

Instalando o Arduino IDE.

Nesse tutorial iremos apresentar as etapas de instalação do **Arduino IDE**, explicando seu modo de funcionamento como, escrita, compilação e Upload do código para o Kit Arduino.

O Arduino IDE é uma aplicação multiplataforma escrita na linguagem Java. É esquematizado para introduzir a programação a artistas e a pessoas não familiarizadas com o desenvolvimento de software. Inclui um editor de código com recursos de realce de sintaxe, parênteses correspondentes e indentação automática, sendo capaz de compilar e carregar programas para a placa com um único clique. [1]

Podemos encontrar a IDE em varias extensões para Download, tanto para Windows como Linux no site oficial www.Arduino.cc, basta clicar na aba download e escolher entre as distribuições disponíveis, se sua máquina for Windows basta clicar <http://downloads.arduino.cc/arduino-1.6.5-r5-windows.exe> que é a ultima versão ate então e seguir com o processo de instalação abaixo.

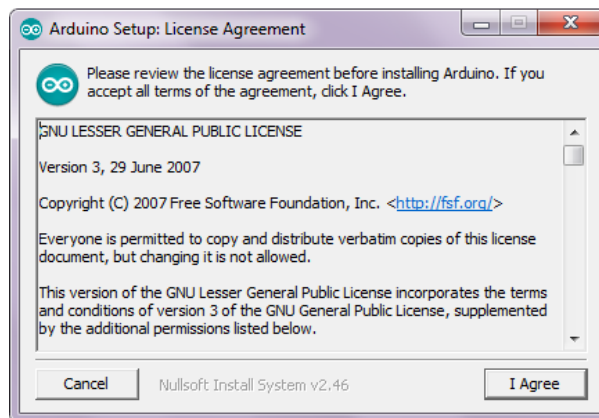


Figura 1 – Iniciando instalação do Arduino IDE

Execute o instalador e clique em **I Agree** até a opção install.

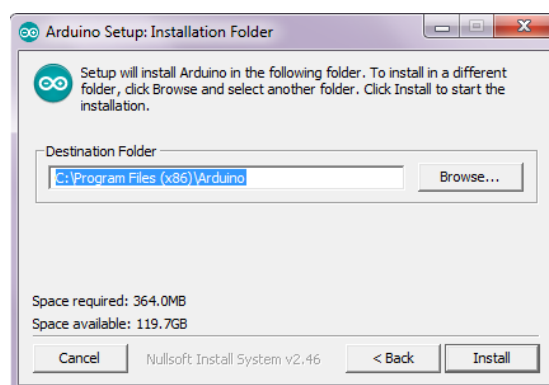


Figura 2 – Final da instalação

Deixe o caminho padrão de instalação do software e clique em **Install**.

Terminando a instalação, vamos abrir nosso programa e conhece-lo um pouco mais.

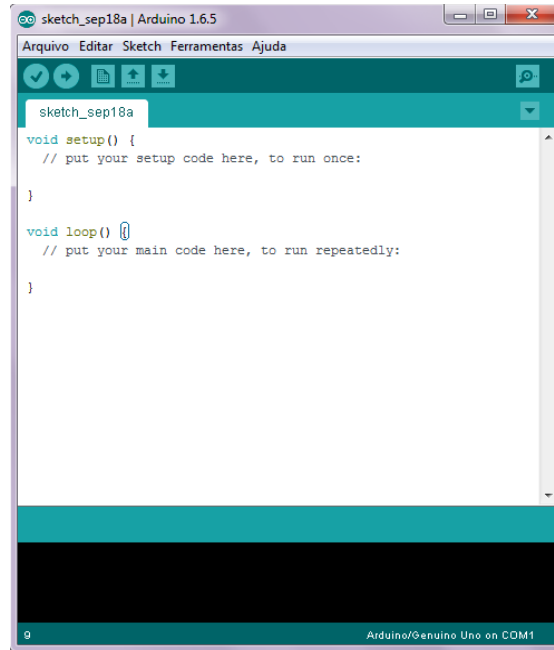


Figura 3 – Tela Inicial Arduino

Na janela você tem o uma barra de **Menu**, outra de **ferramentas rápidas**, uma **caixa de texto** para escrever o seu código e uma caixa de **mensagens de Bug**.

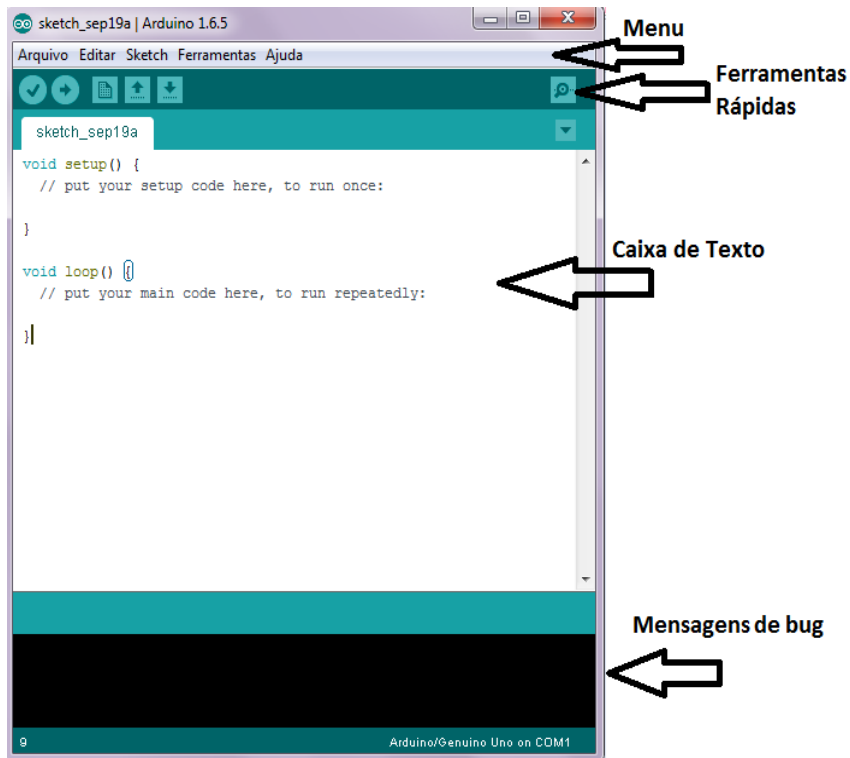


Figura 4 – Descrevendo a Janela

Na barra de ferramentas rápidas temos:

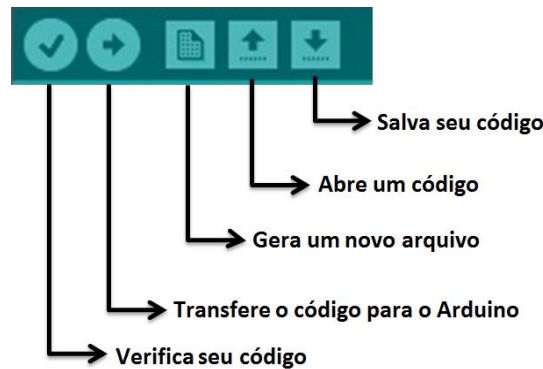


Figura 5 – Barra de Ferramentas Rápidas

Na caixa de texto já temos por padrão a estrutura básica de código no Arduino, se essa estrutura não vier é só escrever.

```

sketch_sep19a
void setup() {
  // put your setup code here, to run once:
}

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

```

Figura 6 – Estrutura do código Arduino.

A estrutura void setup() é utilizada para fazer as configurações iniciais do Arduino como, definição de pinos de I/O, interrupções, iniciar porta serial entre outras funcionalidades.

```

sketch_sep19a$
void setup() {
  // put your setup code here, to run once:
  pinMode(3, OUTPUT);
  pinMode(4, INPUT);
  attachInterrupt(0, blink, FALLING);
  Serial.begin(9600);
}

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

```

Figura 7 – Exemplo de código Setup().

Ex.: **pinMode(3, OUTPUT)** → A palavra **pinMode** [2] serve para configurar os pinos digitais e analógicos do Arduino. A palavra **OUTPUT** [3] identifica a porta como Saída. Então, a linha de código acima serve para dizer que **o pino 3 do Arduino é uma porta digital de Saída de dados**.

attachInterrupt(0, blink, FALLING) → A palavra **attachInterrupt** [4] serve para configurar os pinos de interrupção, que variam de acordo com o Arduino que você esteja utilizando. São utilizados 3 parâmetros para isso, ou seja é necessário 3 informações para utilizar corretamente o comando de interrupção.

- Uno, Nano, Mini, outros baseados - 328 2, 3
- Mega, Mega2560, MegaADK 2, 3, 18, 19, 20, 21
- Micro, Leonardo, outros baseados - 32u4 0, 1, 2, 3, 7
- Zero todos os pinos digitais, exceto 4
- Due todos os pinos digitais

O primeiro parâmetro diz em que porta digital a interrupção será acionada.

- 0 para pino 2
- 1 para pino 3
- Nos Arduinos Uno, Nano, Mini e outros.

O segundo parâmetro é o nome da função que deve ser chamada caso haja o acionamento da interrupção.

O terceiro parâmetro diz respeito ao modo de acionamento da interrupção:

- LOW quando o pino é baixo,
- CHANGE provoca a interrupção sempre que altera o valor do pino.
- RISING quando o pino vai de baixo a alto,
- FALLING quando o pino vai de alto a baixo.
- HIGH quando o pino tem valor alto.

Então, a linha do código da interrupção diz que quando houver mudança de estado do pino 2 do Arduino, **de alto para baixo**, será **acionada a interrupção** chamando a função de nome **blink** no código.

A estrutura **void loop()** é onde você escreve seu código, lembrando que a linguagem que o Arduino IDE utiliza é baseada na **linguagem C/C++** [5], logo é uma linguagem **sequencial**, ou seja, os **comandos são executados linha por linha**.

Após você escrever todo seu código, deve fazer a **compilação** [6]. Essa etapa é de súbita importância, pois nela você verá se cometeu **erros de sintaxe da linguagem de programação** e ira gerar o **código .hex** utilizado para fazer upload para o microcontrolador Atmega da placa Arduino. Então, não pule essa etapa e a cada modificação do código, seja um ponto e vírgula digitado você deve fazer uma nova compilação, para que essa modificação seja atualizada no arquivo .hex.

Qualquer erro encontrado no código será informado em mensagens de bug na parte inferior da janela.

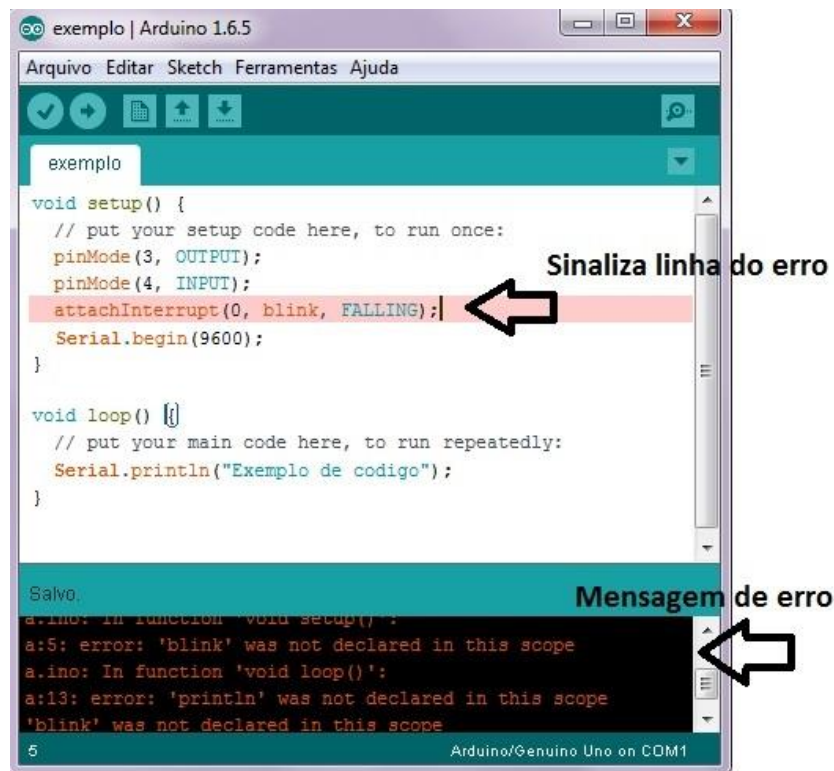


Figura 8 – Debugando Erros.

Para resolvê-los basta ler a mensagem escrita em inglês, no nosso caso o erro apareceu na chamada da interrupção que foi declarada no void setup(), mas especificamente na função blink que esta sendo chamada quando a interrupção é acionada. Ela diz que a função blink não existe então ela não pode ser chamada. E isso é correto, já que não escrevi nenhuma função com esse nome mesmo, para solucionar basta eu escrever a função, se for precisar, ou apagar essa linha de código do meu texto.

Erro resolvido o código irá compilar por completo gerando o arquivo .hex, agora basta você fazer o Upload do arquivo para o microcontrolador do Arduino. Para isso não se esqueça de conectar o Arduino ao seu computador e verificar se:

- os driver foram instalados corretamente;
- seleccionar o tipo de Arduino que você esta utilizando. Basta ir a **Ferramentas** → **Placa: “Arduino/Gnuino Uno”** e seleccionar o tipo em uso.

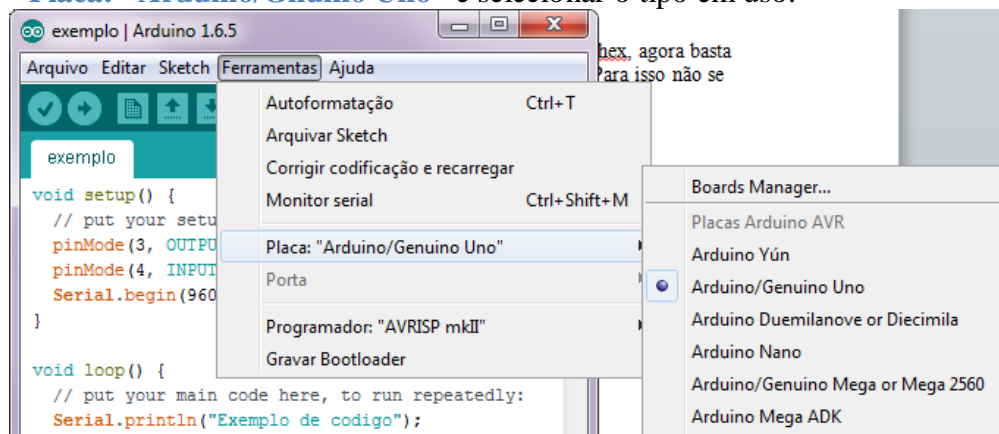


Figura 9 – seleccionando Arduino.

- Selecionar a porta usb correta para a comunicação. Basta ir a **Ferramenta** → **Porta**.

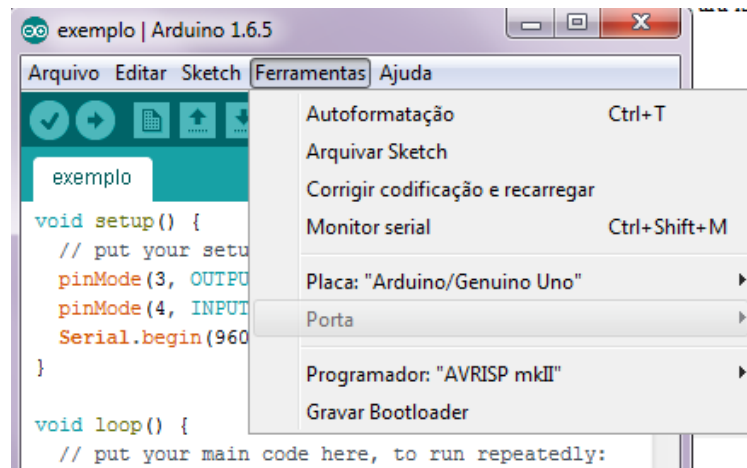


Figura 10 – Selecionando porta usb.

Por fim, basta clicar em fazer upload do código para o Arduino e fazer os testes.

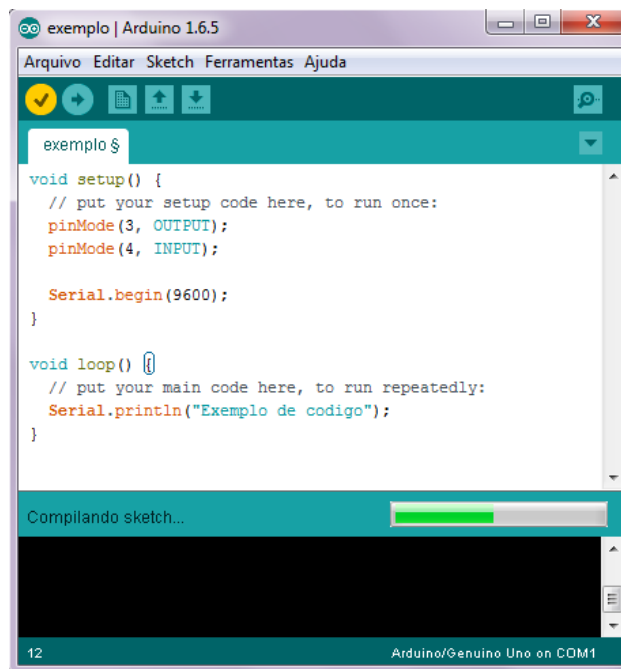


Figura 11 – Fazendo Upload Código.

Novamente não utilizamos todas as funcionalidades do software, mas esse é o básico para se começar a brincar com seu kit Arduino.

Bibliografias

[1] Arduino IDE - <https://pt.wikipedia.org/wiki/Arduino>
 [2] pinMode - <https://www.arduino.cc/en/Reference/PinMode>
 [3] output - <https://www.arduino.cc/en/Reference/PinMode>
 [4] attachInterrupt Arduino - <https://www.arduino.cc/en/Reference/AttachInterrupt>
 [5] Referências da Linguagem - <http://playground.arduino.cc/Portugues/Referencia>
 [6] Compilação do código - <https://pt.wikipedia.org/wiki/Compilador>
 Site Arduino - www.Arduino.cc