

Programación básica

PRÁCTICA 9. FORTRAN ESTRUCTURAS DE REPETICIÓN

¿Dónde estamos en el ciclo de vida del software?



Objetivo de la práctica

- ▶ Elaborar programas en FORTRAN para la resolución de problemas básicos que incluyan estructuras de repetición.
- ▶ Actividades
 - ▶ Crear programa con las estructuras `while` y `do - while` .
 - ▶ Anidar estructuras de selección dentro de estructuras repetitivas .

Estructura de control repetitiva DO WHILE

La estructura repetitiva (o iterativa)) *do - while* primero valida la expresión lógica y si ésta se cumple (es verdadera)) procede a ejecutar el bloque de instrucciones de la estructura. Si la condición no se cumple se continúa el flujo normal del programa sin ejecutar el bloque de la estructura, es decir, el bloque se puede ejecutar de cero a ene veces.



Comparación con el pseudocódigo

Mientras. Do While

Mientras expresion_logica Hacer
 secuencia_de_acciones
Fin Mientras

```
program tablaMultiplicar

c Este programa genera la tabla de multiplicar de un número dado.
c El número se lee desde la entrada estándar (teclado).

integer num, cont

cont = 0

write (*,*) '----- Tabla de multiplicar -----'
write (*,*) 'Ingrese un número:'
read (*,*) num

write (*,*) 'La tabla de multiplicar del', num, 'es'

do while (cont .LT. 10)
    cont = cont + 1
    write (*,*) num, 'x', cont, '=', num*cont
enddo

stop
end
```

Estructura de control de repetición DO

Lenguaje FORTRAN posee la estructura de repetición **do** la cual permite realizar repeticiones cuando se conoce el número de elementos que se quiere recorrer

```
do var = inicio, fin[, incremento]
```

```
!Bloque de código  
!a ejecutar
```

```
enddo
```

Por ejemplo, la cuenta regresiva de un
cohete
o antes de iniciar una película



```
program doConIncremento
```

```
c Este programa debe obtener el promedio de 5 calificaciones  
c ingresadas por el usuario. Las calificaciones se leen  
c desde la entrada estándar (teclado).
```

```
real*8 sum, calif  
integer veces, cont
```

```
sum = 0  
cont = 1
```

```
do veces = 1, 10, 2  
  write (*,*) 'Suma de calificaciones'  
  write (*,*) 'Ingrese la calificación', cont  
  read (*,*) calif  
  sum = sum + calif  
  cont = cont + 1  
end do
```

```
write (*,*) 'El promedio de las calificaciones es:'  
write (*,*) sum/(cont-1)
```

```
stop  
end
```

Escapar de todo el ciclo EXIT

EXIT produce la salida inmediata del ciclo

```
do
    !Bloque de código
    !a ejecutar

    IF (expresión_lógica) EXIT

    !Bloque de código
    !a ejecutar
end do
```



Escapar de una vuelta (iteración) del ciclo CYCLE

CYCLE detiene la ejecución de la iteración actual y devuelve el control al inicio del ciclo, continuando la ejecución de la iteración siguiente del ciclo.

```
PROGRAM sumaPares

c Programa que suma los primeros 5 números pares a partir de un número dado.

INTEGER numero, suma, contador
suma = 0
contador = 1

WRITE (*,*) 'Ingrese el número inicial'
READ (*,*) numero

DO WHILE (contador .LE. 5)
    numero = numero + 1
c si el número es par se devuelve el control al inicio del ciclo,
c es decir, no se ejecutan las líneas que están debajo de la estructura IF
    IF (MOD(numero, 2) .EQ. 1) CYCLE
    contador = contador + 1
    suma = suma + numero
END DO

WRITE (*,*) 'La suma de los números pares es: ', suma

STOP
END
```

Actividades a desarrollar práctica (Individual)

Primera firma, se tiene 30 minutos. Al terminar formarse al frente para revisión:

1. Compilar y ejecutar el programa “Factura” que se pidió tener listo como previo de la práctica. El programa debe tener la funcionalidad de preguntar si quiere calcular otra factura y no salir hasta que el usuario diga que no. (USO DE DO WHILE)
2. Escribir el código de los programas del previo. Compilarlos y ejecutarlos. Hacer las pruebas necesarias para validar su funcionamiento.
 - a) “Promedio de calificaciones” Pag. 3.
 - b) “Calculadora” Pag. 4

Guardar sus programas los vamos a seguir ocupando en clase

Actividades a desarrollar práctica (equipos de dos)

Segunda firma, se tiene 30 minutos. Al terminar formarse al frente para revisión:

1. Modificar el programa "Calculadora" para que en caso de que el usuario ingrese en la división **un cero** en el dividendo, el programa suspenda la ejecución totalmente mandando un mensaje de error general. (Es decir salga del ciclo, USANDO EXIT)
2. Modificar el programa "Promedio de calificaciones" para que en **caso de la calificación sea 5** no sea sumada al promedio. (Saltar una iteración, USANDO CYCLE)

Guardar sus programas los vamos a seguir ocupando en clase