

Instalar sdl no windows
2023
nano



Obtenha a versão estável atual do SDL 2.26.3

Principal

Sobre

Insetos

Licenciamento

Créditos

Opinião

Documentação

Wiki

Fóruns

Listas de discussão

Download

Versões SDL

SDL GitHub

Ligações de idioma

Sobre SDL

Simple DirectMedia Layer é uma biblioteca de desenvolvimento de plataforma cruzada projetada para fornecer acesso de baixo nível a áudio, teclado, mouse, joystick e hardware gráfico via OpenGL e Direct3D. Ele é usado por software de reprodução de vídeo, emuladores e jogos populares, incluindo o premiado catálogo da Valve e muitos jogos do Humble Bundle .

O SDL suporta oficialmente Windows, macOS, Linux, iOS e Android. O suporte para outras plataformas pode ser encontrado no código-fonte.

O SDL é escrito em C, funciona nativamente com C++ e existem [ligações disponíveis](#) para várias outras linguagens, incluindo C# e Python.

O SDL 2.0 é distribuído sob a [licença zlib](#) . Esta licença permite que você use o SDL livremente em qualquer software.



Feito com SDL: Edna & Harvey: os novos olhos de Harvey



Faça o download das versões ou pelo git

Versão atual da biblioteca

Release 2.26.3 · libSDL-org/SDL

github.com/libSDL-org/SDL/releases/tag/release-2.26.3

c++C++C++br-C.orgC++ VISUAL STUDIODomestikaASSEMBLYss64kernel.orgLinux kernelgccarchlinuxdie.netdevdocscOutros favoritos

Assistir 87Garfo 970Estrela 5k

CódigoProblemas 749Requisições pull 51AçõesSegurançaPercepções

Lançamentos / release-2.26.3

2.26.3

Mais recente

Comparar

slouken

lançou isso 2 semanas atrás · 1069 confirma a principal desde esta versão

release-2.26.3

adf31f6

Esta é uma versão estável de correções de bugs, com as seguintes alterações:




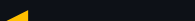











- Corrigido loop infinito desligando controladores WGI
- Corrigida a centralização do D-pad em alguns controladores do Xbox

Ativos 12

SDL2-2.26.3-win32-x64.zip	803 KB	2 semanas atrás
SDL2-2.26.3-win32-x86.zip	707 KB	2 semanas atrás
SDL2-2.26.3.dmg	1.99 MB	2 semanas atrás

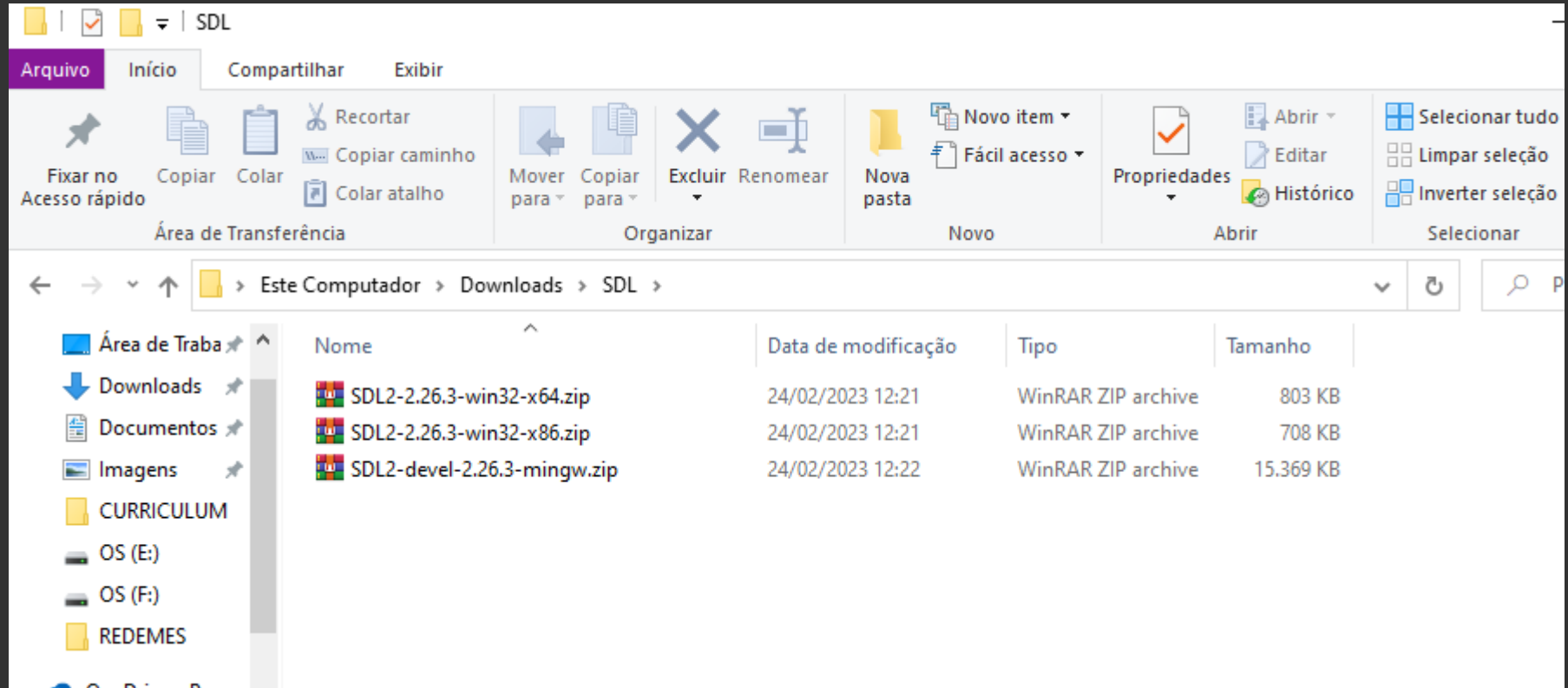
- Corrigido loop infinito desligando controladores WGI
- Corrigida a centralização do D-pad em alguns controladores do Xbox

▼ Ativos 12

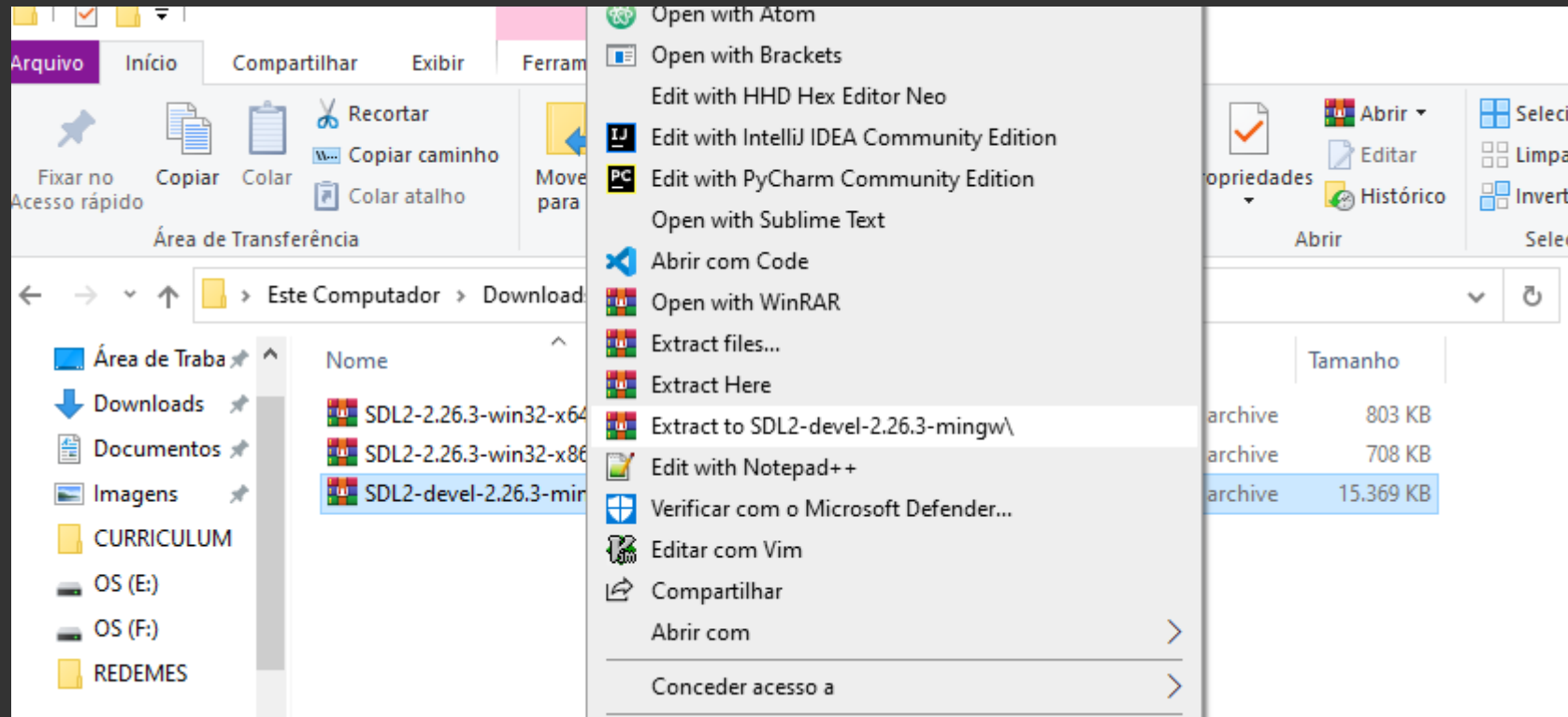
 SDL2-2.26.3-win32-x64.zip		803 KB	2 semanas atrás
 SDL2-2.26.3-win32-x86.zip		707 KB	2 semanas atrás
 SDL2-2.26.3.dmg		1,99 MB	2 semanas atrás
 SDL2-2.26.3.tar.gz		7,71 MB	2 semanas atrás
 SDL2-2.26.3.tar.gz.sig		95 bytes	2 semanas atrás
 SDL2-2.26.3.zip		9MB	2 semanas atrás
 SDL2-2.26.3.zip.sig		95 bytes	2 semanas atrás
 SDL2-devel-2.26.3-mingw.tar.gz		14,9 MB	2 semanas atrás
 SDL2-devel-2.26.3-mingw.zip		15MB	2 semanas atrás
 SDL2-devel-2.26.3-VC.zip		2,58 MB	2 semanas atrás
 Código fonte (fecho eclair)			2 semanas atrás
 Código fonte (tar.gz)			2 semanas atrás

  28  12  14 44 pessoas reagiram

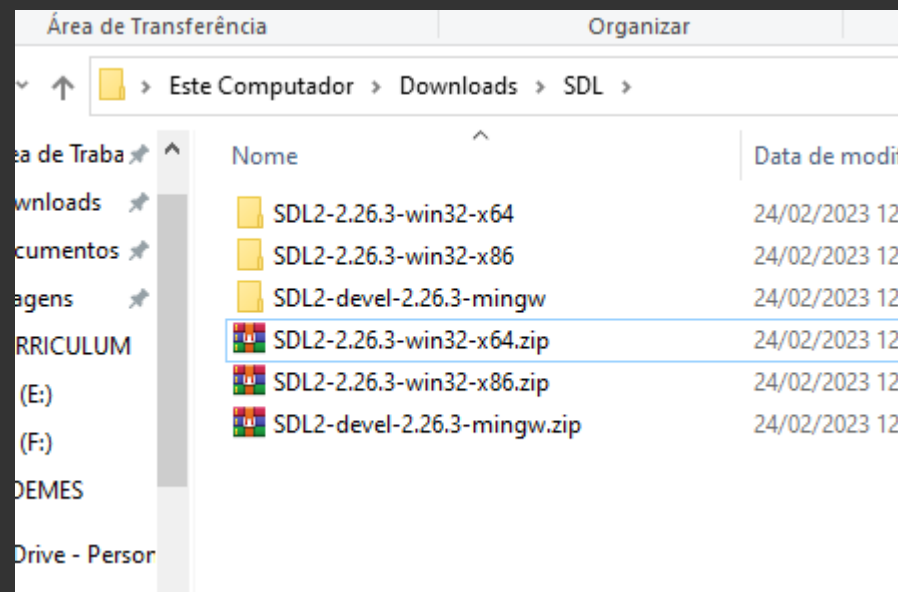
Baixei essas versões Todas .zip



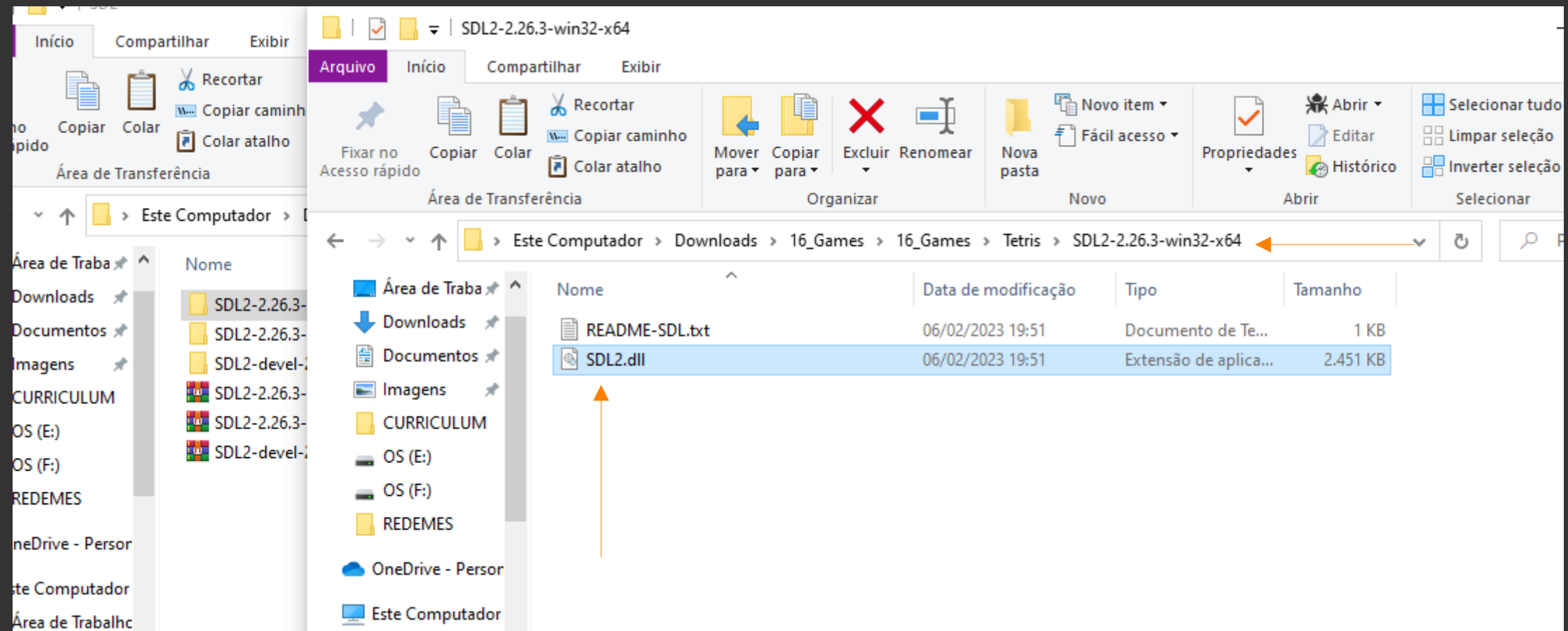
Extraia os arquivos



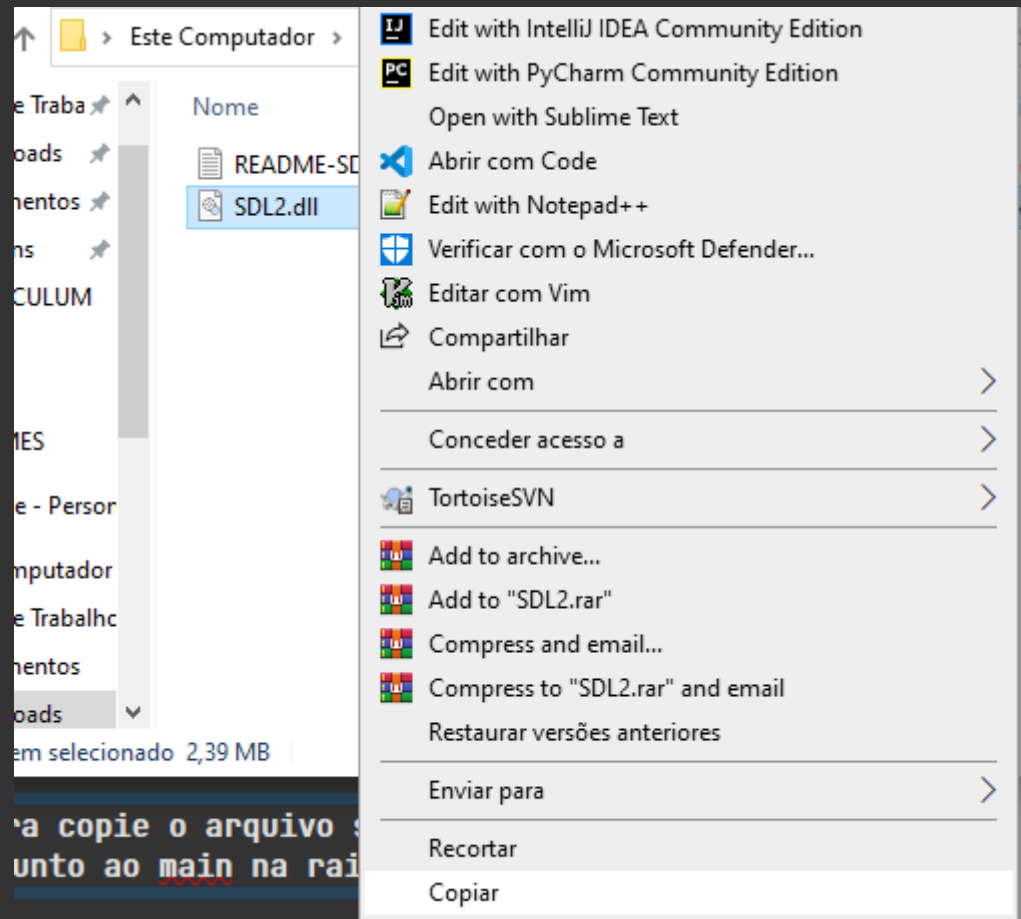
Extraia cada um para seu respectivo pacote



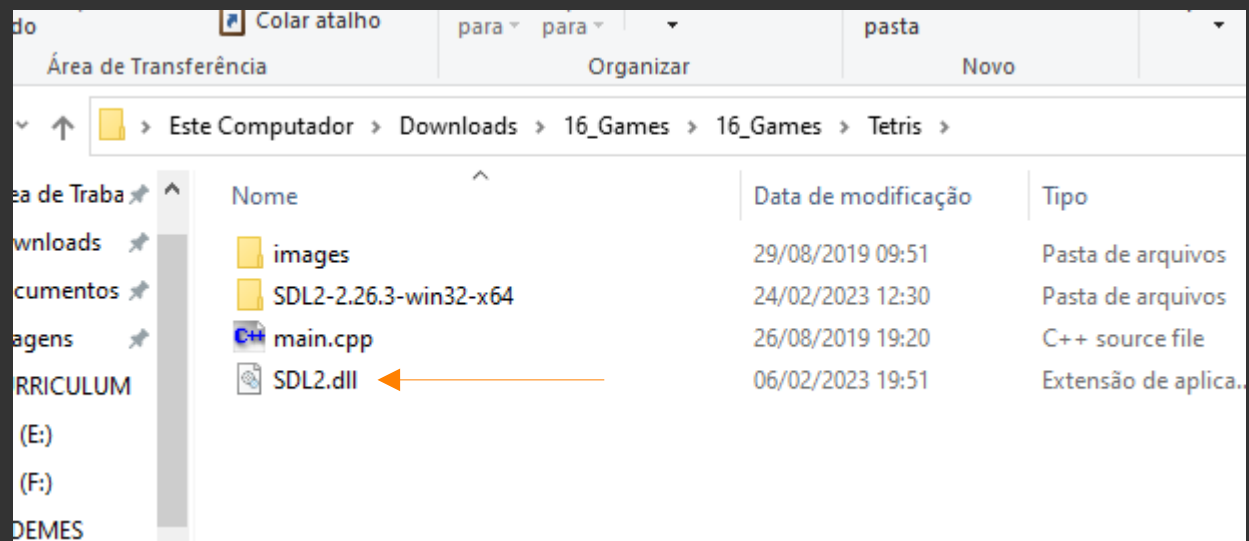
Caso queira usar em um projeto copy alguma dessas pasta e colo junto
ao main.c do projeto
No mesmo diretorio



Agora copie o arquivo sdl2.dll
E cole junto ao main na raiz do projeto

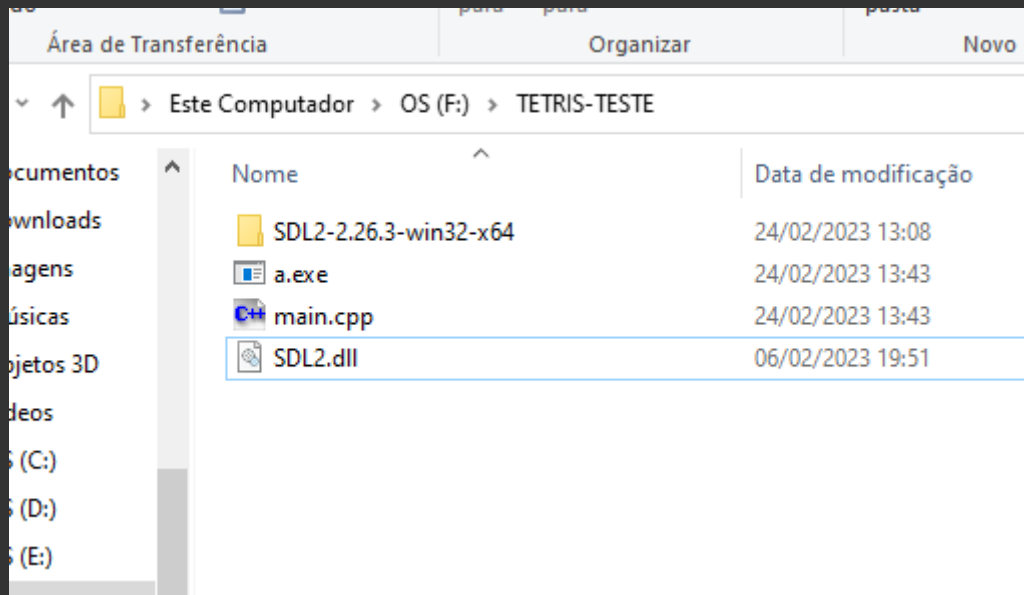


cole



Resolvi seguir esse tutorial e deu certo mais é .cpp
<https://www.youtube.com/watch?v=9IcIKdjSvfc&t=585s>

Estrutura de arquivos



1.0 Colamos a pasta do sdl no projeto

1.1 Colamos o sdl2.dll na raiz

1.2 Damos o comando na linha de comando para gerar o a.exe

1.3 Executamos o a.exe

F:\> mkdir TETRIS-TESTE

Directory: F:\

Mode	LastWriteTime	Length	Name
d----	24/02/2023 13:00		📁 TETRIS-TESTE

F:\> CD TETRIS-TESTE

TETRIS-TESTE> ls

TETRIS-TESTE> vim main.cpp

TETRIS-TESTE> cat main.cpp

```
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_timer.h>

int main(int argc, char *argv[])
{
    returns zero on success else non-zero
    if (SDL_Init(SDL_INIT_EVERYTHING) != 0) {
        printf("error initializing SDL: %s\n", SDL_GetError());
    }

    SDL_Window* win = SDL_CreateWindow("GAME",
                                        SDL_WINDOWPOS_CENTERED,
                                        SDL_WINDOWPOS_CENTERED,
                                        1000, 1000, 0);

    while (1)
        ;
    return 0;
}
```



pwsh in TETRIS-TESTE



TETRIS-TESTE ls

Directory: F:\TETRIS-TESTE

Mode	LastWriteTime		Length	Name
----	-----	-----	-----	----
d----	24/02/2023	13:08		📁 SDL2-2.26.3-win32-x64
-a---	24/02/2023	13:06	976	c++ main.cpp
-a---	06/02/2023	19:51	2509824	📄 SDL2.dll

TETRIS-TESTE

./a.exe

TETRIS-TESTE

cat main.cpp

#include <SDL2/SDL.h>

```
int main(int argc, char *argv[])  
{
```

```
    SDL_Window* window=NULL;  
    window = SDL_CreateWindow(  
        "SDL2 its work!",  
        SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOS_CENTERED,  
        640, 480,  
        SDL_WINDOW_SHOWN  
    );
```

```
    SDL_Renderer * renderer = SDL_CreateRenderer(window, -1, 0);  
    SDL_SetRenderDrawColor(renderer, 9, 20, 33, 255);  
    while(true)  
    {
```

```
        SDL_Event event;  
        while(SDL_PollEvent(&event)){  
            if( event.type == SDL_QUIT ){  
                exit(0);  
            }  
        }
```

```
        SDL_RenderClear(renderer);  
        SDL_RenderPresent(renderer);
```

```
    }  
    SDL_DestroyRenderer(renderer);  
    SDL_DestroyWindow(window);  
    //SDL_QUIT();  
    return 0;  
}
```

brito

TETRIS-TESTE



in pwsh at 13:56:06

main.cpp

buffers

```
>> 1 #include <SDL2/SDL.h>
2
3 int main(int argc, char *argv[])
4 {
5
6     SDL_Window* window=NULL;
7     window = SDL_CreateWindow(
8         "SDL2 its work!",
9         SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOS_CENTERED,
10        640, 480,
11        SDL_WINDOW_SHOWN
12    );
13
14    SDL_Renderer * renderer = SDL_CreateRenderer(window, -1, 0);
15    SDL_SetRenderDrawColor(renderer, 9, 20, 33, 255);
16    while(true)
17    {
18        SDL_Event event;
19        while(SDL_PollEvent(&event)){
20            if( event.type == SDL_QUIT ){
21                exit(0);
22            }
23        }
24        SDL_RenderClear(renderer);
25        SDL_RenderPresent(renderer);
26    }
27    SDL_DestroyRenderer(renderer);
28    SDL_DestroyWindow(window);
29    //SDL_QUIT();
30    return 0;
```

NORMAL main.cpp

cpp

utf-8

41% N:13/31

E:3(L1)

```
return 0;
```

```
}
```

TETRIS-TESTE vim main.cpp

brito TETRIS-TESTE g++ ./main.cpp -I .\SDL2-2.26.3-win32-x64\x86_64-w64-mingw32\include -L .\SDL2-2.26.3-win32-x64-mingw32\lib -lmingw32 -lSDL2main -lSDL2

```
}
```

TETRIS-TESTE vim main.cpp

brito TETRIS-TESTE ./a.exe|

```
#include <SDL2/SDL.h>
```

```
int main(int argc, char *argv[])  
{
```

```
    SDL_Window* window=NULL;  
    window = SDL_CreateWindow(  
        "SDL2 its work!",  
        SDL_WINDOWPOS_CENTERED,  
        640, 480,  
        SDL_WINDOW_SHOWN  
    );
```

```
    SDL_Renderer * renderer = SDL_CreateRenderer(window, -1, SDL_RENDERER_ACCELERATED);  
    SDL_SetRenderDrawColor(renderer, 0, 0, 0, 0);  
    while(true)
```

```
{  
    SDL_Event event;  
    while(SDL_PollEvent(&event))  
        if( event.type == SDL_QUIT )  
            exit(0);  
}
```

```
    SDL_RenderClear(renderer);  
    SDL_RenderPresent(renderer);
```

```
    SDL_DestroyRenderer(renderer);  
    SDL_DestroyWindow(window);  
    //SDL_Quit();  
    return 0;
```

```
}
```

```
TETRIS-TESTE vim main.cpp
```

```
TETRIS-TESTE ./a.exe
```

SDL2 its work!

-

□

x



Tempestade



14:00



FIZ O TESTE NO C
E DEU CERTO
A UNICA COISA FOI QUE
USEI O G++
NÃO SEI SE DA PRA USAR O
GCC

