



Algoritmia

P.PORTO

isep Instituto Superior de
Engenharia do Porto

 DEPARTAMENTO DE ENGENHARIA
INFORMÁTICA
Instituto Superior de Engenharia do Porto

Programação Imperativa

- Um programa é uma sequência de instruções que especifica como realizar uma computação.
- A computação pode ser algo matemático como resolver um sistema de equações ou processamento simbólico como procurar e substituir texto num documento ou compilar um programa ou ...
- **Os programas têm que ser descritos em linguagens de programação.**

Programação Imperativa

- Programar, em certas linguagens não é um processo simples e imediato. Por isso, vamos iniciar a programação usando uma pseudolinguagem de programação (**pseudocódigo**) muito simples, para que nesta fase inicial, possamos estar inteiramente concentrados na conceção da lógica do programa.
- Só depois de concebido o algoritmo, se descreve numa linguagem de programação.

Objetivo

- Um dos objetivos é iniciar os estudantes na produção de software.
- Como primeiro módulo nesta área de conhecimento, começa-se por **desenvolver competências de raciocínio lógico através do desenvolvimento de algoritmos e estruturação de dados.**
- Mas **produzir software não é só programar ...**

O que é programar?

Programar é

Conceber algoritmos

e

Descrevê-los numa dada linguagem.

Algoritmo

- Um algoritmo é um conjunto finito e bem definido (não ambíguo) de instruções que descreve os passos lógicos para a realização de uma tarefa.
- Um algoritmo correto é aquele que perante uma entrada válida deve produzir uma saída única e correta.
- Um algoritmo deve ser eficaz na resolução do problema proposto e eficiente de modo a resolver o problema com o melhor desempenho.

Um algoritmo manipula dados

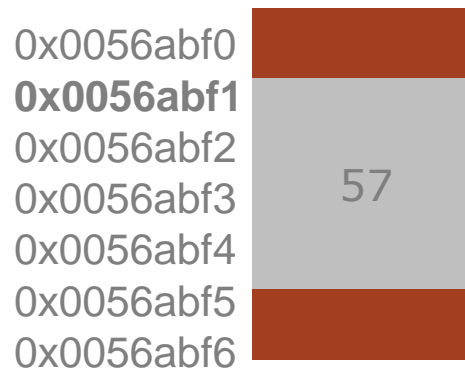
- **Estrutura de Dados** define o modo como os dados estão organizados e como são acedidos e alterados.

Exemplos:

- variáveis simples
- arrays mono e multi-dimensionais
- listas, filas, árvores, grafos, ...
- ficheiros (estruturas de dados em memória secundária)

Noção de variável

- Variável é uma posição de memória caracterizada por:
 - **Nome** (deve ser sugestivo e curto)
 - **Tipo de dados**
 - **Endereço**
 - **Valor ou conteúdo que guarda**



Noção de variável

■ Nomes de variaveis

- Devem:
 - **ser mnemónicos**
 - **ser curtos mas explícitos**
 - **começar por uma letra**
- Não podem:
 - **conter espaços nem caracteres como vírgulas ou pontos**
 - **ser iguais a palavras reservadas**

Noção de variável

- **Variável** é onde o programa guarda os dados
- **Variável tem associado um Tipo de Dados, que define:**
 - O conjunto de valores que a variável pode armazenar, e
 - O tipo de operações em que as variáveis podem ocorrer

Tipos de dados

- Um **tipo** é um conjunto de valores relacionados por um conjunto de operações
- **Tipos primitivos**
 - Números Inteiros (**INTEIRO**)
 - Números reais (**REAL**)
 - Carácter (**CARACTER**)
 - Cadeias de caracteres (**TEXTO**)
 - Booleano (**BOOLEANO**)
- **Tipos não primitivos (complexos)**
 - Tipos indexados mono e multidimensionais
 - referenciar e guardar valores de um tipo primitivo único.

Operadores

OPERADOR DE ATRIBUIÇÃO

←

atribuir valor a uma variável

Exemplo: valor ← 18*32

OPERADORES ATRITMÉTICOS

^	potenciação	prioridade 1
*	multiplicação	prioridade 2
/	divisão real	
DIV	divisão inteira	
MOD	resto da divisão inteira	
+	soma	prioridade 3
-	subtração	

Operadores

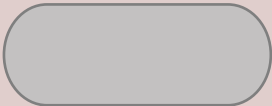
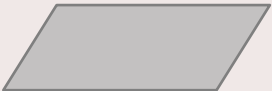


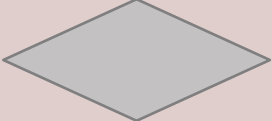

OPERADORES RELACIONAIS

<	menor
<=	menor ou igual
>	maior
>=	maior ou igual
=	igual
<>	diferente

OPERADORES LÓGICOS ou BOOLEANOS

E	conjunção
OU	disjunção
NAO	negação

Descrição de algoritmos

Pseudocódigo	Fluxograma
INICIO / FIM	
Entrada (Input) - LER()	
Saída (Output) - ESCREVER()	
Ações – atribuições / processamento	
Decisão	
Linhas de Fluxo	

Controlo de fluxo de um Algoritmo ou programa

- O controlo de fluxo de um programa refere-se à ordem pela qual são executadas as instruções.
- É possível escrever qualquer programa utilizando exclusivamente as três estruturas básicas de controlo de fluxo:
 - **Sequência** - permite a ordenação em série de instruções;
 - **Seleção/Decisão** - permite a seleção em alternância de um ou outro conjunto de ações por avaliação de uma condição;
 - **Repetição** - permite a execução condicional em circuito fechado (ciclo) de um dado grupo de instruções. A condição é testada em cada iteração para decidir se deve sair ou não do ciclo.

Estrutura de um algoritmo

Descrição da Estrutura de Dados **ED:** //variáveis e tipos de dados

Descrição do Processo

INICIO

...

FIM

Instruções de Entrada e Saída **LER()**
ESCREVER()

Instrução de atribuição **$a \leftarrow b + c$**

Instruções de transferência de controlo de fluxo

Sequência, Selecção/Decisão e Repetição/Iteração

Exemplo

■ Calcular a área de um quadrado

ED:

lado, area REAL

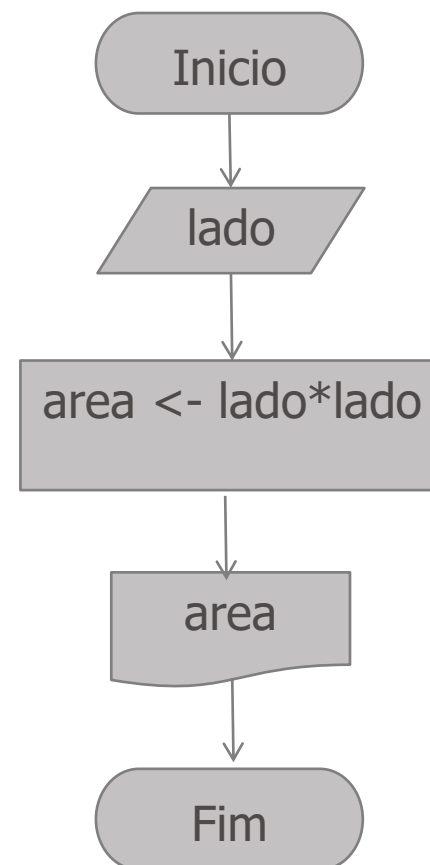
INICIO

LER (lado)

area \leftarrow lado * lado

ESCREVER ("Área do Quadrado=", area)

FIM



Exercícios – Estruturas sequenciais

- Calcular a percentagem de rapazes e raparigas existentes numa turma.
- Ler um valor em segundos e converter para horas, minutos e segundos.

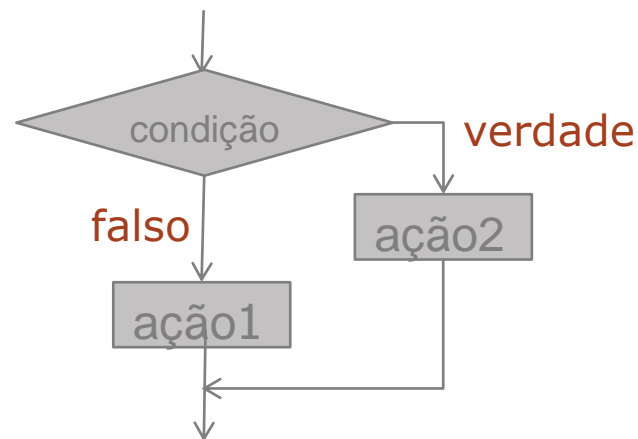
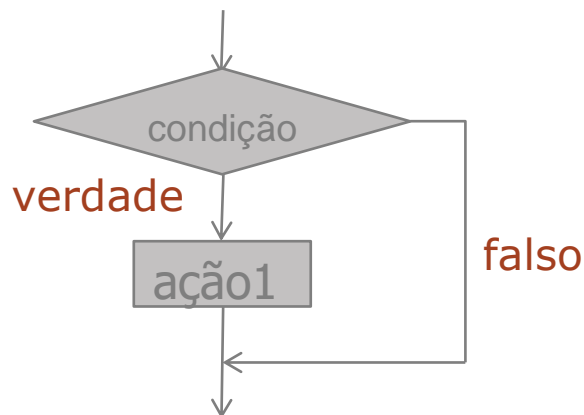
Estrutura de controlo - Decisão

Pseudocódigo

```
SE (condição) ENTAO
    ação1
FIMSE
```

```
SE (condição) ENTAO
    ação1
SENAO
    ação2
FIMSE
```

Fluxograma



Exemplo

■ Determinação da área de um retângulo

ED:

lado1, lado2, area REAL

INICIO

LER (lado1,lado2)

area <- lado1 * lado2

SE (lado1=lado2) ENTÃO

 ESCREVER ("Área Quadrado = ", area);

SENAO

 ESCREVER ("Área Retângulo = ", area);

FIMSE

FIM

Traçagem (valores de entrada 5 e 10)

Exemplo

- Descreva um algoritmo que, dada a sua idade e a idade do seu amigo, determine a relação de idades

ED:

minhaIdade, amigoIdade INTEIRO

INICIO

LER (minhaIdade, amigoIdade)

SE (minhaIdade = amigoIdade) ENTAO

 ESCREVER("São da mesma idade")

SENAO

 SE (minhaIdade > amigoIdade) ENTAO

 ESCREVER("O seu amigo é mais novo")

 SENAO

 ESCREVER("O seu amigo é mais velho")

FIMSE

FIM