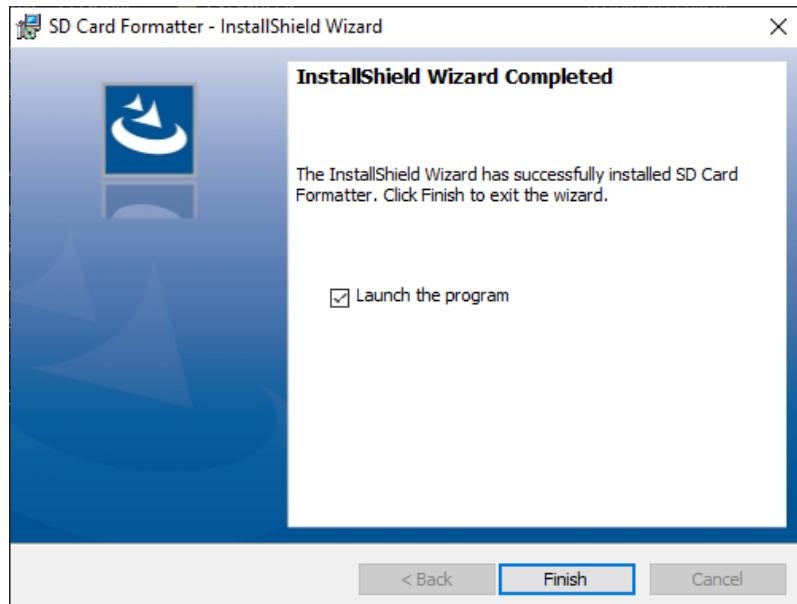
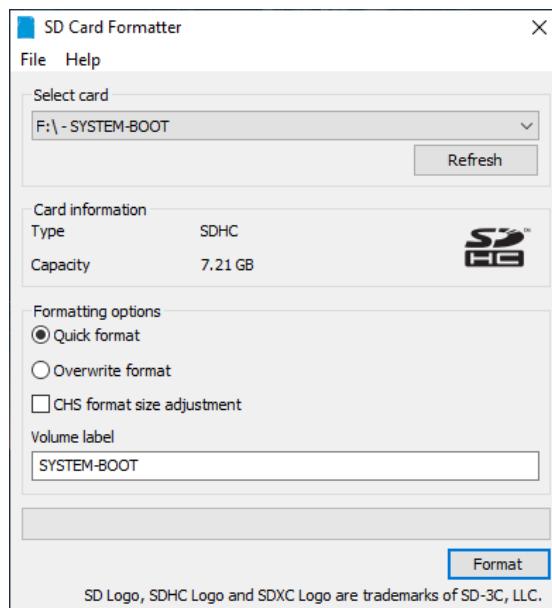


Figura 76 – Ícone do SD Card Formatter criado na área de trabalho.



Abrindo o programa, você se deparará com a tela abaixo. Escolha o cartão que deverá ser formatado na aba “Select card” (clique em “Refresh” caso não o encontre) e clique na opção “Format”.

Figura 77 – Interface do SD Card Formatter.



Concorde com as janelas que aparecerem que pedirão confirmação para apagar os dados pré-existentes naquele sistema e o processo se iniciará.

## 10.INSTALAR UBUNTU 22.04 NO RASPBERRY 3B+

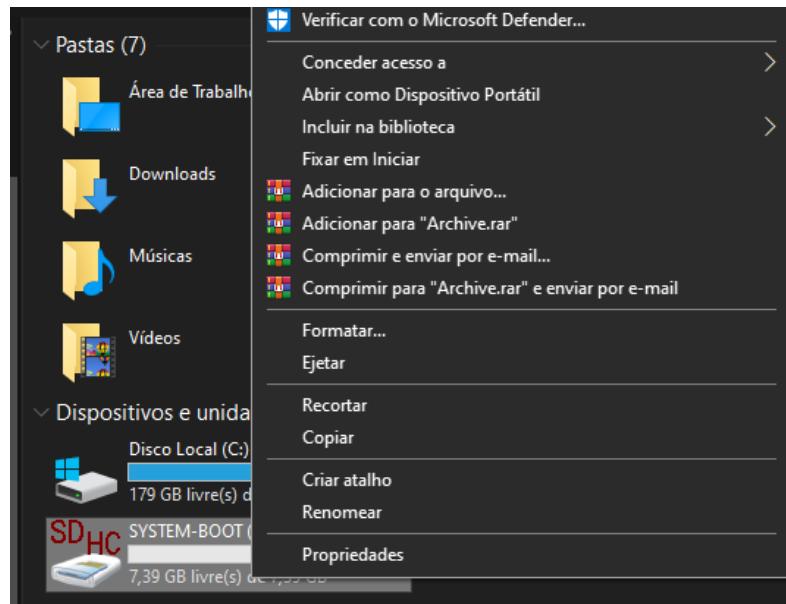
Para essa etapa do tutorial se faz necessário um cartão micro-sd com no mínimo 8gb e classe 10:

Figura 78 – Cartão Micro-SD tipo 10 de 8Gb.



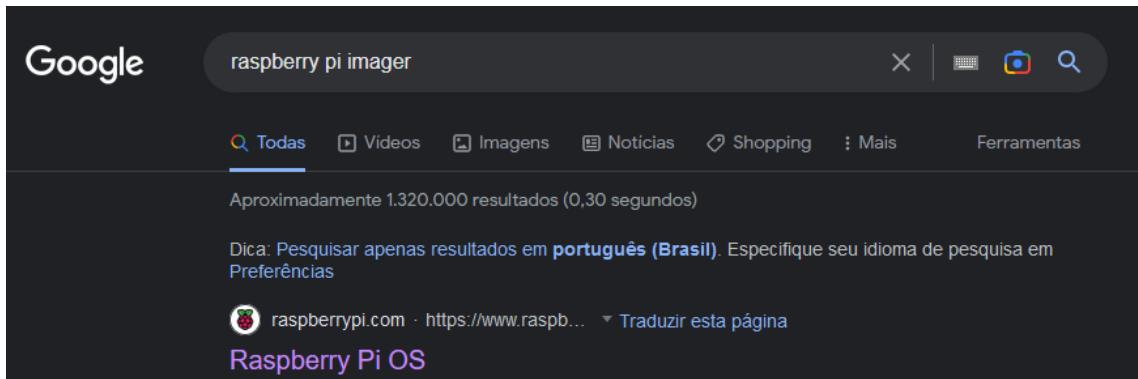
É necessário formatar o cartão antes de começar o procedimento. Pode-se usar o próprio sistema do Windows para isso ou instalar um programa como o “**“SD Card Formatter”**”, disponível em: [\[https://www.sdcards.org/downloads/formatter/\]](https://www.sdcards.org/downloads/formatter/).

Figura 79 – Formatando cartão SD.



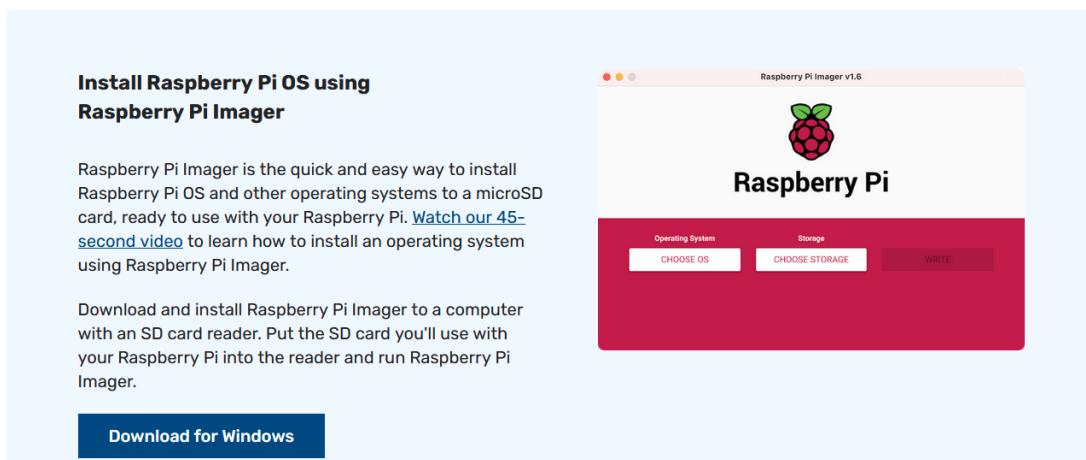
Com o cartão pronto é necessário baixar o “**Raspberry Pi Imager**” para instalarmos o sistema operacional no cartão.

Figura 80 – Instalando Raspberry PI Imager parte 1.



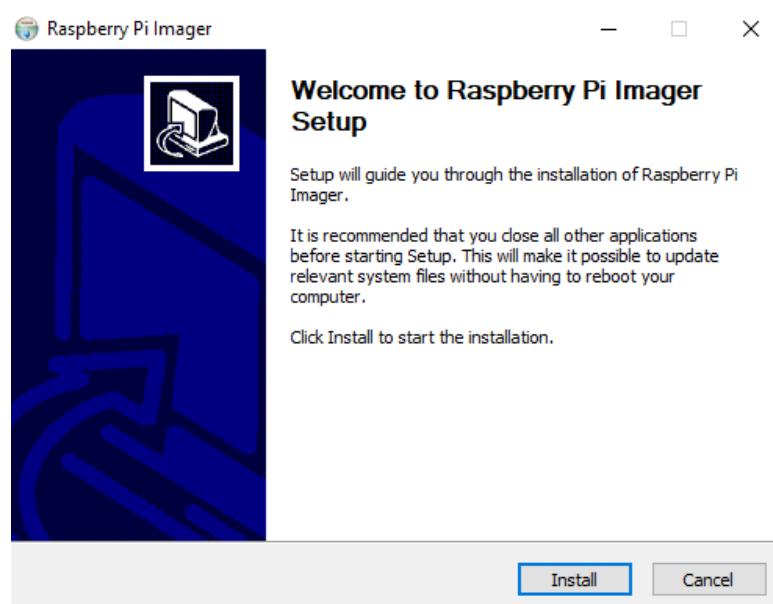
Faça download de acordo com seu sistema operacional.

Figura 81 – Instalando Raspberry PI Imager parte 2.



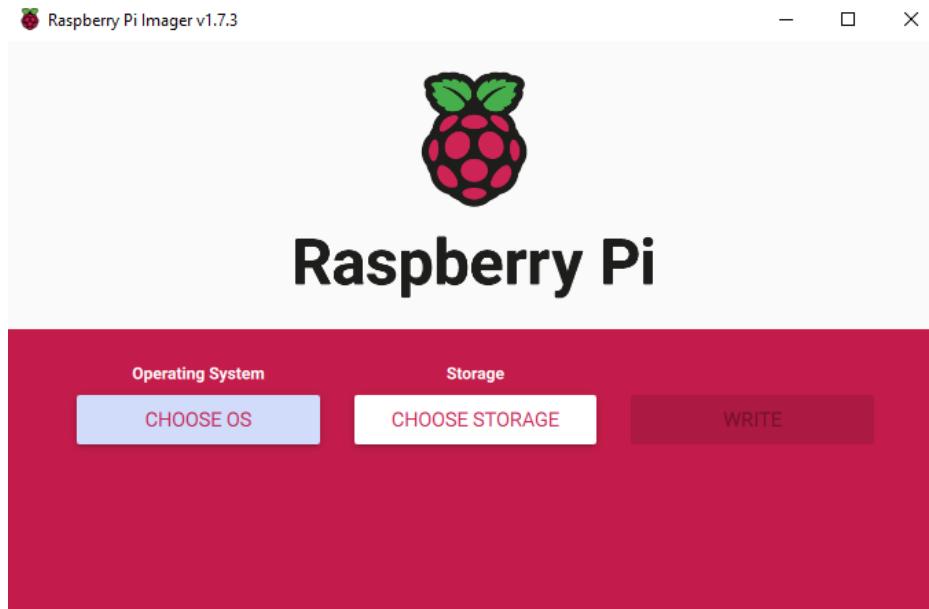
Efetue a instalação, ela é relativamente simples e rápida.

Figura 82 – Instalando Raspberry PI imager parte 3.



Defina o sistema a ser baixado, qual o cartão e algumas configurações adicionais de usuário e rede antes de confirmar.

Figura 83 – Interface Raspberry PI Imager.

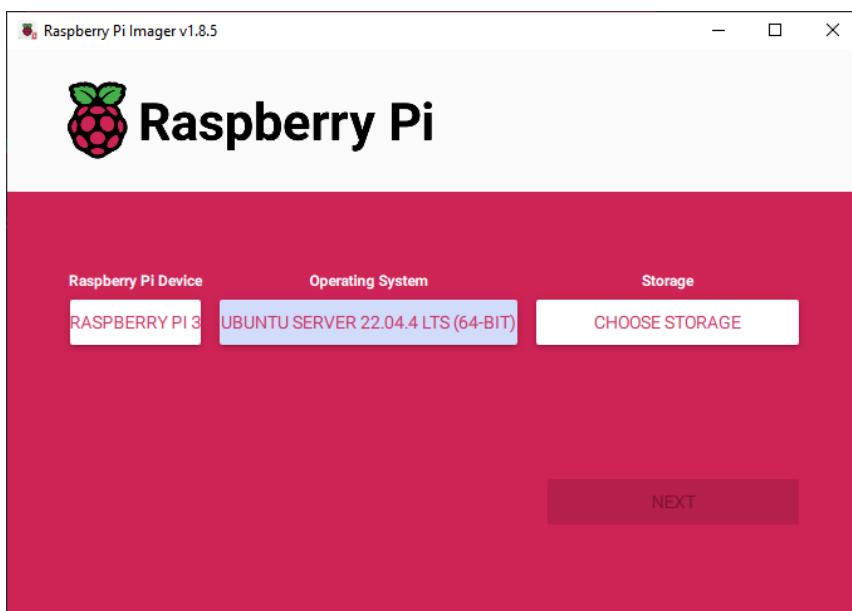


**Raspberry PI Device:** [Modelo da placa utilizada, aqui a 3B+].

**Oprating System:** Other general-purpose OS: Ubuntu: Ubuntu Server 22.04.4 LTS (64-bit).

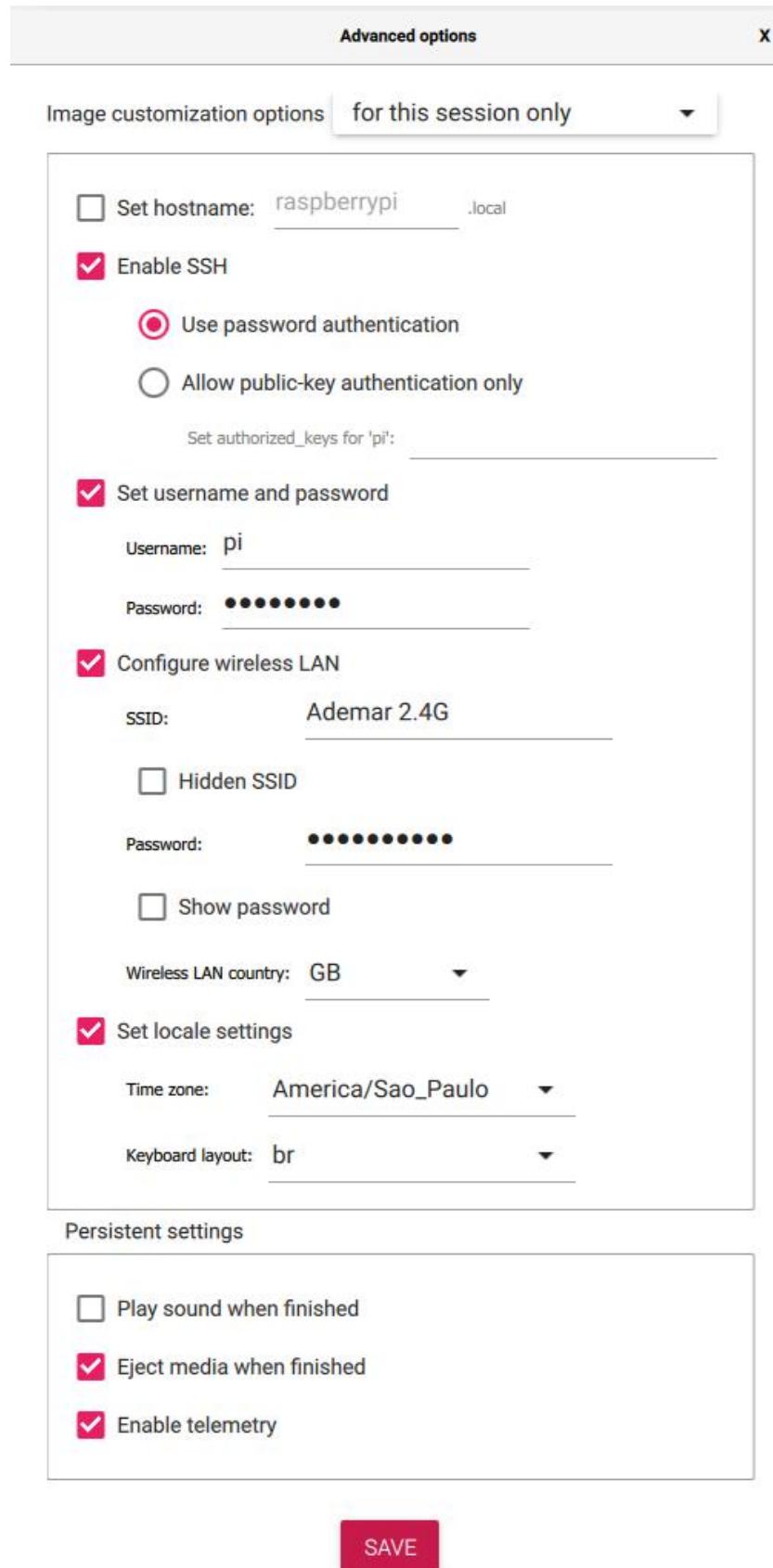
**Storage:** CARTÃO-SSD ESCOLHIDO

Figura 84 – Criando uma imagem Ubuntu 20.04 para SciCoBot.



Na engrenagem marque as opções:

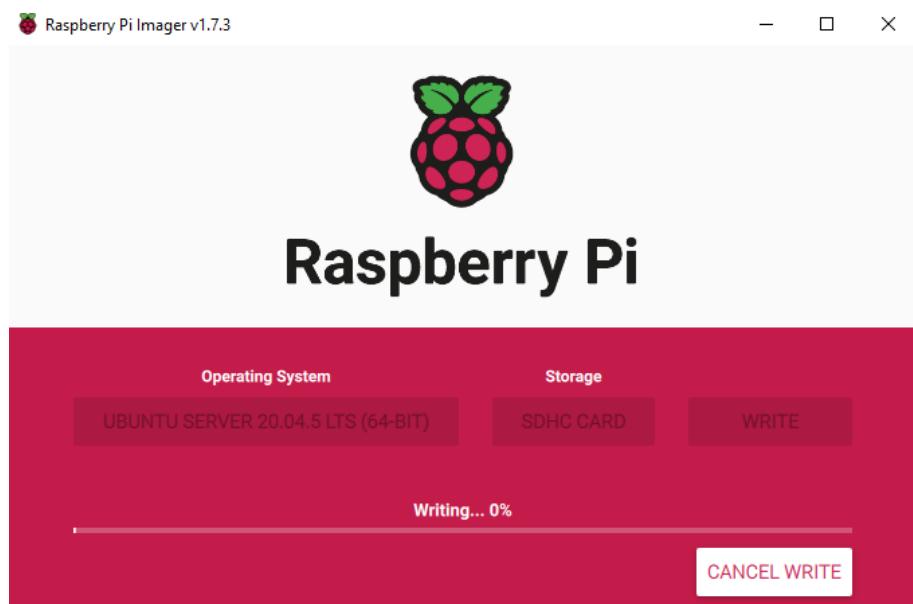
Figura 85 – Configurações adicionais da Imagem.



Habilite a conexão SSH, o usuário e senha, a conexão wifi e a localização. O usuário já vem padronizado, mas pode ser alterado, juntamente com sua senha (esta deve ser definida nessa etapa). A conexão wifi é escolhida com base na conexão de seu computador, caso não esteja conectado na hora de efetuar esse procedimento, escreva o nome da rede que pretende usar e sua senha de acesso. Por último defina a sua região e o layout de seu teclado. Depois só salvar e prosseguir

Confirme com um “**Yes**” e só esperar. Quando finalizar o programa emitirá uma notificação de término e ejectará seu cartão automaticamente.

Figura 86 – Escrita da imagem para SciCoBot.



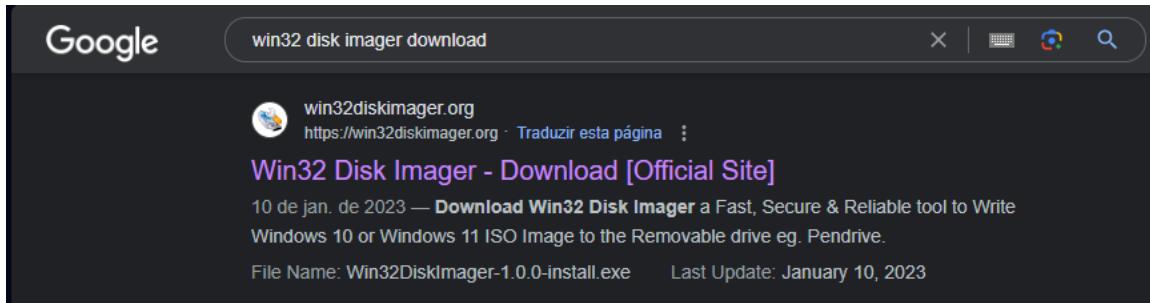
Em seguida coloque o cartão na sua raspberry com um teclado, monitor via HDMI, mouse e uma fonte (de no mínimo 2A, preferencialmente 3A). Estando tudo conectado, alimente o Rasp e espere sua inicialização. Aparecerá uma tela de login, espere mais um pouco porque o Ubuntu irá automaticamente iniciar alguns serviços importantes para a sua inicialização. Quando algo similar aparecer na tela dê “**Enter**” e escreva o login e senha definidos anteriormente:

## **11. CRIAÇÃO CLONES DE UM CARTÃO SD COM O WIN32DISK**

- **INSTALAÇÃO**

Pesquise no google por “**Win32 Disk Imager**” e clique no link indicado na figura abaixo:

Figura 87 – Pesquisa por “win32 disk imager download” no Google.



Procure a opção de download na página.

Figura 88 – Opção de download apresentada pelo site.

File Name	Win32DiskImager-1.0.0-install.exe
Size	12.6 MB
Last Update	January 10, 2023
Author	Tobin Davis

DOWNLOAD

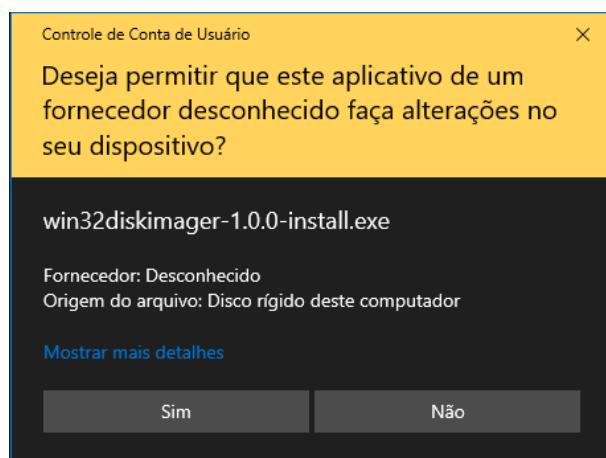
Ao selecioná-la o download se iniciará automaticamente. Dê dois cliques sobre o ícone do executável para o programa se iniciar.

Figura 89 – Arquivo executável baixado.



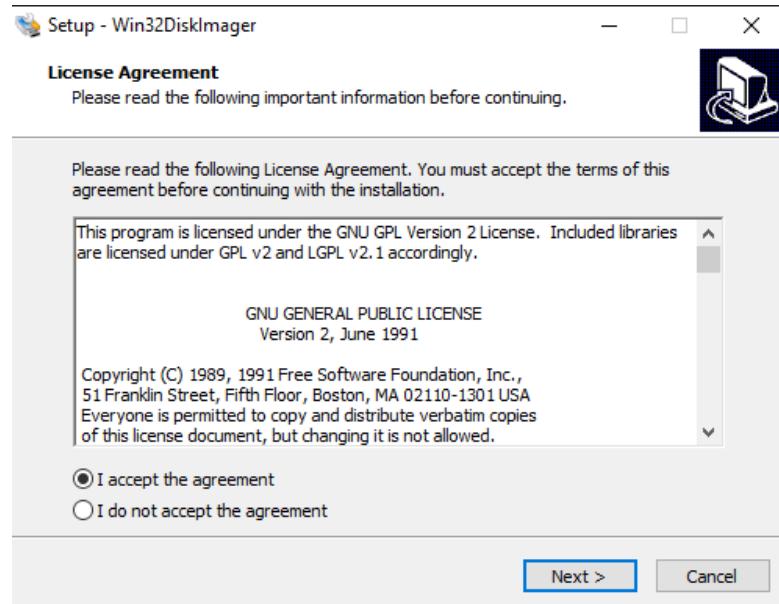
Permita o pedido do Windows.

Figura 90 – Permissão de instalação solicitada pelo sistema.



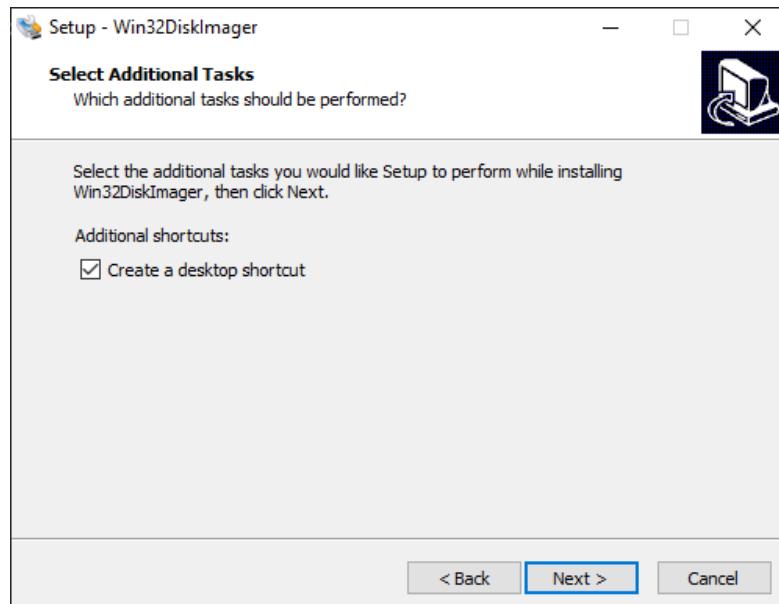
Selecione “I accept the agreement” e em seguida clique em “Next”.

Figura 91 – Primeira etapa da instalação do Win32DiskImager.



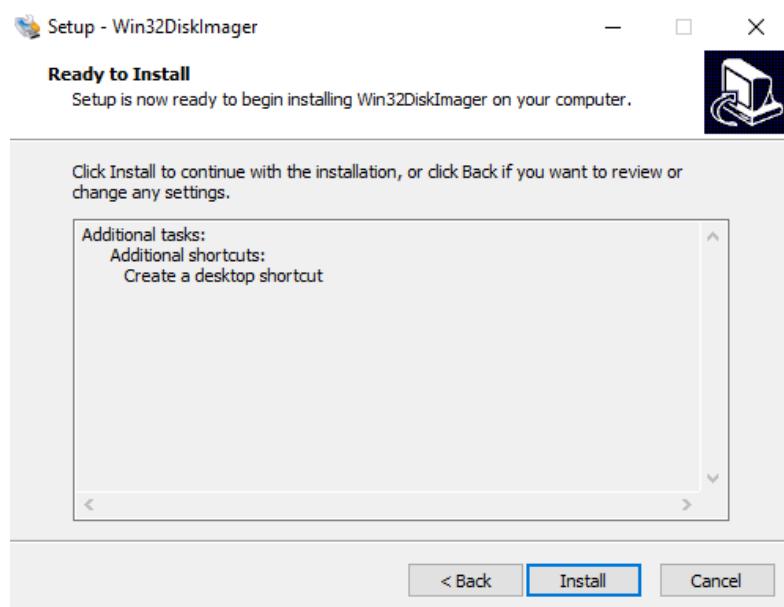
Clique em “Next”.

Figura 92 – Segunda etapa da instalação do Win32DiskImager.



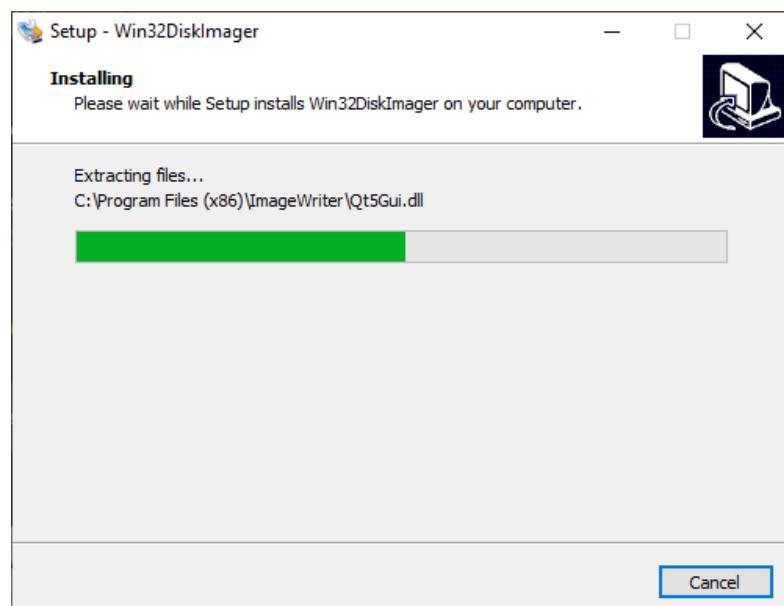
Termine clicando na opção “Install”.

Figura 93 – Terceira etapa da instalação do Win32DiskImager.



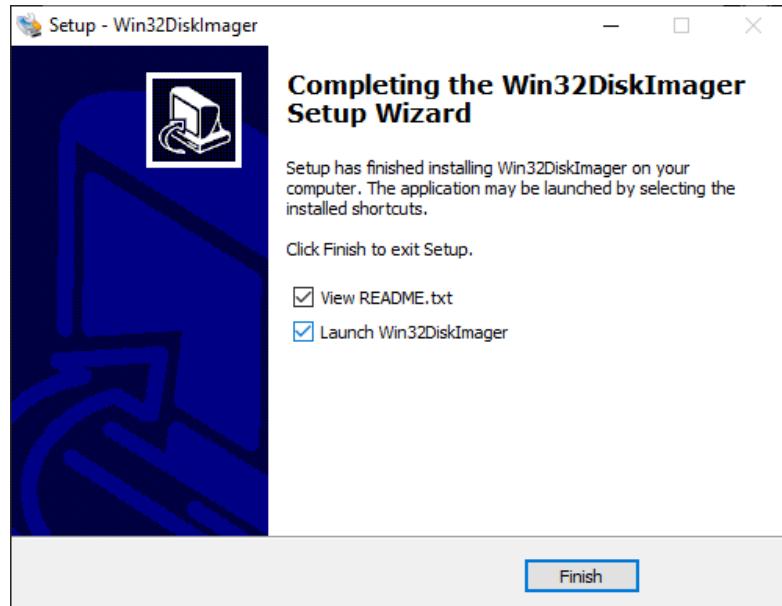
Nessa etapa a instalação se iniciará e em alguns instantes estará finalizada.

Figura 94 – Quarta etapa da instalação do Win32DiskImager.



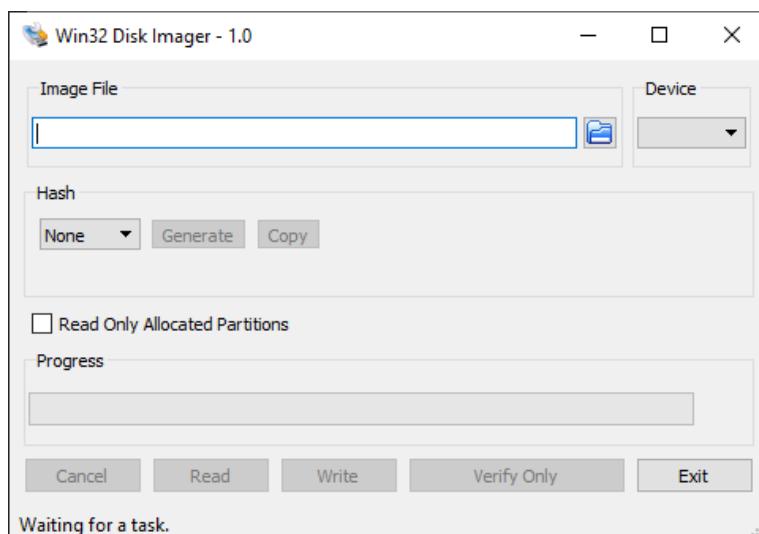
Por fim clique em “Finish”.

Figura 95 – Última etapa da instalação do Win32DiskImager.



**OBSERVAÇÃO:** Infelizmente o programa tem uma incompatibilidade com uma dll do Windows que se ativa sempre que algum serviço de nuvem do computador está ativo. Então se o programa não for inicializado quando solicitado, tente desativar o OneDrive ou GoogleDrive do computador.

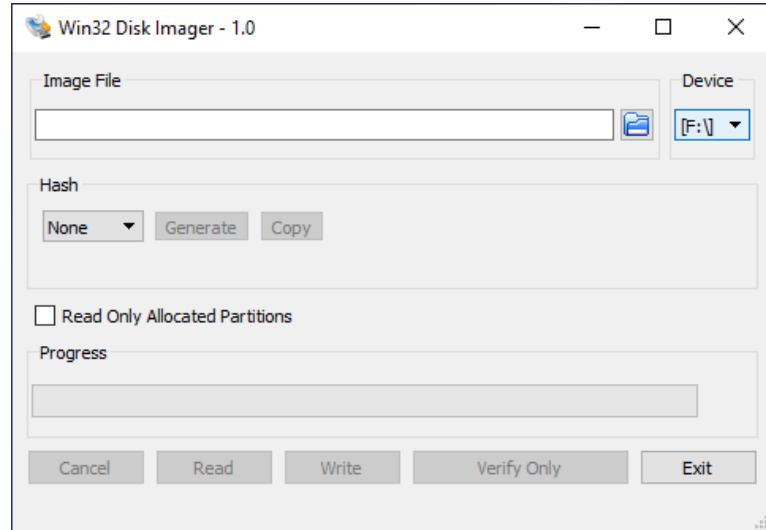
Figura 96 – Interface do Win32DiskImager.



- CRIAR UMA IMAGEM

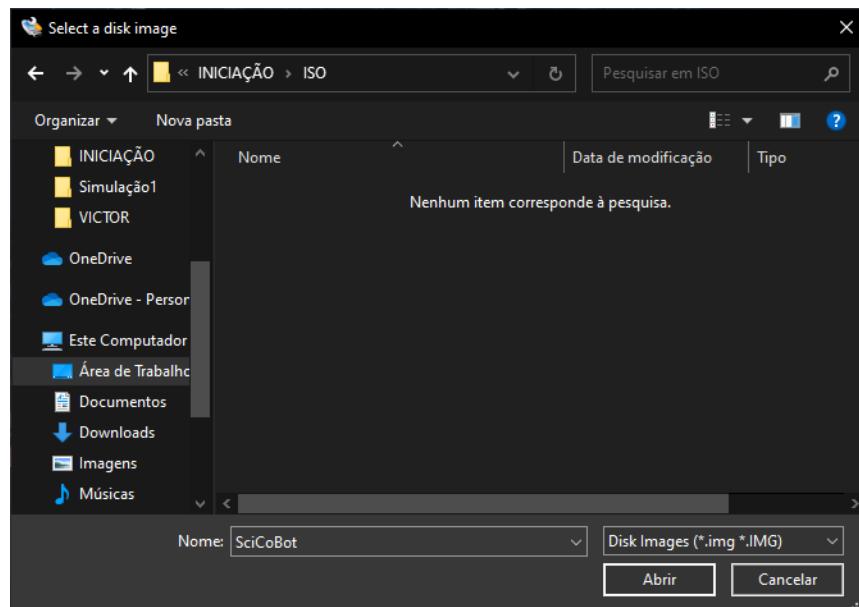
Para a criação de uma imagem o processo é bem simples. Antes de tudo vá até a categoria “**Device**” e selecione qual o cartão deverá ser copiado. Em seguida clique no ícone da pasta azul à sua esquerda e pelo gerenciador de arquivos escolha em que pasta o backup será salvo.

Figura 97 – Criação de arquivo no Win32DiskImager, parte 1.



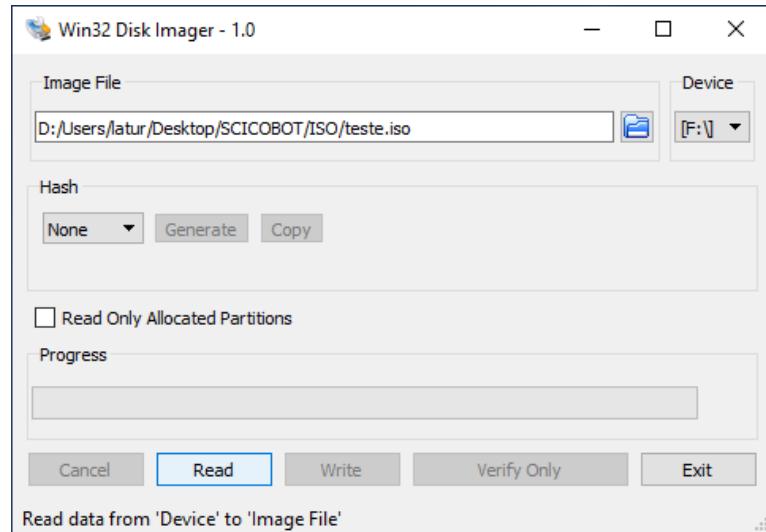
Com isso pronto, uma nova janela se abrirá. Vá na barra de “Nome” escreva o nome do arquivo que deverá ser criado, assim como mostra a imagem abaixo, e clique em “Abrir”.

Figura 98 – Criação de arquivo no Win32DiskImager, parte 2.



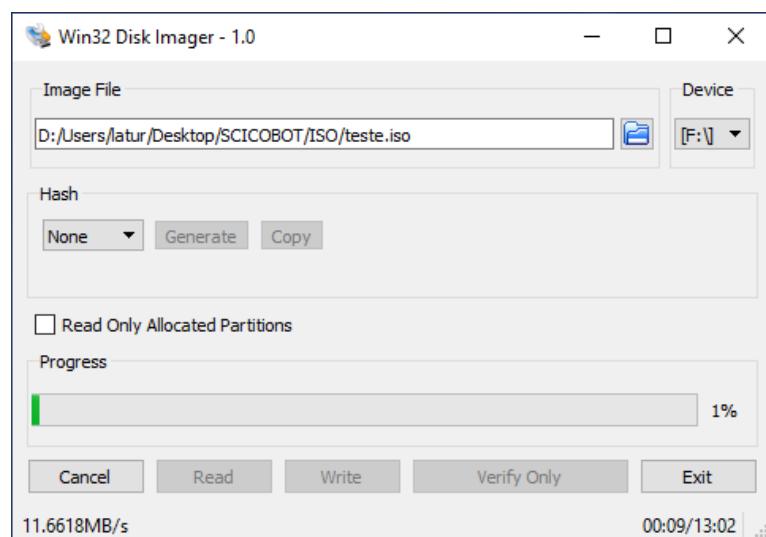
Por fim clique na opção “Read” e espere a criação do arquivo de backup.

Figura 99 – Criação de arquivo no Win32DiskImager, parte 3.



A gravação do sistema na pasta destino começará logo em seguida. Vale ressaltar que essa etapa tende a ser bastante demorada, então não se preocupe se o andamento estiver lento.

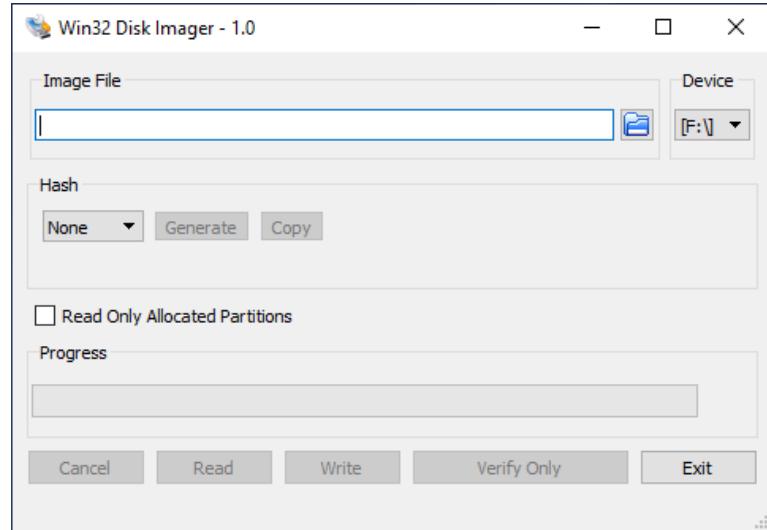
Figura 100 – Criação de arquivo .iso no Win32DiskImager.



- **ALOCAR IMAGEM EM CARTÃO-SD**

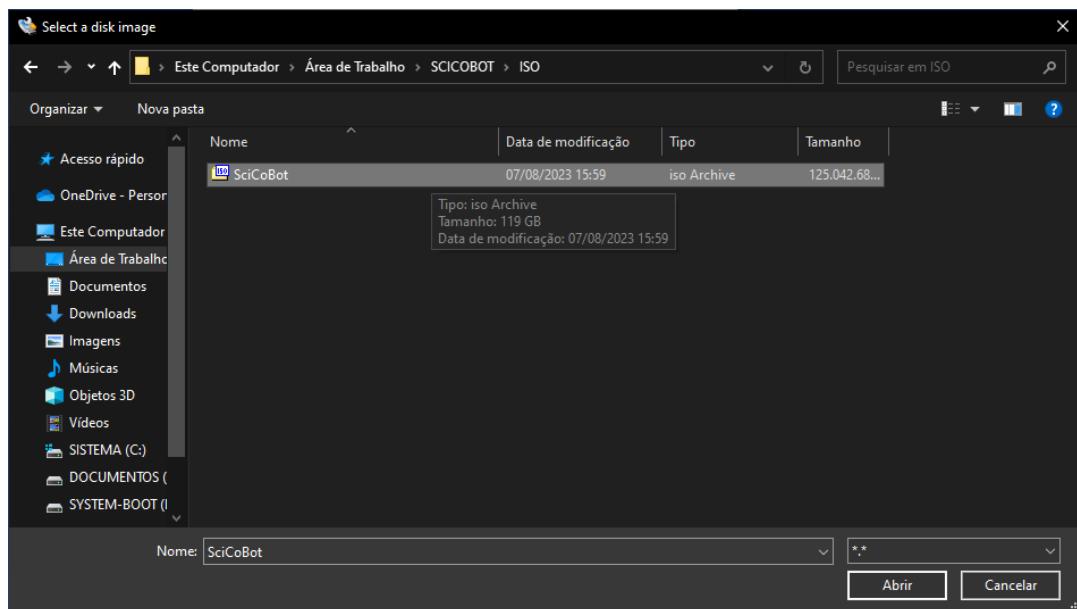
Para salvar um arquivo de imagem previamente criado em seu cartão de memória recém formatado clique abaixo de “Device” e selecione o cartão que deverá receber o backup. Em seguida vá na pasta azul à esquerda e, no gerenciador de arquivos, escolha a pasta onde se localiza o backup.

Figura 101 – Alocação de arquivo .iso no Win32DiskImager, parte 1.



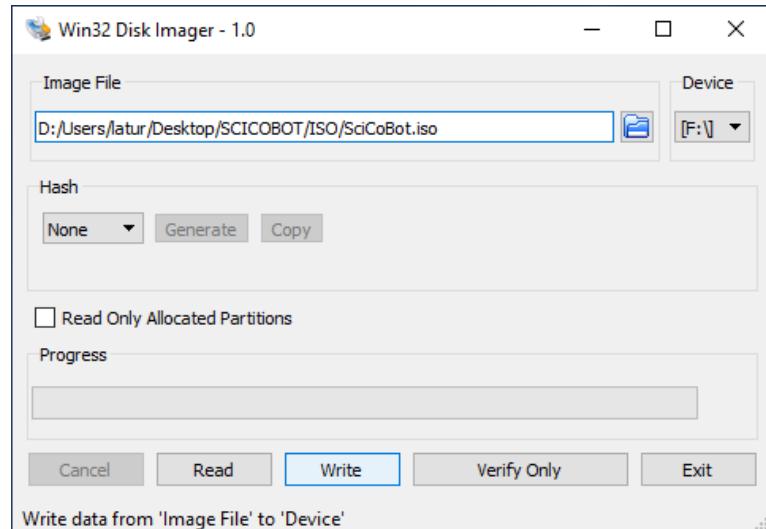
Assim, uma nova janela se abrirá. Vá em “\*.\*” e altere para “\*.img\*, \*.IMG\*” e o arquivo que deseja utilizar deve aparecer para seleção. Clique sobre o arquivo iso e clique em “Abrir”.

Figura 102 – Alocação de arquivo .iso no Win32DiskImager, parte 2.



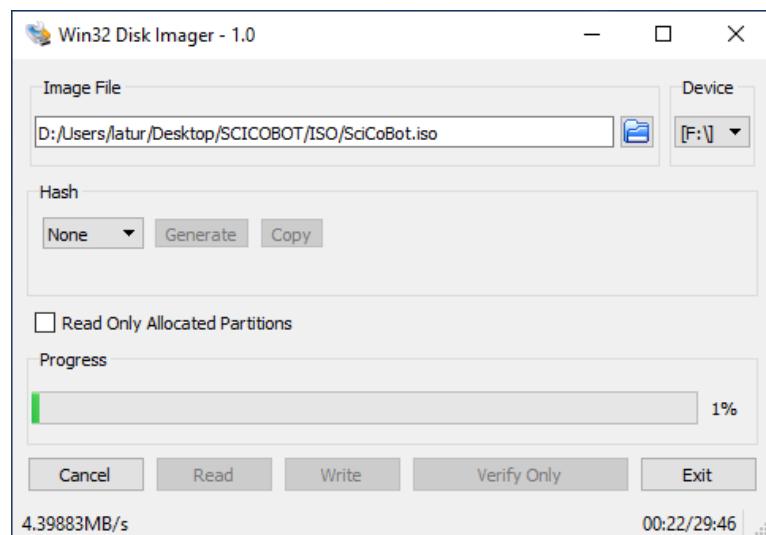
Por fim, clique na opção “Write”.

Figura 103 – Alocação de arquivo .iso no Win32DiskImager, parte 3.



A gravação do sistema em seu cartão SD começará logo em seguida. Mas essa etapa tende a ser consideravelmente demorada, então tenha paciência.

Figura 104 – Alocação de arquivo .iso no Win32DiskImager.



## 12. CONEXÃO COM A INTERNET

Digite “`hostname -I`” na tela de sua raspberry para mostrar seu endereço IP. Em sua máquina virtual, abra o terminal e escreva “`ping +[IP]`” junto com seu IP ou pingue algum site como “`ping Google.com`” no próprio raspberry para verificar se está tudo ok para conexão SSD.

Figura 105 – Ping IP.

```
natanael@natanael-VirtualBox:~$ ping 192.168.100.109
PING 192.168.100.109 (192.168.100.109) 56(84) bytes of data.
64 bytes from 192.168.100.109: icmp_seq=1 ttl=64 time=13.7 ms
64 bytes from 192.168.100.109: icmp_seq=1 ttl=64 time=13.7 ms (DUP!)
64 bytes from 192.168.100.109: icmp_seq=1 ttl=64 time=15.8 ms (DUP!)
64 bytes from 192.168.100.109: icmp_seq=2 ttl=64 time=5.10 ms
64 bytes from 192.168.100.109: icmp_seq=3 ttl=64 time=7.40 ms
64 bytes from 192.168.100.109: icmp_seq=3 ttl=64 time=8.13 ms (DUP!)
64 bytes from 192.168.100.109: icmp_seq=4 ttl=64 time=3.98 ms
64 bytes from 192.168.100.109: icmp_seq=5 ttl=64 time=7.63 ms
64 bytes from 192.168.100.109: icmp_seq=5 ttl=64 time=7.63 ms (DUP!)
64 bytes from 192.168.100.109: icmp_seq=5 ttl=64 time=8.93 ms (DUP!)
64 bytes from 192.168.100.109: icmp_seq=6 ttl=64 time=4.49 ms
64 bytes from 192.168.100.109: icmp_seq=7 ttl=64 time=4.27 ms
64 bytes from 192.168.100.109: icmp_seq=7 ttl=64 time=5.01 ms (DUP!)
64 bytes from 192.168.100.109: icmp_seq=8 ttl=64 time=3.69 ms
```

Aperte “[Ctrl]+[C]” para parar o ping.

Pode acontecer que seu Raspberry se inicie sem conexão de rede, e sem conexão não é possível prosseguirmos com a conexão SSH. Para resolver tal problema escreva o seguinte texto na linha de comando:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

Isso abrirá uma tela com um texto muito similar ao abaixo:

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}

network:
  ethernets:
    eth0:
      dhcp4: true
      optional: true
  version: 2
  wifis:
    wlp3s0:
      optional: true
      access-points:
        "NOME DA REDE WIFI":
          password: "SENHA DA REDE WIFI"
          dhcp4: true
```

Edite o seu “**access-points:**” com o seu usuário e senha da sua rede wifi. Caso a parte de “**ethernets:**” não esteja presente, escreva-a conforme acima, respeitando os espaçamentos, isso permitirá a conexão via cabo além da rede sem fio. Após editado pressione **[CTRL]+[S]** para salvar e **[CTRL]+[X]** para sair da pasta, e em seguida escreva os seguintes comandos:

**sudo netplan apply + [ENTER]** (para aplicar as mudanças feitas)

**sudo netplan --debug apply + [ENTER]**

**sudo reboot” + [ENTER]** (para reiniciar o sistema)

Assim que ele reiniciar, você deve estar conectado à rede wifi escolhida.

OBS: Isso também é feito para mudanças de uma rede para outra.

Link para ajuda: <https://ubuntu.com/tutorials/how-to-install-ubuntu-on-your-raspberry-pi#1-overview>

### **13. CONEXÃO WIFI UFU**

- Endereço IP - Endereço IPv4 da máquina visto pela camada na rede.
- DHCP - Protocolo de distribuição de IPs em uma rede. Toda rede tem um servidor DHCP ativo, geralmente é o roteador/modem.

No caso da internet UFU ou em uma residência que já possui um servidor DHCP, ter 2 servidores DHCP pode gerar conflitos de rede, pois um servidor não sabe quem recebeu endereço pelo outro servidor. Há formas de contornar, deixando faixas de valores para cada servidor DHCP, mas principalmente no caso da UFU, não é possível.

Todos os roteadores têm uma opção de “bridge” ou de encaminhamento DHCP. Nesse modo, o aparelho não distribui IP para a rede e toda requisição que ele receber, ele encaminha para o servidor de fato que faz a distribuição. Para fazer isso ser feito na internet UFU, siga os seguintes passos:

- É necessário entrar no roteador com as configurações de fábrica via cabo conectado diretamente no computador na porta de LAN. É recomendado dar um reset no aparelho principalmente se não souber a senha, e precisa-se entrar no menu do aparelho e não no menu da operadora (como GVT, CTBC etc.). Por conta disso, procure na internet como efetuar o reset no modelo do aparelho que vira a ser utilizado e o IP padrão e o nome/senha padrão do aparelho na Internet.
- Com acesso ao aparelho, procure nas configurações de LAN as opções referentes ao DHCP e encontre a opção ou guia referente a deixar o aparelho em modo Bridge. Não esqueça de configurar o nome e senha da WiFi antes de sair das configurações.

- Uma observação importante é que ao invés de ligar o roteador no computador, ligue ele na rede. Ele deverá pegar IP automático e aparelhos deverão conseguir obter IP automático na rede Wi-Fi.

Link para ajuda: <https://www.youtube.com/watch?v=6ReXrWHrun0>

## 14.CONEXÃO SSH

Para uma conexão SSH entre sua máquina e a sua Raspberry PI ser feita escreva “ssh [USUÁRIO]@[IP]” com seu usuário e IP no terminal de sua máquina virtual. Digite “yes” e por fim coloque sua senha e está feito.

Se seu protocolo de internet for IPv4, deve encontrar algo como:

192.168.10.150

Se for o caso basta escrever “[USUÁRIO]@192.168.10.150”, e autorizar a conexão. Mas se seu protocolo for a versão IPv6, ele será apresentado a você no seguinte formato:

192.168.10.150 3002:0bd6:0000:0000:0000:ee00:0033:6778

Nessa situação escreva “[USUÁRIO]@ 3002:0bd6:0000:0000:0000:ee00:0033:6778”, e prosseguir com a autorização da conexão.

Figura 106 – Conexão SSH entre Raspberry e Máquina Virtual.

```
natanael@natanael-VirtualBox:~$ ssh pi@192.168.100.109
The authenticity of host '192.168.100.109 (192.168.100.109)' can't be established.
ED25519 key fingerprint is SHA256:5wwe4GAyrbtwFv3lR8eS0keqzCYph+7ahLTBGpLRi4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.109' (ED25519) to the list of known hosts.
pi@192.168.100.109's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-1069-raspi armv7l)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Tue Jan  3 10:25:55 -03 2023

 System load:  3.28           Temperature:          51.5 C
 Usage of /:   30.4% of 6.97GB  Processes:             146
 Memory usage: 21%           Users logged in:        1
 Swap usage:   0%            IPv4 address for wlan0: 192.168.100.109
```

Pronto. Agora você está ligado via rede à sua raspberry. Os comandos digitados aqui serão executados nela e assim não se faz mais necessário o uso de trabalhar com mouse, teclado e monitor, apenas com a máquina virtual.

OBS: PARA DESLIGAR A MÁQUINA ESCREVA “shutdown now”, ASSIM ELA SE DESLIGARÁ INSTANANEAMENTE.

OBS: PARA REINICIAR A MÁQUINA ESCREVA “sudo reboot”, ASSIM ELA SE REINICIARÁ INSTANANEAMENTE.

**15.“WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!”**

Essa parte do guia é voltada para aqueles que receberam uma mensagem de aviso como a mostrada abaixo em sua tentativa de conexão:

Figura 107 – Erro “WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!”.

Para resolver tal problema execute o comando apresentado no próprio aviso:

Figura 108 – Solução do erro “WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!”.

```
remove with:  
ssh-keygen -f "/home/natanael/.ssh/known_hosts" -R "192.168.100.109"
```

Apenas copie, cole e dê um **[ENTER]** no próprio terminal que o problema será corrigido. O próprio sistema removerá o arquivo que está provocando o erro, desse modo é só tentar efetuar a conexão novamente.

16. INSTALANDO ROS2 HUMBLE

Se a conexão SSH tiver sido feita, essa etapa é bem simples. Da mesma forma que o ROS2 Humble Hawksbill foi instalado em sua máquina virtual, ele deve ser instalado em seu Raspberry PI. Se atente aos detalhes e a todos os passos.

Certifique-se que o Rasp esteja conectado à uma fonte de 3 amperes, porque essa etapa exigirá um pouco de tempo e de poder de máquina de seu microcomputador.

Outro ponto importante é verificar a versão mais adequada de sistema operacional para aquela versão do ROS específica. Nesse guia usou-se a versão **Ubuntu Jammy (22.04) - arm64** para o ROS2 Humble Hawksbill. Olhe o pesquise por “**ROS 2 Releases and Target Platforms**” [<https://www.ros.org/reps/rep-2000.html>] para uma verificação antes de efetuar a instalação.

Figura 109 – Site de compatibilidade entre ROS2 e sistemas operacionais.

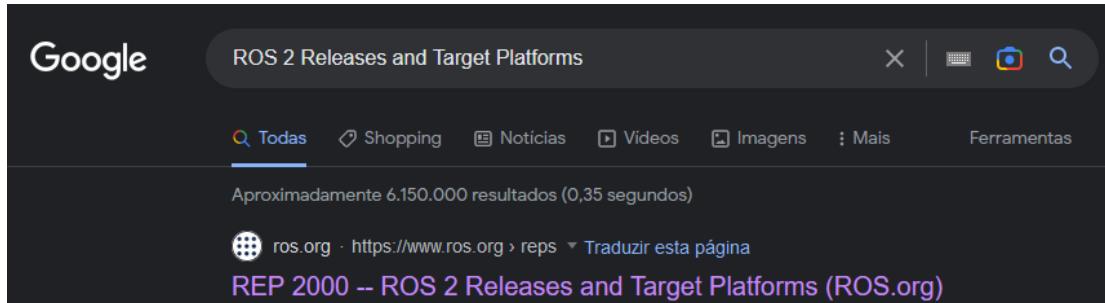


Figura 110 – Versões de Sistemas Operacionais compatíveis com ROS2 Humble Hawksbill.

#### Humble Hawksbill (May 2022 - May 2027)

Targeted platforms:

Architecture	Ubuntu Jammy (22.04)	Windows 10 (VS2019)	RHEL 8	Ubuntu Focal (20.04)	macOS	Debian Bullseye (11)	OpenEmbedded / Yocto Project
amd64	Tier 1 [d][a][s]	Tier 1 [a][s]	Tier 2 [d][a][s]	Tier 3 [s]	Tier 3 [s]	Tier 3 [s]	Tier 3 [s]
arm64	Tier 1 [d][a][s]			Tier 3 [s]		Tier 3 [s]	Tier 3 [s]
arm32	Tier 3 [s]			Tier 3 [s]		Tier 3 [s]	Tier 3 [s]

The following indicators show what delivery mechanisms are available for each platform.

" [d] " Distribution-specific (Debian, RPM, etc.) packages will be provided for this platform for packages submitted to the rosdistro.

" [a] " Binary releases are provided as a single archive per platform containing all packages in the Humble ROS 2 repos file [\[11\]](#).

" [s] " Compilation from source.

Como pode ver a versão “arm” usada no Raspberry PI 3B+ quando de 32bits deve ser compilada da diretamente da fonte. Já a de 64bits pode ser instalada por distribuições e pacotes, o que é ensinado nesse tutorial. Então se quiser usar outro sistema e/ou outra versão pense com calma em qual utilizar.

Link para ajuda: <https://ubuntu.com/tutorials/how-to-install-ubuntu-on-your-raspberry-pi#1-overview>

## 17.“Could not get lock /var/lib/dpkg/lock”

Esse erro aparece porque o sistema já está efetuando algum processo. Antes de tudo cheque se outro programa já está sendo instalado ou atualizado, por isso escreva:

```
ps aux | grep -i apt
```

Caso veja algo como:

```
abhishek@nuc:~$ ps aux | grep -i apt
root    1464  0.0  0.0  4624  772 ?        Ss   19:08  0:00 /bin/sh /usr/lib/apt/apt.systemd.daily
update
root    1484  0.0  0.0  4624 1676 ?        S   19:08  0:00 /bin/sh /usr/lib/apt/apt.systemd.daily
lock_is_held update
_apt    2836  0.8  0.1 96912 9432 ?        S   19:09  0:03 /usr/lib/apt/methods/http
abhishek 6172  0.0  0.0 21532 1152 pts/1  S+  19:16  0:00 grep --color=auto -i apt
```

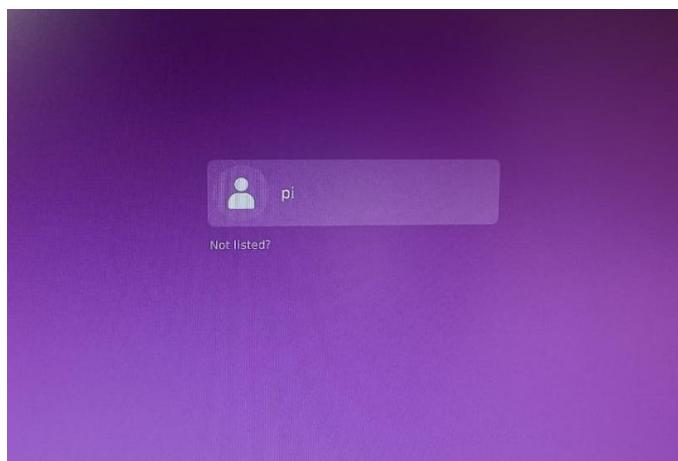
Não se preocupe, o sistema está ocupado trabalhando em algum processo que já está sendo executado. Neste caso o “`apt.systemd.daily update`” é um programa executado em segundo plano que checa por atualizações automaticamente quando o sistema se inicia, presente em versões do Ubuntu 18.04 e posteriores.

Você pode checar se existem atualizações autônomas nos arquivos do Ubuntu na pasta “`/etc/apt/apt.conf.d/20auto-upgrades`”. Se por acaso encontrar o “`apt.systemd.daily update`” usando o processo “`apt`” tudo que será necessário fazer é esperar alguns minutos até as atualizações automáticas se terminarem. Assim feito, você poderá continuar suas instalações normalmente.

## 18.INTERFACE GRÁFICA

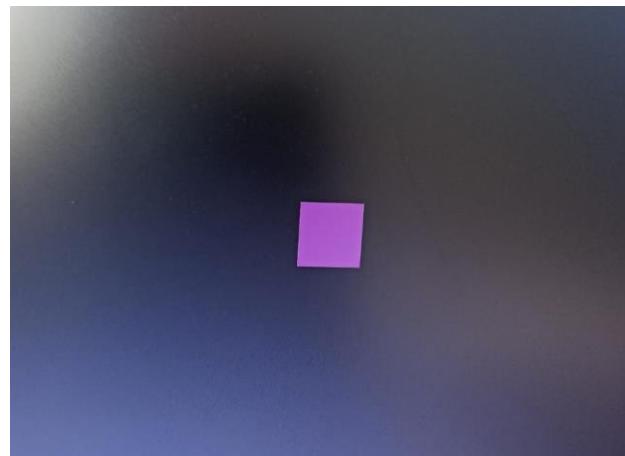
Caso opte por instalar a interface gráfica do ROS no seu Rasp, ao iniciá-lo uma nova tela, como a apresentada abaixo, deve aparecer. Antes de qualquer coisa, entre com seu usuário e senha pré-definidos na instalação do sistema no cartão SD.

Figura 111 – Primeira tela do SciCoBot.



Caso insira os dados corretamente, uma tela de carregamento, como a apresentada abaixo deve aparecer:

Figura 112 – Tela do SciCoBot carregando.



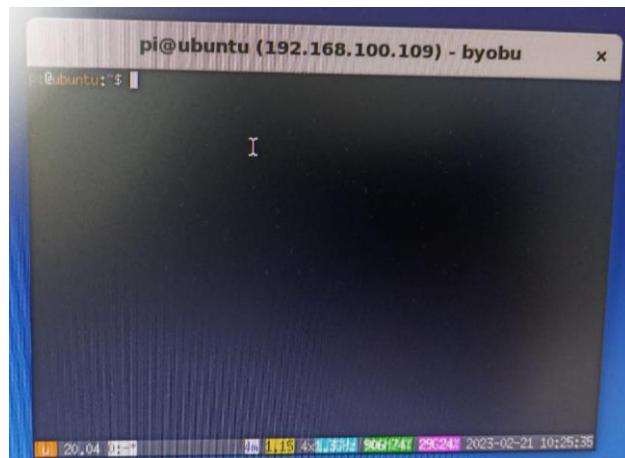
Após o término do carregamento, uma nova tela aparecerá.

Figura 113 – Tela inicial SciCoBot.



Para abrir o terminal pressione **[Ctrl]+[Alt]+[T]**. Esse comando abrirá um terminal automaticamente no sistema operacional Linux. Com o terminal aberto você pode fazer todos os passos ensinados anteriormente, como desligar a máquina, reiniciá-la, pingar ip, realizar a conexão SSH com outro computador etc.

Figura 114 – Terminal SciCoBot.



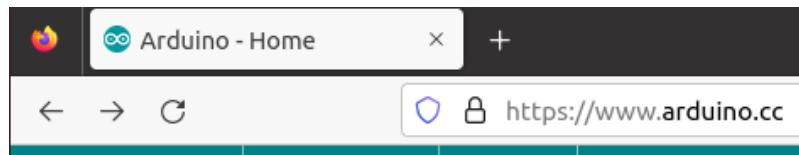
## 19. INSTALAÇÃO DA IDE ARDUINO

Para a configuração do Arduino Due é necessário a instalação da IDE Arduino em seu computador. Aqui está exemplificado essa etapa tanto para a máquina virtual criada, tanto para Windows para aqueles que tem mais familiaridade e preferem trabalhar com este sistema operacional:

- **MÁQUINA VIRTUAL UBUNTU 20.04**

Antes de qualquer coisa, abra seu navegador e pesquise por Arduino no Google, ou simplesmente “[arduino.cc](https://www.arduino.cc)” na barra de pesquisa.

Figura 115 – Página oficial Arduino.



Clique na opção “**SOFTWARE**” e procure pela versão 1.8.19 nas disponibilizadas pela plataforma. Você verá algo semelhante ao apresentado na imagem abaixo. Escolha a versão “**ARM 64bits**” que é compatível com o sistema operacional tanto da Raspberry PI do SciCoBot e apenas 64bits para a máquina virtual.

Figura 116 – Opções de Download do Arduino versão 1.8.19.

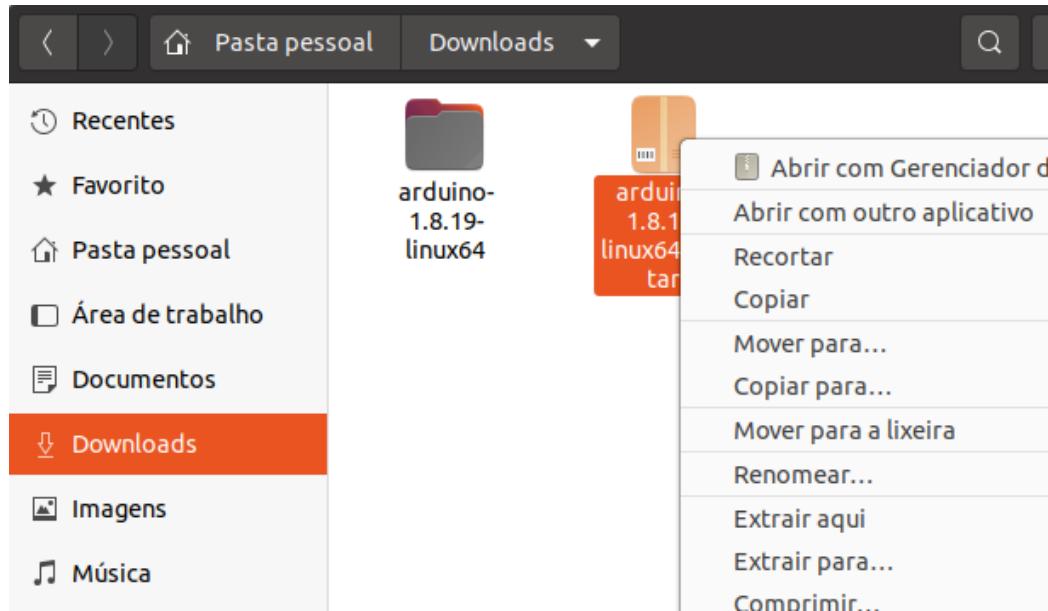
A screenshot of the Arduino IDE 1.8.19 download page. The page features a logo and the text "Arduino IDE 1.8.19". It describes the software as open-source and suitable for any Arduino board. Below this, there's a link to the "Getting Started" page. On the right side, there's a "DOWNLOAD OPTIONS" section with links for Windows (ZIP file and app), Linux (32-bit and 64-bit), ARM 32-bit, ARM 64-bit, and Mac OS X. There are also links for "Release Notes" and "Checksums (sha512)".

Após finalizado o download, o arquivo compactado vai para a pasta destino de seu navegador. Abra-a e faça a descompactação.

Figura 117 – Download do arquivo zip da IDE Arduino versão 1.8.19 para Linux.



Figura 118 – Arquivo da IDE Arduino depois de extraído.



Após descompactar o arquivo, recomenda-se recortar a pasta e colar na sessão “Pasta pessoal”.

Figura 119 – Arquivo da IDE Arduino na pasta Downloads.

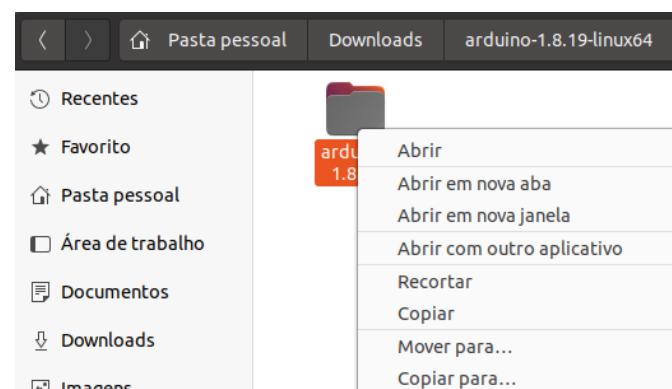


Figura 120 – Arquivo da IDE Arduino na Pasta pessoal.

Em seguida, abra a pasta e clique com o botão direito do mouse de modo que seja possível abrir um terminal ali dentro.

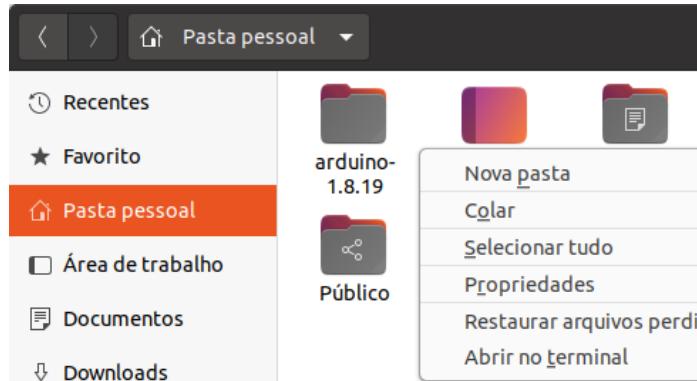
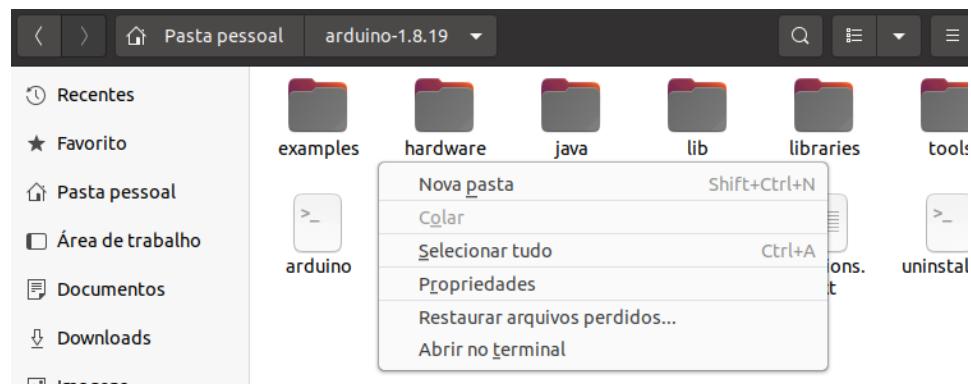


Figura 121 – Pastas internas do arquivo do Arduino IDE.



Com o terminal aberto dentro da pasta escreva “`ls`” para ter acesso a todo o conteúdo da pasta da IDE Arduino. Caso exista um tópico chamado “`install.sh`” prossiga com o guia, caso não, repita o processo desde o começo.

Em seguida, escreva o seguinte comando:

```
sudo sh ./install.sh
```

Por ser um comando com acesso de administrador, uma senha pode ser pedida. Aguarde alguns segundos para a instalação e logo em seguida estará tudo pronto, basta procurar por “Arduino” que sua IDE estará instalada.

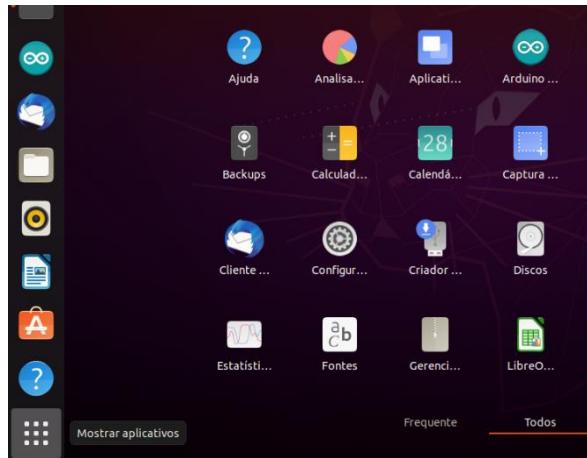
Figura 122 – Instalação da IDE via terminal.

```
natanael@natanael-VirtualBox:~/arduino-1.8.19$ ls
arduino      examples   java      revisions.txt  uninstall.sh
arduino-builder  hardware  lib      tools
arduino-linux-setup.sh  install.sh  libraries  tools-builder
natanael@natanael-VirtualBox:~/arduino-1.8.19$ sudo sh ./install.sh
[sudo] senha para natanael:
sudo: sh./install.sh: comando não encontrado
natanael@natanael-VirtualBox:~/arduino-1.8.19$ sudo sh ./install.sh
Adding desktop shortcut, menu item and file associations for Arduino IDE...

done!
natanael@natanael-VirtualBox:~/arduino-1.8.19$
```

Na barra lateral de sua máquina virtual clique em “**Mostrar aplicativos**” e em seguida em “**Arduino IDE**”.

Figura 123 – IDE do Arduino localizada na aba de aplicativos.



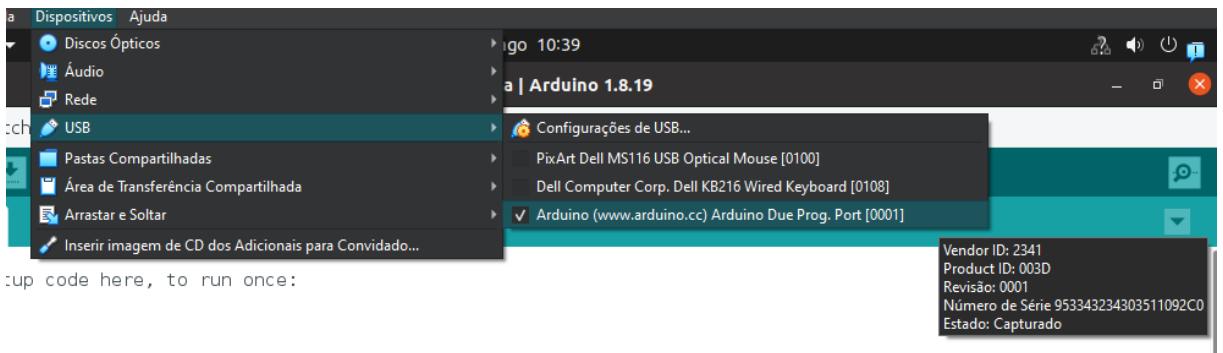
Em alguns instantes sua IDE Arduino se abrirá e você poderá seguir com a identificação da placa e da porta USB na máquina virtual.

Figura 124 – Tela inicial da Arduino IDE na máquina virtual.



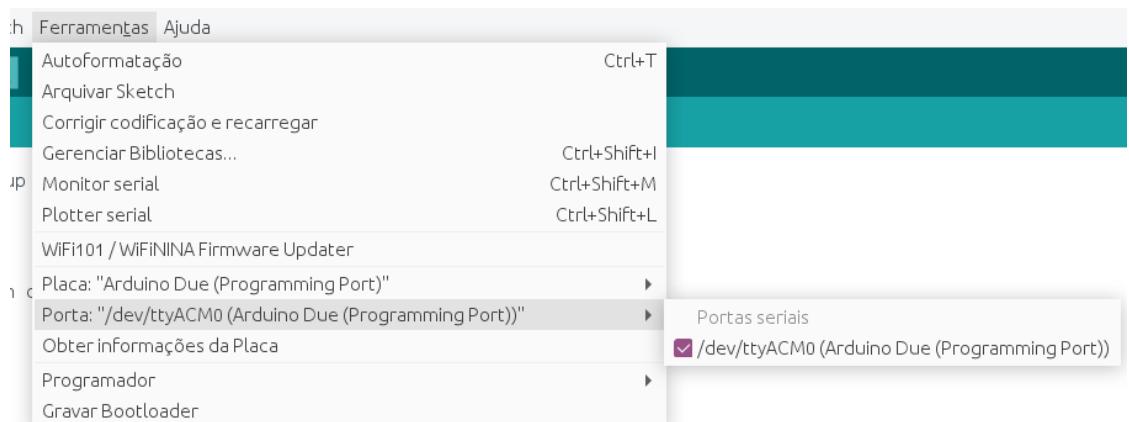
Na barra superior do Virtual Box, se o CD dos adicionais para convidado estiver sido empregado, selecione: “**Dispositivos**”, “**USB**” e selecione a unidade USB conectada relacionada ao Arduino.

Figura 125 – Habilitando portas USB na máquina virtual.



Na IDE Arduino, vá em “**Ferramentas**”, “**Porta**” e escolha a porta indicada pelo Virtual Box.

Figura 126 – Selecionando portas na Arduino IDE.

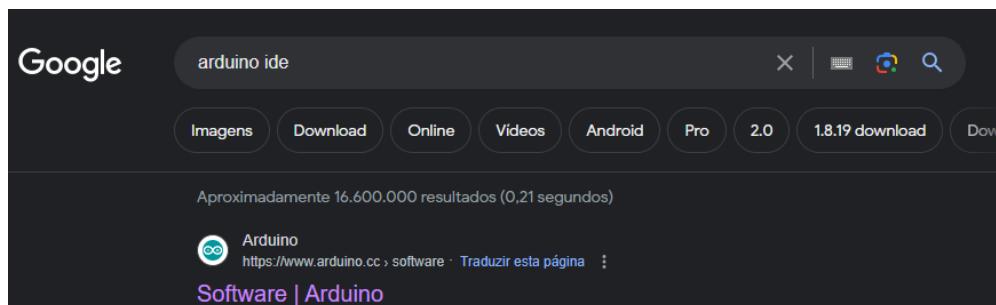


Pronto, sua IDE já pode ser utilizada.

- Windows

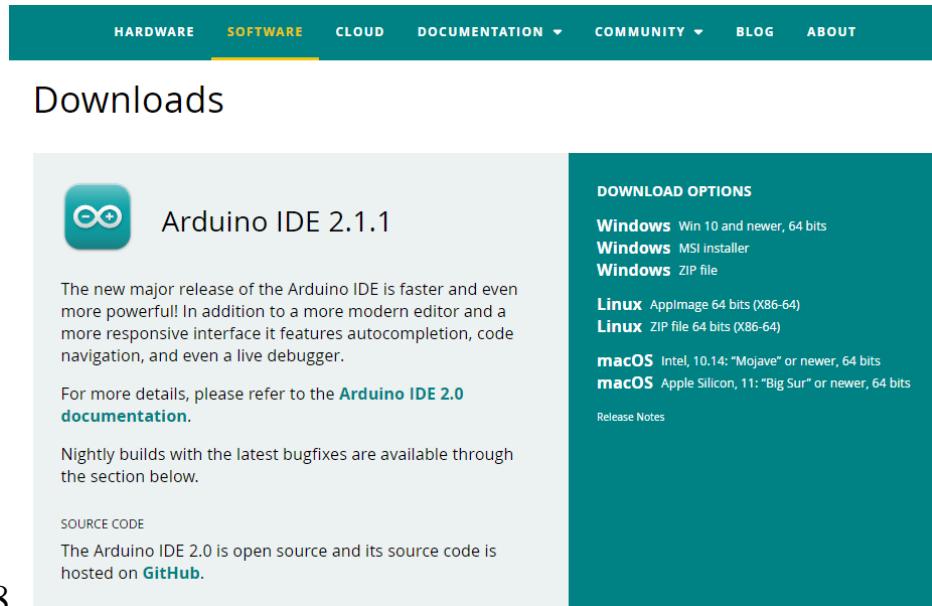
Antes de qualquer coisa, pesquise por “**Arduino IDE**” no Google, ou qualquer outro site de pesquisa. Clique no link indicado na figura abaixo (geralmente é o primeiro a aparecer na pesquisa):

Figura 127 – Pesquisa por “**arduino ide**” no Google.



O link direcionará você direto à página de downloads da IDE do Arduino. Procure a versão mais atual da IDE para a sua plataforma e clique na opção de download.

Figura 128 – Página de downloads da Arduino.



8

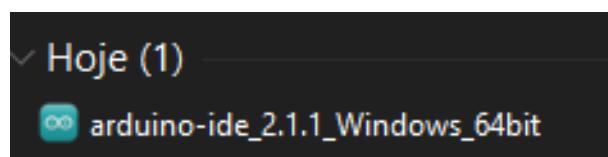
Logo em seguida, você será direcionado para uma página onde é possível contribuir diretamente com o desenvolvimento da Arduino. Caso seja de sua vontade, faça a doação, caso não queira clique em “**JUST DOWNLOAD**” e baixe a IDE normalmente.

Figura 129 – Tela de doações da Arduino.



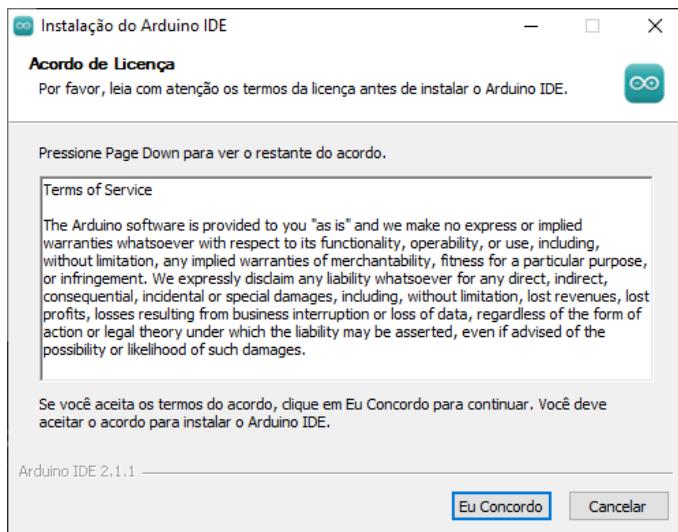
Com o download concluído, vá até a pasta configurada pelo seu navegador e execute o arquivo baixado.

Figura 130 – Arquivo executável da Arduino IDE baixado.



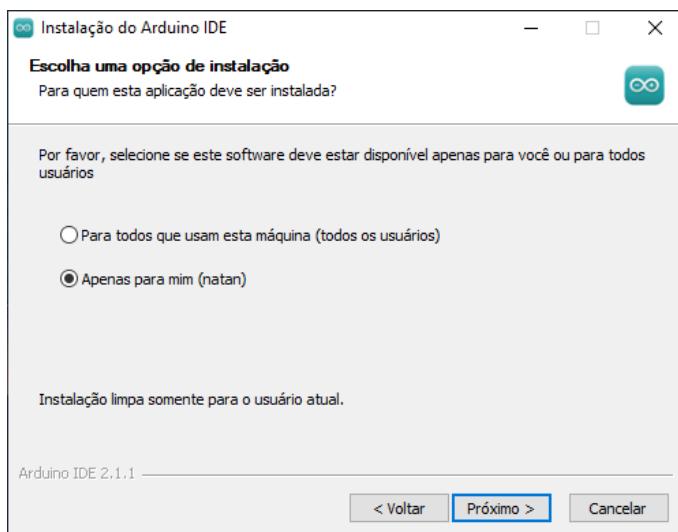
O software apresentará os termos e licenças, concorde para prosseguir com a instalação normalmente.

Figura 131 – Instalação da IDE do Arduino em Windows, parte 1.



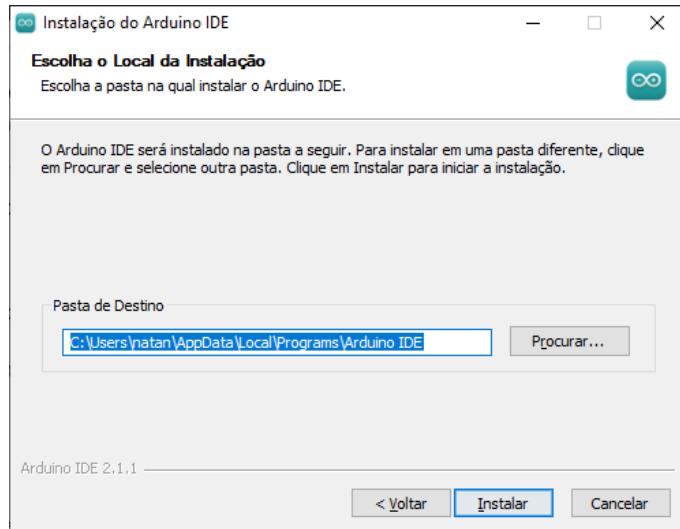
Escolha se todos os usuários da máquina terão acesso a IDE ou se apenas o usuário que está sendo utilizado terá esse benefício.

Figura 132 – Instalação da IDE do Arduino em Windows, parte 2.



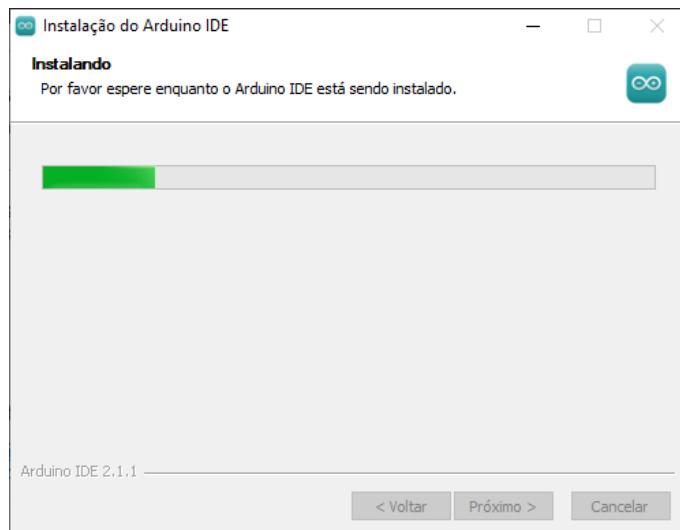
Configure o caminho de instalação dos arquivos da IDE em seu computador e confirme a instalação.

Figura 133 – Instalação da IDE do Arduino em Windows, parte 3.



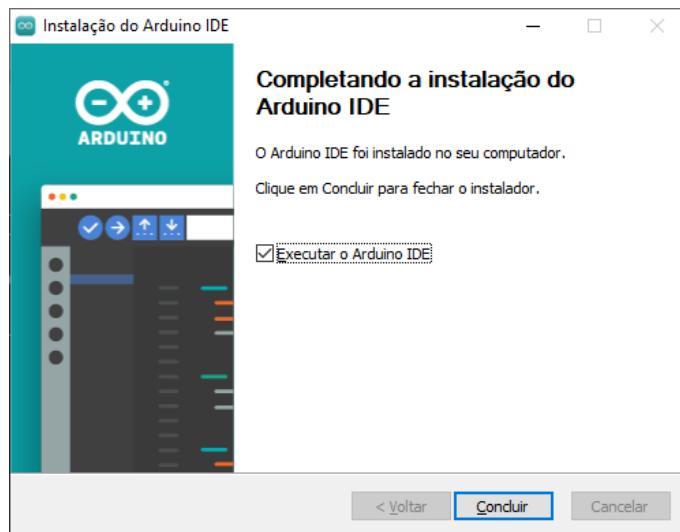
Se tudo for feito como descrito, a instalação se iniciará. Talvez demore alguns minutos para que ela se finalize, isso dependerá da velocidade da máquina utilizada.

Figura 134 – Instalação da IDE do Arduino em Windows, parte 4.



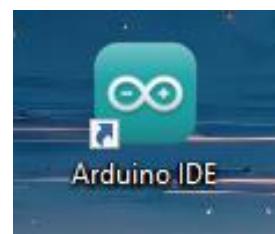
Por fim, conclua a instalação e utilize a interface como desejar.

Figura 135 – Instalação da IDE do Arduino em Windows, parte final.



Um atalho do programa será adicionado à sua área de trabalho. Dê dois cliques com o botão esquerdo do mouse sobre ele.

Figura 136 – Atalho da Arduino IDE na área de trabalho Windows.



Após algum tempo de carregamento a IDE se abrirá e estará totalmente disponível para uso.

Figura 137 – Tela de iniciação da Arduino IDE.

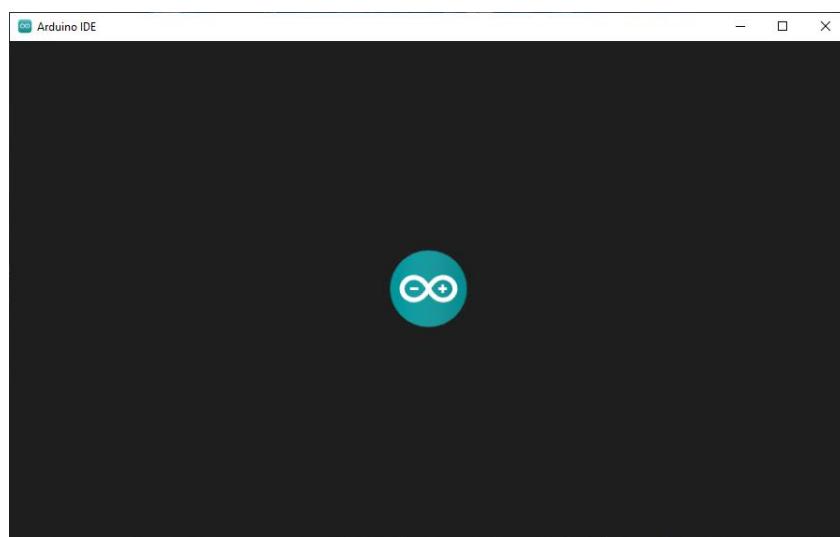
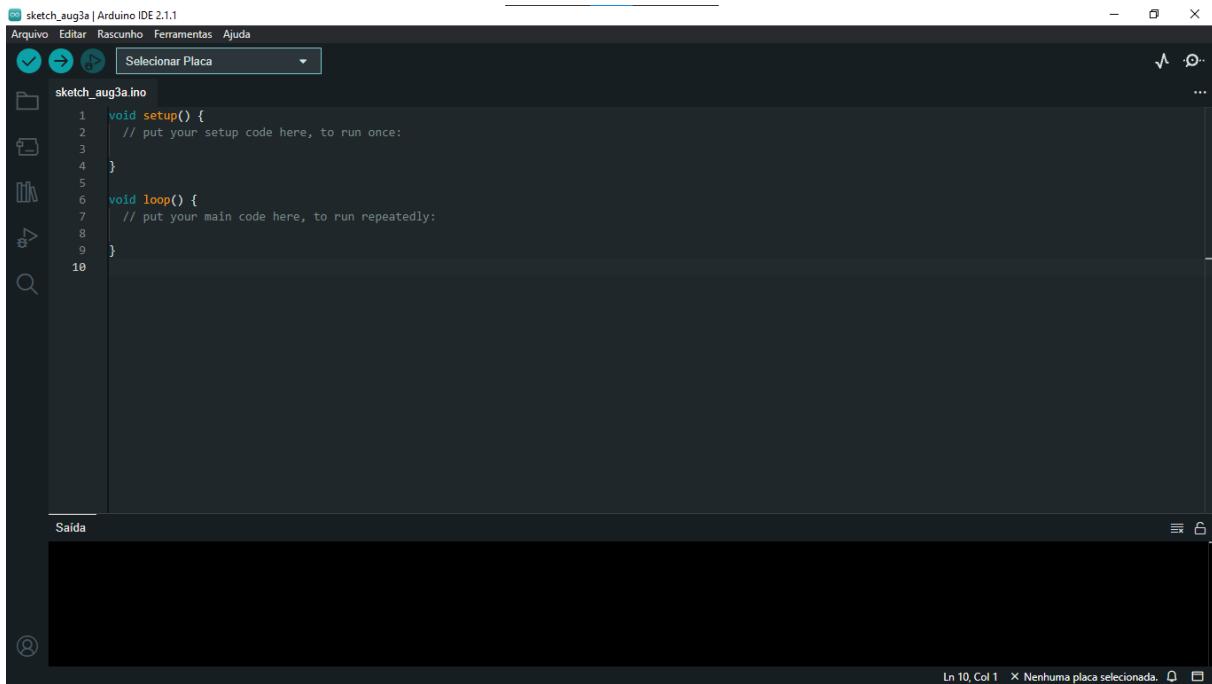


Figura 138 – Tela inicial da Arduino IDE no Windows.



## 20. INSTALAÇÃO DO VS CODE

O Visual Studio Code (VS Code) é um editor de código de código aberto desenvolvido pela Microsoft. Ele foi criado com Electron, ferramenta criada pelo GitHub que permite a criação de softwares Desktop com HTML, CSS e JavaScript. Esse programa é essencial para a produção de material voltado à essa pesquisa, dessa forma, sua instalação se torna crucial.

- Linux:

Em um novo terminal escreva:

```
sudo snap install --classic code
code --version
```

Com isso o VS Code será instalado automaticamente em sua máquina e assim será possível escrever e adaptar seus códigos diretamente no GitHub.

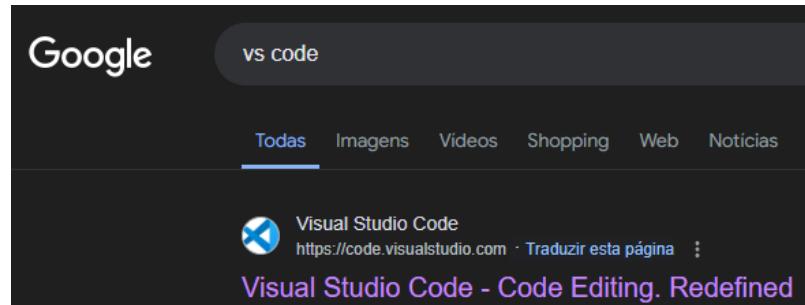
Figura 139 – Instalando VS Code via snap no terminal.

```
natanael@scicobot-virtualbox:~$ sudo snap install --classic code
[sudo] senha para natanael:
code 89de5a8d de Visual Studio Code (vscode) instalado
natanael@scicobot-virtualbox:~$ code --version
1.90.0
89de5a8d4d6205e5b11647eb6a74844ca23d2573
x64
natanael@scicobot-virtualbox:~$
```

- Windows:

Pesquise por “**VS Code**” no Google e vá para o link indicado:

Figura 140 – Pesquisa por “vs code” na plataforma Google.com.



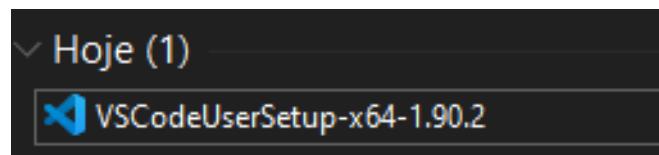
Na página inicial clique em “Download for Windows”:

Figura 141 – Tela inicial do site “Visual Studio Code”.



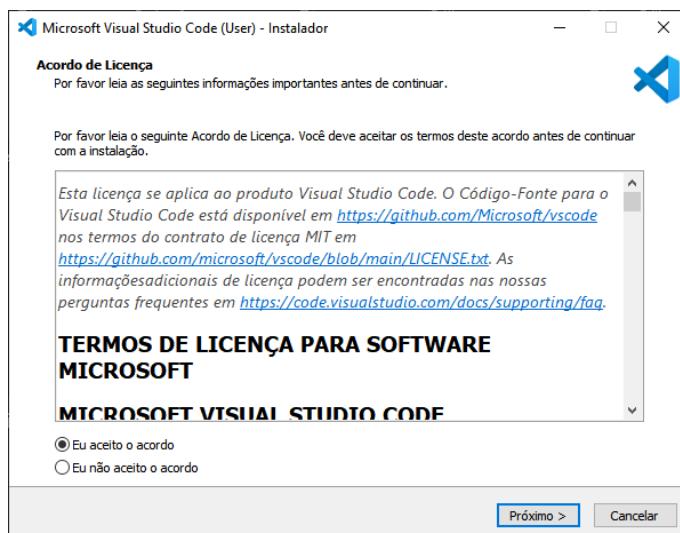
Dê dois cliques sobre o arquivo executável baixado e prossiga com a instalação:

Figura 142 – Arquivo executável baixado do programa.



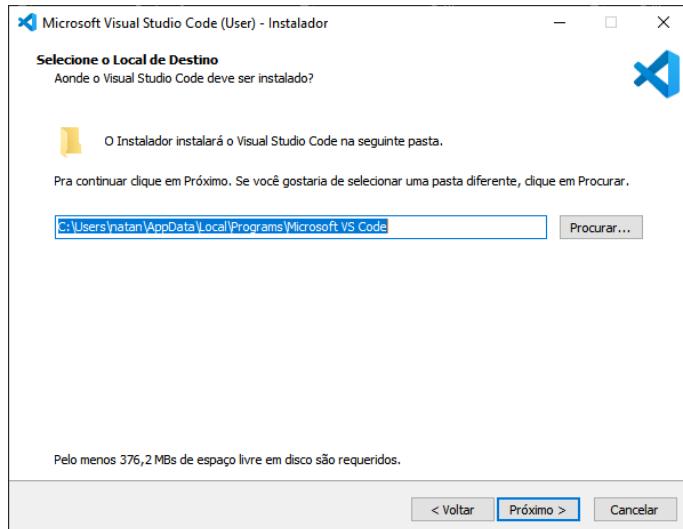
Aceite os termos de licença da Microsoft:

Figura 143 – Licenças do VS Code.



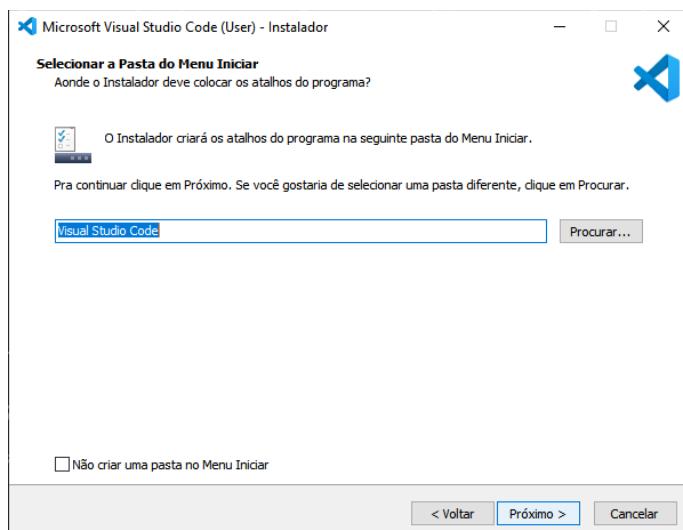
Defina o endereço de instalação dos arquivos:

Figura 144 – Local de destino de arquivos do VS Code.



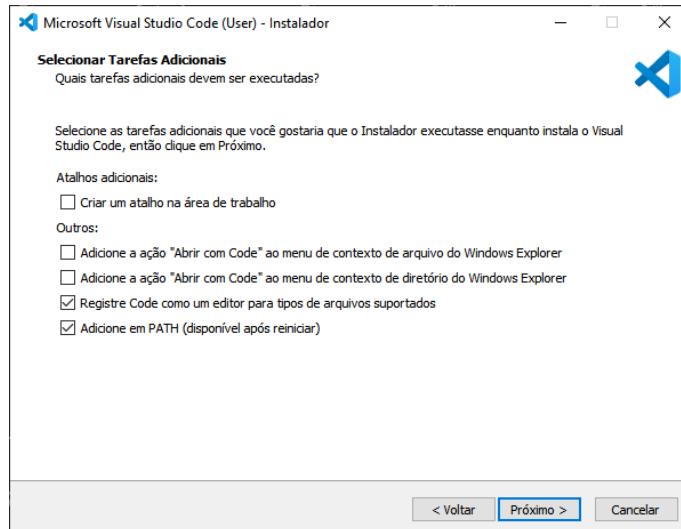
Defina onde, não é obrigatório, criar o atalho do programa:

Figura 145 – Pasta do menu iniciar do programa VS Code.



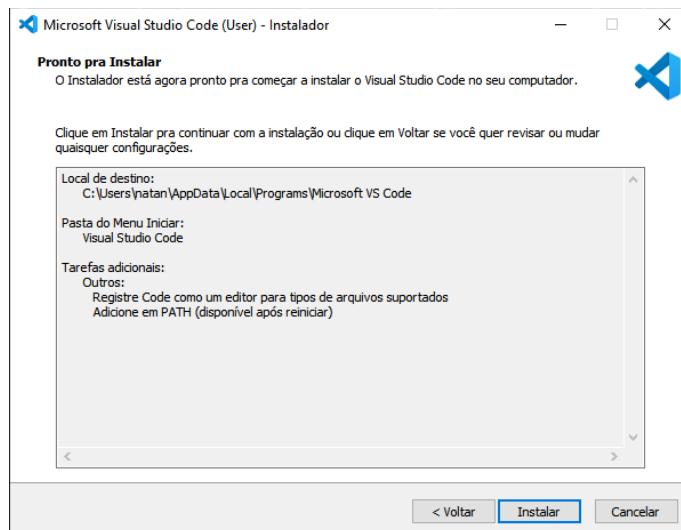
Escolha quais tarefas adicionais serão executadas enquanto o instalador instala o VS Code:

Figura 146 – Tarefas adicionais do Programa VS Code.



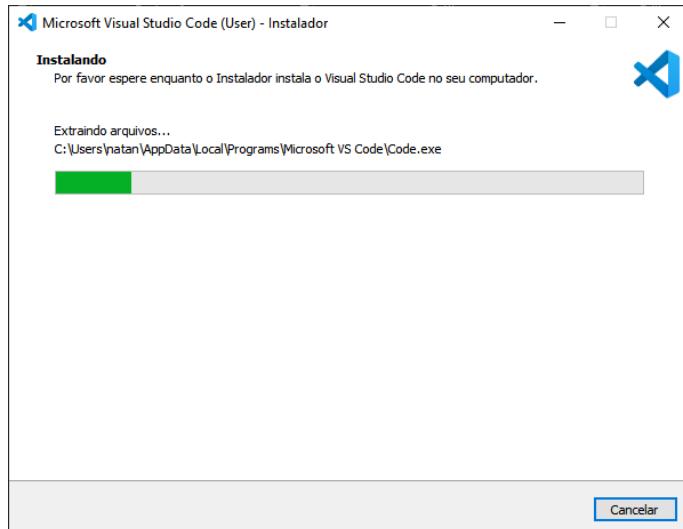
Confirme as opções definidas e prossiga com a instalação:

Figura 147 – Confirmação de configurações de instalação do programa.



Aguarde pacientemente:

Figura 148 – Instalação do VS Code.



Pronto. O VS Code está instalado em sua máquina Linux:

Figura 149 – Finalização da instalação do programa VS Code.

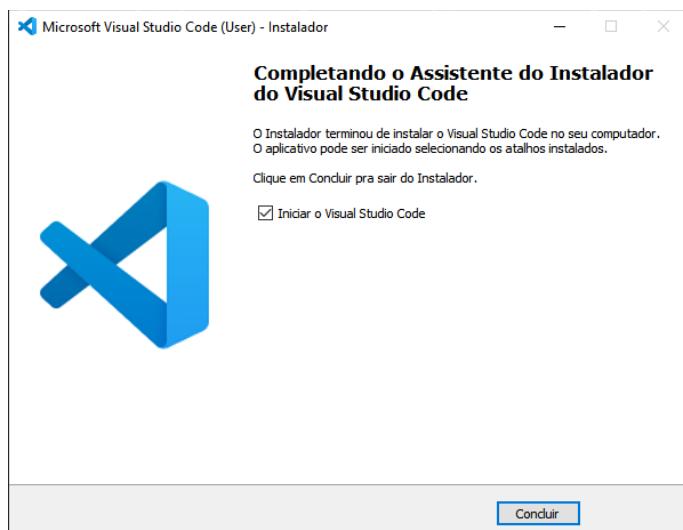
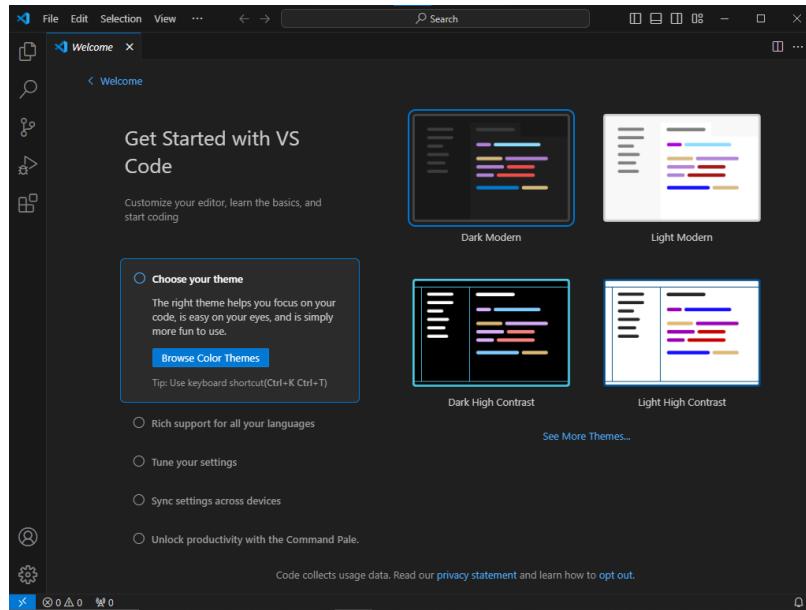


Figura 150 – Tela inicial do programa VS Code.



## 21.CRIAÇÃO DE PACOTES ROS 2

Para criar um pacote ROS 2 primeiro é necessário criar uma pasta onde os arquivos serão escritos. Para isso escreva:

```
mkdir -p [PASTA]/src
cd [PASTA]
colcon build --symlink-install
ls
```

Dessa forma, você deverá se deparar com quatro novas subpastas: `build`, `install`, `log` e `src`. Para criar um pacote entre na pasta `src` e escreva o comando seguinte, tendo em mente o nome do pacote e o tipo de compilação (C++ ou Python).

```
cd src
source /opt/ros/humble/setup.bash, e escolha entre
ros2 pkg create --build-type ament_cmake [NOME DO PACOTE], para C++
ros2 pkg create --build-type ament_python [NOME DO PACOTE], para Python
```

Para criar um código nesse pacote é necessário o programa VS Code. Basta abrir a pasta criada e dois arquivos poderão ser encontrados na pasta `src`: um `.txt` e um `.xml`.

Figura 151 – Inicialização de arquivo no VS Code.

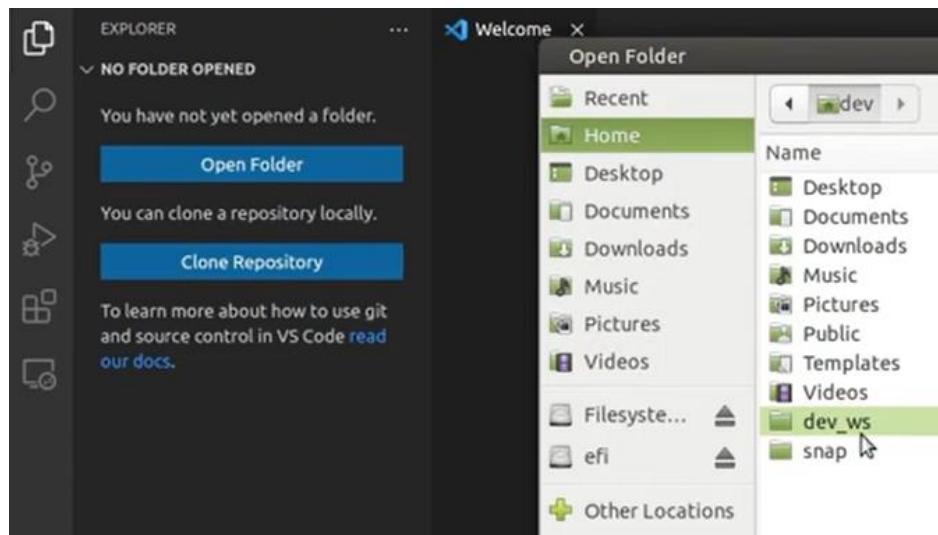
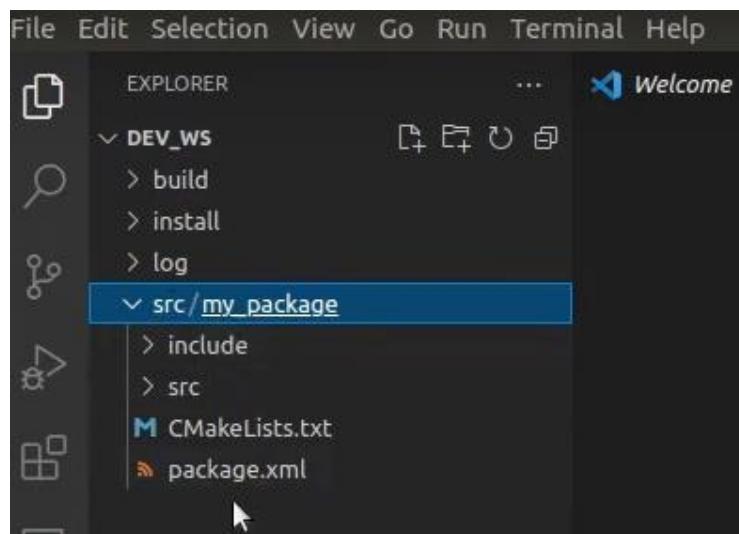


Figura 152 – Pasta criada com os comandos indicados acima.



Para exemplificar, aqui é exemplificado o tipo de pacote mais simples que você poderia criar, um talker-listener entre uma máquina Linux e um Raspberry Pi com Linux instalado, e para isso necessita-se de apenas dois arquivos de inicialização. Dentro do pacote crie uma pasta chamada `launch`, e dentro dela dois arquivos: `talker.launch.pi` e `listener.launch.pi` e suas funções serão inicializar os arquivos demo dos nós de comunicação apresentados como exemplo para testar a instalação do ROS 2. Vale a pena ressaltar que aqui foi trabalhado com a linguagem `C++`.

Figura 153 – Criação de pasta e de arquivo no VS Code.

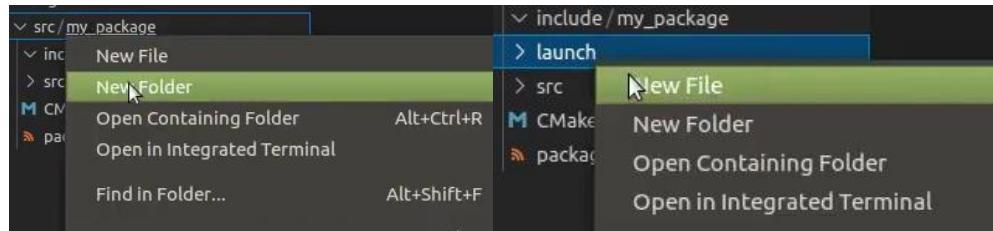


Figura 154 – Estrutura final com a criação de arquivos acima no VS Code.

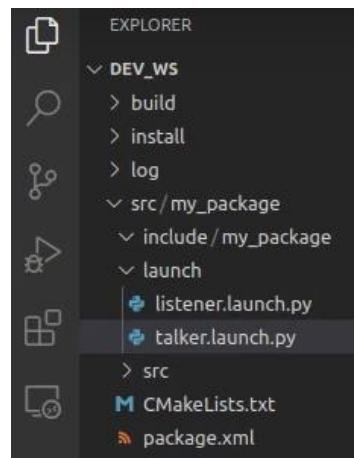


Figura 155 – Estrutura final com a criação de arquivos acima no Terminal.

```
maq@maq-VirtualBox:~$ ls
A                               Documentos      Música          teste_ws
Arduino                         Downloads       Público         Vídeos
arduino-1.8.19                  Imagens        ros2_ws
'Área de Trabalho'               microros_ws  scicobot_rasp
build_scicobot_arduino           Modelos       snap
maq@maq-VirtualBox:~$ cd teste_ws/
maq@maq-VirtualBox:~/teste_ws$ ls
build  install  log  src
maq@maq-VirtualBox:~/teste_ws$ cd src/
maq@maq-VirtualBox:~/teste_ws/src$ ls
my_package
maq@maq-VirtualBox:~/teste_ws/src$ cd my_package/
maq@maq-VirtualBox:~/teste_ws/src/my_package$ ls
CMakeLists.txt  include  launch  package.xml  README.md  src
maq@maq-VirtualBox:~/teste_ws/src/my_package$ cd launch/
maq@maq-VirtualBox:~/teste_ws/src/my_package/launch$ ls
listener.launch.py  talker.launch.py
maq@maq-VirtualBox:~/teste_ws/src/my_package/launch$
```

Dê dois cliques sobre o arquivo e ele será aberto no programa em formato de bloco de notas. Neles escreva os seguintes códigos:

Figura 156 – Código dos arquivos “talker.launch.pi” e “listener.launch.pi”.

```
Ξ talker.launch.py × Ξ listener.launch.py | Ξ talker.launch.py | Ξ listener.launch.py ×
launch > Ξ talker.launch.py
1   from launch import LaunchDescription
2   from launch_ros.actions import Node
3
4   def generate_launch_description():
5       return LaunchDescription([
6           Node(
7               package='demo_nodes_cpp',
8               executable= 'talker'
9           )
10      ])
launch > Ξ listener.launch.py
1   from launch import LaunchDescription
2   from launch_ros.actions import Node
3
4   def generate_launch_description():
5       return LaunchDescription([
6           Node(
7               package='demo_nodes_cpp',
8               executable= 'listener'
9           )
10      ])
```

Com os códigos escritos, edite o `CMakeLists.txt` para incluir o diretório `lunch` e o `package.xml` para incluir os pacotes no qual o pacote que queremos executar depende, nesse caso `demo_nodes`:

Figura 157 – Arquivo “CmakeLists.txt” alterado.

```
M CMakeLists.txt
 1 cmake_minimum_required(VERSION 3.5)
 2 project(my_package)
 3
 4 # Default to C99
 5 if(NOT CMAKE_C_STANDARD)
 6 | set(CMAKE_C_STANDARD 99)
 7 endif()
 8
 9 # Default to C++14
10 if(NOT CMAKE_CXX_STANDARD)
11 | set(CMAKE_CXX_STANDARD 14)
12 endif()
13
14 if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
15 | add_compile_options(-Wall -Wextra -Wpedantic)
16 endif()
17
18 # find dependencies
19 find_packageament_cmake REQUIRED)
20 # uncomment the following section in order to fill in
21 # further dependencies manually.
22 # find_package(<dependency> REQUIRED)
23
24 if(BUILD_TESTING)
25   find_packageament_lint_auto REQUIRED)
26   # the following line skips the linter which checks for copyrights
27   # uncomment the line when a copyright and license is not present in all source files
28   #set(ament_cmake_copyright_FOUND TRUE)
29   # the following line skips cpplint (only works in a git repo)
30   # uncomment the line when this package is not in a git repo
31   #set(ament_cmake_cpplint_FOUND TRUE)
32   ament_lint_auto_find_test_dependencies()
33 endif()
34
35 install(DIRECTORY launch
36 DESTINATION share/${PROJECT_NAME})
37 )
38
39 ament_package()
```

Figura 158 – Arquivo “package.xml” alterado.

```

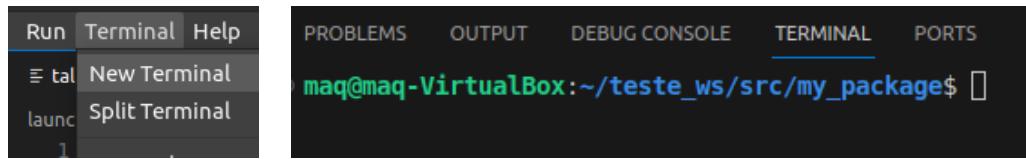
  package.xml
1   <?xml version="1.0"?>
2   <?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematypens="http://www.w3.org/2001/XMLSchema"?>
3   <package format="3">
4     <name>my_package</name>
5     <version>0.0.0</version>
6     <description>PACOTE TESTE SCICOBOT</description>
7     <maintainer email="maq@todo.todo">maq</maintainer>
8     <license>TODO: License declaration</license>
9
10    <buildtool_depend>ament_cmake</buildtool_depend>
11
12    <test_depend>ament_lint_auto</test_depend>
13    <test_depend>ament_lint_common</test_depend>
14
15    <exec_depend>demo_nodes_cpp</exec_depend>
16    <exec_depend>demo_nodes_py</exec_depend>
17
18    <export>
19      <build_type>ament_cmake</build_type>
20    </export>
21  </package>

```

Com tudo isso pronto e atualizado, basta voltar a um terminal, dentro da primeira pasta criada e escrever o comando:

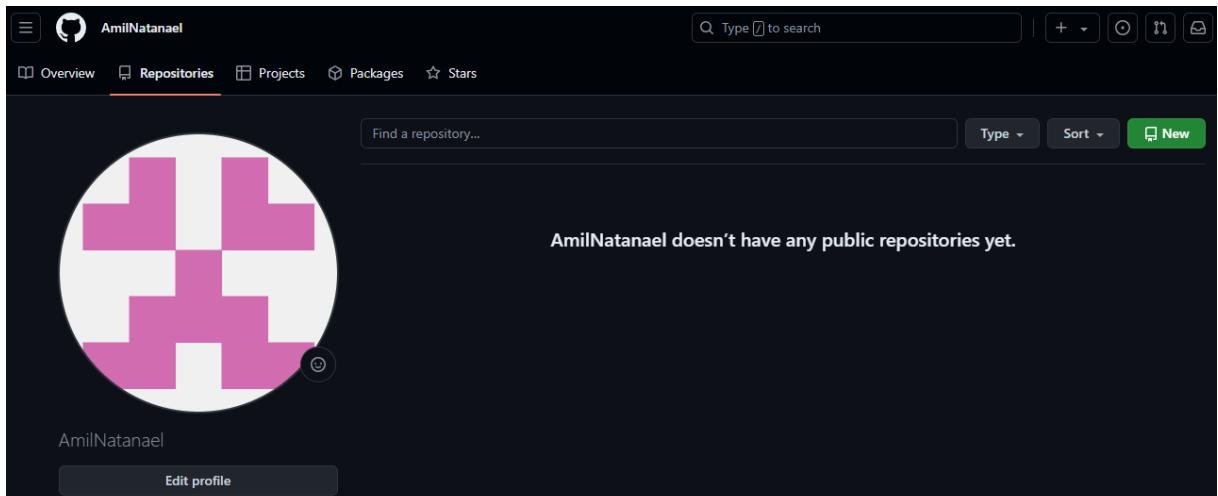
`colcon build --symlink-install`

Figura 159 – Abrir terminal via VS Code.



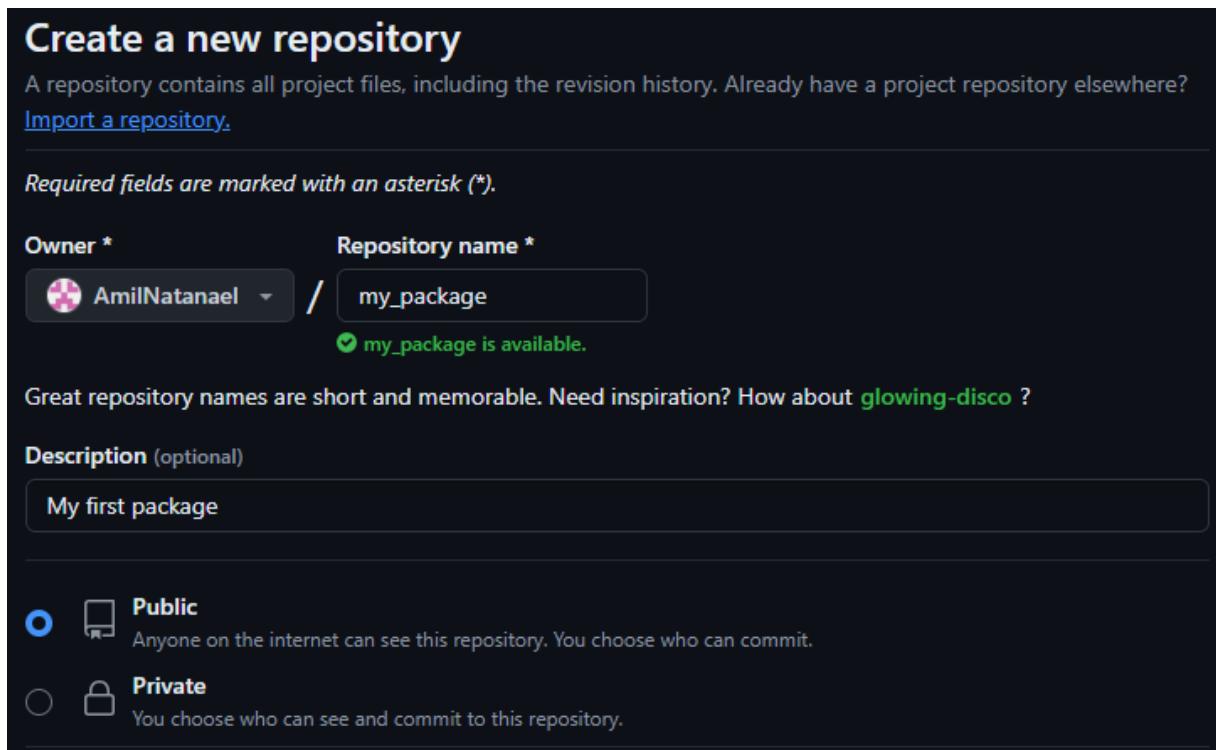
Para uma comunicação entre máquina e Raspberry, o GitHub se faz crucial. Por meio dele todo código e arquivos são salvos por meio de um repositório que posteriormente podem ser baixados em outra máquina. Para isso, crie uma conta na plataforma e crie um repositório:

Figura 160 – Tela inicial do perfil do GitHub.



Defina um nome e uma descrição para o pacote. Como a ideia aqui é um projeto open source, indico manter o projeto como público:

Figura 161 – Página de criação de configuração de um novo repositório Git.



Em seguida, abra o [terminal](#) do VS Code e escreva os comandos indicados pelo GitHub, aconselho upar toda a pasta criada, aqui chamada de [teste\\_ws](#), e não apenas a [src](#):

```
echo "# [NOME DO REPOSITÓRIO]" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin [LINK DO REPOSITÓRIO]
git push -u origin main
```

Figura 162 – Comandos utilizados para adição de arquivos criados no repositório online.

```
...or create a new repository on the command line
echo "# my_package" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/AmilNatanael/my_package.git
git push -u origin main
```

Durante esse processo, um link direcionado à uma autenticação da conta da plataforma aparecerá, por isso recomenda-se o uso do VS Code, por ele basta fazer login em uma aba de

navegador, enquanto por terminal Linux o sistema pode travar ou não entender muito bem o comando.

**OBS:** Se já existir um repositório que queira incluir o código existe a opção de criar uma **Branch**, como se fosse uma segunda camada atualizada (código anterior ainda fica salvo, mas não é mais o principal). Basta seguir os comandos do GitHub:

```
git remote add origin [LINK DO REPOSITÓRIO]
git branch -M main
git push -u origin main
```

Figura 163 – Comandos utilizados para criação de uma Branch no repositório online.

### ...or push an existing repository from the command line

```
git remote add origin https://github.com/AmilNatanael/my_package.git
git branch -M main
git push -u origin main
```

Com tudo isso já feito basta criar uma pasta, do mesmo modo como foi descrito anteriormente e dar os seguintes comandos:

```
git clone [LINK DO REPOSITÓRIO]
colcon build --symlink-install
```

Dessa forma, todos os arquivos online são baixados e instalados dentro da pasta escolhida. Para executar o exemplo siga as instruções abaixo, dentro das respectivas pastas criadas:

[NA MÁQUINA]

```
cd [PASTA]
source /opt/ros/humble/setup.bash
source istall/setup.bash
ros2 launch my_package talker.launch.py
```

[NO RASPBERRY]

```
cd [PASTA]
source /opt/ros/humble/setup.bash
source istall/setup.bash
ros2 launch my_package listener.launch.py
```

**OBS:** o inverso também pode ser feito, o rasp fala e a máquina escuta.

## 22. CÓDIGOS SERIAL\_MOTOR NO ARDUINO

Todo o material para a execução desse projeto foi carregado no GitHub do projeto de extensão RoboPatos [[https://github.com/RoboPatos/scicobot\\_2ino](https://github.com/RoboPatos/scicobot_2ino)]. Na página está disponível o código empregado na montagem indicada aqui. Sua principal função é receber três valores pela entrada serial do Arduino:

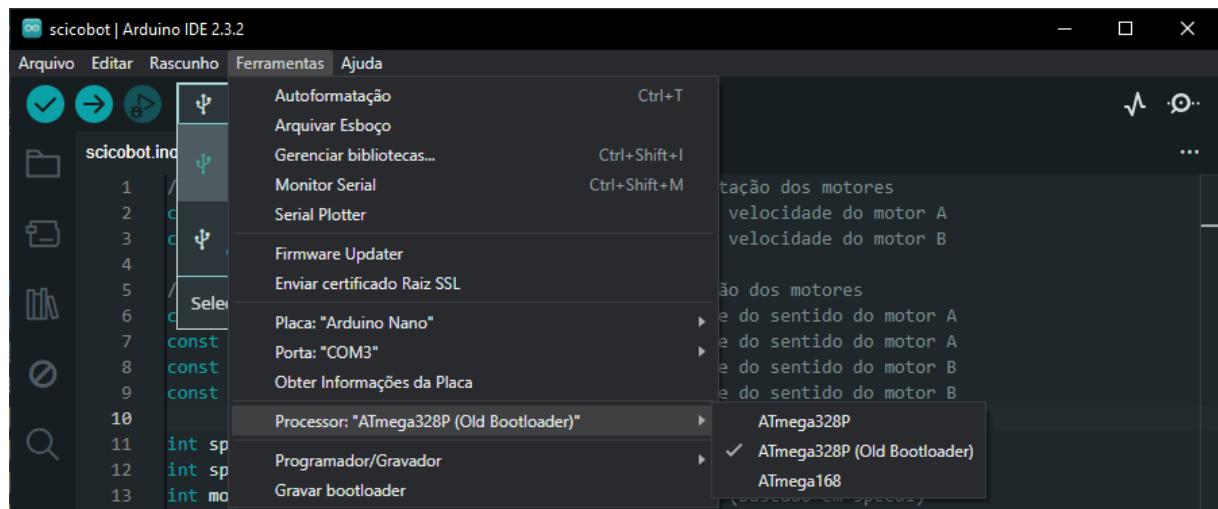
o <velocidade e sentido do motor1> <velocidade e sentido do motor2>

Por exemplo:

- Ambos os motores irão em velocidade máxima e sentido horário: o 255 255;
- Ambos os motores irão em velocidade máxima e sentido anti-horário: o -255 -255;
- Ambos os motores parados: o 0 0;
- Ambos os motores irão em velocidade máxima e sentidos diferentes: o -255 255

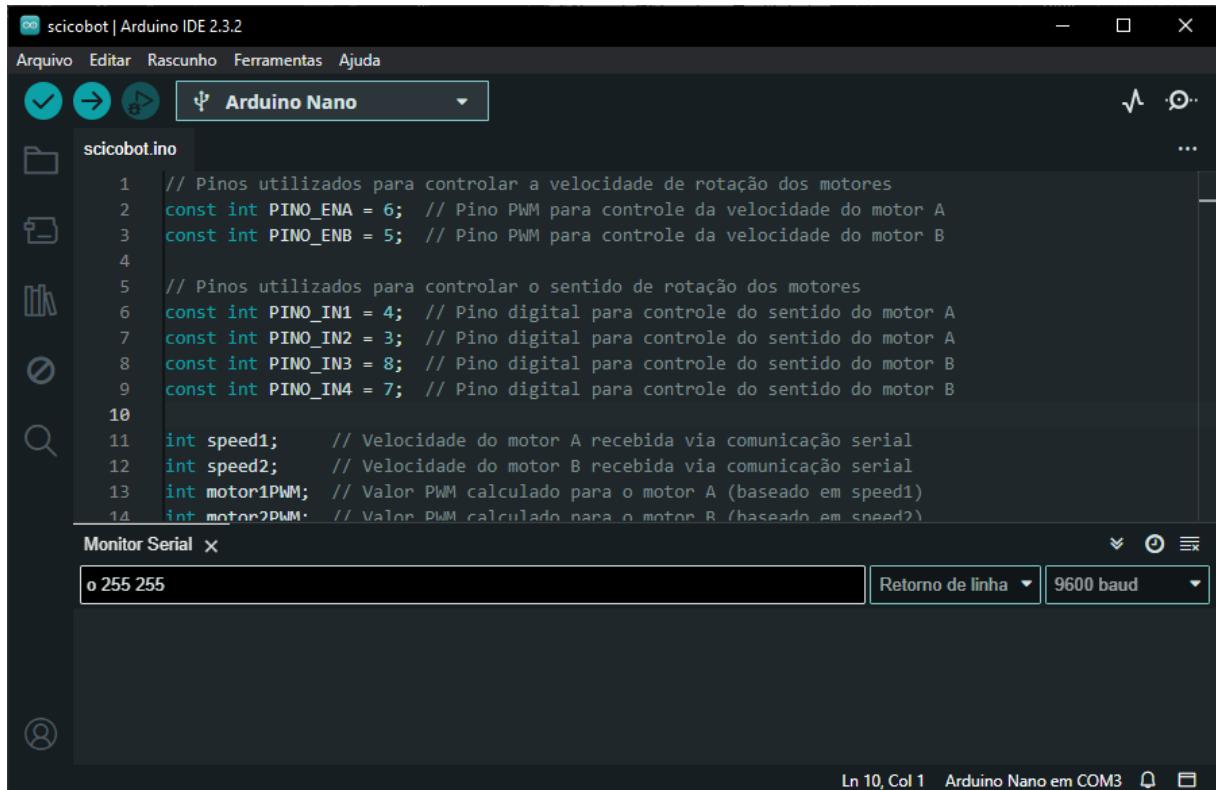
Para sua execução basta carregar o script no Arduino Nano pela IDE, verifique a placa, a porta e a versão do processador da placa que está usando.

Figura 164 – Escolha de placa, porta e processador na IDE Arduino.



Se o projeto estiver montado corretamente, basta abrir a função “Monitor Serial” da IDE e selecionar as opções “Retorno de linha” e “9600 baud”. Feito isso, teste o movimento dos motores conforme apresentado anteriormente, conforme a figura abaixo demonstra:

Figura 165 – Configurações do leitor serial e comando de teste.



## 23. CÓDIGOS SERIAL\_MOTOR NO RASPBERRY

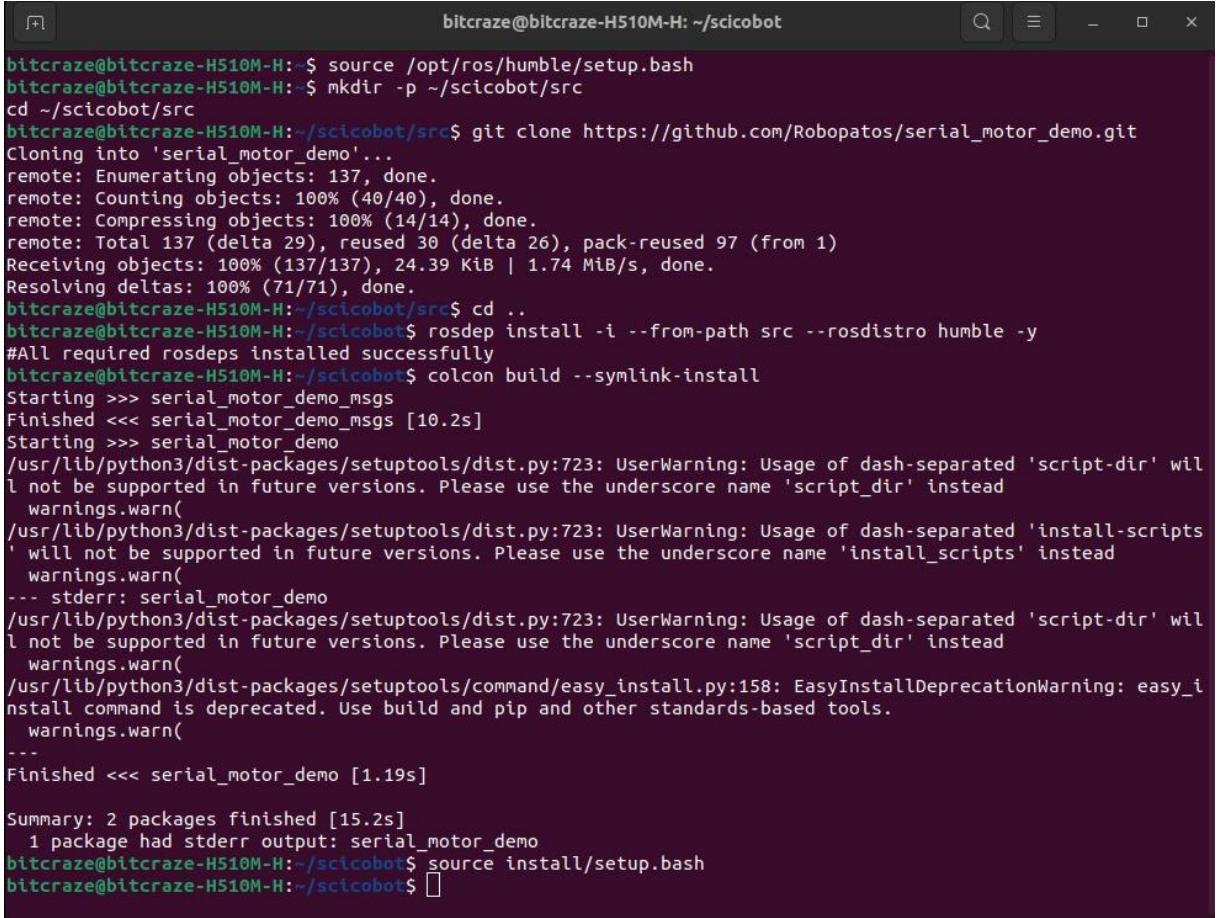
Todo o material para a execução dessa etapa do projeto foi carregado no GitHub do projeto de extensão RoboPatos [[https://github.com/RoboPatos/serial\\_motor\\_demo](https://github.com/RoboPatos/serial_motor_demo)]. Na página está disponível o código empregado no Raspberry Pi para controle do Arduino via comunicação serial. Para a clonagem dos arquivos escreva os seguintes comandos:

```

source /opt/ros/humble/setup.bash
mkdir -p ~/scicobot/src
cd ~/scicobot/src
git clone https://github.com/Robopatos/serial\_motor\_demo.git
cd ..
rosdep install -i --from-path src --rosdistro humble -y
colcon build --symlink-install
source install/setup.bash

```

Figura 166 – Processo de clone e preparação do ambiente para execução do projeto.



```

bitcraze@bitcraze-H510M-H:~$ source /opt/ros/humble/setup.bash
bitcraze@bitcraze-H510M-H:~$ mkdir -p ~/scicobot/src
cd ~/scicobot/src
bitcraze@bitcraze-H510M-H:~/scicobot/src$ git clone https://github.com/Robopatos/serial_motor_demo.git
Cloning into 'serial_motor_demo'...
remote: Enumerating objects: 137, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 137 (delta 29), reused 30 (delta 26), pack-reused 97 (from 1)
Receiving objects: 100% (137/137), 24.39 KiB | 1.74 MiB/s, done.
Resolving deltas: 100% (71/71), done.
bitcraze@bitcraze-H510M-H:~/scicobot/src$ cd ..
bitcraze@bitcraze-H510M-H:~/scicobot$ rosdep install -i --from-path src --rosdistro humble -y
#All required rosdeps installed successfully
bitcraze@bitcraze-H510M-H:~/scicobot$ colcon build --symlink-install
Starting >>> serial_motor_demo_msgs
Finished <<< serial_motor_demo_msgs [10.2s]
Starting >>> serial_motor_demo
/usr/lib/python3/dist-packages/setuptools/dist.py:723: UserWarning: Usage of dash-separated 'script-dir' will not be supported in future versions. Please use the underscore name 'script_dir' instead
    warnings.warn(
/usr/lib/python3/dist-packages/setuptools/dist.py:723: UserWarning: Usage of dash-separated 'install-scripts' will not be supported in future versions. Please use the underscore name 'install_scripts' instead
    warnings.warn(
--- stderr: serial_motor_demo
/usr/lib/python3/dist-packages/setuptools/dist.py:723: UserWarning: Usage of dash-separated 'script-dir' will not be supported in future versions. Please use the underscore name 'script_dir' instead
    warnings.warn(
/usr/lib/python3/dist-packages/setuptools/command/easy_install.py:158: EasyInstallDeprecationWarning: easy_install command is deprecated. Use build and pip and other standards-based tools.
    warnings.warn(
---
Finished <<< serial_motor_demo [1.19s]

Summary: 2 packages finished [15.2s]
  1 package had stderr output: serial_motor_demo
bitcraze@bitcraze-H510M-H:~/scicobot$ source install/setup.bash
bitcraze@bitcraze-H510M-H:~/scicobot$ 
```

Para a execução de qualquer código siga o padrão a seguir:

`ros2 run <nome_do_pacote> <nome_do_executável>`

- Find\_ports

Isso imprimirá no terminal os caminhos e nomes das portas disponíveis no computador, planejado para verificar qual a porta do Raspberry Pi o Arduino está conectado. Para executá-lo escreva:

`ros2 run serial_motor_demo find_ports`

- GUI

Interface gráfica onde pode-se controlar dois motores de forma independente, por entrada PWM (-255 a 255), definindo assim sua orientação e velocidade. Esse código é executado em paralelo com o Driver, porém no computador físico, e é responsável por receber os dois valores definidos pelo usuário para a comunicação entre o Raspberry e Arduino. Pode ser acessada com:

`ros2 run serial_motor_demo gui`

- Driver

Realiza a comunicação entre a interface gráfica e o Arduino. Responável pela comunicação serial propriamente dita, logo é executado no Raspberry Pi em paralelo ao GUI. Ele recebe os valores introduzidos na GUI e envia pela porta serial definida com a baud rate especificada.

Execute com:

```
ros2 run serial_motor_demo driver --ros-args -p serial_port:=/dev/ttyUSB0 -p
baud_rate:=9600
```

Figura 167 – Terminais do Computador e do Raspberry se comunicando e executando o código exemplo.

```
bitcraze@bitcraze-H510M-H: ~/scicobot
bitcraze@bitcraze-H510M-H:~$ source /opt/ros/humble/setup.bash
bitcraze@bitcraze-H510M-H:~$ cd scicobot/
bitcraze@bitcraze-H510M-H:~/scicobot$ source install/setup.bash
bitcraze@bitcraze-H510M-H:~/scicobot$ ros2 run serial_motor_demo gui
```

```
scicobot@ubuntu: ~/scicobot
scicobot@ubuntu:~$ source /opt/ros/humble/setup.bash
scicobot@ubuntu:~$ cd scicobot/
scicobot@ubuntu:~/scicobot$ source install/setup.bash
scicobot@ubuntu:~/scicobot$ ros2 run serial_motor_demo find_ports
/dev/ttyUSB0 - USB2.0-Serial
/dev/ttyAMA0 - ttyAMA0
scicobot@ubuntu:~/scicobot$ ros2 run serial_motor_demo driver --ros-args -p serial_port:=/dev/ttyUSB0 -p baud_rate:=9600
Connecting to port /dev/ttyUSB0 at 9600.
Connected to Serial<id=0xfffffae395870, open=True>(port='/dev/ttyUSB0', baudrate=9600, bytesize=8, parity='N', stopbits=1, timeout=1.0, xonxoff=False, rtscts=False, dsrdr=False)
cmd recebido
o 255 255
cmd recebido
o -255 -255
cmd recebido
o 0 0
```

**SciCoBot**

Controle individual dos motores

Motor 1	-255
Motor 2	-255

**ENVIAR** **PARAR**

## 24. CIRCUITO PARA PRÓXIMAS VERSÕES

- Scicobot\_1-7

Nomeou-se o projeto abaixo de Scicobot\_1-7. Inicialmente, dividiu-se o projeto em duas partes independentes: uma de criação e montagem do Arduino com uma ponte H, que deveria controlar dois motores à partir de uma entrada serial usando uma função “Serial.read()”, e a segunda fazendo um Raspberry ler um comando escrito no teclado e enviá-lo serialmente para o Arduino ligado em alguma de suas portas usando ROS 2.

Para a montagem do circuito do projeto primeiramente focou-se na ligação do Arduino com a ponte H, modelo L9110s. Optou-se pelo uso do Arduino Nano por este ser mais compacto, com baixo custo e por suprir as portas necessárias para o funcionamento do circuito, mas qualquer outra placa com 4 portas PWM pode ser empregada.

Tabela 4 – Pinagem dos componentes;

<b>DRIVER PONTE H L9110S</b>	<b>MOTOR</b>	<b>MICROCONTROLADOR ARDUINO</b>	<b>SENTIDO</b>
IB1	1	D6	ANTIHORÁRIO
IA1	1	D9	HORÁRIO
IB2	2	D10	ANTIHORÁRIO
IA2	2	D11	HORÁRIO

Figura 168 – Montagem da sugestão de projeto.

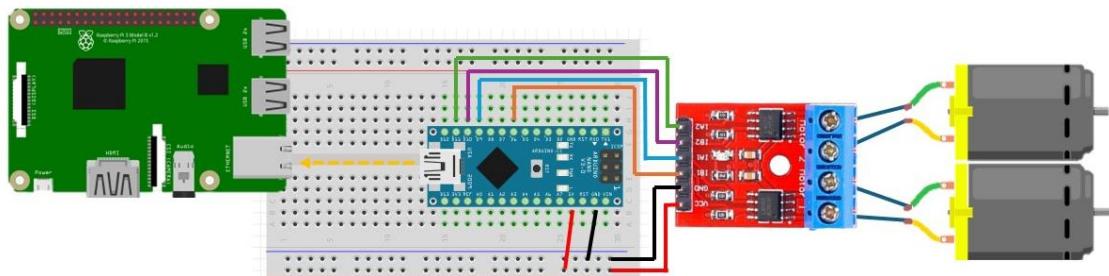


Figura 169 – Pinagem da Ponte H modelo L9110s.

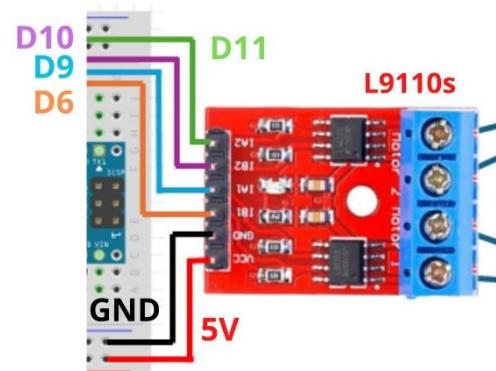


Figura 170 – Pinagem do Arduino Nano.

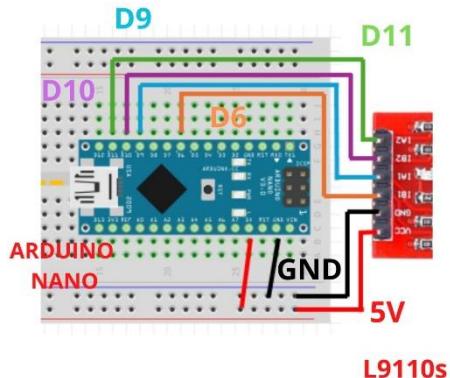
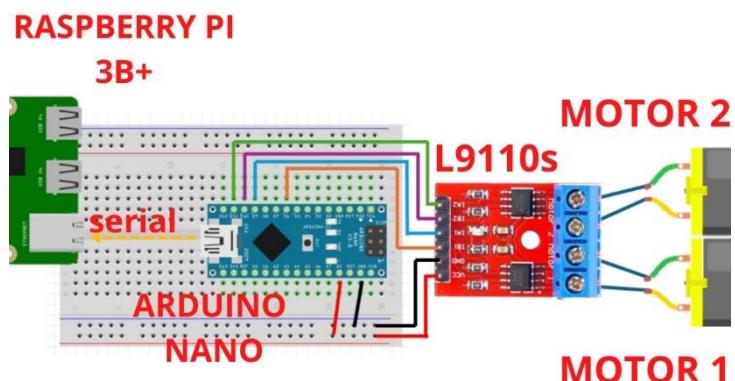


Figura 171 – Montagem completa comentada.



A ideia aqui é a criação de um executável ROS 2 para complementar a montagem apresentada acima e o código disponível no GitHub da RoboPatos, que recebe via serial números entre 1 e 7 e executa uma ação para cada um desses valores. Imagina-se a execução de um script python que apresente ou não uma interface gráfica, mas seria de grande aproveitamento se empregasse o código disponível em:

[https://github.com/RoboPatos/scicobot\\_2ino/blob/main/outros\\_códigos/scicobot\\_1-7/scicobot\\_1-7.ino](https://github.com/RoboPatos/scicobot_2ino/blob/main/outros_códigos/scicobot_1-7/scicobot_1-7.ino)

- Serial\_motor\_demo (Botão Dinâmico)

Também era previsto a existência de outros dois botões na interface gráfica: “Enviar dinâmico” e “Parar dinâmico”. Sua função seria enviar indefinidamente os valores lidos na interface, de forma de ao serem alterados o Arduino executaria automaticamente. Infelizmente isso não foi concretizado, mas existem alguns comentários no código do raspberry referentes ao que se planejou.

Por conta disso, acredito que essa seria uma boa continuação do projeto, tanto para aproveitamento da montagem como da ideia de fácil visualização do que está acontecendo.