



ALGORITMIA

Resolução De Problemas Em Informática

Docente: Ana Paula Afonso

2000-2001



Resolução de Problemas

- 1. Analisar o problema
 - Conhecer o bem o problema
 - Descrever o problema: subdividir, detalhar
- 2. Resolver o problema passo a passo, verificar se não há ambiguidade na solução apresentada, ou seja, escrever o algoritmo.
- 3. Implementar a solução numa linguagem de programação

2



Resolução De Problemas

- Um algoritmo pode ser definido como uma sequência de instruções, ordenada e sem ambiguidade, que conduzem à resolução de um problema, geralmente em português estruturado.
- Propriedades básicas
 - Constituídos por passos simples
 - Eficazes, devem resolver o problema num número finito de passos

3



Resolução De Problemas

Problema → **Algoritmo** → **Programa**

Possuo £310.
Qual é o
equivalente
em escudos?

Enunciado
claro e preciso

Início
Libra = 328
Esc = 310 * Libra
Escrever("Possuo", Esc)
Fim

Português
estruturado

```
Sub Libras_em_Esc()  
  Libra = 328  
  Esc = 310 * Libra  
  MsgBox("Possuo " & Esc)  
End Sub
```

Linguagem de
programação

4



Algoritmos

■ Tipos de dados

■ Numéricos:

- Inteiros (12, 1, 1908654, ...)
- Reais (-12.4, 0.0000765, ...)

■ Cadeias de caracteres

- Alfabéticos, algarismos ("Carlos", "Semestre2," ...)
- Outros símbolos (!, #, @, £, ...)

As cadeias de caracteres são representadas entre aspas.

5



Algoritmos

■ Variáveis

Em regra, é necessário guardar os dados utilizados num algoritmo. Para esse efeito existem as **variáveis**.

■ Características

- **Nome** : sugestivo e curto, iniciado sempre por uma letra
- **Tipo** : real, inteiro, cadeia de caracteres)
- **Tamanho**: Cada variável ocupa um determinado espaço na memória do computador.
- **Única**: Guarda um valor de cada vez.

- Os dados são "armazenados" nas variáveis através da operação de atribuição, é representada pelo operador ←.

Ex. Nome ← "Ana"

6



Algoritmos

- Operadores

- **Aritméticos**

- * ; / ; $\uparrow$: multiplicação, divisão e exponenciação
- + ; - : adição e subtracção

- **Relacionais**

- <= ; >= ; <> : menor ou igual, maior ou igual, diferente
- = ; < : igualdade e menor

7



Algoritmos

- Operadores

- **Lógicos**

- Negar : Negação
- E : conjunção
- Ou : disjunção

8



Algoritmos

■ Precedência dos operadores

A avaliação de expressões aritméticas é feita de acordo com a precedência atribuída a cada dos operador.

Prioridade

Máxima	■ ↑
	■ * ; /
	■ + ; -
	■ = ; < , > ; <= ; >=
	■ Negar ; E ; Ou
Mínima	■ ↓

9



Algoritmos -Funções pré-definidas

- **ABS (e)** e: expressão real inteira; Valor absoluto de e
 - Ex: ABS(-5)=5
- **Sqrt (e)** e: expressão real ou inteira e > 0; raiz quadrada de e
 - Ex: Sqrt(9)=3
- **Trunc (e)** e: expressão real; Valor truncado
 - Trunc(3,5)=3
- **Round(e)** e: expressão real; Valor arredondado
 - Round(4,5)=5
- **Mod (a,b)** a,b números reais; Resto da divisão de a por b
 - Mod (5,4)=1

10



Algoritmos

■ Notações

Pseudocódigo – Português estruturado, mais próximo de uma linguagem de programação.

Palavras chave

Significado

- | | |
|------------|-----------------------|
| ■ Início | - início do algoritmo |
| ■ Fim | - fim do algoritmo |
| ■ Ler | - operação de leitura |
| ■ Escrever | - operação de escrita |

11



Algoritmos

■ Estruturas de controlo

Pode escrever-se qualquer programa recorrendo apenas a três formas de controlo da sua execução : **sequencial**; **condicional** e **repetitiva**.

Execução Sequencial – As instruções de um programa são executadas pela ordem em que se encontram escritas.

12



Algoritmos

- Estruturas de controlo sequencial

Exemplo

Ínicio

Escrever (“ Digite o seu ano de nascimento,
no formato aaaa”)

Ler (ano_actual, ano_de_nascimento)

Idade = ano_actual - ano_de_nascimento

Escrever (“ Já tem ou vai fazer “, Idade ,
“anos”)

Fim

13



Algoritmos

- Estruturas de controlo

Execução **Condicional** – As instruções de um programa são executadas pela ordem em que se encontram escritas, respeitando determinadas condições .

Se <condição>

Então <conjunto de instruções>

Senão <outro conjunto de instruções>

FimSe

14



Algoritmos

■ Estruturas de controlo condicional

Exemplo

Início

Escrever (“ Digite o seu ano de nascimento, no formato aaaa”)

Ler (ano_actual, ano_de_nascimento)

Se (ano_actual e ano_de_nascimento) > 0

Então

Idade = ano_actual - ano_de_nascimento

Escrever (“ Já tem ou vai fazer “, Idade , “anos”)

Senão Escrever (“Valores de entrada inválidos”)

FimSe

Fim

15



Algoritmos

■ Estruturas de controlo

Execução **Repetitiva** – As instruções de um programa são executadas pela ordem em que se encontram escritas, respeitando determinadas condições e podendo executar-se mais do que uma vez.

Fazer Enquanto <condição>

<conjunto de instruções>

Fim Fazer

16



Algoritmos

■ Estruturas de controlo Repetitiva

Exemplo

Início

Escrever(" Digite o numero de elementos da turma: ")

Ler(Numero)

Fazer Enquanto Numero>0

Escrever ("digite a idade: ")

Ler (Idade)

Se Idade >= 21

Então Escrever (" O aluno tem 21 anos ou mais")

Senão Escrever ("O aluno tem menos de 21 anos")

FimSe

Numero= Numero - 1

Fim Fazer

Fim

17



Algoritmos

■ Traçagens

A traçagem consiste em testar um algoritmo para determinados valores de entrada, observando o seu comportamento. Todos os passos do algoritmo devem ser numerados: P1, P2 ...

A forma mais simples é construir uma tabela, em que os títulos das colunas são constituídas por todas as operações efectuadas pelo algoritmo, desde atribuições, condições, repetições.

Os títulos das linhas correspondem aos passos P1, P2, etc.

O algoritmo pode a partir de agora ser executado manualmente. A ideia é determinar se a sua lógica é válida e corresponde aos objectivos que se pretendem atingir.

18



Programação em VBA

O que é o Visual Basic for Applications - VBA?

- O VBA é uma linguagem de programação integrada no Excel e noutras aplicações da Microsoft.
- Para activar a barra de ferramentas do VBA :
 - Ver + Barra de ferramentas + Visual Basic
- Para activar o editor de texto VBE do VBA :
 - Alt +F11 (onde vai ser possível escrever o código)

19



Programação em VBA

- O VBA vai informando o utilizador relativamente ao código que este escreve, permitindo detectar erros antecipadamente:
 - A capitalização das letras é indiferente.
 - Os comentários começam por ' e apresentam cor verde.
 - As palavras chave têm cor azul.
 - As instruções com erro têm cor vermelha.

20



Programação em VBA

- A programação em VBA é constituída por módulos. O elemento básico denomina-se procedimento.
- Existem dois tipos de procedimentos:
 - Funções — retorna ou devolve um valor
 - Sub-rotinas
 - Pode retornar um valor ou mais
 - Começa com Sub e termina com End Sub
 - Entre estas duas palavras chave escreve-se o código.

21



Programação em VBA

■ Funcções

Cálculo da média de dois valores através de uma função.

```
Function média (C , A , B)  
    C=(A+B)/2  
End Function
```

■ Sub-Rotinas

Cálculo da média de dois valores através de uma sub-rotina.

```
Sub média()  
    a = Application.InputBox("Valor 1= ? ")  
    b = Application.InputBox(" Valor 2 =? ")  
    c = (a + b) / 2  
    MsgBox ("valor médio = " & c)  
End Sub
```

22



Programação em VBA

- Como executar uma sub-rotina:
 - No menu Ferramentas, seleccionar Macro, escolher a sub-rotina e optar por Executar.
 - Associando a sub-rotina a:
 - uma **tecla de atalho**, do tipo Alt+ Letra (Ferramentas/Macro/Macros/Opções)
 - ou a um **Botão** (Ferramentas/Personalizar/Macro).
 - ou ainda a um **item do menu** (processo idêntico ao item anterior).

23



Programação em VBA

- Tipos de variáveis
 - **String** - Sequência de caracteres (\$)
 - **Integer** - Inteiros de -32768 a 32767 (%)
 - **Long** - Inteiros de -2147483648 a 2147483647 (&)
 - **Single** - Negativos e positivos de baixa precisão (!)
 - **Double** - Negativos e positivos de alta precisão (#)
 - **Currency** - Moedas (@)
 - **Date** - Datas
 - **Boolean** - Verdadeiro/Falso
 - **Array** - Armazena matrizes

24



Programação em VBA

- Definir o tipo de uma variável com a instrução DIM
Dim Numero As Single
Dim Nome As String
Dim Data_de_Nascimento As Date
- Em alternativa, numa única linha:
Dim Num As Single, Nome As String, Data_Abertura As Date
- Definir o conteúdo de uma variável explicitamente
Num= 25 ; Nome= "João Silva"

25



Programação em VBA

Pseudocódigo	VBA
Escrever ("Nome :", Nome) Ler (Nome)	MsgBox ("Nome :" & Nome) Nome = InputBox ("")
Sub Calculo_da_Idade() Dim Idade, Ano As Integer Ano = Application.InputBox ("Digite o seu ano de nascimento, no formato aaaa") If (Ano > 1900) Then Idade = 2001 - Ano MsgBox ("Já tem ou vai fazer " & Idade & " anos") Else MsgBox ("O ano de nascimento é um valor inválido") End If End Sub	

26



Programação em VBA

Pseudocódigo

Exemplos em VBA

```
Se <condição> então  
    <instruções>  
FimSe
```

```
If Dn < 1900 Then  
    MsgBox ( "Ano inválido"  
End If
```

```
Se <condição> então  
    <Grupo de instruções>  
Senão  
    <Novo grupo de instruções>  
FimSe
```

```
If Dn < 1900 Then  
    MsgBox ("Ano inválido")  
Else  
    Idade= 2000-Dn  
EndIf
```

27



Programação em VBA

Estruturas repetitivas

Pseudocódigo

Exemplos em VBA

```
Repetir Enquanto <condição>  
    <Grupo de instruções>  
Fim Repetir
```

```
Do While x < 10  
    x=x+1  
Loop
```

Nota: Executa enquanto a condição for VERDADEIRA

```
Fazer Até <condição>  
    <instruções>  
Fim Fazer
```

```
Do Until x>10  
    x=x+1  
Loop
```

Nota: Executa enquanto a condição for FALSA

28

Programação em VBA

Estruturas repetitivas

Pseudocódigo

Para Variável = Constante até Valor ou Expressão numérica
<Grupo de instruções>
Próxima Variável

Ex. em VBA

```
For I=1 to 10  
    x=x+1  
Next I
```

Nota: Executa até que I atinja o valor 10

29

Programação em VBA

Seleção de casos

Pseudocódigo

Seleccionar caso Opção
Caso Opção=1
 Somar
Caso Opção =2
 Multiplicar
Caso Opção =3
 Dividir
Caso Opção =4
 Subtrair
Caso Senão
 Erro
Fim Seleccionar

Exemplo em VBA

```
Select Case Opção  
Case Opção = 1  
    Somar ()  
Case Opção = 1  
    Multiplicar()  
Case Opção = 1  
    Dividir()  
Case Opção = 1  
    Subtrair()  
Case Else  
    Erro()  
End Case
```

30