

CodeBlocks

Roberto Rocha

Linguagem C

Instalando o CodeBlocks

Acesse o site:

<http://codeblocks.org/>



Code :: Blocks 20.03 está aqui!

Escrito por MortenMacFly
Quinta-feira, 19 de março de 2020 05:09

Finalmente, mude para wxWidgets 3.x, muitas melhorias, novos recursos, mais estável, aprimorado para HiDPI, o novo Code :: Blocks versão 20.03 chegou. Obtenha na seção de [downloads](#) ! Um [changelog](#) resume novos recursos e correções.

Nós fornecemos binários para as principais plataformas suportadas por Code :: Blocks, com mais por vir na próxima vez.

Atualizado pela última vez na quinta-feira, 19 de março de 2020 05:10

Instalando o CodeBlocks

Transferências


Existem diferentes maneiras de baixar e instalar Code :: Blocks no seu computador:

- **Baixe a versão binária** 

Esta é a maneira fácil de instalar o Code :: Blocks. Baixe o arquivo de instalação, execute-o no seu computador e Code :: Blocks será instalado, pronto para você trabalhar com ele. Não pode ser mais fácil do que isso!

- **Baixar um nightly build** : Há também mais recente chamado *nightly builds* disponível nos **fóruns** . Observe que geralmente consideramos as compilações noturnas *estáveis* , a menos que indicado de outra forma.
- Outras distribuições geralmente seguem fornecidas pela **comunidade** (grande "Obrigado!" Por isso!). Se você quiser fornecer algum, certifique-se de anunciá-lo nos fóruns para que possamos colocá-lo na página inicial oficial do C :: B.

Selecione um pacote de instalação dependendo da sua plataforma:


- Windows XP / Vista / 7 / 8.x / 10 
- Linux 32 e 64 bits
- Mac OS X

NOTA: Para sistemas operacionais mais antigos, use versões mais antigas. Existem lançamentos para muitas versões e plataformas do sistema operacional na página Sourceforge.net .

NOTA: Existem também *compilações noturnas* mais recentes disponíveis nos **fóruns** ou (para usuários do Ubuntu) no **repositório Ubuntu PPA** . Observe que geralmente consideramos as compilações noturnas *estáveis* .

NOTA: Temos um **Changelog para 20.03** , que oferece uma visão geral das melhorias e correções que colocamos na nova versão.

NOTA: As compilações padrão são de 64 bits (começando com a versão 20.03). Também fornecemos compilações de 32 bits para sua conveniência. Os pacotes de 32 bits têm o mesmo compilador de 64 bits que a versão de 64 bits incluída ou (o que é recomendado) uma versão de 32 bits do compilador.

[codeblocks-20.03mingw-setup.exe](#) 

29 de março de 2020

O arquivo codeblocks-20.03mingw-setup.exe inclui *adicionalmente* o compilador GCC / G ++ / GFortran e o depurador GDB do [projeto MinGW-W64](#) (versão 8.1.0, 32/64 bits, SEH).

Instale o codeBlocks

Escolha a origem do download



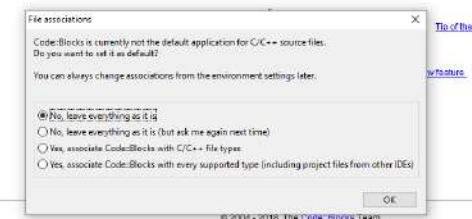
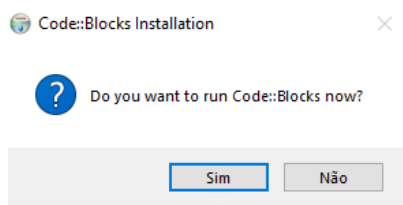
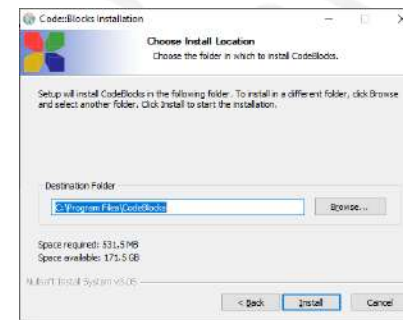
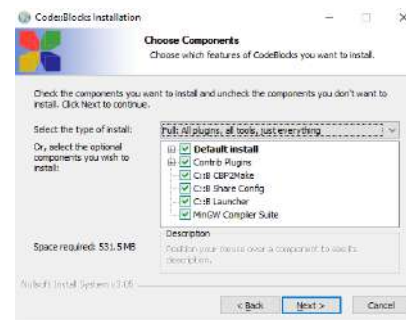
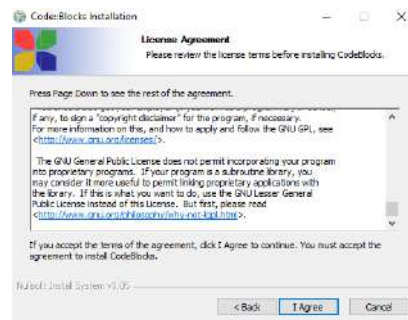
FossHUB ou Sourceforge.net

<https://www.codeblocks.org/downloads/binaries/>

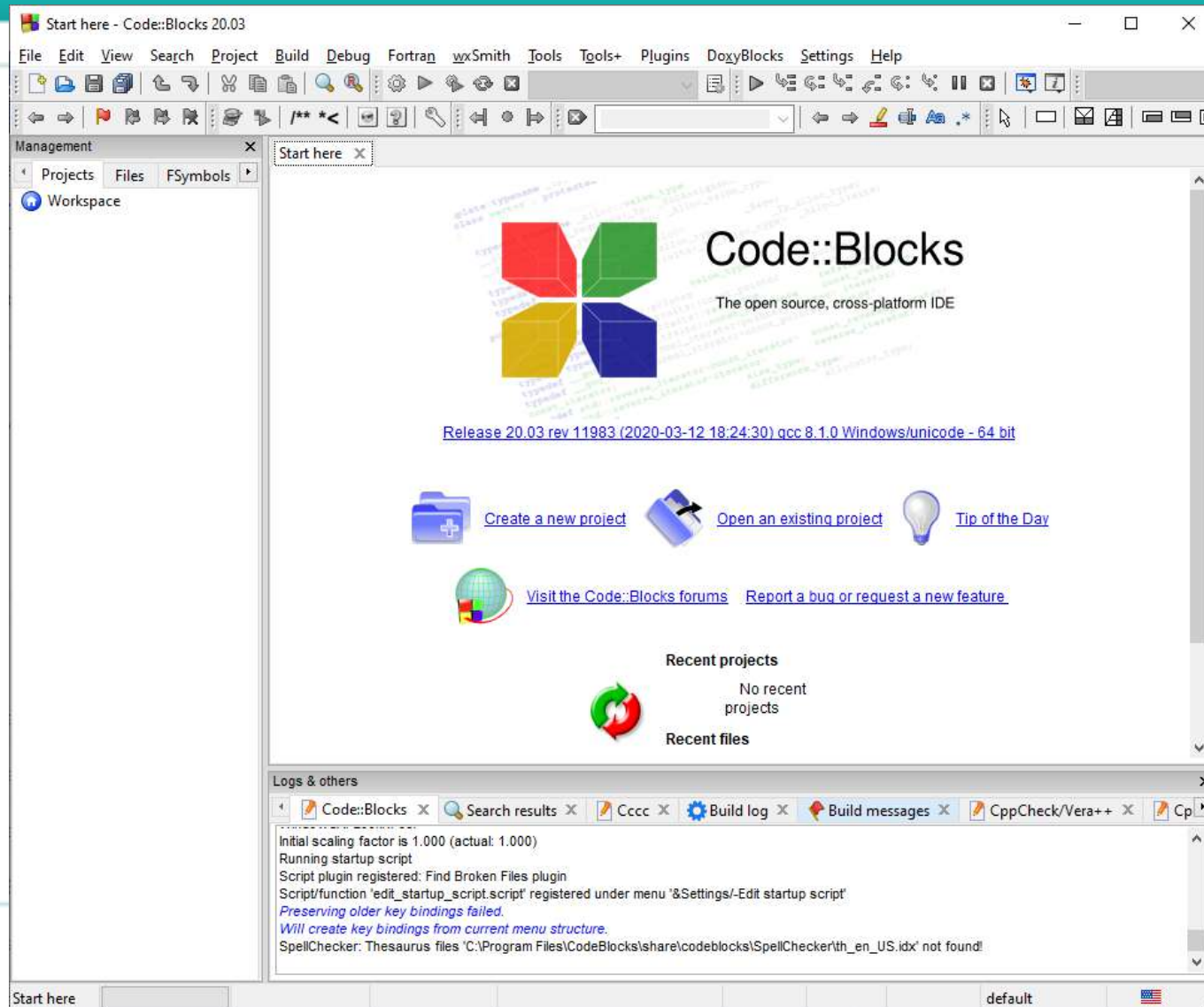
PUC Minas Virtual

Instalando o CodeBlocks

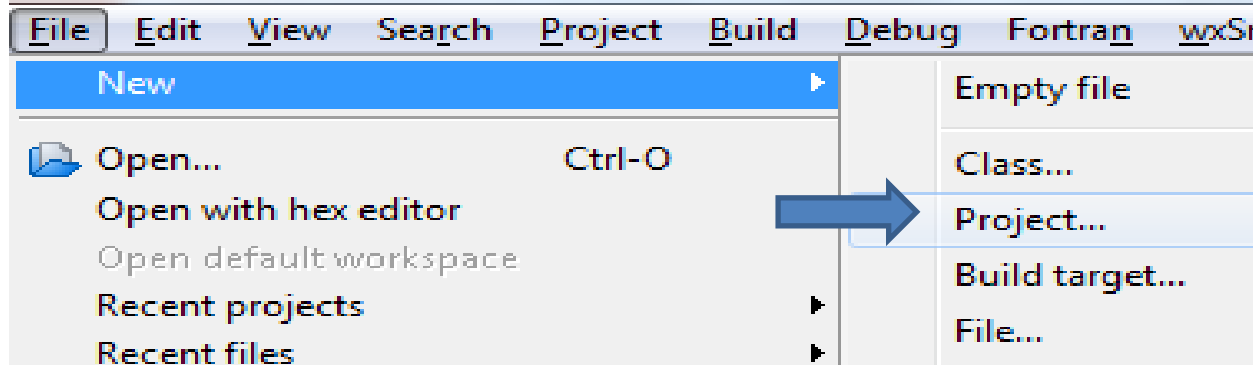
Instale o codeBlocks



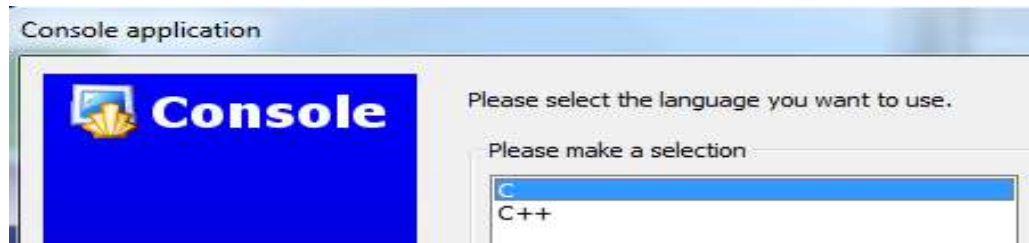
Abra o CodeBlocks



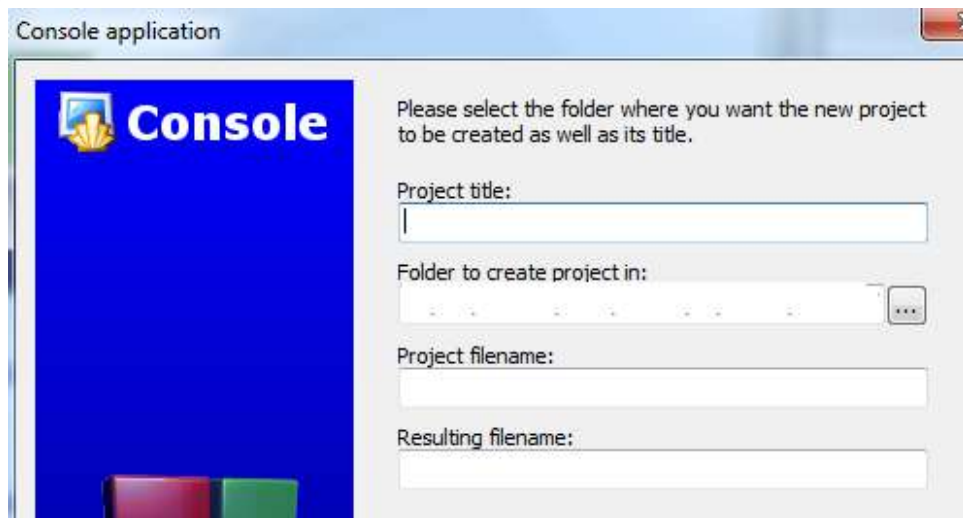
Criando um projeto



em seguida escolha o tipo de projeto
e a linguagem : C



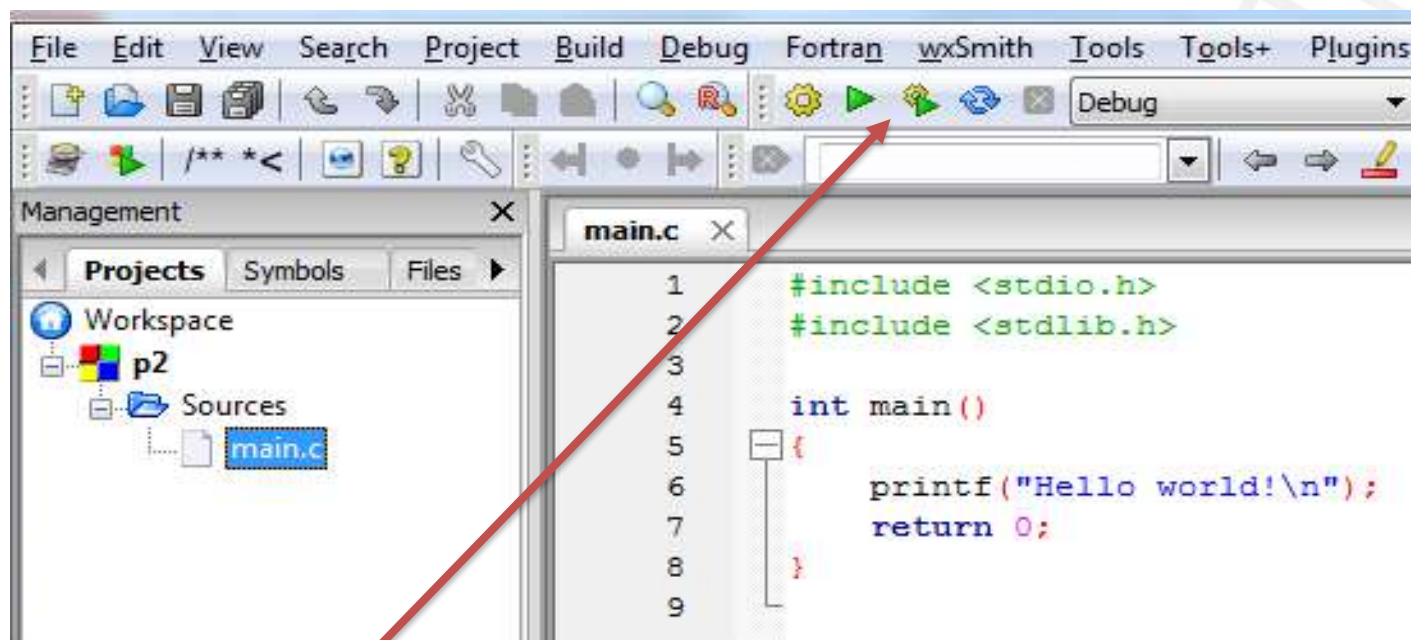
Informe a localização e o nome do projeto:



Informe o compilador a ser utilizado e finalize.



Na aba Projets aparecerá seu projeto e o programa main.c



Clique em  para compilar, linkar o programa e executá-lo.

Se tudo sair ok, deverá aparecer a seguinte tela:

```
Hello world!  
Process returned 0 (0x0)   execution time : 7.221 s  
Press any key to continue.
```

Altere o programa para

```
/* Um Primeiro Programa */  
int main ()  
{  
    printf ("Primeiro programa em C!\n");  
    return(0);  
}
```

O programa

```
int main ()  
{  
    printf ("Primeiro programa em C!\n");  
    return(0);  
}
```

int main()

O nome das Funções – A função main

Este programa compõe-se de uma única função chamada **main**.

O nome de uma função pode ser qualquer um, com exceção de **main**, reservado para a função que inicia a execução do programa.

Em todo programa C / C++ deve existir uma e uma única função chamada de **main**.

A função **main** marca o ponto de partida do programa.

O programa termina quando for encerrada a execução da função **main**.

int main()

Os parênteses após o nome da função

Os parênteses após o nome **main** indica que se trata de uma função.

Sem os parênteses, o compilador poderia pensar que esse nome se refere a uma variável.

Nosso programa eles estão vazios, mas nem sempre isso ocorre.

O tipo int

`int main()`

A função **main** de nosso programa é do tipo **int**. Isso significa que ela deverá retornar um **número inteiro**. A instrução de retorno é a seguinte.

```
return 0;
```

Chaves

Toda função em C/ C++ deve começar com uma chave de abertura de bloco `{` e terminar uma fechamento de bloco `}`. As chaves delimitam o corpo da função.

Espaços em branco

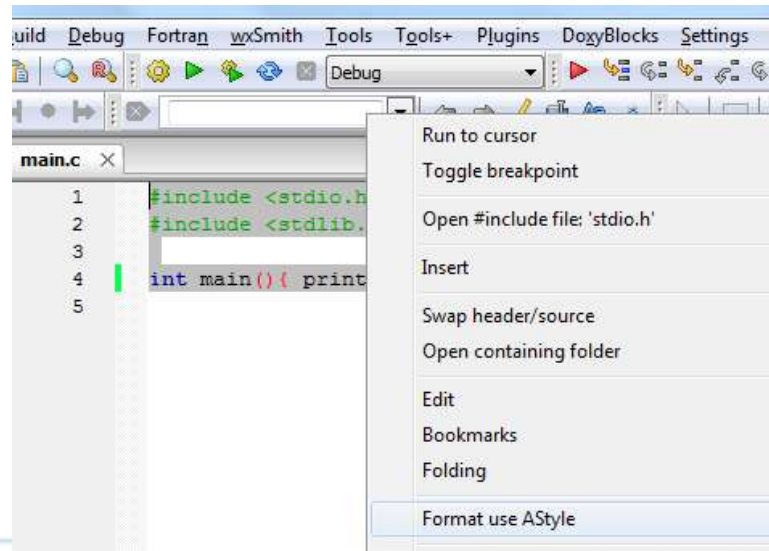
- Pode-se inserir espaços em branco, tabulações e pular linhas à vontade em seus programas. O compilador ignora esses caracteres.
- É possível escrever várias instruções em uma única linha, separadas por qualquer número de espaços ou tabulações, ou escrever uma instrução em várias linhas.
- Não há um estilo obrigatório para a escrita de programas C/C++.
- Porém não se pode inserir espaços em branco no meio de uma palavra ou de um símbolo da linguagem.

O programa anterior poderia ficar assim escrito:

```
/* Um Primeiro Programa */
```

```
int main () { printf ("Primeiro programa em C!\n"); return(0); }
```

Para que a formatação seja automática tecle <alt><a> para selecionar todo o texto e com o botão direito do mouse escolha “Format use Astyle”



Diretivas do pré-processador

A primeira linha do programa inicia-se com # **não** é uma instrução da linguagem C/C++.

Observe que não há ponto-e-vírgula ao seu final.

É uma diretiva do pré-processador.

Toda diretiva é iniciada pelo símbolo # e o seu texto deve ser escrito em uma única linha.

Se o texto for muito grande, pode-se terminar a linha com a barra invertida (\) e continuar em outra linha, as diretivas servem para auxiliar no desenvolvimento do programa-fonte.

Bibliotecas

A linha `#include <stdio.h>` diz ao compilador que ele deve incluir o arquivo-cabeçalho `stdio.h`.

Neste arquivo existem declarações de funções úteis para entrada e saída de dados (`std` = standard, padrão em inglês; `io` = Input/Output, entrada e saída ==> `stdio` = Entrada e saída padronizadas).

Toda vez que você quiser usar uma destas funções deve-se incluir este comando.

O C/C++ possui diversos arquivos-cabeçalho.

Instruções de Programa

O programa mostrado possui duas instruções.

```
printf ("Primeiro programa em C!\n");
```

imprime a frase entre aspas duplas na tela do computador.

```
return 0;
```

solicita à função **main** que retorne o valor **0** a quem a chamou.

No caso, o sistema operacional ou o compilador

- Toda instrução em C++ deve terminar com um **ponto-e-vírgula**.
Obs: toda instrução não quer dizer toda linha!!!
- O **ponto-e-vírgula** é crucial da sintaxe da linguagem e facilmente esquecido o que acarreta a apresentação de um erro de compilação.
- Uma **função** pode ter qualquer número de instruções.
- As instruções devem ser escritas entre as **chaves** que delimitam o corpo da função e executadas na ordem em que as escrevemos.

Letras Minúsculas

Em C++ as letras maiúsculas e minúsculas são consideradas distintas.

Assim, a função principal que inicia o programa deve ser grafada em letras minúsculas.

Os nomes `main`, `Main`, `mAIN`, `MAIN`, `MAin` são diferentes.

Exercício

Crie um novo projeto que escreva a mensagem “segundo programa em C/C++”.

Exercícios de fixação:

1. Escreva um programa que solicite ao usuário a altura e o raio de um cilindro circular e imprima o volume do cilindro.

O volume de um cilindro circular é calculado por meio da seguinte fórmula:

$$\text{Vol} = \text{PI} * \text{raio}^2 * \text{altura}$$

Obs: em algoritmo o operador de potência é o ^ em C utiliza-se a função `pow(a,b)`. exemplo a^2 `a^2` `pow(a,2)`

2. Uma empresa contrata um encanador a R\$ 20.00 por dia. Crie um programa que solicite o número de dias trabalhados pelo encanador e imprima a quantia líquida que deverá ser paga, sabendo-se que são descontados 8% de impostos.
3. O cardápio de uma lanchonete é dado abaixo. Prepare um programa que leia a quantidade de cada item que você consumiu e calcule a conta final.

Hambúrguer..... R\$ 30,00

Cheeseburger..... R\$ 35,50

Fritas..... R\$ 20,50

Refrigerante..... R\$ 10,00

Milkshake..... R\$ 30,00



PUC Minas
Virtual