

ALGORITMO vs PROGRAMA

- Um **algoritmo** é um conjunto finito de regras bem determinadas para a resolução de um problema, através de um número finito de operações;
- Por sua vez, um **programa** é um conjunto de instruções destinadas a ser processadas num sistema informático.

Algoritmo

- Conjunto de regras e operações bem definidas e ordenadas, destinadas à resolução de um problema com um número finito de etapas;
- Pseudo-código ou fluxograma
- Usada a linguagem nativa (de quem o escreve)

Programa

- Sequência completa de instruções a serem executadas por um computador numa determinada linguagem. (De acordo com um algoritmo).

EXEMPLO DE ALGORITMO: Algoritmo que verifica números positivos

Variáveis:

inteiro: número

início

escrever (“Digite um número”)

ler (número)

se (número > 0)

escrever (“É positivo”)

fimSe

fim



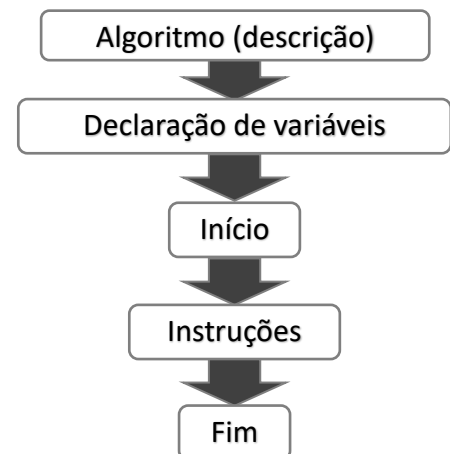
Um algoritmo pode ser dividido em duas partes principais: a declaração de variáveis, primeiro, e o algoritmo propriamente dito, sempre definido entre as instruções “INICIO” e “FIM”.



Quando queremos comentar uma linha, ou seja, que uma linha seja apenas explicativa e não uma instrução do algoritmo, devemos colocar # no seu início.

ESTRUTURA DE UM ALGORITMO

- (1) Descrição do algoritmo
- (2) Declaração das variáveis necessárias
- (3) Processamento dos dados
- (4) Escrita dos resultados



- Tipos de operações básicas e fundamentais de qualquer sistema informático:
 - a) **Operações de input** ou entrada de dados;
 - b) **Operações de processamento** interno;
 - c) **Operações de output** ou saída de dados;
- Para a **operação de input** de dados devemos usar a palavra **Ler** ou **Obter**.
- Para a **operação de output** de dados devemos usar a palavra **Escrever** ou **Apresentar**.

Pseudocódigo

- Os algoritmos podem ser escritos em pseudocódigo (linguagem natural) permitindo a sua leitura simples e direta sem grande necessidade de conhecimentos de programação.
- Cada algoritmo deve conter duas áreas:

Estrutura de dados –onde devem ser declaradas todas as variáveis e os seus valores iniciais, caso existam;

Algoritmo – onde são apresentadas todas as ações que serão executadas linha-a-linha.

EXEMPLO:

Pretende-se criar um algoritmo em pseudo-código que permita a um utilizador inserir dois números inteiros. Após a leitura dos dois números o algoritmo deve fazer a soma dos dois números e apresentar o resultado.

ESTRUTURA DE DADOS:

INTEIRO numero1, numero2, soma;

ALGORITMO:

```

INÍCIO
  ESCREVER("Insira o 1º número");
  LER(numero1);
  ESCREVER("Insira o 2º número");
  LER(numero2);
  soma = numero1 + numero2;
  ESCREVER("A soma é:" + soma);
FIM
  
```

OPERADORES

Operador Atribuição

=

Operadores Aritméticos

+	-	*	/	%
Adição	Subtração	Multiplicação	Divisão inteira	Resto da divisão inteira

Operadores Relacionais

<	>	<=	>=	==	!=
Menor	Maior	Menor ou igual	Maior ou igual	Igual	Diferente

Operadores Lógicos

and	or	not
Conjunção	Disjunção	Negação

DADOS

► Um algoritmo manipula dados:

•Tipo de Dados

Cada dado é de um determinado tipo e têm um conjunto de operações associadas, por exemplo

- Números inteiros
- Números reais
- Caracteres,
- Cadeias de caracteres, ...

•Estrutura de Dados

Modo como os dados estão organizados e como são acedidos e alterados, Por exemplo:

- Variáveis simples,
- Arrays mono e multidimensionais,
- listas, filas, árvores, grafos, dicionários,...

Tipo	Descrição	Exemplo
Booleano	Pode assumir o valor True ou o valor False	is_active = True
Numérico inteiro	Aceita qualquer valor numérico sem parte decimal	quantity = 10
Numérico decimal	Aceita qualquer valor numérico com parte decimal	price = 10.5
Carateres	Aceita qualquer cadeia de carateres (letras, números, símbolos, ...)	text = 'Isto é um texto'

Programação em Python

– Leitura

- O comando abaixo permite a leitura, para a variável “nome”, do valor inserido, em tempo de execução, pelo utilizador.

```
nome = input("Insira um nome")
```

- Variáveis

```
nome = "Meu nome"  
numero = 123  
outro_numero = 12.3  
verdadeiro_falso = True
```

- Ao contrário de muitas linguagens de programação o Python não necessita que seja colocado o tipo de dados antes da sua declaração.
- Isto não significa que não se trata de uma linguagem de “tipagem” forte!
- O Python identifica o tipo de dados pelo seu conteúdo.

➔ Exercícios de exploração em ficheiro [ALG-Sessao3-Exemplos_Exploração.py](#)