

# Tutorial de criação de projetos no *Code::Blocks*

Por Matheus Almeida, para Introdução ao Desenvolvimento de Jogos – UnB

*Code::Blocks* é uma IDE, ou seja, é um ambiente de trabalho que auxilia em projetos, mais especificamente, para programação. Por ser Multi Plataforma e compatível com Windows, Linux Mac, e acredite, até *raspberry pi*!

Nesse tutorial será mostrado como usar e criar um projeto feito no *Code::Blocks*.

## Instalação

O tutorial irá apenas ensinar a utilizar, instalar e criar/compilar projetos, não irá abordar a compilação do próprio codeblocks em si.

### Windows e Mac

A forma mais fácil é fazer o download do binário nesse [LINK](#)

**IMPORTANTE:** O code::blocks tem uma versão que já vem com o mingw, é uma versão diferente, e esse mingw pode dar conflito com o mingw atual. O correto é o codeblocks-16.01-setup.exe

### Linux

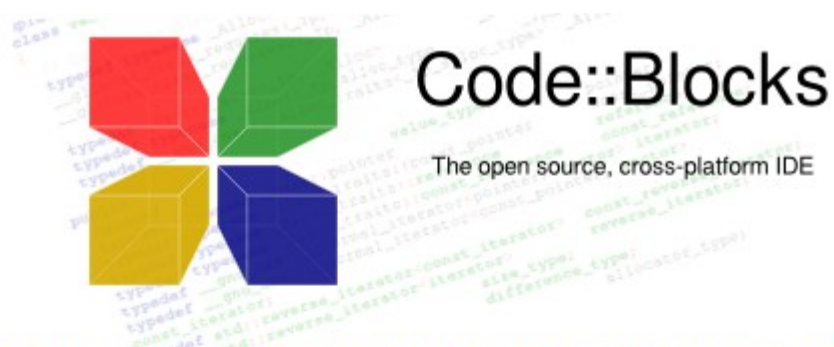
Caso você esteja usando Ubuntu, Fedora ou qualquer versão do Unix que tenha algum tipo de “app store”, basta instalar por lá mesmo.

Mas também tem como baixar nesse [Link](#)

## Abrindo o codeblocks pela primeira vez

Quando for aberto pela primeira vez, ele irá fazer uma checagem rápida procurando por compiladores. Se você usa o Mingw ou o TDM-GCC, ele deve aparecer como a primeira opção. Confirme e espere aparecer a janela, o que pode levar alguns segundos.

## Criando um novo projeto



[Release 13.12 rev 9501 \(2013-12-25 18:25:45\) gcc 4.8.2 Linux/unicode - 6](#)



[Open an existing project](#)



[Tip of the day](#)

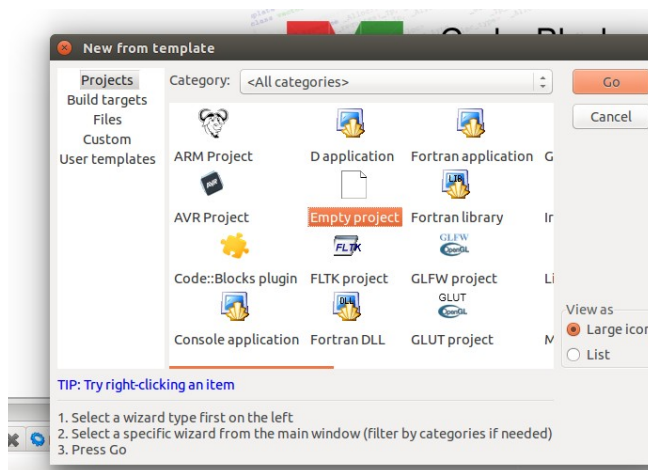


[Visit the Code::Blocks forums](#)

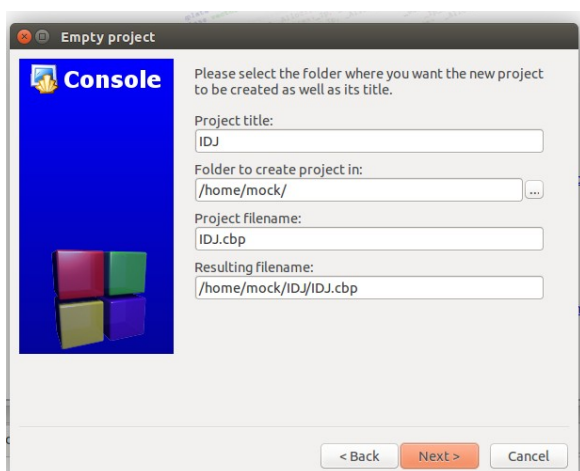
[Report a bug](#)

[Request a new feature](#)

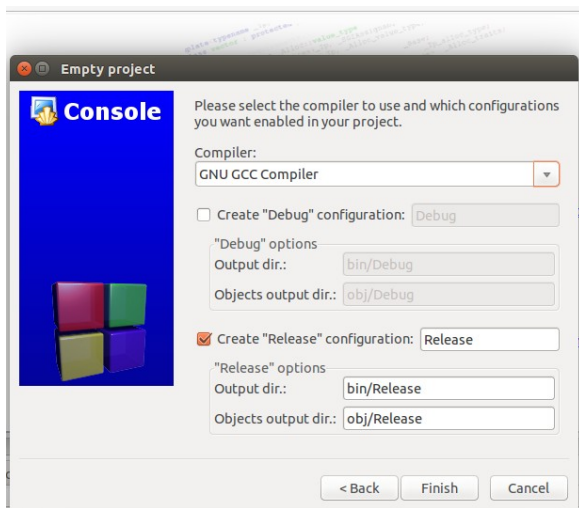
Selezione Empty project



Insira o nome e diretório do projeto



Selecione o compilador, no caso por ser o GGC (mingw ou tdm), deve ser a primeira opção.



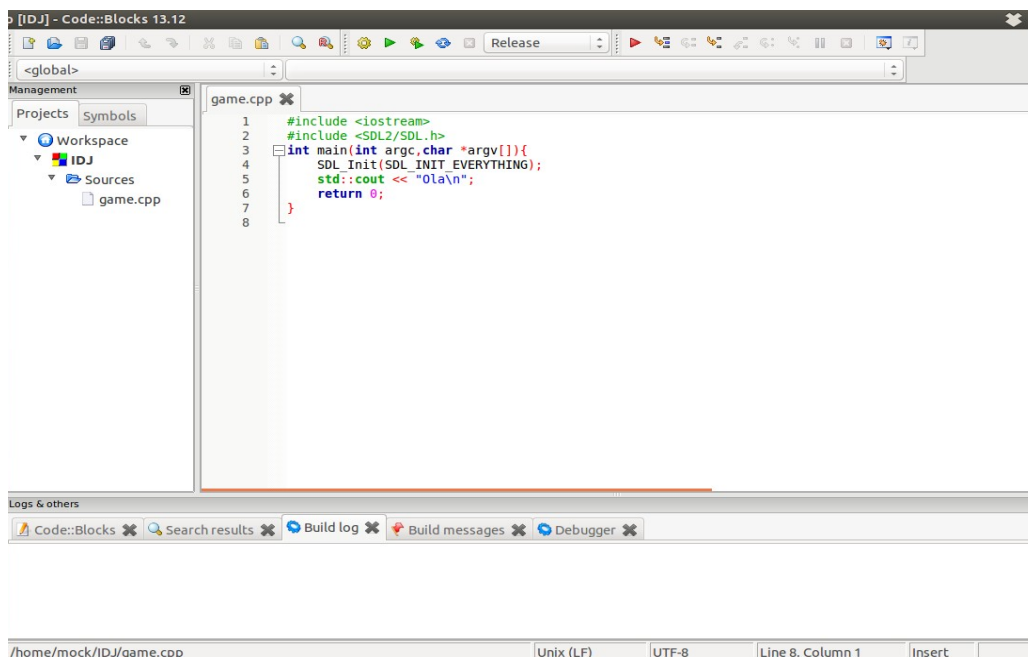
Ele vai criar o projeto, e em seguida, vá no menu do topo e selecione new file. Ele irá perguntar se você quer salvar o arquivo no projeto, selecione sim.

Crie um arquivo chamado game.cpp

Coloque esse conteudo no arquivo:

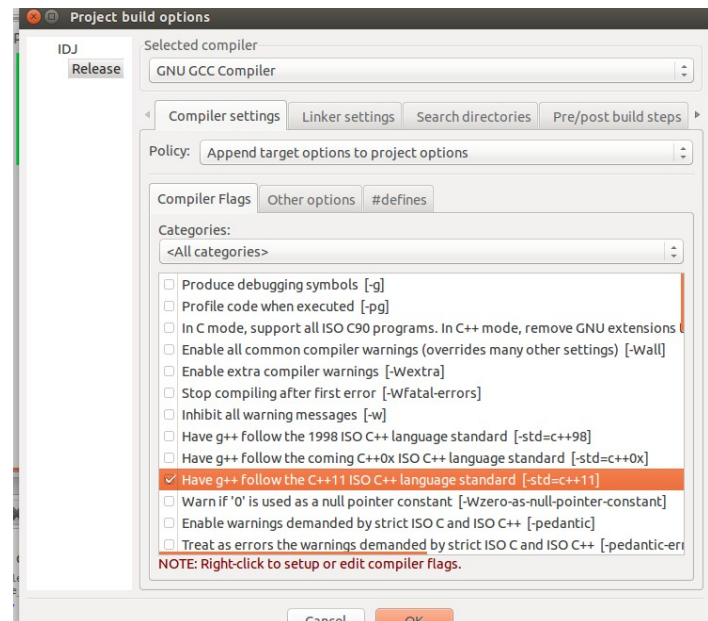
```
#include <iostream>
#include <SDL2/SDL.h>
int main(int argc, char *argv[]){
    SDL_Init(SDL_INIT_EVERYTHING);
    std::cout << "Ola\n";
    return 0;
}
```

Em seguida, no menu aa arvore de arquivos do projeto:



Clique com o botão direito no icone escrito IDJ e selecione Build Options. Na primeira janela que

aparecer, procure a opção que estiver escrito c++11 e marque-a.



Nessa mesma janela, selecione a aba Linker Settings.

No campo “Linker options”, coloque as flags para linkar os arquivos do projeto. Nesse pequeno exemplo, as flags necessárias são:

```
-lSDL2 -lSDL2main
```

As flags da SDL e seus plug-ins são:

```
-lSDL2main -lSDL2 -lSDL2_image  
-lSDL2_ttf -lSDL2_mixer
```

**Atenção:** No windows pode ser preciso colocar uma flag a mais, isto é

```
-lMinGW32
```

Depois basta apertar F9 para compilar e rodar.

O próprio Code::Blocks vai exibir uma janela de console, mesmo que o projeto não tenha uma, então fica mais fácil de ver o output.

Caso tenha problemas para abrir arquivos, por padrão, o codeblocks salva os executáveis na pasta “IDJ/bin/release/” ou “IDJ/bin/debug”, mas na hora de executar, ele seta o diretório para “IDJ/”  
Para alterar essas opções, basta ir nas propriedades do projeto e alterar.