

- Introdução
- *Hardware versus Software*
- *Arquitetura Básica do Computador*
- Fases de Desenvolvimento de um Programa
- Algoritmia e Modelação de Problemas

História e Evolução das Linguagens de Programação

<https://www.youtube.com/watch?v=sqjxrNUSBm4>

<https://www.youtube.com/watch?v=Pn5znSOGHcs>

Evolução das Linguagens de Programação

<https://www.youtube.com/watch?v=Og847HVwRSI>

O que é um Computador?

- **Computador** – máquina que processa informação sob o controlo de um programa.
- **Hardware** -- componentes electrónicos e mecânicos.
- **Software** -- programas que “controlam” o computador.
- **Computados com objectivos Genéricos** – é capaz de alterar o seu programa de controlo.
- **Computados com objectivos Especiais** – possuem um programa de controlo fixo.

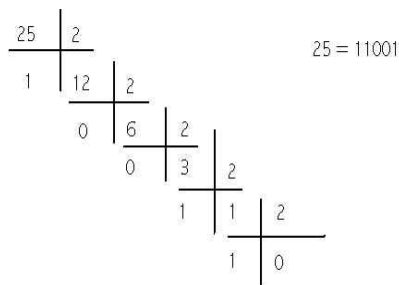
Representação da Informação

- Apesar da diversidade de informação que pode ser apresentada num computador (texto, fotos, sons, vídeo,...), internamente existem apenas 2 valores:
 - bit (**B**inary **digiT**): pode assumir os valores 0 ou 1 (correspondente aos estados de aberto ou fechado de um interruptor)
 - ASCII (**A**merican **S**tandard **C**ode **I**nformation **I**nterchange)

Carácter	Código Binário
A	0100 0001
B	0100 0010
C	0100 0011

Representação da Informação

■ Conversão Decimal \leftrightarrow Binário:

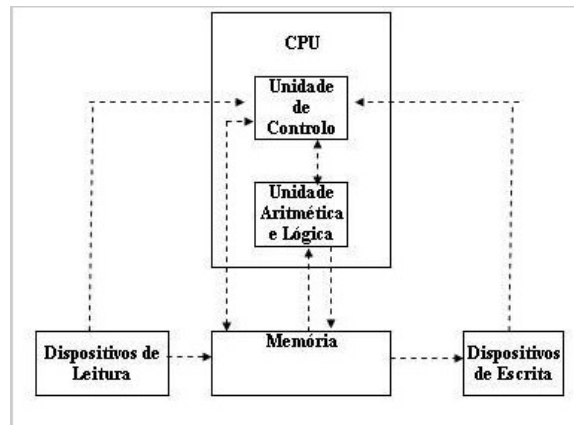


$$101011_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 43_{10}$$

Unidades

Unidade	Equivalência
bit	Unidade de informação mais pequena. Pode assumir o valor 0 ou 1
Byte	8 bits. Usada na representação de caracteres. Por exemplo: A \rightarrow 1 byte; Pedro \rightarrow 5 bytes
KiloByte	Também representado por KB. 1KB = 1024 Bytes (equivale à potência de 2 mais próxima de 1000, ou seja 2^{10})
MegaByte	Também representado por MB. 1MB = 1024KB = 1048576 Bytes
GigaByte	Também representado por GB. 1GB = 1024MB = 1048576KB = 1073741824 Bytes
TeraByte	Também representado por TB. 1TB = 1024GB = 1048576MB = 1073741824KB = 1099511627776 Bytes

Componentes de um Computador



Hardware (Entrada/Saída)

- Dispositivos de Saída
 - Mostram a informação numa forma perceptível para os seres humanos.
 - Exemplos: Impressoras, monitores,...
- Dispositivos de Entrada
 - Fornecem dados ao computador.
 - Exemplos : Teclado, rato, *scanner*.

Hardware (Memória)

■ Memória Primária ou Principal

- *Temporária* Armazenam dados e programas temporariamente durante o processamento.
- *Volátil*: Dados perdem-se quando se desliga a alimentação.
- *Rápida*: Completamente electrónica.

■ Memória Secundária

- *Memória não volátil* de armazenamento de dados e programas
- *Lenta*: Requer movimentos mecânicos.
- *Exemplos*: Discos duros, Compact Disk(CD), Pen(CDs).

Memória

- *Memória* é um componente essencial em qualquer computador.
- A memória consiste numa sequência ordenada de localizações de armazenamento – células de memória.
- Cada célula de memória possui um *endereço* único que indica a posição relativa na memória principal do computador.

0000	-25.2
0001	H
0002	ADD 3
...	
998	X
999	81.3

Hardware (Processador)

■ CPU - Central Processing Unit

(Unidade Central de Processamento)

- O *microprocessador*, tal como Intel Pentium, Apple Power PC, i7, ...
- Controlado por *software*.
- *Unidade de Controlo*
 - Envia sinais para controlar o hardware.
- *Unidade Aritmética e Lógica (ALU)*
 - Realiza operações aritméticas (+), relacionais (>) e lógicos (&).

Tipos de *Software*

■ *Software de Aplicação*

- Programas que efectuem uma tarefa ou serviço específico.
- *Exemplos*: Processador de Texto, folha de cálculo, jogos de computador, programa de E-mail.

■ *Software de Sistema*

- Programas que efectuem operações básicas.
- *Exemplos* : O *Sistema Operativo* permite que o utilizador corra os seus programas, guarde o seu trabalho em ficheiros, envie mensagens utilizador, ...

Linguagens de Programação

■ Linguagens de Alto Nível

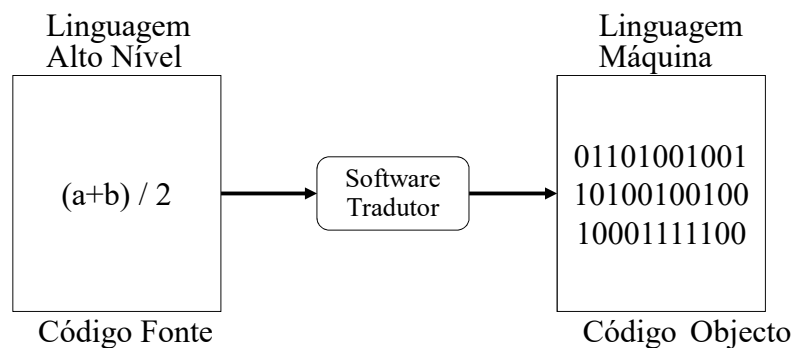
- Facilmente lidas por humanos -- $(a + b) / 2$
- Usada para escrever a maioria do software.
- Não pode ser compreendida directamente pelo computador.
- Exemplos: FORTRAN, Pascal, Java, C, C++, C#, BASIC, Python, Swift.

■ Linguagem Máquina

- A única linguagem compreendida pela CPU.
Código Binário -- 0010010010100010101

Tradução de uma Linguagem

Uma expressão em linguagem de alto nível $(a+b)/2$ requer várias operações primitivas em linguagem máquina.



Interpretação versus Compilação

■ Interpretação

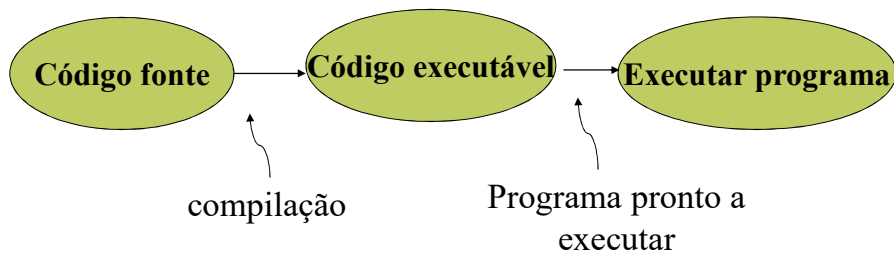
- ❑ Cada linha de código é decodificada e interpretada pelo computador à medida que o programa é executado.
- ❑ Instruções em ciclo podem ter de ser interpretadas várias vezes.
- ❑ Ineficiente e torna a execução mais lenta.
- ❑ Python, BASIC, Perl,... são interpretadas.

Interpretação *versus* Compilação

■ Compilação

- ❑ A maioria das linguagens é compilada C, C++, COBOL, FORTRAN ...
- ❑ Todo o programa é primeiro analisado pelo compilador.
- ❑ Conversão em Linguagem Máquina.
- ❑ Execução muito mais rápida.
- ❑ Duas partes – compilação em código objecto e depois ligação para executável.

Na linguagem C



Programa de Computador

O que é?

- Sequência de Instruções que o computador deve seguir, com vista à execução de uma tarefa específica.
- A estratégia a seguir deve respeitar:
 - **Fase de Resolução do Problema**
 - **Fase de Implementação**

Fase de Resolução do Problema

- **Análise** - Compreender (definir) o problema.
- **Algoritmo (Solução Geral)** – Sequência lógica de passos utilizados para resolver o problema.
- **Prova** - Verificar se a sequência de passos resolve realmente o problema .

Fase de Implementação

- **Programa (Solução Específica)** Traduzir o algoritmo numa Linguagem de Programação específica.
- **Prova** – Após a execução das instruções por parte de computador, comprovar os resultados e efectuar as correcções, se necessário.
- **Utilização** – Usar o programa.

Algoritmo

- Sequência Ordenada e Precisa de acções que conduzem à solução de um problema.
- Principais características:
 - Ausência de Ambiguidade
 - Número Finito de Acções
 - Genérico

Algoritmos

- Técnicas de Representação
 - Pseudocódigo
 - Fluxogramas
- Estruturas Algorítmicas
 - Sequência
 - Repetição
 - Selecção

Algoritmos

Pseudocódigo

- ❑ INICIO / FIM
- ❑ LER / ESCREVER
- ❑ SE... ENTÃO ...SENÃO
- ❑ ESCOLHA ...
- ❑ FAZ ... ENQUANTO...
- ❑ ENQUANTO...FAZ...FIM ENQUANTO
- ❑ PARA ... ATÉ ...FAZ...FIM PARA

Algoritmos Fluxograma

■ Decisão



■ Decisão Múltipla



■ Início/Fim



■ Cálculo



■ Leitura/Escrita



■ Linhas de Fluxo



Algoritmos

Exemplo – Elabore um algoritmo (pseudocódigo e fluxograma) que calcule o dobro de um número escolhido pelo utilizador.

ALGORITMO CalculaDobro

Dados: numero (inteiro)

Auxiliares:

Resultados: dobro (inteiro)

INICIO

LER numero

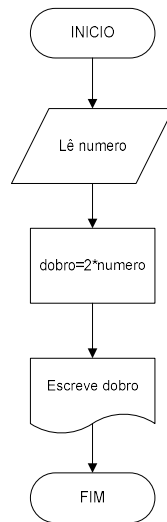
$\text{dobro} \leftarrow 2 * \text{numero}$

ESCREVER dobro

FIM

Pseudocódigo

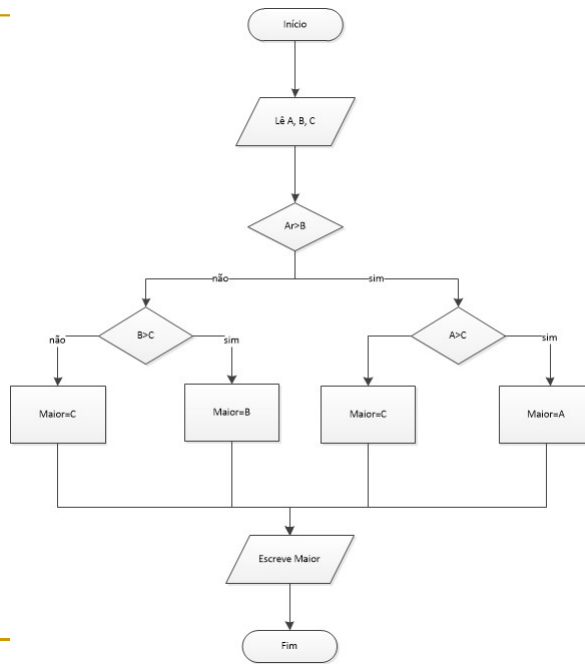
Fluxograma



Algoritmos

Exemplo – Elabore um algoritmo (pseudo código e fluxograma) que, dados 3 números, todos diferentes, determine e mostre o maior dos três.

Fluxograma



Introdução à Programação

29

ALGORITMO Maior

Dados: A,B,C (inteiro)

Auxiliares:

Resultados: maior (inteiro)

INICIO

LER (A,B,C)

SE (A>B) **ENTÃO**

SE (A>C) **ENTÃO**

maior ← A

SENÃO

maior ← C

SENÃO

SE (B>C) **ENTÃO**

maior ← B

SENÃO

maior ← C

ESCREVER ("O número maior é", maior)

FIM

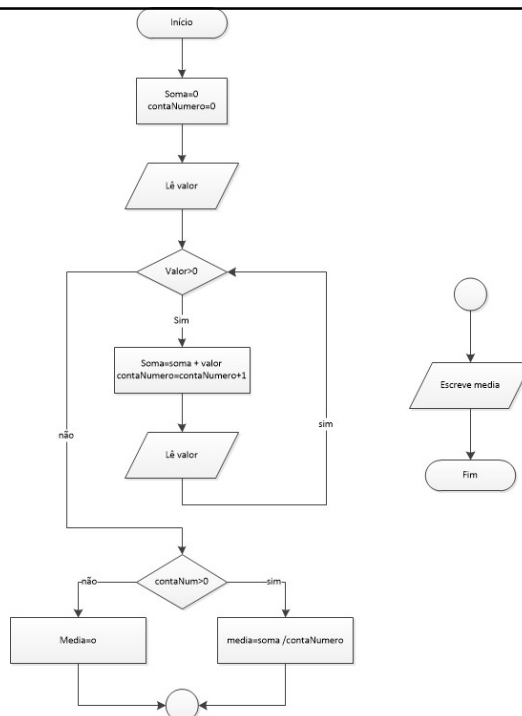
Introdução à Programação

30

Algoritmos

Exemplo – Elaborar um algoritmo que calcule a média de um conjunto de números. Quando for digitado um número negativo ou nulo termina a introdução de dados.

Fluxograma



Algoritmos

Exemplo #1 - Pseudocódigo

```
INICIO
    soma ← 0
    contaNumero ← 0
    LER valor
    ENQUANTO ( valor > 0 ) FAZ
        soma ← soma + valor
        contaNumero ← contaNumero + 1
        LER valor
    FIM ENQUANTO
    SE (contaNumero = 0) ENTÃO
        media ← 0
    SENÃO
        Media ← soma / contaNumero

    ESCREVER Media
FIM
```

Introdução à Programação

33

Algoritmo

Arranque de um Automóvel

1. Colocar a chave.
2. Verificar se o carro está em ponto morto.
3. Dar à chave.
4. Se o motor arranca antes de 6 s, podemos andar.
5. Se o motor não arranca antes de 6 s, esperar 10 s e repetir os passos 3, 4 e 5.
6. Se o carro não arranca chamar um mecânico.

Introdução à Programação

35

Algoritmo

Somar dois números pedidos ao
utilizador

1. Ler o primeiro número
2. Ler o segundo número
3. Somar os dois números
4. Mostrar resultado