

COMPILAÇÃO, VARIÁVEIS E TIPOS DE DADOS

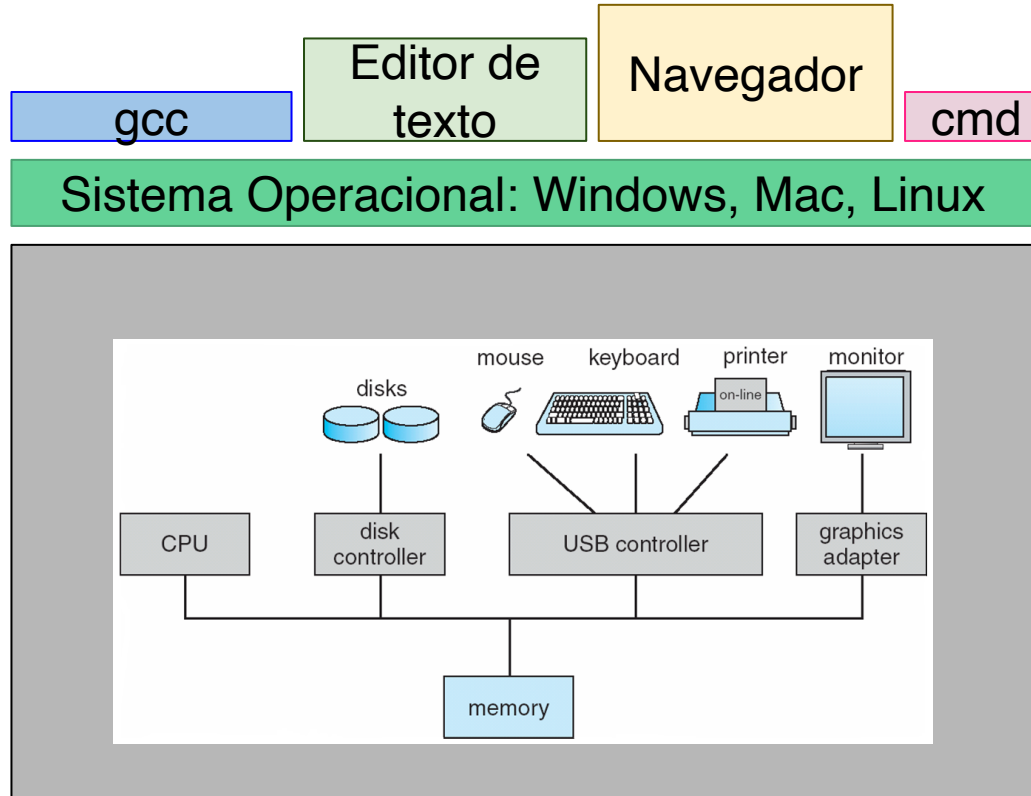
Programação e Desenvolvimento de Software I

Heitor Ramos

Antes de iniciar a aula

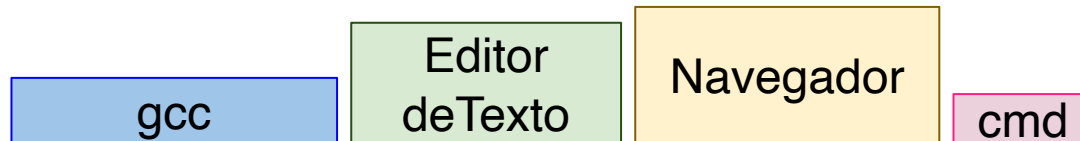
- O que é um computador?
 - Um computador é uma máquina que, a partir de uma entrada, realiza um número muito grande de cálculos matemáticos e lógicos, gerando uma saída.

Organização Básica



PDS-I

Um passo inicial para a criação de programas de aplicações...



Definições

- Para resolver um problema de computação é preciso escrever um **texto**.
- Este texto, como qualquer outro, obedece **regras de sintaxe**.
- Estas regras são estabelecidas por uma **linguagem de programação**.
- Este texto é conhecido como:

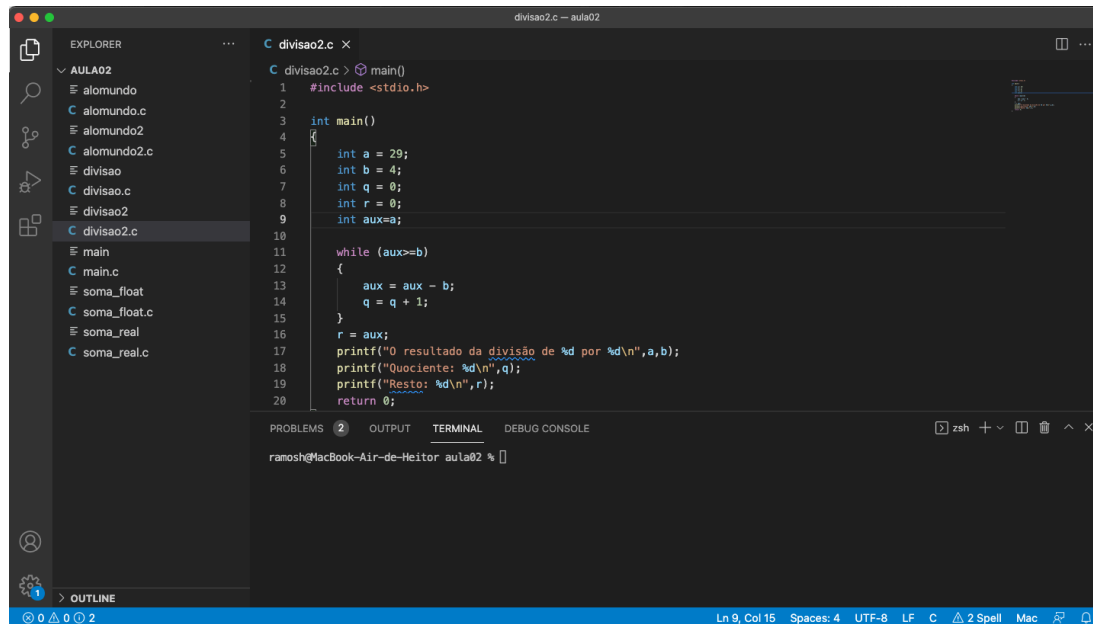
Programa

Definições

- Neste curso, será utilizada a [linguagem C](#).
- As [linguagem C](#) e [C++](#) possuem relações fortes, e, por isso, geralmente, os [ambientes de programação](#) da linguagem C são denominados ambientes C/C++.
- Um [ambiente de programação](#) contém:
 - [Editor de programas](#): viabiliza a escrita do programa.
 - [Compilador](#): verifica se o texto digitado obedece à sintaxe da linguagem de programação e, caso isto ocorra, traduz o texto para uma sequência de instruções em [linguagem de máquina](#).

Definições

- Que ambiente de programação iremos utilizar?
- Existem muitos, por exemplo: VSCode



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project named 'AULA02' with several files, including 'divisao2.c' which is currently selected. The main editor window shows the code for 'divisao2.c'. The code is a C program that calculates the division of two integers 'a' and 'b'. It includes the standard input/output library, defines a 'main' function, and uses a 'while' loop to perform the division by repeatedly subtracting 'b' from 'a'. It also uses 'printf' to output the result, quotient, and remainder. The bottom status bar indicates the current cursor position is at line 9, column 15.

```
divisao2.c - aula02
C divisao2.c x
C divisao2.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     int a = 29;
6     int b = 4;
7     int q = 0;
8     int r = 0;
9     int aux=a;
10
11     while (aux>=b)
12     {
13         aux = aux - b;
14         q = q + 1;
15     }
16     r = aux;
17     printf("O resultado da divisao de %d por %d\n",a,b);
18     printf("Quociente: %d\n",q);
19     printf("Resto: %d\n",r);
20     return 0;

```

PROBLEMS 2 OUTPUT TERMINAL DEBUG CONSOLE

ramosh@MacBook-Air-de-Meitor aula02 %

Ln 9, Col 15 Spaces: 4 UTF-8 LF C 2 Spell Mac

Linguagens de programação

- Linguagem de Máquina
 - Computador entende apenas pulsos elétricos
 - Presença ou não de pulso
 - 1 ou 0
- Tudo no computador deve ser descrito em termos de 1's ou 0's (binário)
 - Difícil para humanos ler ou escrever
 - $00011110 = 30$

Linguagem Assembly

- Um programa chamado montador (assembler) faz a transformação em código binário (código de máquina)

```
LOOP:  MOV A, 3  
        INC A  
        JMP LOOP
```

Linguagens de programação

- Linguagens de Alto Nível
 - Programas são escritos utilizando uma linguagem parecida com a linguagem humana
 - Independente da arquitetura do computador
 - Mais fácil programar
 - Uso de compiladores

Compilador

- Porque o compilador traduz o programa escrito na **linguagem de programação** para a **linguagem de máquina**?

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

int main(int argc, char *argv[])
{
    float y;
    y = sin(1.5);
    printf("y = %f", y);
    printf("\n");
    system("PAUSE");
    return 0;
}
```



Compilador



```
0101010110100010011
1000101010111101111
1010100101100110011
0011001111100011100
0101010110100010011
1000101010111101111
1010100101100110011
0011001111100011100
```

- Os computadores executam instruções que estejam escritas na forma de códigos binários.
- Um programa em **linguagem de máquina** é chamado de **programa executável**.

Linguagem C

- FORTRAN (FORmula TRANsform)
 - Em 1950, um grupo de programadores da IBM liderados por John Backus produz a versão inicial da linguagem;
 - Primeira linguagem de alto nível;
- Várias outras linguagens de alto nível foram criadas
 - Algol-60, Cobol, Pascal, etc

Linguagem C

- Uma das mais bem sucedidas foi uma linguagem chamada C
 - Criada em 1972 nos laboratórios por Dennis Ritchie
 - Revisada e padronizada pela ANSI em 1989
 - Padrão mais utilizado

COMPILANDO EM C

Compilando

```
#include <stdio.h>
int main(void) {
    printf("Alo Mundo\n");
    return 0;
}
```

Use o comando

```
gcc -Wall -std=c11 main.c -o alomundo
```

Compilando

```
gcc -Wall -std=c11 main.c -o alo
```

gcc → compilador

-Wall → exibir todos os erros (Warnings all)

-std=c11 → habilita C11 (2011)

main.c → seu arquivo

-o alo → nome do programa final

COMPUTANDO EM C

Problema

- Vamos supor por um momento que só sabemos fazer operações de:
 - Soma (+)
 - Subtração (-)
- Dado dois inteiros positivos **A** e **B**
- Determine o **quociente** e o **resto** da divisão de A por B.

Algoritmos

- Consistem de uma sequência finita de passos executados para obter um resultado
- Os passos não podem ser ambíguos

Algoritmo: Bolo de Chocolate

- Aqueça o forno a 180° C
- Unte uma forma redonda
- Numa taça
 - Bata
 - 75g de manteiga
 - 250g de açúcar
 - Até ficar cremoso
- Junte
 - 4 ovos, um-a-um
 - 100 g de chocolate derretido
- Adicione aos poucos 250g de farinha
- Deite a massa na forma
- Leve ao forno por 40 min

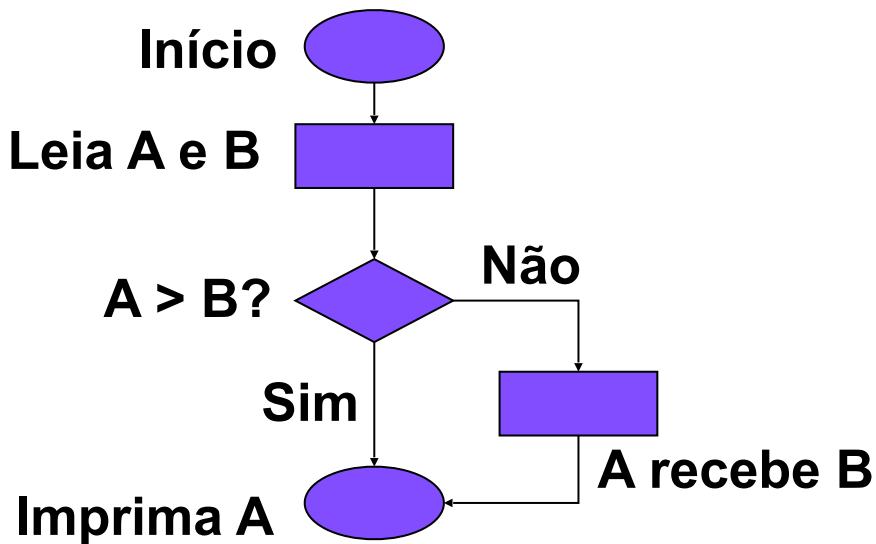
Algoritmo: Ambiguidade

- Em culinária, é comum ter receitas com:
 - Sal a gosto
 - Bata os ovos até atingir o ponto...
 - Açúcar o quanto baste
- E tem aquela da mulher do programador:
 - Pediu ao marido: vá à padaria e traga pães;
 - Se houver ovos, traga 12
 - Na volta, ele trouxe 12 pães

Pseudocódigo e Fluxograma

- Ex.: imprimir maior valor

```
Leia A;  
Leia B;  
Se A > B então  
  Imprima A;  
Senão  
  Imprima B;  
Fim Se
```



Algoritmos

- Consistem de uma sequência de passos executados para obter um resultado
- Lembrando do ensino fundamental, como resolver o problema anterior?

$a = 29$

$b = 4$

enquanto $a \geq b$:

$a = a - b$

$q = q + 1$

$r = a$

Algoritmos

- Consistem de uma sequência de passos executadas para obter um resultado
- Lembrando do ensino fundamental, como resolver o problema anterior?

a = 29

b = 4

enquanto a >= b:

 a = a - b

 q = q + 1

r = a

```
#include <stdio.h>
int main() {
    int a = 29;
    int b = 4;
    int q = 0;
    int r = 0;
    while (a >= b) {
        a = a - b;
        q = q + 1;
    }
    r = a;
    return 0;
}
```


Algoritmos

Pseudocódigo:

a = 29

b = 4

enquanto a >= b:

 a = a - b

 d = d + 1

r = a

C

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

```
int main() {
```

```
    int a = 29;
```

```
    int b = 4;
```

```
    int q = 0;
```

```
    int r = 0;
```

```
    while (a >= b) {
```

```
        a = a - b;
```

```
        q = q + 1;
```

```
    }
```

```
    r = a;
```

```
    return 0;
```

```
}
```

Algoritmos em C

- A linguagem oferece um conjunto de palavras chaves
 - if, for, while, else, ...
- Tipos de dados
 - int, float, double, long, char, ...
- Durante o curso vamos explorar cada uma delas. No momento, caso você não entenda uma delas, sem problemas.
 - Pergunte ao professor no contexto do exemplo/slide
 - Vai facilitar sua vida no futuro

```
#include <stdio.h>
int main() {
    int a = 27;
    int b = 4;
    int q = 0;
    int r = 0;
    while (a >= b) {
        a = a - b;
        q = q + 1;
    }
    r = a;
    printf("Ao dividir %d por %d ", a, b);
    printf("o quociente e igual a %d. ", q);
    printf("O resto e igual a %d", r);
    return 0;
}
```

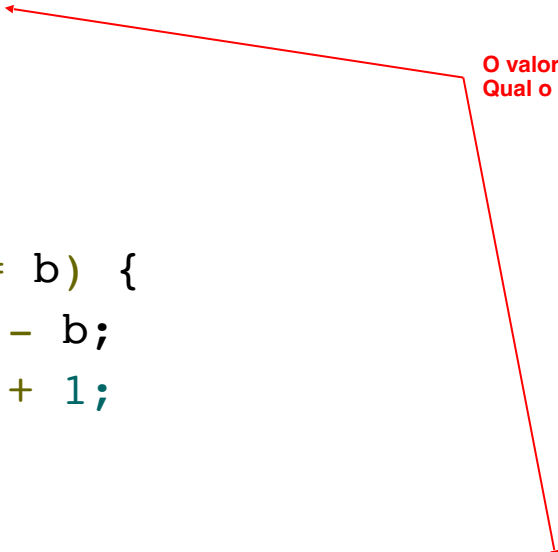
Qual vai ser a saída?

```
#include <stdio.h>
int main() {
    int a = 27;
    int b = 4;
    int q = 0;
    int r = 0;
    while (a >= b) {
        a = a - b;
        q = q + 1;
    }
    r = a;
    printf("Ao dividir %d por %d ", a, b);
    printf("o quociente e igual a %d. ", q);
    printf("O resto e igual a %d", r);
    return 0;
}
```

Parece que tem um erro no código!

Imprimindo o resultado

```
#include <stdio.h>
int main() {
    int a = 27;
    int b = 4;
    int q = 0;
    int r = 0;
    while (a >= b) {
        a = a - b;
        q = q + 1;
    }
    r = a;
    printf("Ao dividir %d por %d ", a, b);
    printf("o quociente e igual a %d. ", q);
    printf("O resto e igual a %d", r);
    return 0;
}
```



O valor de A mudou.
Qual o motivo?

VARIÁVEIS EM C

Variáveis

- Os dados que um programa utiliza precisam ser armazenados na **memória** do computador.
- Cada posição de memória do computador possui um **endereço**.

Variáveis

- Os dados que um programa utiliza precisam ser armazenados na **memória** do computador.
- Cada posição de memória do computador possui um **endereço**.



Cada gaveta tem uma etiqueta e um espaço bem delimitado. No entanto, você pode guardar diversas coisas dentro delas.

Variáveis

- No exemplo anterior, quais são as variáveis?

```
#include <stdio.h>
int main() {
    int a = 27;
    int b = 4;
    int q = 0;
    int r = 0;
} 4 vars: a, b, q, r.
while (a >= b) {
    a = a - b;
    q = q + 1;
}
r = a;
printf("Ao dividir %d por %d ", a, b);
printf("o quociente e igual a %d. ", q);
printf("O resto e igual a %d", r);
return 0;
}
```

Resolvendo o erro

- Como resolver o mesmo com uma variável a mais?

```
#include <stdio.h>
int main() {
    int a = 27;
    int b = 4;
    int q = 0;
    int auxiliar = a;
    while (auxiliar >= b) {
        auxiliar = auxiliar - b;
        q = q + 1;
    }
    int r = auxiliar;
    printf("Ao dividir %d por %d ", a, b);
    printf("o quociente e igual a %d. ", q);
    printf("O resto e igual a %d", r);
    return 0;
}
```

Quais foram as mudanças?

Tipos de erro

- Erros de compilação:
 - Caso o programa não esteja de acordo com as regras da linguagem, erros de compilação ocorrerão. Ler e entender estes erros é muito importante.
- Erros de lógica/execução:
 - Seu programa está errando por algum motivo
 - Variáveis inicializadas de forma errada
 - Tipos de dados diferentes
 - ...

Erros de Compilação

```
#include <stdio.h>
int main(void) {
    printf("Alo Mundo\n");
    return 0;
```

```
#include <stdio.h>
int main(void) {
    printf("Alo Mundo\n");
    return 0;
}
```

Outro Problema

- Imagine que você tem duas variáveis:
 - X e Y
- Cada uma dessas representa um número real
- Queremos computar a soma dessas duas variáveis
- Como fazer?

Solução A

```
#include <stdio.h>
int main() {
    int x = 7.2;
    int y = 2.3;
    int resultado = x + y;
    printf("O resultado de %d + %d foi %d", x, y,
resultado);
    return 0;
}
```


Solução A: 7 + 2 = 9?

```
#include <stdio.h>
int main() {
    int x = 7.2;
    int y = 2.3;
    int resultado = x + y;
    printf("O resultado de %d + %d foi %d", x, y,
resultado);
    return 0;
}
```