

**PROGRAMA QUALI-PIM-CAP  
SUBÁREA AUTOMAÇÃO DE PROCESSOS INDUSTRIAIS**

<b>Atividade Prática</b>	Instalação do Arduino IDE	<b>Período</b>	09 de set à 20 de out de 2022
<b>Monitor</b>		<b>Módulo</b>	1
<b>Professor</b>	Andrey Ruben Ribeiro Bessa		

**UTILIZAÇÃO DO SOFTWARE ARDUINO IDE**

**Sumário**

1. OBJETIVOS.....	1
2. INFORMAÇÃO TEÓRICA .....	1
3. SOFTWARE ARDUINO IDE – NOÇÕES BÁSICAS .....	2
<b>3.1. Menu principal</b> .....	5
3.1.1. Arquivo.....	5
3.1.2. Editar .....	6
3.1.3. Sketch.....	7
3.1.4. Ferramentas .....	8
3.1.5. Ajuda .....	10
3.1.6. Acesso Rápido .....	10
<b>3.2. Exemplo de Programa</b> .....	11
3.2.1. Estrutura de um programa Arduino .....	11
3.2.2. Exemplo Pisca LED.....	12

## 1. OBJETIVOS

- Exibir as principais funcionalidades do software de programação Arduino IDE.

## 2. INFORMAÇÃO TEÓRICA

O Arduino Integrated Development Environment (IDE) é um aplicativo de plataforma cruzada (para Windows, macOS, Linux) escrito na linguagem de programação Java. Ele é usado para escrever e carregar programas em placas compatíveis com Arduino, mas também, com a ajuda de núcleos de terceiros, pode ser usado com placas de desenvolvimento de terceiros.

O código fonte do IDE é publicado sob a GNU General Public License, versão 2. O Arduino IDE suporta as linguagens C e C++ usando regras especiais de estruturação de código. O Arduino IDE fornece uma biblioteca de software para o projecto de fiação, que fornece muitos procedimentos comuns de E/S. O código escrito pelo usuário requer apenas duas funções básicas, para iniciar o esboço e o ciclo principal do programa, que são compilados e vinculados a um apêndice do programa main() em um ciclo com a cadeia de ferramentas GNU, que também

está incluída. O Arduino IDE usa o programa *avrdude* para converter o código executável em um arquivo de texto em codificação hexadecimal carregada na placa Arduino usando um programa carregado no *firmware* da placa.

### 3. SOFTWARE ARDUINO IDE – NOÇÕES BÁSICAS

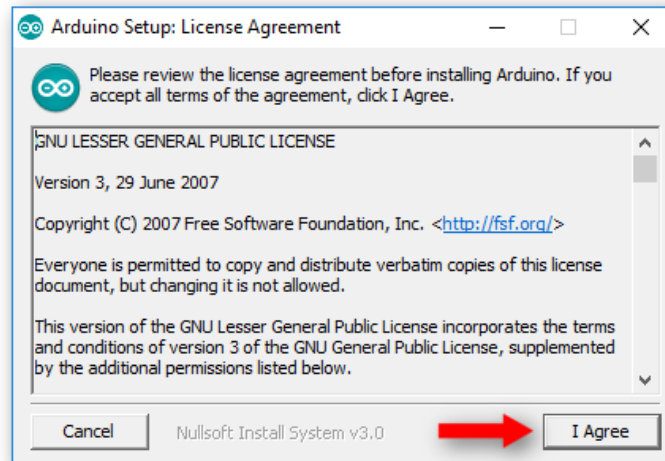
Antes de iniciar o estudo é necessário obter o programa Arduino IDE, o mesmo pode ser baixado pelo site <https://www.arduino.cc/en/main/software>. A Arduino disponibiliza um arquivo instalável: **Windows Installer, for Windows XP and up**, que será estudado neste documento. Em seguida, é preciso instalar o software no computador.



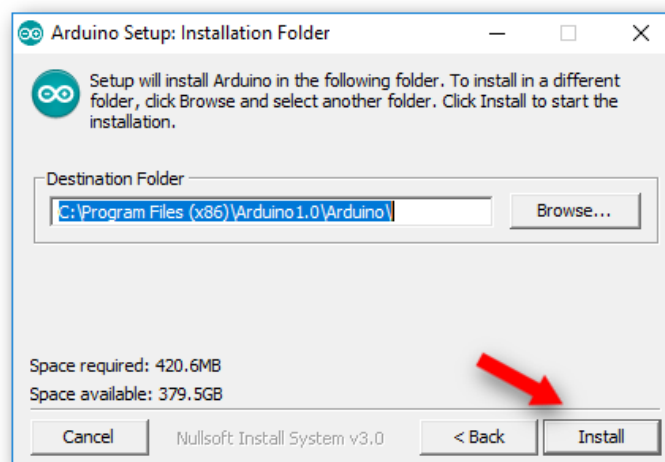
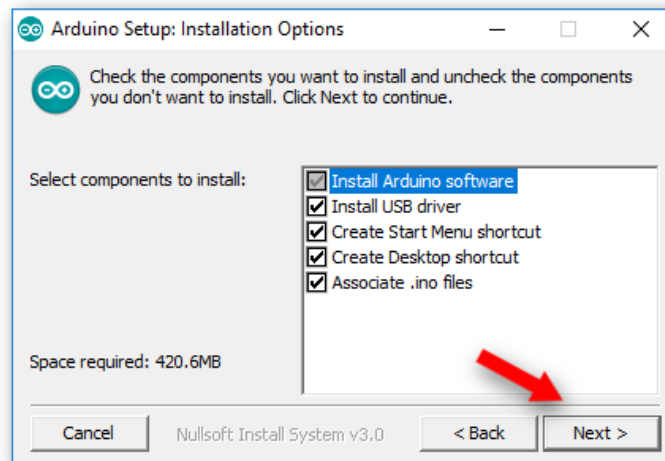
Se quiser fazer uma doação para o software Arduino este é o momento. Se não, basta clicar em Just Download.

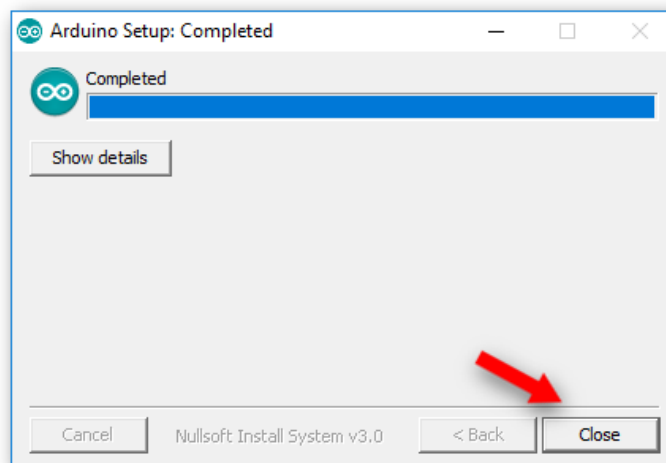
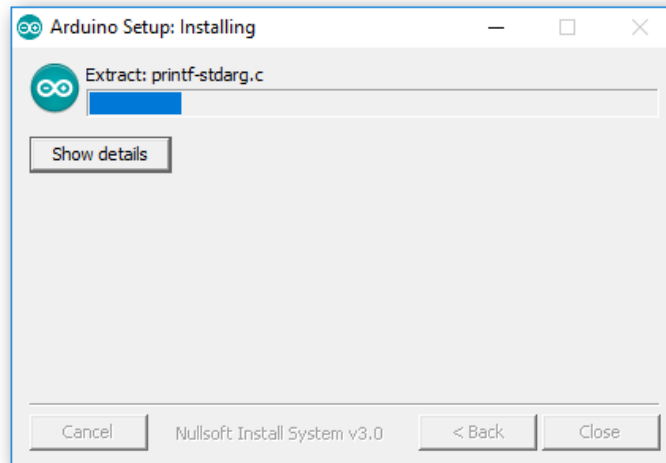


Execute o arquivo instalador '.exe' e siga as instruções de instalação.



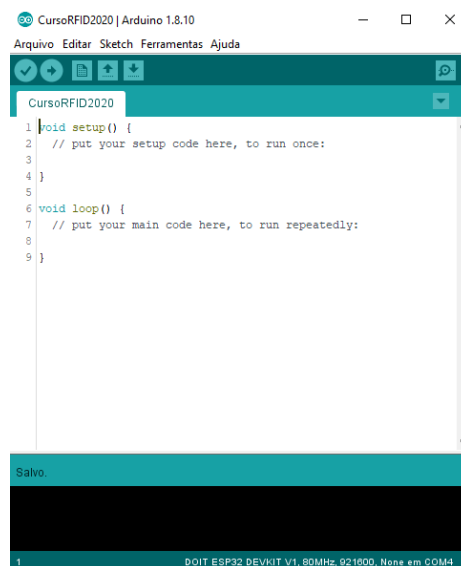
Certifique-se de que todos os componentes na tela abaixo estejam selecionados.





Após a instalação, abra a IDE Arduino pelo atalho criado na sua área de trabalho.

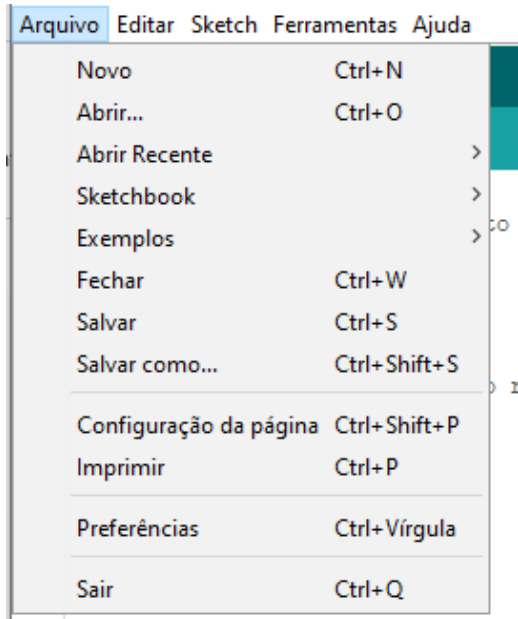
Ao abrir o programa a tela de visão geral que aparece é a seguinte:



### 3.1. Menu principal

Arquivo Editar Sketch Ferramentas Ajuda

#### 3.1.1. Arquivo



- **Novo**: Cria uma nova instância do editor, com a estrutura mínima de um esboço já em vigor.
- **Abrir**: Permite carregar um arquivo de esboço navegando pelas unidades e pastas do computador.
- **Abrir Arquivo Recente**: Fornece uma pequena lista dos esboços mais recentes, prontos para serem abertos.
- **Sketchbook** (caderno de esboços): Mostra os esboços salvos dentro da estrutura de pastas do sketchbook (caderno de esboços); clicar em qualquer nome abre o esboço correspondente em uma nova instância do editor.
- **Exemplos**: Qualquer exemplo fornecido pelo software do Arduino (IDE) ou pelas bibliotecas aparece neste item de menu. Todos os exemplos são estruturados em uma árvore que permite o acesso fácil por tópico ou biblioteca.
- **Fechar**: Fecha a instância do software Arduino do qual é clicado.
- **Salvar**: Salva o esboço com o nome atual. Se o arquivo não tiver sido nomeado anteriormente, um nome será fornecido em uma janela de “Salvar como...”.
- **Salvar como...**: Permite salvar o esboço atual com um nome diferente.
- **Configuração de página**: Mostra a janela de configuração de página para impressão.
- **Imprimir**: Envia o esboço atual para a impressora de acordo com as configurações definidas na janela Page Setup.
- **Preferências**: Abre a janela de preferências, onde algumas configurações do IDE podem ser personalizadas, como, por exemplo, o idioma da interface do IDE.

- **Sair**: Fecha todas as janelas do IDE. Os mesmos esboços que estiverem abertos quando o botão **Sair** for pressionado serão reabertos automaticamente na próxima vez que o IDE for iniciado.

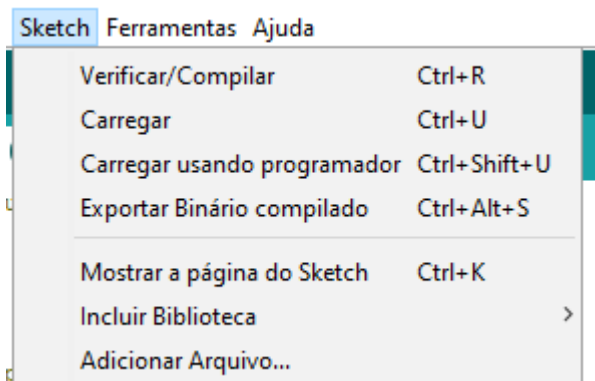
### 3.1.2. Editar

Editar	Sketch	Ferramentas	Ajuda
Desfazer		Ctrl+Z	
Refazer		Ctrl+Y	
Cortar		Ctrl+X	
Copiar		Ctrl+C	
Copiar para Fórum		Ctrl+Shift+C	
Copiar como HTML		Ctrl+Alt+C	
Colar		Ctrl+V	
Selecionar tudo		Ctrl+A	
Vá para a linha...		Ctrl+L	
Comentar/descomentar		Ctrl+Barra	
Aumentar indentação		Guia	
Diminuir indentação		Shift+Guia	
Aumentar Tamanho da Fonte		Ctrl+Mais	
Diminuir Tamanho da Fonte		Ctrl+Menos	
Localizar...		Ctrl+F	
Localizar próximo		Ctrl+G	
Localizar anterior		Ctrl+Shift+G	

- **Desfazer/Refazer**: O botão **Desfazer** desfaz a última ação que você fez durante a edição; quando uma ação é desfeita dessa forma, você ainda pode recuperá-la com o botão **Refazer**.
- **Cortar**: Remove o texto selecionado e o coloca na área de transferência.
- **Copiar**: Copia o texto selecionado no editor para a área de transferência.
- **Copiar para o Fórum**: Copia o código do seu esboço para a área de transferência em um formato adequado para ser postado no fórum: completo e com coloração de sintaxe.
- **Copiar como HTML**: Copia o código do seu esboço para a área de transferência como HTML, adequado para incorporação em páginas da Web.
- **Colar**: Coloca o conteúdo da área de transferência na posição do cursor, no editor.
- **Selecionar tudo**: Seleciona e destaca todo o conteúdo do editor.
- **Comentar/Descomentar**: Coloca ou remove o // (marcador de comentário) no início de cada linha selecionada.
- **Aumentar/Diminuir Recuo**: Adiciona ou subtrai um espaço no início de cada linha selecionada, movendo o texto um espaço à direita ou eliminando um espaço no início.

- **Encontrar**: Abre a janela Find and Replace (Localizar e Substituir), onde é possível especificar um texto a ser pesquisado dentro do esboço atual e seguindo um conjunto de várias opções configuráveis.
- **Encontrar Próximo**: Destaca a próxima ocorrência - se houver - da string (seqüência de caracteres) especificada como item de pesquisa na janela Find (Localizar), relativo à posição do cursor.
- **Encontrar Anterior**: Encontrar o anterior: destaca a ocorrência anterior - se houver - da string (seqüência de caracteres) especificada como item de pesquisa na janela Find (Localizar), relativo à posição do cursor.

### 3.1.3. Sketch



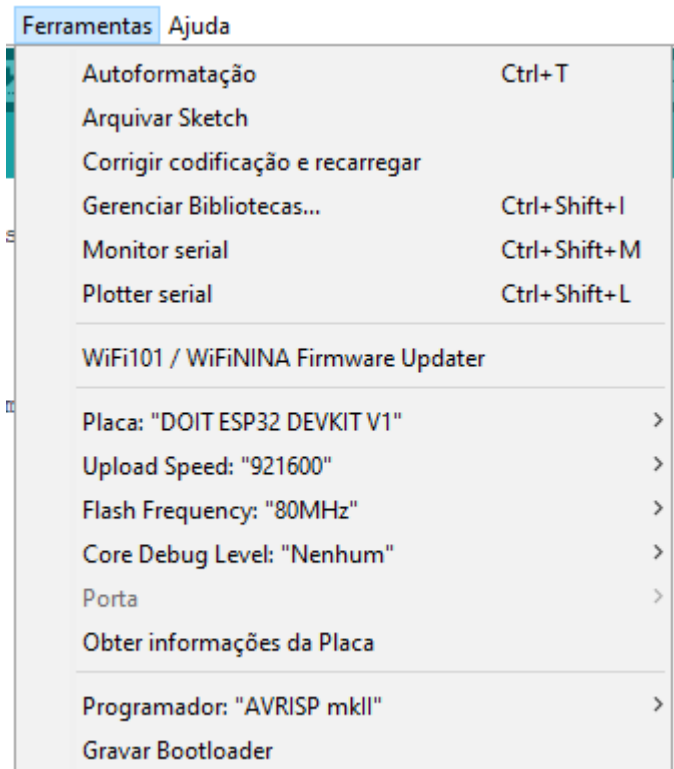
## Sketch (esboço)

- **Verify/Compile** (verificar/compilar): verifica se o seu esboço contém erros compilando-o; ele relatará o uso da memória do código e as variáveis na área do console.
- **Upload** (envio): compila e carrega o arquivo binário na placa configurada através da porta configurada.
- **Upload Using Programmer** (carregar usando programador): isso substituirá o bootloader na placa; você precisará usar Tools (Ferramentas) -> Burn Bootloader (Gravar Bootloader) para restaurá-lo e ser capaz de fazer upload para a porta serial USB novamente. No entanto, ele permite que você use a capacidade total da memória Flash para o seu esboço. Tenha em mente que este comando NÃO irá queimar/gravar os fusíveis. Para fazer isso, um comando Tools -> Burn Bootloader deve ser executado.
- **Export Compiled Binary** (exportar binário compilado): salva um arquivo .hex que pode ser mantido como arquivo ou enviado para a placa usando outras ferramentas.
- **Show Sketch Folder** (mostrar pasta de esboços): abre a pasta atual de esboços.
- **Include Library** (incluir biblioteca): adiciona uma biblioteca ao seu esboço inserindo instruções #include no início do seu código. Falaremos mais sobre bibliotecas ainda nesse tutorial. Além de incluir bibliotecas (já importadas para o IDE) em seu projeto, você também

pode usar esse menu para acessar o Library Manager (Gerenciador de Bibliotecas) e importar novas bibliotecas a partir de arquivos .zip.

- **Add File...** (adicionar arquivo...): adiciona um arquivo de código ao esboço (ele será copiado do seu local atual). O novo arquivo aparece em uma nova guia na janela do esboços. Os arquivos podem ser removidos do esboço usando o menu de abas que pode ser acessado clicando no pequeno ícone triangular abaixo do monitor serial, do lado direito da barra de ferramentas.

### 3.1.4. Ferramentas



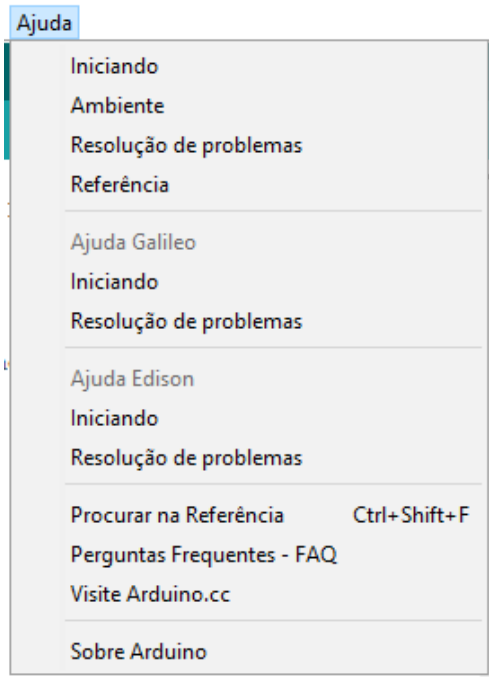
## Tools (ferramentas)

- **Formatar automaticamente**: Formata o seu código de uma forma agradável, indentando-o de forma que as aberturas e fechamentos de chaves (“{”, “}”) fiquem alinhados e as instruções contidas entre as chaves estejam corretamente indentadas.
- **Arquivar esboço**: arquiva uma cópia do esboço atual no formato .zip. O arquivo resultante é colocado no mesmo diretório que o esboço.
- **Corrigir codificação e recarregar**: corrige possíveis discrepâncias entre a codificação do mapa de caracteres do editor e os mapas de caracteres de sistemas operacionais.
- **Monitor serial**: abre a janela do monitor serial e inicia a troca de dados com qualquer placa conectada na porta atualmente selecionada. Isso normalmente reinicia a placa, se a placa suportar a reinicialização a partir da abertura da porta serial.



- **Placa**: selecione a placa que você está usando. Falaremos mais sobre as diferentes placas ainda nesse tutorial.
- **Porta**: este menu contém todos os dispositivos seriais (reais ou virtuais) da sua máquina. Ele deverá ser atualizado automaticamente sempre que o menu Tools (ferramentas) for acessado.
- **Programador**: permite selecionar um programador hardware ao programar uma placa ou um chip e não usar a conexão USB serial em série. Normalmente, você não precisará disto, mas se você estiver [gravando um bootloader](#) em um novo microcontrolador, você usará essa funcionalidade.
- **Gravar Bootloader**: os itens neste menu permitem gravar um [Bootloader](#) (“carregador de inicialização”) no microcontrolador em uma placa Arduino. Isso não é necessário para o uso normal de um Arduino ou placa Genuíno, mas é útil se você comprar um novo microcontrolador ATmega (que normalmente vem sem um bootloader). Certifique-se de que selecionou a placa correta no menu **Placas** antes de gravar o bootloader na placa de destino.

### 3.1.5. Ajuda



## Ajuda

Aqui você encontra fácil acesso a vários documentos que vêm com o software do Arduino (IDE). Você tem acesso à página de Introdução, à página de Referência e até mesmo a este guia para o IDE, além de outros documentos. O acesso é feito localmente, sem uma conexão à Internet. Os documentos são uma cópia local dos documentos online e podem possuir links para os tutoriais no site do Arduino.

- Localizar nas Referências Esta é a única função interativa do menu Help (ajuda): essa função seleciona, diretamente na cópia local da Referência do Arduino, a página relevante para a função ou comando que estiver sob o cursor.

### 3.1.6. Acesso Rápido



**Verificar:** compila seu código e busca os erros existentes.



**Carregar:** compila seu código e faz o upload para a placa configurada. Falaremos mais sobre upload ainda nesse tutorial. Nota: Se você estiver usando um programador externo com sua placa,

you can secure (keep pressed) the “shift” key of the computer when using this icon. If you do this, the text will change to “Upload using Programmer” (Send using Programmer).



**Novo:** creates a new sketch.



**Abrir:** presents a menu with all the sketches of your sketchbook (notebook of sketches). When clicking on a sketch, it will be loaded into the current window, replacing its content. Note: due to a bug in Java, this menu does not scroll; if you need to open a sketch that is at the end of the list, use the menu **Arquivo > Sketchbook**.



**Salvar:** saves your sketch.



**Monitor Serial:** opens the serial monitor. Displays the serial data sent by the Arduino or Genuino (USB or serial board). To send data to the board, type the text and click the **send** button or press **enter**. Choose the transmission rate that corresponds to the rate passed for **Serial.begin** in your sketch. Note that in Windows, Mac or Linux, the Arduino or Genuino board will be restarted (execute again the sketch from the beginning) when you connect to the serial monitor.

## 3.2. Exemplo de Programa

### 3.2.1. Estrutura de um programa Arduino

You do not need to be an expert in C language to program with Arduino. When opening the IDE Arduino you will face a standard program structure containing the functions `setup()` and `loop()`.

**setup()** – It is in this part of the program that you configure the initial options of your program: the initial values of a variable, if a port will be used as input or output, messages for the user, etc. This function will execute only once at the beginning of the program.

**loop()** – Diferente da função setup(), essa parte do programa repete uma estrutura de comandos de forma contínua ou até que alguma comando de “parar” seja enviado ao Arduino.

Vamos ver exatamente como isso funciona, levando em consideração o programa abaixo, que acende e apaga o led embutido na placa Arduino em intervalos de 1 segundo:

```
1 //Programa : Pisca Led Arduino
2 //Autor : FILIPEFLOP
3
4 void setup()
5 {
6     //Define a porta do led como saida
7     pinMode(13, OUTPUT);
8 }
9
10 void loop()
11 {
12     //Acende o led
13     digitalWrite(13, HIGH);
14
15     //Aguarda o intervalo especificado
16     delay(1000);
17
18     //Apaga o led
19     digitalWrite(13, LOW);
20
21     //Aguarda o intervalo especificado
22     delay(1000);
23 }
```

### 3.2.2. Exemplo Pisca LED

Vimos acima um programa exemplo para piscar nosso LED, o famoso “Hello World” da eletrônica e um dos primeiros passos com arduino. O exemplo Blink LED.

A primeira coisa que fazemos no início do programa é colocar uma pequena observação sobre o nome do programa, sua função e quem o criou:

```
1 // Programa : Pisca Led Arduino
2 // Autor : FILIPEFLOP
```

Comece uma linha com barras duplas ( // ) e tudo o que vier depois dessa linha será tratado como um comentário. Uma das boas práticas de programação é documentar o seu código por meio das linhas de comentário. Com elas, você pode inserir observações sobre como determinada parte

do programa funciona ou o que significa aquela variável **xyz** que você criou. Isso será útil não só para você, se precisar alterar o código depois de algum tempo, como também para outras pessoas que utilizarão o seu programa.

Após os comentários, vem a estrutura do **setup()**. É nela que definimos que o pino 13 do Arduino será utilizado como saída.

```
1 void setup()
2 {
3   //Define a porta do led como saída
4   pinMode(13, OUTPUT);
5 }
```

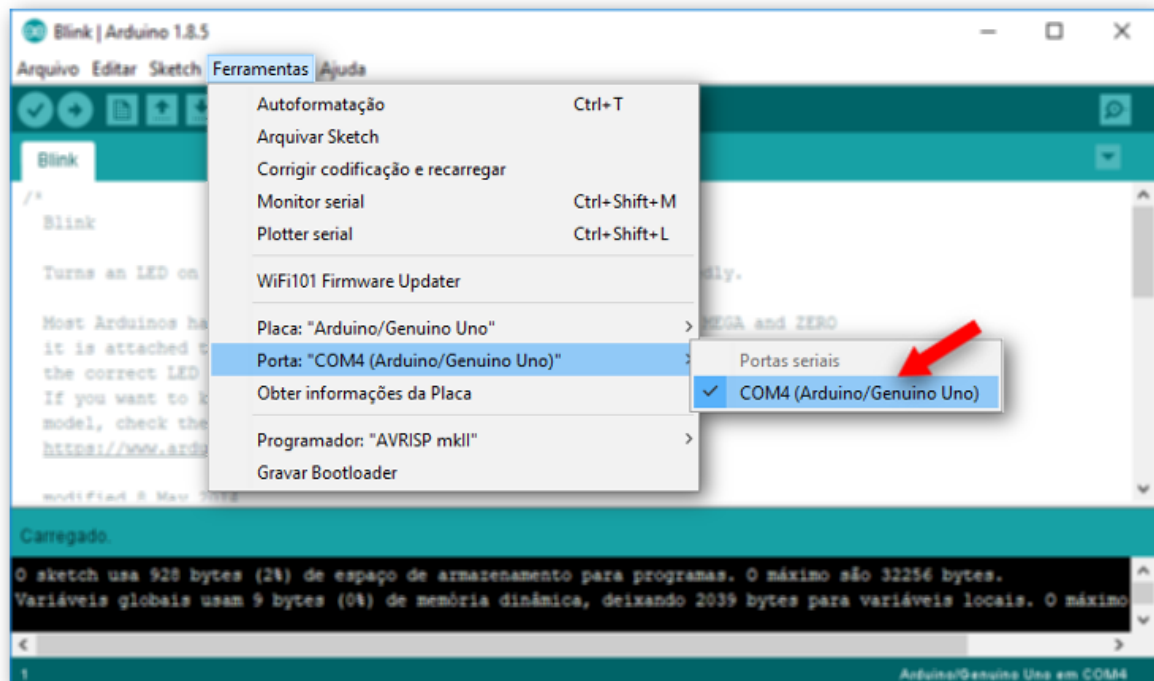
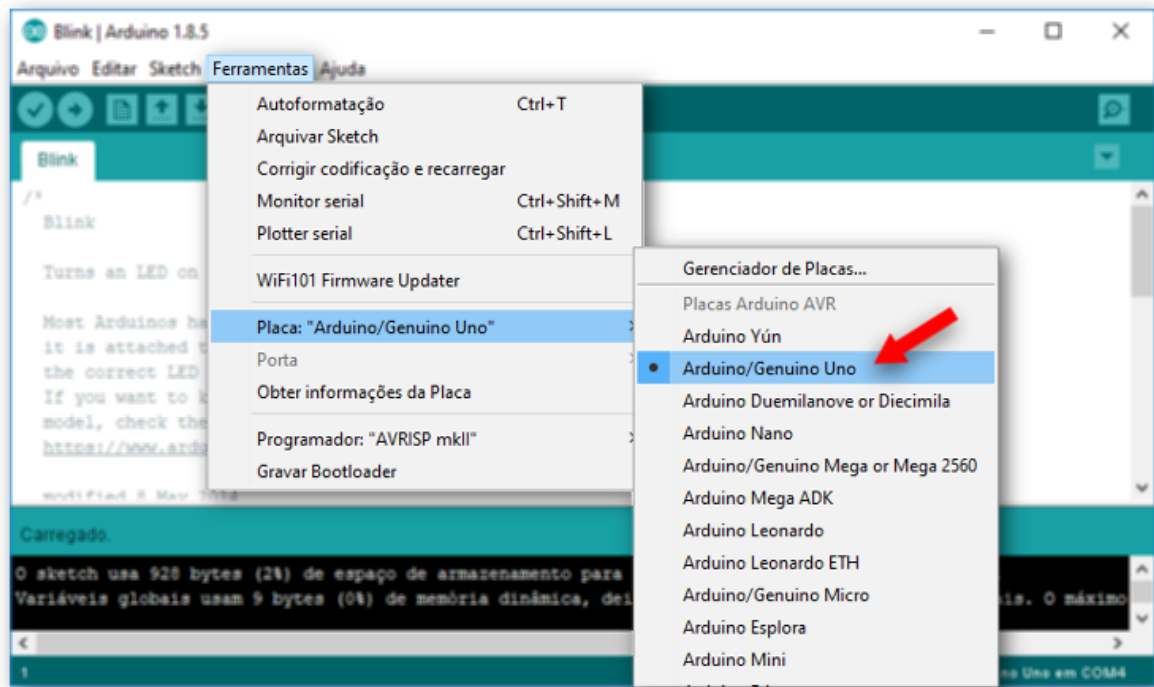
Por último, temos o **loop()**, que contém as instruções para acender e apagar o led, e também o intervalo entre essas ações:

```
1 void loop()
2 {
3   //Acende o led
4   digitalWrite(13, HIGH);
5
6   //Aguarda o intervalo especificado
7   delay(1000);
8
9   //Apaga o led
10  digitalWrite(13, LOW);
11
12  //Aguarda o intervalo especificado
13  delay(1000);
14 }
```

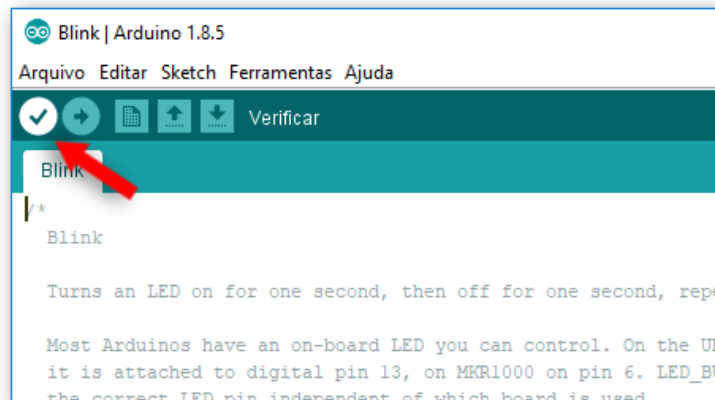
A linha do código contendo **digitalWrite(13, HIGH)** coloca a porta 13 em nível alto (**HIGH**, ou **1**), acendendo o led embutido na placa. O comando **delay(1000)**, especifica o intervalo, em milissegundos, no qual o programa fica parado antes de avançar para a próxima linha.

O comando **digitalWrite(13, LOW)**, apaga o led, colocando a porta em nível baixo (**LOW**, ou **0**), e depois ocorre uma nova parada no programa, e o processo é então reiniciado.

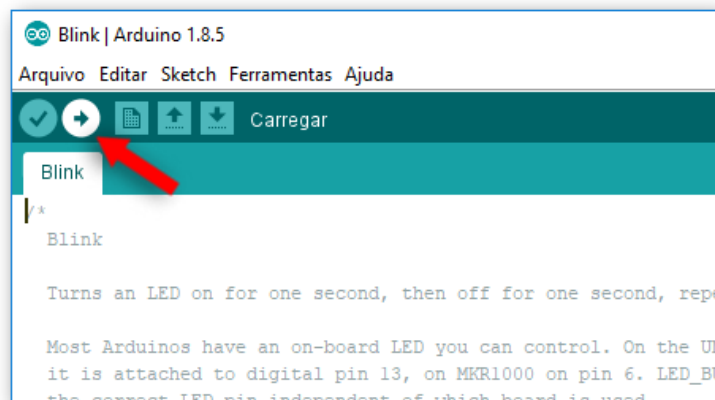
Quando o código estiver pronto para ser carregado na placa, conecte a placa Arduino no seu computador, entre no menu ferramentas, escolha o modelo da placa e a porta na qual a mesma está conectada:



Então clique no botão de compilação. Isso irá verificar se existe algum erro no código e gerar o arquivo de programa necessário para carregar na placa.



Se estiver tudo OK, clique no botão carregar. Isso irá gravar o programa na placa.



Caso não apareça nenhum erro, você deverá ver o LED piscando em um intervalo de 1 segundo.