

PROGRAMACIÓN ESTRUCTURADA EN FORTRAN

As mulleres na programación e na informática

set ENJOY SAFER TECHNOLOGY™

Actriz de Hollywood, inventora de la tecnología precursora del Wifi, Bluetooth y GPS.

Hedy Lamarr

La primera programadora y madre de la programación informática.

Ada Lovelace

Pionera en la automatización de tareas paralelas. En 2007, recibió el premio Turing, equivalente al Nobel de Informática.

Frances E. Allen

Creadora del ciberpunk, programadora, escritora, belde, defensora de los ciberderechos.

Jude Milhon

Sin las mujeres, la informática no existiría tal como la conocemos

Desarrolló el primer compilador para un lenguaje de programación.

Grace Murray H.

Inventó en 1953 el ordenador de oficina. Desarrolló el primer sistema de reserva de vuelos. Conocida como la madre de los procesadores de texto.

Evelyn Berezin

Seis especialistas en matemáticas que, en 1946, programaron el primer computador ENIAC.

Top Secret Rosies

Pionera en el campo de diseño de chips microelectrónicos.

Lynn Conway

Proceso de programación en Fortran

- Escribe o programa fonte *programa.f90* no **entorno** *Visual Code*
- Compilación do programa dende a **terminal** do VSCode:



A rectangular box with a thin black border containing the text `gfortran programa.f90`. An arrow points from the right side of the box back to the list of steps, specifically towards the 'Compilación do programa' step.

```
gfortran programa.f90
```

- Corrección dos erros de compilación
- Creación de programa executábel: *a.exe*
- Execución de programa na terminal: *.\a.exe*
- Corrección de erros de execución e lóxicos (edición, compilación, execución)
- Se executas *gfortran programa.f90 -o programa.exe*, o executable chamarase *programa.exe*
- **Coidado:** non poñas *programa.f90* no canto de *programa.exe*: se o fas, sobrescribes o *programa.f90* co executable

Programa básico en Fortran

- Resumo do proceso: *programa.f90*  Compilador
gfortran  *a.exe* 

```
program proba  
print *, 'ola!'  
end program proba
```

- Comeza con sentenza: “*program nome*”
- Remata con “*end program nome*”
- “*nome*” é o nome do programa: non pode haber variábeis nin subprogramas con ese nome
- Logo de *program*:
 - declaración de variábeis (ao principio)
 - sentenzas executábeis: operacións, entrada / saída, ...

Programa básico en Fortran

- Sentenza “*stop*”: remata a execución:
- *stop ‘mensaxe’*: remata e imprime a mensaxe
- Exemplo de programa básico:

program basico

integer :: x

x=5

*print *, x*

end program basico

Imprime o
nº 5 na
terminal de
comandos



- Liñas de comentarios:
con ! ao comezo da liña

Programación en Fortran

```
File Edit Selection View ... programas
Welcome basico.f90 x
basico.f90 > {} basico
1 program basico
2 integer :: x
3 x=5
4 print *, x
5 end program basico

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
delgado@ctdesk209:~/docencia/matematicas/material/fortran/interactivas
• /programas$ gfortran basico.f90
delgado@ctdesk209:~/docencia/matematicas/material/fortran/interactivas
• /programas$ a.exe
5
delgado@ctdesk209:~/docencia/matematicas/material/fortran/interactivas
• /programas$
```

Entrada e saída estándar

- Entrada de datos estándar: le datos dende o teclado e almacénaos en variábeis
 - Sentenza *read*: *read *, x*
 - Pódese producir un erro se a variábel é numérica e introducimos un carácter (p.ex.)
- Saída de datos estándar: visualiza na terminal (pantalla)
 - Sentenza *print*: *print *, 'resultado=', 2*x*
 - Con formato: *print '(“n=",i0)',n*
print '(“x=",f10.6)',x
 - ← Ancho do nº enteiro
 - ← Ancho e nº de decimais

Algoritmo e diagrama de fluxo

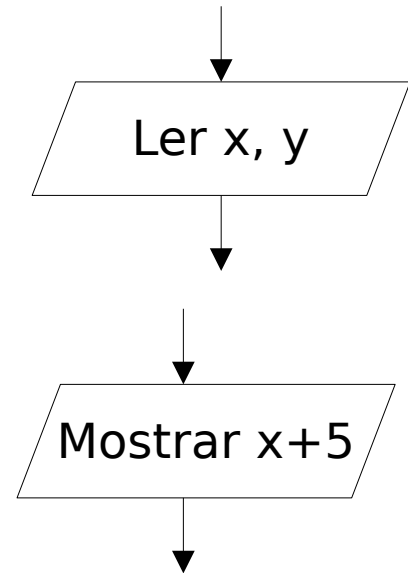
Algoritmo: método para a resolución dun problema, detallado completamente en tódolos seus pasos.

- **Entrada** de datos: dende teclado ou arquivos
- **Procesamento:** operacións cos datos
- **Saída** de resultados: por pantalla ou a arquivos

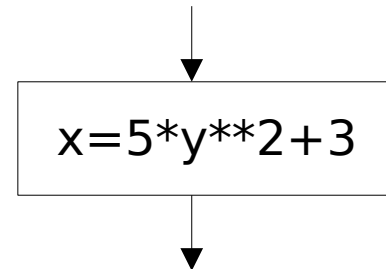
Diagrama de fluxo:

- Forma de representar gráficamente un algoritmo como unha secuencia de operacións
- Diagrama de caixas (operacións) e flechas que as conectan (orden de execución).

Lectura de
datos por
teclado
ou escritura
de datos
na terminal:



Operación
aritmética
ou asignación:



Estructura de selección básica

- IF/ELSE: avalía unha serie de condicións e executa unhas sentenzas ou outras dependendo de que condicións se cumpren

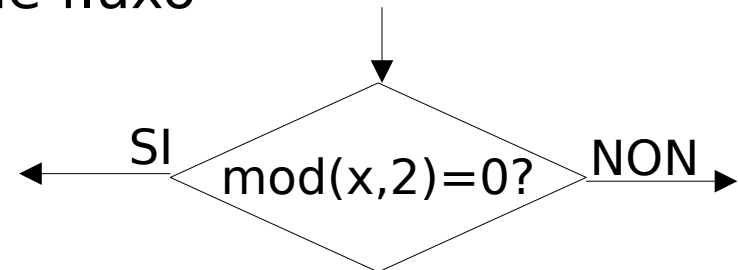
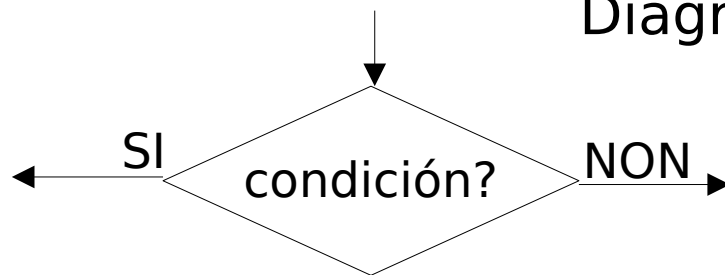
Sintaxe:

```
if(condicion) then
    sentenzas1
else if(condición2) then
    sentenzas2
else
    sentenzas3
endif
```

Exemplo:

```
if (x > 3) then
    print *, 'alto'
else
    print *, 'baixo'
end if
```

Diagrama de fluxo



Estrutura iterativa *do* definida

- Permite repetir unha ou varias sentenzas un certo número de veces

Sintaxe:

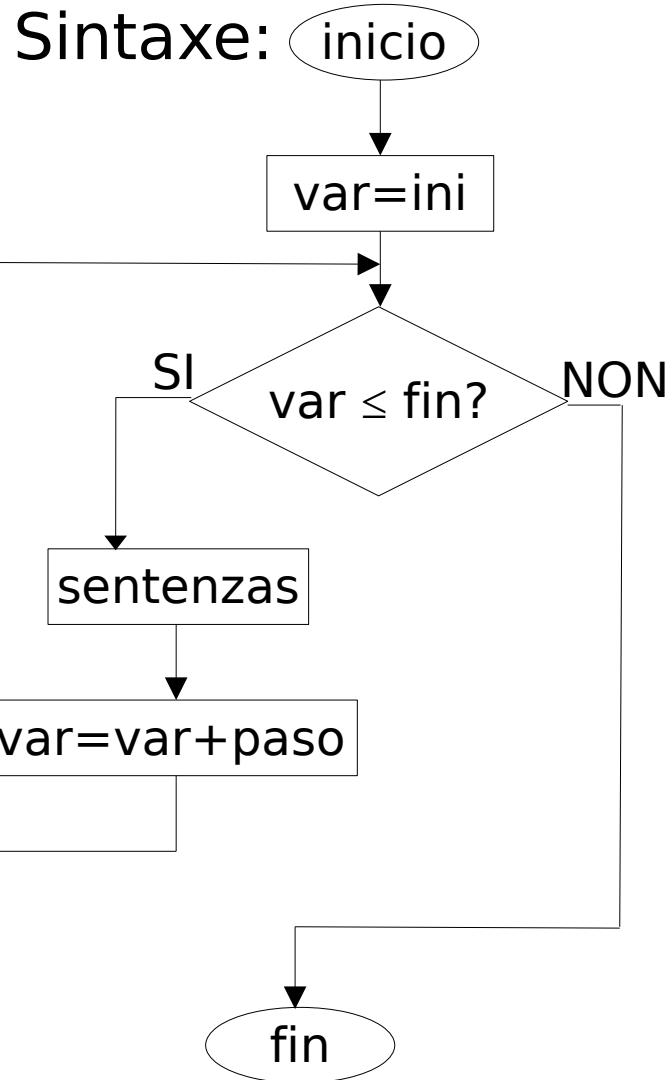
```
do var=ini, fin, paso  
  sentenzas  
end do
```

Exemplo:

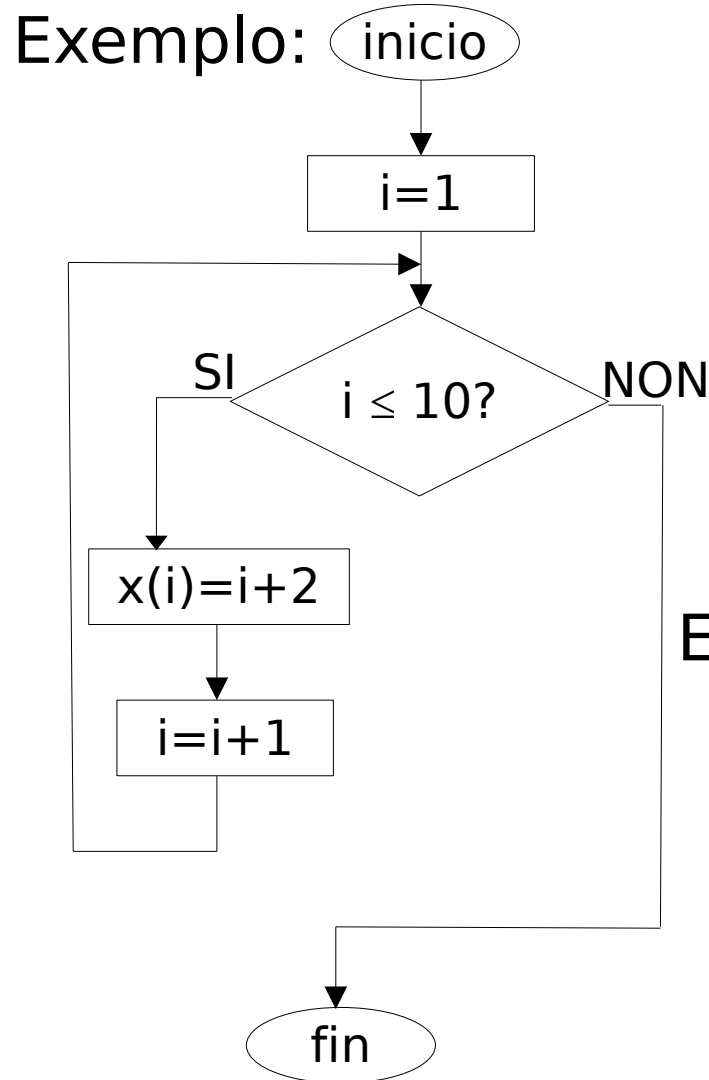
```
do i=1, 10  
  print *,i+2  
end do
```

- Repite as sentenzas dende que a variábel *var* adopta o valor *ini*, ata que adopta o valor *fin*.
- En cada repetición, *var* incrementábase en *paso*
- O incremento *paso* vale 1 por defecto

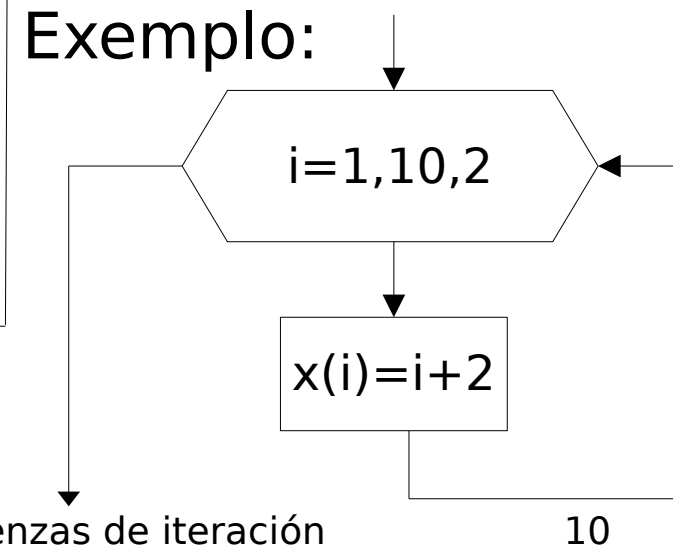
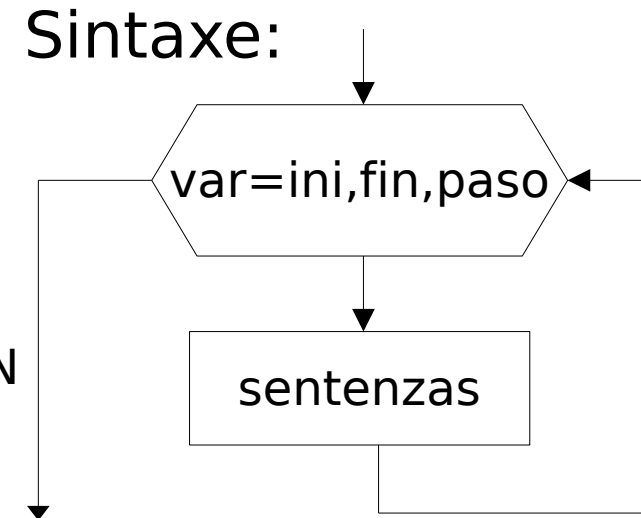
Diagrama de flujo *do* definido



Programación en Fortran



Versión simplificada:



Acumulador

- Variable que aparece na esquerda e na dereita dunha asignación dentro dun bucle *do*: Exemplo: $x=x+i$; $p=p*j$
- Emprégase en operacións cun número variábel de operandos, que require un bucle *do*.
- O acumulador debe inicializarse co elemento neutro da operación.
- Acumulador de suma e de produto dun número variábel (n) de valores:

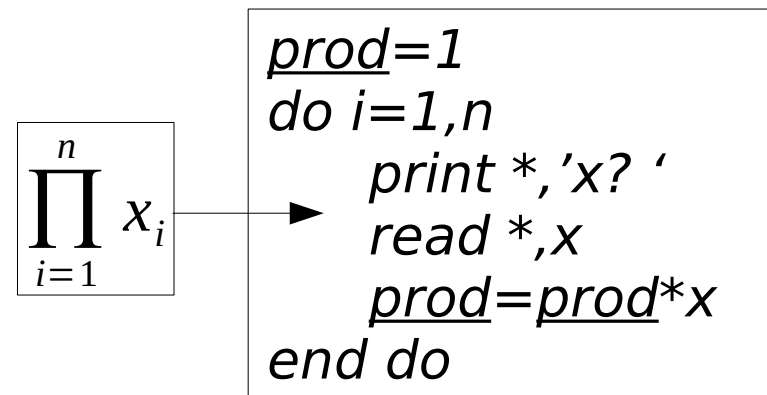
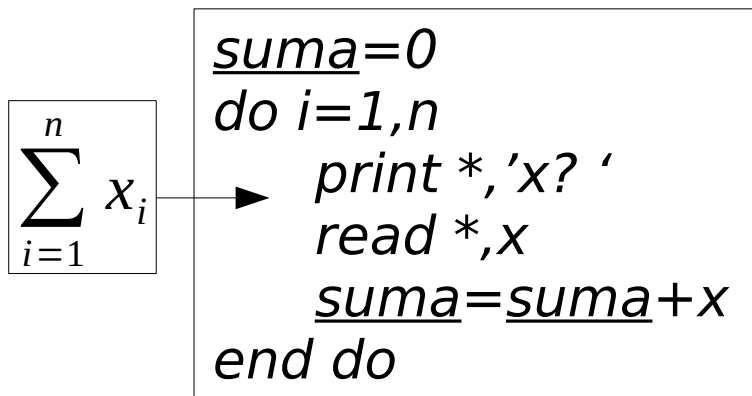
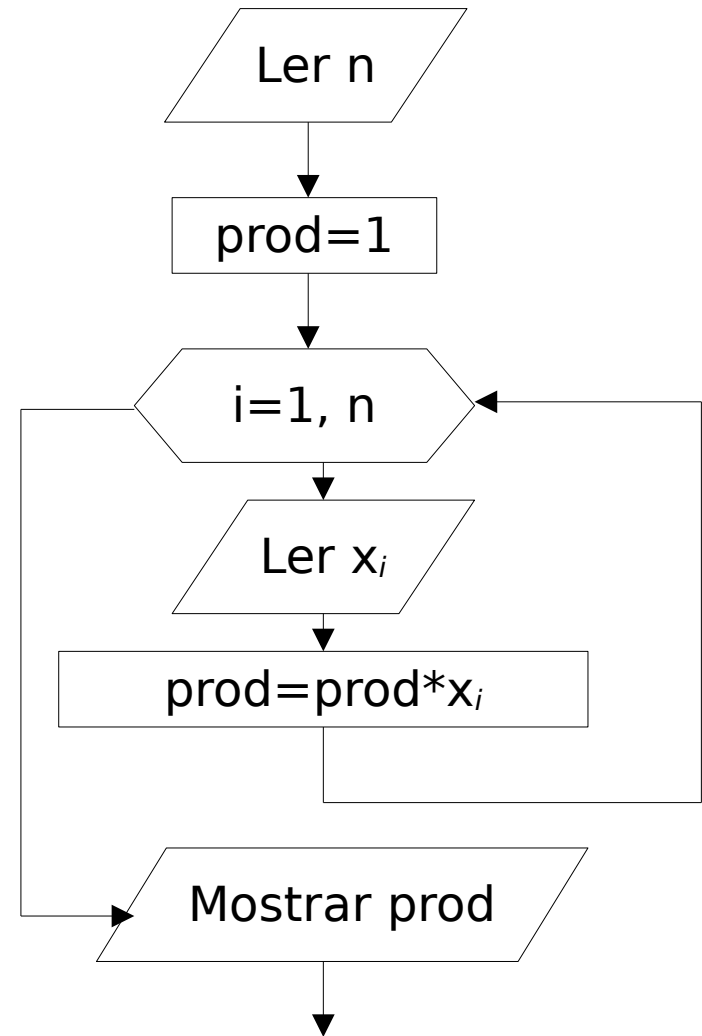
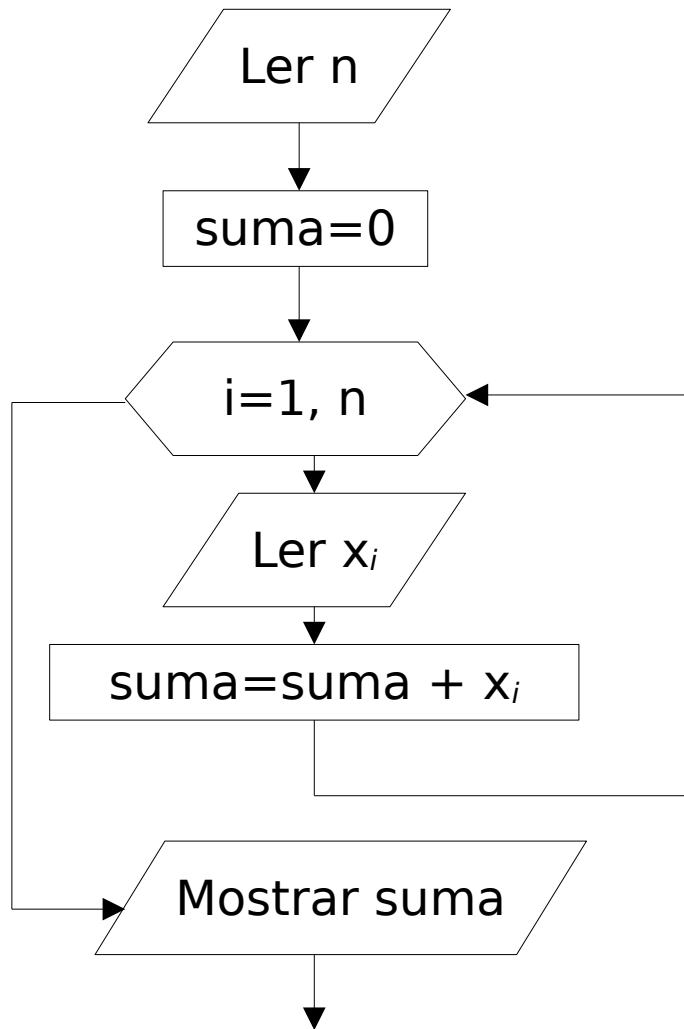


Diagrama de flujo de suma e producto



Estructuras iterativas *do* indefinidas

- Bucle *do/exit*: repite las sentencias e remata cuando se cumple una condición.

```
do
  sentencias
  if(cond) exit
  sentencias
end do
```

```
x=0;dx=0.1
do
  x=x+dx
  print *,x,x*x
  if(x>1) exit
end do
```

```
x=0;dx=0.1
do
  if(x>1) exit
  x=x+dx
  print *,x,sin(x)
end do
```

- Bucle *do while*: repite mientras se cumple la condición.

```
do while(condición)
  sentencias
end do
```

```
x=0;dx=0.1
do while(x<1)
  print *,x,x*x
  x=x+dx
end do
```

