

Flávio Augusto de Freitas
Introdução à Programação em Linguagem C/C++

<http://flavioaf.blogspot.com>

C/C++

Tutorial 1 (usando Dev-C++ versão 4.9.9.2)



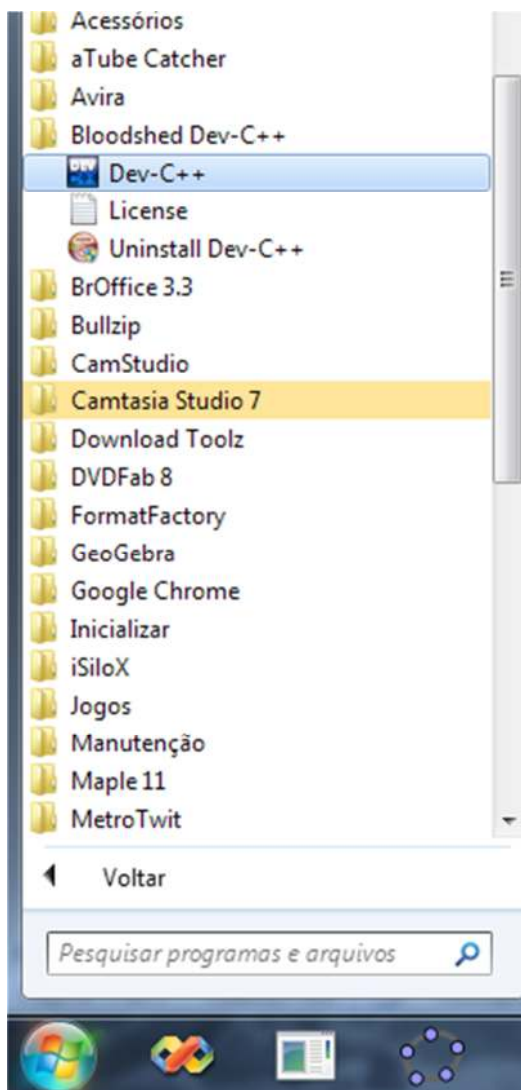
2011

1 INTRODUÇÃO

Esta série de tutoriais foi escrita usando o **Microsoft Windows 7 Ultimate** e o **Bloodshed Dev-C++** versão 4.9.9.2, que pode ser baixada em <http://www.bloodshed.net>.

2 EXECUTANDO O DEV-C++

Clique no botão Iniciar, depois clique em Todos os programas; em seguida clique em Bloodshed Dev-C++; finalmente clique em Dev-C++.




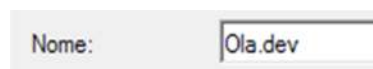
3 O AMBIENTE DO DEV-C++

Depois de executar o programa, clique no menu Arquivo, depois aponte o mouse para a opção Novo e clique em Projeto... . Na janela que se abre, clique em

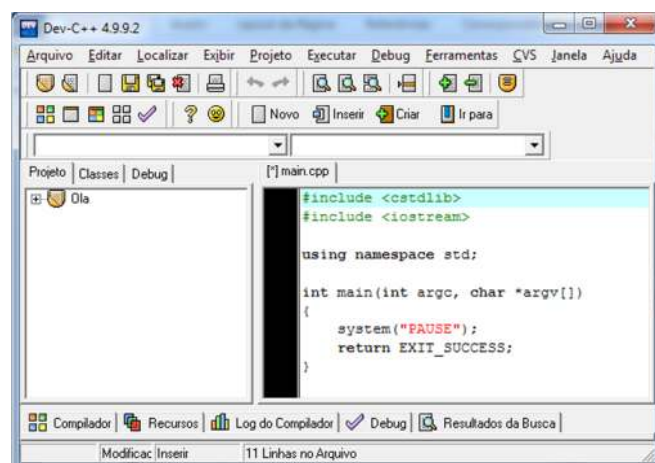


Ainda na mesma janela, onde está escrito Projeto1, mude para **Ola**. Clique no botão Ok. Agora, vamos salvar previamente nosso projeto. Na janela que se abriu, no lado esquerdo, clique no botão Área de Trabalho (ou Desktop, se estiver escrito).

Clique no botão  para criar uma nova pasta na Área de Trabalho. Digite o nome ProjetoOla e pressione a tecla Enter. Dê dois cliques na pasta ProjetoOla, recém criada. Em seguida, na mesma janela, observe que o projeto já tem nome (Ola.dev).



Clique no botão Salvar. A seguinte tela é mostrada:

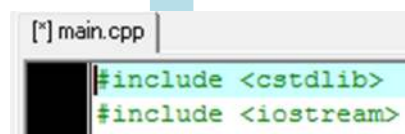


3.1 ENTENDENDO O AMBIENTE DO DEV-C++

Observe na tela que há um menu principal e algumas barras de ferramentas. Logo abaixo a janela é dividida em duas partes.

A parte da esquerda é o navegador do projeto. Clique no sinal de + ao lado de Ola. Veja que o Dev-C++ mostra os arquivos que compõem o projeto. Em nosso caso, há apenas o main.cpp.

A parte da direita mostra o arquivo selecionado. Observe que a aba mostra o nome do arquivo e um sinal de (*).



Isto significa que ainda não salvamos o arquivo corrente. Vamos salvá-lo agora. Clique no menu Arquivo, depois clique em Salvar. Na caixa que

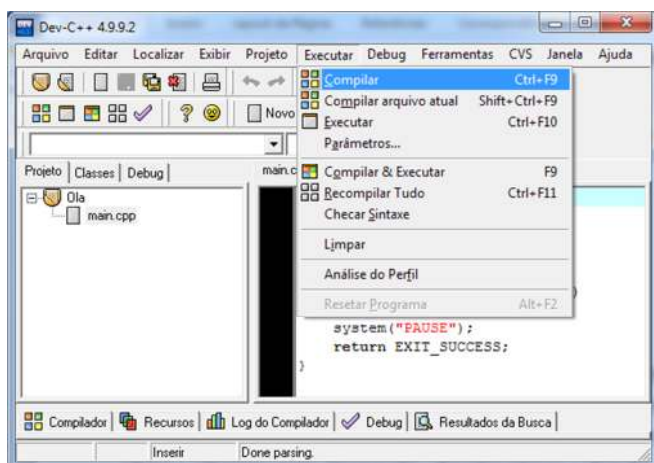
aparece, não modifique nada, apenas clique no botão Salvar. Observe que o (*) sumiu. Sempre que alterarmos o código o Dev-C++ mostra o (*), indicando que precisamos salvar o arquivo.

3.1 EDITANDO O TEXTO

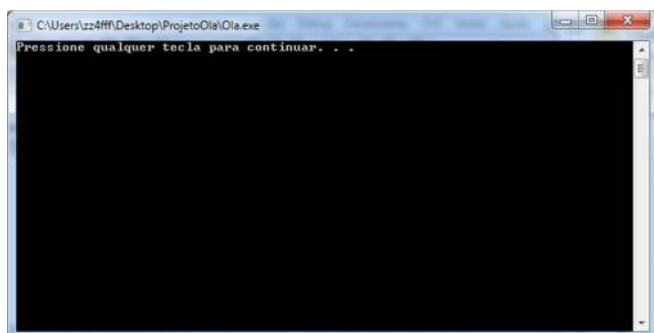
Notou a linha destacada (com fundo azulado). Esta é a linha corrente de edição do texto. Use as teclas de setas do teclado para mover-se pelo texto. Experimente agora. Para inserir texto basta deslocar o cursor usando as teclas de setas e digitar. Mas não faça nada ainda. Veja também que várias palavras aparecem ou coloridas, ou em **negrito**. O Dev-C++ destaca as palavras reservadas da linguagem com diversas cores e negritos ou *itálicos*. Assim você não se perde com muita frequência. Siga lendo.

3.3 EXECUTANDO O PROGRAMA

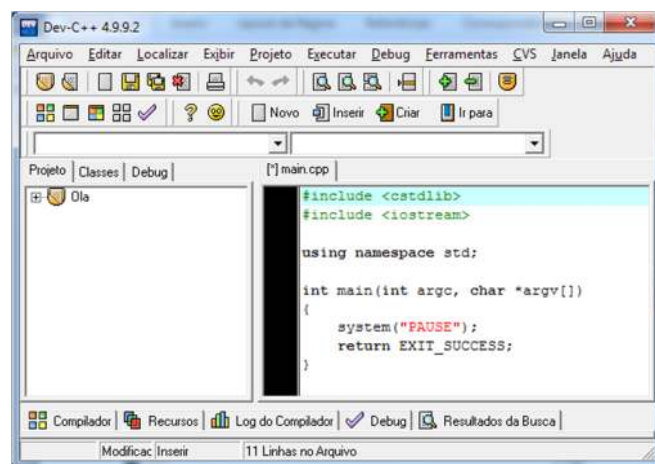
Vamos executar o programa e ver o que acontece. Clique no menu Executar e depois em Compilar & Executar, como abaixo:



Ou pressione F9, que é a tecla de atalho. Aliás, vários menus têm alguma tecla de atalho. Aprenda as principais: Ctrl-S salva o arquivo selecionado; Ctrl-F localiza texto; Para compilar e executar o programa, basta pressionar F9, sem ter de ir até o menu Executar. Após a execução, a seguinte tela se apresenta:



Pressione qualquer tecla e o programa termina, voltando ao Dev-C++.



Mas o que está escrito? O que são estes códigos?

```
#include <cstdlib>
```

```
#include <iostream>
```

Estas duas linhas incluem as bibliotecas de códigos necessárias para que o programa execute. Cada programa terá tantas bibliotecas quantas forem preciso. Há centenas delas.

```
using namespace std;
```

Isto é padrão no Dev-C++. Todo programa deste tutorial terá esta linha de código.

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

Esta é a declaração da função main. Todo programa em C/C++ tem pelo menos uma função com o nome main. Isto é obrigatório.

Vamos analisar esta linha:

O **int**, diz que a função main deve retornar um valor inteiro ao final da execução. Isto é feito pela linha de código **return EXIT_SUCCESS;**. Entre parênteses existem dois argumentos, que são valores que podem ser passados para o programa, quando ele é executado. Aqui **int argc** indica quantos argumentos foram passados e **char *argv[]** é a lista de argumentos que foram passados.

Finalmente **system(\"PAUSE\");** indica ao compilador para mostrar uma mensagem padronizada do sistema operacional. É aquela mensagem **“Pressione qualquer tecla para continuar. . .”** que apareceu quando você executou o programa.

4 UM EXEMPLO COMPLETO

Modifique o código do programa para que fique como na imagem abaixo:

```
#include <cstdlib>
#include <iostream>

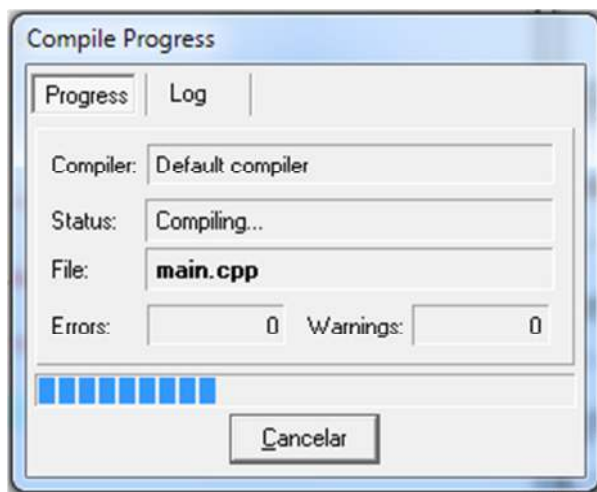
using namespace std;

int main(int argc, char *argv[])
{
    char nome[10];
    int idade;

    printf("Insira o seu nome: ");
    scanf("%s", &nome);
    printf("\n\nInsira a sua idade: ");
    scanf("%d", &idade);
    printf("O seu nome eh %s e tem %d anos.\n", nome, idade);

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Após a digitação, pressione Ctrl-S para salvar o arquivo. Pressione F9 para executar o programa. A seguinte janela aparece:



Ela mostra o progresso da compilação. É importante você observar as partes

Errors: 0 e Warnings: 0

que mostram se houve erros (*Errors*) ou avisos (*Warnings*), respectivamente. Avisos não precisam ser levados tão a sério, ao menos por enquanto para você, mas erros sim, pois impedem o programa de executar. Caso haja um ou mais erros, após a compilação, o compilador se mostrará assim:



Observe que na parte em que editamos o código, uma linha ficou marcada com fundo vermelho escuro. É a linha onde o Dev-C++ encontrou um erro. Mas que erro? Abaixo da janela de código, observe a linha destacada com fundo cinza. Na coluna Linha, vemos qual linha está o erro encontrado (linha 11). Na coluna Mensagem, podemos ler a mensagem

`'printf' undeclared (first use this function)`

que indica que o Dev-C++ não encontrou a função 'printf'. Voltando ao código, descobrimos o erro: digitamos incorretamente o nome da função printf (faltando a letra t) e o Dev-C++ nos avisou. Basta corrigir e tentar compilar novamente, pressionando F9. Para ver um erro ocorrendo, experimente trocar a palavra **char** por **chr** e tente executar novamente o programa.

Se o programa compilar e executar corretamente, você deverá ver uma tela preta, com uma mensagem solicitando o seu nome:

Insira o seu nome:

Digite seu nome (Flavio, por exemplo) e pressione Enter.

Em seguida, o programa solicitará sua idade:

Insira a sua idade:

Digite sua idade (42, por exemplo) e pressione Enter.

Em seguida o programa mostra uma mensagem, parecida com o exemplo abaixo.

O seu nome eh Flavio e tem 42 anos.

Pressione qualquer tecla para continuar. . .

Pressione qualquer tecla, Enter, por exemplo, e o programa termina e retorna ao compilador.

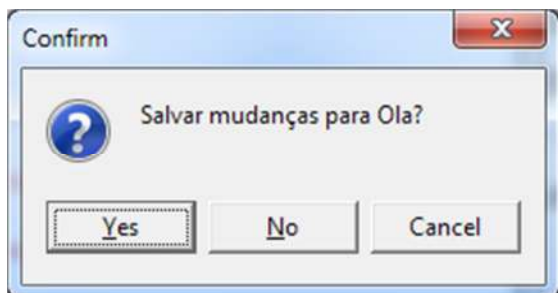
5 ENDENTAÇÃO

É o alinhamento do texto. Observe que o código entre { } está deslocado para a direita. Isto é importante para a visualização e entendimento do código escrito. Quanto maior for seu código, ou seja, quanto mais linhas de código tiver seu programa, mais importante será a endentação.

6 SAINDO DO DEV-C++

Agora que terminamos este programa, podemos sair do ambiente do Dev-C++. Para isto, clique no

menu Arquivo e depois clique na opção Sair. Caso alguma coisa não esteja salva, o Dev-C++ mostrará a seguinte tela:



Basta clicar no botão Yes para o Dev-C++ salvar o arquivo e terminar. Se desistir de sair do Dev-C++, basta clicar no botão Cancel e você continuará com o Dev-C++ aberto.

7 REABRINDO O ÚLTIMO PROJETO

Entre novamente no Dev-C++ (se tiver esquecido, volte ao tópico número 2). Com o Dev-C++ aberto, clique no menu Arquivo e depois aponte o mouse para a opção Reabrir. Clique na opção com o número 0 (zero). Pronto, reabrimos o projeto Ola.

Se não estiver vendo o código do programa, clique no sinal de + ao lado de Ola (lado esquerdo da tela, no navegador de projeto), depois clique em `main.cpp`, que aparece embaixo de Ola. O código deve aparecer no lado direito.

7 TERMINAMOS

Terminamos por aqui. Clique no menu Arquivo, depois clique na opção Sair.

Corra para o próximo tutorial.