

Guia de Instalação e Configuração da IDE do Arduino na Labrador e Primeiros Passos com Computação Física

**MAKER
SPACE IOT**

Guia de Instalação e Configuração da IDE do Arduino na Labrador e Primeiros Passos com Computação Física

Realização

Escola Politécnica da Universidade de São Paulo
Associação do Laboratório de Sistemas Integráveis Tecnológico
Furnas Centrais Elétricas S.A

Autores

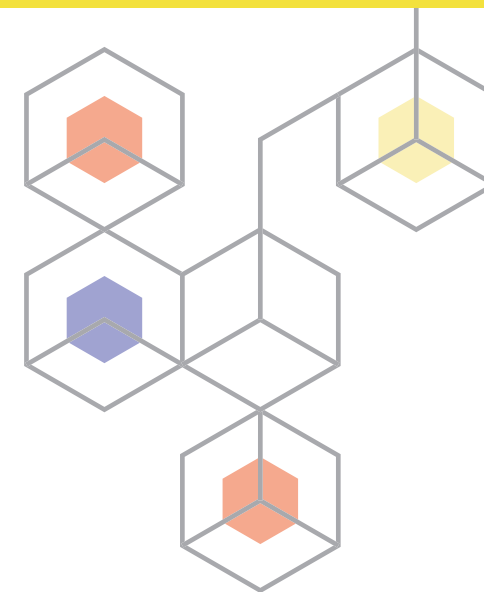
Nathan Rabinovitch
Rodrigo Lima Azeredo

Edição

Coordenação Científica: Roseli de Deus Lopes e Irene Karaguilla Ficheman
Coordenação editorial: Valkiria Venancio
Capa, projeto gráfico, editoração e ilustrações: Andrea Sofia Majjul Fajardo

Agradecimento

Agradecemos a Equipe da Caninos Loucos (<https://caninosloucos.org/pt/>) por apoiarem a iniciativa do Programa Maker Space IoT para alunos de escolas públicas paulistas com a doação de 60 placas Labrador, que oportunizaram aos estudantes e professores conhecerem essa tecnologia brasileira. Além de proporcionar aos professores durante a formação uma visita a Central Multiusuário em Manufatura Avançada em Internet das Coisas no Centro Interdisciplinar em Tecnologias Interativas - CITI (<https://www.lsi.usp.br/citi/>).



Introdução

Este guia foi criado especialmente para apoiar professores e estudantes participantes do programa **Maker Space IoT para estudantes de escolas públicas paulistas** e consiste em um passo a passo para instalar a IDE Arduino na placa Labrador e nos primeiros passos para começar a desenvolver projetos interativos com computação física.

Para iniciar espera-se que você já tenha conectado na labrador um display ou um monitor auxiliar juntamente com um mouse e um teclado para conseguir fazer a instalação.

Autores:

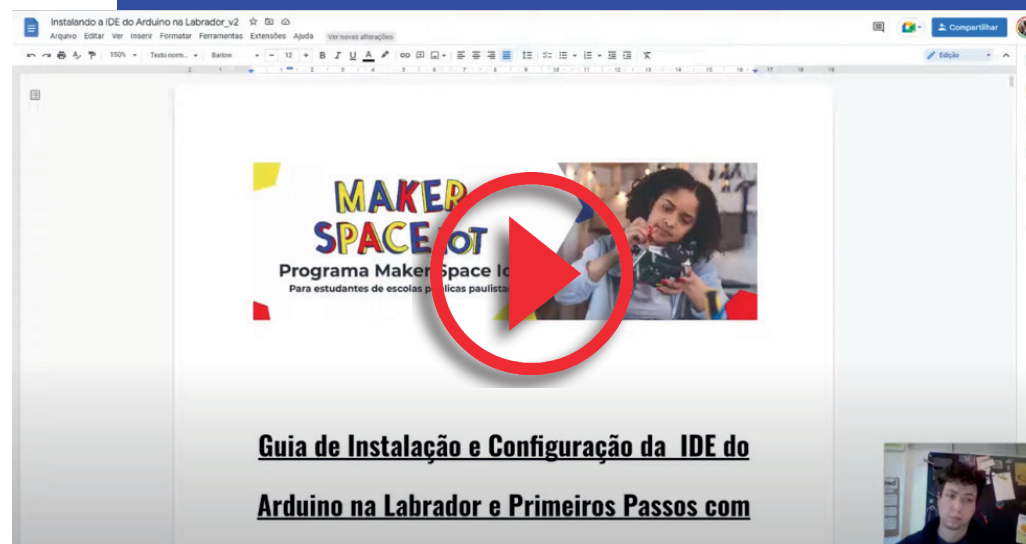


**Nathan
Rabinovitch**



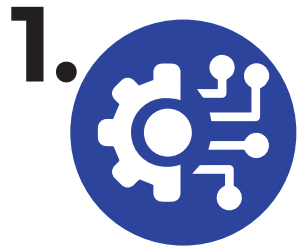
**Rodrigo Lima
Azeredo**

**Antes de começar, veja este vídeo:
[Guia Labrador + Arduino | Makerspace IoT LSITec USP](https://youtu.be/9avhgfFbJjA)**



<https://youtu.be/9avhgfFbJjA>

Sumário



Componentes e
conexão

Pg. 6



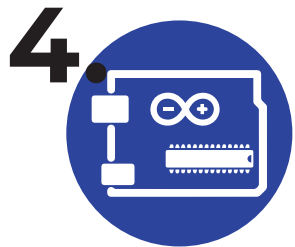
Como conectar a
uma rede Wi-Fi

Pg. 7



Como alterar
configurações de
data e hora

Pg. 8



Instalação da IDE
do Arduino

Pg. 10



Como abrir
projetos salvos na
Labrador dentro
da IDE do Arduino

Pg. 12



Mudança do
idioma do sistema
operacional: Inglês
para o Português

Pg. 13

Parte 1

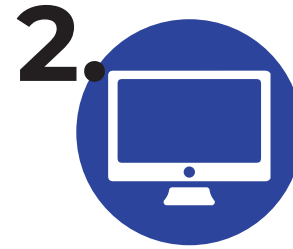
Labrador e Arduino IDE

Parte 2

Primeiros passos com computação física



Apresentação e materiais
Pg. 16



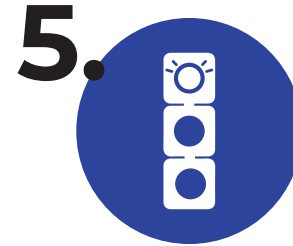
Configurando o ambiente
Pg. 18



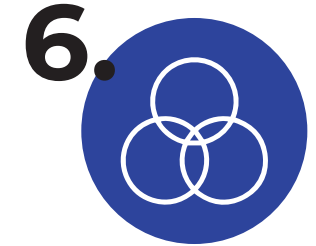
Atividade 1: Blink do LED
Pg. 20



Atividade 2: Dois LEDs piscando alternadamente
Pg. 22



Atividade 3: Construa e programe um semáforo
Pg. 23



Atividade Opcional: Desafio das cores
Pg. 24



O uso da laboratório no programa “Maker Space IoT para estudantes de escolas públicas Paulistas”
Pg. 26

Parte 1: Labrador e Arduino IDE

1. Componentes e conexão



- Placa Labrador;
- Monitor ou Display;
- Cabo HDMI;
- Cabo USB B;
- Teclado;
- Mouse;
- Fonte de alimentação da Labrador.
- Conexão 1: Alimentação da Placa;
- Conexão 2: Entrada para fone de ouvido;
- Conexão 3: Entrada para o cabo HDMI;
- Conexão 4: Entrada USB para teclado e mouse;
- Conexão 5: Conexão de internet.

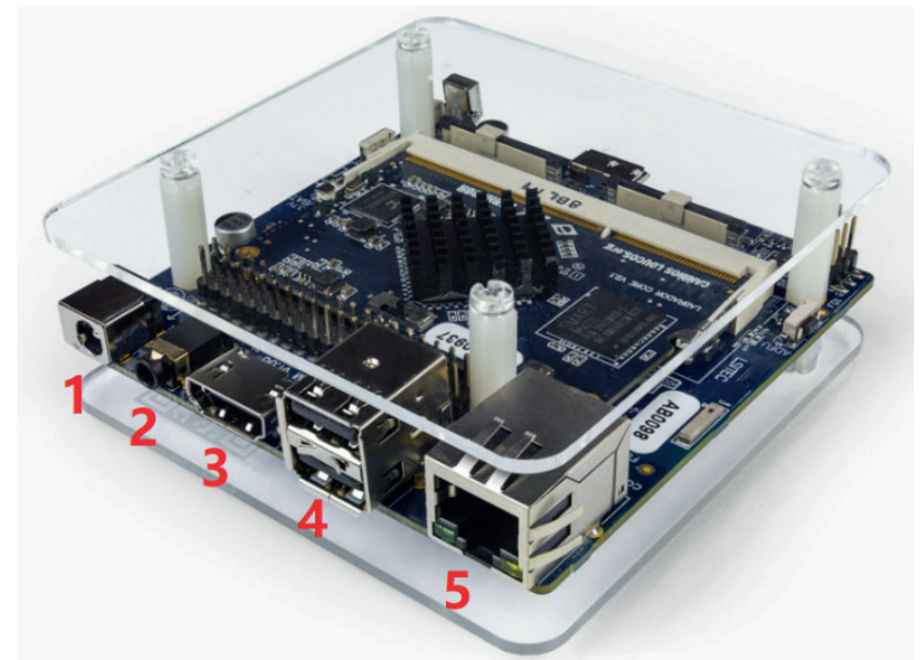


Imagem 1: Conexões da placa Labrador

2. Como conectar a uma rede Wi-Fi



Inicialmente, partindo do pressuposto que sua Labrador ainda não esteja conectada a uma rede, iremos desenvolver o passo a passo de como conectar sua placa a uma rede Wi-Fi. Dessa forma, ao abrir a tela inicial do Debian (Sistema Operacional disposto na caninos), você vai se deparar com um **login** e **senha** (imagem 2), basta colocar tanto no login quando na senha “**caninos**”, em seguida já com a tela inicial aberta vá na barra de tarefas no canto inferior esquerdo no menu iniciar (Imagem 3).

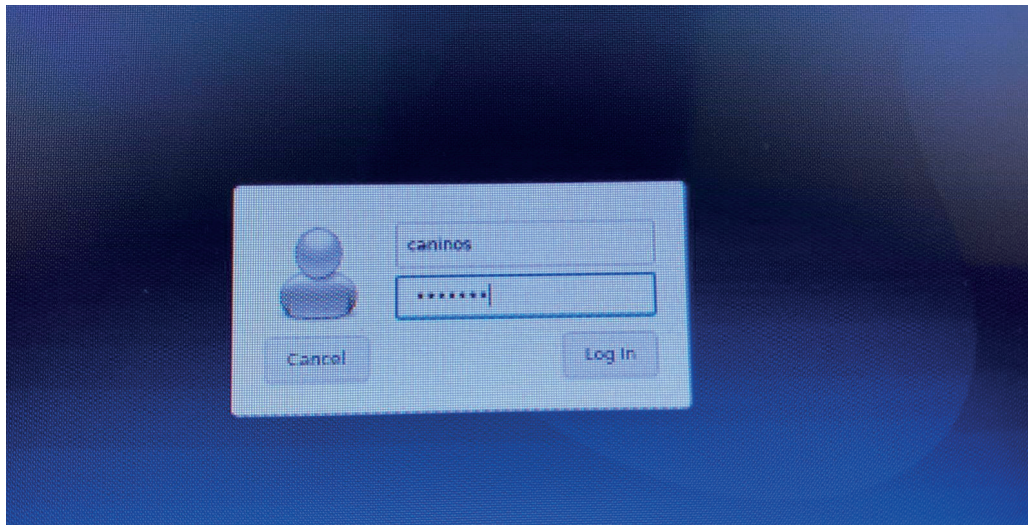


Imagem 2: Login da Caninos

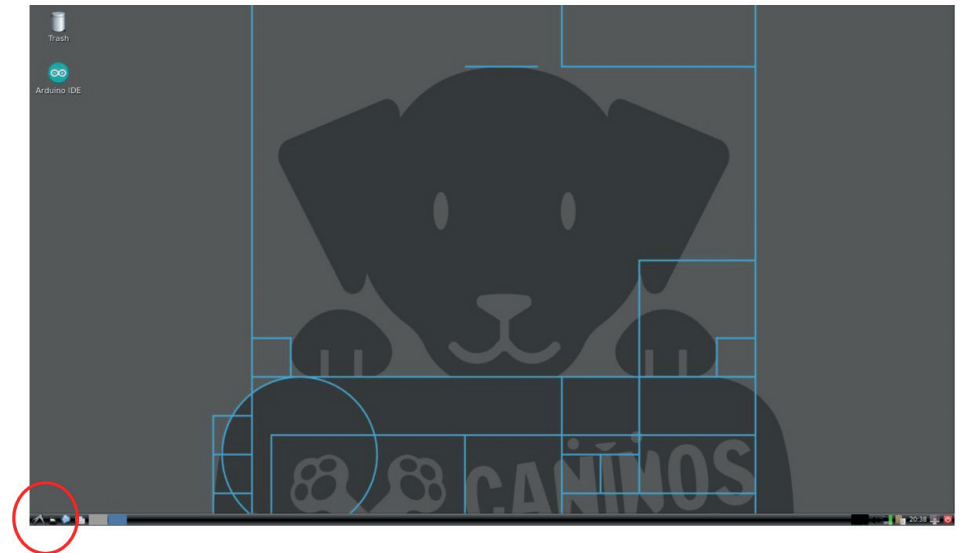


Imagem 3: Área de Trabalho da Caninos

3. Como alterar configurações de data e hora



Para fazer as alterações da data e hora, para isso clique em “**Preferences**” depois em “**Time and Date**”, logo em seguida abrirá em sua tela a imagem 4, agora basta clicar em “**Unlock**” colocar a senha “**caninos**” e fazer a alteração da data e hora referente ao momento em que esteja fazendo a alteração. Por fim, basta clicar em “**Lock**”, que as alterações serão salvas.

Segue o passo a passo a seguir (Imagens 4 e 5):

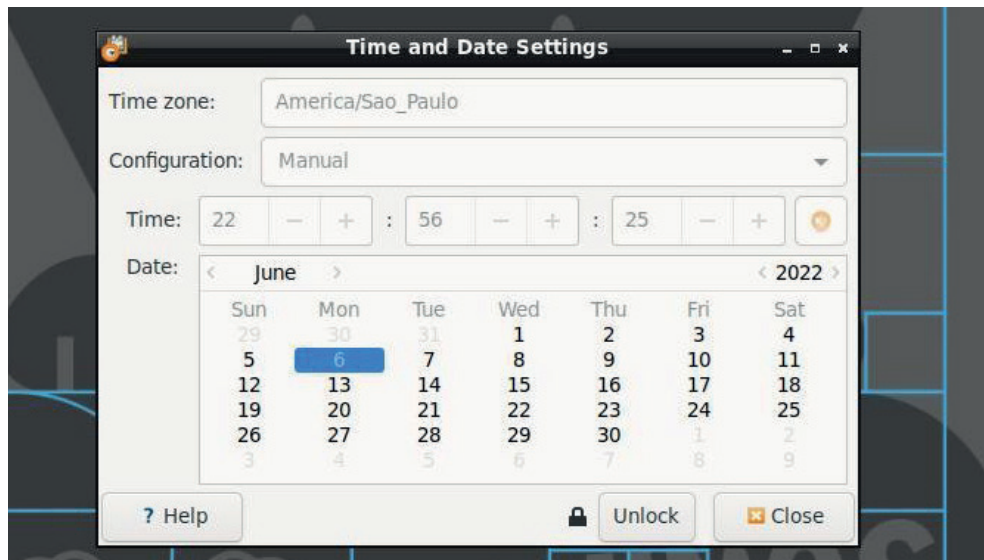


Imagem 4: Configuração da data e hora - Fazendo alterações.

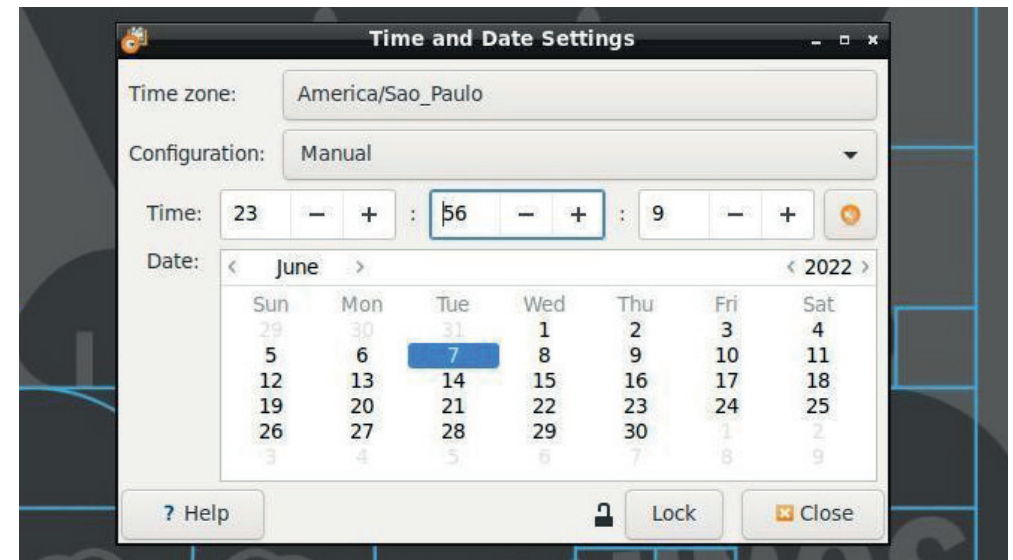


Imagem 5: Configuração da data e hora - Salvando as alterações feitas.

Depois dos ajustes feitos, volte no menu **iniciar**, clique em **“internet”**, em seguida em **“wicd Network Manager”**. Depois disso, abrirá em sua tela o gerenciador de redes Wi-Fi, onde você irá conectar na rede de internet disponível (no meu caso é “Raposada”), clique em **“connect”** e depois em **“ok”** (Imagem 6 e 7).

Logo em seguida no espaço vazio na frente de **“key”**, insira a senha da sua rede Wi-Fi (Imagem 8 e 9).

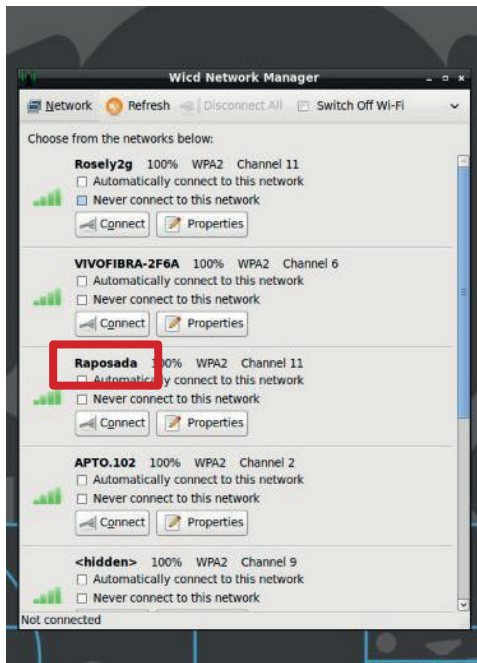


Imagem 6: Gerenciador de redes Wi-fi Wicd.

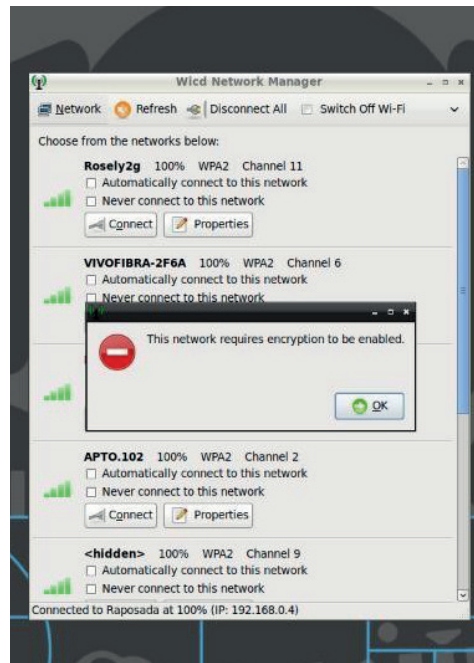


Imagem 7: Gerenciador de redes Wi-Fi Wicd.

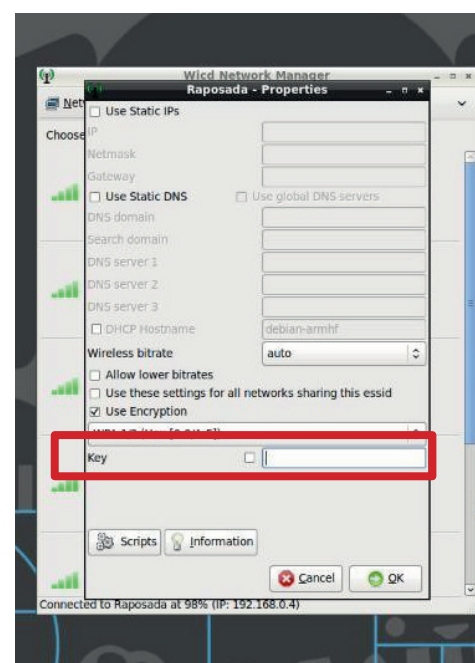


Imagem 8: Gerenciador de redes Wi-Fi Wicd.

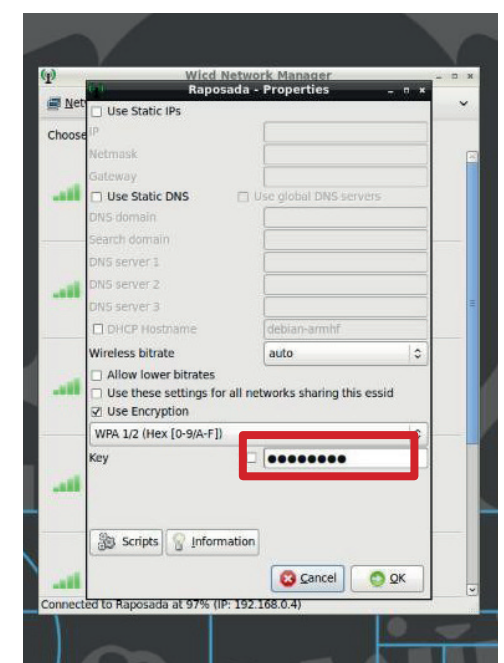


Imagem 9: Gerenciador de redes Wi-fi Wicd.

4. Instalação da IDE do Arduino



Após conectar sua Labrador a internet, vamos iniciar o processo de instalação da IDE Arduino. Nisso, será necessário que você volte em sua tela inicial e vá no menu iniciar (Imagem 3), clique em **“Internet”**, depois em **“Firefox ESR”**.

Irá abrir uma guia do FireFox, assim, basta acessar o seguinte link: <https://www.arduino.cc/en/software>

Em seguida, clique em **“Linux ARM 32 bits”** (Imagem 10).

Após a finalização do download, volte na tela inicial da caninos, vá no menu iniciar (Imagem 3), clique em **“System Tools”**, depois em **“File Manager PCManFM”**. Logo em seguida, acesse a pasta de downloads, aperte com o botão direito do mouse em cima da pasta selecionada (Imagem 11) escolhendo depois a opção **“Extract Here”**.



Imagem 10: Site do Arduino.

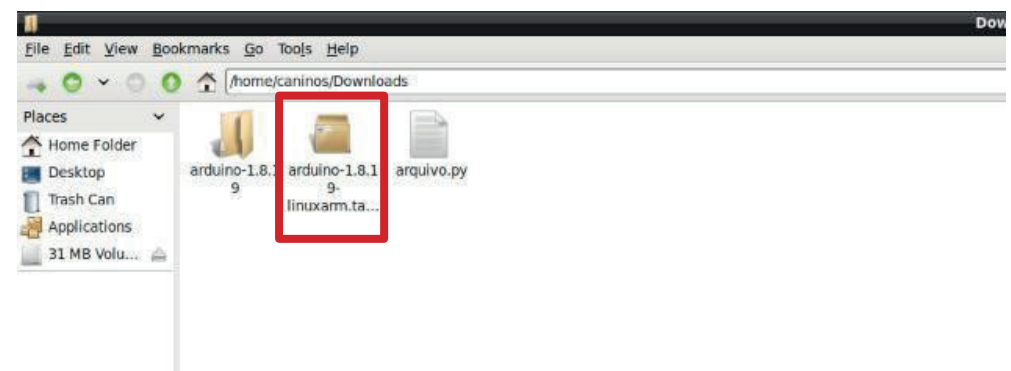
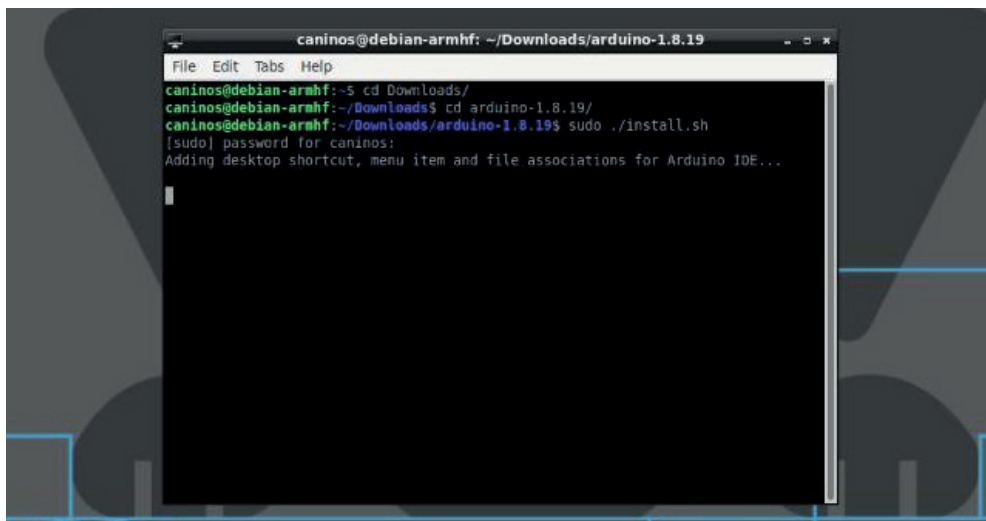


Imagem 11: Pasta de downloads.

Feito isso, volte a tela inicial e acesse o menu iniciar (Imagem 3). Clique em “**System Tools**” em seguida em ‘**LXTerminal**’. Com o terminal aberto, digite “**cd Downloads**” e aperte enter, digite “**cd arduino- 1.8.19**”, aperte enter, em seguida digite “**sudo ./install.sh**” e aperte enter. Agora, o terminal pedirá uma senha, digite “**caninos**” (nessa parte você não consegue visualizar o que está digitando, basta digitar e apertar enter).

Segue o passo a passo nas figuras a seguir:



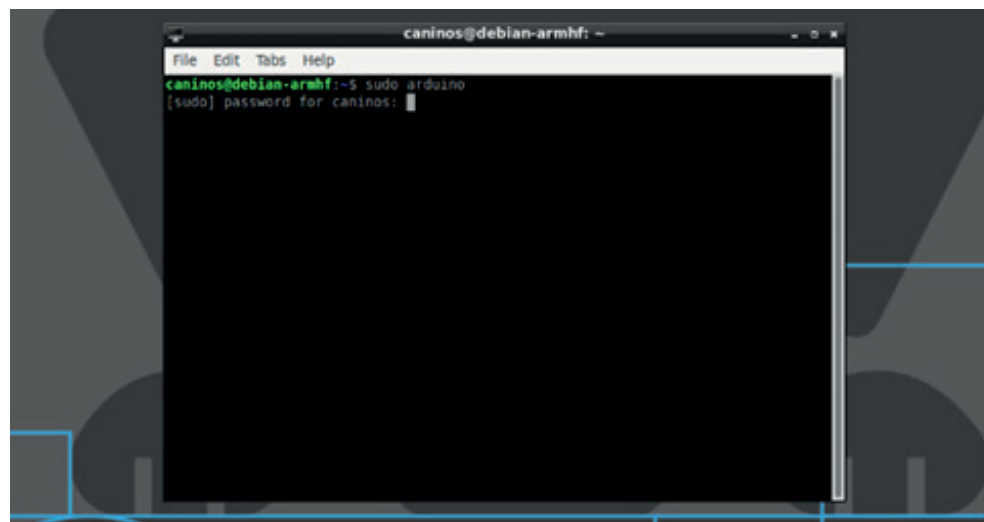
```
caninos@debian-armhf: ~/Downloads/arduino-1.8.19
File Edit Tabs Help
caninos@debian-armhf:~$ cd Downloads/
caninos@debian-armhf:~/Downloads$ cd arduino-1.8.19/
caninos@debian-armhf:~/Downloads/arduino-1.8.19$ sudo ./install.sh
[sudo] password for caninos:
Adding desktop shortcut, menu item and file associations for Arduino IDE...
```

Imagem 12: Terminal do Debian - Instalação do Arduino.



IMPORTANTE

Após a instalação da IDE do Arduino na Labrador, toda vez que quiser abri-lo, basta clicar no menu iniciar (Imagem 3), clique em “**System Tools**” em seguida em “**LXTerminal**”, então basta digitar “**sudo arduino**”, apertar enter e digitar a senha “caninos” que sua IDE irá abrir (Imagem 13).



```
caninos@debian-armhf: ~
File Edit Tabs Help
caninos@debian-armhf:~$ sudo arduino
[sudo] password for caninos: 
```

Imagem 13: Terminal do Debian - Processo inicial para abrir a IDE arduino.



IMPORTANTE

Caso você **não tenha acesso à internet** em sua escola, ou caso o acesso seja ruim, é possível fazer o download do **instalador** da IDE do Arduino e salvá-lo em um **pen-drive**. Com o arquivo em mãos, é possível seguir os passos de instalação, copiando o instalador para dentro da Labrador.

5. Como abrir projetos salvos na Labrador dentro da IDE do Arduino

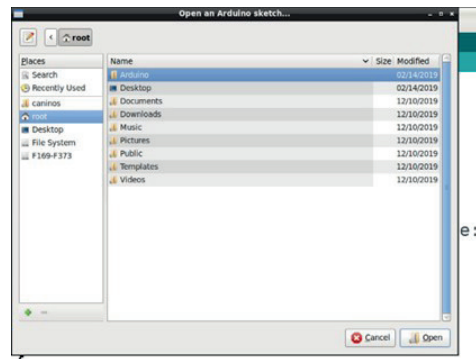


Importante sabermos que depois que você tem o primeiro contato com a Labrador e faz seu primeiro programa utilizando o Arduino, caso queira abrir este ou um outro programa feito anteriormente. Basta repetir o processo mencionado acima e depois com a IDE aberta, vá na parte superior esquerda em “File” ou “Arquivo”, em seguida em “Open” ou “Abrir”. Nisso, vai abrir opções de pasta, agora basta clicar na pasta que você salvou antes e abrir a sua programação, segue o passo a passo a seguir (Imagens 14, 15 e 16):



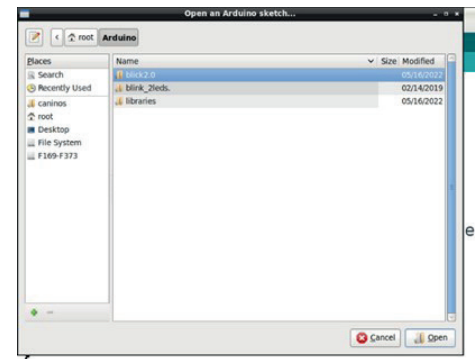
Obs.:

É importante sempre manter cada programação feita organizada, para isso recomendamos que sempre salve a programação feita em uma única pasta. Isso para que você sempre saiba onde estão seus projetos (no meu caso salvei no “root” e criei uma pasta chamada “Arduino”)



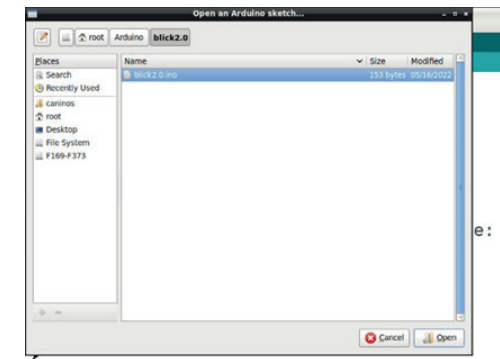
```
void loop() {  
  digitalWrite(vermelho,HIGH);  
  
  Serial.println("Vermelho ACESSO!");  
  Serial.println("-----");  
}
```

Imagem 14: Pasta raiz “root”.



```
void loop() {  
  digitalWrite(vermelho,HIGH);  
  
  Serial.println("Vermelho ACESSO!");  
  Serial.println("-----");  
}
```

Imagem 15: Programação salva na pasta “Arduino”



```
void loop() {  
  digitalWrite(vermelho,HIGH);  
  
  Serial.println("Vermelho ACESSO!");  
  Serial.println("-----");  
}
```

Imagem 16: Abrindo a programação feita.

6. Mudança do idioma do sistema operacional: Inglês para o Português



O tutorial a seguir consiste na mudança do idioma inglês do sistema operacional debian para o português. Mudança a qual é optativa, lembrando que as alterações a seguir não impactam na utilização da placa labrador.

Inicialmente você vai no seu menu iniciar (Imagem 3) e clique em “**System Tools**” em seguida em “**LXTerminal**”. Com o terminal aberto digite “**sudo su**” em seguida aperte enter e digite a senha “**caninos**”(senha a qual quando estiver digitando não conseguirá ver o que está digitando).

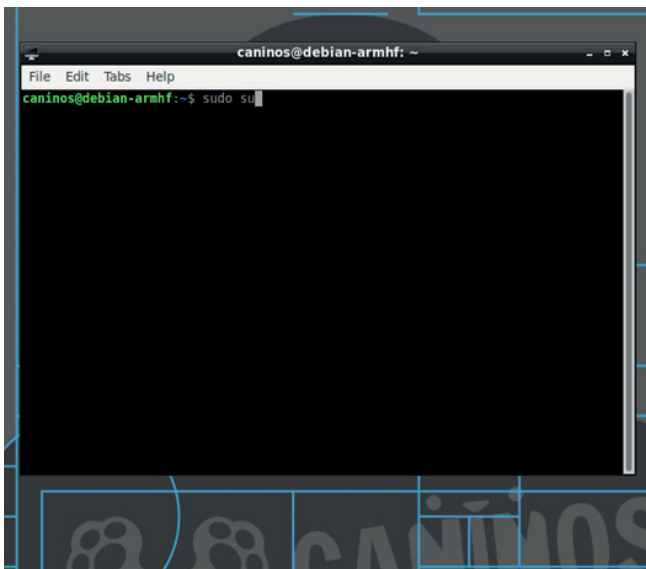


Imagem 17: Terminal do Debian - Virando administrador do Sistema Operacional.

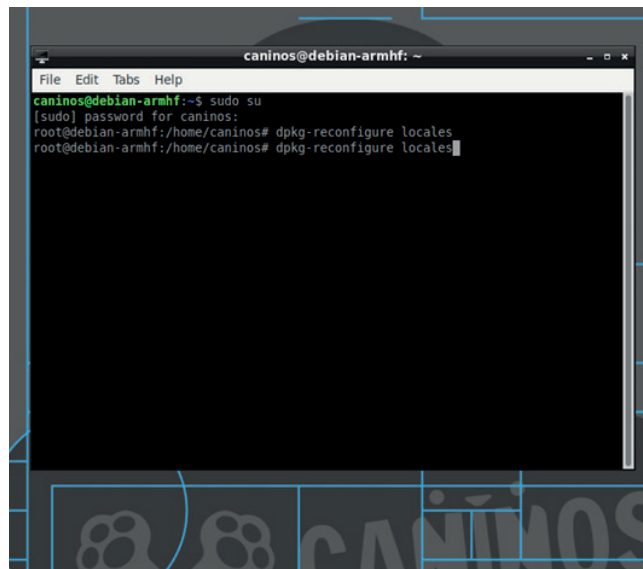


Imagem 18: Terminal do Debian - Configurando as variáveis locais.

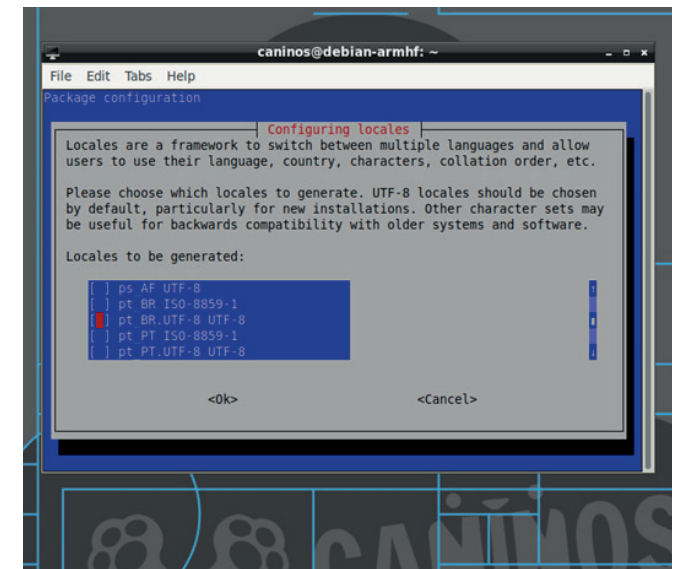


Imagem 19: Configurações locais.

Feito isso, e ainda no terminal, digite “dpkg-reconfigure locales” para configurar as variáveis locais em seguida enter (Imagem 18).

Depois desse processo aguarde um momento e assim que abrir em sua tela a imagem 19, utilizando a seta do seu teclado clique para baixo até chegar em “pt_BR.UTF-8 UTF-8”. Com essa opção selecionada, aperte espaço e em seguida enter.

Depois desse processo abrirá em sua tela a imagem 20, com ela aberta e utilizando a seta do seu teclado escolha a opção “pt BR.UTF-8” para configurar para português.

Depois dessa etapa, aguarde até a configuração terminar (Imagem 21).

Terminado a configuração digite em seu terminal “reboot” para reiniciar e aplicar as mudanças feitas (Imagem 22).

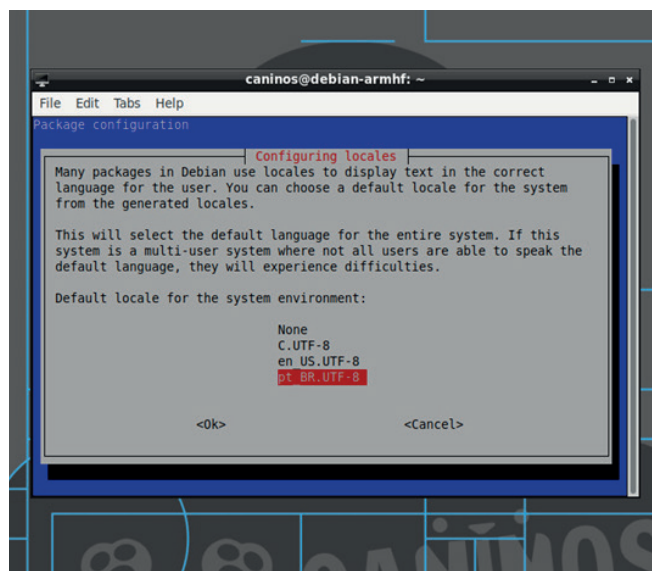


Imagem 20: Configurando o Debian para Português.

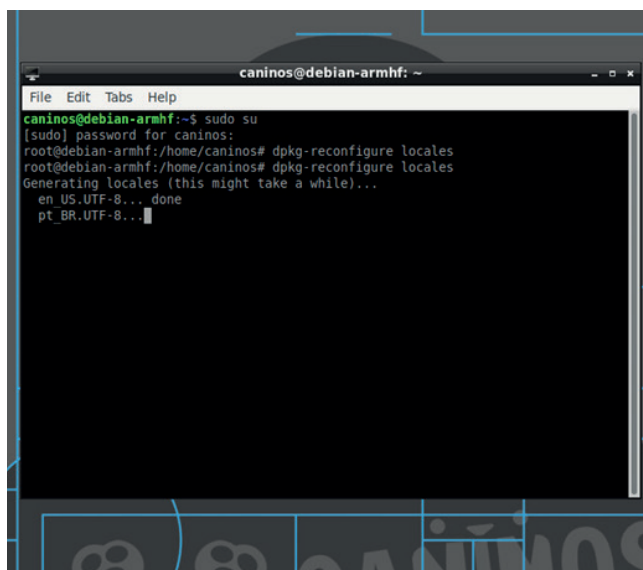


Imagem 21: Configurando a linguagem.

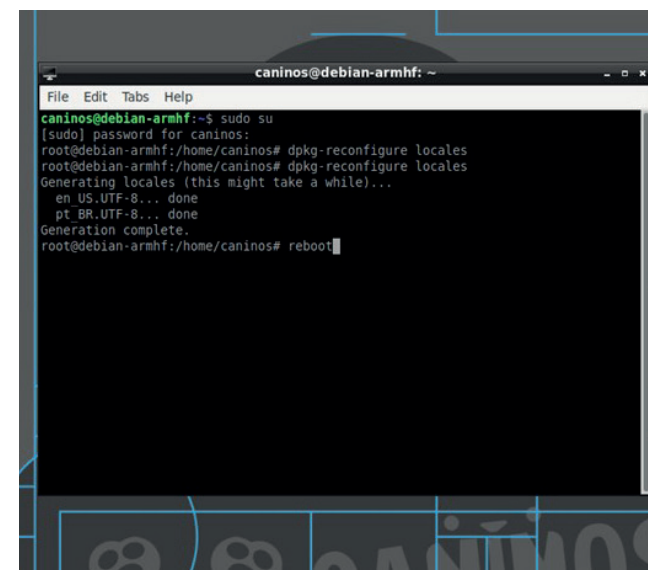


Imagem 22: Reiniciando o Sistema.

E fim, agora seu Sistema Operacional já está em português.



Parte 2: Primeiros passos com computação física

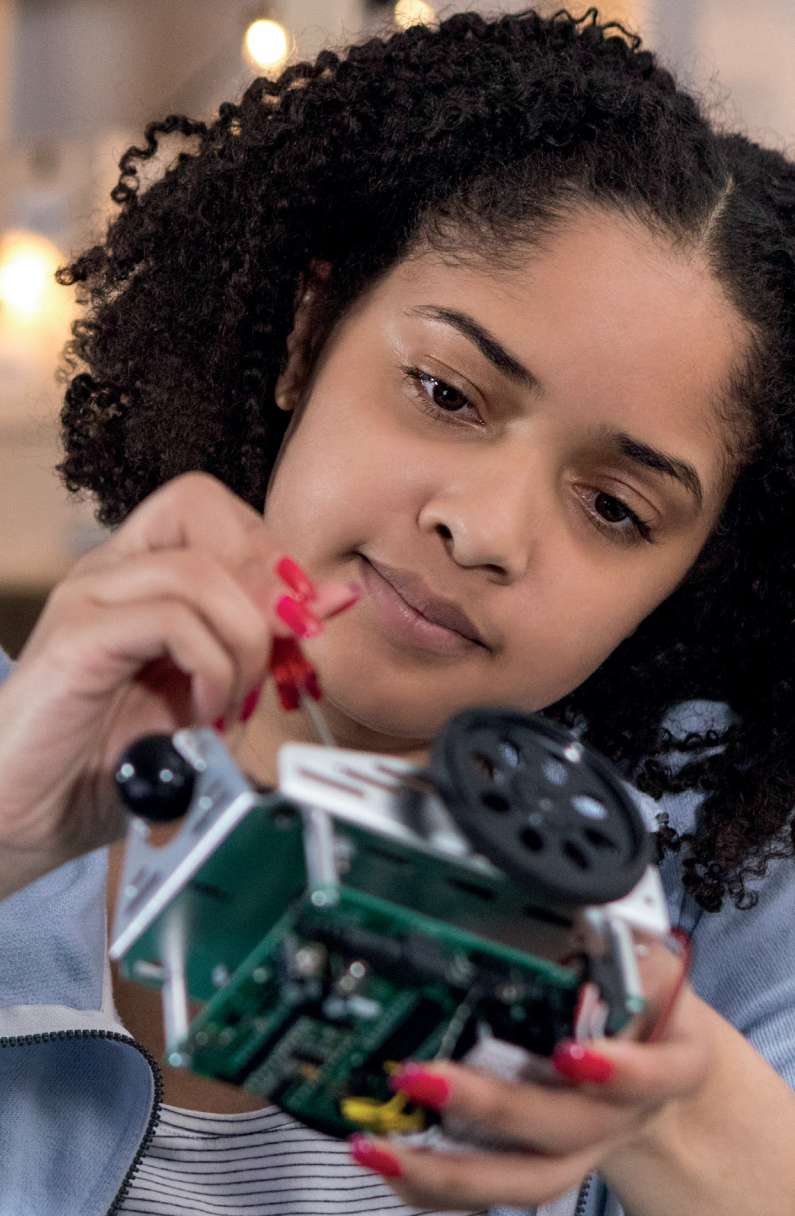
1. Apresentação e materiais



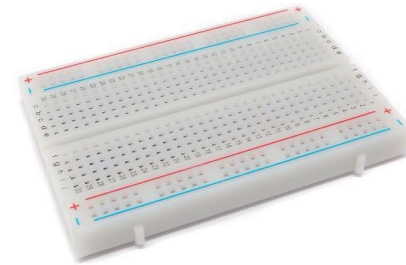
Estas atividades foram pensadas como ponto de partida para nos aventurarmos no maravilhoso universo da computação física. Uma vez que a IDE do Arduino estiver instalada no seu computador, seja ele um notebook ou um Labrador, experimente desenvolver os projetos descritos nos tutoriais a seguir.

Os dois principais objetivos desta etapa são:

- Verificar se todo o sistema (Computador + Arduino IDE + Circuitos + Arduino) está funcionando de forma correta;
- Desenvolver os primeiros projetos de computação física com Arduino para que seja possível desenvolver projetos mais complexos e interessantes no encontro síncrono do dia 22 de junho.



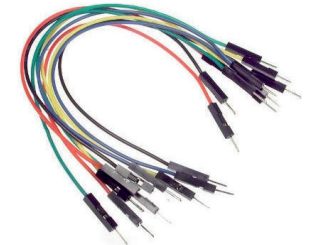
Materiais



- ◆ 1 Protoboard de 400 furos



- ◆ 1 placa de Arduino Uno e 1 cabo USB Serial para o Arduino



- ◆ 20 jumpers macho macho para protoboard (destaque cada jumper, pois eles costumam vir grudados)



- ◆ 3 resistores de 220 Ohms (destaque cada resistor, pois eles costumam vir grudados)



- ◆ 1 LED RGB com cátodo comum



- ◆ 1 LED verde, 1 LED amarelo, 1 LED vermelho

Orientações de condução

2. Configurando o ambiente



Passo 1: Conecte a placa Arduino na porta USB da sua Labrador ou computador.

Passo 2: Clique em ferramentas e em seguida identifique sua placa arduino. Em nosso caso, Arduino Uno.



Imagem 23: Placa Arduino e cabo USB Serial.

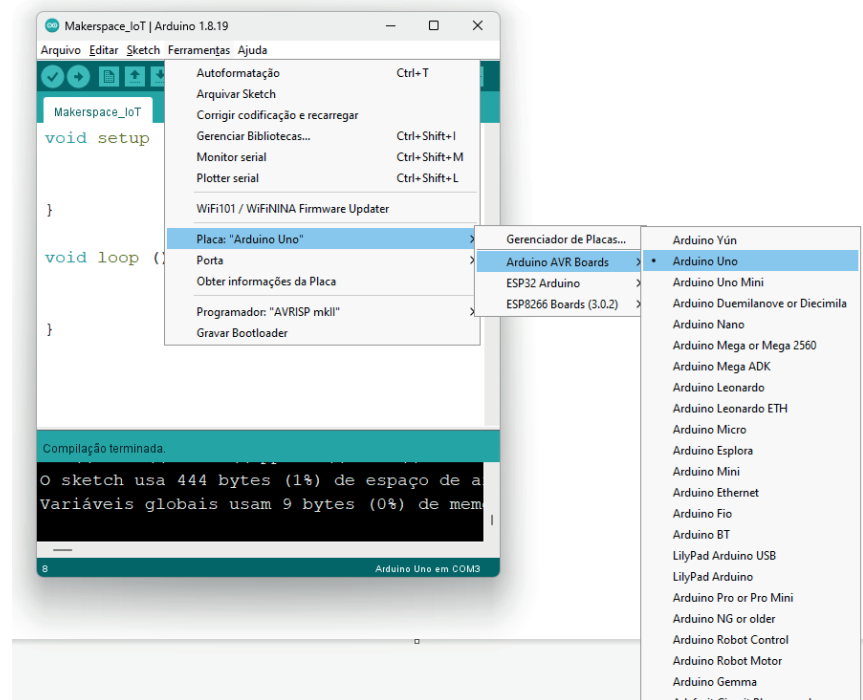


Imagem 24: Caminho para seleccionar modelo de placa na Arduino IDE.

Passo 3: Ainda em **ferramentas**, clique em porta e clique na porta que aparecerá para você. Em nosso exemplo, apareceu COM 8 (mas pode ser um número diferente).

Passo 4: Agora que já configuramos o ambiente de desenvolvimento, vamos testá-lo! Para isso, utilizaremos o programa Blink, que é um pisca-pisca. Para acessar este programa clique no canto superior esquerdo em **Arquivo > Exemplos > 01. Básicos > Blink**. Em seguida, envie o código para o Arduino utilizando o botão de upload conforme imagem 26:

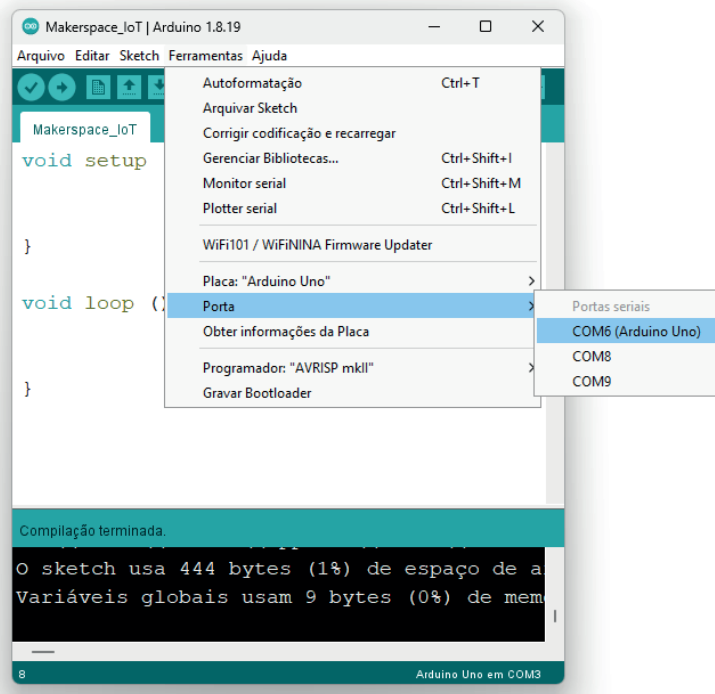


Imagem 25: Caminho para selecionar porta USB onde o Arduino está conectado.

Este botão é responsável por enviar o seu código para o Arduino!

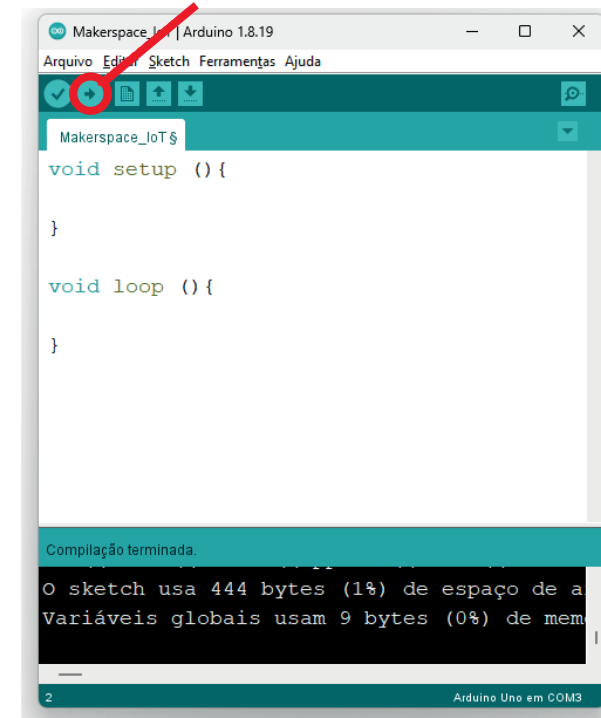
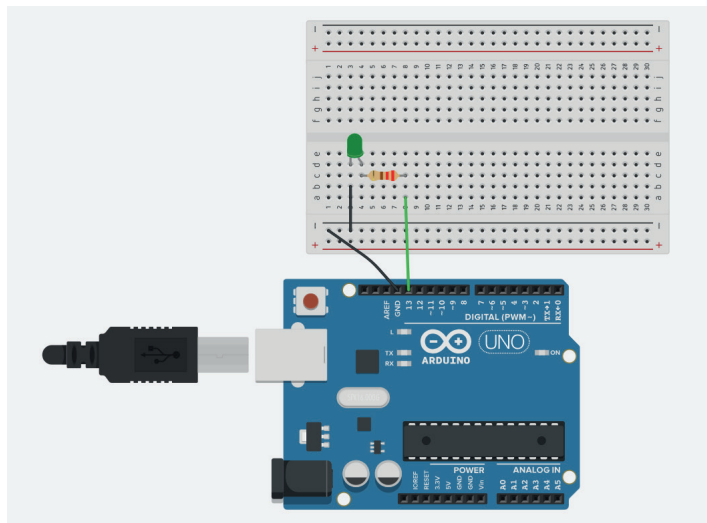


Imagem 26: Botão de upload de códigos da IDE para o Arduino.

3. Atividade 1: Blink do LED



Monte o circuito abaixo na protoboard. Uma vez montado o circuito, é hora de programá-lo na IDE do Arduino na Labrador para que o **LED** pisque com intervalos de 1 segundo. Para isso, utilize o código abaixo:



Fonte: Tinkercad

Imagem 27: Circuito de 1 LED na porta 13

Código:

```
void setup(){
  pinMode(13, OUTPUT);
}

void loop(){
  digitalWrite(13,1);
  delay(1000);
  digitalWrite(13,0);
  delay(1000);
}
```



DICA

Caso ao final da programação que você fará a seguir o LED não acenda, inverta a posição dos terminais do LED na protoboard, pois pode ser que a polaridade dele esteja invertida.

Note que este código funciona em dois blocos principais chamados de **void setup** e **void loop**. Ao ser ligado, a placa **Arduino** executa o que estiver dentro do bloco setup uma única vez, que nesse caso é a declaração da **porta 13** como um **atuador** (OUTPUT). Em seguida entramos no bloco loop, que funciona como uma laço infinito no qual o que estiver em seu interior se repetirá de forma indefinida.

Ele inicia com a função **digitalWrite(pino, estado)** que informa o estado do **LED** da **porta 13**. Como o **estado** usado é **1**, esta função irá **acender** o **LED** conectado à **porta 13**.

Em seguida, a função **delay** faz a placa pausar por 1 segundo (1000 milissegundos).

Na sequência, o **LED** é apagado com a função **digitalWrite(pino, estado)** na **porta 13** e o **número zero**, e há uma pausa de **1 segundo** com **delay(1000)**.

Neste momento, a **função loop** chega ao seu final e reinicia o laço de forma indefinida. Esta repetição faz com que o **LED** fique piscando.

Em resumo o bloco **void setup** configura o comportamento das portas (pin, ou pino) e bloco **void loop** repete a ação indefinidamente.

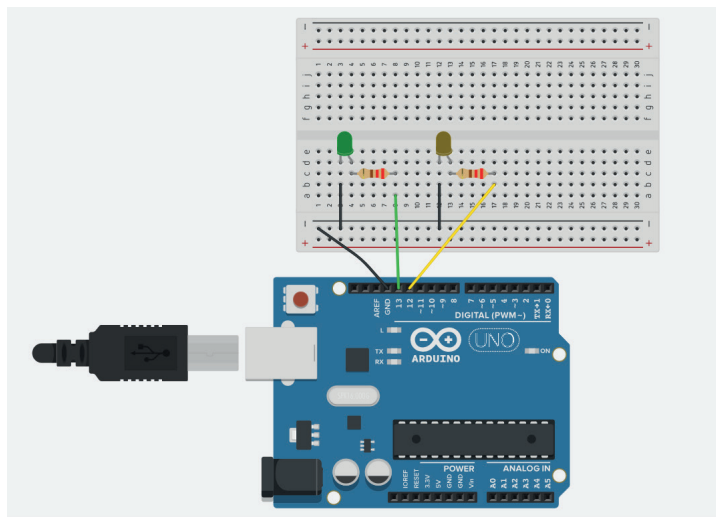
Uma vez montado o circuito e com a programação funcionando é hora de explorar um pouco as possibilidades deste arranjo. Altere os valores encontrados nas funções **delay**, lembrando que o valor descrito é o tempo da pausa em milissegundos e que 1000 milissegundos é igual a um segundo. Uma possibilidade é diminuir este valor e verificar se ainda consegue perceber que o LED está piscando. Quando o valor da pausa é muito baixo, o olho humano não percebe o LED acendendo e apagando, e parece que ele fica aceso com intensidade mais baixa.

4. Atividade 1: Blink do LED



Agora é hora de complexificar um pouco. Que tal adicionar mais um **LED** e fazer com que o arranjo pisque de **forma alternada**? Ou seja, enquanto um **LED** fica apagado o outro acende e vice e versa, sem deixar os dois acesos ou apagados ao mesmo tempo. Para isso, monte o **circuito** do segundo **LED** usando como exemplo o primeiro, e o mesmo procedimento para o código. Todas as informações já estão disponíveis na **Atividade 1**, agora é o caso de repetir o processo para o segundo **LED**.

Seguem abaixo os gabaritos tanto para o circuito como para o código. Lembre que os riscos azuis e pretos representam os jumpers:



Fonte: Tinkercad

Imagem 28: Circuito de 2 leds pinos 12 e 13

Código:

```
void setup(){
  pinMode(13, OUTPUT);
  pinMode(12,OUTPUT);
}

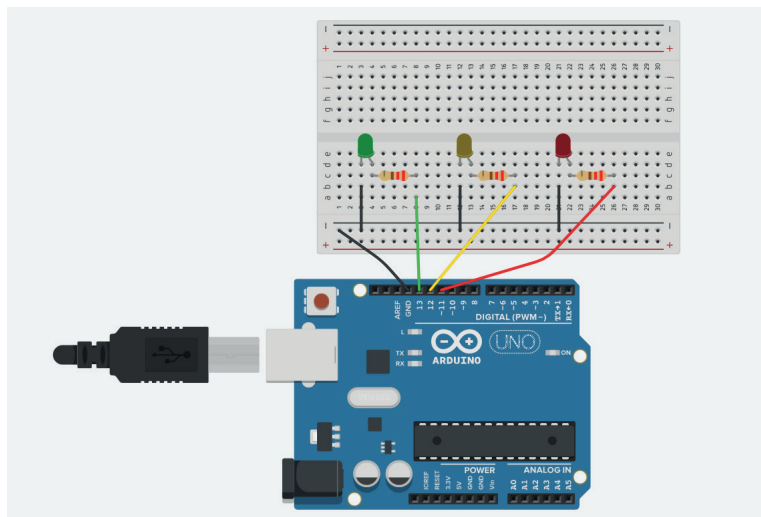
void loop(){
  digitalWrite(13,1);
  digitalWrite(12,0);
  delay(1000);
  digitalWrite(13,0);
  digitalWrite(12,1);
  delay(1000);
}
```

5. Atividade 3: Construa e programe um semáforo



Como desafio para esta atividade, construa e programe um semáforo que se comporte conforme os semáforos que temos nas ruas.

Esta etapa funciona como validação dos conceitos vistos até agora, uma vez que já aprendemos a programar um LED sozinho e dois LEDs ao mesmo tempo. Abaixo segue uma forma possível de se construir e programar o projeto do semáforo:



Fonte: Tinkercad

Imagem 29: Circuito de 3 leds pinos 11, 12 e 13

Código:

```
void setup(){
    pinMode(13, OUTPUT);
    pinMode(12,OUTPUT);
    pinMode(11, OUTPUT);
}
```

```
void loop(){
    digitalWrite(13,1);
    digitalWrite(12,0);
    digitalWrite(11,0);
    delay(1000);
    digitalWrite(13,0);
    digitalWrite(12,1);
    digitalWrite(11,0);
    delay(1000);
    digitalWrite(13,0);
    digitalWrite(12,0);
    digitalWrite(11,1);
    delay(1000);
}
```

6. Atividade Opcional: Desafio das cores



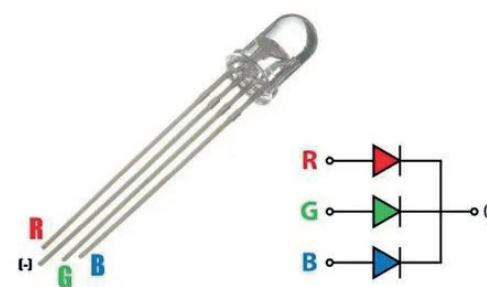
Uma possibilidade de exploração vai em direção ao estudo das cores luz, que são diferentes das cores pigmentos, aquelas das tintas. Nas cores luz, temos três cores primárias que, se combinadas em diferentes proporções, podem fazer praticamente qualquer outra cor. Como exemplo, temos o diagrama de cores mostrando como **vermelho**, **verde** e **azul** combinados podem gerar cores diferentes.

Para trabalhar com as cores e o **Arduino**, podemos utilizar o **LED** chamado **LED RGB com cátodo comum**. Isso quer dizer que dentro desse **LED** de quatro pernas existem na verdade **3 LEDs** diferentes ao mesmo tempo, com as cores **vermelho**, **verde** e **azul**. Note que ele possui três terminais positivos, um para cada cor, e um negativo comum (cátodo comum).



Fonte: Google

Imagem 30: Diagrama de cores



Fonte: Google

Imagem 31: Led RGB catodo comum

Para testar muitas cores diferentes, experimente utilizar a função:

`analogWrite(pino, intensidade);`

Onde a porta é a porta do Arduino no qual o terminal está ligado e a intensidade varia de 0 a 255 como equivalente a 0% a 100%. Segue abaixo o circuito para montar:

Por exemplo:

Acender a cor vermelha com intensidade máxima:

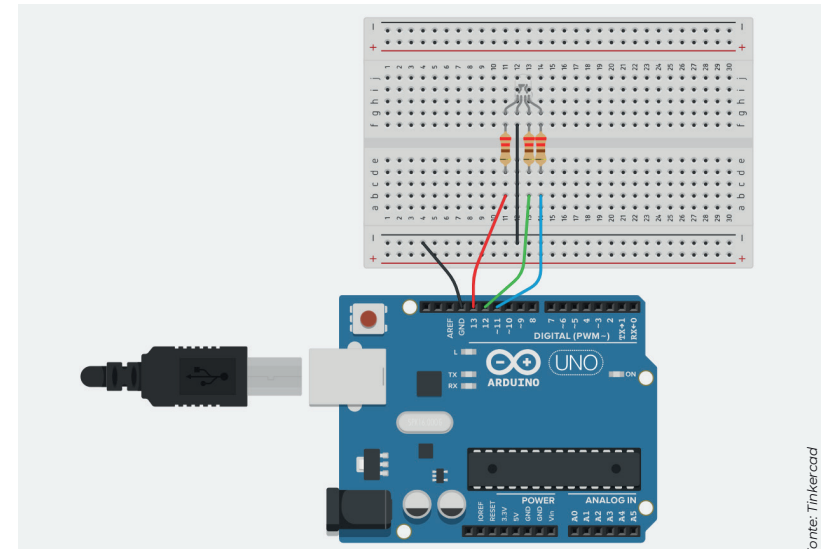
`analogWrite(13,255);`

Acender a cor vermelha com intensidade média:

`analogWrite(13,125);`

Apagar a cor vermelha com intensidade igual a zero:

`analogWrite(13,0);`



Fonte: TinkerCAD

Imagem 32: Circuito led RGB

7. O uso da laborador no programa “Maker Space IoT para estudantes de escolas públicas Paulistas”



A placa laborador, juntamente com teclado, mouse e display, foi distribuída às 10 escolas participantes do programa fase 2 com a intenção inicial de possuírem um mini computador no espaço destinado ao ambiente maker.

Ao final do programa ouvimos alguns professores



Saiba mais do programa Maker Space IoT em <https://makerspaceiot.febrace.org.br/> e visite nossas publicações em <https://febrace.org.br/acervo/outros/colecao-maker-space-iot/>



A placa Labrador é, na verdade, um mini PC, pois é possível instalar o Sistema Operacional Debian e uma diversidade enorme de aplicativos. Foi devido a essa funcionalidade que nos permitiu ensinar programação para uma turma de ensino fundamental 1 (de 9 a 10 anos) utilizando o *Scratch*.

Possui vários periféricos como placa de vídeo, placa de som, saídas USB e outros. No entanto, o periférico mais importante para nosso trabalho, foi a *GPIO*. A *GPIO* possui várias saídas e entradas digitais, que foram essenciais para o desenvolvimento de nosso projeto, que era controlar de forma remota os movimentos de uma luneta.



Fonte: EE Frederico Brotero

O uso da labrador na EE Frederico Brotero



Fonte: EE Frederico Brotero

O uso da labrador na EE Frederico Brotero

Para utilizar algumas saídas digitais escrevemos um código em C com a biblioteca *wiringPi.h*, o que possibilitou simular um *joystick* em uma interface simples na linha de comando, mas que conseguiu mover a luneta tanto no azimute como na altura.

Foi um grande instrumento de aprendizagem para nossos alunos e tem o potencial de se transformar na Raspberry Pi brasileira, desde que melhore a sua comercialização.

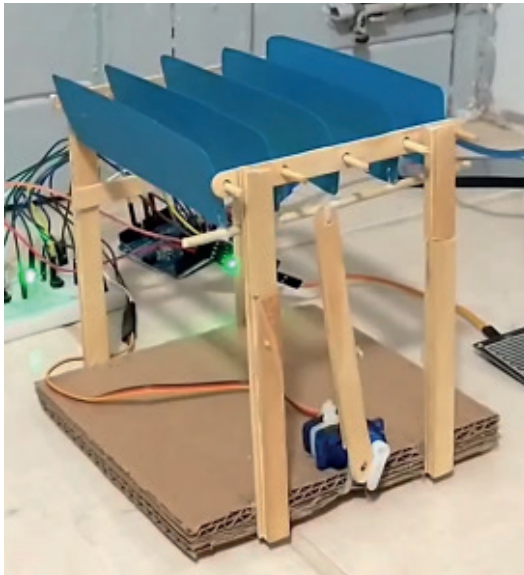
Ubiratan de Campos

Prof. de Física da EE Frederico Brotero - Guarulhos

Aqui na escola usamos a placa labrador para programação e como mini computador para realização de pesquisas. Assim os alunos desenvolveram o protótipo do projeto cobertura retrátil automatizada com Arduino por exemplo

O equipamento está disponível para os estudantes e professores que queiram utilizar no espaço Maker da escola, mas acredito que seria interessante uma formação técnica para aperfeiçoarmos seu uso.

Bruno Ruivo de Oliveira
Professor de Química da EE Lino Vieira Ruivo - Ibiúna



Cobertura retrátil automatizada



O uso da labrador na EE Lino Vieira Ruivo



Fonte: EE Lino Vieira Ruivo



Fonte: ETEC Guarulhos

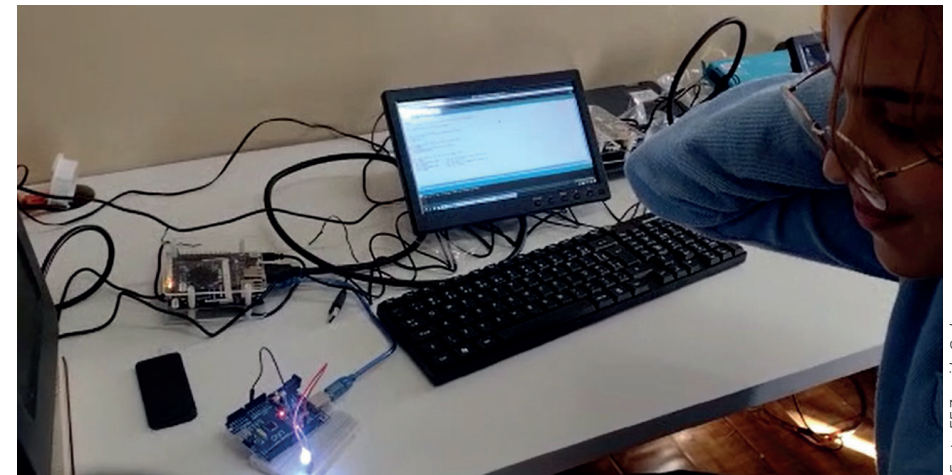
O uso da labrador na ETEC Guarulhos

Para um dos projetos da escola, o Compostura - uma composteira onde se pode acompanhar o processo de compostagem via internet, a ideia é que seja controlada via arduíno ou labrador. Com a labrador tentamos a troca do sistema operacional e a instalação de algumas bibliotecas, não conseguimos, ou travava ou não funcionava mais. Ainda não deu certo, mas continuamos na luta.

Marcel Thomé Filho
Prof. orientador de TCC e sistemas embarcados da
ETEC Guarulhos

Começamos a usar a labrador com os alunos inscritos para trabalhar programação no espaço maker. No início tiveram dificuldades por estarem acostumados com a interface Windows que facilita, já com a labrador tivemos que digitar comandos. Usamos muito o guia que vocês encaminharam junto com o apoio técnico. Foi bem interessante e desafiador e quando você vê o final, o resultado que conseguiram atingir, observo nos olhos deles aquela sensação de vitória, de superação e isso é bastante gratificante.

Vicente Lopes da Silva Neto
Prof. de física e matemática da EE Zenaide Godoy –
São Paulo



Fonte: EE Zenaide Godoy

Prof. Vicente com a aluna Amanda de Albuquerque Lima

