

CCF 110 – Programação

Aula 02 – Linguagem C

Prof. José Augusto Nacif – jnacif@ufv.br



Tradução



- ▶ **MONTADOR (assembler)**
 - ▶ Tradutor para linguagens de 2a geração.
- ▶ **COMPILADOR:**
 - ▶ Traduz todo o programa de uma vez.
- ▶ **INTERPRETADOR:**
 - ▶ Traduz o programa instrução por instrução.





Histórico da Linguagem C

- ▶ Criada por Denis Ritchie, na década de 1970, para uso em um computador DEC PDP-11 em Unix
- ▶ C++ é uma extensão da linguagem C
- ▶ O sistema Unix é escrito em C e C++





Estrutura básica de um programa C

diretivas para o pré-processador

declaração de variáveis globais

main ()

{

 declaração de variáveis locais da função main

 comandos da função main

}





Diretivas para o processador - Bibliotecas

- ▶ Diretiva `#include` permite incluir uma biblioteca
- ▶ Bibliotecas contêm funções pré-definidas, utilizadas nos programas
- ▶ Exemplos

<code>#include <stdio.h></code>	Funções de entrada e saída
<code>#include <stdlib.h></code>	Funções padrão
<code>#include <math.h></code>	Funções matemáticas
<code>#include <string.h></code>	Funções de texto





Code::Blocks

- ▶ O uso de uma IDE visa aumentar a produtividade e ter à disposição uma série de ferramentas de auxílio num único ambiente.
- ▶ Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado.
- ▶ Minimalistic GNU for Windows, um port do GNU Compiler Collection para Windows.





Code::Blocks

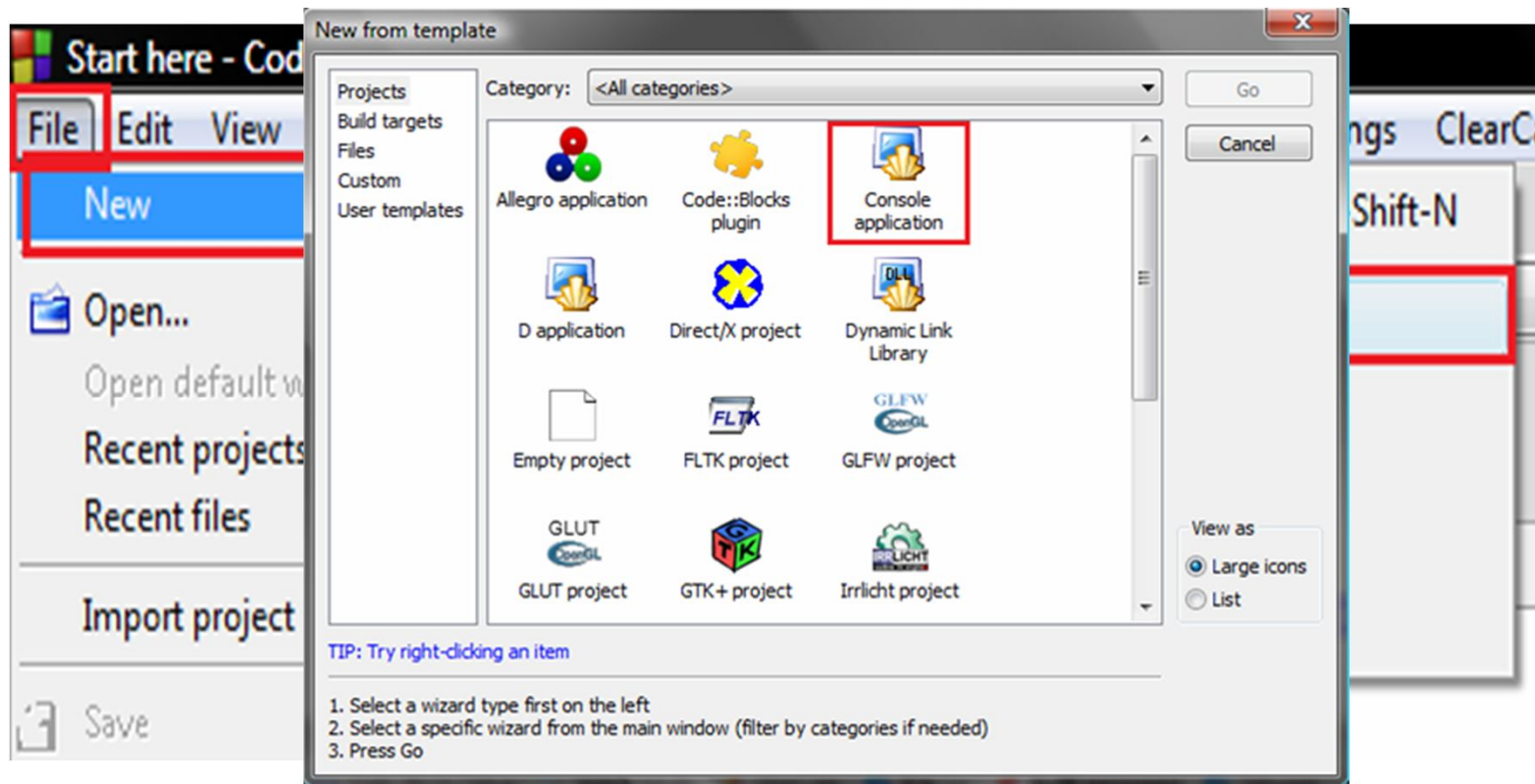
- ▶ Pode ser usada a versão mais recente do Code::Blocks, disponível no site oficial:
- ▶ <http://www.codeblocks.org/downloads>
 - ▶ No site oficial há versões para Windows, Linux e MacOS.





Usando o Code::Blocks

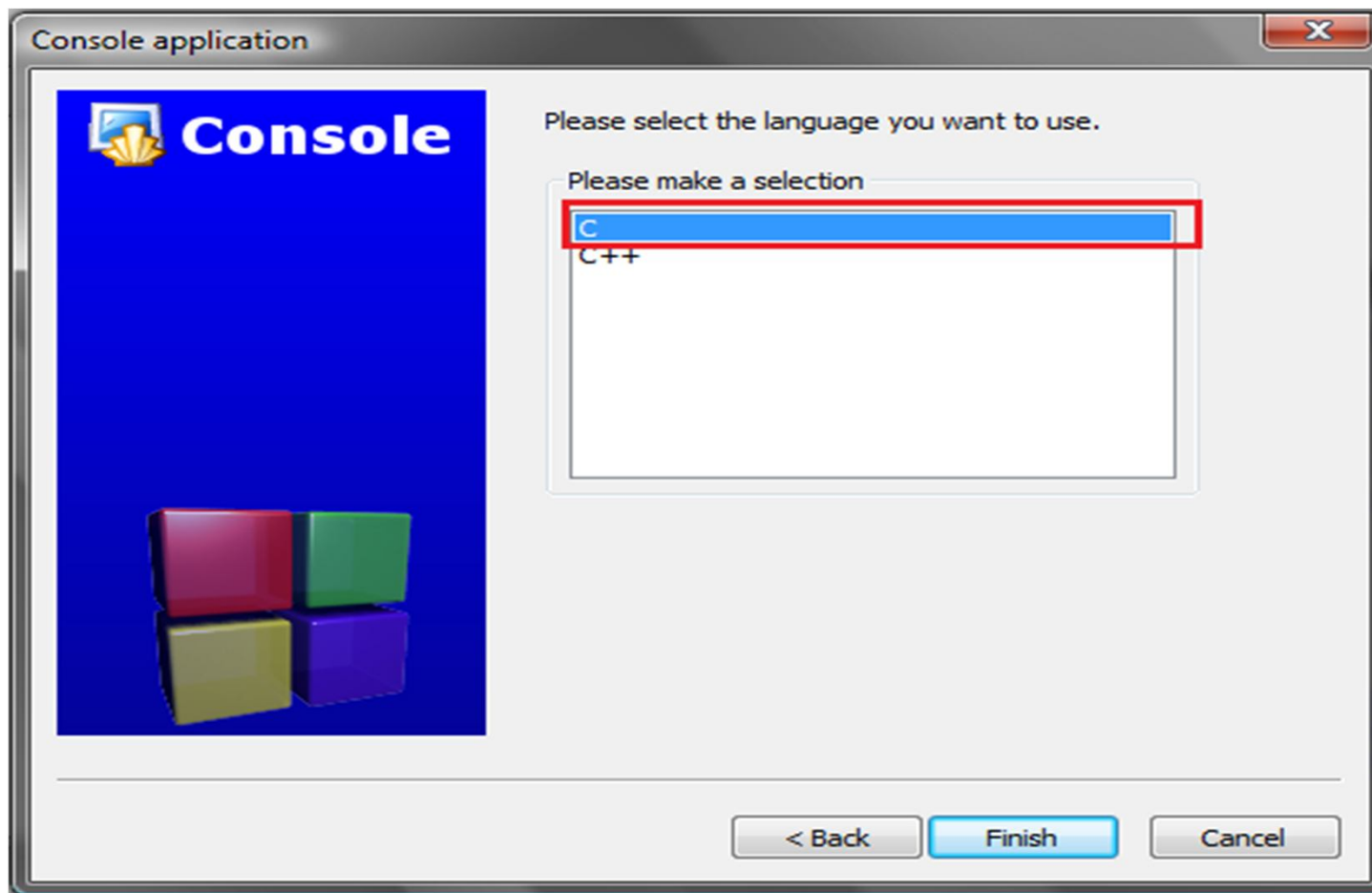
- ▶ Execute o arquivo "codeblocks.exe".
- ▶ Crie um projeto do tipo "Console Application".





Usando o Code::Blocks

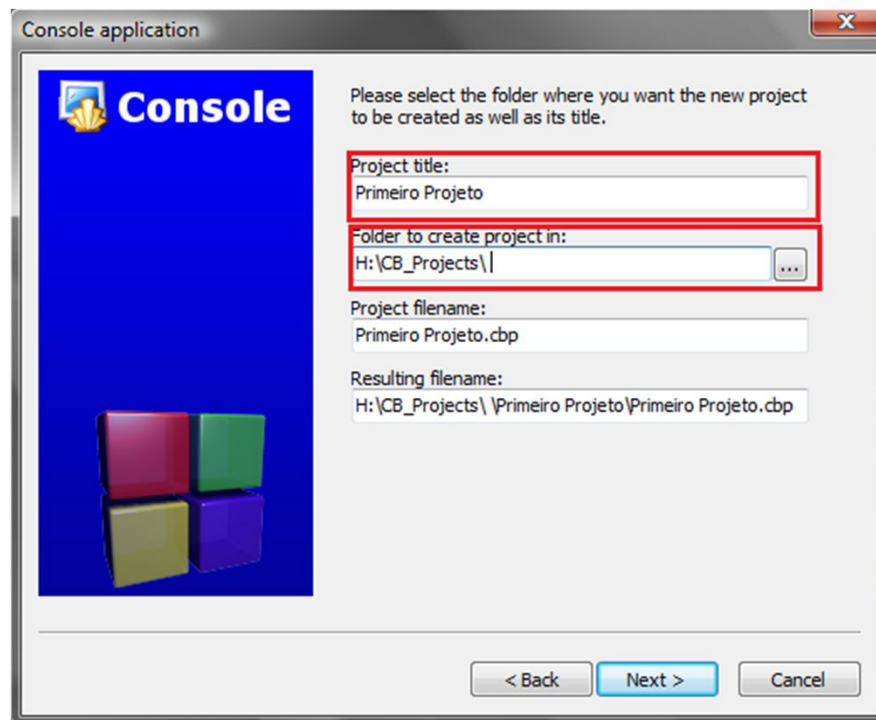
- Selecione c, em seguida clique em Next.





Usando o Code::Blocks

- ▶ Em Project Title, coloque qualquer nome.
- ▶ Em Folder... In, coloque uma pasta e aperte em Next.

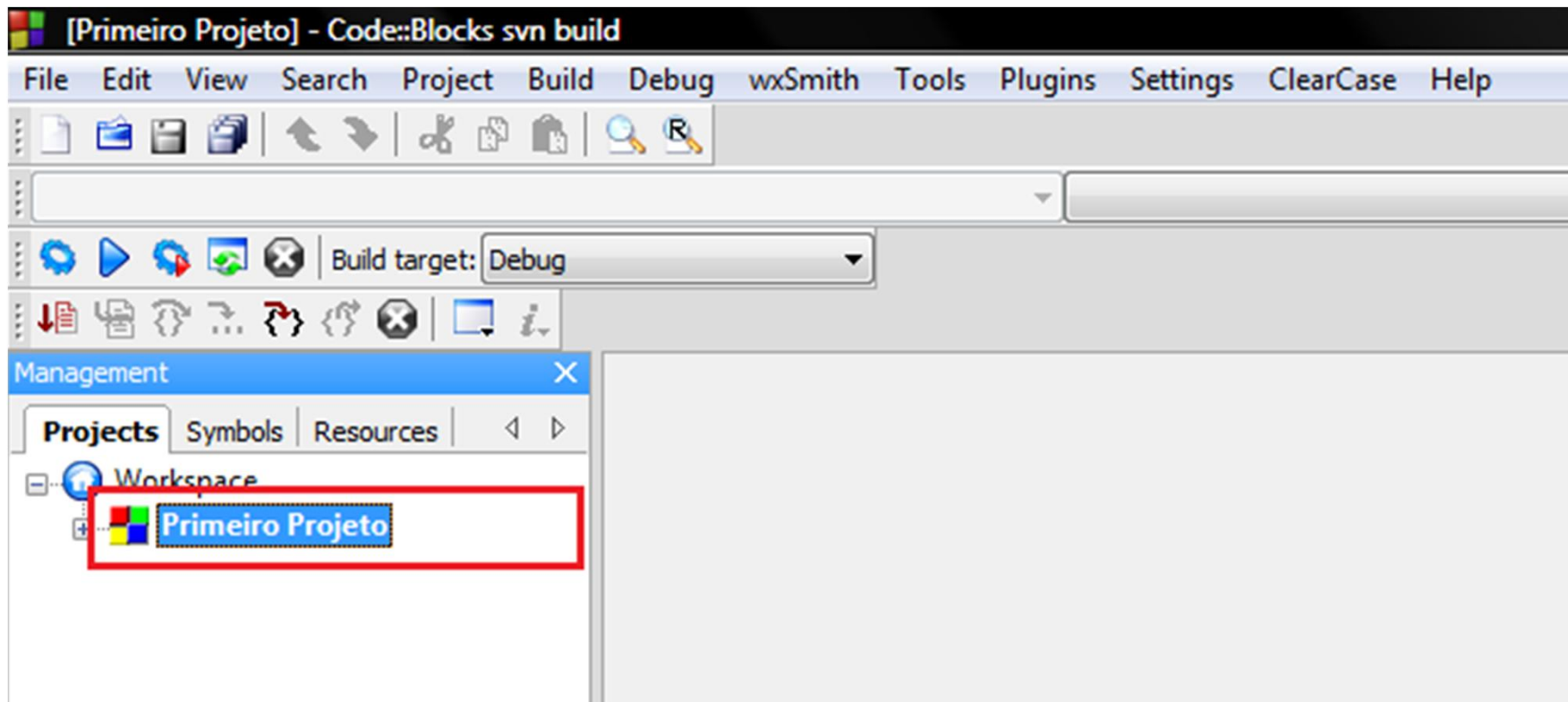


- ▶ Clique em Next e na tela seguinte, Finish.



Usando o Code::Blocks

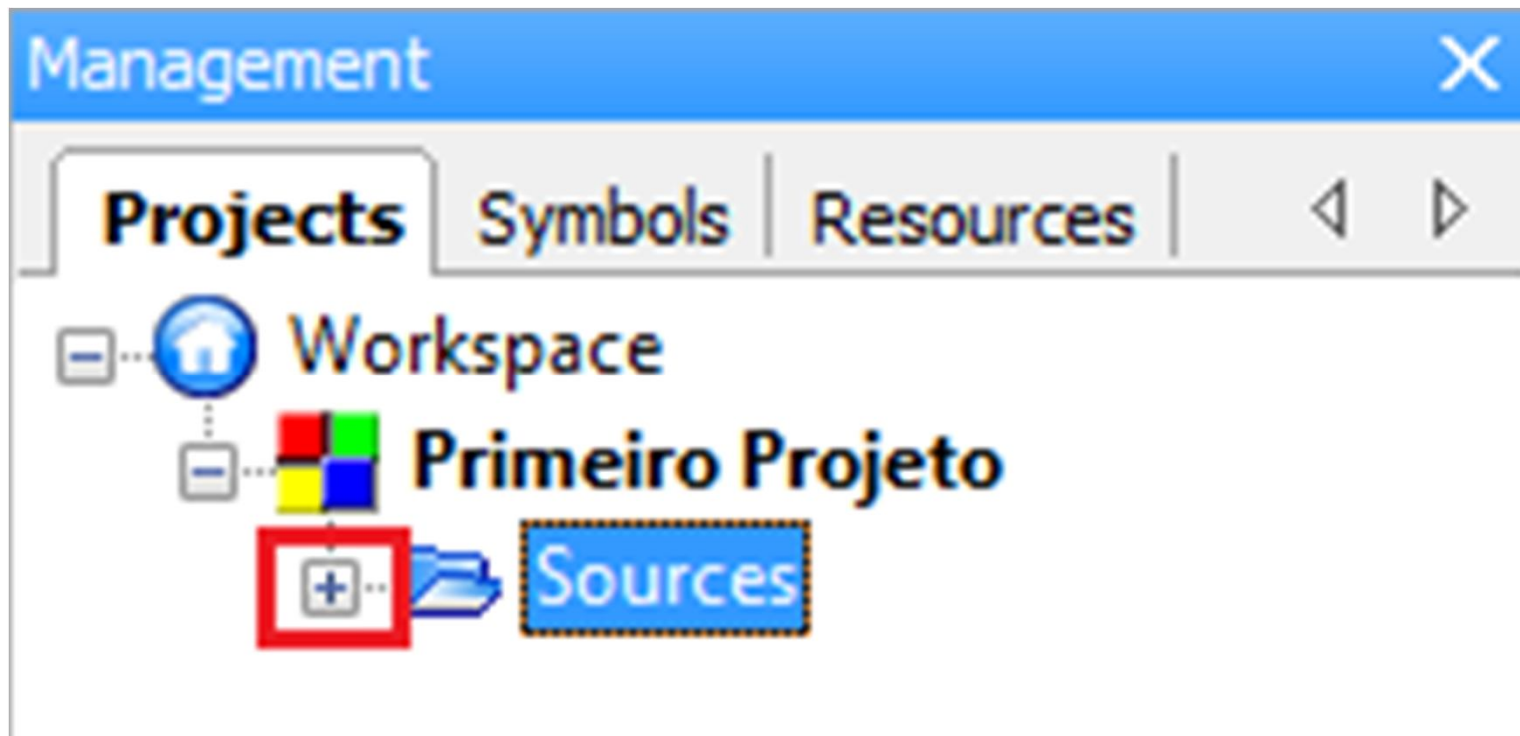
- ▶ Na lista lateral, clique no projeto.





Usando o Code::Blocks

- ▶ Para editar o arquivo contendo o código expanda a pasta “Primeiro projeto”.
- ▶ Expanda a pasta “Sources”.





Usando o Code::Blocks

- ▶ Aperte F9 para executar o projeto atual.

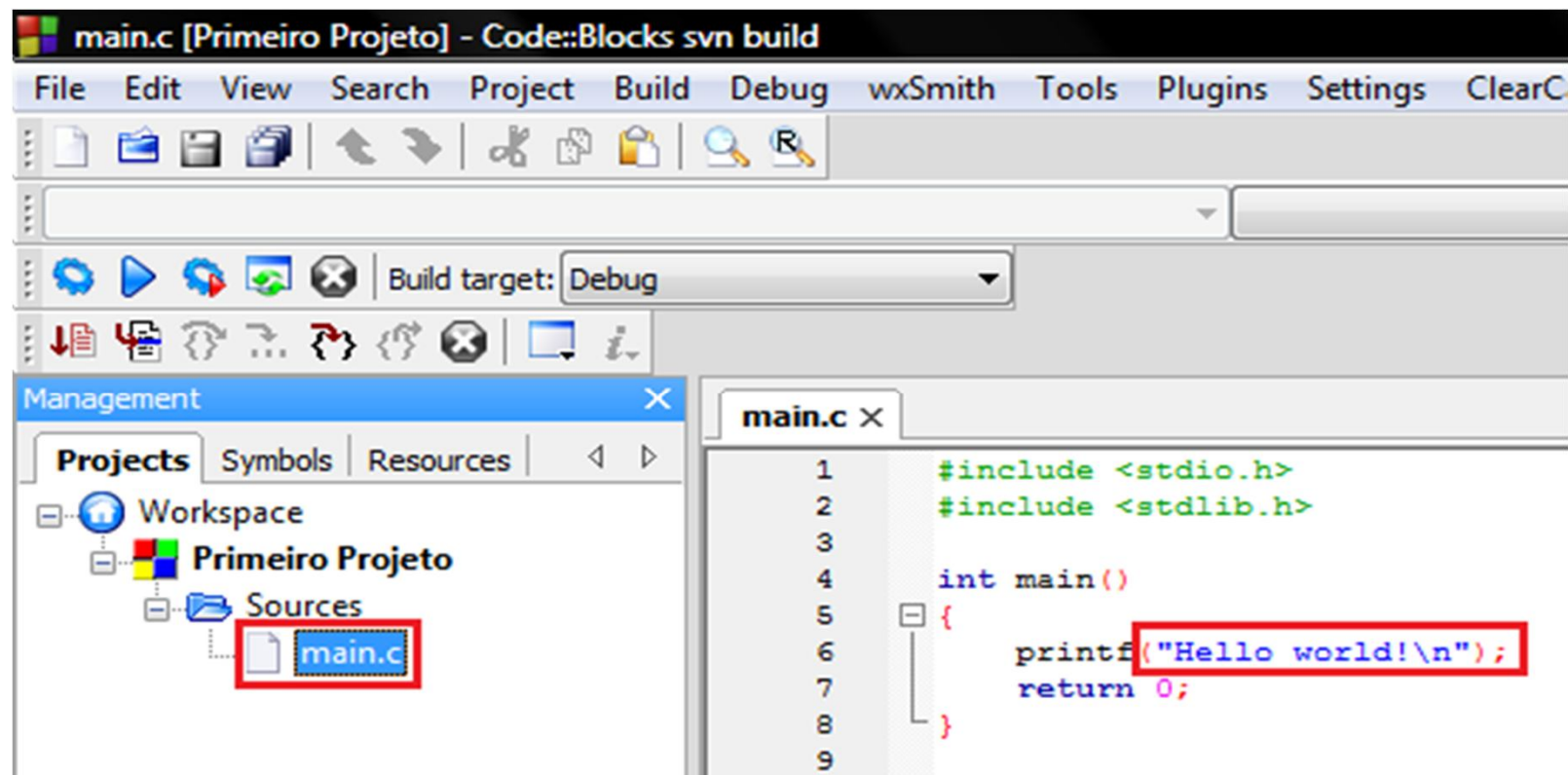
A screenshot of a Windows command prompt window. The title bar shows the file path: "f:\Desktop\Primeiro Projeto\bin\Debug\Primeiro Projeto.exe". The window has standard Windows window controls (minimize, maximize, close). The command prompt displays the following text:

```
Hello world!  
Process returned 0 (0x0)   execution time : 0.022 s  
Press any key to continue.  
-
```



Usando o Code::Blocks

- ▶ Abra o arquivo "main.c", e o editor do Code::Blocks será aberto.
- ▶ Modifique "Hello world!\n" por "Oi, mundo!\n"





Usando o Code::Blocks

- Compile e execute (Apertando F9).

A screenshot of a Code::Blocks console window. The title bar shows the file path "f:\Desktop\Primeiro Projeto\bin\Debug\Primeiro Projeto.exe". The console output displays "Oi, mundo!" followed by "Process returned 0 (0x0) execution time : 0.025 s" and "Press any key to continue.". A cursor is visible on the line "Press any key to continue.". The window has standard Windows-style minimize, maximize, and close buttons in the top right corner.

```
C:\ "f:\Desktop\Primeiro Projeto\bin\Debug\Primeiro Projeto.exe"
Oi, mundo!
Process returned 0 (0x0) execution time : 0.025 s
Press any key to continue.
-
```



Dicas

- ▶ Termine todas as linhas com ;
- ▶ Sempre salve o programa antes de compilar
- ▶ Sempre compile o programa antes de executar
- ▶ Quando ocorrer um erro de compilação, dê um duplo clique sobre a mensagem de erro para destacar o comando errado no programa
- ▶ Verifique também a linha anterior, que pode ser a responsável pelo erro, especialmente se faltar o ;
- ▶ Use comentários, iniciados por //





Template (Exemplo 1)

```
#include <stdio.h>
#include <stdlib.h>
main( )
{
    printf ( "Alo mundo!" );
    system( "PAUSE" );
}
```





Declarações

- ▶ Declaram as variáveis e seus tipos
- ▶ Os nomes das variáveis devem conter apenas letras, dígitos e o símbolo _
- ▶ Os principais tipos são: **int**, **float**, **double** e **char**
- ▶ Exemplos

```
int n;  
int quantidade_valores;  
float x, y, somaValores;  
char sexo;  
char nome[40];
```

C diferencia letras maiúsculas de minúsculas!





Algoritmo

Real: n1, n2, n3, media

Na Linguagem C...

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    double n1, n2, n3, media;

    system("PAUSE");
}
```





Comando de atribuição

- ▶ Atribui o valor da direita à variável da esquerda
- ▶ O valor pode ser:
 - ▶ uma *constante*,
 - ▶ uma variável ou
 - ▶ uma expressão
- ▶ Exemplos

```
x = 4;    --> lemos x recebe 4
```

```
y = x + 2;
```

```
y = y + 4;
```

```
valor = 2.5;
```

```
sexo = 'F'
```





Entrada de Dados

► Função **scanf**

```
scanf ("formatos", &var1, &var2,...)
```

Exemplos:

```
int i, j;  
float x;  
char c;  
scanf ("%d", &i);  
scanf ("%d %f", &j, &x);  
scanf ("%c", &c);  
scanf ("%s", nome);
```

%d	inteiro
%f	float
%lf	double
%c	char
%s	palavra





Entrada de Dados (Exemplo 2)

Algoritmo

Real: n1, n2, n3, media

leia(n1)

leia(n2)

leia(n3)

leia (n1, n2, n3)

Na Linguagem C...

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
main()
```

```
{
```

```
double n1, n2, n3, media;
```

```
scanf ("%lf",&n1);
```

```
scanf ("%lf",&n2);
```

```
scanf ("%lf",&n3);
```

```
scanf ("%lf %lf %lf",&n1, &n2, &n3);
```

```
system("PAUSE");
```

```
}
```



Operadores Matemáticos

Operador	Exemplo	Comentário
+	$x + y$	Soma x e y
-	$x - y$	Subtrai y de x
*	$x * y$	Multiplica x e y
/	x / y	Divide x por y
%	$x \% y$	Resto da divisão de x por y
++	$x++$	Incrementa em 1 o valor de x
--	$x--$	Decrementa em 1 o valor de x





Entrada de Dados (exemplo 3)

Algoritmo

Real: n1, n2, n3, media

ler n1, n2, n3

media=(n1+n2+n3)/3

Na Linguagem C...

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    double n1, n2, n3, media;
    scanf ("%lf %lf %lf",&n1, &n2, &n3);
    media=(n1+n2+n3)/3;

    system("PAUSE");
}
```





Saída de Dados

► Função **printf**

```
printf ("formatos", var1, var2,...)
```

Exemplos:

```
int i, j;  
float x;  
char c;  
printf("%d", i);  
printf("%d, %f", j, x);  
printf("%c", c);  
printf("%s", nome);
```

%d	inteiro
%f	float
%lf	double
%c	char
%s	palavra





Saída de Dados (Exemplo 4)

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int i, j;
    float x;
    char c;
    char nome [10];
    i = 1; j = 2; x = 3;
    printf("%d", i);
    printf("%d, %f", j, x);

    system("PAUSE");
}
```





Saída de Dados (Exemplo 5)

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    // definicao de variaveis
    double n1, n2, n3, media;
    // entrada de dados
    scanf ("%lf %lf %lf", &n1, &n2, &n3);
    // operacao
    media=(n1+n2+n3)/3;
    // saida de dados
    printf("%lf", n1);
    printf("%lf", n2);
    printf("%lf", n3);
    printf("%lf", media);

    system("PAUSE");

}
```



Saída de Dados (Exemplo 6)

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    // definicao de variaveis
    int i,j;
    float x;
    char c;
    char nome[30];
    //entrada de dados
    scanf("%d", &i);
    scanf("%d %f", &j, &x);
    scanf("%c", &c);
    scanf("%s", nome);
    // exibicao de dados
    printf("%d", i);
    printf("%d, %f", j, x);
    printf("%c", c);
    printf("%s", nome);

    system("PAUSE");
}
```



Exemplo 7

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    //declaracao de variaveis
    int n;
    int quantidade_valores;
    float x, y, somaValores;
    char sexo;
    char nome[40];
    // operacao
    somaValores = 1+1;
    //saida de dados
    printf("somaValores=%f", somaValores);

    system("PAUSE");
    }
▶
```



Exemplo 8

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    // definicao de variaveis
    int n;
    int quantidade_valores;
    float x, y, somaValores;
    char sexo;
    char nome[40];
    x = 1; y = 2;
    //atribuicao
    somaValores = x + y;
    // mostra mensagem de texto na tela
    printf("alo mundo");
    // mostra valor na tela
    printf("somaValores=%f", somaValores);
    system("PAUSE");
}
```



Entrada e Saída

Exemplo 9

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    double n1, n2, n3, media;
    scanf ("%lf %lf %lf",&n1, &n2,
&n3);
    media=(n1+n2+n3)/3;
    printf ("%lf",media);

    system("PAUSE");
}
```

Exemplo 10

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    double n1, n2, n3, media;
    printf("Digite 3 notas: ");
    scanf ("%lf %lf %lf",&n1, &n2, &n3);
    media=(n1+n2+n3)/3;
    printf ("Media: %.2lf",media);

    system("PAUSE");
}
```





Exemplo 11

1. Tendo como dado de entrada a altura de uma pessoa, construa um programa que calcule seu peso ideal, utilizando a seguinte fórmula: peso ideal = $(72.7 * h) - 58$



Exemplo 11

- 1) Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um programa que calcule seu peso ideal, utilizando a seguinte fórmula: $\text{peso ideal} = (72.7 * h) - 58$

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
main()
{
    float altura, pesoIdeal;
    printf ("\nDigite sua altura em metros: ");
    scanf ("%f",&altura);
    pesoIdeal=(72.7*altura)-58;
    printf ("\nO peso ideal desta pessoa eh %.2f quilos\n", sexo, pesoIdeal);
    system("PAUSE");
}
```





Operadores de Atribuição

Operador	Exemplo	Comentário
=	$x = y$	Atribui o valor de y a x
+=	$x += y$	Equivale a $x = x + y$
-=	$x -= y$	Equivale a $x = x - y$
*=	$x *= y$	Equivale a $x = x * y$
/=	$x /= y$	Equivale a $x = x / y$
%=	$x \% = y$	Equivale a $x = x \% y$





Funções Matemáticas

Função	Exemplo	Comentário
<code>ceil</code>	<code>ceil(x)</code>	Arredonda o número real para cima; <code>ceil(3.2)</code> é 4
<code>cos</code>	<code>cos(x)</code>	Cosseno de x (x em radianos)
<code>exp</code>	<code>exp(x)</code>	e elevado à potencia x
<code>fabs</code>	<code>fabs(x)</code>	Valor absoluto de x
<code>floor</code>	<code>floor(x)</code>	Arredonda o número deal para baixo; <code>floor(3.2)</code> é 3
<code>log</code>	<code>log(x)</code>	Logaritmo natural de x
<code>log10</code>	<code>log10(x)</code>	Logaritmo decimal de x
<code>pow</code>	<code>pow(x, y)</code>	Calcula x elevado à potência y
<code>sin</code>	<code>sin(x)</code>	Seno de x
<code>sqrt</code>	<code>sqrt(x)</code>	Raiz quadrada de x
<code>tan</code>	<code>tan(x)</code>	Tangente de x

► `#include <math.h>`



2) Construa um programa que tendo como entrada dois pontos quaisquer do plano $P(x_1, y_1)$ e $Q(x_2, y_2)$, imprima a distância entre eles. A fórmula da distância é:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main()
{
    float x1, y1, x2, y2;
    float distancia;
    printf ("\nDigite o valor de x1: ");
    scanf ("%f", &x1);
    printf ("\nDigite o valor de y1: ");
    scanf ("%f", &y1);
    printf ("\nDigite o valor de x2: ");
    scanf ("%f", &x2);
    printf ("\nDigite o valor de y2: ");
    scanf ("%f", &y2);
    distancia=sqrt(pow((x2-x1),2)+pow((y2-y1),2));
    printf ("\nA distancia entre os pontos P1 e P2 eh %.2f\n", distancia);
    system("PAUSE");
```



Exercicio I

3) Construa um programa que calcule a quantidade de latas de tinta necessárias e o custo para pintar tanques cilíndricos de combustível, onde são fornecidos a altura e o raio desse cilindro.

Sabendo que:

- a lata de tinta custa R\$20,00
- cada lata contém 5 litros
- cada litro de tinta pinta 3 metros quadrados.

Sabendo que:

$\text{Área do cilindro} = 3.14 * \text{raio}^2 + 2 * 3.14 * \text{raio} * \text{altura}$
e que raio e altura são dados de entrada.





Resposta do Exercício I

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
main()
{
    float altura, raio, areaCilindro, qtdadeLitros, qtdadeLatas, custo;
    printf ("\nDigite o valor da altura em metros: ");
    scanf ("%f",&altura);
    printf ("\nDigite o valor do raio em metros: ");
    scanf ("%f",&raio);
    areaCilindro=3.14*raio*raio + 2*3.14*raio*altura;
    printf ("\nA area do cilindro eh %.2f metros quadrados", areaCilindro);
    qtdadeLitros=areaCilindro/3;
    printf ("\nA qtidade de litros necessaria eh de %.2f ", qtdadeLitros);
    qtdadeLatas=qtdadeLitros/5;
    printf ("\nA qtidade de latas necessaria eh de %.2f ", qtdadeLatas);
    custo=qtdadeLatas*20;
    printf ("\nO valor total das tintas eh de R$ %.2f \n", custo);
    system("PAUSE");
}
```



Operadores Relacionais

Operador	Exemplo	Comentário
<code>==</code>	<code>x == y</code>	O conteúdo de x é igual ao de y
<code>!=</code>	<code>x != y</code>	O conteúdo de x é diferente do de y
<code><=</code>	<code>x <= y</code>	O conteúdo de x é menor ou igual ao de y
<code>>=</code>	<code>x >= y</code>	O conteúdo de x é maior ou igual ao de y
<code><</code>	<code>x < y</code>	O conteúdo de x é menor que o de y
<code>></code>	<code>x > y</code>	O conteúdo de x é maior que o de y

As expressões relacionais em C retornam :

- **1** se verdadeiro e;
- **0** se falso.





Operadores Lógicos

- ▶ **&& (E lógico)**: retorna verdadeiro se ambos os operandos são verdadeiros e falso nos demais casos.
Exemplo: `if(a>2 && b<3)`.
- ▶ **|| (OU lógico)**: retorna verdadeiro se um ou ambos os operandos são verdadeiros e falso se ambos são falsos.
Exemplo: `if(a>1 || b<2)`.
- ▶ **! (NÃO lógico)**: usada com apenas um operando. Retorna verdadeiro se o operando é falso e vice-versa.
Exemplo: `if(!var)`.





Operadores Lógicos

Tabela E	Tabela OU	Tabela NÃO
$V \text{ e } V \rightarrow V$	$V \text{ ou } V \rightarrow V$	$\text{Não } V \rightarrow F$
$V \text{ e } F \rightarrow F$	$V \text{ ou } F \rightarrow V$	$\text{Não } F \rightarrow V$
$F \text{ e } V \rightarrow F$	$F \text{ ou } V \rightarrow V$	
$F \text{ e } F \rightarrow F$	$F \text{ ou } F \rightarrow F$	