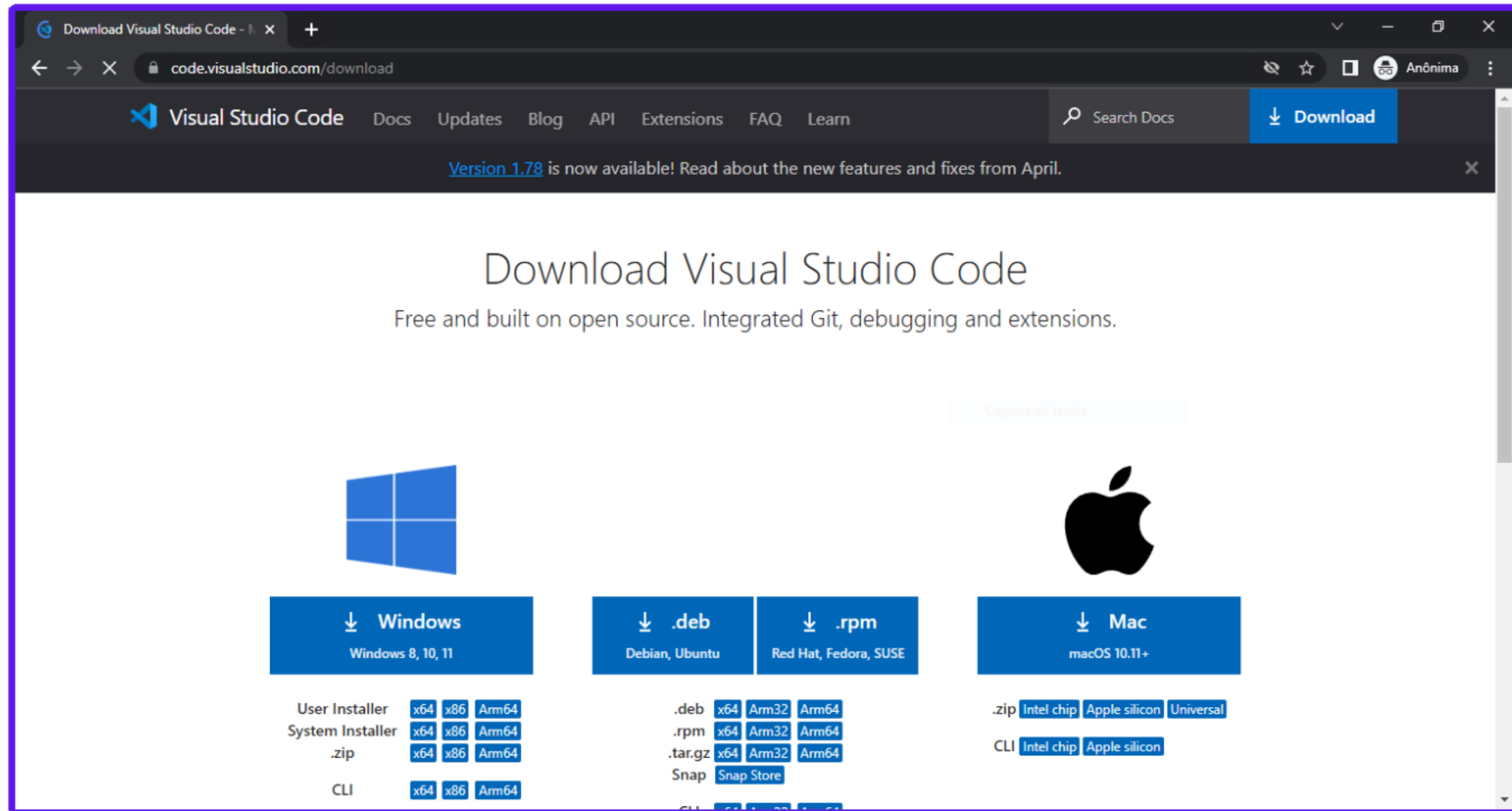


Como Instalar o Allegro no Visual Studio Code

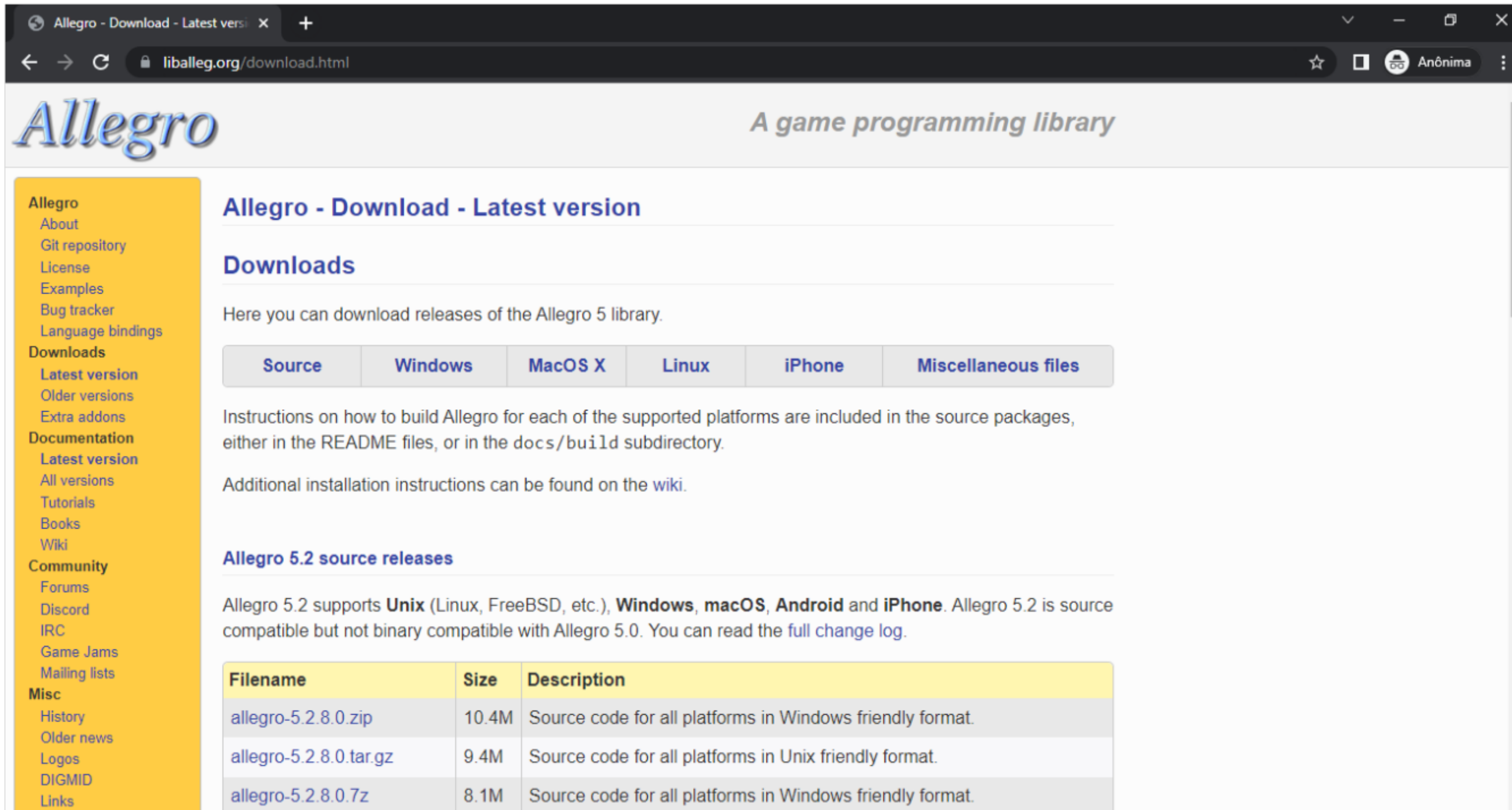
ATENÇÃO

**Todas as Images referentes a
alguma pagina da web, caso
cliquem em cima delas servira como
um link para a respectiva pagina**

1) Baixe o VS Code pelo link abaixo e faça a instalação



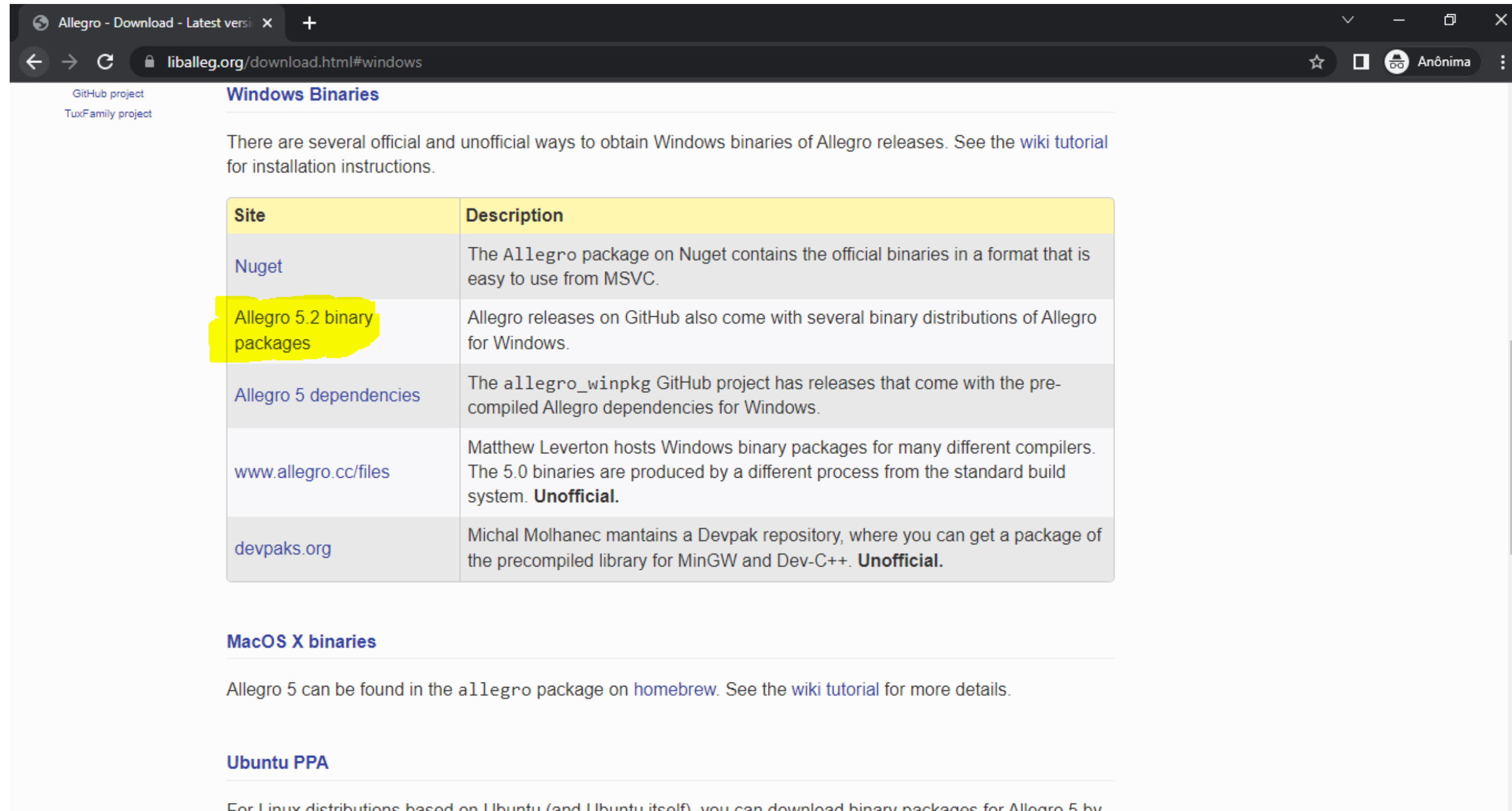
2) Acesse a página de downloads do Allegro e selecione seu sistema operacional



The screenshot shows a web browser window with the URL `liballeg.org/download.html`. The page features the Allegro logo and the tagline "A game programming library". A left sidebar contains navigation links for Allegro, Downloads, Documentation, Community, and Misc. The main content area is titled "Allegro - Download - Latest version" and "Downloads". It includes a paragraph about downloading releases of the Allegro 5 library, a set of tabs for different platforms (Source, Windows, MacOS X, Linux, iPhone, Miscellaneous files), and instructions on how to build Allegro. Below this, there is a section for "Allegro 5.2 source releases" with a table listing three source code packages.

Filename	Size	Description
allegro-5.2.8.0.zip	10.4M	Source code for all platforms in Windows friendly format.
allegro-5.2.8.0.tar.gz	9.4M	Source code for all platforms in Unix friendly format.
allegro-5.2.8.0.7z	8.1M	Source code for all platforms in Windows friendly format.

3) No caso do Windows clique em **Allegro 5.2 binary packages**

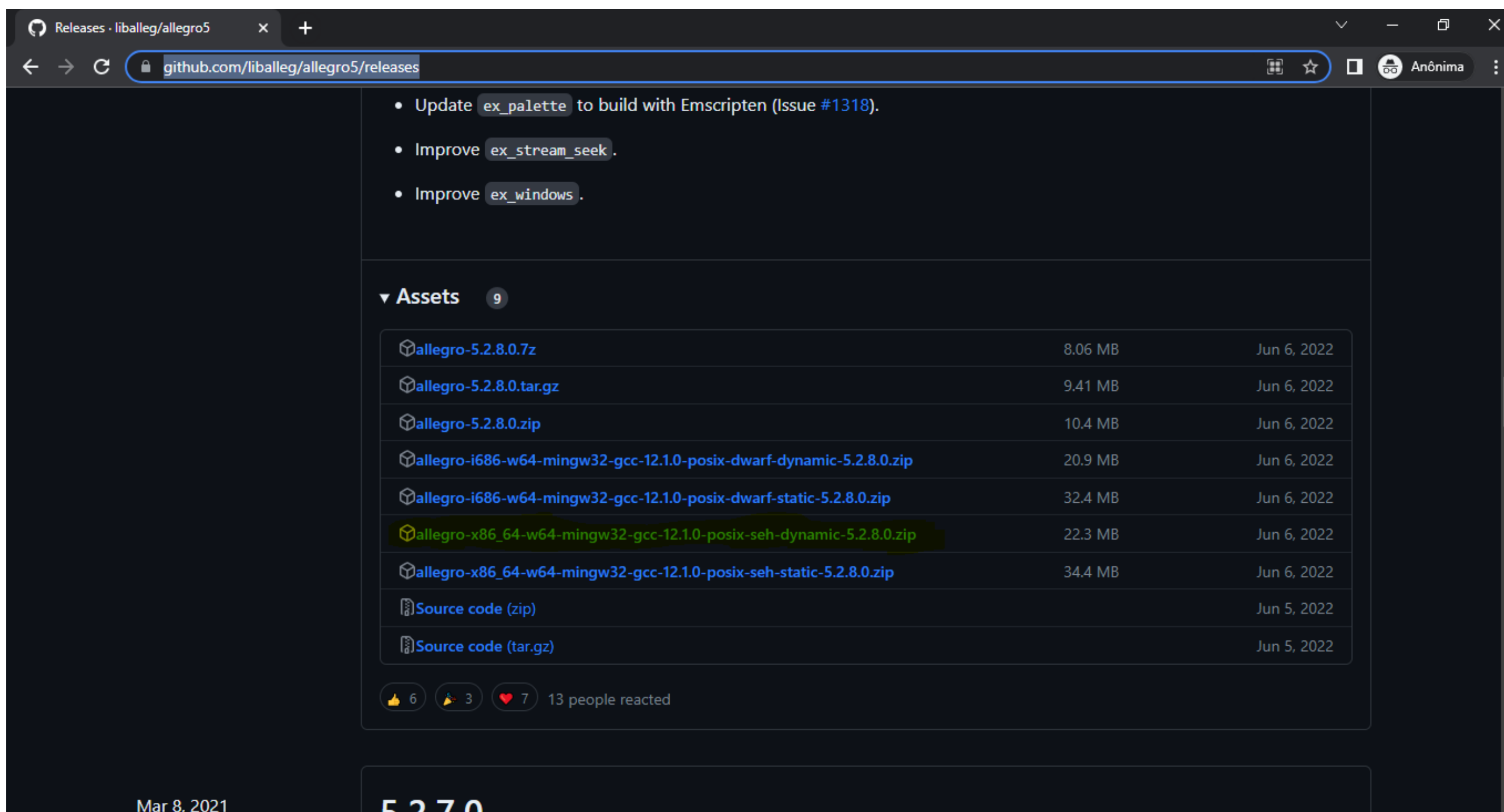


The screenshot shows a web browser window with the address bar displaying `liballeg.org/download.html#windows`. The page title is "Allegro - Download - Latest version". The main heading is "Windows Binaries". Below the heading, there is a paragraph: "There are several official and unofficial ways to obtain Windows binaries of Allegro releases. See the [wiki tutorial](#) for installation instructions."

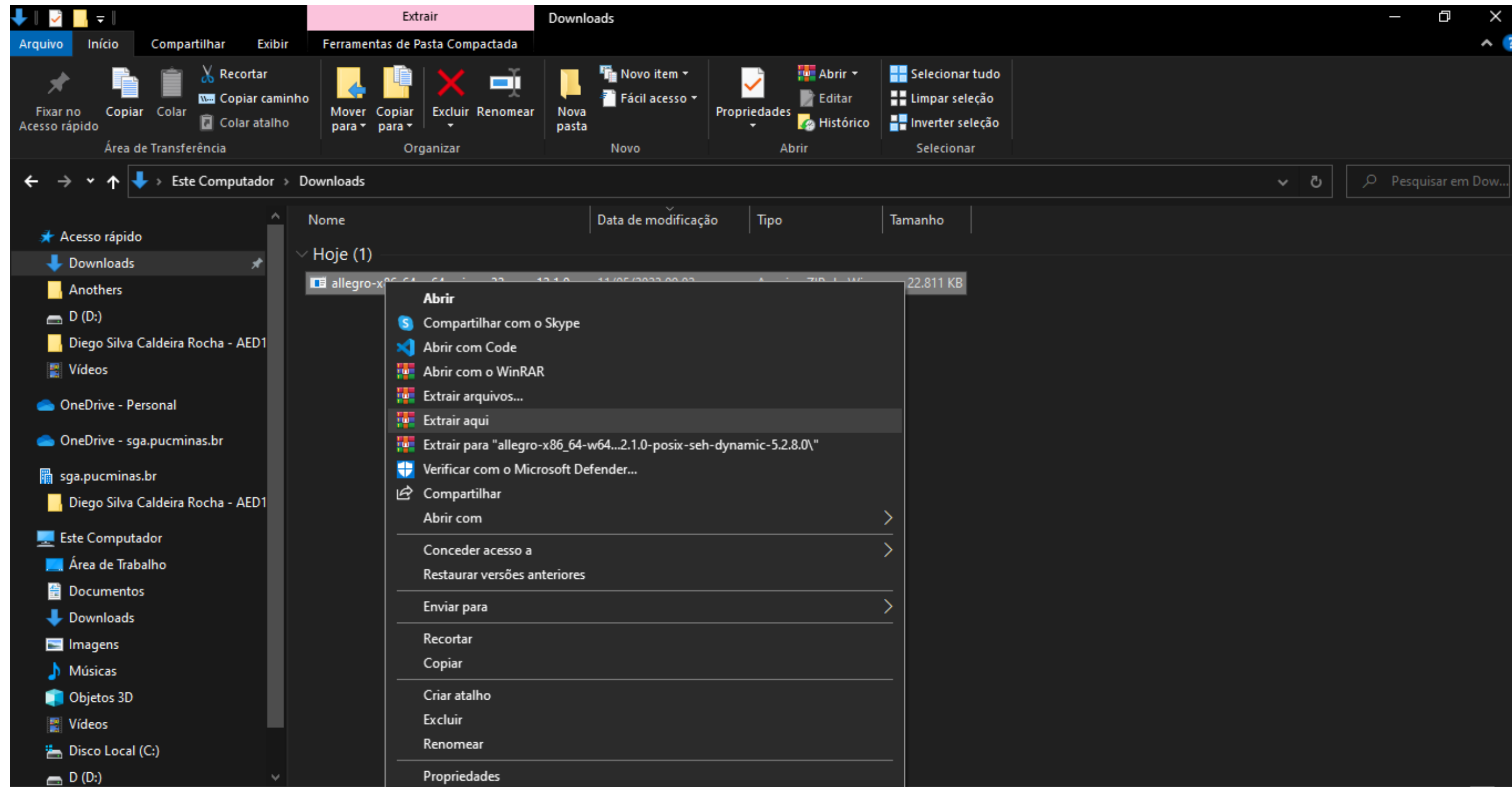
Site	Description
Nuget	The Allegro package on Nuget contains the official binaries in a format that is easy to use from MSVC.
Allegro 5.2 binary packages	Allegro releases on GitHub also come with several binary distributions of Allegro for Windows.
Allegro 5 dependencies	The <code>allegro_winpkg</code> GitHub project has releases that come with the pre-compiled Allegro dependencies for Windows.
www.allegro.cc/files	Matthew Levertton hosts Windows binary packages for many different compilers. The 5.0 binaries are produced by a different process from the standard build system. Unofficial.
devpaks.org	Michal Molhanec maintains a Devpak repository, where you can get a package of the precompiled library for MinGW and Dev-C++. Unofficial.

Below the table, there are sections for "MacOS X binaries" and "Ubuntu PPA". The "MacOS X binaries" section states: "Allegro 5 can be found in the `allegro` package on [homebrew](#). See the [wiki tutorial](#) for more details." The "Ubuntu PPA" section states: "For Linux distributions based on Ubuntu (and Ubuntu itself) you can download binary packages for Allegro 5 by..."

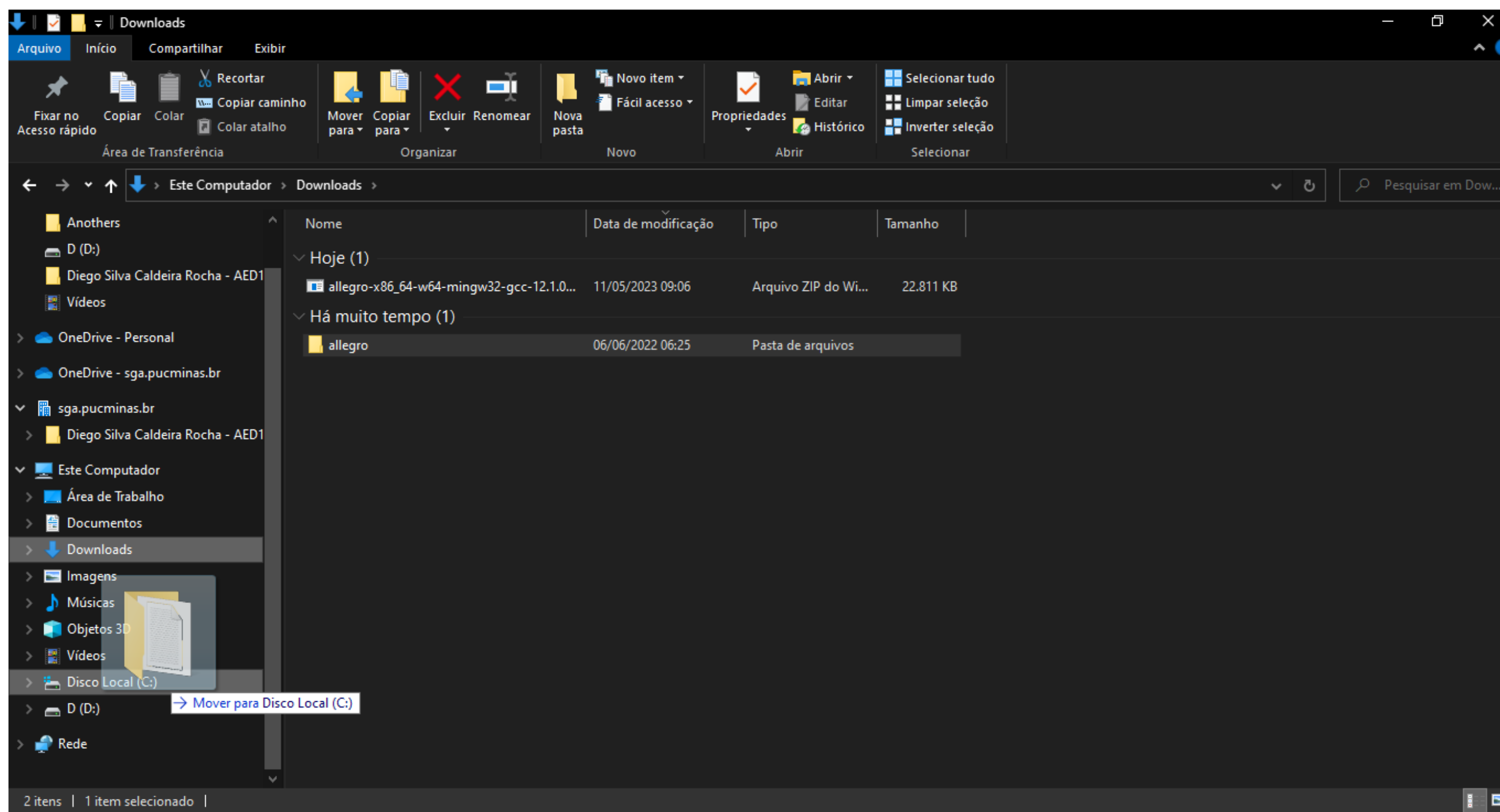
4) Dessa até o meio da página na aba Assets e baixe a versão marcada abaixo (ou clique na imagem)



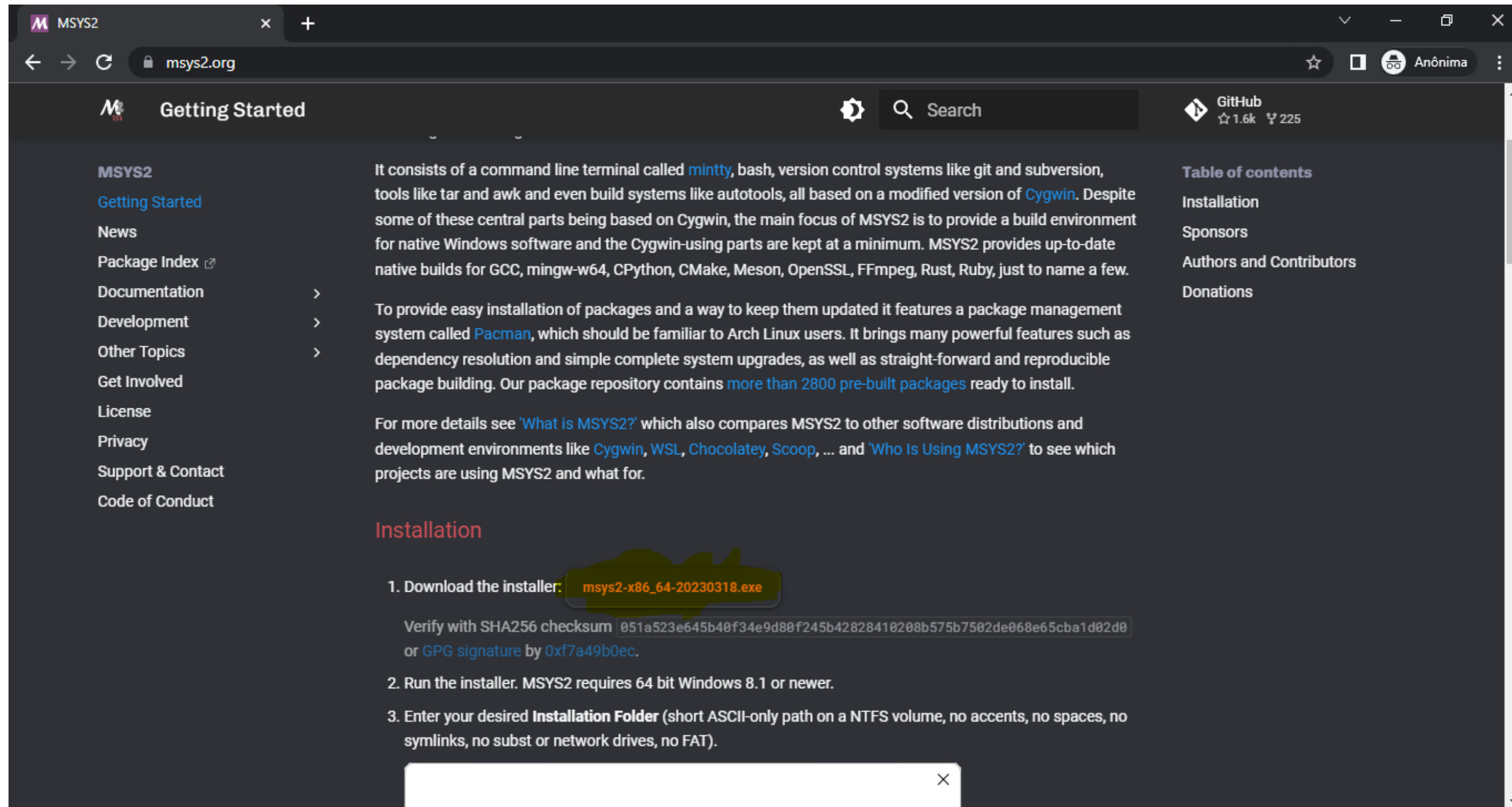
5) Extraia ou descompacte o arquivo que foi baixado



6) Mova allegro pra uma pasta que você se lembrará recomendamos que seja na pasta raiz: Disco Local (C:)



7) Acesse a página principal do MSYS2, faça o download e execute o instalador

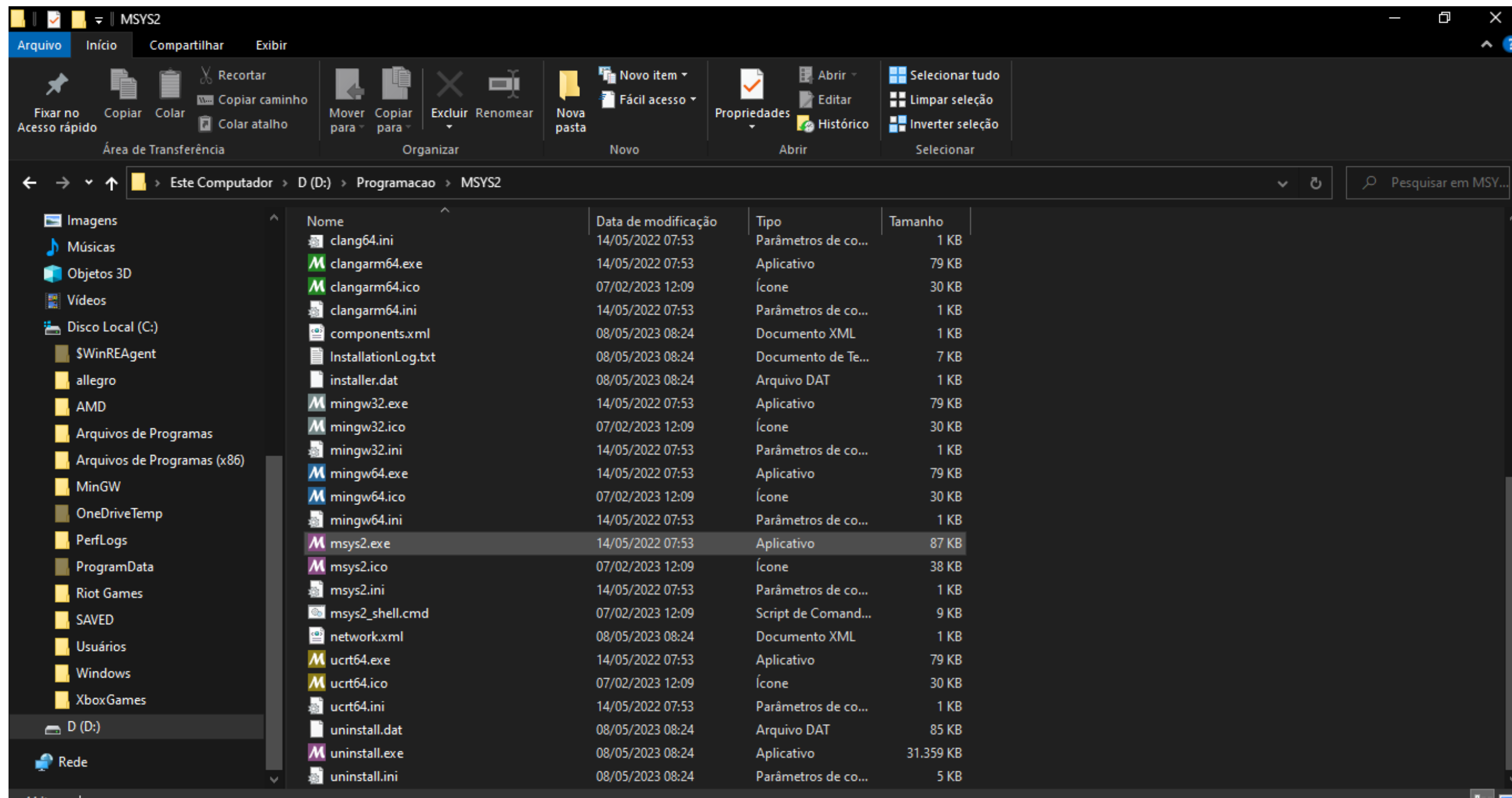


8) Com o MSYS2 aberto execute o seguinte comando no terminal gerado por ele: pacman -Syu

A screenshot of a terminal window with a black background. The window title bar shows a yellow icon with a white 'M' and a minus sign. The terminal text shows the prompt 'pedro@DESKTOP-E79SRV9 UCRT64 ~' in green and purple, followed by the command '\$ pacman -Syu' in white. A vertical scrollbar is visible on the right side of the terminal area.

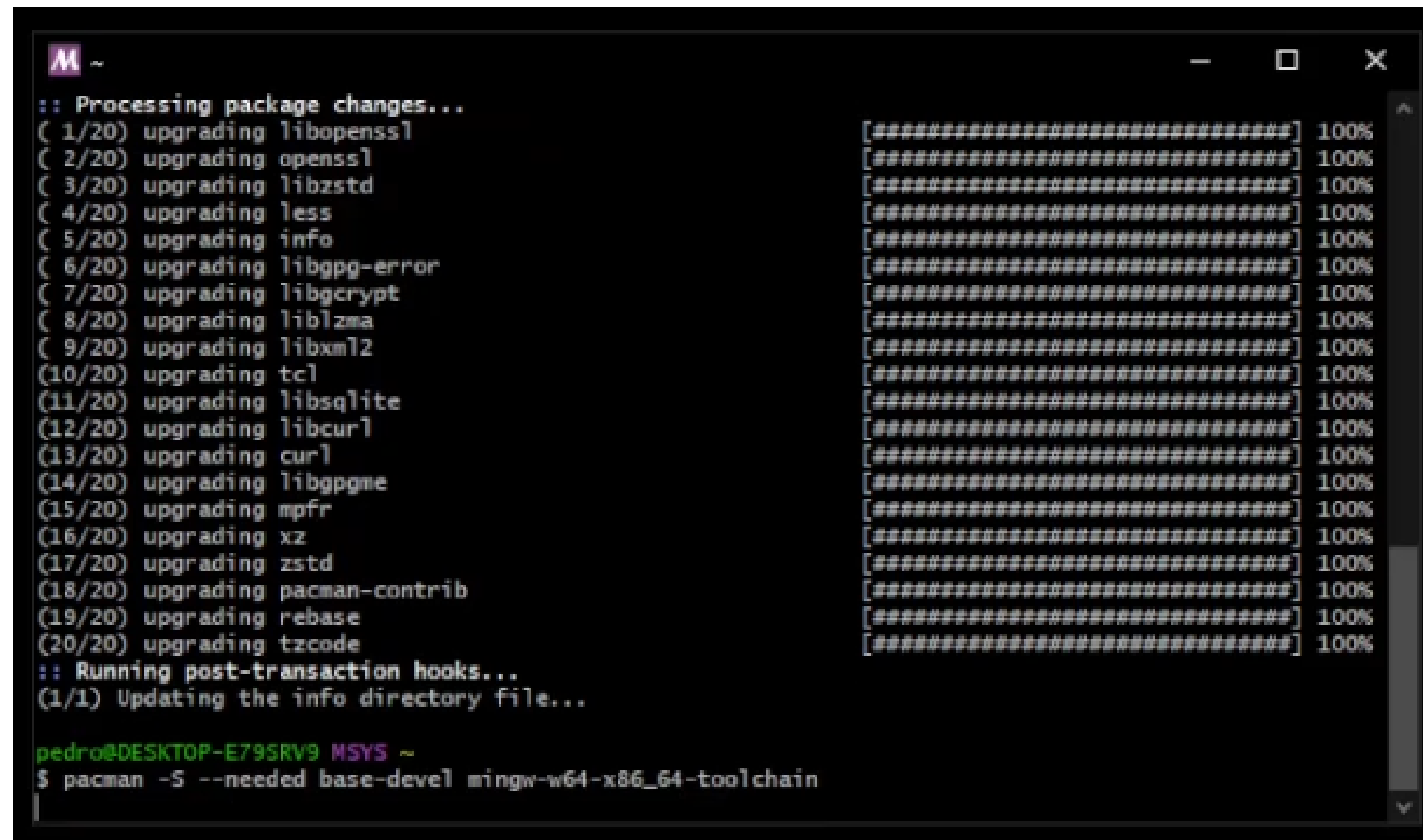
```
M -
pedro@DESKTOP-E79SRV9 UCRT64 ~
$ pacman -Syu
|
```

9) Vá até onde você baixou o MSYS2 e execute o msys2.exe novamente



10) Agora execute esses dois comandos em ordem:

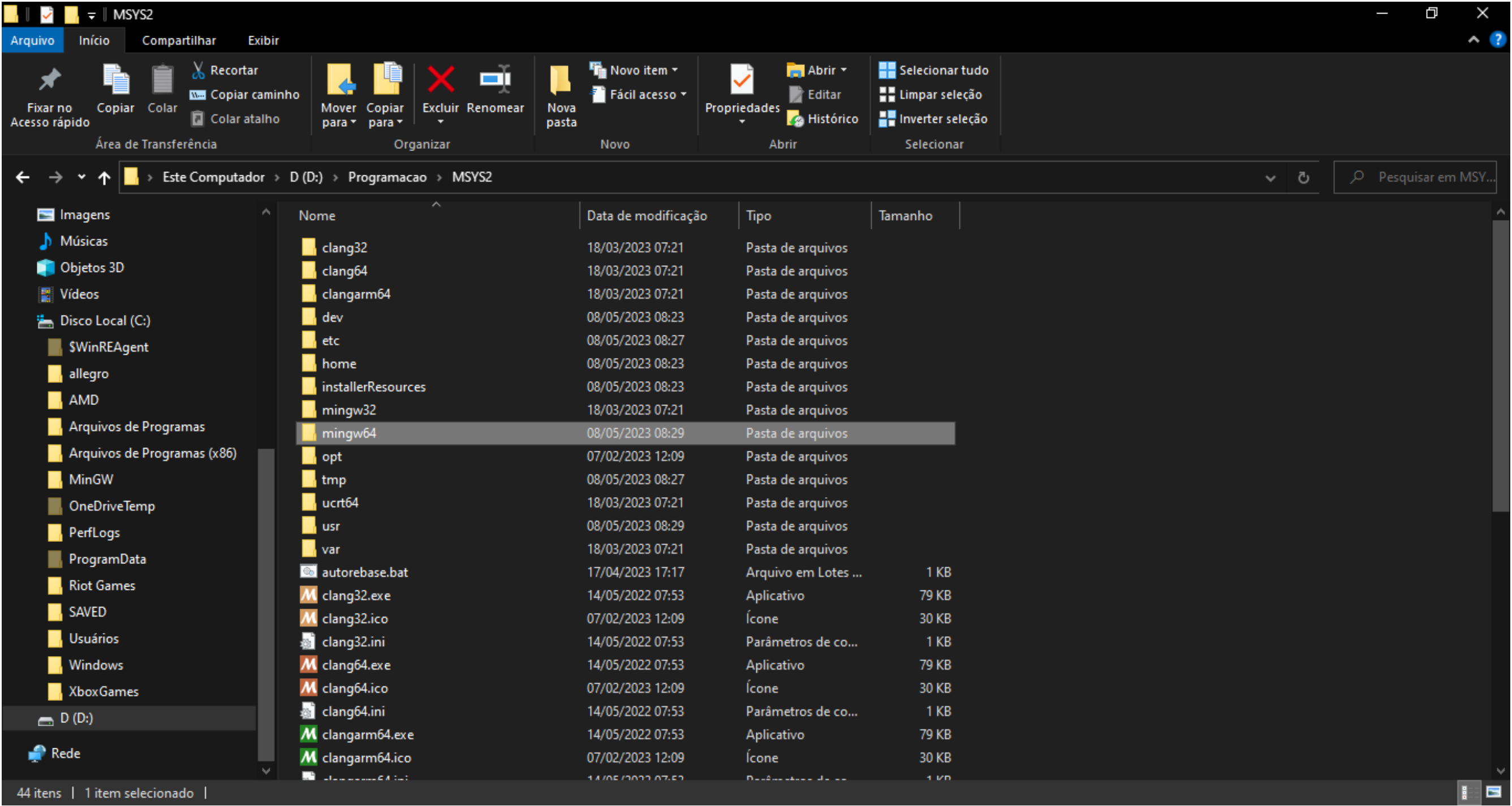
- **pacman -Su**
- **pacman -S --needed base-devel mingw-w64-x86_64-toolchain**



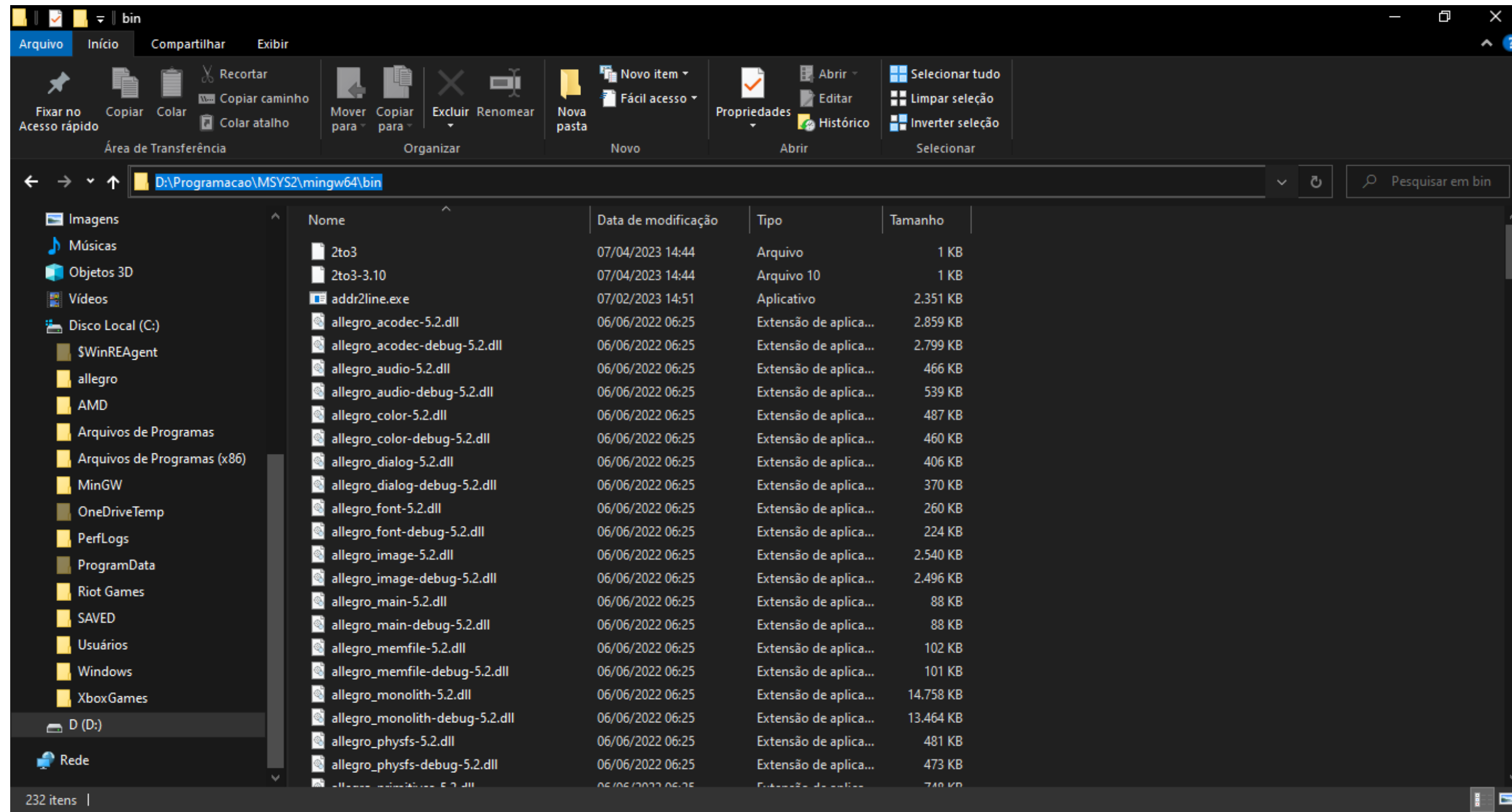
```
M ~
:: Processing package changes...
( 1/20) upgrading libopenssl [#####] 100%
( 2/20) upgrading openssl [#####] 100%
( 3/20) upgrading libzstd [#####] 100%
( 4/20) upgrading less [#####] 100%
( 5/20) upgrading info [#####] 100%
( 6/20) upgrading libgpg-error [#####] 100%
( 7/20) upgrading libgcrypt [#####] 100%
( 8/20) upgrading liblzma [#####] 100%
( 9/20) upgrading libxml2 [#####] 100%
(10/20) upgrading tcl [#####] 100%
(11/20) upgrading sqlite [#####] 100%
(12/20) upgrading libcurl [#####] 100%
(13/20) upgrading curl [#####] 100%
(14/20) upgrading libgpgme [#####] 100%
(15/20) upgrading mpfr [#####] 100%
(16/20) upgrading xz [#####] 100%
(17/20) upgrading zstd [#####] 100%
(18/20) upgrading pacman-contrib [#####] 100%
(19/20) upgrading rebase [#####] 100%
(20/20) upgrading tzcode [#####] 100%
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

pedro@DESKTOP-E79SRV9 MSYS ~
$ pacman -S --needed base-devel mingw-w64-x86_64-toolchain
```

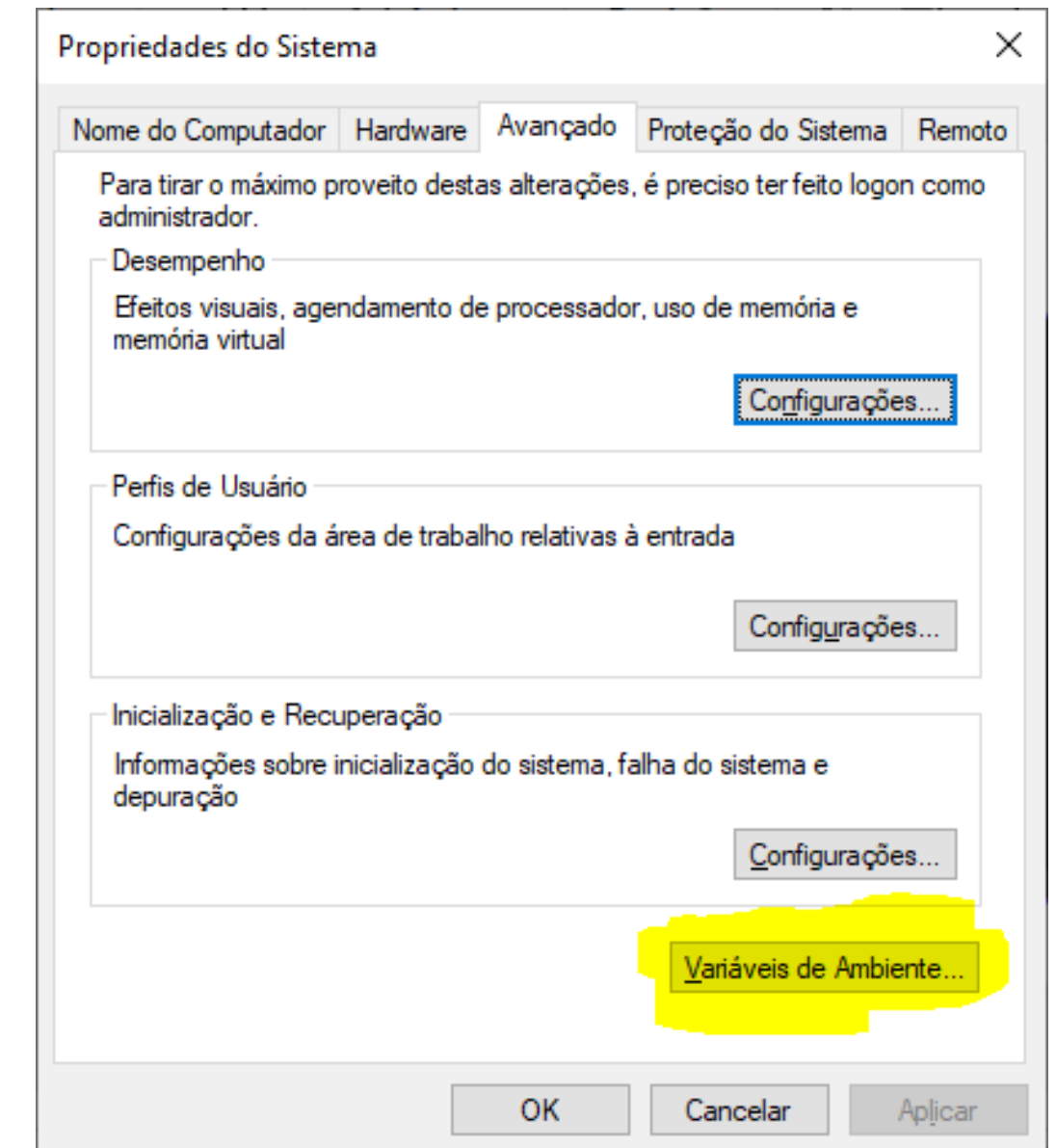
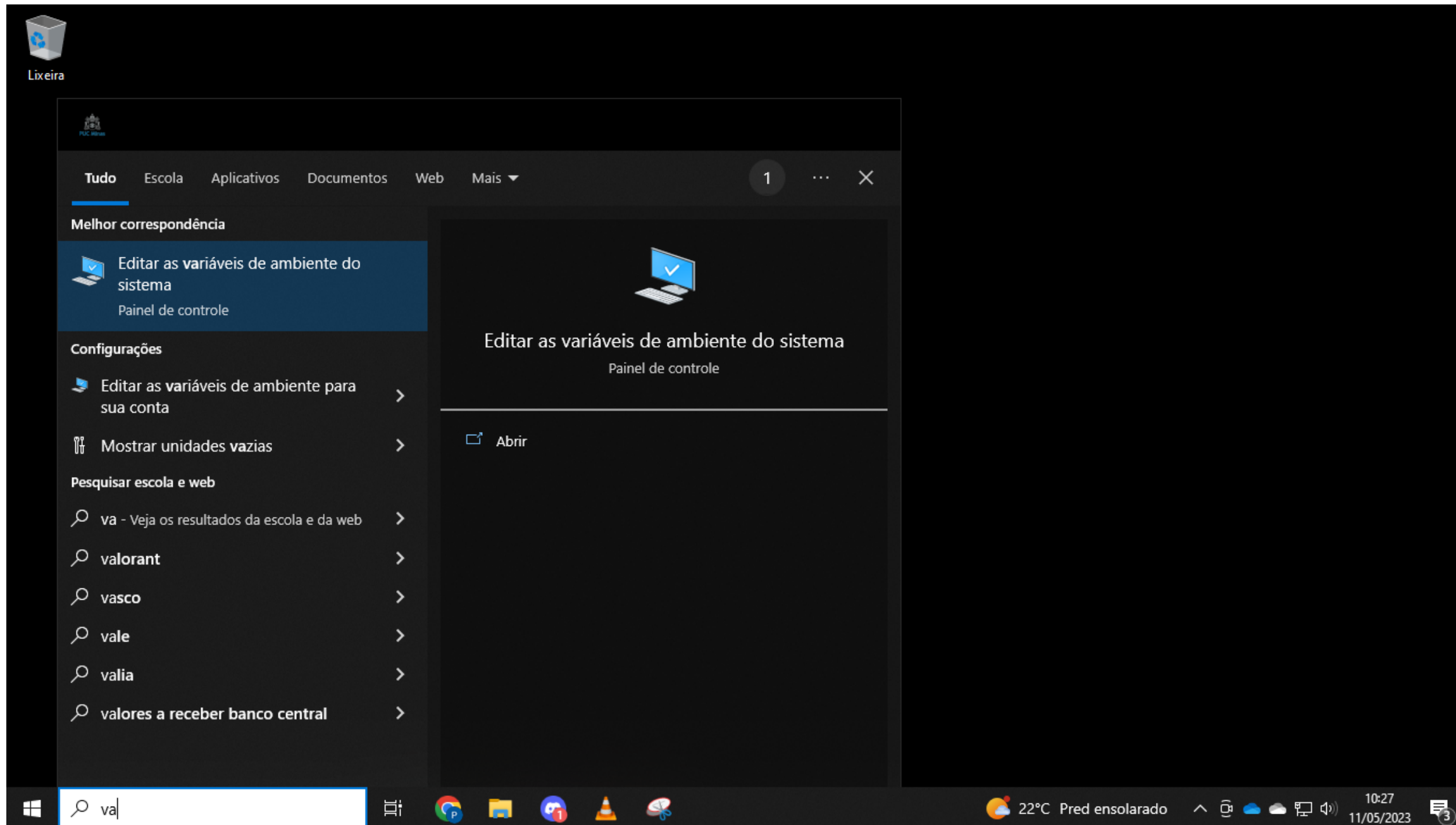
11) Perceba que foi gerada uma nova pasta no MSYS2 chamada mingw64



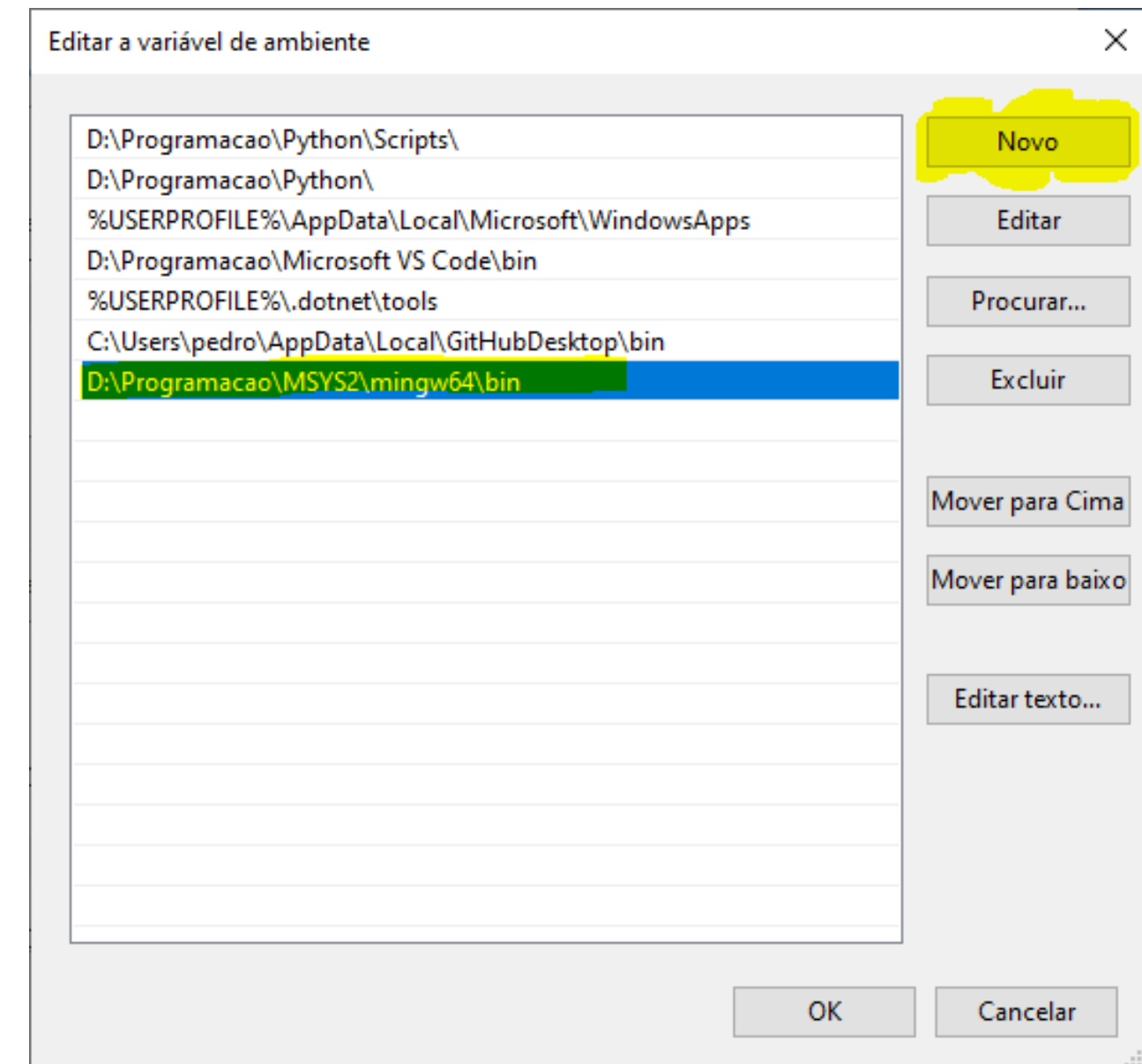
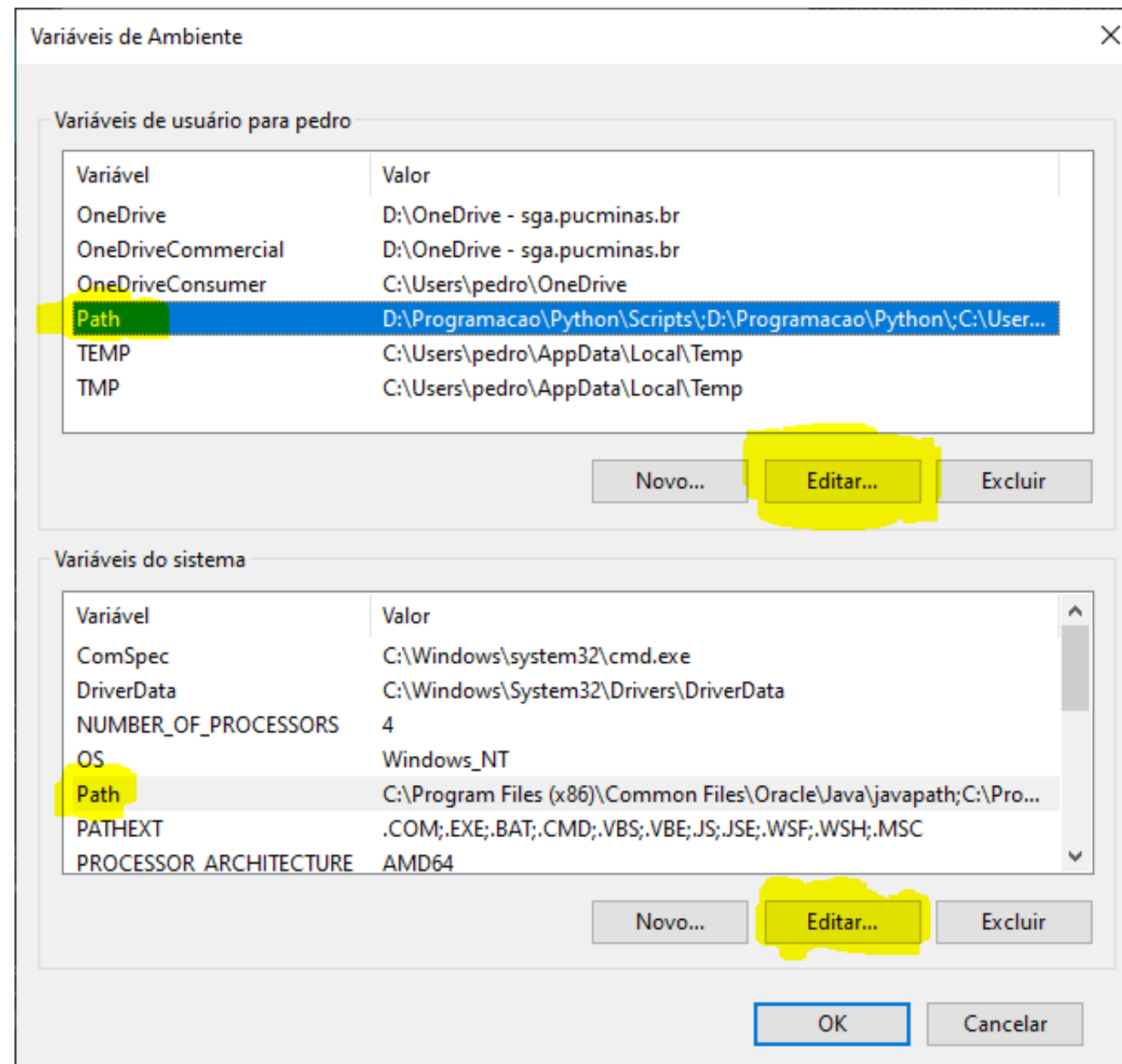
12) Copie o caminho da pasta "bin" que esta dentro de mingw64



13) Abra as variáveis de ambiente

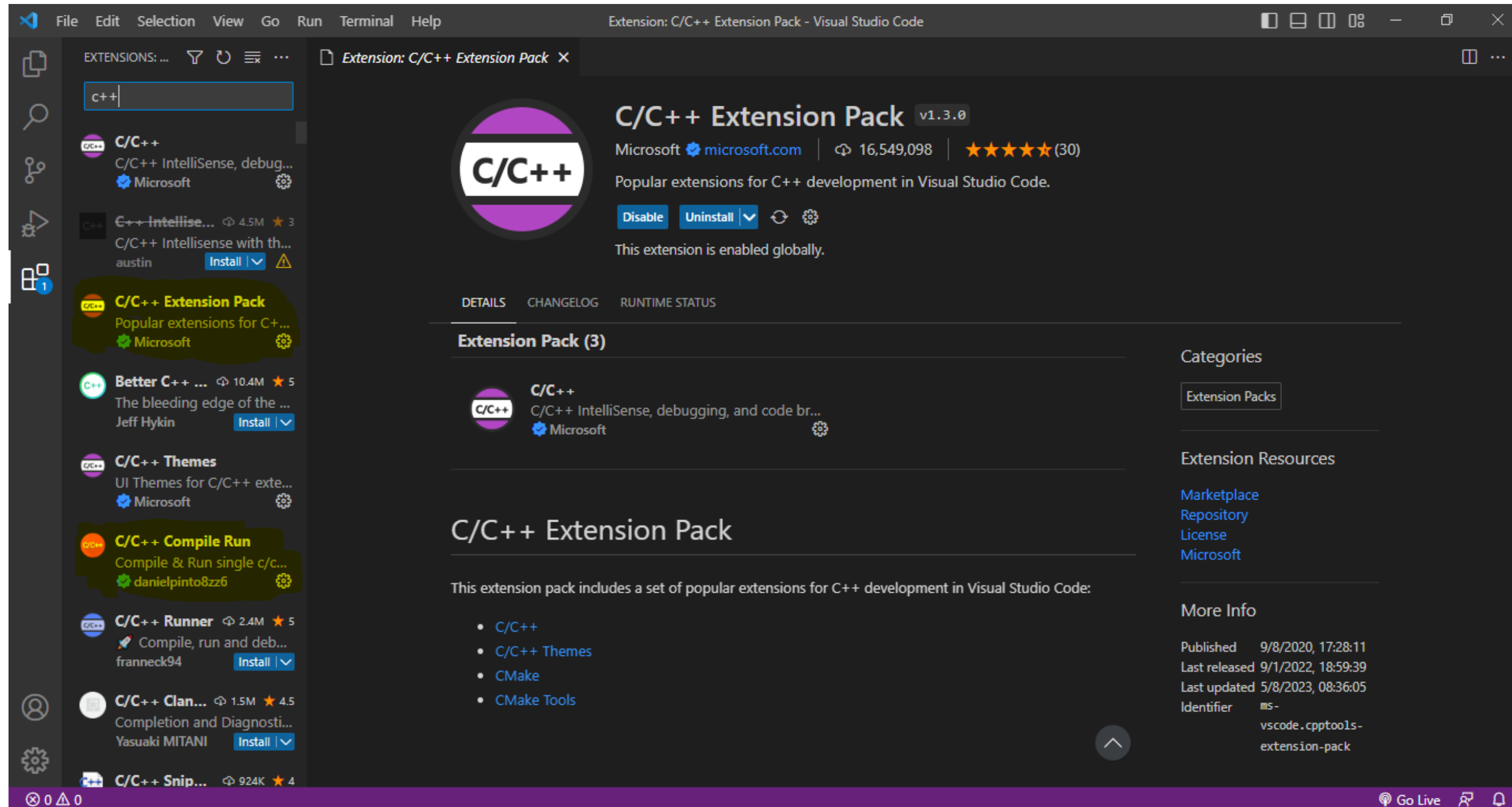


14) Em ambos os "path" separadamente você deve ir em "editar" e logo em seguida em "novo" colando o caminho da pasta "bin" copiado anteriormente

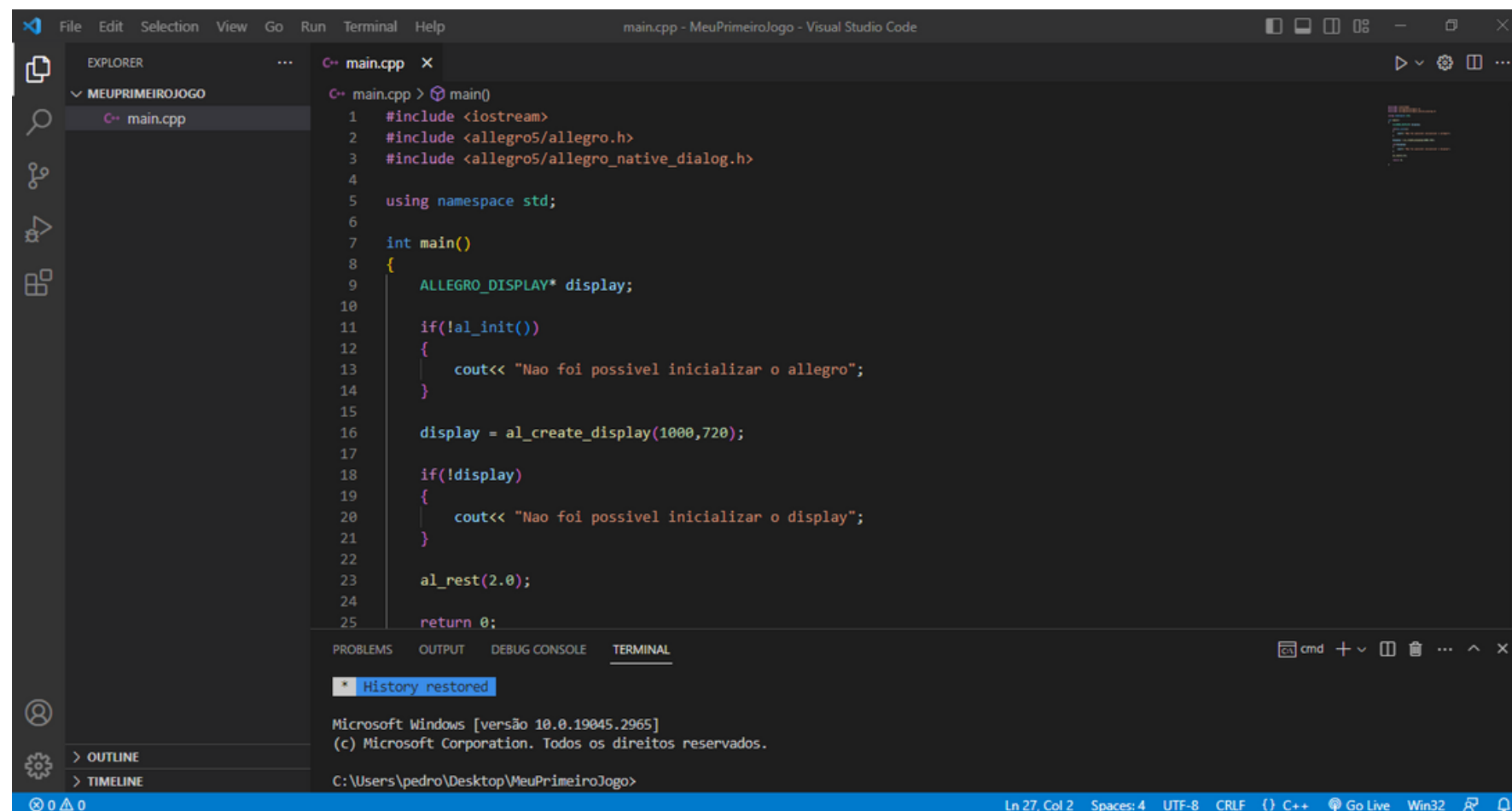


15) Abra o VS Code e instale as extensões:

- C/C++ Extension Pack
- C/C++ Compile Run



16) Crie uma pasta vazia e abra ela com o VS Code, após isso crie um arquivo "main.cpp" e cole nele o código ao lado



```
#include <iostream>
#include <allegro5/allegro.h>
#include <allegro5/allegro_native_dialog.h>

using namespace std;

int main()
{
    ALLEGRO_DISPLAY* display;

    if(!al_init())
    {
        cout<< "Nao foi possivel inicializar o allegro";
    }

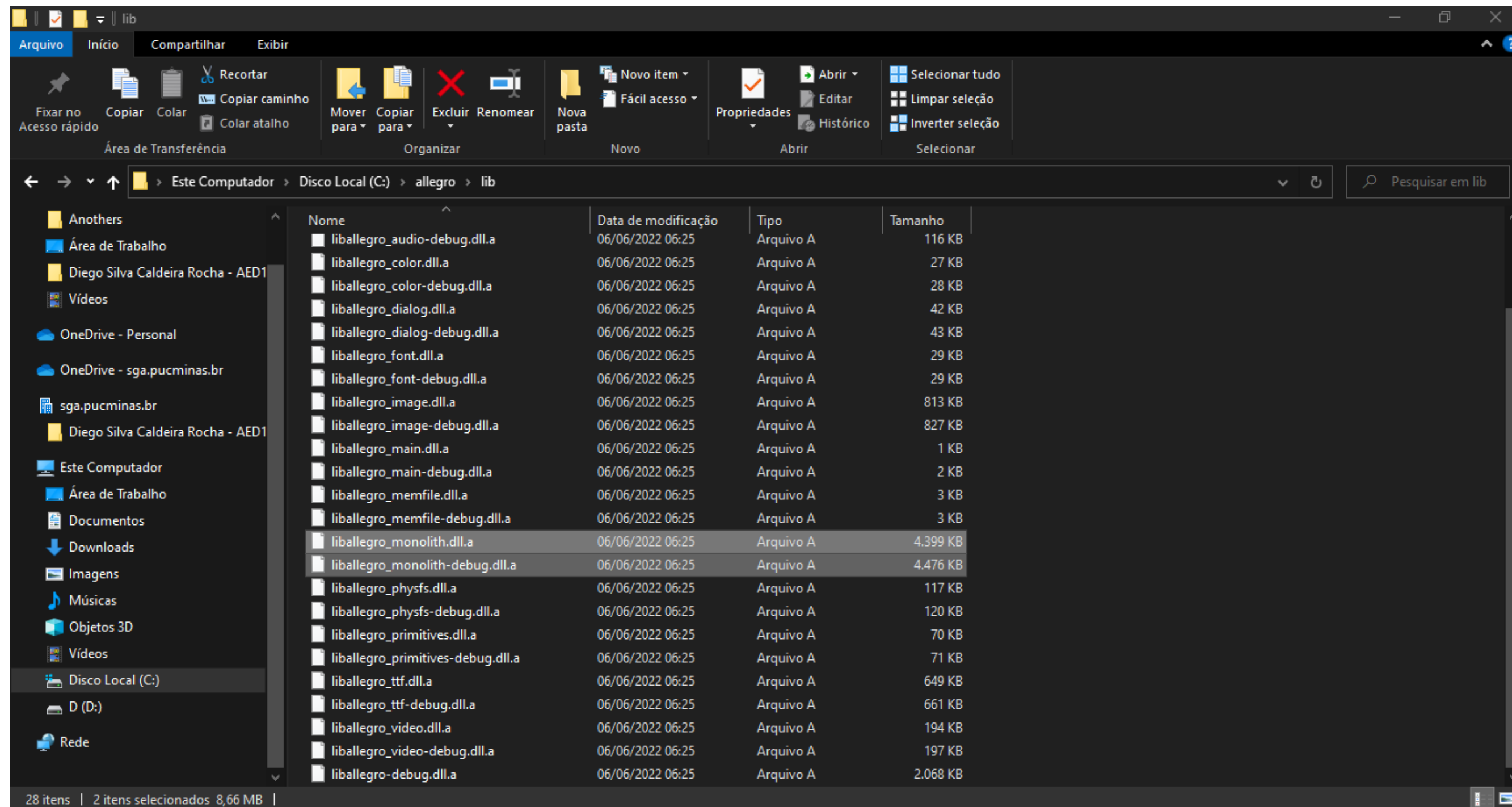
    display = al_create_display(1000,720);

    if(!display)
    {
        cout<< "Nao foi possivel inicializar o display";
    }

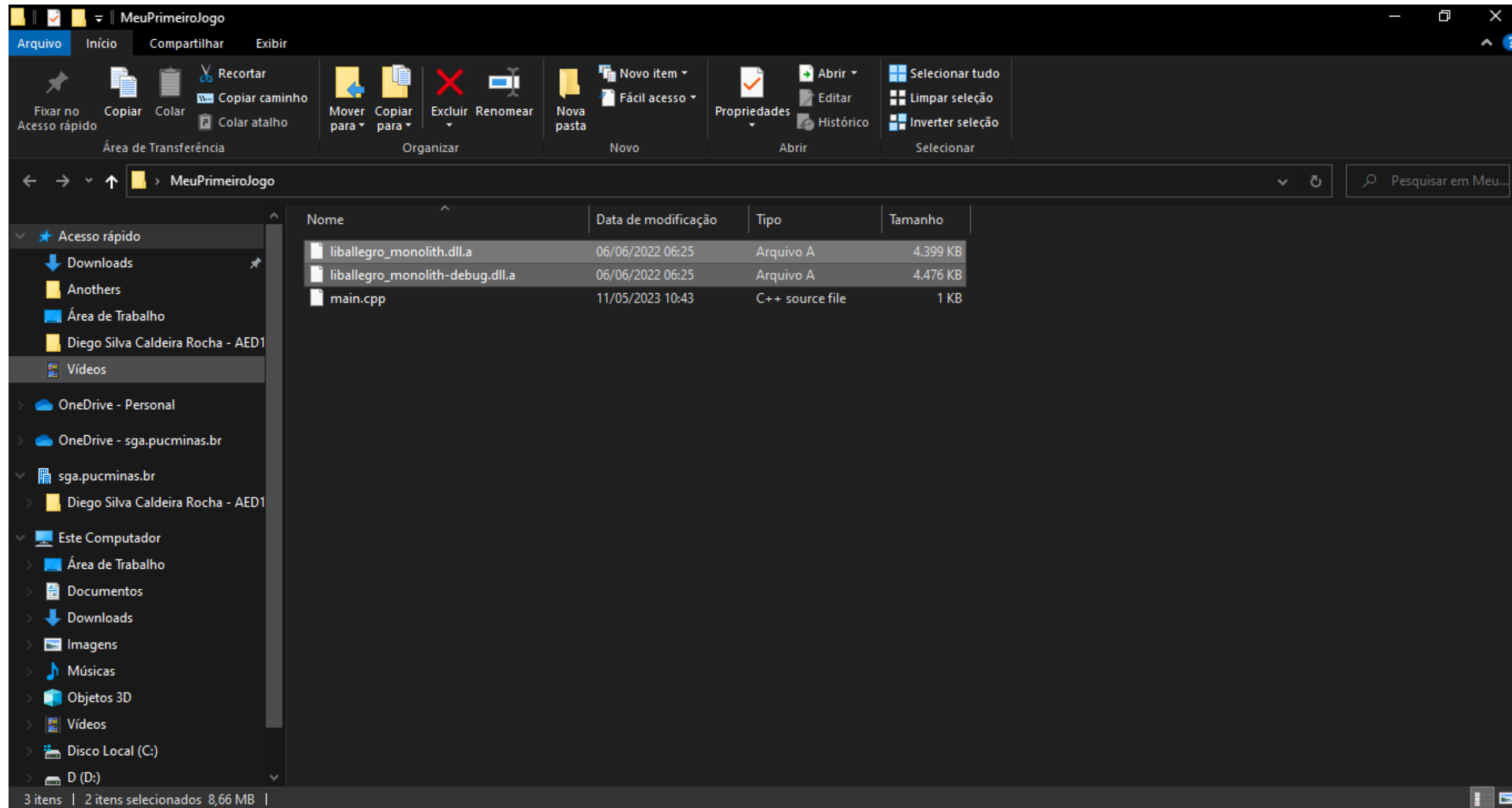
    al_rest(2.0);

    return 0;
}
```

17) Vá até a pasta "lib" que esta dentro da pasta "allegro" e copie os arquivos "liballegro_monolith.dll.a" e "liballegro_monolith-debug.dll.a"



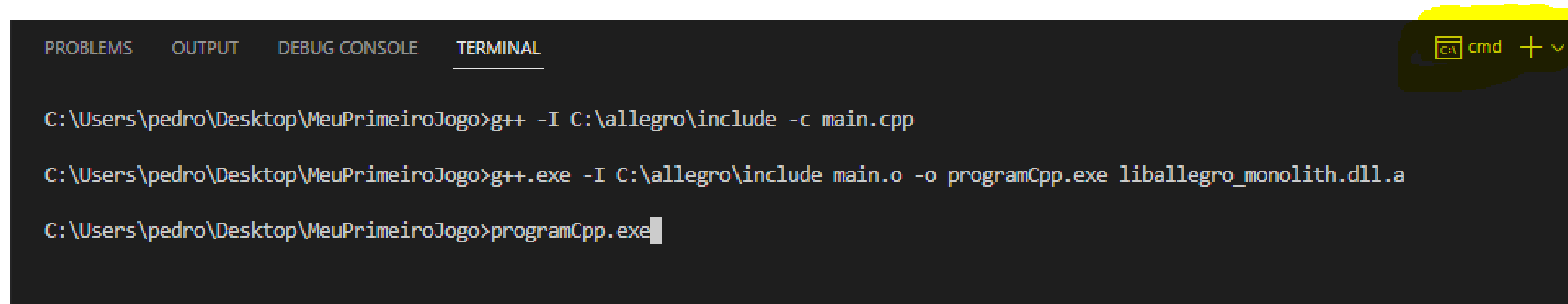
18) Cole os arquivos copiados no passo anterior na pasta do projeto que criamos a pouco



19) Abra um novo terminal no VS Code, selecione "cmd" e execute as seguintes linhas de comando no terminal

g++ -I C:\allegro\include -c main.cpp

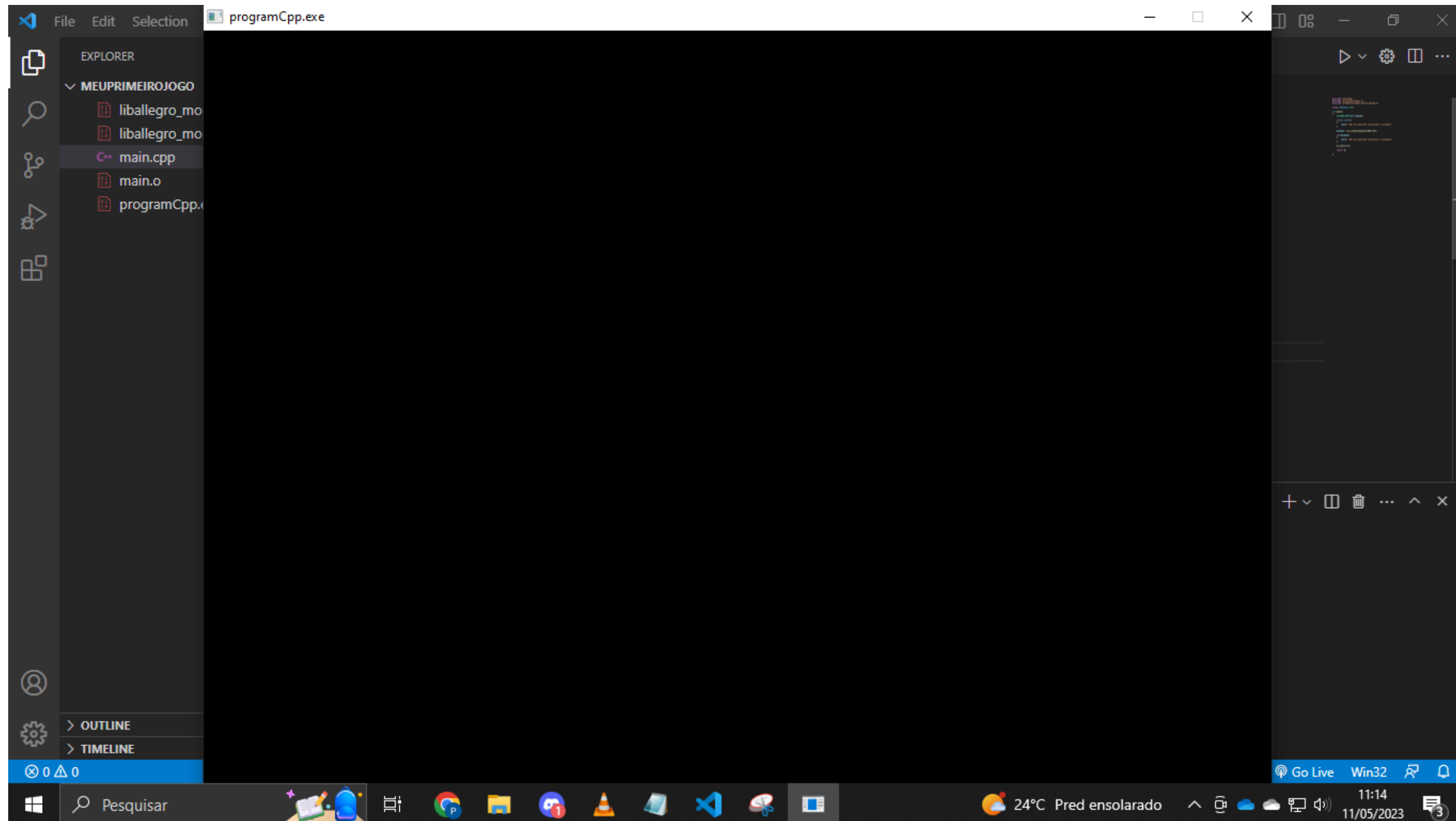
**g++.exe -I C:\allegro\include main.o -o programCpp.exe liballegro_monolith.dll.a
programCpp.exe**



The screenshot shows a VS Code terminal window with the 'TERMINAL' tab selected. The terminal title bar indicates it is a 'cmd' window. The terminal content shows the following commands and their execution paths:

```
C:\Users\pedro\Desktop\MeuPrimeiroJogo>g++ -I C:\allegro\include -c main.cpp  
C:\Users\pedro\Desktop\MeuPrimeiroJogo>g++.exe -I C:\allegro\include main.o -o programCpp.exe liballegro_monolith.dll.a  
C:\Users\pedro\Desktop\MeuPrimeiroJogo>programCpp.exe
```

20) Se aberto um display preto que fecha logo em seguida, parabéns você concluiu seu primeiro projeto em Allegro!



Comandos

MSYS2

- **pacman -Syu**
- **pacman -Su**
- **pacman -S --needed base-devel mingw-w64-x86_64-toolchain**

Terminal VS Code

```
g++ -I C:\allegro\include -c main.cpp  
g++.exe -I C:\allegro\include main.o -o programCpp.exe liballegro_monolith.dll.a  
programCpp.exe
```

```
#include <iostream>  
#include <allegro5/allegro.h>  
#include <allegro5/allegro_native_dialog.h>  
  
using namespace std;  
  
int main()  
{  
    ALLEGRO_DISPLAY* display;  
  
    if(!al_init())  
    {  
        cout<< "Nao foi possivel inicializar o allegro";  
    }  
  
    display = al_create_display(1000,720);  
  
    if(!display)  
    {  
        cout<< "Nao foi possivel inicializar o display";  
    }  
  
    al_rest(2.0);  
  
    return 0;  
  
}
```