**Linguagem**

Linguagem Algorítmica

A linguagem algorítmica utiliza o português para a definição dos comandos e tem as seguintes característica:

1 - A linguagem não á sensível á capitulação ( não faz distinção entre letras minúsculas e maiúsculas)  
2 - Todos os algoritmos começam por inicio e terminam com fim  
3 - cada linha contém apenas um comando (o enter funciona como sinalizador de fim de comando)  
4 - O sinal de atribuição é a seta ( <- )  
5 - A definição de símbolos é feita em qualquer local do algoritmo

Sintaxe

inicio  
  Algoritmo  
fim

**Decisão**

**Escolhe**

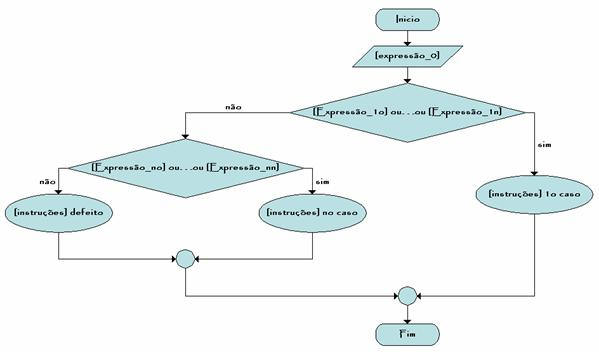
**Sintaxe**

Escolhe [expressão\_0]  
    Caso [Expressão\_1o],. . .,[Expressão\_1n],:  
        [instruções]  
     . . . . . . . . . . .        
    Caso [Expressão\_no],. . .,[Expressão\_nn],:  
        [instruções]      
    Defeito:  
        [instruções]      
FimEscolhe

**Descrição**

Esta estrutura compara a expressão\_0 com as restantes expressões (expressão\_1 . . . expressão\_n) e se a comparação for positiva esxecuta as instruções do Caso e sai.  
A estrutura tem um caso por defeito (opcional) que é executado quando todas as outras condições falharem.

**Fluxograma**



**Exemplo**

inicio  
    caracter sexo  
    escrever "Qual o seu sexo :"  
    ler sexo  
    escrever "Olá "  
    escolhe sexo  
        caso "m" :  
            escrever "Homem"  
        caso "f" :  
            escrever "Mulher"  
        defeito :  
            escrever "Sexo indefinido"   
    fimescolhe  
    escrever ", bemvindo ao portugol"  
fim

**Resultado**

Qual o seu sexo ?: m  
Olá Homem, benvindo ao Portugol

| **Anexos** | **Tamanho** |
| --- | --- |
| [ultimo\_dia.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/ultimo_dia.alg) | 468 bytes |

**Se**

**Sintaxe**

se [condição] então  
    [instruções]  
senão  
    [instruções]  
fimSe

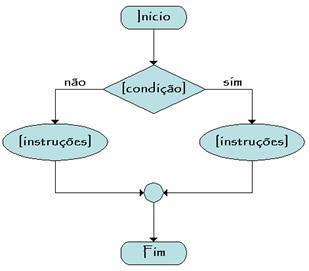
ou

**se [condição] então  
    [instruções]  
fimSe**

**Descrição**

Selecção de instruções. A condição deve ser uma variável lógica ou uma expressão com resultado lógico (operadores lógicos ou relacionais).

**Fluxograma**



**Exemplo**

inicio  
    inteiro numero, r  
    escrever " introduza um numero: "  
    ler numero  
    r <- numero % 2  
    se r = 0 entao  
        escrever "numero par"  
    senao  
        escrever "numero impar"  
    fimse  
fim

**Resultado**

introduza um numero: 5  
numero impar

| **Anexos** | **Tamanho** |
| --- | --- |
| [eq2g.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/eq2g.alg) | 1.24 KB |
| [se\_par\_impar.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/se_par_impar.alg) | 209 bytes |
| [nome\_sexo.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/nome_sexo.alg) | 321 bytes |

**Entrada/Saída**

**Escrever**

**Sintaxe**

escrever variável  
escrever constante  
escrever "texto"  
escrever [expressão]  
escrever variável , "texto" , expressão , . . .,

Escreve texto variáveis, constantes, expressões  ou texto na consola.  
A mesma instrução pode escrever vario elementos desde que sejam separados por virgulas (**,**)  
Existem três símbolos predefinidos no texto:

* **\n** - mudança de linha (enter)
* **\t** - tubulação
* **\"** - caracter aspas (")

**Descrição**

Solicita a introdução de valores para as variáveis através da consola

**Exemplo**

inicio  
    texto nome  
    escrever "qual o seu nome:"  
    ler nome  
    escrever "\nbem-vindo ao portugol \t\"" , nome , "\""  
fim

**Resultado**

qual o seu nome: António  
bem-vindo ao portugol     "António"

**Ler**

**Sintaxe**

ler variável  
ler variavel1, variavel2

**Descrição**

Solicita a introdução de valores para as variáveis através da consola. É possível definir a leitura de várias variáveis desde que estas estejam separadas por virgulas (,). A leitura de uma variável deve ser precedida da sua declaração. Os valores introduzidos para cada variável devem ser compatíveis com o tipo definido na declaração da mesma.

**Exemplo**

inicio  
    texto nome  
    escrever "qual o seu nome:"  
    ler nome  
    escrever "bem-vindo ao portugol " , nome  
fim

**Resultado**

qual o seu nome: António  
bem-vindo ao portugol António

**Operadores**

**Aritméticos**

**Inteiros**

**Operadores disponíveis para números inteiros**

+    Adição  
-    Subtracção  
/    Divisão inteira  
\*    Multiplicação  
%    Resto da divisão inteira   
^    Potenciação

**Descrição**

Operadores que utilizam exclusivamente variáveis ou valores inteiros

**Exemplo**

inicio  
    **inteiro** i1 <- 5 , i2 <- 2   
    escrever "operadores inteiros"  
    escrever "\n" , i1 , " + " , i2 , " = " , i1 + i2  
    escrever "\n" , i1 , " - " , i2 , " = " , i1 - i2  
    escrever "\n" , i1 , " \* " , i2 , " = " , i1 \* i2  
    escrever "\n" , i1 , " / " , i2 , " = " , i1 / i2  
    escrever "\n" , i1 , " % " , i2 , " = " , i1 % i2  
    escrever "\n" , i1 , " ^ " , i2 , " = " , i1 ^ i2  
fim

**Resultado**

operadores inteiros  
5 + 2 = 7  
5 - 2 = 3  
5 \* 2 = 10  
5 / 2 = 2  
5 % 2 = 1  
5 ^ 2 = 25

| **Anexos** | **Tamanho** |
| --- | --- |
| [operadores\_inteiros.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/operadores_inteiros.alg) | 346 bytes |

**Reais**

**Operadores disponíveis para números reais**

+    Adição  
-    Subtracção  
/    Divisão inteira  
\*    Multiplicação  
^    Potenciação

**Descrição**

Operadores que utilizam variáveis ou valores numéricos em que **pelo menos um** elemento é do tipo real.

**Exemplo**

inicio  
    **real** i1 <- 5.0 , i2 <- 2.0   
    escrever "operadores inteiros"  
    escrever "\n" , i1 , " + " , i2 , " = " , i1 + i2  
    escrever "\n" , i1 , " - " , i2 , " = " , i1 - i2  
    escrever "\n" , i1 , " \* " , i2 , " = " , i1 \* i2  
    escrever "\n" , i1 , " / " , i2 , " = " , i1 / i2  
    escrever "\n" , i1 , " ^ " , i2 , " = " , i1 ^ i2  
fim

**Resultado**

operadores real  
5.0 + 2.0 = 7.0  
5.0 - 2.0 = 3.0  
5.0 \* 2.0 = 10.0  
5.0 / 2.0 = 2.5  
5.0 ^ 2.0 = 25.0

| **Anexos** | **Tamanho** |
| --- | --- |
| [operadores\_real.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/operadores_real.alg) | 149 bytes |

**Texto**

**Operadores disponíveis para texto**

+    Concatenação de texto

**Descrição**

Operadores que utilizam variáveis ou valores do tipo texto

**Exemplo**

inicio  
    texto nome1 <-"Linguagem"  
    texto nome2 <-"Algoritmica"  
    texto nome3  
    nome3 <- nome1 + " " + nome2  
    escrever nome3   
fim

**Resultado**

Linguagem Algoritmica

| **Anexos** | **Tamanho** |
| --- | --- |
| [operadores\_texto.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/operadores_texto.alg) | 146 bytes |

**Lógicos**

**Operadores disponíveis**

E    Disjunção   
OU   Conjunção   
XOU  Conjunção Exclusiva   
NAO  Negação

**Descrição**

Os operadores lógicos fornecem com resultado um valor lógico (verdadeiro ou falso).   
Os operadores E e OU são operadores binários e  o NAO é um operador unário.

**Exemplo**

inicio  
    logico l1 , l2  
    l1 <- verdadeiro  
    l2 <- falso  
    escrever "\n não " , l1 , "\t= " , nao l1  
    escrever "\n não " , l2 , "\t= " , nao l2  
    escrever "\n\n" , l1 , " e \t" , l2 , " = " , l1 e l2  
    escrever "\n" , l1 , " ou \t" , l2 , " = " , l1 ou l2  
    escrever "\n" , l1 , " xou \t" , l2 , " = " , l1 xou l2  
    l1 <- verdadeiro  
    l2 <- verdadeiro  
    escrever "\n\n" , l1 , " e \t" , l2 , " = " , l1 e l2  
    escrever "\n" , l1 , " ou \t" , l2 , " = " , l1 ou l2  
    escrever "\n" , l1 , " xou \t" , l2 , " = " , l1 xou l2  
    l1 <- falso  
    l2 <- falso  
    escrever "\n\n" , l1 , " e \t" , l2 , " = " , l1 e l2  
    escrever "\n" , l1 , " ou \t" , l2 , " = " , l1 ou l2  
    escrever "\n" , l1 , " xou \t" , l2 , " = " , l1 xou l2  
fim

**Resultado**

não VERDADEIRO = FALSO  
não FALSO = VERDADEIRO  
VERDADEIRO e FALSO = FALSO  
VERDADEIRO ou FALSO = VERDADEIRO  
VERDADEIRO xou FALSO = VERDADEIRO  
VERDADEIRO e VERDADEIRO = VERDADEIRO  
VERDADEIRO ou VERDADEIRO = VERDADEIRO  
VERDADEIRO xou VERDADEIRO = FALSO  
FALSO e FALSO = FALSO  
FALSO ou FALSO = FALSO  
FALSO xou FALSO = FALSO

| **Anexos** | **Tamanho** |
| --- | --- |
| [operadores\_logico.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/operadores_logico.alg) | 777 bytes |

**Relacionais**

**Operadores disponíveis**

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

**Descrição**

Os operadores lógicos fornecem com resultado um valor lógico (verdadeiro ou falso)

* Se os operandos forem numéricos a comparação é feita com base no seu valor
* Se os operadores forem texto a comparação é feita lexicograficamente
* Se os operadores forem lógicos apenas estão disponíveis a igualdade e a diferença

**Exemplo**

inicio  
    real l1 <- 5.0 , l2 <- 3   
    escrever "\n" , l1 , " = " , l2 , "\t= " , l1 = l2  
    escrever "\n" , l1 , " =/= " , l2 , "\t= " , l1 =/= l2  
    escrever "\n" , l1 , " > " , l2 , "\t= " , l1 > l2  
    escrever "\n" , l1 , " >= " , l2 , "\t= " , l1 >= l2  
    escrever "\n" , l1 , " < " , l2 , "\t= " , l1 < l2  
    escrever "\n" , l1 , " <= " , l2 , "\t= " , l1 <= l2  
fim

**Resultado**

5.0 = 3.0 = FALSO  
5.0 =/= 3.0 = VERDADEIRO  
5.0 > 3.0 = VERDADEIRO  
5.0 >= 3.0 = VERDADEIRO  
5.0 < 3.0 = FALSO  
5.0 <= 3.0 = FALSO

| **Anexos** | **Tamanho** |
| --- | --- |
| [operadores\_relacionais.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/operadores_relacionais.alg) | 385 bytes |

**Repetição**

**Enquanto**

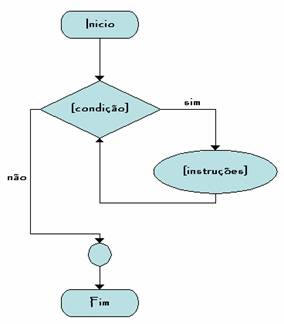
**Sintaxe**

enquanto [condição] faz  
    [intruções]  
fimEnquanto

**Descrição**

Repeta as intruções enquanto a condição for verdadeira

**Fluxograma**



**Exemplo**

inicio  
    inteiro contador  
    contador <- 0  
    **enquanto** **contador < 10** **faz**  
        escrever contador , "\t"  
        contador <- contador + 1  
    **fimenquanto**  
fim

**Resultado**

0    1    2    3    4    5    6    7    8    9

| **Anexos** | **Tamanho** |
| --- | --- |
| [enquanto\_conta10.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/enquanto_conta10.alg) | 164 bytes |

**Faz**

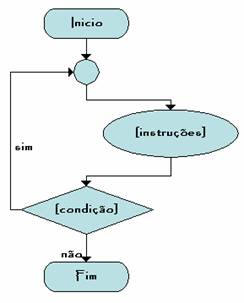
**Sintaxe**

Faz  
    [instruções]  
Enquanto[condição]

**Descrição**

Repete as instruções até que a condição seja falsa.

**Fluxograma**



**Exemplo**

Solicita a introdução de um mês até o utilizador introduzir um valor válido ( entre 1 e 12)

inicio  
    inteiro mes  
    faz  
        escrever "introduza um mes :"  
        ler mes  
    enquanto mes < 0 ou mes > 13  
    escrever "\nmes introduzido :" , mes  
fim

**Resultado**

introduza um mes :-10  
introduza um mes :20  
introduza um mes :5

mes introduzido :5

**Outros exemplos**

* Informação nao disponível

| **Anexos** | **Tamanho** |
| --- | --- |
| [faz\_pede\_mes.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/faz_pede_mes.alg) | 187 bytes |

**Para**

**Sintaxe**

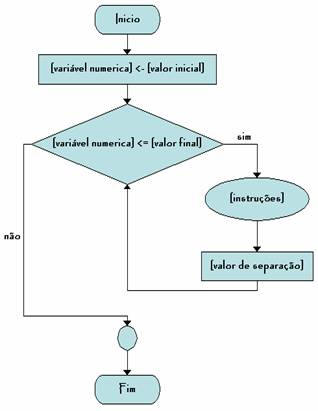
para [variável numerica] de [valor inicial] ate [valor final] passo [valor de separação]  
    [intruções]  
próximo

para [variável numerica] de [valor inicial] ate [valor final]   
    [intruções]  
próximo

**Descrição**

Repete as instruções controlado por uma variável numérica que percorre os valores entre dois limites utilizando o passo definido. Caso o passo não seja definido, o passo é de mais ou menos uma unidade de forma a poder percorrer o intervalo entre o valor inicial e o final de forma crescente ou decrescente.

**Fluxograma**



**Exemplo**

inicio  
    inteiro contador  
    **para** **contador** **de** **0** **ate** **9** **passo** **1**  
        escrever contador , "\t"  
    **proximo**  
fim

**Resultado**

0    1    2    3    4    5    6    7    8    9

| **Anexos** | **Tamanho** |
| --- | --- |
| [para\_conta10.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/para_conta10.alg) | 115 bytes |

**Repete**

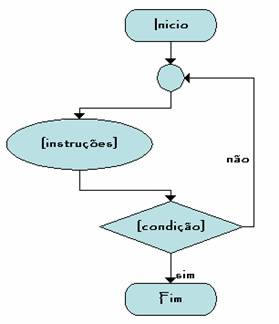
**Sintaxe**

repete  
    [instruções]  
até[condição]

**Descrição**

Repete as instruções até que a condição seja verdadeira.

**Fluxograma**



**Exemplo**

Solicita a introdução de um mês até o utilizador introduzir um valor válido ( entre 1 e 12)

inicio  
    inteiro mes  
    repete  
        escrever "introduza um mes :"  
        ler mes  
    ate mes > 0 e mes < 13  
    escrever "\nmes introduzido :" , mes  
fim

**Resultado**

introduza um mes :-10  
introduza um mes :20  
introduza um mes :5

mes introduzido :5

| **Anexos** | **Tamanho** |
| --- | --- |
| [DataNascimento.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/DataNascimento.alg) | 1.59 KB |
| [repete\_pede\_mes.alg](http://www.dei.estt.ipt.pt/portugol/sites/dei.estt.ipt.pt.portugol/files/repete_pede_mes.alg) | 161 bytes |

**Tipos de dados**

**Arrays**

**Sintaxe**

<variavel/constante> <tipo dados> nome[dimensão]  
<variavel/constante> <tipo dados> nome[dim1][dim2]. . . [dimn]  
<tipo dados> nome[dimensão]  
<tipo dados> nome[dimensão] <- {valor\_1, valor\_2, . . . , valor\_n}

**Estruturas homogéneas de dados (Arrays)**

define um conjunto de dados todos do mesmo tipo que podem ser acedidos através de um índice.  
regras:

1- Se a declaração omitir o termo <variavel/constante> o array é variavel  
2- A definição do nome obedeçe ás mesmas regras das [variáveis](http://www.dei.estt.ipt.pt/tipo_dados_variaveis.html)  
3- A dimensão tem de ser um valor inteiro ou uma expressão com resultado inteiro  
4 - As dimensões estão contidas dentro de parênteses rectos  
5 - São permitidas tantas dimensões quantas as desejadas  
6- Se o valor de inicialização for omitido todas o conjunto é inicializado com os valores por [defeito](http://www.dei.estt.ipt.pt/tipo_dados_basicos.html)  
7 - A inicialização é feita com os dados compreendidos entre chavetas e separados por virgulos  
8 - Se o número de valores for insuficiente os últimos elementos são incializado com os valores por [defeito](http://www.dei.estt.ipt.pt/tipo_dados_basicos.html)

**Exemplo**

constante texto estacoes[4] <- {"inverno", "outono", "primavera" , "verão" }  
variavel inteiro notas[3] <- {10,20}  
real salarios[12]  
logico bits[8]

**Básicos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo** | **Descrição** | **Valores** | **Valor por defeito** |
| **Inteiro** | valores ordinais definidos com quatro bits | -2 147 483 648  2 147 483 647 | 0 |
| **Real** | Valores com parte decimal definidos com 64 bits | -1.7 E 308 1.7 E 308 | 0.0 |
| **Lógico** | Valore lógicos - 1 bit | verdadeiro falso | falso |
| **Carácter** | Caracteres da Tabela ASCII | ASCII(0) ASCII(255) | " " (espaço) |
| **Texto** | Conjuntos de caracteres | "Sequências de caracteres"  "entre aspas" | "" (vazio) |

**Constantes**

**Sintaxe**

constante [tipo] [nome] <- [valor]  
constante [tipo] [nome] <- [expressão]  
constante [tipo] [nome] <- [valor] ,[nome] <- [expressão]

**Definição**

Define um símbolo cujo valor permanece inalterável durante os seu ciclo de vida.  
Segue as mesmas regras que a definição de variáveis excepto que **não é possível omitir o valor de inicialização**

**Exemplos**

constante inteiro meses <- 12  
constante real pi <- 3.14

**Variáveis**

**Sintaxe**

variavel [tipo] [nome] <- [expressão]  
variavel [tipo] [nome] <- [valor] ,[nome] <- [expressão]   
[tipo] [nome] <- [expressão]  
[tipo] [nome]

**Definição**

Define um símbolo cujo valor pode ser alterado durante os seu ciclo de vida.  
As variáveis são o elemento básico de processamento. A sua declaração permite definir que tipo de informação irão conter.   
A declaração de variaveis segue as seguintes regras:  
1- O nome tem de começar por uma letra ou pelo carácter  undescore (\_) e não pode ser uma palavra reservada da linguagem.  
2- O valor de inicialização tem de ser compativel com tipo de variável definido.  
3- Se o valor de inicialização for omitido a variavel é inicializada com os [valores por defeito](http://www.dei.estt.ipt.pt/tipo_dados_basicos.html)  
4- É possivel definir mais que uma variavel utilizando o caracter virgula (,)  
5- É possível omitir a palavra variavel.

**Exemplos**

variavel inteiro idade <- 18  
variavel real peso <- 23.14  
logico repetente  
texto nome <- "Antonio" , apelido <- "Sousa"  
caracter sexo