



PROGRAMAÇÃO \_

FRONT-END \_

DATA SCIENCE \_

INTELIGÊNCIA ARTIFICIAL \_

DEVOPS \_

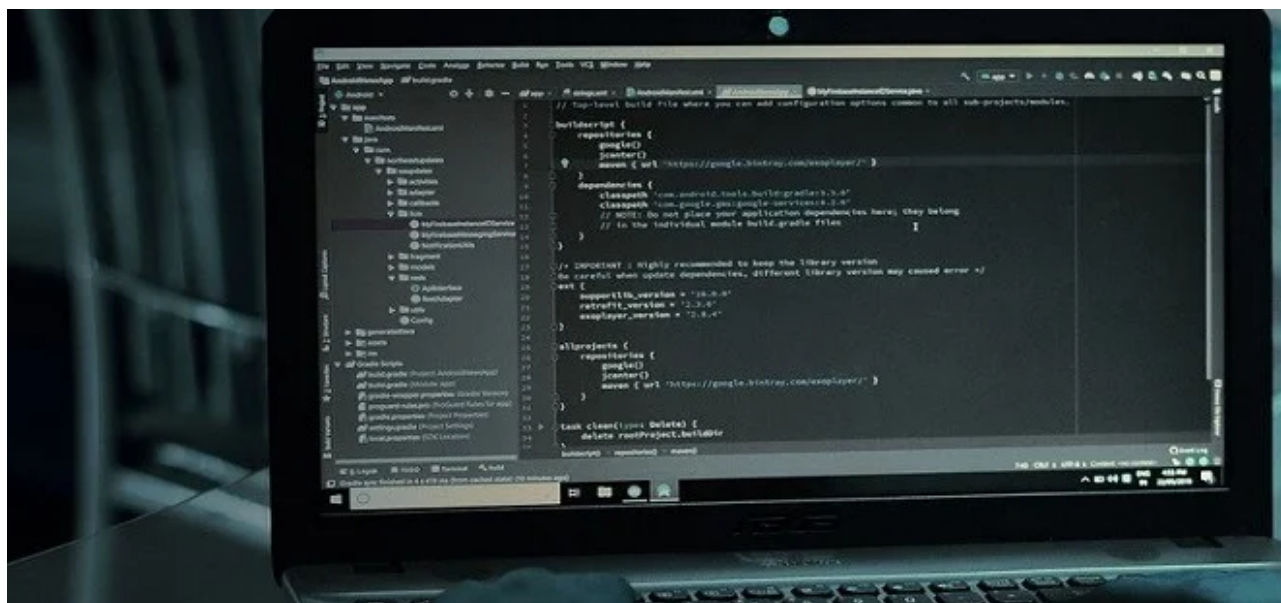
UX &amp; DESIGN \_

MOBILE \_

INOVAÇÃO &amp; GESTÃO \_

Artigos > **Programação**

# Compilando e executando programas em C/C++ no Windows

**Larissa Gabriela**

30/11/2021

COMPARTILHE



- [MinGW](#)
- [Variáveis de ambiente](#)
- [Executando no Prompt de Comando](#)
- [Ferramentas para aqueles que utilizam o Visual Studio Code](#)
- [Atalho para quem utiliza o Sublime Text](#)
- [Conclusão](#)

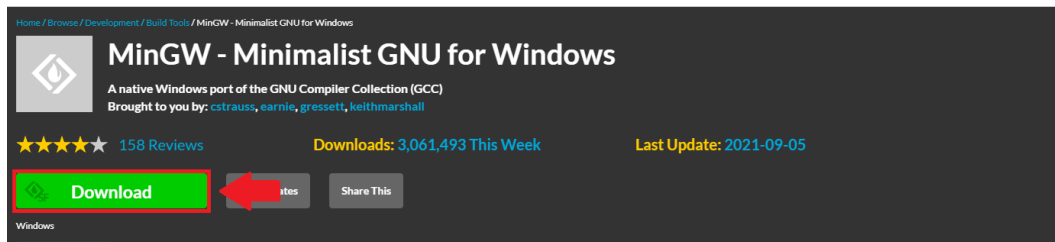
Uma dúvida muito comum quando começamos a aprender as linguagens C ou **C++** é como compilamos e executamos nossos programas no Windows. Esse artigo tem como objetivo mostrar como podemos fazer isso além de contar com algumas dicas de ferramentas e atalhos disponíveis em alguns editores de código.

A primeira coisa que devemos fazer para isso é fazer o Download do MinGW que se encontra [nesse link](#).

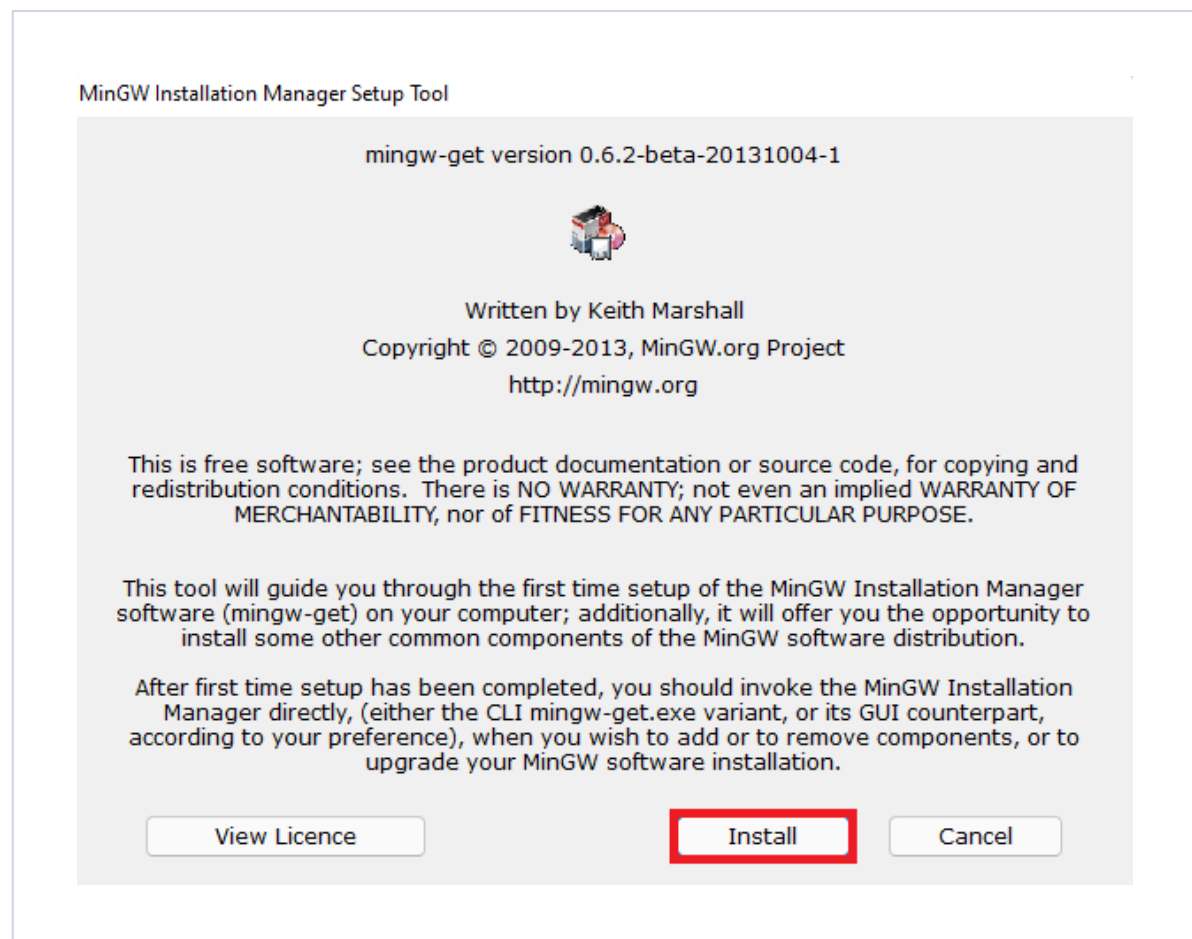
## MinGW

Este software inclui um conjunto de arquivos cabeçalho para a API do Windows que permite a pessoa dev usar o GCC. O GCC (GNU Compiler Collection) é um conjunto de compiladores de linguagens de programação. Nesses compiladores estão incluídos o de C e C++.

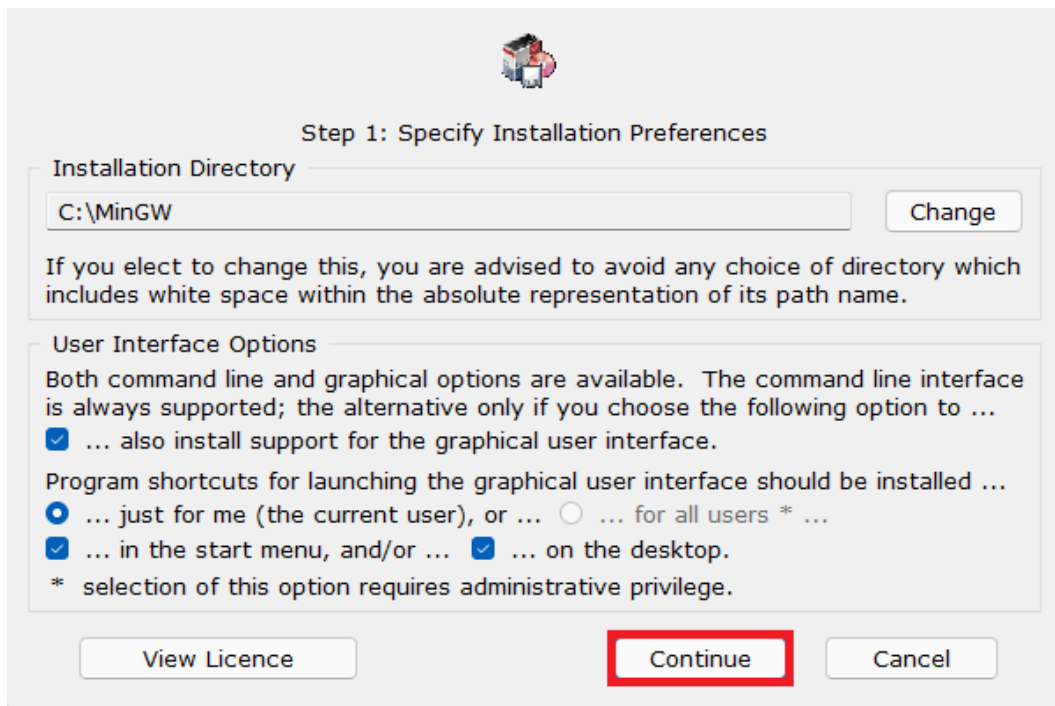
1) Assim que acessar o link disponível acima, você verá a seguinte tela e deverá clicar em **Download**.



2) Quando o Download for finalizado vá até a pasta do seu computador em que o arquivo foi salvo e execute. Assim que executar o programa você verá a guia “MinGW Installation Manager Setup Tool”, mostrada abaixo. Clique em **Install** (Instalar).

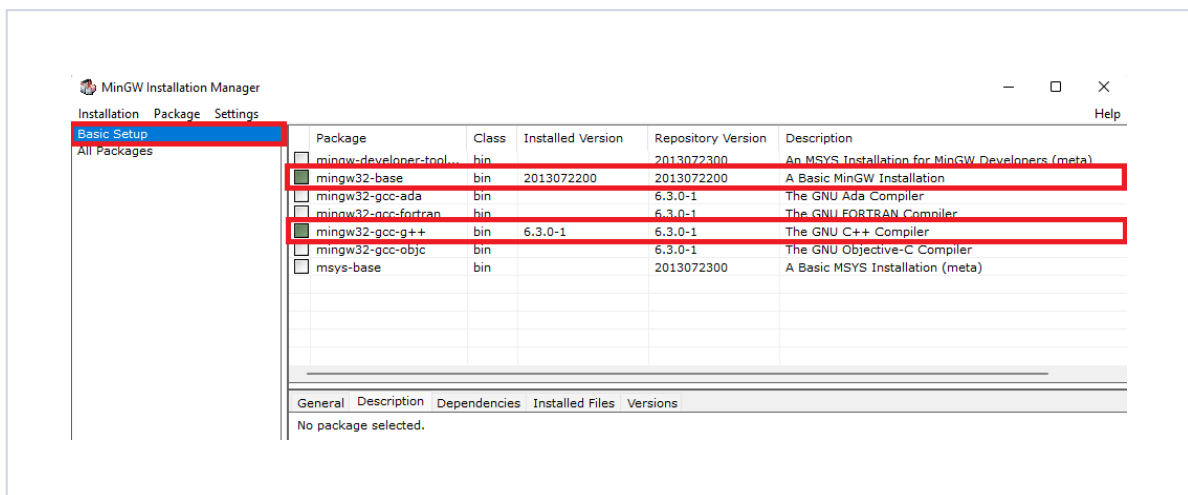


3) Na página seguinte, clique em **Continue** (Continuar).

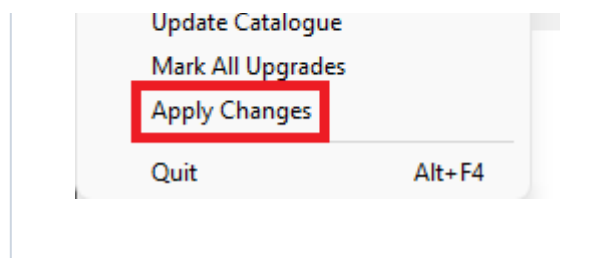


4) Com isso, a instalação do MinGW está concluída e podemos executar o MinGW.

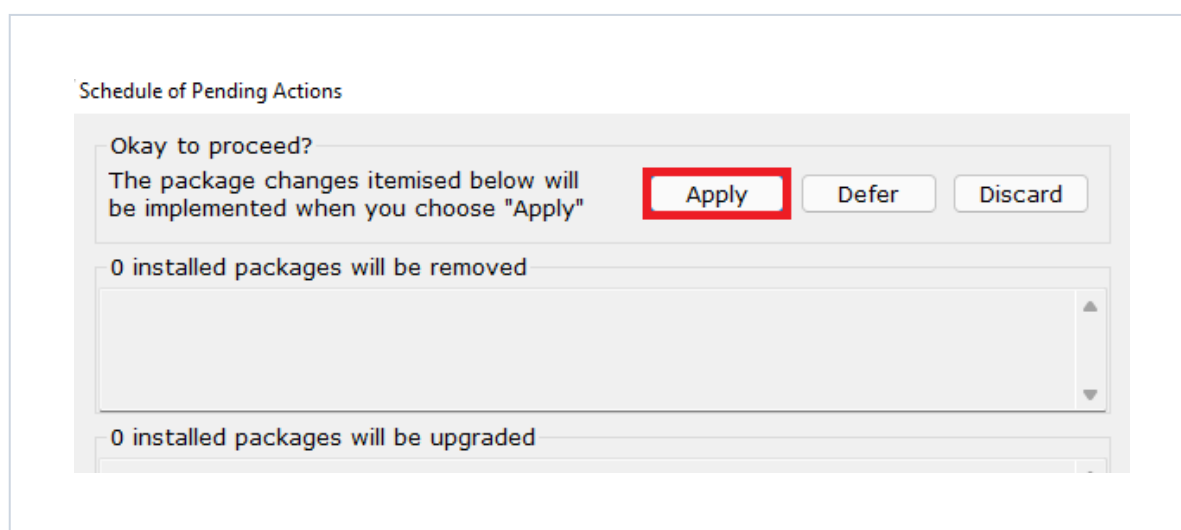
5) Na aba **Basic Setup**, localizada no canto esquerdo, selecione as opções **mingw32-base** e **mingw32-gcc-g++**.



6) No canto superior esquerdo da tela você verá o que está mostrado na imagem a seguir. Clique em **Installation** (Instalação) e em seguida **Apply Changes** (Aplicar Mudanças).



7) Você verá na tela o texto abaixo, em inglês. Clique em **Apply** (Aplicar).

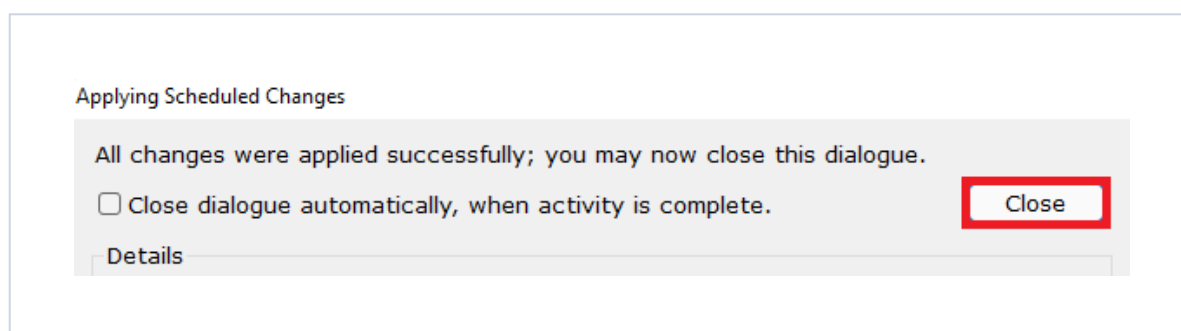


### Tradução:

"Tudo bem para continuar?"

As mudanças do pacote mostradas abaixo serão implementadas assim quando você escolher 'Aplicar'

8) Aguarde a instalação até que apareça a tela com o conteúdo a seguir. Clique em **Close** (Fechar).



### Tradução:



Agora, nosso próximo passo é configurar nossas variáveis de ambiente.

**Matricule-se na escola de PROGRAMAÇÃO**

Junte-se a uma comunidade de **+500 mil** estudantes

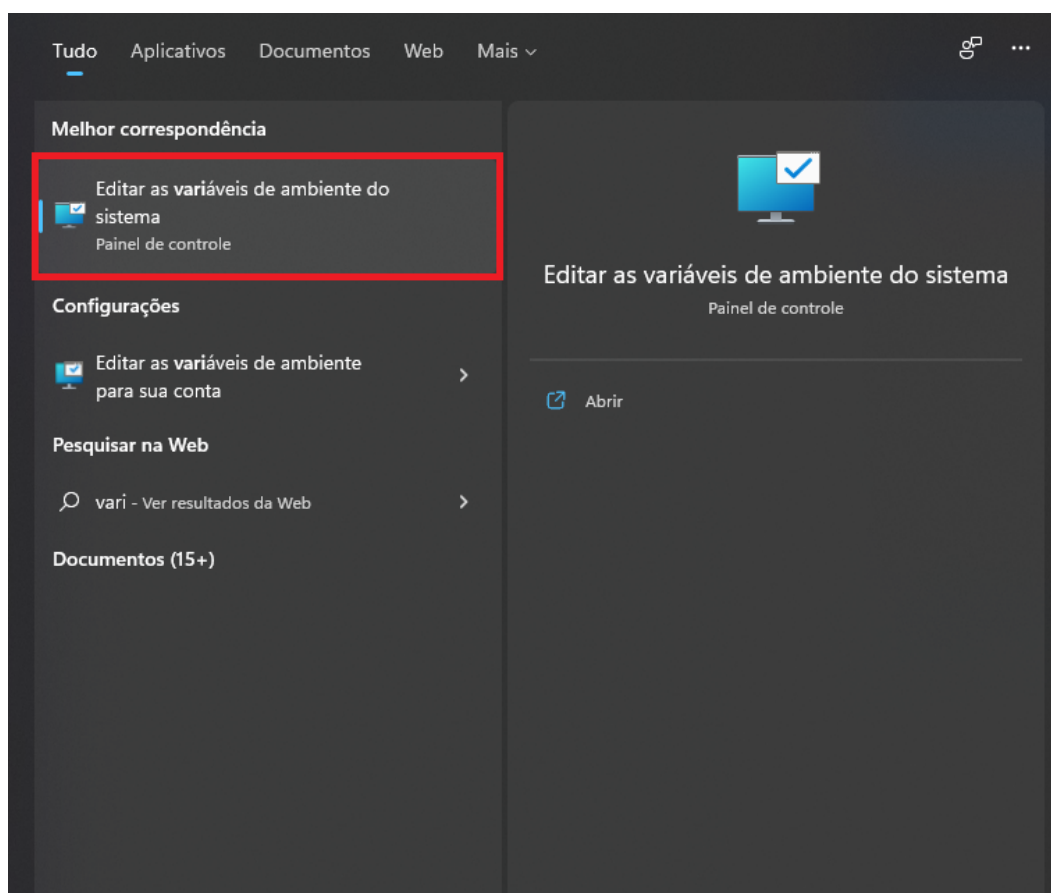
- Acesso a **TODOS** os cursos em uma única assinatura
- Novos lançamentos a cada semana
- Desafios práticos

**SAIBA MAIS**

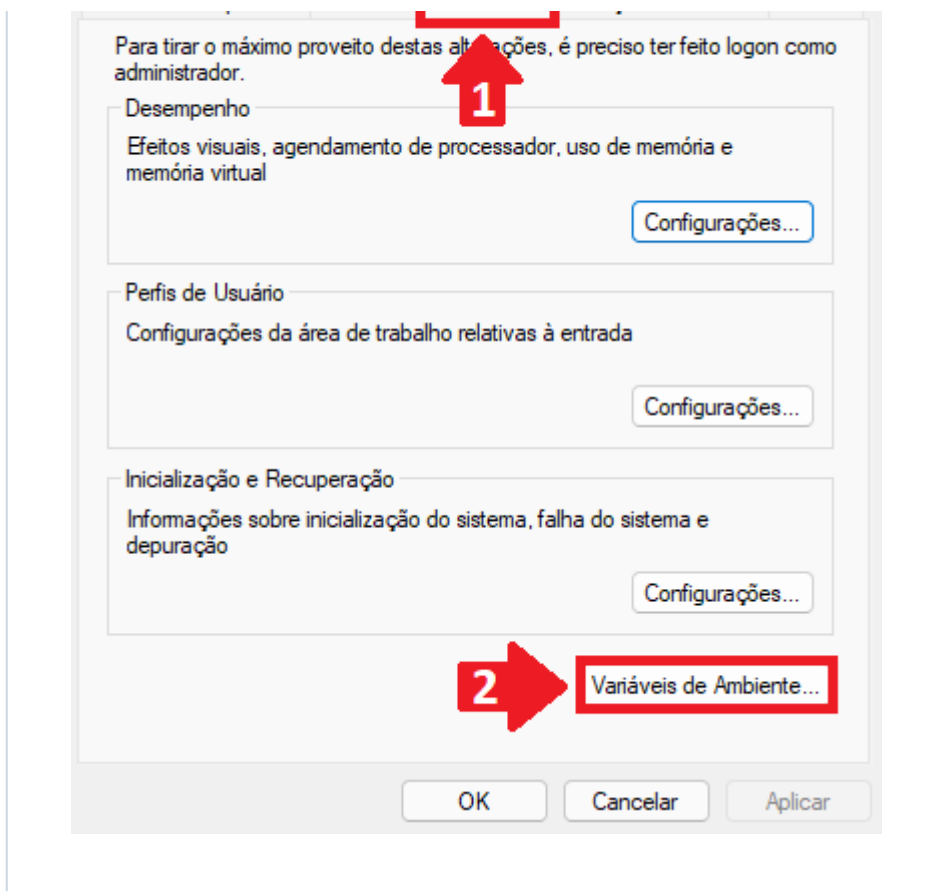
## Variáveis de ambiente

Quando um programa é executado ele precisa receber certas informações. Essas informações devem ser passadas por alguém e nesse caso que entram as **variáveis de ambiente**. A variável de ambiente que vamos modificar é a **PATH**. Ela guarda informações de onde estão nossos arquivos executáveis para que possamos executar um comando sem a necessidade de digitar o caminho absoluto.

9) Vá até a barra de pesquisa do seu Windows e pesquise “Editar as variáveis de ambiente do sistema”. Selecione a opção destacada.

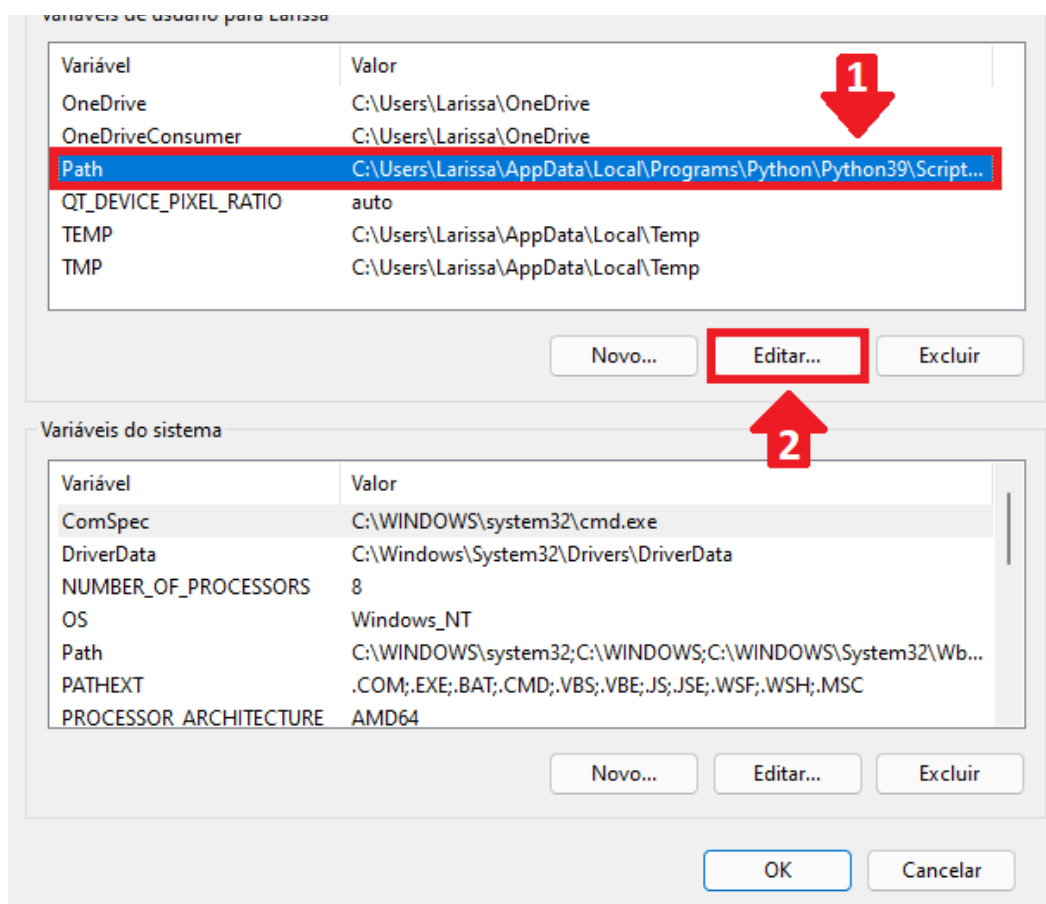


10) Abrirá a tela abaixo. Na aba **Avançado** clique em **\*\*Variáveis de Ambiente...\*\***.

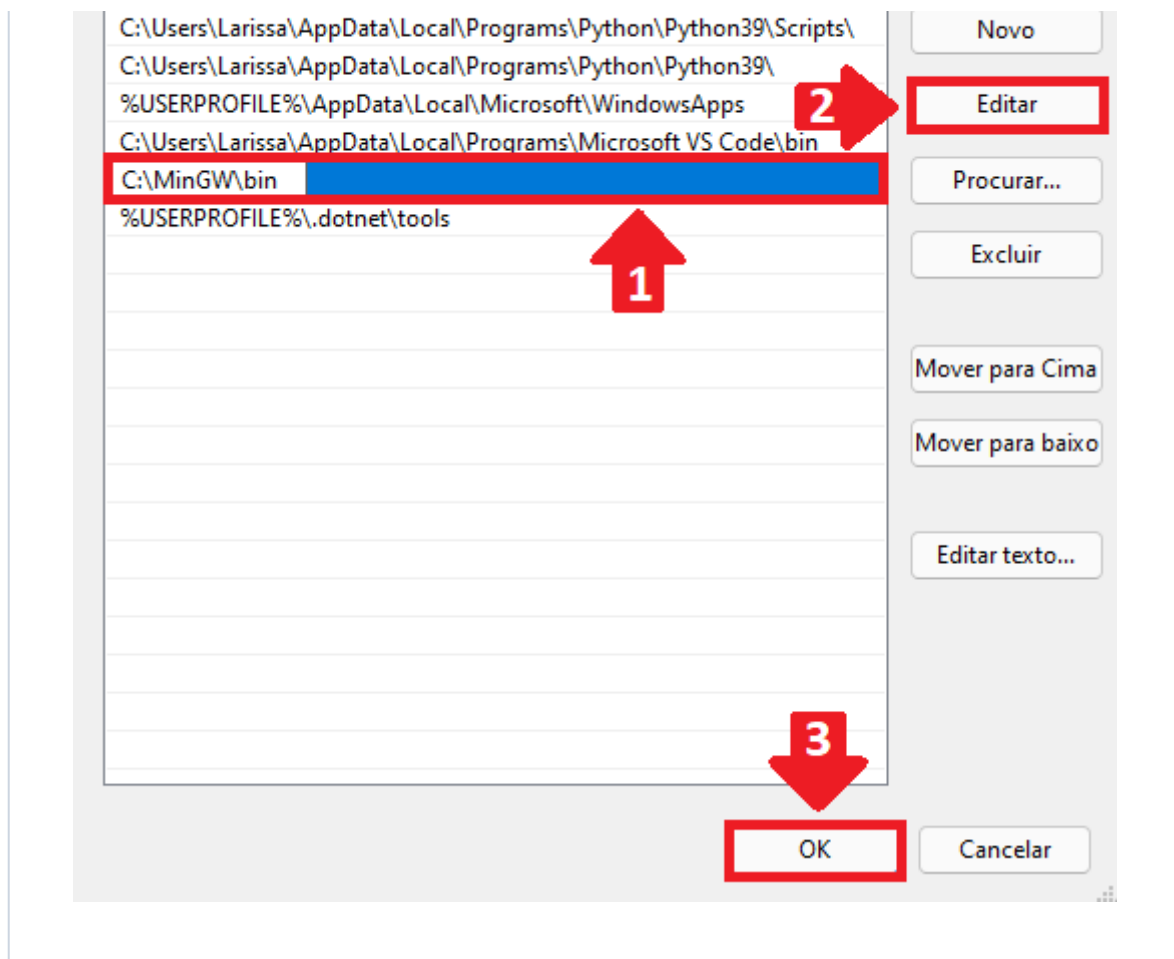


11) Teremos algumas variáveis de usuário. Selecione **Path** (ela ficará azul) e vá em **Editar...**





12) Vá até uma linha em branco e selecione (você pode dar dois cliques ou clicar em “Editar” após selecionar). Digite **C:\MinGW\bin**. Após, clique em **OK**.



Com isso finalizamos as configurações e já podemos compilar e executar códigos em C/C++ no Windows. Caso ainda assim ocorra algum erro para compilar seus programas, recomendo verificar novamente se a variável de ambiente foi alterada com sucesso. Acontecem casos em que a variável acaba não sendo salva ou teve algum erro de digitação.

## Executando no Prompt de Comando

1) Vá até o diretório onde seu arquivo `.c` ou `.cpp` está salvo através do comando `cd`. Para conferir se realmente está na pasta desejada use o comando `dir`.



```
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\Larissa>cd OneDrive\Documentos\Alura\C  
  
C:\Users\Larissa\OneDrive\Documentos\Alura\C>dir  
O volume na unidade C é Windows-SSD  
O Número de Série do Volume é 0C56-EA33  
  
Pasta de C:\Users\Larissa\OneDrive\Documentos\Alura\C  
  
22/11/2021 12:51 <DIR> .  
17/11/2021 08:30 <DIR> ..  
24/10/2017 14:31      131 forca.h  
24/10/2017 14:31      28 palavras.txt  
22/11/2021 11:39      64 teste.c  
22/11/2021 12:16      64 teste2.c  
                4 arquivo(s)      287 bytes  
                2 pasta(s) 66.899.836.928 bytes disponíveis  
  
C:\Users\Larissa\OneDrive\Documentos\Alura\C>_
```

2) Após, compile o arquivo utilizando `gcc teste.c -o teste.exe`. Nenhum resultado é obtido. Isso significa que a [compilação](#) foi feita com sucesso.

**Observação:** Lembrando que teste é o nome do arquivo que eu escolhi para usar como exemplo. Você utilizará o nome que você escolheu. Verá que um novo arquivo `.exe` aparecerá nessa mesma pasta. Esse é o arquivo executável.



```
22/11/2021 12:16          64 teste2.c
          4 arquivo(s)          287 bytes
          2 pasta(s) 66.899.836.928 bytes disponíveis

C:\Users\Larissa\OneDrive\Documentos\Alura\C>gcc teste.c -o teste.exe

C:\Users\Larissa\OneDrive\Documentos\Alura\C>dir
O volume na unidade C é Windows-SSD
O Número de Série do Volume é 0C56-EA33

Pasta de C:\Users\Larissa\OneDrive\Documentos\Alura\C

22/11/2021 12:57    <DIR>          .
17/11/2021 08:30    <DIR>          ..
24/10/2017 14:31          131 forca.h
24/10/2017 14:31          28 palavras.txt
22/11/2021 11:39          64 teste.c
22/11/2021 12:57    40.766 teste.exe
22/11/2021 12:16          64 teste2.c
          5 arquivo(s)    41.053 bytes
          2 pasta(s) 66.896.797.696 bytes disponíveis

C:\Users\Larissa\OneDrive\Documentos\Alura\C>
```

4) Agora, podemos executar utilizando `.\"teste.exe"`.

```
Prompt de Comando

22/11/2021 12:16          64 teste2.c
          4 arquivo(s)          287 bytes
          2 pasta(s) 66.899.836.928 bytes disponíveis

C:\Users\Larissa\OneDrive\Documentos\Alura\C>gcc teste.c -o teste.exe

C:\Users\Larissa\OneDrive\Documentos\Alura\C>dir
O volume na unidade C é Windows-SSD
O Número de Série do Volume é 0C56-EA33

Pasta de C:\Users\Larissa\OneDrive\Documentos\Alura\C

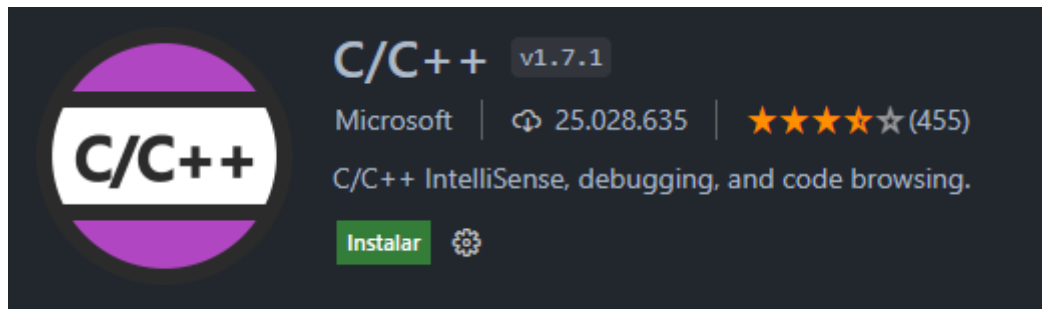
22/11/2021 12:57    <DIR>          .
17/11/2021 08:30    <DIR>          ..
24/10/2017 14:31          131 forca.h
24/10/2017 14:31          28 palavras.txt
22/11/2021 11:39          64 teste.c
22/11/2021 12:57    40.766 teste.exe
22/11/2021 12:16          64 teste2.c
          5 arquivo(s)    41.053 bytes
          2 pasta(s) 66.896.797.696 bytes disponíveis

C:\Users\Larissa\OneDrive\Documentos\Alura\C>.\"teste.exe"
Olá, mundo!

C:\Users\Larissa\OneDrive\Documentos\Alura\C>
```



Caso utilize o Visual Studio Code como editor de texto, recomendo a instalação de duas extensões que facilitarão sua experiência programando com essas linguagens. A primeira delas é a **C/C++**.



Essa extensão irá permitir o uso do IntelliSense, debugging e code browsing. O IntelliSense nos permite o preenchimento automático quando estamos codando. É possível, por exemplo, listar variáveis ou informações do parâmetro. Veja:

```
1  #include <stdio.h>
2
3  int main(){
4
5      for
6  }
```

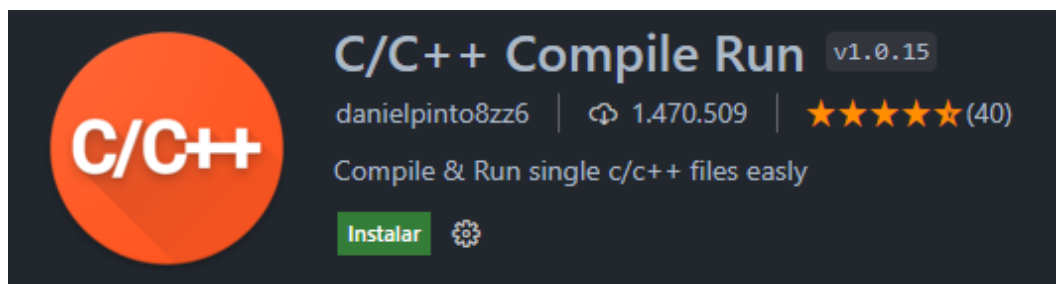
☐ for Code snippet for for loop  
for  
fortran  
\_Format\_string\_impl\_  
\_\_format\_string  
ferror  
\_Scanf\_format\_string\_  
\_Scanf\_format\_string\_impl\_  
\_Scanf\_format\_string\_params\_  
\_Printf\_format\_string\_  
\_Printf\_format\_string\_impl\_  
\_Printf\_format\_string\_params\_



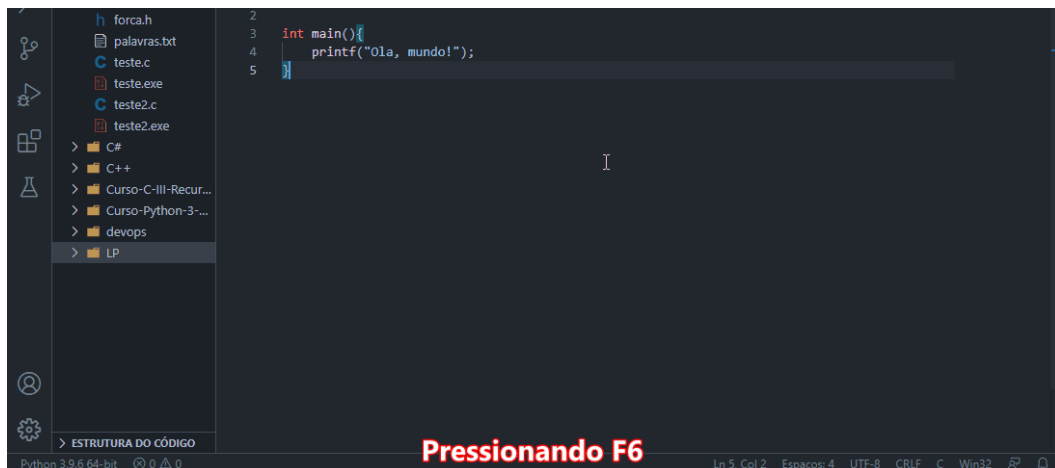
```
4
5     for (size_t i = 0; i < count; i++)
6     {
7         /* code */
8     }
9
10 }
```

O debugging (depuração, em português) será o processo de encontrar algum bug em seu código. Ou seja, ele irá facilitar a vida da desenvolvedora ou do desenvolvedor, permitindo que o erro seja encontrado com mais facilidade.

A segunda extensão que recomendo é a **C/C++ Compile Run**. Ela permite que o código seja compilado e executado pressionando apenas a tecla **F6**. Portanto, no Visual Studio Code, pressione F6 no programa que deseja executar e você verá o resultado no terminal disponível na parte inferior da tela.

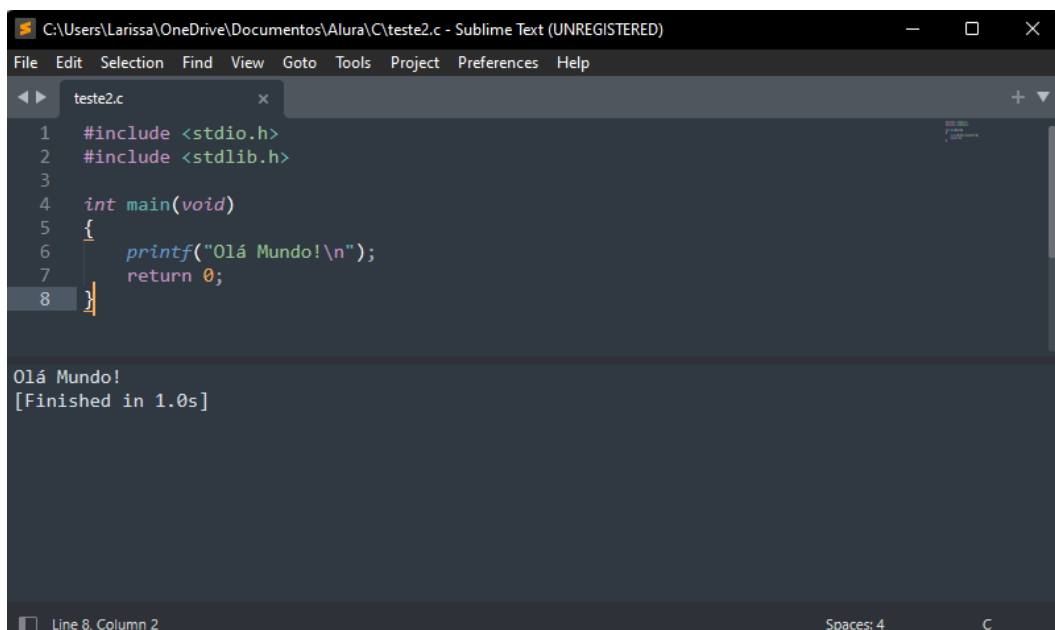


**Veja um exemplo de compilação e execução com o Visual Studio Code**



# Atalho para quem utiliza o Sublime Text

Uma dica, para aqueles que [utilizam o Sublime Text](#), conseguimos compilar e executar utilizando o **CTRL+B**.



**Veja um exemplo de compilação e execução com o Sublime Text**



```
1  #include <stdio.h>
2
3  int main(){
4      printf("Ola, mundo!");
5  }
```

**Pressionando CTRL + B**

Line 5, Column 2      Spaces: 4      C

## Conclusão

Agora seu ambiente está preparado para que você possa estudar e se aprofundar ainda mais no que essas linguagens podem te oferecer.

Quer mergulhar ainda mais no mundo da tecnologia? Leia também:

- [As linguagens C e C++: qual a diferença entre elas?](#)
- [Começando a programar com C](#)
- [C++: Vale a Pena Aprender em 2021? Por que e onde Usar?](#)
- [Como criar um README para o seu perfil do GitHub](#)



**Larissa Gabriela**





MATRICULE-SE

Nas horas vagas adoro jogar FPS e ver streams.

[Artigo Anterior](#)

**Diminuindo conta de luz com Flutter, Jogos 2D, Python Pandas e Seletores CSS**

[Próximo Artigo](#)

**Ada Lovelace, Charles Babbage e ENIAC: uma breve história da computação**

Veja outros artigos sobre  
[Programação](#)

## Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

**ME INSCREVA**

Nossas redes e apps



**MATRICULE-SE**

## Institucional

Sobre nós

Trabalhe conosco

Para Empresas

Para Sua Escola

Política de Privacidade

Compromisso de Integridade

Termos de Uso

Status

## A Alura

Formações

Como Funciona

Todos os cursos

Depoimentos

Instrutores(as)

Dev em <T>

Luri by ChatGPT

## Conteúdos

Alura Cases

Imersões

Artigos

Podcasts

Artigos de educação  
corporativa

## Fale Conosco

Email e telefone

Perguntas frequentes

## Novidades e Lançamentos

Email\*

ENVIAR



## **CURSOS**

### **Cursos de Programação**

Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

### **Cursos de Front-end**

HTML, CSS | React | Angular | JavaScript | jQuery

### **Cursos de Data Science**

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

### **Cursos de Inteligência Artificial**

IA para Programação | IA para Dados

### **Cursos de DevOps**

AWS | Azure | Docker | Segurança | IaC | Linux

### **Cursos de UX & Design**

Usabilidade e UX | Vídeo e Motion | 3D

### **Cursos de Mobile**

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

### **Cursos de Inovação & Gestão**

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas