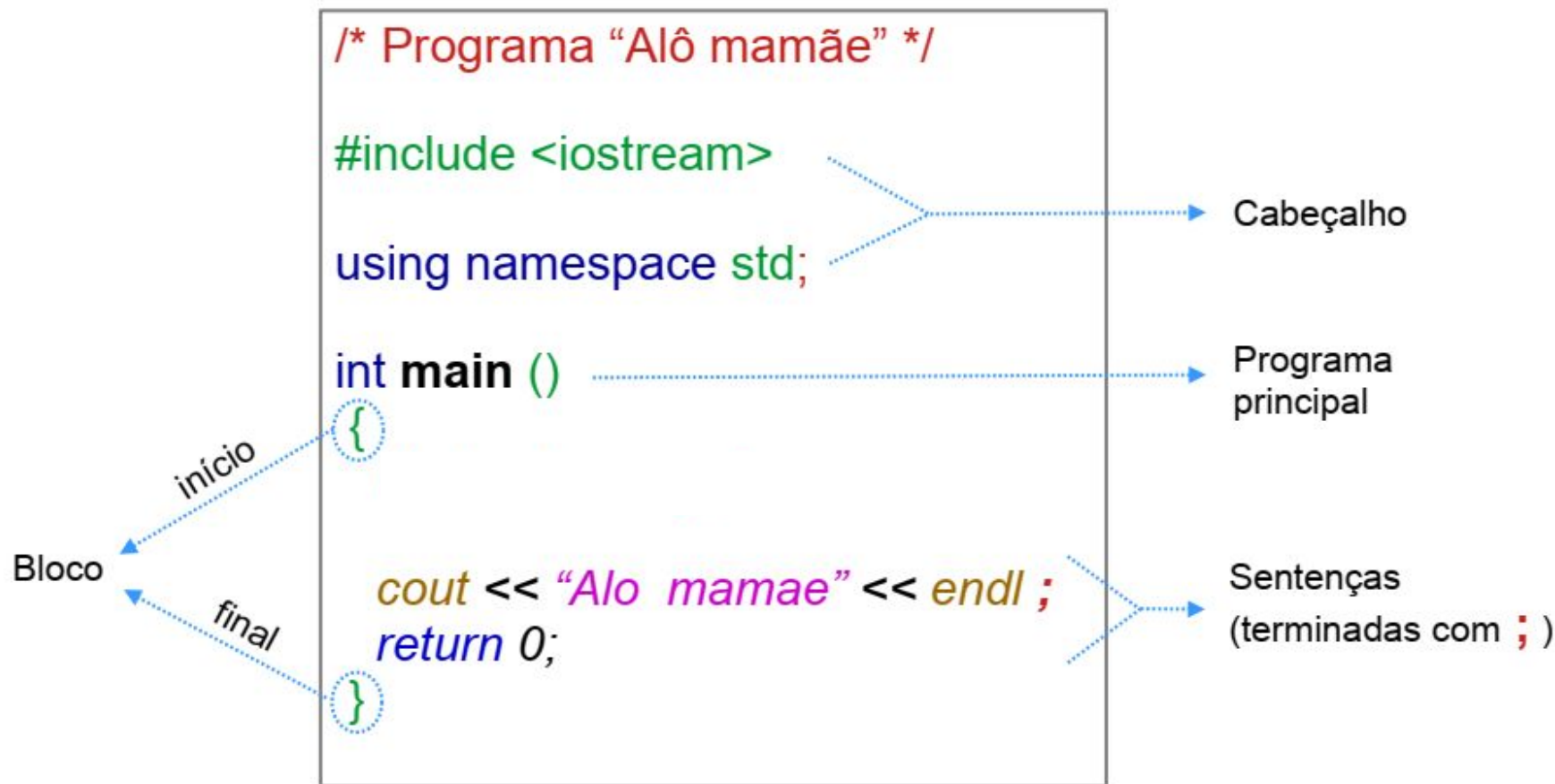


Programação de Computadores - CI-208

Prof. Murilo da Silva - DINF/UFPR

Aula 05

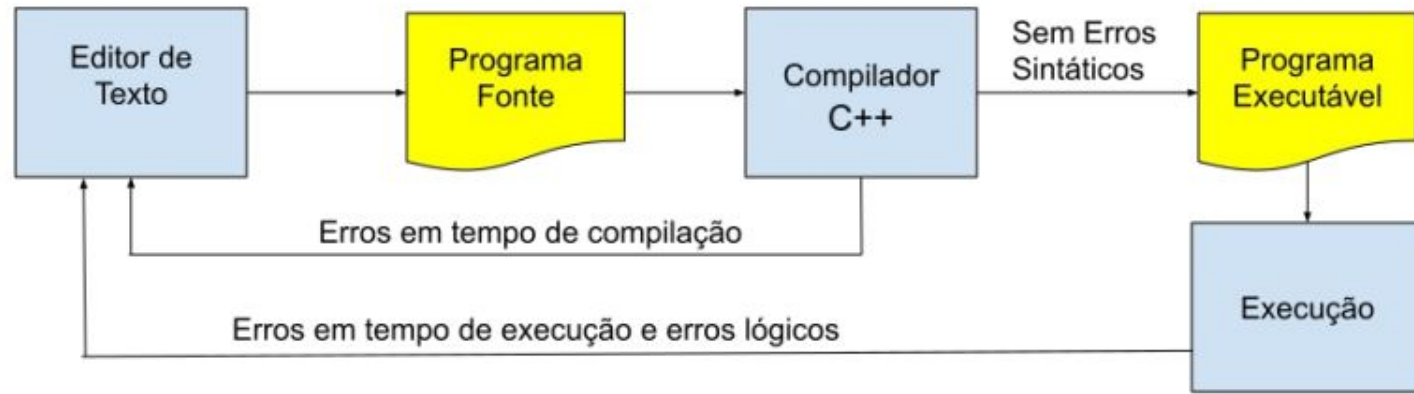
Estrutura de um Programa



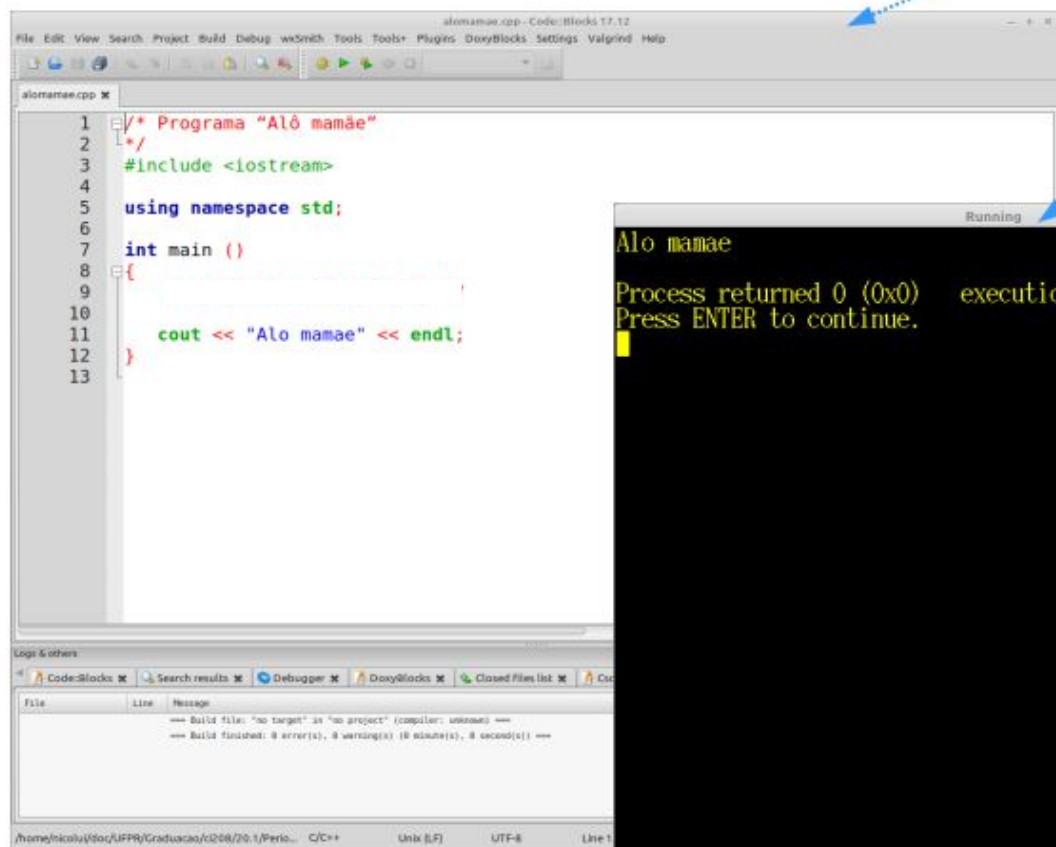
Processo de compilação

Código-fonte → Programa executável

```
/* Programa "Alô mamãe" */  
  
#include <iostream>  
  
using namespace std;  
  
int main ()  
{  
      
    cout << "Alo mamae" << endl;  
    return 0;  
}
```

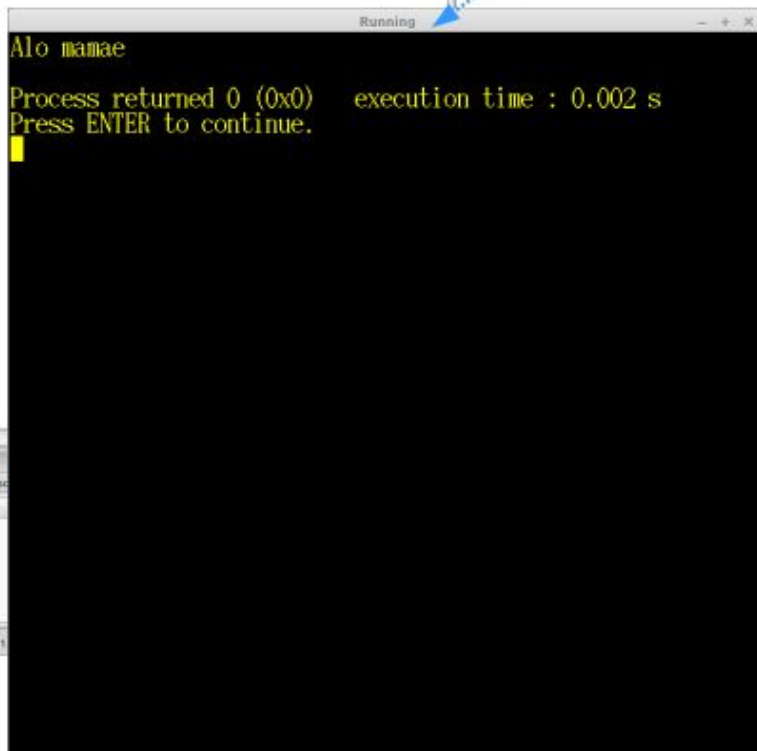


Processo de compilação



Janela do Code::Blocks

Janela do resultado
(aparece ao executar
o programa)



Tipos de Erros

Linha onde
compilador
"percebeu" um
erro

```
1  /* Programa "Alô mamãe"
2  */
3  #include <iostream>
4
5  using namespace std;
6
7  int main ()
8  {
9
10
11     cout << "Alo mamae" << endl;
12 }
```

Log & others

File	Line	Message
/home/nicolui/...		=== Build file: "no target" in "no project" (compiler: unknow...
/home/nicolui/...		In function 'int main()':
/home/nicolui/...	12	error: expected ';' before '}' token
/home/nicolui/...		=== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 s...

/home/nicolui/doc/UFPB/Graduacao/c208/20.1/Perio... C/C++ Unix (LF) UTF-8 Line 13, Col 1, Pos 156 Insert Read/Wri... default

Descrição de erro
de compilação

Variáveis

- Elementos usados para guardar valores em um programa
 - ▷ Cada variável representa um espaço em memória
 - ▷ TIPO de valor a representar → quantas posições de memória
 - inteiro, real, caracter, etc.
 - ▷ DECLARADAS antes das sentenças do programa (normalmente...)

```
/* Programa "Peso da Idade" */
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
    /* declarações */
```

```
    int idade;
```

```
    char inicial;
```

```
    float peso;
```

```
    bool casado;
```

```
    .....
```

```
}
```

Memória RAM

32	'a'	45.5	1			←..... conteúdo
100	104	105	109			←..... endereço
idade	inicial	peso	casado	←..... identificador no programa		

- **letra** seguida de sequência de letras, números e ' _
- palavras reservadas → NÃO
- sinais da linguagem → NÃO
- letras acentuadas → NÃO

Variáveis

Faixas de valores possíveis para cada tipo de variável

Tipo	Tamanho na memória (bytes)	Faixa de valores
int	4	-2147483648 → 2147483647
float	4	1.5e-45 → 3.4e38
double	8	5.0e-324 → 1.7e308
char	1	-128 → 127
bool	1	0 (false) 1 (true)

- Para expressar valores fixos como números e textos:
 - ▷ **Constantes** para valores numéricos: 3, 4.5, -23
 - ▷ **Literais** para caracteres de texto: 'a', 'B', '0', '1'
 - Delimitados por *apóstrofes*
 - ▷ **Strings** pra representar textos: "Meu nome eh Armando"
 - Delimitados por **aspas**

Comando de Atribuição

Sintaxe

variável = expressão ;

- Usado para definir o valor de uma variável
 - **variável** RECEBE valor de **expressão**

```
#include <iostream>
using namespace std;

int main ()
{
    int idade;
    float peso;
    char inicial;
    bool casado;

    idade = 32;
    inicial = 'a';
    peso = 45.5;
    casado = true;

    .....
}
```

Operadores Aritméticos

Para inteiros

- [unário]
* / % (MOD)
+ -

- `id = 10 * 2;`
- `id = id + 1;`
- `negativo = -id;` → **-21**

-[unário] → troca sinal do operando

Para reais

- [unário]
* /
+ -

- `peso = -7.5;`
- `peso = peso + 2.5;`
- `peso = peso / 2;` → **-2.5**

Temos um operador diferente: %

Precedência de Operadores Aritméticos

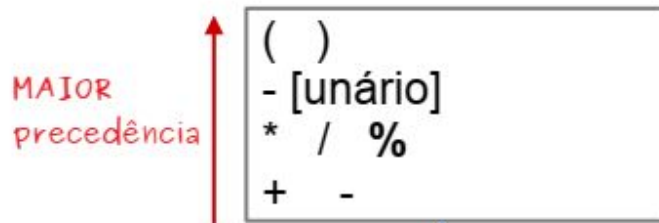
```
/* Programa "Precedencias" */
#include <iostream>
using namespace std;

int main ()
{
    float a, b, c;

    a = 10 * 2;
    b = 7.5 / 2.8;
    c = 2.0;

    b = b * 2.0 + a;    // * antes de +
    b = b * (2.0 + a);  // + antes de *
    b = a / c * 3;      // / antes de *
    b = a / (c * 3);    // * antes de /
    .....
}
```

Precedência de operadores:
prioridade de aplicação de operações



Lembrando que o operador %
(resto da divisão) só é válido
para operandos **inteiros**

Escrita

Comandos de escrita: exibe valores na tela (saída)

`cout << expr1 << expr2 << .... << exprn ;` [exibe e cursor fica na mesma linha]

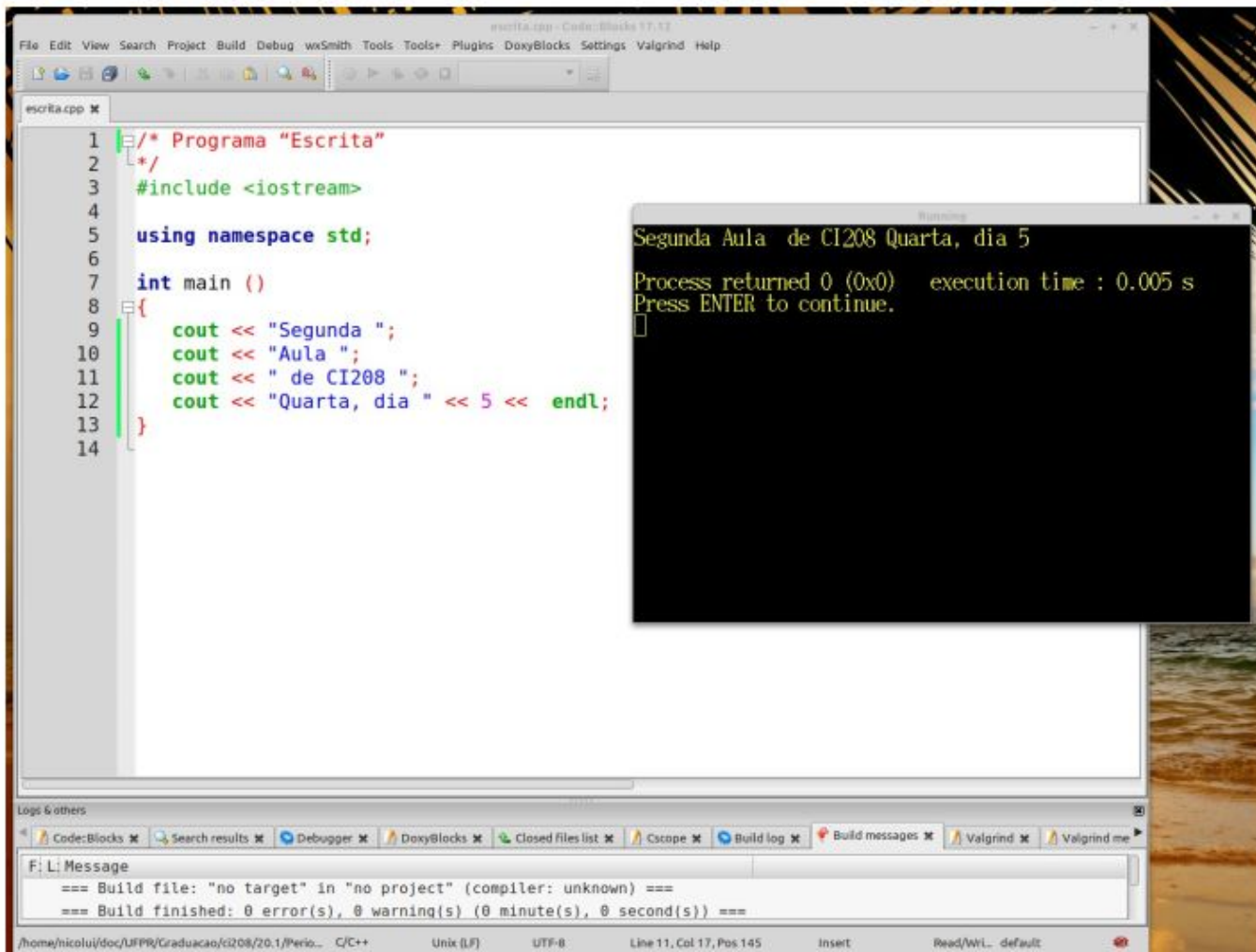
`cout << expr1 << expr2 << .... << exprn << endl ;` [exibe e cursor muda de linha]

ATENÇÃO: Exibe os valores um após o outro, SEM ESPAÇO entre eles

```
/* Programa "Escrita" */
#include <iostream>
using namespace std;

int main ()
{
    cout << "Segunda " ;
    cout << "Aula " ;
    cout << "de C1208 " ;
    cout << "Quarta , dia " << 5 << endl;
    return 0;
}
```

Escrita



The screenshot displays the Code::Blocks IDE interface. The main editor window shows a C++ program named "escrita.cpp". The code is as follows:

```
1  /* Programa "Escrita"
2  */
3  #include <iostream>
4
5  using namespace std;
6
7  int main ()
8  {
9      cout << "Segunda ";
10     cout << "Aula ";
11     cout << " de CI208 ";
12     cout << "Quarta, dia " << 5 << endl;
13 }
14
```

Overlaid on the right side of the IDE is a terminal window titled "Running". It shows the output of the program:

```
Segunda Aula de CI208 Quarta, dia 5
Process returned 0 (0x0)   execution time : 0.005 s
Press ENTER to continue.
```

At the bottom of the IDE, the "Logs & others" panel is visible, showing a message from the build system:

```
F:\L: Message
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
```

The status bar at the very bottom indicates the file path: `/home/nicolui/doc/UFPR/Graduacao/CI208/20.1/Perio...`, the editor is in C/C++ mode, UTF-8 encoding, and the cursor is at Line 11, Col 17, Pos 145.

Leitura

Comandos de leitura: lê valores para variáveis (entrada)

`cin >> var1 >> var2 >> .... >> varn ;`

```
/* Programa "Leitura" */
#include <iostream>
using namespace std;

int main ()
{
    int anoNasc;
    cin >> anoNasc;
    cout << "Em 31/12/2020 voce terah " , 2020 - anoNasc << " anos." << endl;
    return 0;
}
```

Leitura

```
/* Programa "Leitura" */  
#include <iostream>  
using namespace std;  
  
int main ()  
{  
    int valor1, valor2;  
  
    cin >> valor1 >> valor2;  
  
    cout << "Media = " << (valor1+valor2) / 2.0  
        << endl;  
    return 0;  
}
```

```
/* Programa "Leitura" */  
#include <iostream>  
using namespace std;  
  
int main ()  
{  
    int valor1, valor2;  
  
    cin >> valor1;  
    cin >> valor2;  
  
    cout << "Media = " << (valor1+valor2) / 2.0  
        << endl;  
    return 0;  
}
```



As duas formas de leitura são equivalentes