

Esquemas repetitivos en Fortran 90

Los esquemas repetitivos permiten implementar iteraciones del lenguaje algorítmico (mientras, repetir, para, hasta).

- Podemos dividir los esquemas según:
 - Se conozcan el número de iteraciones a priori,
 - La acción se realice de cero o más veces.

❖ Tipos de Esquema

1. Esquema DO
2. Esquema DO WHILE
3. Esquema DO + EXIT
4. Esquema DO - CYCLE

Esquemas repetitivos en Fortran 90

1. Esquema DO

Presenta novedades frente al bucle DO de FORTRAN 77 (y sus extensiones):

```

DO [etiqueta [,]] <vble-do>=<inicio>,<fin>[, <inc>]
  <sentencias>
END DO |

DO [etiqueta [, ] ] WHILE (<expr. lógica>)
  <sentencias>
END DO
  
```

Esquemas repetitivos en Fortran 90

1. Esquema DO

1.1 Sintaxis:

```

DO <variable> = <inicio>, <fin> [, <inc>]
  <sentencias>
END DO
  
```

1.2 Semántica

- Equivalente al esquema para
- <variable> entera, se conoce como índice
- Inicialmente se asigna el valor <inicio> a la <variable>
- Se comprueba que <variable> no haya llegado al valor <fin>
- El paso de la iteración viene dado por <inc>
- Si <inc> no se incluye, se supone que es 1
- No deberían modificarse los valores <variable>, <inicio>, <fin> ni <inc> dentro del bucle

Esquemas repetitivos en Fortran 90

1.3 Ejemplos

```

program prueba_do
  implicit none
  integer::i, j
  write(*,*) "Escribo cuadrados perfectos hasta 20"
  do i=1, 20, 1
    write(*,*) i, i**2
  end do
  stop
end program prueba_do

```

Esquemas repetitivos en Fortran 90

```

program prueba_do2
  implicit none
  integer::i, j
  write(*,*) "Escribo caracteres ASCII en orden inverso"
  read(*,*)
  j = 0
  do i=255,1,-1
    write(*,*) i, char(i)
    j = j + 1
    if (j == 24) then ! Sería lo mismo que (mod(255 - i + 1, 24) == 0)
      j = 0
      read(*,*)
    end if
  end do
  stop
end program prueba_do2

```

Esquemas repetitivos en Fortran 90

2. ESQUEMA DO WHILE

2.1. Sintaxis

```

DO WHILE (<expr. lógica>)
  <sentencias>
END DO

```

2.2. Semántica

- Es equivalente al bucle "mientras" del lenguaje algorítmico
- No se sabe a priori el número de iteraciones
- Sólo si <expr. lógica> es cierta, se ejecuta el loop
- Cada vez que se alcanza END DO vuelve a evaluarse la expresión lógica
- El bucle puede ejecutarse de cero o infinitas veces
- Se mantiene por compatibilidad con otros lenguajes

Esquemas repetitivos en Fortran 90

2.3 Ejemplos

```

program cuentopipa
implicit none
character (len=1):: entrada

write(*," ¿Deseas saber sobre el cuento de la buena pipa? (s/n)"
read(*,") entrada
do while ((entrada == "s") .OR. (entrada == "n"))
write(*," Te pregunté sobre el cuento de la buena Pipa... (s/n)"
  read(*,") entrada
end do

stop
end program cuentopipa
  
```

Esquemas repetitivos en Fortran 90

3. ESQUEMA DO + EXIT

3.1. Sintaxis

```

DO
  <sentencias-1>
  IF (<expr. lógica>) EXIT
  <sentencias-2>
END DO

  DO
    IF (<expr. lógica>) EXIT
    <sentencias-2>
  END DO

  DO
    <sentencias-1>
    IF (<expr. lógica>) EXIT
  END DO
  
```

Esquemas repetitivos en Fortran 90

3. ESQUEMA DO + EXIT

3.2 Semántica

- Equivalente al esquema general de repeticiones (iterarsalirsi)
- También, implementa los esquemas mientras y repetir
- Puede utilizarse en lugar del bucle DO-WHILE-END DO
- Si existen, se ejecutan el conjunto de <sentencias-1>
- Después se comprueba si la <expr. lógica> es cierta:
 - Si es cierta en ese caso se sale del bucle (avanza a la instrucción tras END DO)
 - Si es falsa, se ejecutan las <sentencias-2> y se vuelve a entrar al DO

Esquemas repetitivos en Fortran 90

3.3 Ejemplos

```

program suma1
implicit none
integer::i, suma, n
suma = 0
i = 0

write(*,*) "Cuantos números desea sumar?"
read (*, *) n
do while (i <= n )
    suma = suma + i
    i = i + 1
end do

write(*,*) "La suma de ", n, " enteros es ", suma

stop
end program suma1
    
```

Esquemas repetitivos en Fortran 90

3.3 Ejemplos

```

program suma2
implicit none
integer::i, suma, n
suma = 0
i = 0
! Comprueba que los límites sean correctos = repetir
do
    write(*,*) "Cuantos números quieres sumar (max. 100)?"
    read (*, *) n
    if ((n >= 0) .and. (n <= 100)) exit
end do
! Equivalente al do while = mientras
do
    if (i > n) exit
    suma = suma + i
    i = i + 1
end do
write(*,*) "La suma de ", n, " enteros es ", suma

stop
end program suma2
    
```

Esquemas repetitivos en Fortran 90

4. DO CYCLE

4.1. Sintaxis

```

IF (<exp. lógica>) CYCLE [NOMBRE]
    
```

4.2. Semántica

- Puede utilizarse dentro de los esquemas DO – END DO
- Si se cumple la expresión lógica, se pasa directamente al END DO y se vuelve al inicio: DO o DO WHILE.
- Puede llevar asociado el nombre del bucle DO asociado.
- Su uso no es necesario, disponiendo de IF-THEN-ELSE y variables lógicas adecuadas NO DEBE USARSE.

Esquemas repetitivos en Fortran 90

4.3 Ejemplo

```
fuera: do i=1,n
  medio: do j=1,m
    dentro: do k= 1, l
      read (*,*) x
      if (x < 0.0) exit fuera
      if (j == 5) cycle medio
      if (K == 7) cycle fuera
    end do dentro
  end do medio
end do fuera
```
