



# Introdução à Computação

## *História do computador:*

Computação:

- Do latim *computacione*: ato ou efeito de computar
- Dicionário Aurélio: “disciplina que estuda o uso de computadores em cálculos, simulações, etc., com o objetivo de auxiliar cientistas em suas pesquisas”.

Informática:

- Informação + automática
- Dicionário Aurélio: “ciência que visa o tratamento da informação através do uso de equipamentos e procedimentos da área de processamento de dados”.

Surgiu há milhares de anos pela necessidade humana de tornar as atividades/funções mais ágeis, como por exemplo, Povos primitivos utilizavam os dedos para contar: origem do sistema de numeração decimal. Logo, a computação sempre foi utilizada para facilitar atividades.

Também é útil para atividades comuns, como:

- Comércio, agricultura, períodos de plantio, colheita, etc.

## *Evolução dos Computadores:*

A palavra “computador” vem do verbo “computar” que, por sua vez, significa “calcular”. Sendo assim, podemos pensar que a criação de computadores começa na idade antiga, já que a relação de contar já intrigava os homens.

## *Primeira geração:*

- 1951-1959;
- Os computadores funcionavam por meio de circuitos e válvulas eletrônicas;

- Possuíam o uso restrito, além de serem imensos e consumirem muita energia.

### *Segunda geração:*

- 1959-1965;
- Dimensões muito grandes;
- Funcionavam por meio de transistores, os quais substituíram as válvulas que eram maiores e mais lentas. Nesse período já começam a se espalhar o uso comercial.

### *Terceira geração:*

- 1965-1975;
- Os computadores funcionavam por meio de circuitos integrados;
- Esses substituíram os transistores e já apresentavam uma dimensão menor e maior capacidade de processamento;
- Foi nesse período que os chips foram criados e a utilização de computadores pessoais começou.

### *Quarta geração:*

- 1975 - até os dias atuais;

Com o desenvolvimento da tecnologia da informação, os computadores diminuem de tamanho, aumentam a velocidade e capacidade de processamento de dados. São incluídos os microprocessadores com gasto cada vez menor de energia.

Além disso, surgem os softwares integrados e a partir da virada do milênio, começam a surgir os computadores de mão. Ou seja, os smartphones, iPod, iPad e tablets, que incluem conexão móvel com navegação na web.

Alguns estudiosos preferem acrescentar a “Quinta Geração de Computadores” com o aparecimento dos supercomputadores, utilizados por grandes corporações como a NASA. Nessa geração, é possível avaliar a evolução da tecnologia multimídia, da robótica e da internet.

### *Conceitos básicos de informática e arquitetura dos computadores:*

Para iniciar a discussão sobre computadores é necessário se estabelecer dois conceitos iniciais:

- **HARDWARE** – são os componentes físicos do computador, ou seja, a máquina propriamente dita. É composto também pelos periféricos (dispositivos de entrada e saída) como teclado, mouse, monitor, etc.

- **SOFTWARE** – é a parte lógica do computador, representado pelos programas que nos permitem utilizar a máquina (hardware). Sua principal função é realizar a interface entre um computador e o usuário.

Podemos agrupar os elementos do hardware em diferentes categorias: **dispositivos de entrada de dados; dispositivos de saída; CPU; Placa-mãe; HD; memória.**

**Os dispositivos de entrada** - nos permitem alimentar os programas que serão utilizados fornecendo dados que serão processados e nos são “devolvidos” pelos dispositivos de saída. Exemplo: teclado, mouse, webcam, disquete, CD, DVD, etc.

**Dispositivo de saída** - são dispositivos que exibem dados e informações processadas pelo computador, também chamados de unidades de saída. Exemplo: impressoras, caixas de som, projetores multimídia, etc.

**A Unidade Central de Processamento – CPU** - tem sua sigla decorrente da expressão em inglês: Central Processing Unit. A CPU é responsável pelo processamento das informações, sendo entretanto muitas vezes confundida com o “gabinete” da máquina. Porém ela é o próprio processador ou chip, onde são processados os dados.

**Placa mãe** - para entender melhor precisamos conhecer o processador, o processador tem 3 funções básicas:

- Realizar cálculos de operações aritméticas e comparações lógicas.
- Manter o funcionamento de todos os equipamentos e programas, pois a unidade de controle interpreta e gerencia a execução de cada instrução do programa;
- Administrar na memória central (principal) além do programa submetido, os dados transferidos de um elemento ao outro da máquina, visando o seu processamento.

Agora que entendemos o processador vamos conhecer a placa mãe. Se o processador pode ser considerado o “cérebro” do computador, a placa-mãe (do inglês motherboard) representa a espinha dorsal, interligando os demais periféricos ao processador. Antigamente, a placa-mãe funcionava apenas como um ponto central, contendo os ocupantes pelos demais componentes, comprados separadamente: processador, pentes de memória, placa de vídeo, modem, placa de som e rede, entre outros. Cada um desses componentes era uma placa à parte, interligados pela placa-mãe. Atualmente as placas “onboard” já apresentam muitos desses

componentes, porém para se aumentar o desempenho ou recursos, é possível instalar placas adicionais ampliando os componentes originais.

**O disco rígido** - do inglês hard disk, também conhecido como HD serve como unidade de armazenamento permanente, guardando dados e programas. Ele armazena os dados em discos magnéticos que mantêm a gravação por vários anos. Esses discos giram a uma alta velocidade e têm seus dados gravados ou acessados por um braço móvel composto por um conjunto de cabeças de leitura capazes de gravar ou acessar os dados em qualquer posição nos discos.

**Memória** - basicamente há dois tipos de memória em um computador. ROM e RAM

- Memória ROM (Read-Only Memory) – também conhecida como memória secundária ou ainda auxiliar do computador, é responsável pelo armazenamento permanente de informações como dados ou programas, que não estão não estão sendo usados ativamente por outros dispositivos. Exemplos de memória ROM são os CDs, DVDs e o próprio HD.
- Memória RAM (Random Access Memory) – também conhecida como memória principal do computador é responsável pelas informações que estão sendo processadas/utilizadas. É uma memória de acesso aleatório com conteúdo volátil, que é perdido pelo desligamento do computador ou com a falta de energia elétrica.

**Software** - é o conjunto de programas que permite o funcionamento e utilização da máquina (hardware).

Sistema operacional - o primeiro software necessário para o funcionamento de um computador é o Sistema Operacional, que é, na verdade, "invisível". Sua principal função é reconhecer e utilizar o hardware da máquina e permitir a utilização dos diferentes programas que serão utilizados. Na verdade, o usuário não usa o sistema operacional, mas sim os programas instalados. Exemplo: Windows (Microsoft), Macintosh (Apple) e Linux Educacional.

Software Livre - "Software Livre" não significa "não-comercial". "Software livre" se refere à liberdade dos usuários executarem, copiarem, distribuírem, estudarem, modificarem e aperfeiçoarem o software. Mais precisamente, ele se refere a quatro liberdades, para os usuários do software:

- A liberdade de executar o programa, para qualquer propósito (liberdade no. 0);

- A liberdade de estudar como o programa funciona, e adaptá-lo para as suas necessidades (liberdade no. 1). Acesso ao código-fonte é um pré-requisito para esta liberdade;
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade no. 2);
- A liberdade de aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade no. 3). Acesso ao código-fonte é um pré-requisito para esta liberdade.

Software gratuito - também conhecido como *freeware* é todo programa de computador cuja utilização não exige o pagamento de licenças de uso ou royalties. • Portanto um software pode ser gratuito sem ser livre, ou seja, não ter seu código aberto. Pode ser acompanhado de uma licença que restringe o tipo de uso como, por exemplo, uso para fins não lucrativos, não comerciais, uso acadêmico, entre outros.

**Noções de lógica:** A lógica (do grego clássico logos, que significa palavra, pensamento, ideia, argumento, relato, razão lógica ou princípio lógico), é uma ciência de índole matemática e fortemente ligada à Filosofia. Já temos alguns noções de lógica com base em nossas vivências, por exemplo:

**Se todo mamífero é um animal e todo cavalo é um mamífero. Portanto, todo cavalo é um animal.**

Usamos a lógica no nosso dia a dia sem perceber, por exemplo:

**Uma gaveta está fechada, tem caneta dentro da gaveta. Precisamos primeiro abrir a gaveta para depois pegar a caneta.**

A lógica na programação é importante porque é ela quem nos dá as ferramentas necessárias para executar o processo mais básico no desenvolvimento de alguma aplicação: a criação de seu **algoritmo**. Antes de criar um software do zero, ou de resolver um problema de um já existente, é necessário descascá-lo até chegarmos ao seu núcleo. Em outras palavras, precisamos compreendê-lo completamente, desde suas funções a seus objetivos finais — ou seja, pesquisar, rascunhar, dominá-lo na íntegra.

**Algoritmos:** É a descrição de uma sequência de ações para realizar alguma tarefa. Em geral, estaremos interessados em algoritmos computacionais, que descrevem uma sequência

de ações que podem ser traduzidos para alguma linguagem de programação.

Uma maneira para calcular o MDC (Máximo Divisor Comum) de dois números inteiros positivos é através do Algoritmo de Euclides apresentado no quadro seguinte:

**Passo 1:** Adote  $x = m$  e  $y = n$ ;  
**Passo 2:** Adote  $r = (\text{resto de } x \text{ dividido por } y)$ ;  
**Passo 3:** Adote novos valores  $x = y$  e  $y = r$ ;  
**Passo 4:** Se  $r$  é diferente de 0, volte ao passo 2; senão pare com a resposta  $x$ .

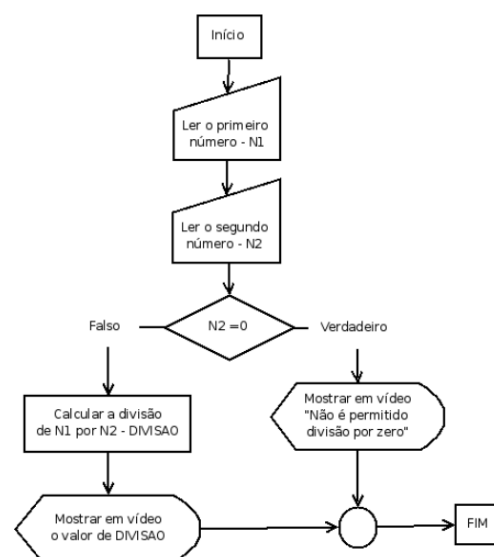
Algoritmo de Euclides

### Características que definem um algoritmo:

- Partir de um ponto inicial e chegar a um ponto final;
- Não ser ambíguo (ter dupla interpretação);
- Poder receber dados externos e ser capaz de retornar resultados aos mesmos;
- Ter todas suas etapas alcançáveis em algum momento do programa.

### Algoritmo para o final de semana

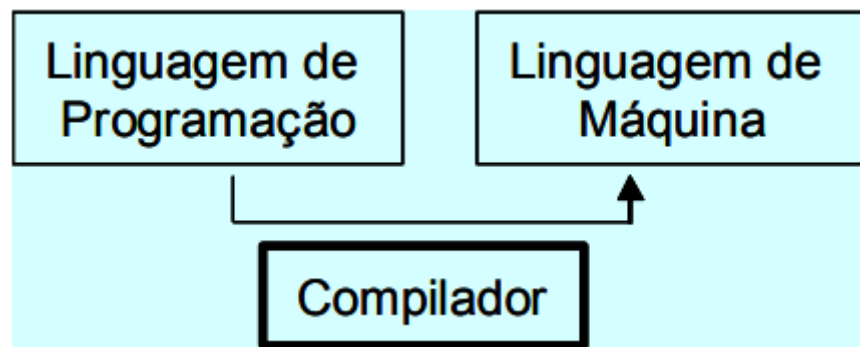
- 1 – Ver a previsão do tempo;
- 2 – Se fizer sol  
    ir a praia;  
    Senão  
    Estudar;
- 3 – Almoçar;
- 4 – Ver televisão;
- 5 – Dormir.



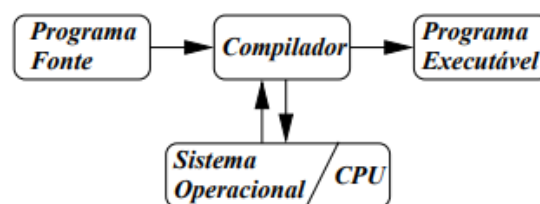
**Linguagens de Programação:** Vimos que podemos representar um algoritmo usando uma forma textual ou gráfica para podermos compreender melhor, mas uma máquina não consegue interpretar um texto ou uma imagem sem auxílio de um software.

Para suprir esses problemas, foram desenvolvidas linguagens em que a programação é feita

através de uma notação matemática e de algumas palavras reservadas.



**Compilador:** Tradutor de programas escritos em uma linguagem de programação para programas em linguagem de máquina. Uma vez que o programa foi convertido para código de máquina, este pode ser executado independente do compilador e do programa original.



Como exemplo de linguagens de programação podemos citar o Fortran, C, C++, Pascal, Cobol, MATLAB, Python, etc.

**O que é Python?** é uma linguagem de programação extremamente simples e versátil. Isso porque sua sintaxe é moderna e objetiva, ou seja, ela foi elaborada para que as pessoas desenvolvedoras escrevam instruções com menos linhas de código. Por isso, é fácil aprender Python. Ela é uma linguagem eficiente tanto para a construção de pequenos sistemas quanto para o desenvolvimento de aplicações complexas.

Python é uma linguagem multiplataforma, pois tem versões para diversos sistemas operacionais, entre eles: Windows, Linux e macOS. Além disso, ela é mantida pela Python Software Foundation (PSF), uma organização sem fins lucrativos.

Conhecendo mais sobre Python:

O Python é uma linguagem interpretada, pois ela não gera arquivos executáveis como acontece na linguagem C, por exemplo. Em vez disso, ela contém um interpretador, que é responsável por traduzir o código fonte em linguagem de máquina e, assim, executar o programa.

Na prática, o interpretador transforma o código fonte para o formato bytecode, que corresponde à linguagem de máquina ou código binário, e envia para um ambiente chamado PVM — Python Virtual Machine —, que é uma máquina virtual Python que contém o ambiente capaz de executar o programa

***Como surgiu a linguagem Python?*** O desenvolvimento da linguagem Python começou no final da década de 1980 por Guido Van Rossum, enquanto ele trabalhava no Centrum Wiskunde & Informatica — CWI — em Amsterdã. Seu objetivo era criar uma linguagem de script mais eficiente, capaz de acessar recursos do sistema operacional Amoeba.

Em 1994, surgiu a primeira versão da linguagem Python. O nome escolhido não tem nenhuma relação com a espécie de cobras de mesmo nome, apesar de seu logotipo ser o de duas cobras entrelaçadas. Em vez disso, a escolha foi em função de um programa de televisão da época criado por um grupo de comediantes chamado Monty Python.

Desde então, a linguagem Python recebeu diversas atualizações. Ela tem o código aberto e é disponibilizada sob a licença PSF License Agreement, que é semelhante à General Public License (GPL). Atualmente, está na versão 3.9.5 e é possível fazer o Python download gratuitamente a partir da página oficial da linguagem.

### **1. É fácil de aprender**

A linguagem Python tem a sintaxe simples e o mais objetiva possível. Um exemplo é que não precisamos utilizar o sinal de ponto e vírgula ao final de cada declaração e as chaves para delimitar os comandos de estruturas de repetição e condicionais, como acontece em outras linguagens de programação.

### **2. É portátil e multiplataforma**

A linguagem Python pode ser instalada em diferentes sistemas operacionais, como Windows, Linux



e macOS. Além disso, existem distribuições portáteis, que podem ser utilizadas em pen drives, o que é ideal para facilitar a vida das pessoas programadoras na fase de desenvolvimento da aplicação.

### 3. É open source e gratuito

A linguagem Python é de código aberto, o que significa que é possível baixar seu código fonte escrito em linguagem C e realizar alterações. Também não é preciso pagar por sua licença de uso. Além disso, ele contém inúmeras bibliotecas e frameworks gratuitos que podem ser utilizados para obter recursos adicionais.

### 4. Oferece múltiplas possibilidades de desenvolvimento

É possível desenvolver diferentes tipos de aplicações com a linguagem Python, tanto sistemas comerciais para desktop, quanto softwares mais complexos de Machine Learning e Inteligência Artificial. As bibliotecas e frameworks disponíveis para a linguagem ampliam ainda mais a possibilidade de desenvolvimento. Um exemplo é o framework Django, que é utilizado para o desenvolvimento de aplicações web. Por toda essa abrangência, ela foi escolhida em primeiro lugar como a linguagem de programação mais desejada pelas pessoas programadoras na pesquisa Stack Overflow de 2020.

### 5. É uma linguagem “curinga”

A linguagem Python é extremamente poderosa e oferece uma ampla possibilidade de desenvolvimento de sistemas. Como consequência, amplia as oportunidades de trabalho às pessoas programadoras, que podem escolher entre as diferentes possibilidades de desenvolvimento como a de aplicações para web. Além disso, sua sintaxe é simples e intuitiva, o que facilita o aprendizado.

## *Comandos básicos*

**Entrada:** Entrada de dados é a forma como enviamos informações para dentro de aplicações escritas em Python. A forma mais primitiva de se fazer isto é através do teclado. Isto é realizado a partir do comando **input**.

**Saída:** Todas as vezes em que desejamos imprimir uma mensagem na tela, utilizaremos o comando **print ('...')**.

## *Quais os tipos de dados Python?*

**Inteiros (int):** O tipo inteiro representa os números não decimais positivos ou negativos e de tamanho ilimitado. Para uma variável ser considerada do tipo inteiro, basta atribuir um valor com essas características a ela ou indicar o tipo antes de atribuir o valor.

```
a = 1
b:int = 1
print(type(a)) #Resultado: <class 'int'>
print(type(b)) #Resultado: <class 'int'>
```

**Decimais (float):** O tipo float, que é chamado de número de ponto flutuante, também faz parte dos tipos numéricos e é representado pelos números que contêm uma ou mais casas decimais.

```
i = 199.90
print(i) #Resultado: 199.9
print(type(i)) #Resultado: <class 'float'>
```

**Booleana (bool):** O tipo bool é representado por uma entre duas possibilidades de valores, são eles: True ou False, que indica verdadeiro ou falso. Além de declarados em variáveis, os valores do tipo bool são utilizados para comparar e avaliar expressões.

```
i = 10
resultado = (i == 10)
print(resultado) #Resultado: True

print(i == 10) #Resultado: True
print(i >= 10) #Resultado: True
print(i < 10) #Resultado: False
```

**Palavras/frases (string):** Em Python, o tipo string funciona como um array de caracteres. Por isso, é possível acessar cada caractere por meio de seu índice correspondente. Para declarar uma string, basta informar o conteúdo delimitado por aspas ou aspas dupla.

```
nome = 'João da Silva'
print(nome) #Resultado: João da Silva
print(type(nome)) #Resultado: <class 'str'>
```

## Estruturas condicionais em Python: if, elif e else

Toda linguagem de programação precisa ter estruturas condicionais, que são comandos utilizados para verificar se uma expressão é verdadeira ou falsa e executar determinados blocos de código em cada situação específica. Confira como a estrutura **if**, **elif** e **else** funcionam em Python.

**if/else:** O comando Python **if** é utilizado para verificar a condição de uma expressão e executar ou não um determinado bloco de código. A instrução **if** pode ser executada sozinha ou em conjunto com a instrução **else**, que possibilita a execução de outro conjunto de códigos caso a condição avaliada seja falsa.

```
nome = 'João'
if nome == 'João':
    print(True)
else:
    print(False)
#Resultado: True
```

**Elif:** Existem situações em que pode ser necessário avaliar outra condição caso a primeira expressão seja falsa. Nesse cenário, podemos utilizar a instrução **elif**, que avalia se outra condição é verdadeira. Veja a sintaxe abaixo:

```
numero = 20
if numero < 10:
    print(True)
elif numero == 20:
    print("O número é igual a 20")
else:
    print(False)
#Resultado: O número é igual a 20
```

## Programas em Python:

*Calcule a média aritmética de duas notas quaisquer fornecidas para um aluno.*

```
#-----
def media():
    n1 = (float)(input('Digite a nota 1: '))
    n2 = (float)(input('Digite a nota 2: '))

    md = (n1+n2)/2.0

    print( "media: %f" %md )

#-----


media()
```

Faça um programa que leia os coeficientes de uma equação do primeiro grau do tipo  $y = ax + b$  e calcule a sua raiz. Imprimir o resultado na tela (questão da lista).

```
#-----  
  
def raiz():  
  
    a = (float)(input('Coeficiente A: '))  
    b = (float)(input('Coeficiente B: '))  
  
    x = -b/a  
  
    print( "raiz: %f" %x )  
  
#-----  
  
raiz()
```

#### Tarefinha para os feras:

O cálculo do ponto de interseção entre duas retas pode ser obtido igualando-se às suas equações. Baseado nessa informação, faça um programa que leia os coeficientes das equações das retas e calcule o ponto de interseção. As retas são definidas pelas equações  $y = Ax+B$  e  $y = Cx + D$ . Imprimir o resultado.


$$Ax+B = Cx + D; (A-C)x=(D-B)$$
$$x=(D-B)/(A-C)$$
$$y=Ax+B$$

# *Links úteis*

## **Instalação do Python:**

**Windows:** <https://python.org.br/instalacao-windows/>

**MacOS:** <https://python.org.br/instalacao-mac/>

**Linux:** <https://python.org.br/instalacao-linux/>

**Documentação:** <https://docs.python.org/pt-br/3/tutorial/>

**Pycharm:** <https://www.jetbrains.com/pt-br/pycharm/download/>

**Notepad++:** <https://notepad-plus-plus.org/downloads>