



Informática

Control de flujo

Dpto. Matemática Aplicada a la Ingeniería Aeroespacial

curso 2017-2018



1 Introducción

2 Estructura Condicional



1 Introducción

2 Estructura Condicional



Estructura de un programa



```
program main
implicit none
    ! Declaración de datos
real :: x
real :: y

    ! Definición de datos
x = 2.5

    ! Sentencias de ejecución
y = x**0.5

    ! Visualización de resultados
write(*,*) ' La raíz cuadrada de', x, 'es', y
end program main
```



Flujo secuencial





Estructuras de alteración de flujo

- **Estructura condicional**

Una estructura condicional es aquella que ejecuta un conjunto u otro de sentencias, dependiendo de si se cumple o no una determinada condición.

- **Estructura iterativa**

Una estructura iterativa (*bucle*) es aquella que ejecuta repetidas veces un conjunto de sentencias (*rango del bucle*).



1 Introducción

2 Estructura Condicional



Estructuras de alteración de flujo

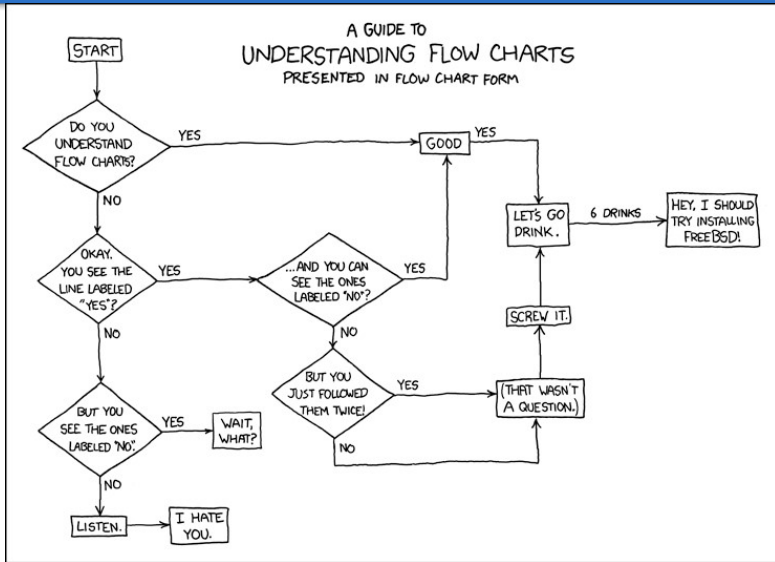
- **Estructura condicional**

Una estructura condicional es aquella que ejecuta un conjunto u otro de sentencias, dependiendo de si se cumple o no una determinada condición.

- Existen tres tipos de estructuras condicionales:
 - *simples*
 - *dobles*
 - *múltiples*



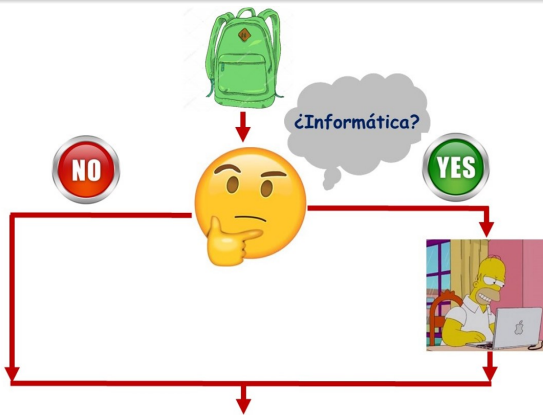
Estructura condicional





Estructura condicional simple

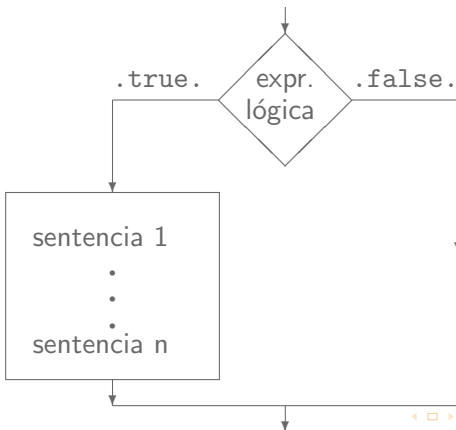
- Evalúa una única condición dada por una expresión lógica escalar.





Estructura condicional simple

- Evalúa una única condición dada por una expresión lógica **escalar**.





Estructura condicional simple

- Evalúa una única condición dada por una expresión lógica **escalar**.

```
if (expr. lógica) then
```

```
    sentencia 1
```

```
    •
```

```
    •
```

```
    •
```

```
    sentencia n
```

```
end if
```



Ejemplo 1

Escribir un programa que, dadas las variables x e y , tras la ejecución, x tenga el valor máximo de los dos e y el mínimo.

```

program condicion
implicit none
real :: x
real :: y
real :: temp

x = 0.0
y = 2.0

if (x < y) then
    temp = x
    x = y
    y = temp
end if

end program condicion
  
```



Estructura condicional simple

- Si solo hay una sentencia de ejecución dentro de la estructura, el lenguaje permite la siguiente variación

`if (expr. lógica) sentencia`



Ejemplo 2

Escribir un programa que calcule y escriba por pantalla el valor absoluto de un número real.

```
program valor_absoluto
implicit none
real :: x

x = -2.0

if (x < 0.0) x = -x

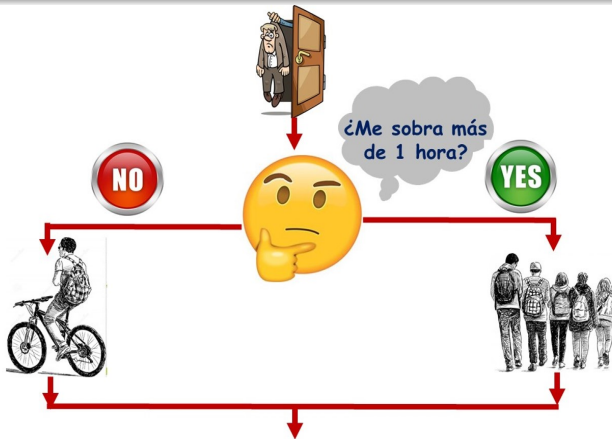
write(*,*) 'Valor absoluto de x:', x

end program valor_absoluto
```



Estructura condicional doble

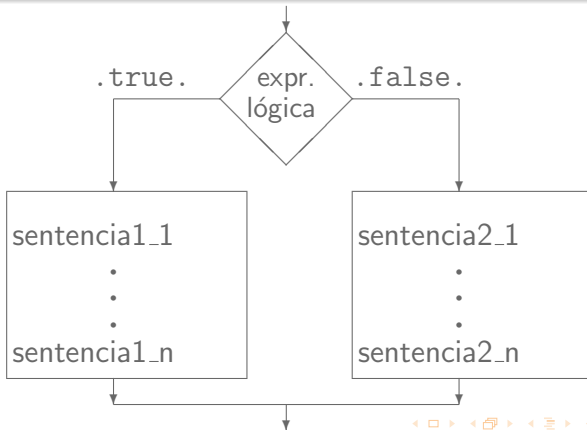
- Evalúa una única condición dada por una expresión lógica escalar.





Estructura condicional doble

- Evalúa una única condición dada por una expresión lógica **escalar**.





Estructura condicional doble

- Evalúa una única condición dada por una expresión lógica **escalar**.

```
if (expr. lógica) then
    sentencia1_1
    .
    .
    sentencia1_n
else
    sentencia2_1
    .
    .
    sentencia2_n
end if
```



Ejemplo 3

Escribir un programa que, dado un número real x , calcule \sqrt{x} . En el caso en que no sea posible, que lo indique por pantalla.

```

program raiz_cuadrada
implicit none
real :: x

write(*,*) 'Introduzca un real'
read(*,*) x

if (x > 0.0) then
    write(*,*) 'Raiz cuadrada:', x**0.5
else
    write(*,*) 'Debe ser positivo'
end if

end program raiz_cuadrada

```

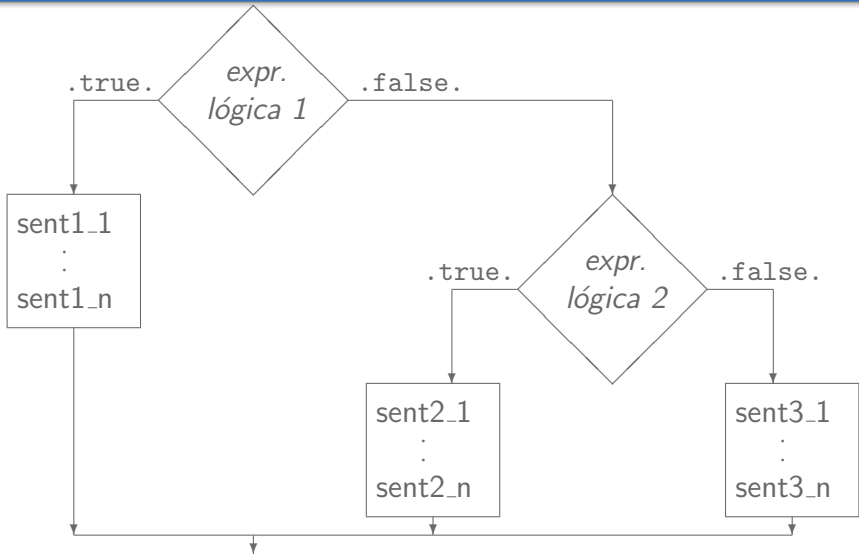


Estructura condicional múltiple

- Construcción `if`
 - Se evalúan diversas codiciones, todas ellas dadas por una expresión lógica **escalar**
- Construcción `select`
 - Se evalúa el valor de un dato simple de tipo `integer`, `character` o `logical`



Estructura condicional múltiple





```
if (expr. lógica 1) then
    sentencia1_1
    :
    sentencia1_n
else if (expr. lógica 2) then
    sentencia2_1
    :
    sentencia2_n
:
[else
    sentenciaN_1
    :
    sentenciaN_n]
end if
```



Ejemplo 4

Escribir un programa que calcule las raíces de:

$$ax^2 + 2bx + c = 0$$

```
program raices
implicit none
real    :: a, b, c
real    :: disc
real    :: x1, x2
complex :: z1, z2

write(*,*) 'Introducir a, b y c'
read(*,*) a, b, c

disc = b*b - 4.0*a*b
```



Ejemplo 04

Escribir un programa que calcule las raíces de:

$$ax^2 + 2bx + c = 0$$

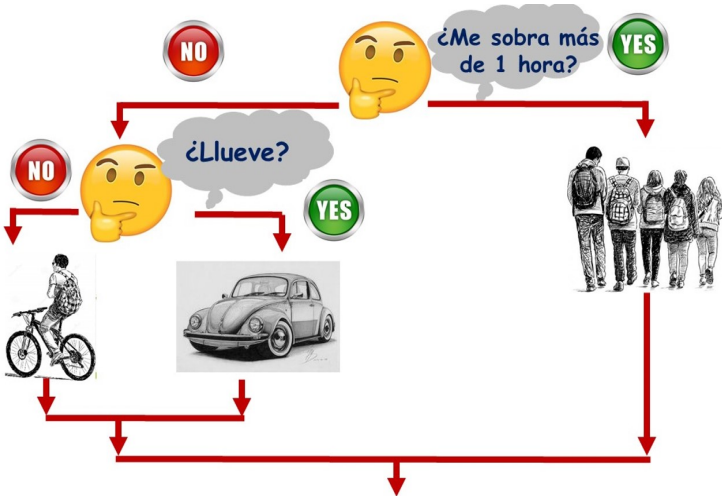
```

if (disc > 0.0) then
    x1 = (-b + disc**0.5)/(2.0*a)
    x2 = (-b - disc**0.5)/(2.0*a)
else if (disc > 0.0) then
    z1 = cmplx(-b, (-disc)**0.5)/(2.0*a)
    z2 = cmplx(-b, -(-disc)**0.5)/(2.0*a)
else
    x1 = -b/(2.0*a)
    x2 = x1
end if

end program raices|
  
```




Estructura condicional anidada





```
if (expr. lógica 1) then
    if (expr. lógica 1.1) then
        sentencia 1.1_1
        :
        sentencia 1.1_n
    else
        sentencia 1.2_1
        :
        sentencia 1.2_n
    end if
else if (expr. lógica 2) then
    sentencia2_1
    :
    sentencia2_n]
end if
```



Ejemplo 4

Escribir un programa que calcule las raíces de:

$$ax^2 + 2bx + c = 0$$

```

if ((disc > 0.0) .or. (disc < 0.0)) then
  if (disc > 0.0) then
    x1 = (-b + disc**0.5)/(2.0*a)
    x2 = (-b - disc**0.5)/(2.0*a)
  else
    z1 = cmplx(-b, (-disc)**0.5)/(2.0*a)
    z2 = cmplx(-b, -(-disc)**0.5)/(2.0*a)
  end if
else
  x1 = -b/(2.0*a)
  x2 = x1
end if

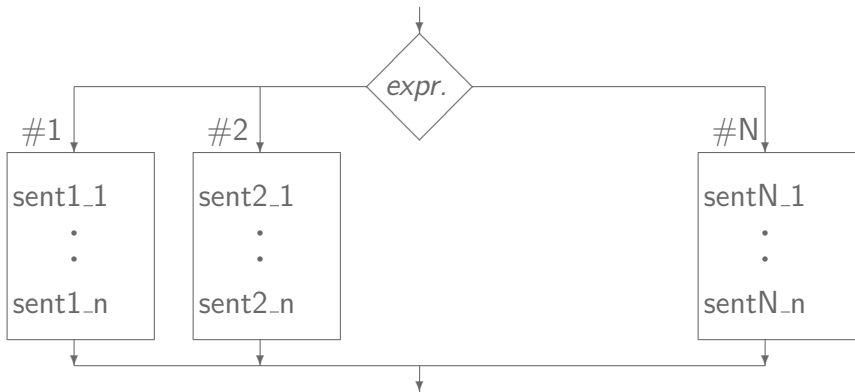
end program raices

```



Estructura condicional múltiple

- Construcción `if`
 - Se evalúan diversas codiciones, todas ellas dadas por una expresión lógica **escalar**
- Construcción `select`
 - Se evalúa el valor de un dato simple de tipo `integer`, `character` o `logical`





Estructura condicional múltiple



```
select case (expr. case)
case (rango de valores #1)
    sentencia1_1
    .
    sentencia1_n
case (rango de valores #2)
    sentencia2_1
    .
    sentencia2_n
.
[case default
    sentenciaN_1
    .
    sentenciaN_n]
end select
```



Construcción select

- *expr. case* es una expresión **escalar** de tipo `integer`, `character` o `logical`
 - Un dato simple: `i`
 - Una expresión aritmética: `i*j+2`



Construcción select

- *rango de valores* puede tomar un único valor, varios valores o un rango de valores

dato	$expr = \text{dato}$
dato_1, dato_2, ..., dato_n	$expr = \text{dato}_1 \text{ ó } \text{dato}_2 \text{ ó } \dots \text{ ó } \text{dato}_n$
	\vdots
	dato_n
dato_1 : dato_2	$\text{dato}_1 \leq expr \leq \text{dato}_2$
: dato_2	$expr \leq \text{dato}_2$
dato_1 :	$\text{dato}_1 \leq expr$



Estructura condicional múltiple



```
select case (num)
case (:-1)
    write(*,*) ' num <= -1 '

case (1, 3, 5, 7, 9)
    write(*,*) ' 1 <= num <= 9 (impar) '

case (2, 4, 6, 8)
    write(*,*) ' 2 <= num <= 8 (par) '

case (0, 10:200)
    write(*,*) ' 10 <= num <= 200 o 0 '

case (205:)
    write(*,*) ' num >= 205 '
case default
    write(*,*) ' 201 <= num <= 204 '
end select
```