Arara

Ensinando os primeiros passos para programar sua placa Arara.

Instalação

Neste capítulo será descrito como fazer a instalação completa do software utilizado pela Arara.

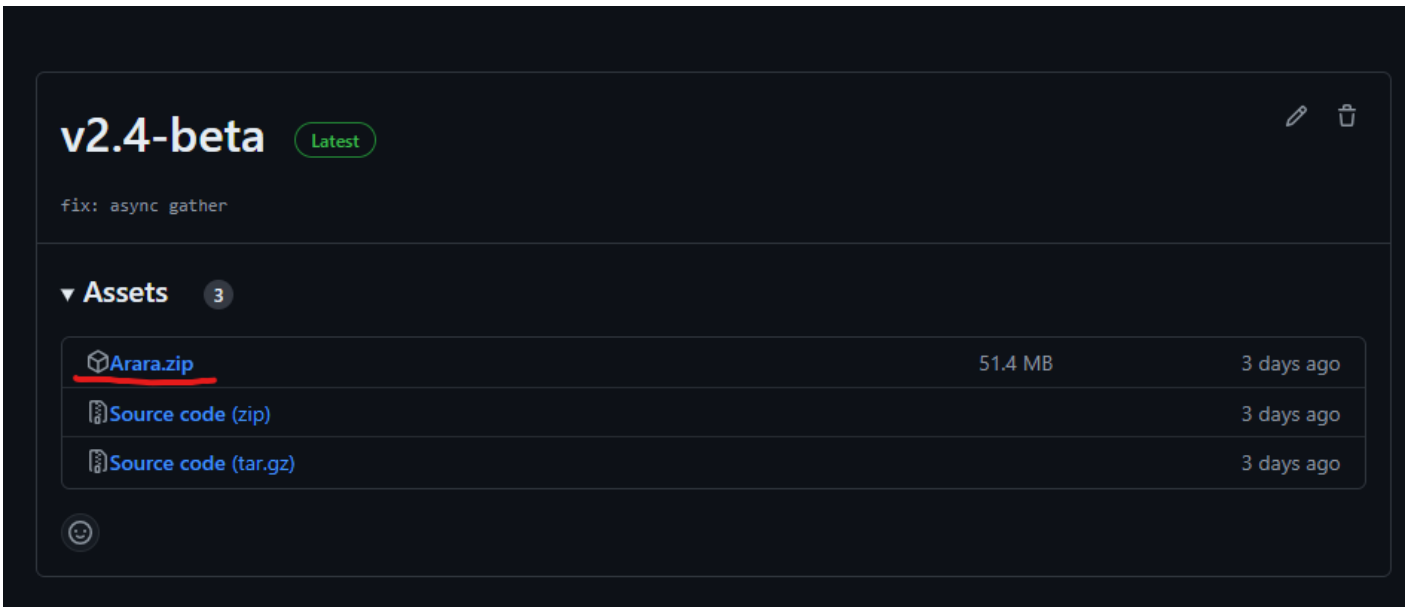
Arara Driver Station

O programa Arara Driver Station é o software que faz a conexão wireless entre a placa (Arara) e o computador, dessa forma obtendo os valores do controle conectado via USB.

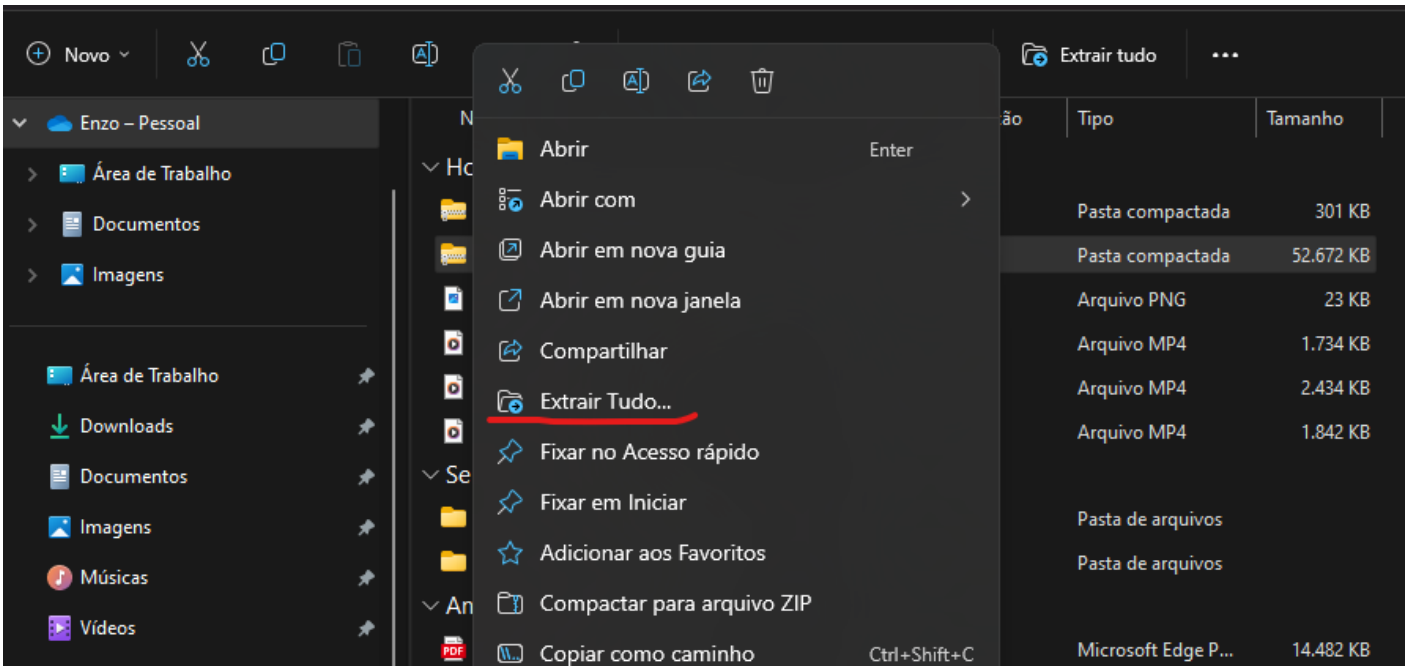
Agora, para começar o processo de instalação, entre no seguinte endereço:

Arara Driver Station

Ao abrir o *link* você será direcionado para uma página no github na qual é possível observar os lançamentos do software, baixe o mais recente clicando em Arara.zip, como indicado abaixo.



Após clicar, iniciará o *download* de uma pasta .zip, aguarde o término dela. Por fim, extrai-a em uma pasta desejada, como indicado abaixo.



Ao clicar em "Extrair Tudo..." é necessário indicar um caminho (local onde a pasta será armazenada), escolha um onde ficará organizado.

> Teste o executável abrindo-o em um primeiro momento

Arduino IDE

Nesta página estaremos ensinando como baixar a Arduino IDE, assim como as depêndencias da Arara.

Para começar, entre no seguinte endereço: Arduino IDE

Em "*Download options*" escolha a que for compatível com seu computador, caso seja windows eu recomendo a primeira opção **"Win 10 and newer, 64 bits"**, dessa forma é feito o *download* de um executável que fará a instalação da Arduino IDE. Potanto, apenas abra o executável e siga as etapas de instalação que serão mostradas.

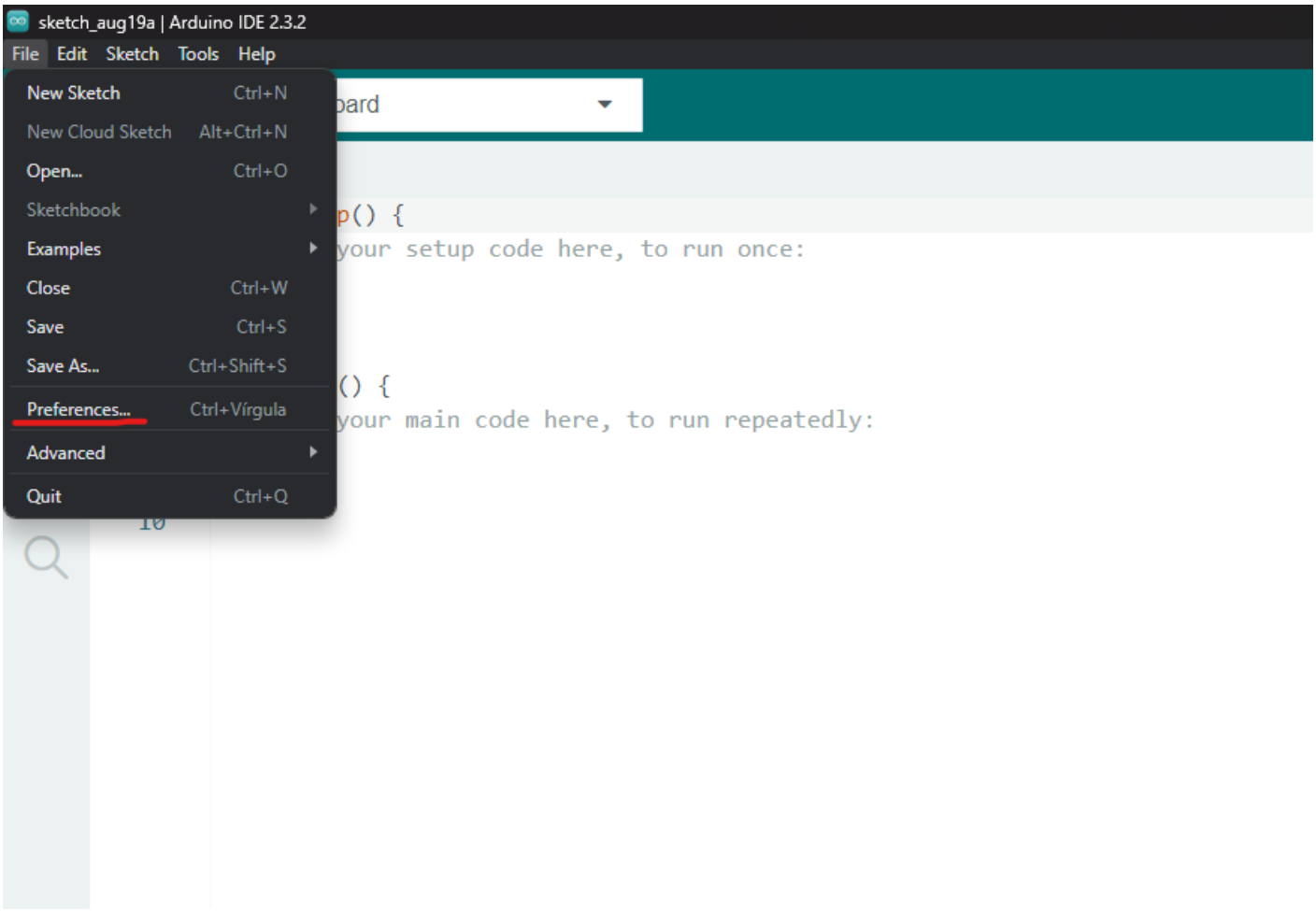
> Pode ser necessário dar permissão na hora de instalar a Arduino IDE, nesses casos apenas clique em "Permitir" ou "Sim"

Dependências de placa

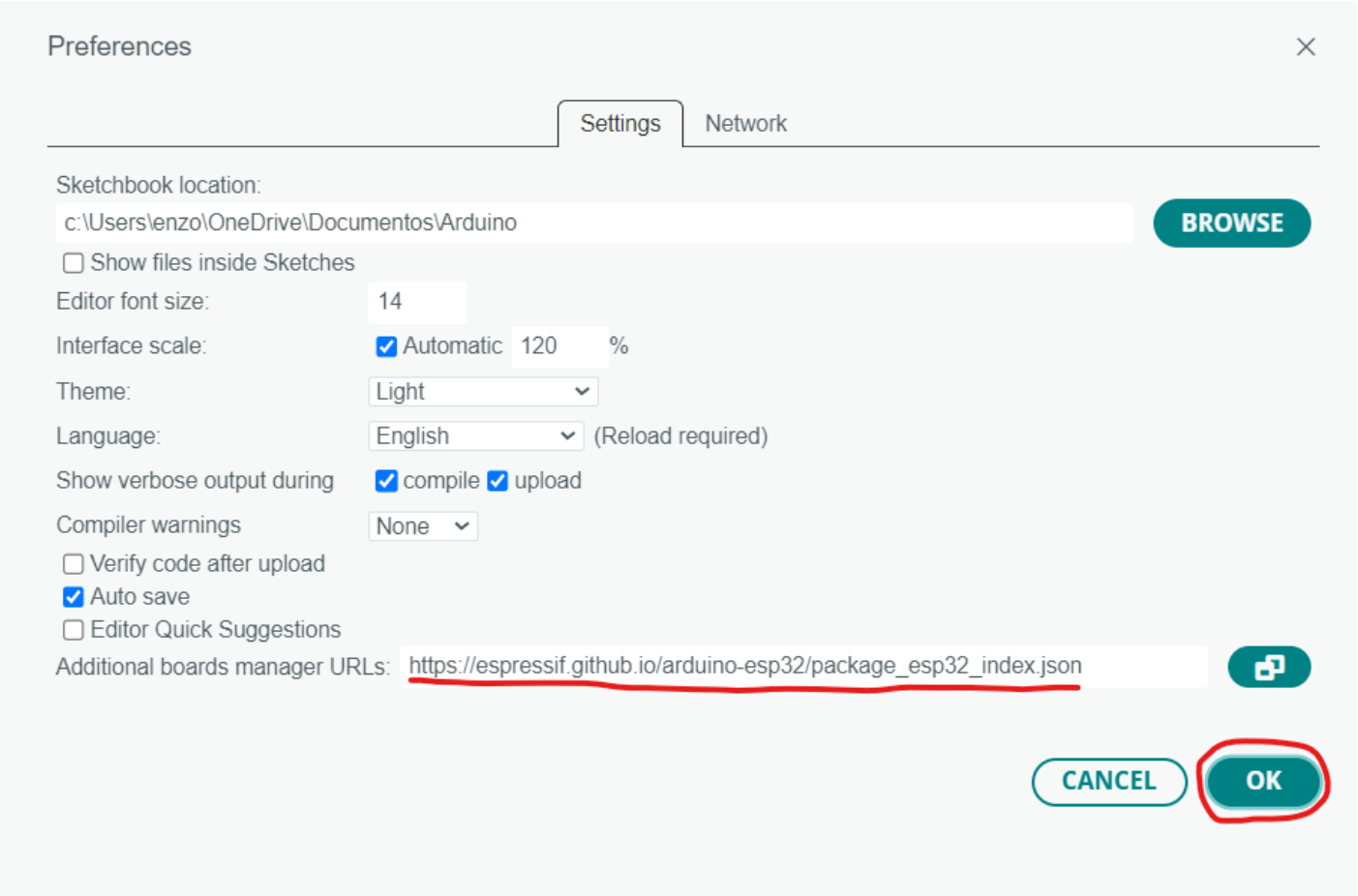
O Arduino IDE é capaz de compilar e fazer *upload* dos códigos escritos em sua interface, para tal, é necessário que ele faça o reconhecimento da placa que está conectado ao seu computador pela porta USB. Como a placa Arara não tem reconhecimento inato pela IDE, é necessário que seja feito a instalação das placas compatíveis.

Portanto, comece copiando o seguinte *link*: https://espressif.github.io/arduino-esp32/package_esp32_index.json

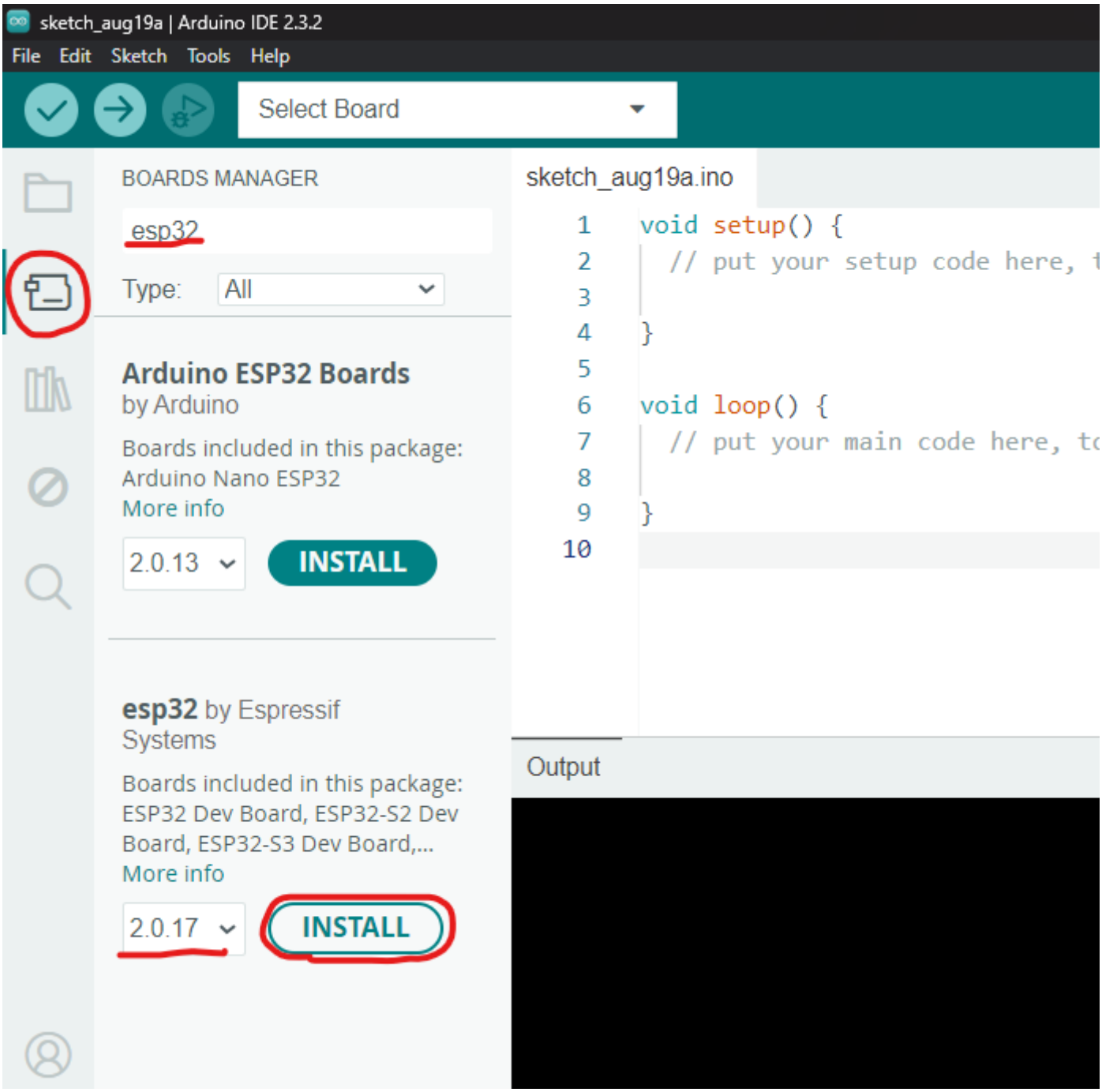
Agora, abra sua Arduino IDE instalada anteriormente. Com ela aberta, clique Ctrl + , para abrir a janela de preferências. Ou siga as etapas abaixo.



No fim, você deve parar em uma janela semelhante a essa abaixo. Agora, coloque o *link* copiado anteriormente no local indicado e pressione *OK*



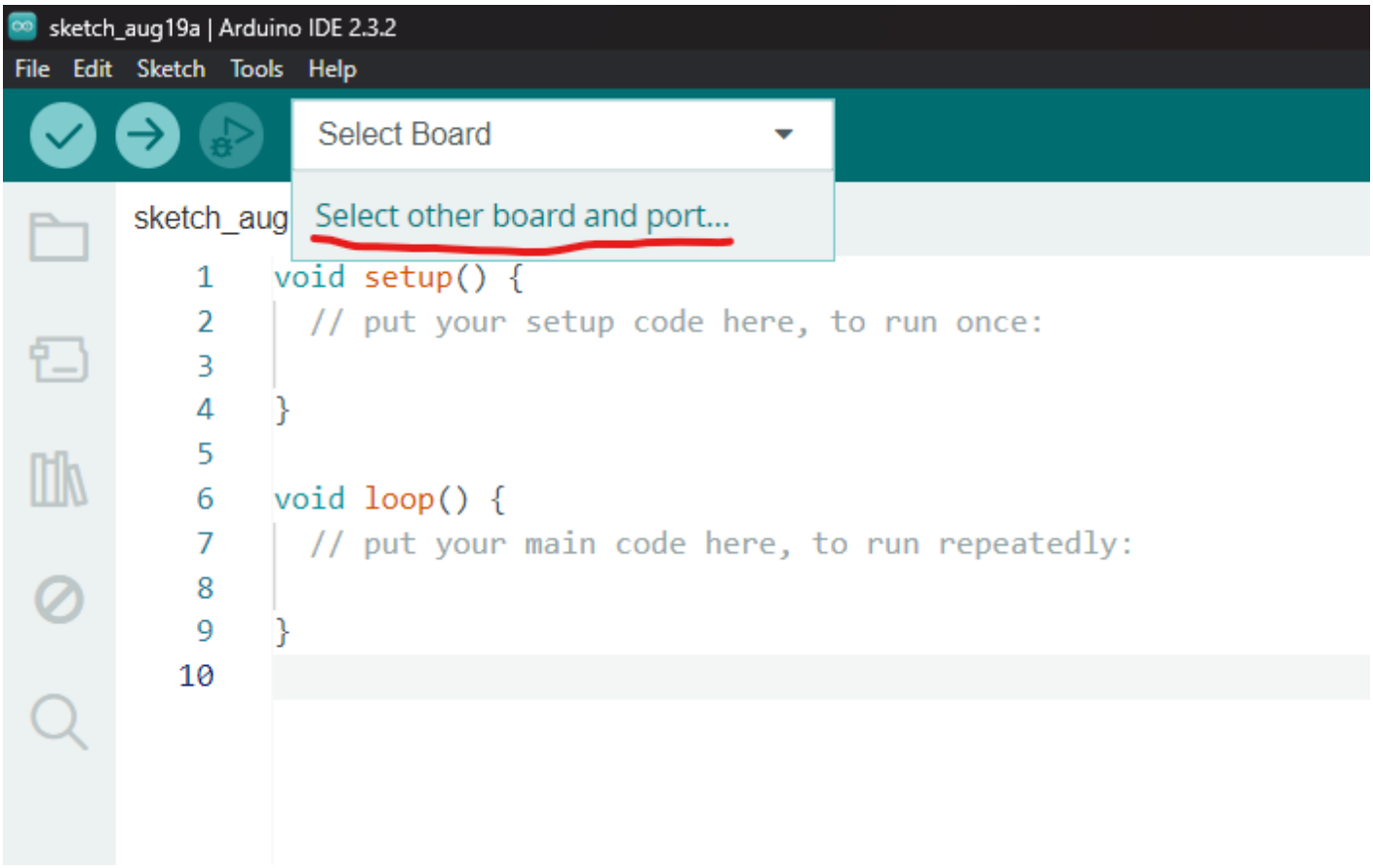
Agora, abra a aba de *boards manager* e pesquise por ***esp32***, após encontrar as placas esp32, escolha a versão **2.0.17** e clique em **Install**. Como indicado abaixo.



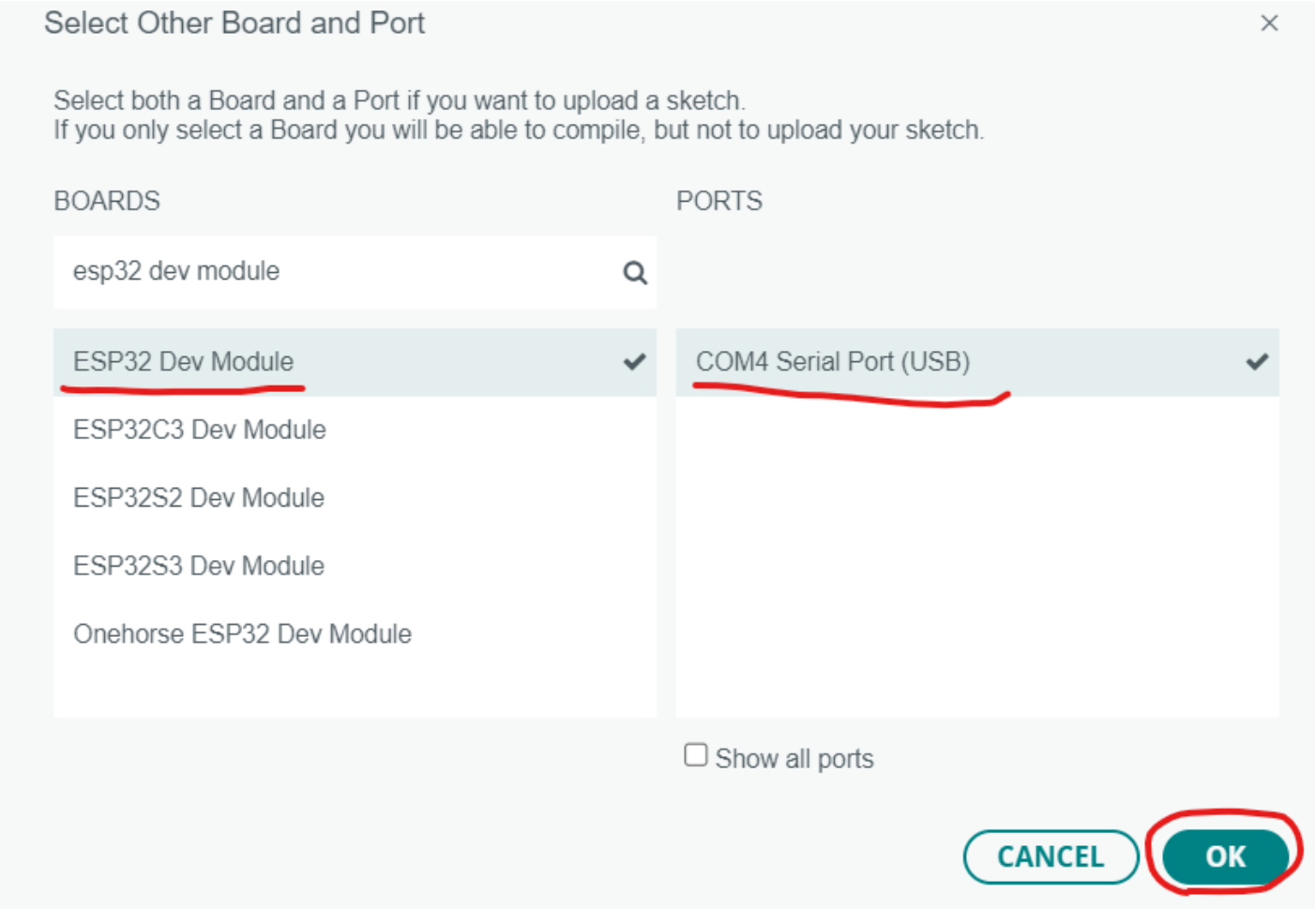
Aguarde o término da instalação, pode demorar um certo tempo.

Selecionando a placa

Agora que a interface de programação foi instalada, conecte a Arara ao computador e clique em ***Select Board*** na Arduino IDE. Como mostrado abaixo.



Agora, escolha **ESP32 Dev Module** e a porta **COM** a qual a placa foi conectada. Como mostrado abaixo.



Caso esteja tudo certo, a aba de placa deve ficar da seguinte forma.

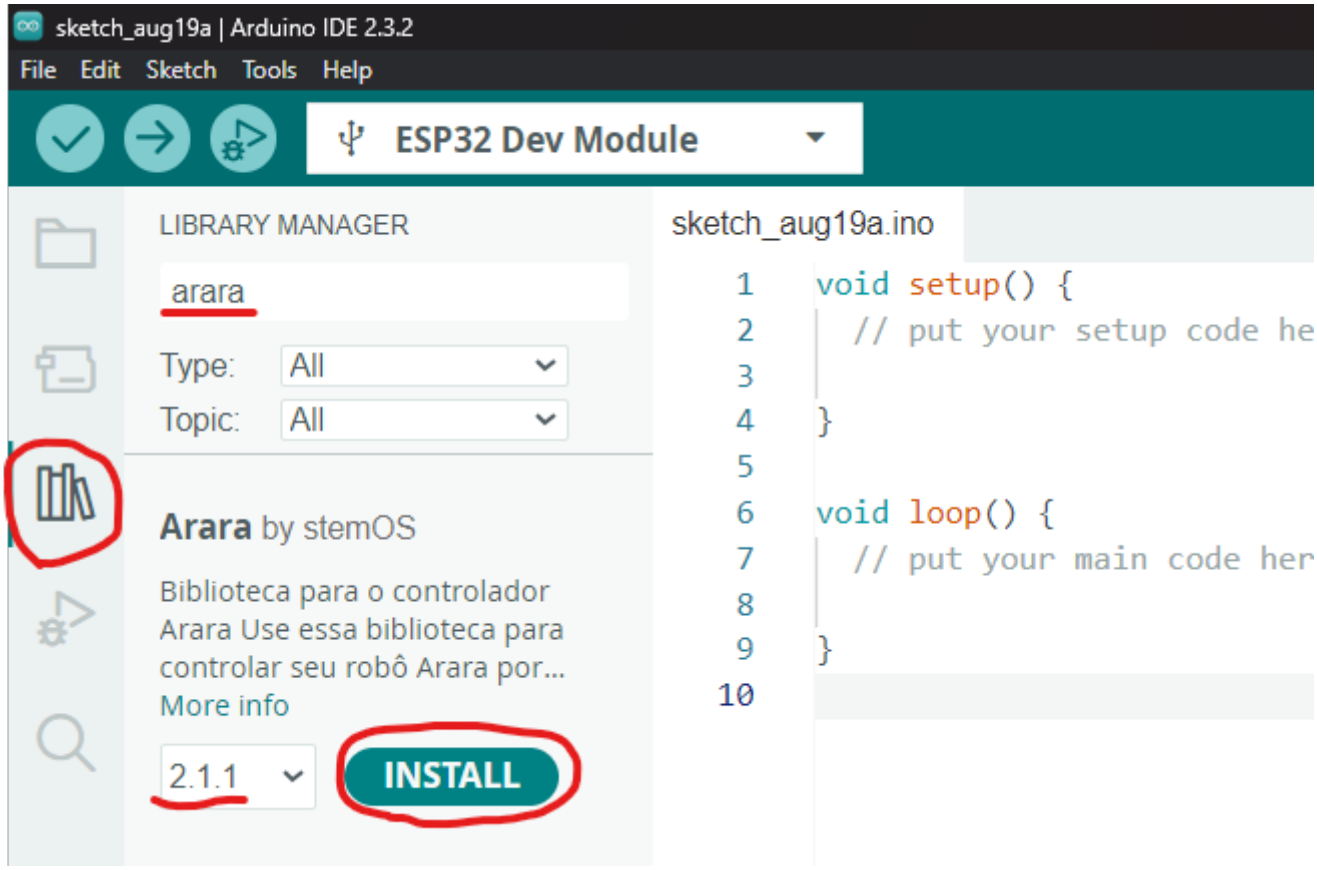
> Parabéns, você terminou a parte mais complicada do processo de instalação, agora siga para a próxima parte para instalarmos a biblioteca da Arara!

Biblioteca Arara

Seja bem vindo a última parte do processo de instalação, mas também a mais fácil caso os processos anteriores tenham sido feitos corretamente.

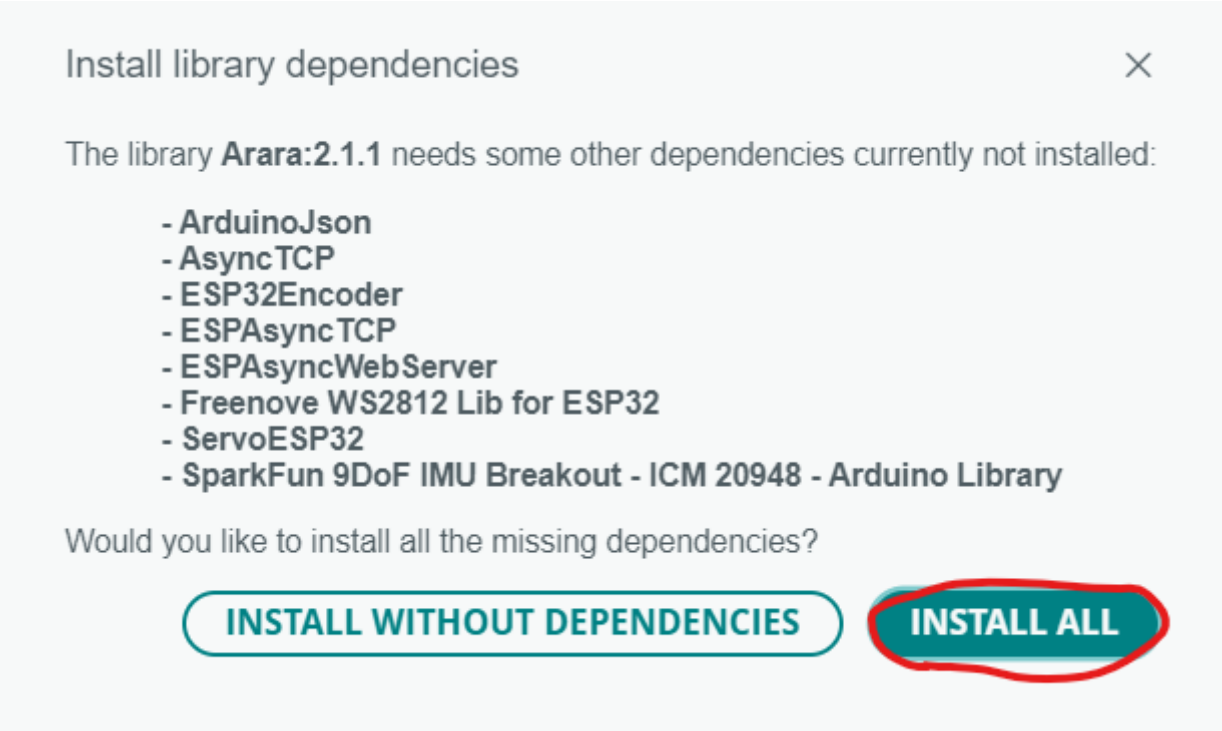
A biblioteca da Arara é aquela que faz com que sejamos capazes de movimentar motores, servos e utilizar sensores, portanto é ela quem faz o controle do *hardware* da placa. Com isso, ela é de suma importância.

Para começar a instalação, abra o arduino IDE e siga o processo indicado abaixo.



> Escolha sempre a versão mais recente de preferência, no meu caso, ao escrever essa página era 2.1.1, mas caso para você apareça uma versão maior, selecione ela.

Ao clicar no botão **Install** mostado anteriormente, será perguntado sobre as dependências da biblioteca, isto é, outras bibliotecas utilizadas pela biblioteca principal da Arara, clique em **Install All**



Aguarde o término da instalação da biblioteca.

> Parabéns, caso tenha seguido todos os processos anteriores corretamente, você terminou toda instalação do software da Arara, dessa forma podemos finalmente começar a programa-lá!

Programação

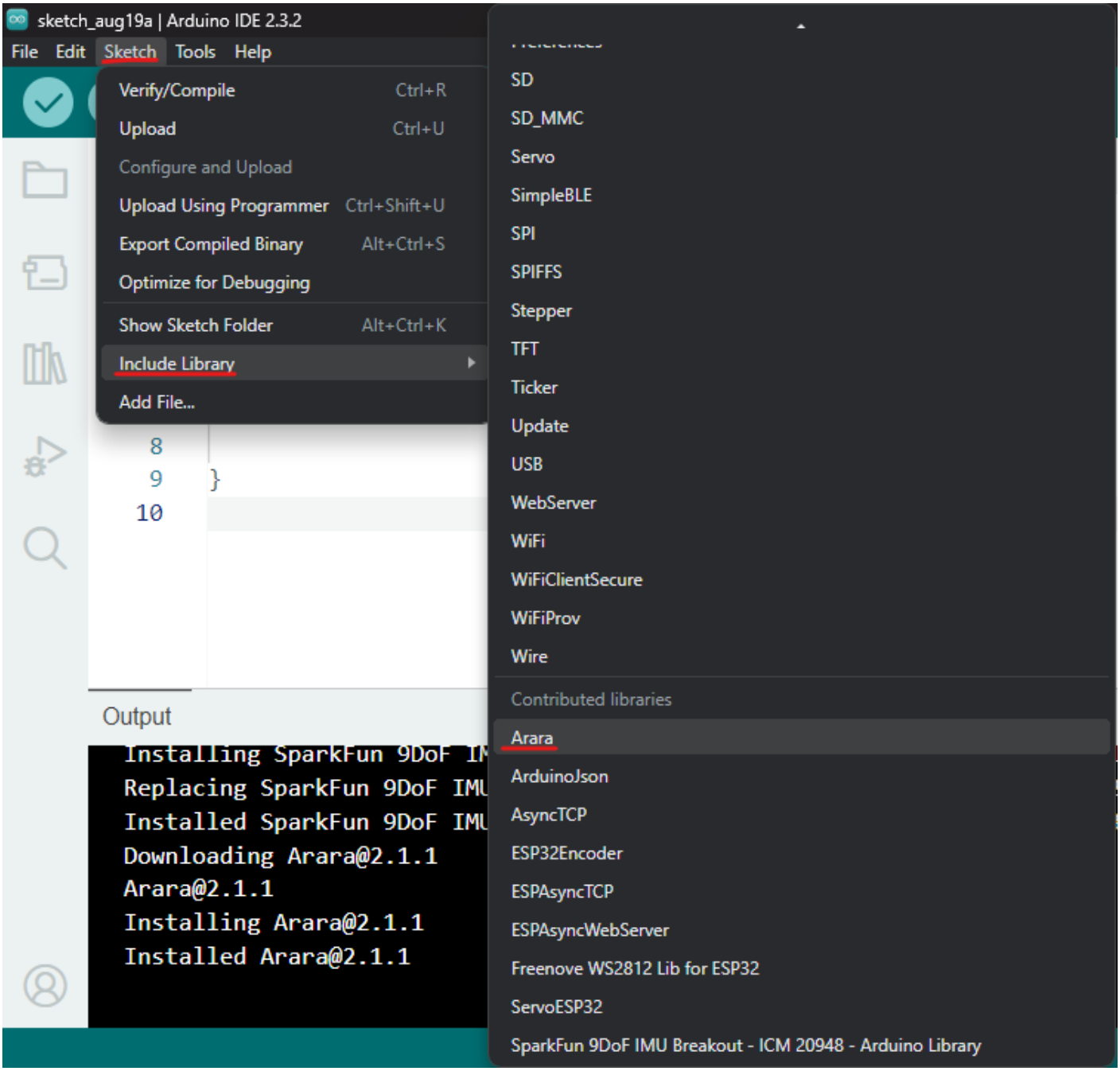
Neste capítulo veremos como podemos utilizar o hardware disponível pela Arara

Programação

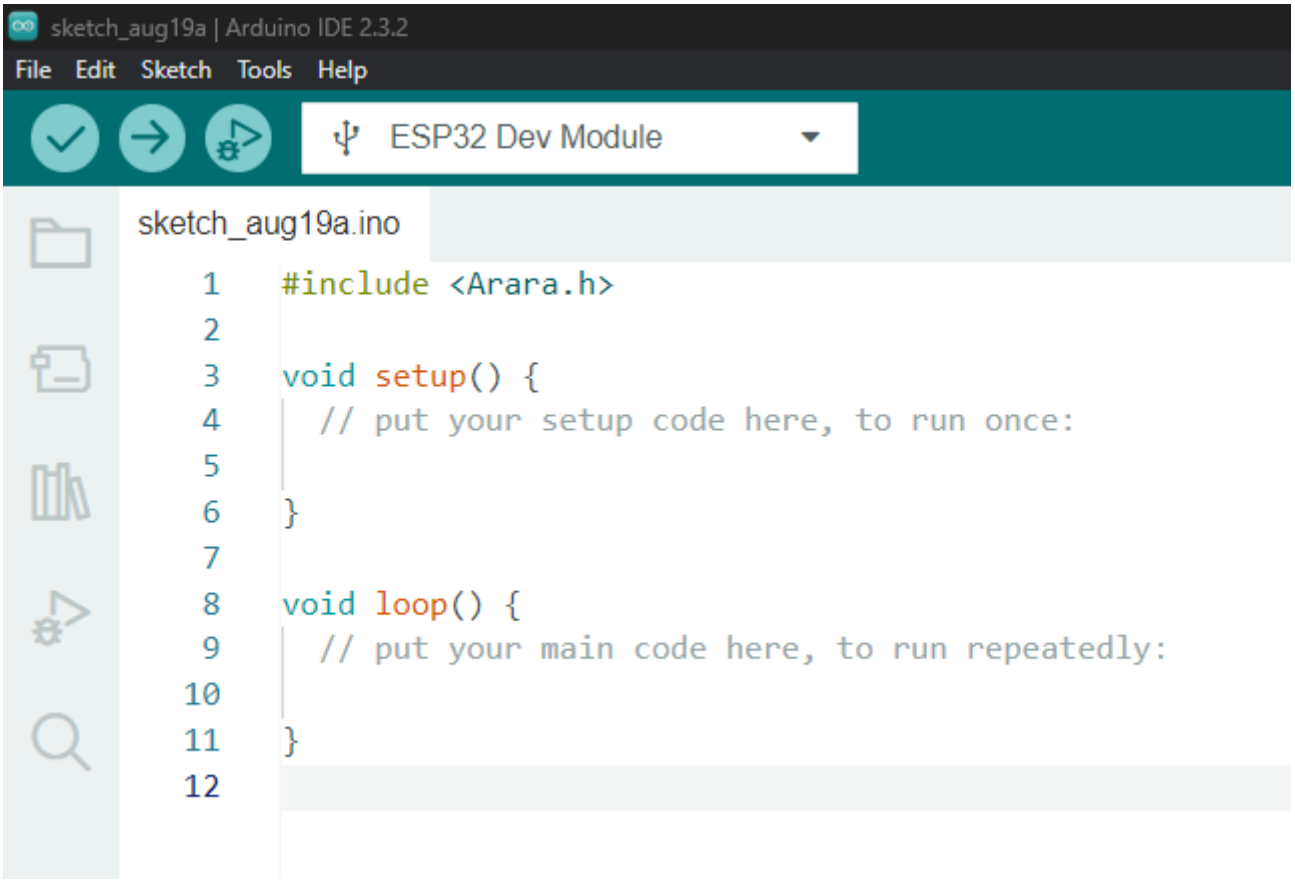
Importação da biblioteca

Esta página pode ser considerada como introdutória a todas as outras, pois devemos realizar esse processo antes de começar a programar qualquer esboço (nome de um projeto na Arduino IDE).

O processo dito anteriormente, é a importação da biblioteca, ou seja, utiliza-lá em nosso código. Para fazer isso é siga os processos abaixo.



Seu esboço deve ficar da seguinte maneira.



> Você também pode escrever diretamente essa linha #include, é ela quem faz a importação da biblioteca ao projeto

Motores

Para programar os motores é bem simples, afinal, eles já estão declarados na biblioteca com suas portas correspondentes.

Portanto, podemos apenas fazer o seguinte para acionar um motor.

```
#include <Arara.h>

void setup() {
  // put your setup code here, to run once:
  motor1.setPower(0.5);
}

void loop() {
  // put your main code here, to run  repeatedly:

}
```

Esse código fará com que o motor opere a, aproximadamente, 50% de sua velocidade máxima. E é claro que como dito anteriormente, temos um objeto correspondente para cada porta, como indicado a seguir.

> Com esse parágrafo anterior é importante entender o seguinte, o parâmetro setPower só aceita valores entre -1.0 até 1.0, então 1.0 para 100%, vale dizer que é o mesmo para valores negativos, mas agora o motor irá girar em outra direção.

```
#include <Arara.h>

void setup() {
  // put your setup code here, to run once:
  motor1.setPower(0.5);
  motor2.setPower(1.0);
  motor3.setPower(0.0);
  motor4.setPower(-0.5);
}
```

```
void loop() {
  // put your main code here, to run  repeatedly:


}
```

Tente nesse momento entender o que cada motor fará.

> Resposta:
>
> Motor da porta 1: operando a 50% de sua velocidade para "frente";
>
> Motor da porta 2: operando a 100% de sua velocidade para "frente";
>
> Motor da porta 3: parado;
>
> Motor da porta 4: operando a 50% de sua velocidade para "trás".

Encoder

De forma breve, um encoder é um sensor digital que tem como função medir a posição/velocidade de um motor.

Em nossa biblioteca ele já é declarado como objeto inerente do motor, portanto, para utiliza-lo podemos fazer o seguinte.

```
#include <Arara.h>

void setup() {
  // put your setup code here, to run once:
  motor1.setPower(0.5);
}

void loop() {
  // put your main code here, to run  repeatedly:
  double position = motor1.encoder.getPosition();
}
```

> Agora essa variável position ficará armazenando o valor do encoder repetidamente

Podemos verificar seu valor imprimindo-a no monitor serial. Da seguinte forma.

```
#include <Arara.h>

void setup() {
  // put your setup code here, to run once:
  motor1.setPower(0.5);
}

void loop() {
  // put your main code here, to run  repeatedly:
  double position = motor1.encoder.getPosition();
  Serial.print("Encoder: ");
```

```
  Serial.println(position);
}
```

Servos

Um servo também é um acionador, como um motor, a diferença é que seu movimento tem como intenção ser mais preciso, de forma que podemos dizer para qual ângulo de seu escopo ele pode se mover.

Então, para movimentar ele para uma posição específica, podemos fazer da seguinte forma.

```
#include <Arara.h>

void setup() {
  // put your setup code here, to run once:
  servo1.setPosition(270);
}

void loop() {
  // put your main code here, to run  repeatedly:

}
```

Nesse código movimentamos o servo da porta para a posição de 270º de seu escopo (no caso do servo da REV Robotics esse é considerado o limite).

> Importante adicionar que é a mesma ideia de programar um motor, já existem os 3 servos declarados na biblioteca, podemos apenas chama-los no código.

```
#include <Arara.h>

void setup() {
  // put your setup code here, to run once:
  servo1.setPosition(270);
  servo2.setPosition(0);
  servo3.setPosition(180);
}

void loop() {
```

```
  // put your main code here, to run  repeatedly:


}
```

Sensores digitais

Um sensor digital é um dispositivo que mede apenas dois valores, verdadeiro ou falso.

Para utiliza-lo no código o seguinte pode ser feito.

```
#include <Arara.h>

Digital di1(PortasDigitais::PORTA_1);

void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run  repeatedly:
  bool valorDigital = di1.getInput();
}
```

> Observe que diferente dos outros dispositivos vistos anteriormente, a porta digital deve ser declarada

Também podemos imprimir seu valor no monitor serial.

```
#include <Arara.h>

Digital di1(PortasDigitais::PORTA_1);

void setup() {
  // put your setup code here, to run once:

}
```

```
void loop() {
  // put your main code here, to run  repeatedly:
  bool valorDigital = di1.getInput();
  Serial.println(valorDigital);
}
```

Programação

IMU

IMU é um dispositivo, portanto, um conjunto de componentes que tem como intenção final indicar a posição/velocidade angular de um mecanismo. Seu nome pode ser traduzido como dispositivo de medição inercial.

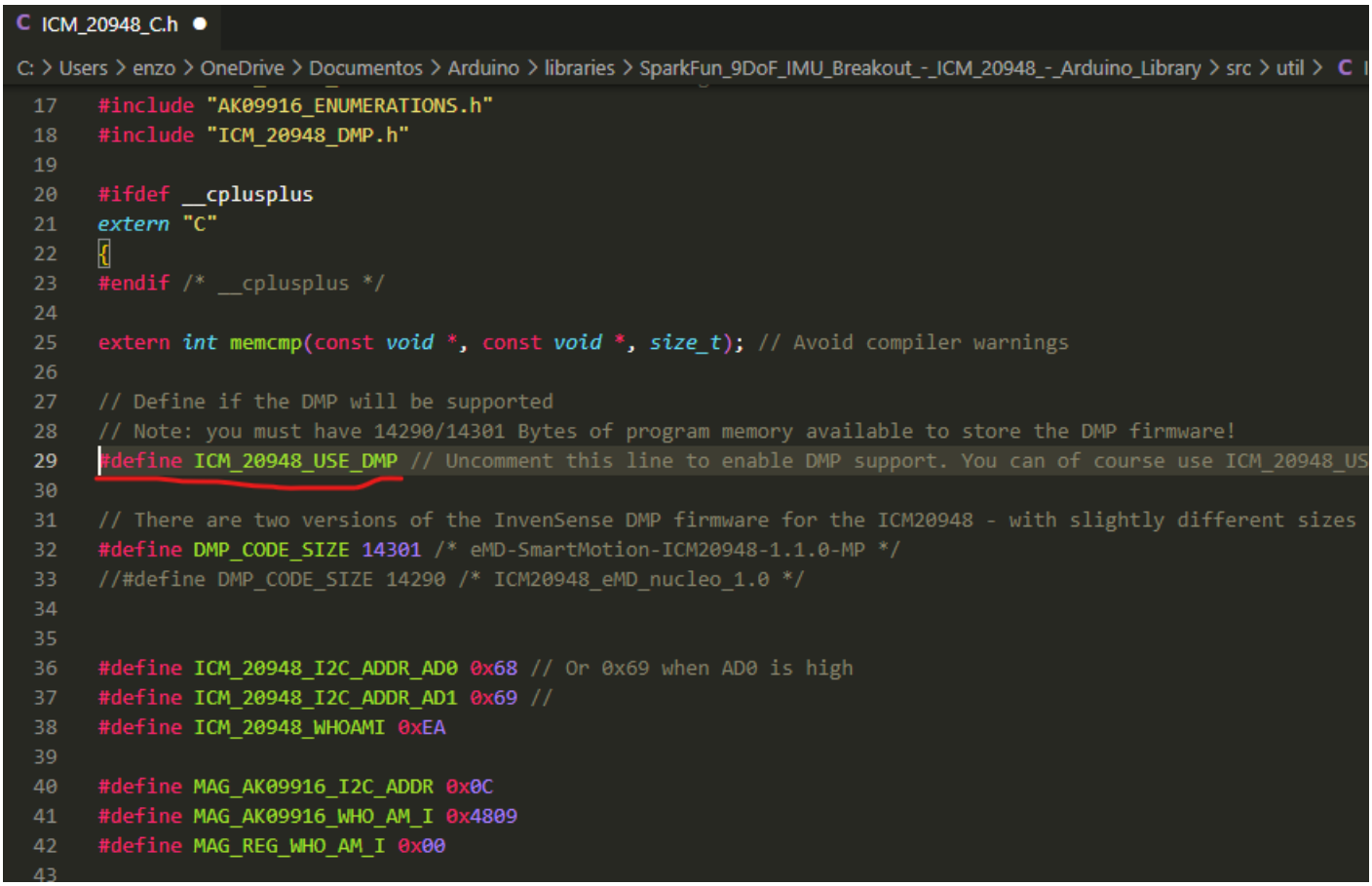
Para utiliza-lo é necessário editar um arquivo da biblioteca do Arduino IDE.

Faça o seguinte caminho em seu computador:

C:\Users\"User"\Documentos\Arduino\libraries\SparkFun_9DoF_IMU_Breakout_-*ICM_20948*-_Arduino_Library\src\util

Abra o arquivo ICM_20948_C.h, caso ele peça algum aplicativo para abrir, use o editor de texto do windows.

Por fim, descomente (remova os caracteres //) da seguinte linha.



Agora em nosso código podemos fazer o seguinte.

```
#include <Arara.h>

IMU imu;

void setup() {
  // put your setup code here, to run once:
  imu.init();
}

void loop() {
  // put your main code here, to run  repeatedly:
  Serial.print("Pitch: ");
  Serial.println(imu.getPitch());
  Serial.print("Roll: ");
  Serial.println(imu.getRoll());
  Serial.print("Yaw: ");
  Serial.println(imu.getYaw());
}
```

Gamepad

Na primeira parte desse livro foi ensinado como instalar a Arara Driver Station, software utilizado para fazer a comunicação wireless entre a Arara e o computador. Essa comunicação tem como finalidade entregar os valores do gamepad a Arara. Portanto, vamos aprender como se da inicio a essa comunicação pela parte de programação da Arara.

Para começar é preciso entender que a comunicação começa chamando a seguinte linha de código.

```
#include <Arara.h>

void setup() {
  Arara.start();
}

void loop() {

}
```

Essa função **start()** faz com que o Wi-Fi da placa seja iniciado, assim como os processos de comunicação.

> Será ensinado como fazer a conexão a placa pela Driver Station em um capítulo posterior desse livro

Agora, nosso objeto de gamepad poderá ter seu valor atualizado quando a driver station for utilizada.

Então, como ele já está declarado, podemos chama-lo da seguinte forma.

```
#include <Arara.h>

void setup() {
  Arara.start();
}
```

```
void loop() {
  double y = gamepad.getLeftAxisY();
  double x = gamepad.getLeftAxisX();
}
```

Esses são os eixos analógicos do controle. Podemos também chamar os botões, como indicado.

```
#include <Arara.h>

void setup() {
  Arara.start();
}

void loop() {
  bool a = gamepad.getButtonA();
  bool b = gamepad.getButtonB();
}
```

> Verifique por si mesmo quais são as funções disponíveis

Código exemplo

Com essas etapas que seguimos anteriormente já é possível programarmos um motor para ser acionado conforme o gamepad se movimenta. Portanto, nessa página está disponibilizado um código de exemplo para mostrar como isso pode ser feito.

```
#include <Arara.h>

void setup() {
  Arara.start();
}

void loop() {
  motor1.setPower(gamepad.getLeftAxisY());
  motor2.setPower(gamepad.getRightAxisY());
}
```

> Isso pode ser traduzido como um acionamento do estilo Tank

Interface

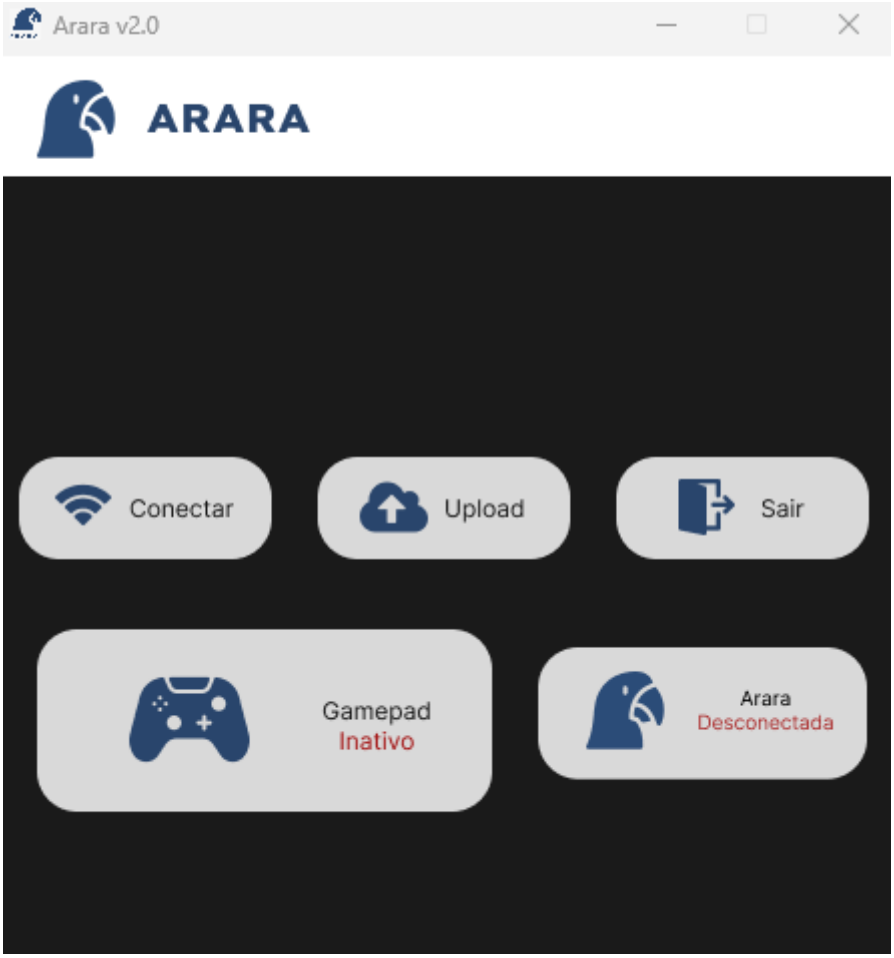
Aqui será abordado como fazer uso da interface de controle instalada anteriormente

Driver Station

Estarei utilizando como exemplo o código de exemplo apresentado no último capítulo. Entretanto, os passos aqui seguidos serão o mesmo para qualquer código que tente utilizar o gamepad.

Para começar, abrindo seu aplicativo ele deve ser semelhante ao mostrado abaixo.

> Aceite caso o aplicativo peça permissão



Descrição do que cada botão/indicador faz:

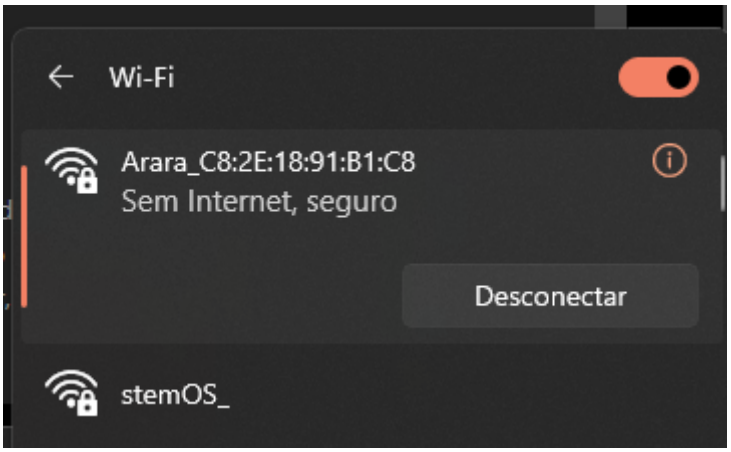
- Conectar: estabelece a comunicação com a placa
  - É necessário que o computador esteja conectado na rede da placa.
- Upload: faz *upload* de códigos já prontos.
- Sair: fecha a página
- Gamepad: indica se o gamepad está ou não conectado ao computador.

- Arara: indica se a placa está conectada ou desconectada.

Conexão

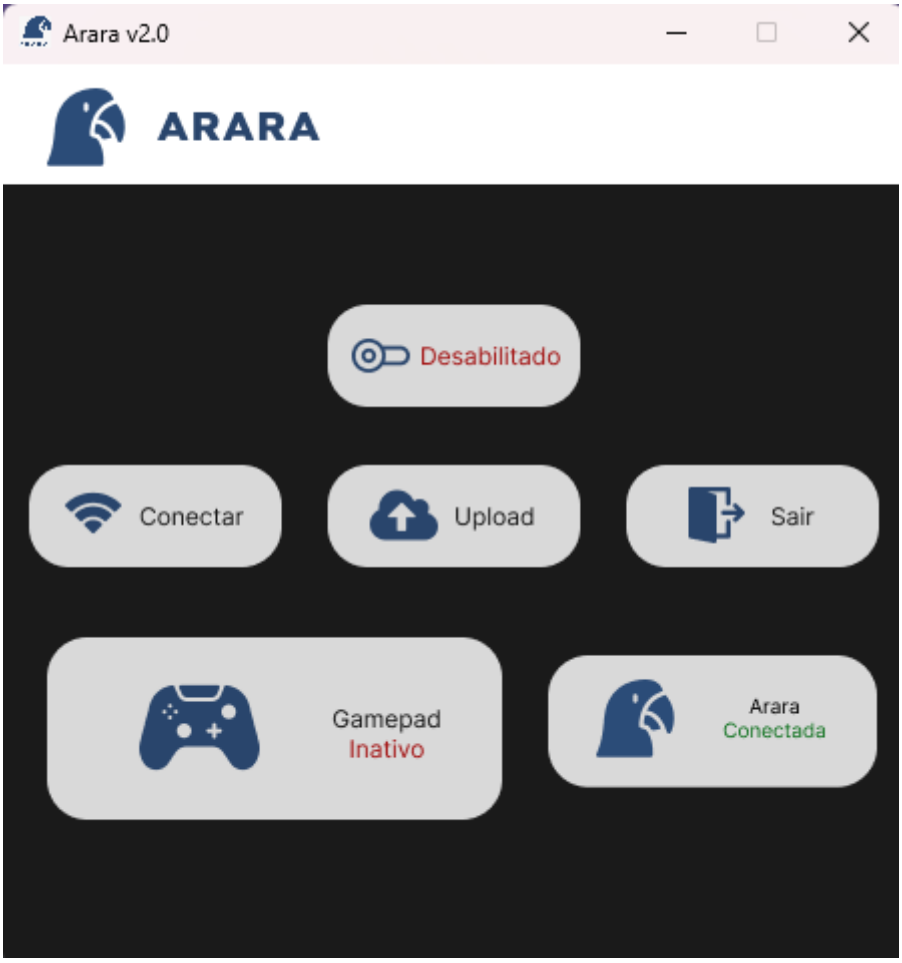
Ligue a placa em 12V, estou considerando que o código utilizado é o último apresentado neste livro. Agora, espere o LED piscar azul e amarelo, isso indicará que ele está pronto para ser conectado, abra a lista de Wi-Fi em seu computador, deve apararecer uma rede com o nome de Arara e alguns caracteres aleatórios, como indicado.

Faça a conexão com essa rede. Caso solicite senha, ela é **password**



Agora, em sua Driver Station, clique no botão "Conectar"

Deve aparecer uma mensagem indicando o sucesso da conexão estabelecida. A sua janela deve ficar da seguinte forma.



Conecte o gamepad ao seu computador via USB e veja o indicado do gamepad ficar ativo também. Com isso, é possível perceber que depois de conectar surge um botão Habilitado/Desabilitado, ele serve para controlar quando as mensagens serão enviadas. Portanto, após conectar o gamepad, clique nesse botão e ele começara a enviar os valores do controle para a placa. Lembre-se de que para utilizar o código de exemplo anterior, a Arara deve ser alimentada em 12V.

> Observe quais indicadores mudam conforme você conecta o gamepad ou aperta o botão desabilitado

Com tudo equipado, movimente o analógico esquerdo para frente, o motor da porta 1 deve acionar, caso o analógico direito seja movido, o motor da porta 2 deve girar.

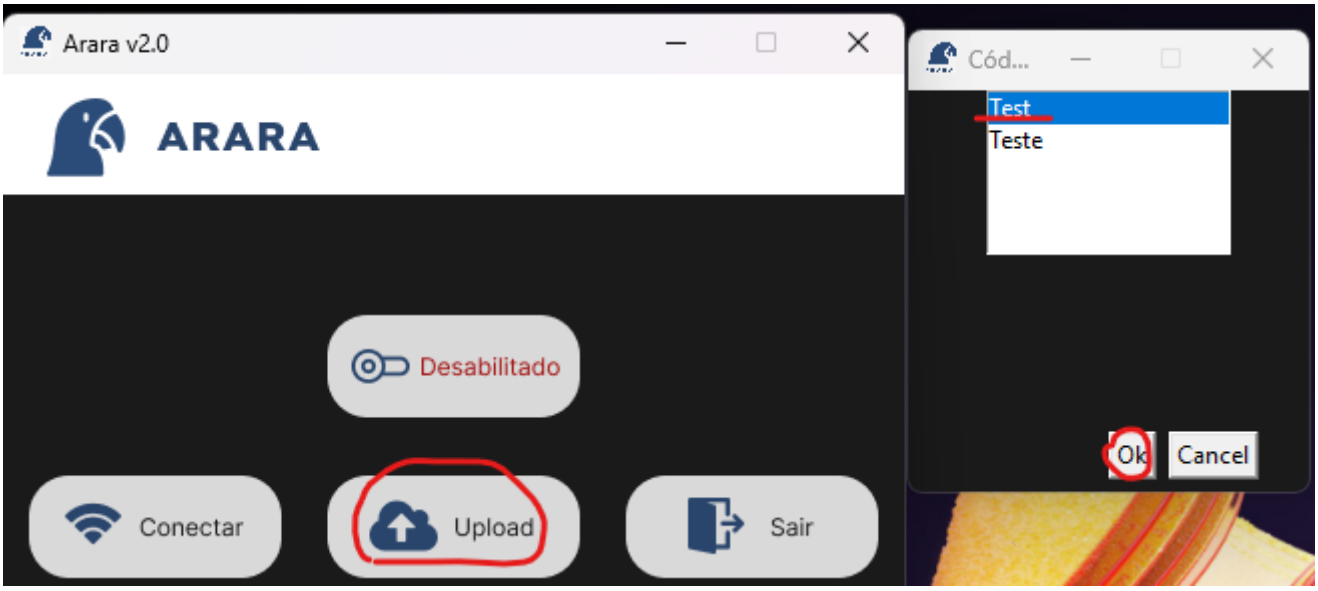
> Parabéns, com isso você conseguiu realizar seu primeiro código teleoperado

> Você deve notar que após a conexão o LED, antes amarelo e azul, ficou na cor verde, isso indica uma conexão bem sucedida

---

Upload

O botão de upload disponibiliza códigos já prontos para podermos utilizar. Caso queira escolher um faça o seguinte processo.



> Dependendo do código pode ser que apareça uma nova janela explicando o que aquele código executa, o indicado acima por exemplo, é uma programação estilo tank