

## PSEUDOCÓDIGO

O traballo con pseudocódigo presenta algunhas vantaxes sobre a utilización de Diagramas de Fluxo:

- Vainos permitir representar de forma moi sinxela operacións repetitivas complexas.
- Ocupa menos espazo que o Diagrama de Fluxo.
- O paso de pseudocódigo a Linguaxe de programación soe ser moi sinxelo.
- Pódense ver perfectamente os niveis e anidamento das estruturas do programa.

Non hai un pseudocódigo normalizado, por exemplo aquí tedes tres formas de representar o pseudocódigo do algoritmo suma:

```
Proceso suma_de_dos_numeros
Escribir "Ingrese los dos numeros a sumar";
Leer num1,num2;
suma<-num1+num2;
Escribir "la suma de ",num1," mas ",num2," es igual a ",suma;
FinProceso
```

```
Algoritmo Suma_de_dous_números
Definir num1,num2,suma Como Entero
Escribir "Introduce o primeiro número"
Leer num1
Escribir "Introduce o segundo número"
Leer num2
suma=num1+num2
Escribir "A suma é: ",suma
FinAlgoritmo
```

Inicio

```
num1, num2, suma: ENTERO
Escribir "Introduza número: "
Ler num1
Escribir "Introduza número: "
Ler num2
suma ← num1 + num2
Escribir "Resultado: ", suma
```

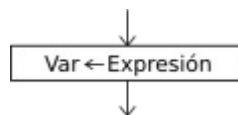
Fin

Aquí imos resumir os termos en pseudocódigo que utiliza o programa pseint:

### Definición de variables

Definir A Como Real, Entero, Logico, Caracter.

### Asignación



A instrucción de asignación permite almacenar un valor nunha variable.

<variable> <- <expresión>;

Ao executarse a asignación, primeiro evalúase a expresión da dereita e logo asignase o resultado á variable da esquerda. O tipo da variable e da expresión deben coincidir.

Entre as principais funcións atopamos a asignación que consiste no paso de valores a unha zona de memoria, a cal recoñecemos polo nome da variable á que se lle asigna ese valor. Existen diferentes convencións á hora de realizar esta tarefa.

O formato a utilizar é un dos seguintes (depende da convención ou aplicación utilizada para representalo):

```
< Variable > ← <valor ou expresión >  
< Variable > := <valor ou expresión >  
< Variable > = <valor ou expresión >
```

Deste xeito indicamos que a unha variable se lle asigna un valor ou unha expresión.

Vexamos algún exemplo. Supoñamos que queremos asignar á variable Prezo o valor 123. Segundo o autor, podemos atopar estas tres formas de facelo:

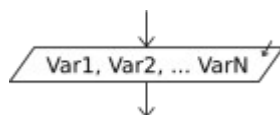
```
Prezo ← 123  
Prezo := 123  
Prezo = 123
```

O mesmo ocorre coas expresións, por exemplo coas operacións aritméticas.

Supoñamos que queremos sumar o valor contido en dúas variables (num1 e num2) e asignarllas a unha terceira variable (result). Estas tres expresións son equivalentes (só depende da convención ou aplicación para xeración de pseudocódigo utilizada):

```
Result ← num1 + num2 // nos tres casos  
Result := num1 + num2 // sumamos num1 + num2 e  
Result = num1 + num2 // o resultado é gardado en Result
```

### **Lectura ou entrada de datos**



A instrución Leer permite ingresar información . Consiste en recibir dende un dispositivo de entrada como pode ser o teclado, un valor ou dato.

**Leer** <variable1> , <variable2> , ... , <variableN> ;

### **Escritura ou saída de datos**



A instrución Escribir permítenos enviar a un dispositivo de saída, como pode ser pantalla ou impresora, un resultado (valor dunha variable) ou un texto ou mensaxe.

**Escribir** <expr1> , <expr2> , ... , <exprN> ;

Escribir "O resultado da suma é: " Result

// Neste caso escribe un texto e a continuación o contido da variable Result

// é dicir, se Result tivese o valor 23, mostraría: O resultado da suma é 23

Como vedes no exemplo anterior é posible atopar no medio do código dos algoritmos os símbolos "//" seguidos de texto. Ésta é a forma que temos en pseudocódigo de incluír comentarios que non teñen ningún valor no código, tan só son informativos, para aclarar e documentar diferentes operacións e estruturas utilizadas no algoritmo.

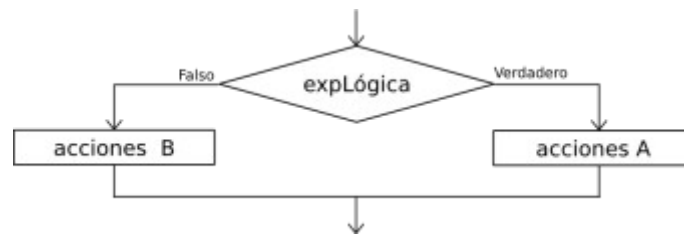
### **Estruturas selectivas ou condicionais**

As instrucións condicionais ou selectivas representan instrucións que poden executarse ou non, segundo o cumprimento dunha condición. Existen tres tipos de estruturas alternativas:

- Estruturas alternativas ou selectivas simples
- Estruturas alternativas ou selectivas múltiples
- Estruturas alternativas ou selectivas CASO

### **Estruturas alternativas ou selectivas simples**

#### Condiciona Si-Entonces



Si <condición> Entonces  
    <instrucións>  
Sino  
    <instrucións>  
FinSi

Sino pode non estar.

As estruturas condicionais comparan unha variable contra outro ou outros valores. En función do resultado desa comparación, sigue un curso de acción dentro do programa.

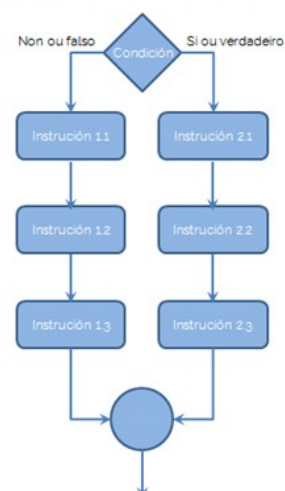
#### **Pseudocódigo**

Se <condición> entón  
    Instrucións<sub>1</sub>;  
Senón  
    Instrucións<sub>2</sub>;  
Fin se

Às veces non hai ningunha instrución na parte "Senón"

#### **Diagrama de Fluxo**

ESTRUTURA ALTERNATIVA OU SELECTIVA



Exemplo: Algoritmo que determina o máximo de dous números.

```
Algoritmo máximo
Definir num1,num2 Como Real
  Escribir 'Introduce o primeiro número'
  Leer num1
  Escribir 'Introduce o segundo número'
  Leer num2

  Si num1>num2 Entonces
    Escribir 'O máximo é: ' num1
  Sino
    Escribir 'O máximo é: ' num2
  FinSi
FinAlgoritmo
```

### **Estructuras alternativas ou selectivas múltiples**

As estruturas alternativas pódense encadear, de xeito que despois dunha condición apareza outra e outra...

Exemplo 1: Algoritmo que determina o máximo de dous números e avisa se os números son iguais.

```
Algoritmo máximo
  Definir num1,num2 Como Real
  Escribir 'Introduce o primeiro número'
  Leer num1
  Escribir 'Introduce o segundo número'
  Leer num2
  Si num1=num2 Entonces
    Escribir 'Os números son iguais, tes que introducir números distintos'
  Sino
    Si num1>num2 Entonces
      Escribir 'O máximo é: ' num1
    Sino
      Escribir 'O máximo é: ' num2
    FinSi
  FinSi
FinAlgoritmo
```

Exemplo 2: Diseña un algoritmo que lea tres números enteiros. Se todos son negativos, mostra o produto dos tres. Se algún é negativo pero non todos, mostra a suma dos tres. No caso de que todos sexan positivos, suma os dous primeiros e multiplica dita suma polo terceiro.

```
Algoritmo Operacións_con_tres_números
  Definir num1,num2,num3,resultado Como Real
  Escribir 'Introduce el primer número:'
  Leer num1
  Escribir 'Introduce el segundo número:'
  Leer num2
  Escribir 'Introduce el tercer número:'
  Leer num3
  Si num1<0 Y num2<0 Y num3<0 Entonces
    resultado <- num1*num2*num3
```

```

    SiNo
        Si num1<0 O num2<0 O num3<0 Entonces
            resultado <- num1+num2+num3
        SiNo
            resultado <- (num1+num2)*num3
        FinSi
    FinSi
    Escribir 'O resultado é: ',resultado
FinAlgoritmo

```

Exemplo 3: Diseña un algoritmo ao que se introduzan como entradas dous números e un carácter. O algoritmo deberá sumar (s), restar (r) , multiplicar (m) ou dividir (d) os valores dos dous números dependendo do código indicado no terceiro parámetro, e mostrar o resultado.

Algoritmo Operacións\_matemáticas

```

    Definir num1,num2,resultado Como Real
    Definir operacion,error Como Caracter
    Escribir "Introduce o primeiro número:"
    Leer num1
    Escribir "Introduce o segundo número:"
    Leer num2
    Escribir "Que operación queres realizar? (s-suma, r-resta, m-multiplicación e d-división)"
    Leer operacion
    Si operacion='s' Entonces
        resultado=num1+num2
    SiNo
        Si operacion='r' Entonces
            resultado=num1-num2
        SiNo
            Si operacion='m' Entonces
                resultado=num1*num2
            SiNo
                Si operacion='d' Entonces
                    Si num2=0 Entonces
                        Escribir "Non se pode dividir por cero. A operación non
é válida"
                        error='0'
                    SiNo
                        resultado=num1/num2
                    Fin Si
                SiNo
                    Escribir "Introduce unha letra correcta. A operación non é
válida"
                    error='0'
                FinSi
            FinSi
        FinSi
    FinSi
    Si error <> '0' Entonces
        Escribir "O resultado é: ", resultado
    FinSi
FinAlgoritmo

```

## Selección Múltiple

### Estructuras alternativas, condicionais ou selectivas CASO

Existe outro tipo de estrutura, a alternativa múltiple, que permite simplificar algunhas estruturas alternativas aniñadas, no caso de que todas sexan excluíntes entre si.

#### Pseudocódigo

Segun <variable> Facer

  caso <valor<sub>1</sub>>

    Instrucións<sub>1</sub>

  caso <valor<sub>2</sub>>

    Instrucións<sub>2</sub>

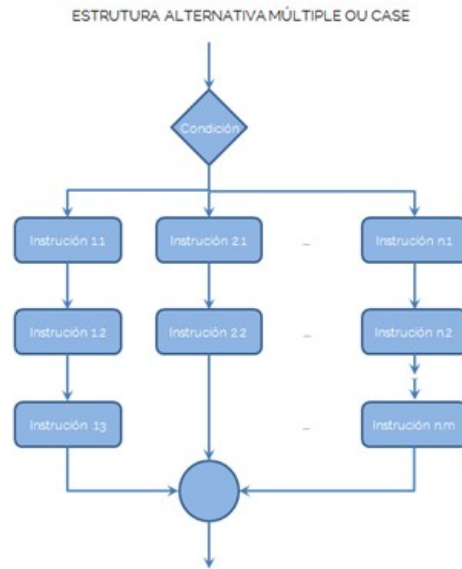
    ...

  de Outro Modo

    Instrucións<sub>n</sub>

Fin Segun

#### Diagrama de Fluxo



A secuencia de instrucións executada por unha instrución *Segun* depende do valor dunha variable numérica.

Segun variable\_numerica Hacer

  opcion\_1:

    secuencia\_de\_acciones\_1

  opcion\_2:

    secuencia\_de\_acciones\_2

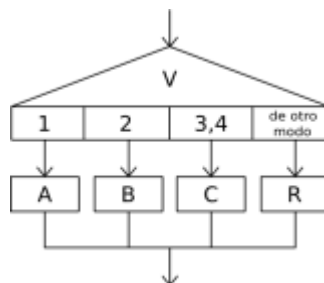
  opcion\_3:

    secuencia\_de\_acciones\_3

  De Otro Modo:

    secuencia\_de\_acciones\_dom

Fin Segun



Repetimos o exercicio anterior utilizando a instrución Segun:

### Algoritmo Operacións\_matemáticas

```
Definir num1,num2,resultado Como Real
Definir operacion Como Entero
Definir error Como Caracter
Escribir 'Introduce o primeiro número:'
Leer num1
Escribir 'Introduce o segundo número:'
Leer num2
Escribir 'Que operación queres realizar? (1-suma, 2-resta, 3-multiplicación e 4-división)'
Leer operacion
Segun operacion Hacer
  1:
    resultado <- num1+num2
  2:
    resultado <- num1-num2
  3:
    resultado <- num1*num2
  4:
    Si num2=0 Entonces
      Escribir 'Non se pode dividir por cero. A operación non é válida'
      error <- '0'
    SiNo
      resultado <- num1/num2
    FinSi
  De Otro Modo:
    Escribir 'A operación introducida non é válida. Introduce un número correcto'
    error <- '0'
FinSegun
Si error<>'0' Entonces
  Escribir 'O resultado é:',resultado
FinSi
FinAlgoritmo
```

### **Estructuras repetitivas, iterativas ou bucles**

As estruturas repetitivas son aquelas nas que é necesario utilizar un mesmo conxunto de accións que se podan executar unha cantidade específica de veces.

Esta cantidade pode ser fixa (previamente determinada polo programador) ou pode ser variable (en función dalgún dato dentro do programa).

O número de veces indícase mediante unha condición, se a condición é verdadeira executa o bloque de código que conteña, cando a condición sexa falsa sairá do bucle.

Temos tres estruturas repetitivas:

Repetir ... ata

Mientras ...facer

Para

## Estrutura repetitiva Repetir ...ata

Utilízase habitualmente cando é necesario que o grupo de instrucións se executen polo menos unha vez e ata que se cumpra a condición:

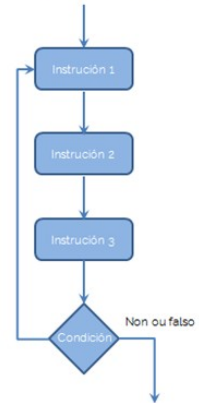
### Pseudocódigo

```

Repetir
    Instrucións;
Ata que <Condición>
    
```

### Diagrama de Fluxo

ESTRUTURA REPETITIVA TIPO 1



En pseudocódigo:

```

Repetir
    <instrucións>
Hasta Que <condición>
    
```

Neste caso as instrucións do corpo do ciclo execútanse polo menos unha vez.

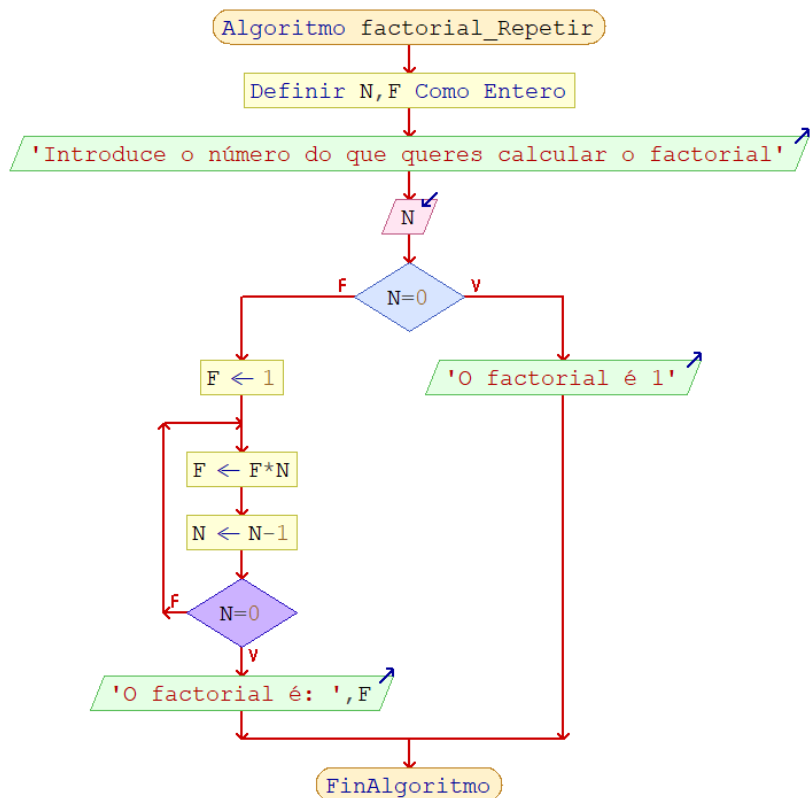
Exemplo: Deseña un algoritmo que calcule o factorial dun número N.

Algoritmo factorial\_Repetir

```

Definir N,F como Entero
Escribir "Introduce o número do que queres calcular o factorial"
Leer N
Si N=0 Entonces
    Escribir "O factorial é 1"
SiNo
    F=1
    Repetir
        F=F*N
        N=N-1
    Hasta Que N=0
    Escribir "O factorial é: ",F
Fin Si
FinAlgoritmo
    
```

FinAlgoritmo





## Estrutura repetitiva Mientras ... facer

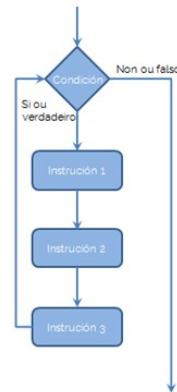
O bucle repitirase *mentras* sexa certa a condición, de xeito que se ao chegar por primeira vez ao bucle "*mentras*" a condición é falsa, o grupo de instrucións do bucle non se executará ningunha VEZ.

### Pseudocódigo

```
Mientras <condición> Facer
    Instrucións;
Fin Mientras
```

### Diagrama de Fluxo

ESTRUTURA REPETITIVA TIPO 2



A instrución *Mientras* executa unha secuencia de instrucións mentras unha condición sexa verdadeira. En pseint:

```
Mientras <condición> Hacer
    <instrucións>
FinMientras
```

No exemplo anterior:

Algoritmo factorial\_Mientras

Definir N,F Como Entero

Escribir 'Introduce o número do que queres calcular o factorial'

Leer N

F <- 1

Mientras N>1 Hacer

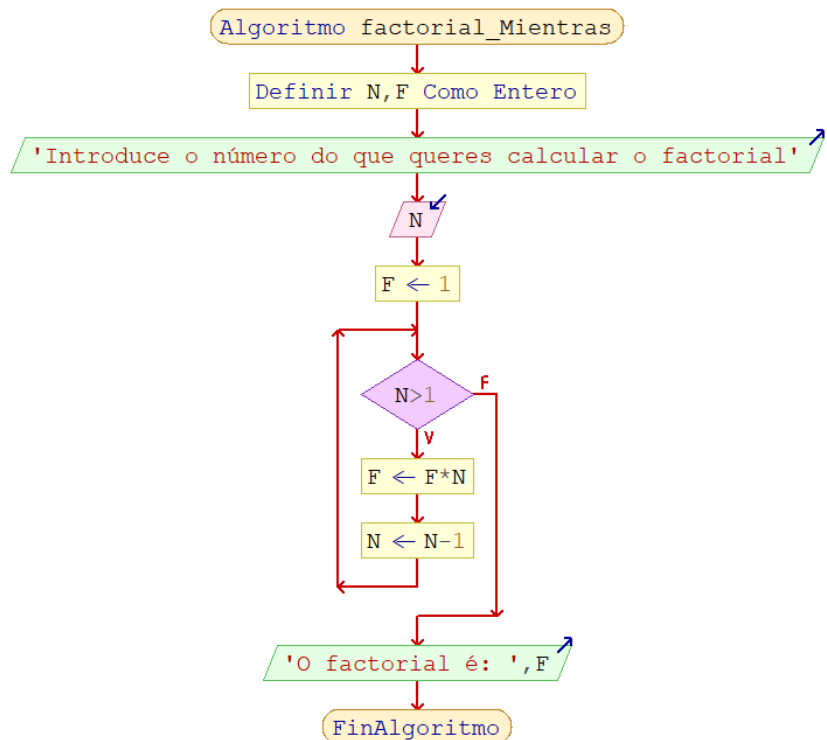
F <- F\*N

N <- N-1

FinMientras

Escribir 'O factorial é: ',F

FinAlgoritmo



## Estrutura repetitiva Para

Este bucle interpretámolo como "Dende o contador inicial  $contador=x$  ata o límite  $n$ , Repetimos un conxunto de instrucións, modificando en cada bucle o valor de contador segundo indiza  $s$ ."

### Pseudocódigo

Para contador  $\leftarrow x$  ata  $n$  a incremento de  $s$  facer

Instrucións

fin para

ou se prefires:

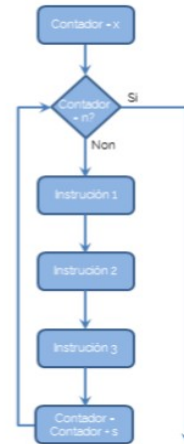
Para contador desde  $x$  ata  $n$  paso  $s$

Instrucións

fin para

### Diagrama de Fluxo

ESTRUTURA REPETITIVA TIPO 3



A instrucción *Para* executa unha secuencia de instrucións un número determinado de veces.

En pseint:

Para variable\_numerica <- valor\_inicial Hasta valor\_final Con Paso paso Hacer  
 secuencia\_de\_acciones

Fin Para

No exemplo do factorial

Algoritmo factorial\_Para

Definir N,F,contador Como Entero

Escribir 'Introduce o número do que queres calcular o factorial'

Leer N

F <- 1

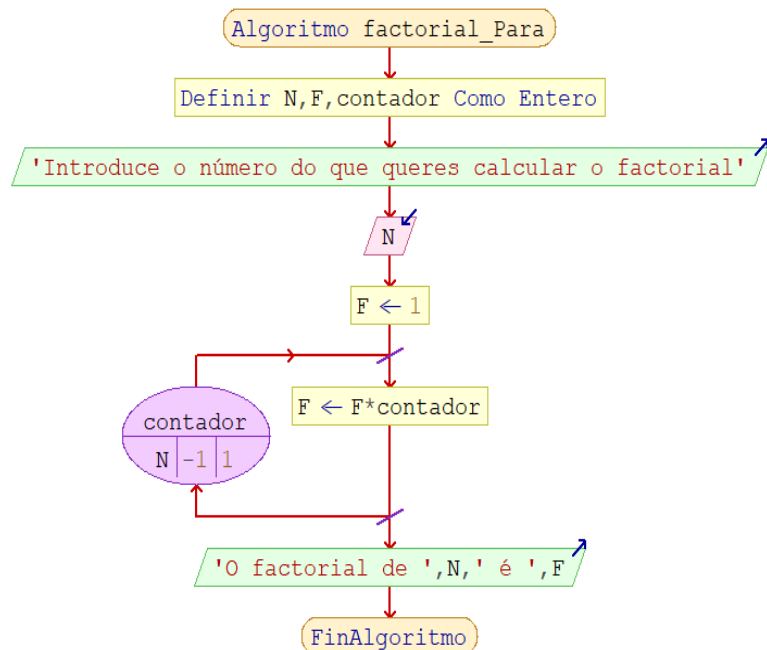
Para contador <- N Hasta 1 Con Paso -1 Hacer

F <- F\*contador

FinPara

Escribir 'O factorial de ',N,' é ',F

FinAlgoritmo



## Operadores e Funcións

As seguintes táboas amosan os principais operadores desta linguaxe:

<i>Operador</i>	<i>Significado</i>	<i>Exemplo</i>
<i>Relacionais</i>		
>	Maior que	3>2
<	Menor que	'ABC'<'abc'
=	Igual que	4=3
<=	Menor ou igual que	'a'<='b'
>=	Maior ou igual que	4>=5
<>	Distinto que	error<>'0'
<i>Lóxicos</i>		
& ó Y	Conxunción (y).	(7>4) & (2=1) //falso
ó O	Disxunción (o).	(1=1   2=1) //verdadero
~ ó NO	Negación (no).	~(2<5) //falso
<i>Alxebraicos</i>		
+	Suma	total <- cant1 + cant2
-	Resta	stock <- disp - venta
*	Multiplicación	area <- base * altura
/	División	porc <- 100 * parte / total
^	Potenciación	sup <- 3.41 * radio ^ 2
% ó MOD	Módulo (resto da división enteira)	resto <- num MOD div

### Observacións:

- Podense introducir comentarios logo dunha instrución, o en liñas separadas, mediante o uso da dobre barra (//). Todo o que segue a //, non será tido en conta ao interpretar o algoritmo.
- Os identificadores, ou nomes de variables, deben comezar sempre cunha letra, e non poden ser palabras reservadas (como para, mientras, y, no, etc...)
- As constantes de tipo carácter escribíense entre comillas ( " ).
- Nas constantes numéricas, o punto ( . ) é o separador decimal.
- As constantes lóxicas son *Verdadeiro* e *Falso*.

## Más programas en Pseudocódigo:

### **//Programa que calcula a suma de N números introducidos por teclado con Repetir**

```
Algoritmo suma_n_numeros_repetir
  Definir n como entero
  Definir s,a Como Real
  s=0
  Escribir "Introduce cantos números quieres sumar: "
  Leer n
  Repetir
    Escribir "Introduce un número: "
    Leer a
    s=s+a
    n=n-1
  Hasta Que n=0
  Escribir "A suma total é: ",s
FinAlgoritmo
```

### **//Programa que calcula a suma de N números introducidos por teclado con Mientras**

```
Algoritmo suma_n_numeros_mientras
  Definir n como entero
  Definir s,a Como Real
  s=0
  Escribir "Introduce cantos números quieres sumar: "
  Leer n
  Mientras n>0 Hacer
    Escribir "Introduce un número: "
    Leer a
    s=s+a
    n=n-1
  Fin Mientras
  Escribir "A suma total é: ",s
FinAlgoritmo
```

### **//Programa que calcula a suma de N números introducidos por teclado con Para**

```
Algoritmo suma_n_numeros_para
  Definir n,i como entero
  Definir s,a Como Real
  s=0
  Escribir "Introduce cantos números quieres sumar: "
  Leer n
  Para i=1 Hasta n Con Paso 1 Hacer
    Escribir "Introduce un número: "
    Leer a
    s=s+a
  Fin Para
  Escribir "A suma total é: ",s
FinAlgoritmo
```

### **Algoritmo táboa\_de\_multiplicar\_con\_mientras**

```
Definir N,i,resultado Como Entero
Escribir "Introduce o número do que queredes facer a táboa de multiplicar"
Leer N
Mientras i<=10 Hacer
    resultado=N*i
    Escribir N "*" i "=" resultado
    i=i+1
Fin Mientras
FinAlgoritmo
```

### **Algoritmo táboa\_de\_multiplicar\_con\_repetir**

```
Definir N,i,resultado Como Entero
Escribir "Introduce o número do que queredes facer a táboa de multiplicar"
Leer N
Repetir
    resultado=N*i
    Escribir N "*" i "=" resultado
    i=i+1
Hasta que i>10
FinAlgoritmo
```

### **Algoritmo táboa\_de\_multiplicar\_con\_para**

```
Definir N,i,resultado Como Entero
Escribir "Introduce o número do que queredes facer a táboa de multiplicar"
Leer N
Para i<-0 Hasta 10 Con Paso 1 Hacer
    resultado=N*i
    escribir N "*" i "=" resultado
Fin Para
FinAlgoritmo
```

### **Algoritmo máximo\_de\_N\_números**

```
Definir N,A,max como entero
Escribir "Introduce o número de números dos que queres calcular o máximo"
Leer N
Escribir "Introduce un número"
Leer A
max=A
N=N-1
Mientras N>0 Hacer
    Escribir "Introduce outro número"
    Leer A
    Si A> max Entonces
        max=A
    Fin Si
    N=N-1
Fin Mientras
Escribir "O máximo dos números introducidos é: ",max
FinAlgoritmo
```

## //Programa que nos din se un número é ou non primo

Algoritmo primos

```
Definir n,d Como Entero
Escribir "Que número queres saber se é primo: "
Leer n
d=1
Repetir
    d=d+1
Hasta Que n%d=0
Si n=d Entonces
    Escribir n, " é un número primo"
SiNo
    Escribir n, " non é un número primo"
Fin Si
FinAlgoritmo
```

## Algoritmo Fibonacci

```
Definir N,A,B,S Como Entero
A=1
B=1
Escribir "Que termo da serie de Fibonacci queres calcular: "
Leer N
Si N=1 | N=2 Entonces
    Escribir "O valor do termo de fibonacci é: 1"
SiNo
    Repetir
        S=A+B
        A=B
        B=S
        N=N-1
    Hasta Que N=2
    Escribir " O valor do termo de fibonacci é: " S
FinSi
FinAlgoritmo
```