

Lógica de Programação

Aula 01 - Introdução

Prof. JOSE PICOVSKY

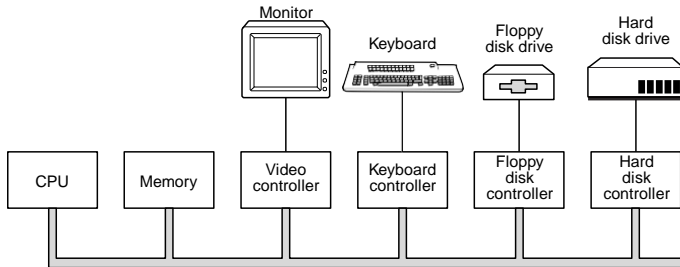
Primeiro Semestre de 2021

- 1 Organização de um computador
- 2 Algoritmos
- 3 A linguagem C

- Computador: o que computa, calculador, calculista. (dicionário Houaiss).
- Os primeiros “computadores” eram humanos que calculavam tabelas de logaritmos ou trajetórias pra canhões, seguindo procedimentos bem definidos.
- Um computador é uma máquina que, a partir de uma entrada, realiza um número muito grande de cálculos matemáticos e lógicos, gerando uma saída.

A linguagem nativa do computador é codificada numericamente, de forma binária:

- Bit → Pode assumir valores 0 ou 1.
- Byte → Agrupamento de 8 bits em uma palavra.
- Letras e símbolos são representados por números.



- Computadores realizam tarefas complexas por meio de um número enorme de operações simples.
- Para gerenciar a complexidade das soluções, existe uma hierarquia de funções, onde cada uma apresenta uma interface mais simples.

Programas de Aplicação		
Shell	Compiladores	Editores
Sistema operacional		
Hardware		

- Como usuários, interagimos com os programas de aplicação.
- Neste curso iremos descer nesta hierarquia, para construirmos novos programas de aplicação.

- Lógica: é a técnica de encadear pensamentos para atingir determinado objetivo;
- Sequência Lógica: são passos executados até atingir um objetivo ou solução de um problema.
- Instruções: um conjunto de regras ou normas definidas para a realização ou emprego de algo. É o que indica a um computador uma ação elementar a executar.

- Sequência de passos, precisos e bem definidos, para a realização de uma tarefa.
- Não pode ser redundantes, nem subjetivos e não ambíguo;

Exemplo 1: Como chupar uma bala?

- 1 Pegar a bala;
- 2 Retirar o papel;
- 3 chupar a bala;
- 4 Jogar o papel no lixo.

- Algoritmos podem ser especificados de várias formas, inclusive em português.

Exemplo 2: Fritar um ovo?

- 1 .
- 2 ..
- 3 ...
- 4

Exemplo 3: Somar dois valores?

- 1 .
- 2 ..
- 3 ...
- 4

- Crie uma sequência lógica para tomar banho.
- Faça um algoritmo para somar dois números e multiplicar o resultado pelo primeiro número.
- Descreva com detalhes a sequência lógica para Trocar um pneu de um carro.
- Faça um algoritmo para trocar uma lâmpada. Descreva com detalhes.

- Pseudocódigo;
- Diagrama de blocos ou fluxograma.

- São independentes das linguagens de programação;
- Devem ser fácil de se interpretar e fácil de codificar;
- Devem ser o intermediário entre a linguagem falada e a linguagem de programação (e.g., C, Java e Python).
- Regras para construção:

- Regras para construção:
 - Usar somente um verbo por frase;
 - Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática;
 - Usar frases curtas e simples;
 - Ser objetivo;
 - Procurar usar palavras que não tenham sentido dúbio.







- Exemplo: Calcular a média aritmética dos alunos. Os alunos realizarão quatro provas: P1, P2, P3 e P4.
 - Quais são os dados de entrada?
 - Qual será o processamento a ser utilizado?
 - Quais serão os dados de saída?

- Algoritmo:
 - Receba a nota da prova 1
 - Receba a nota da prova 2
 - Receba a nota da prova 3
 - Receba a nota da prova 4
 - Some todas as notas e divida o resultado por 4
 - Mostre o resultado da divisão

- Identifique os dados de entrada, processamento e saída no algoritmo abaixo:
 - Receba código da peça;
 - Receba valor da peça; Receba
 - Quantidade de peças;
 - Calcule o valor total da peça ($\text{Quantidade} * \text{Valor da peça}$);
 - Mostre o código da peça e seu valor total.

- É uma forma padronizada e eficaz para representar os passos lógicos de um determinado processamento;
- Pode-se definir uma sequência de símbolos, com significado bem definido;
- Principal função é facilitar a visualização dos passos de um processamento.

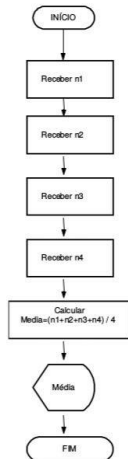
Símbolo	Função
 TERMINAL	Indica o INÍCIO ou FIM de um processamento Exemplo: Início do algoritmo
 PROCESSAMENTO	Processamento em geral Exemplo: Calculo de dois números
 ENTRADA DE DADO MANUAL	Indica entrada de dados através do Teclado Exemplo: Digite a nota da prova 1
 EXIBIR	Mostra informações ou resultados Exemplo: Mostre o resultado do calculo

- Fluxograma “Chupar uma bala”?

- Fluxograma “Chupar uma bala”?



- Fluxograma “Calcular média aritmética dos alunos”?



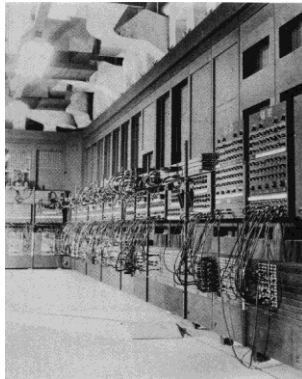
- Construa um diagrama de blocos que:
 - Leia a cotação do dólar;
 - Leia um valor em dólares;
 - Converta esse valor para Real;
 - Mostre o resultado;
- Desenvolva um diagrama que:
 - Leia 4 (quatro) números;
 - Calcule o quadrado para cada um;
 - Somem todos os números; Mostre o resultado;

- Construa um algoritmo para pagamento de comissão de vendedores de peças, levando-se em consideração que sua comissão será de 5% do total da venda e que você tem os seguintes dados:
 - Identificação do vendedor;
 - Código da peça;
 - Preço unitário da peça;
 - Quantidade vendida;
- Construa o diagrama de blocos do algoritmo desenvolvido no exercício anterior.

- Como transformar um algoritmo em linguagem que o computador entenda?
- Deve ser capaz de expressar tudo o que o computador pode fazer.
- Não pode ser ambígua.

Programação em código absoluto ou binário (apenas 0s e 1s).

ENIAC



Um programa, chamado montador ou assembler, faz a transformação em código absoluto.

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

- Mais distantes da máquina e mais próximas de linguagens naturais (inglês, português, etc.).
- Mesmo mais compreensíveis, elas não são ambíguas.
- Um **compilador** as transforma em código executável.

Exemplos de linguagens

- C
- Pascal
- Java

Um programa em C é um arquivo texto, contendo declarações e operações da linguagem. Isto é chamado de **código fonte**.

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

```
main() {  
    printf("Hello, world!\n");  
}
```

Para executar um programa a partir do seu código fonte é necessário compilá-lo, gerando **código binário** ou **executável**. Este pode ser executado como qualquer outro programa de aplicação.

```
$ gcc hello.c -o hello
$ hello
Hello, world!
```

Caso o programa não esteja de acordo com as regras da linguagem, erros de compilação ocorrerão. Ler e entender estes erros é muito importante.

```
#include <stdio.h>
main() {
    printf("Hello, world!\n");
```

```
$ gcc hello.c -o hello
hello.c: In function 'main':
hello.c:5: error: syntax error at end of input
```

Acontecem quando o comportamento do programa diverge do esperado e podem acontecer mesmo quando o programa compila corretamente.

```
#include <stdio.h>
main() {
    printf("Hello, world! $#%#@%\n");
}
```

```
$ gcc hello.c -o hello
$ hello
Hello, world! $#%#@%
```

- Ferramenta que executa um programa passo a passo.
- Ajuda a encontrar erros de execução (bugs).

Exemplo

- gdb


```
#include <stdio.h>
main() {
    int x, y;
    printf("x: ");
    scanf("%d", &x);
    printf("y: ");
    scanf("%d", &y);
    if (x > y)
        printf ("O maior número é x = %d\n", x);
    else
        printf ("O maior número é y = %d\n", y);
}
```