

# Fundamentos de Programação de Computadores

## Introdução aos Algoritmos

Docente: Fátima Leal

**DCT** DEPARTAMENTO CIÊNCIA  
E TECNOLOGIA

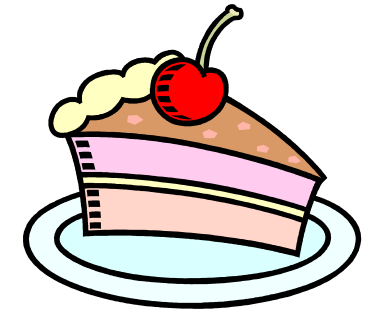
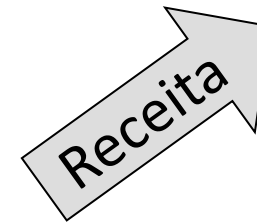
# Introdução aos Algoritmos

- **Algoritmo** por definição é uma **sequência** finita de **passos ou instruções**, ordenadas de forma lógica, que levam à execução de uma tarefa ou à solução de um problema.

- **Exemplo:**

- 250 g de farinha
- 150 g de margarina
- 5 ovos
- 2 colheres de fermento
- 200 gramas de açúcar

1. Misturar os ingredientes
2. Cozinhar o bolo



# Introdução aos Algoritmos

- **Refinamento** melhora a sequência do algoritmo de modo a obter a solução ou o resultado pretendido.
- **Exemplo:**
- Misturar os ingredientes:
  - Juntar a margarina e a farinha e bater até obter um creme
  - Juntar os ovos e mexer
  - Juntar o fermento
- Cozinhar o bolo
  - Aquecer o forno a  $180^{\circ}$
  - Cozer o bolo durante 45 min



# Introdução aos Algoritmos

- Diariamente **criamos algoritmos** nas nossas rotinas:
  - Resolver uma operação matemática
  - Tomar medicamentos segundo as orientações médicas
  - Cozinhar seguindo as receitas
- Os algoritmos são capazes de realizar tarefas como:
  - Ler e escrever dados
  - Avaliar expressões algébricas, relacionais e lógicas
  - Tomar decisões com base nos resultados das expressões avaliadas
  - Repetir um conjunto de ações de acordo com uma condição

# Introdução aos Algoritmos

- Os **algoritmos** são mais importantes que as **linguagens de programação** ou que o **computador**. A linguagem de programação é um meio para expressar um algoritmo e um computador é um meio para executá-lo.
- Os algoritmos devem ser:
  - **Precisos** e indicar a ordem de realização de cada passo
  - **Eficazes** e contemplar diferentes problemas
  - **Eficiente** utilizando o menor número de passos possível
  - **Finitos**, ou seja, deve terminar em algum momento incorporando um número finito de passos.
- Um **algoritmo** é constituído por três partes:
  - **Entrada**
  - **Processamento**
  - **Saída**

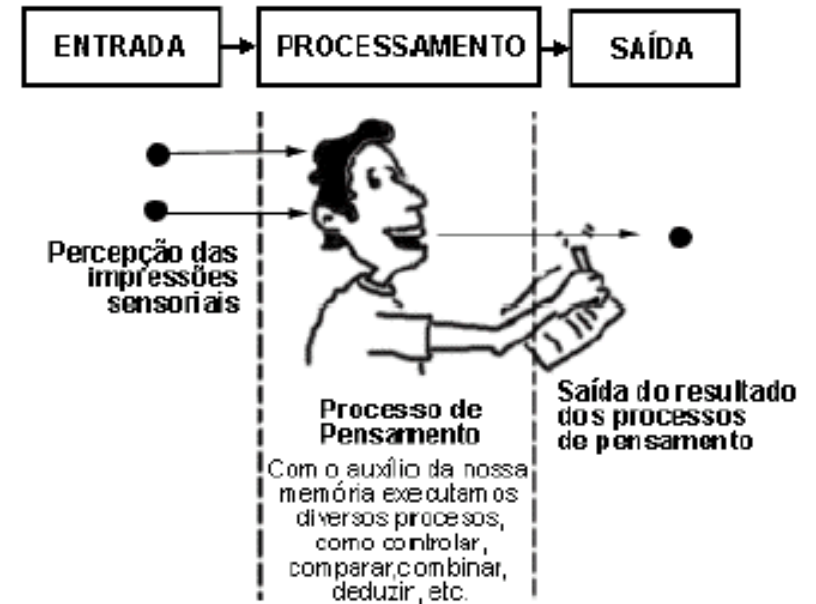
# Componentes de um Algoritmo

## ■ Problema

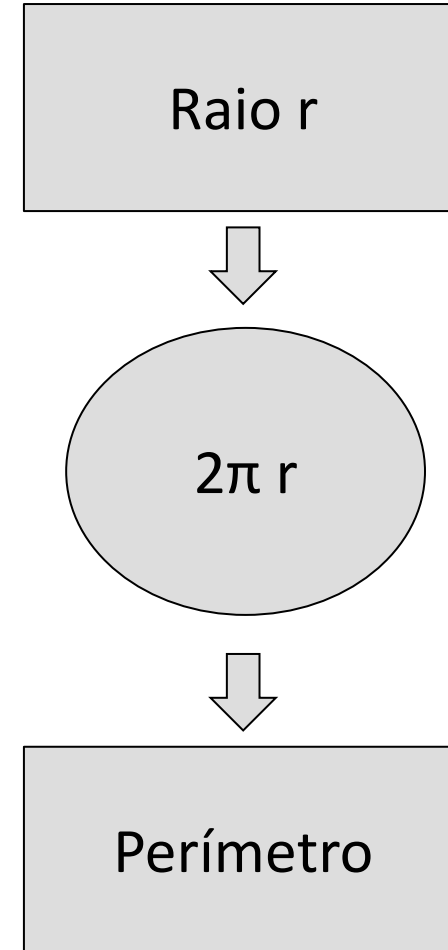
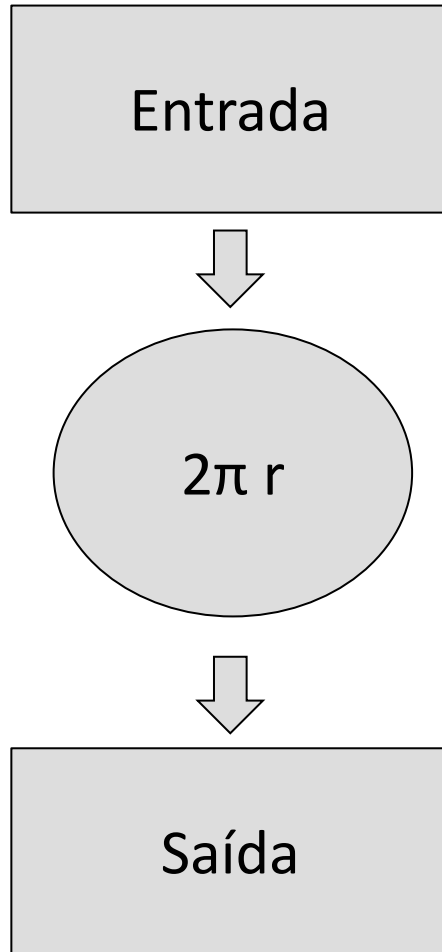
- Conjunto de possíveis **entradas**
- A **saída** pretendida
- Conjunto de **operações** válidas

## ■ Solução algorítmica

- **Conjunto ordenado de operações** que transformam o conjunto de entradas na saída desejada

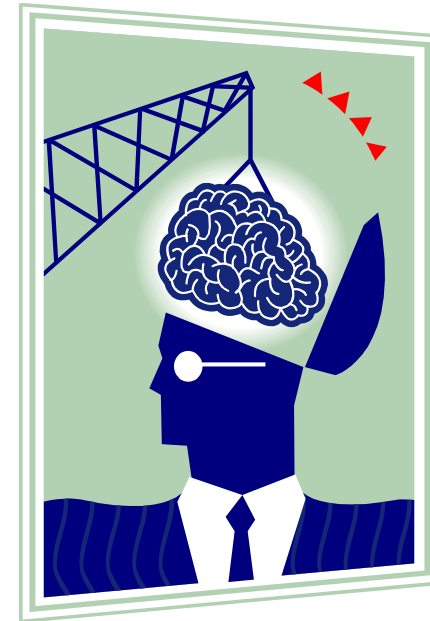


# Introdução aos Algoritmos



# Introdução aos Algoritmos

- O **algoritmo não é a solução** de um problema mas o **caminho** para a solução.
- **Não se aprende:**
  - a **copiar** algoritmos
  - ler **algoritmos** terminados
  - a **decorar** algoritmos
- **Aprende-se:**
  - **Construindo**
  - **Testando**
  - **Praticando** o pensamento





# Algoritmia: Conceitos

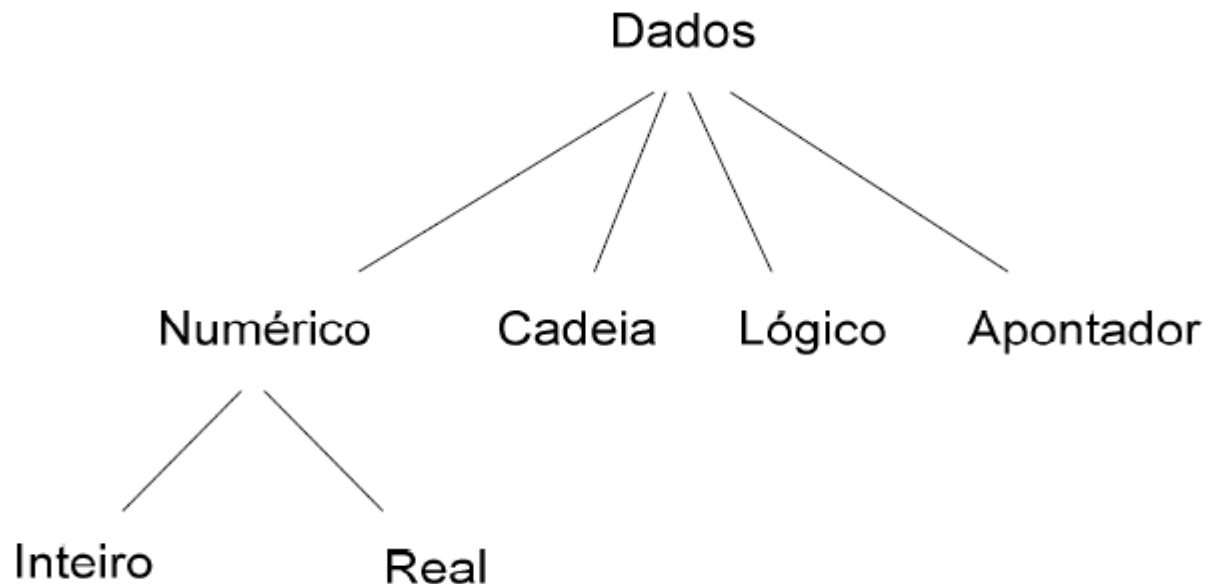
- **Dados** correspondem à porção da informação a ser processada pelo computador
- **Variável** representa uma entidade na qual é possível guardar temporariamente valores processados durante a execução de um programa.

	Endereço	Conteúdo
a	1100	40
z	1101	35
c	1102	5930
g	1103	68

- Neste exemplo, o valor da variável z é 35 que está armazenado no endereço 1101.

# Algoritmia: Tipos de Dados

- Os **dados** não são todos do mesmo **tipo** e para os programas torna-se necessário diferenciar os vários **tipos de dados**.
- Tipo de **dados primitivos** ou **básicos**:



# Algoritmia: Tipos de Dados

- **Inteiro:** valores inteiros, e.g., 13.
- **Real:** valores reais, e.g., 5.14356
- **Cadeia de caracteres:** valores alfanuméricos, e.g., “Fatima Leal”
- **Lógico:** valores booleanos, e.g., FALSO.
- **Apontador:** endereços de memória
- Cada **variável** irá guardar um tipo de dados. Deverá ter um nome de preferência indicativo.
- O valor da variável fica alterado depois de uma **atribuição**.
- **Exemplos:**

Atribuição

`nomeProfessor <- “Fatima Leal”`

`disciplina <- “Fundamentos Programacao Computadores”`

`classificacao <- 17`

# Algoritmia: Tipos de Dados

- Nomes de variáveis:

- NUM
- N\_ALUNO
- NOMEALUNO
- NOME\_ALUNO
- X1
- Y2



- Nomes Inválidos

- 7NUM
- Y-Z
- U\*Z
- DUAS PALAVRAS
- DUAS-PAL



# Algoritmia: Operadores

- Na construção de algoritmos é inevitável o recurso a operadores:
  - **Aritméticos**
  - **Relacionais**
  - **Lógicos**
- **Operadores Aritméticos**

Operador	Descrição	a	b	x
+	Adição	4	5	$4 + 5 = 9$
-	Subtração	4	5	$4 - 5 = -1$
*	Multiplicação	4	5	$4 * 5 = 20$
/	Divisão	18	3	$18 / 3 = 6$
%	Resto de divisão	20	3	$20 \% 3 = 2$

# Algoritmia: Operadores

## ■ Operadores Relacionais

Operador	Descrição	a	b	x
==	Verifica uma igualdade	4	5	4 == 5 R:
!=	Verifica se é diferente	4	5	4 != 5 R:
>	Verifica se é Maior	4	5	4 > 5 R:
<	Verifica se é Menor	4	5	4 < 5 R:
>=	Verifica se é Maior ou igual	4	4	4 >= 4 R:
<=	Verifica se é Menor ou igual	3	4	3 <= 4 R:

# Algoritmia: Operadores

## ■ Operadores Lógicos

Operador	Descrição
&&	E / and
	Ou / or
!	Não / not

### Exemplo:

```
Se (x > y && y > 0)
    entao y <- x +1
```

Neste exemplo encontramos:

**Variáveis**

**Operadores:** aritméticos, relacionais e lógicos

**Atribuição**

# Algoritmia: Operadores

- **Atribuição** é uma operação utilizada para **armazenar** um valor numa determinada **variável**
- Em **algoritmia** representa-se por uma **seta**:
  - $x \leftarrow 23$  a variável x vai ficar com o valor 23
  - $x \leftarrow y$  a variável x vai ficar com o conteúdo de y
  - $x \leftarrow x + 1$  a variável x é incrementada de uma unidade
- Na **maioria** das **linguagens de programação** atribuição é realizada pelo **=**.



# Algoritmia: Tipos de Dados

- **Tipos de dados não primitivos** são definidos através de **conjuntos ou estruturas** de dados.
- **Vetores** permitem representar um **conjunto de valores** de diferentes tipos (inteiros, reais, caracteres). Têm um **nome, um índice e uma dimensão**. É apresentado através de um nome e de um índice entre parêntesis retos.
- Exemplo...

# Algoritmia: Tipos de Dados

	I = 1 .. 8				N = 8			
	1	2	3	4	5	6	7	8
VNUM [I]	34	121	7	78	0	90	3	15

VNUM [2] = 121

VNUM [7] = 3

VNUM [12] - indefinido

I <- 5

VNUM [I] = 0

I <- 1

VNUM [I] = 32

# Algoritmia: Tipos de Dados

- **Matrizes** pode ser interpretada como um **vetor bidimensional** (com dimensão 2). Uma matriz é representada através de um **nome e dois índices** que permitem indexar e aceder os elementos que constam na matriz.
- Exemplo ...

# Algoritmia: Tipos de Dados

Índice representativo das colunas da matriz → **J** M = 6

Índice representativo das linhas da matriz → **I**

**MAT [ I , J ]**

N = 7

		1	2	3	4	5	6
1	55	12	72	8	15	99	
2	121	67	17	78	12	123	
3	34	4	71	7	54	212	
4	56	12	12	7	56	33	
5	34	21	15	8	0	79	
6	76	32	78	78	56	7	
7	43	221	321	77	45	7	

MAT [2,3] = 17 I <- 3

MAT [7,1] = 43 J <- 5

MAT [4,12] - indefinido MAT [I,J] = 54

# Introdução aos Algoritmos

- Os algoritmos podem ser descritos em diferentes formalismos:
  - Linguagem natural
  - Linguagem gráfica - Fluxogramas
  - Pseudocódigo
- Construção de algoritmos implica conhecer um conjunto de instruções:
  - **Leitura e escrita** - “ponte do exterior” para dentro do algoritmo e do algoritmo para o “exterior”, respetivamente.
  - **Decisão** permite alterar o fluxo de controlo (ordem pela qual as instruções de um algoritmo são executadas).
  - **Repetição** mecanismo de repetição de instruções que também permite alterar o fluxo de controlo do algoritmo.
  - **Atribuição** é uma operação elementar que permite alterar o valor de uma variável.

# Algoritmia: Conceitos

- Resumindo...
- Os algoritmos contêm:
  - **Instruções**
  - **Variáveis**
  - **Operadores**
  - **Atribuições**
  - **Dados**

# Programa

Programa = Algoritmo + dados

# Pseudocódigo

- **Pseudocódigo** é uma notação **algorítmica** muito próxima da linguagem natural.
- Para traduzir o **pseudocódigo** para uma **linguagem de programação**, basta conhecer o vocabulário e regras sintáticas da linguagem.



# Pseudocódigo

- **Algoritmo** é a palavra que indica o início da definição de um algoritmo em forma de pseudocódigo.
- **<declaração\_de\_variáveis>** é uma forma opcional onde são declaradas as variáveis globais usadas no algoritmo
- **Início e Fim** são respectivamente as palavras que delimitam o início e o fim do conjunto de instruções do corpo do algoritmo.

# Pseudocódigo: Variáveis

- **Declaração de variáveis:** todas as variáveis que serão utilizadas pelo algoritmo devem ser declaradas entre as palavras reservadas **Variaveis** (ou Var) e **Inicio**

```
Algoritmo Exemplo
Variaveis
    a: Inteiro;
    b: Real;
Inicio
    <bloco de comandos>
Fim
```

- **Atribuição:** <variável>  $\leftarrow$  <valor>

- Exemplos:           salario  $\leftarrow$  1000;  
  
                      nome  $\leftarrow$  fatima ;

# Pseudocódigo: Comando de Entrada e Saída

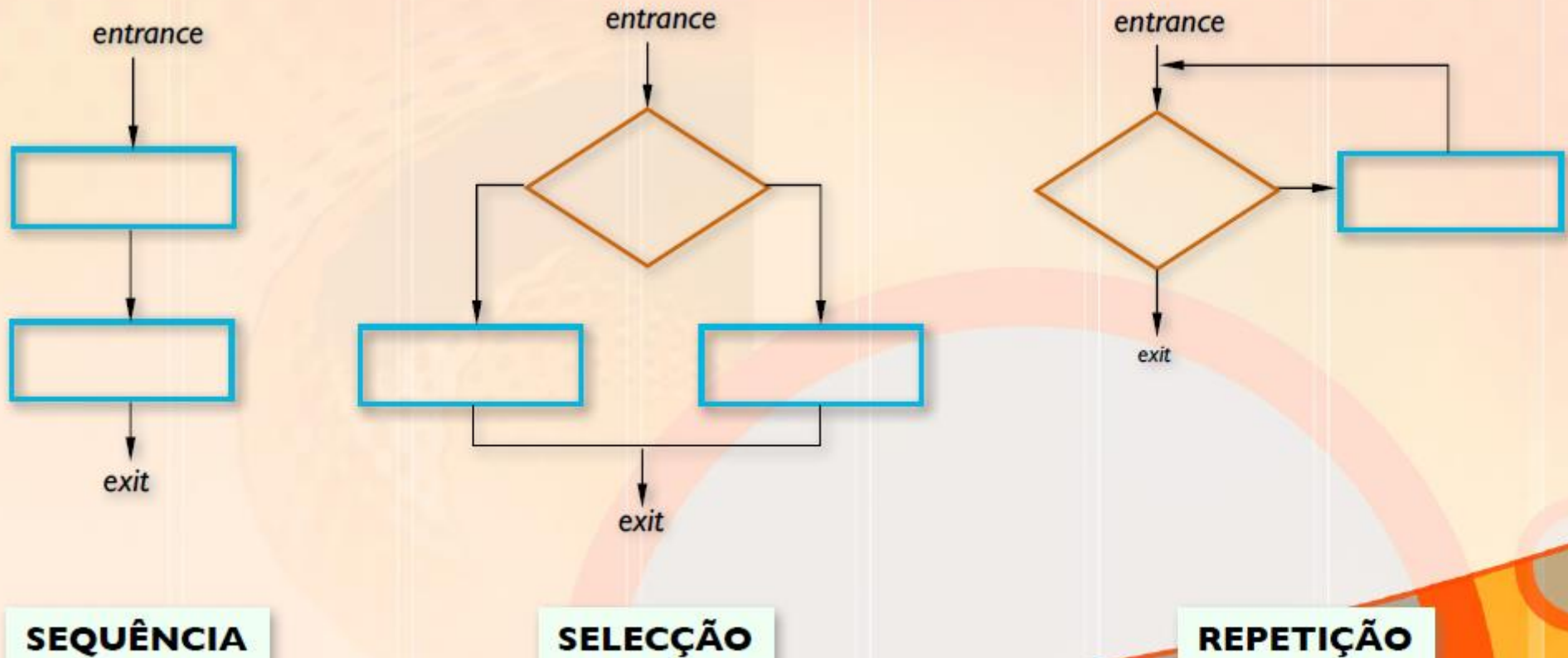
- **Entrada:** sempre que precisamos que o usuário introduza algum dado para ser processado, teremos uma entrada de dados. O comando usado para receber essa entrada de dados em pseudocódigo é o comando **Ler**.
  - **Exemplo:**
    - **Ler** (nome)
    - **Ler** (num1, num2)
- **Saída:** usado para exibir ou retornar qualquer valor ou mensagem ao usuário. O comando utilizado para produzir essa saída de dados em pseudocódigo é o comando **Escrever**.
  - **Exemplo**
    - **Escrever** (“Olá Mundo”)

# Fluxogramas

- **Fluxogramas** são representações **gráficas** de algoritmos
- Utilizam **formas geométricas** para representar o **início**, o **fim** e as **instruções** a serem executadas. As setas indicam o fluxo das ações



# Estruturas de Programação



# Estruturas de Programação

- **Instruções de Sequência** - as instruções de sequência são instruções atômicas (simples) permitem a leitura/escrita de dados, bem como o cálculo e atribuição de valores;
- **Instruções de Decisão** - as instruções de decisão, ou seleção, permitem a seleção em alternância de um ou outro conjunto de ações após a avaliação lógica de uma condição;
- **Instruções de Repetição** - as instruções de repetição, ou ciclos, permitem a execução, de forma repetitiva, de um conjunto de instruções. Esta execução depende do valor lógico de uma condição que é testada em cada iteração para decidir se a execução do ciclo continua ou termina.

# Representação de algoritmos

## Pseudocódigo

INÍCIO ou FIM

LER()

ESCREVER()

SE...ENTÃO...SENÃO

PARA...ATÉ...FAZER

ENQUANTO...FAZER

FAZER...ENQUANTO

PROCEDIMENTO/FUNÇÃO

## Fluxograma



Início ou fim



Entrada de dados



Saída de resultados



Processamento



Tomada de decisão



módulo

# Conceitos fundamentais

## Estrutura do algoritmo

- (1) Declaração das variáveis necessárias
- (2) Leitura dos dados
- (3) Processamento ...
- (4) Escrita dos resultados

INICIO

ED: (1)

variavel1, variavel2 INTEIRO

variavel3, variavel4 REAL

LER (variavel1, variavel2) (2)

variavel3  $\leftarrow$  variavel1 + variavel2 (3)

ESCREVER (variavel3) (4)

FIM



# Exemplo 1

- Elabora um algoritmo que receba 2 números inteiros do utilizador e apresenta a sua soma.

## Pseudocódigo

INICIO

ED: numero1, numero2, soma INTEIRO

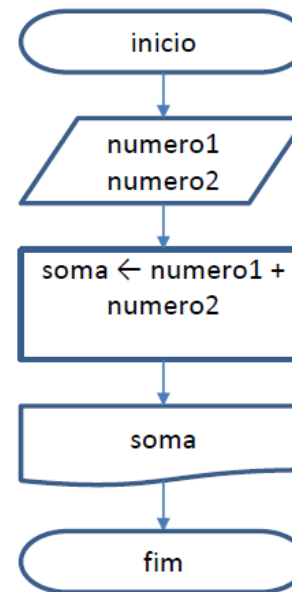
LER (numero1, numero2)

soma  $\leftarrow$  numero1 + numero2

ESCREVER (soma)

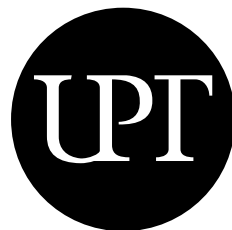
FIM

## Fluxograma



# Exemplo 2

- Dois atletas fizeram uma corrida solidária. Por cada quilómetro era doado 1,5€ para uma instituição. Elabora um algoritmo que calcule o total de dinheiro angariado.



UNIVERSIDADE  
PORTUCALENSE

Do conhecimento à prática.