



Física Computacional

Escuela de Física

M.R.Fulla¹

¹Escuela de Física, Universidad Nacional de Colombia Sede
Medellín

marlonfulla@yahoo.com- Oficina:21-408

<https://sites.google.com/view/fiscomunalmed/>

August 28, 2023

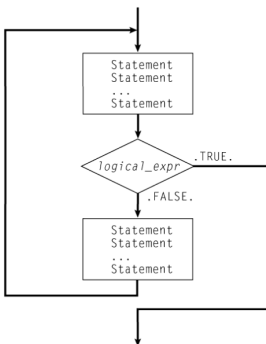
Estructura de Repetición DO (while)



UNIVERSIDAD
NACIONAL
DE COLOMBIA

```
DO  
...  
IF (logical_expr) EXIT  
...  
END DO
```

} Code Block



```
DO WHILE (logical_expr)  
...  
...  
...  
END DO
```

} Statement 1
Statement 2
...
Statement n

Alternativa

DO Iterativo

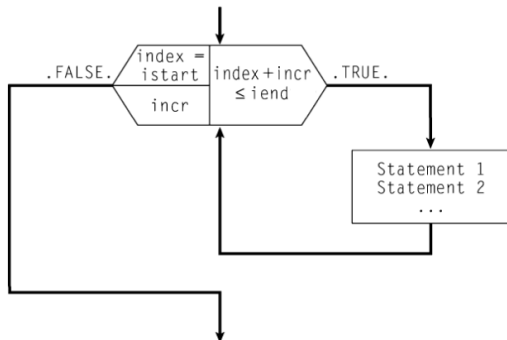
```
DO index = istart, iend, incr  
  Statement 1  
  ...  
  Statement n  
END DO
```

} Code Block



Sangría

$$\text{iter} = \frac{\text{iend} - \text{istart} + \text{incr}}{\text{incr}}$$



Sugerencia: Nunca modificar el valor del índice dentro del ciclo DO.

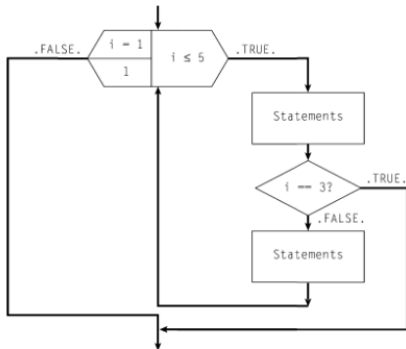
Instrucciones de Salto CYCLE y EXIT



UNIVERSIDAD
NACIONAL
DE COLOMBIA

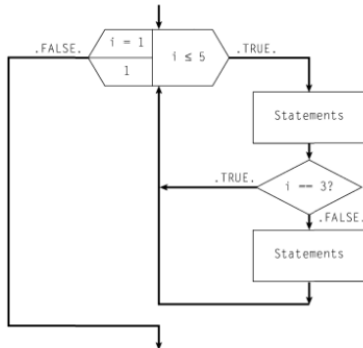
```
PROGRAM test_exit
INTEGER :: i
DO i = 1, 5
  IF ( i == 3 ) EXIT
  WRITE (*,*) i
END DO
WRITE (*,*) 'End of loop!'
END PROGRAM test_exit
```

```
C:\book\chap4>test_exit
1
2
End of loop!
```



```
PROGRAM test_cycle
INTEGER :: i
DO i = 1, 5
  IF ( i == 3 ) CYCLE
  WRITE (*,*) i
END DO
WRITE (*,*) 'End of loop!'
END PROGRAM test_cycle
```

```
C:\book\chap4>test_cycle
1
2
4
5
End of loop!
```



Ciclos DO Anidados

```
PROGRAM nested_loops
INTEGER :: i, j, product
DO i = 1, 3
    DO j = 1, 3
        product = i * j
        WRITE (*,*) i, ' * ', j, ' = ', product
    END DO
END DO
END PROGRAM nested_loops
```

1 *	1 =	1
1 *	2 =	2
1 *	3 =	3
2 *	1 =	2
2 *	2 =	4
2 *	3 =	6
3 *	1 =	3
3 *	2 =	6
3 *	3 =	9

```
outer: DO i = 1, 3
    ...
    inner: DO j = 1, 3
        ...
    END DO outer
    ...
END DO inner
```

- ▶ Hacer uso de los diagramas de flujo y/o pseudocódigo para el diseño de algoritmos, especialmente aquellos que involucren estructuras repetitivas y selectivas en un mismo ciclo.
- ▶ Implementar WRITE
- ▶ Chequear condiciones lógicas (escritura de operadores lógicos)
- ▶ Comparaciones entre números reales

IF($x == 1.0$) THEN, una mejor opción es
IF($\text{abs}(x - 1.0) \leq 0.0001$) THEN

Asignación:

CHARACTER(len=int-val):: var

CHARACTER(int-val):: var

Operadores con caracteres $\left\{ \begin{array}{l} \textit{Substring} \\ \textit{Concatenación} \end{array} \right.$

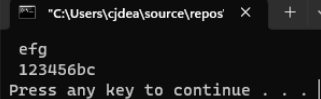
Substring:

CHARACTER(int-val)::char-name

char-name(char-ini:char-end)

Substring ()

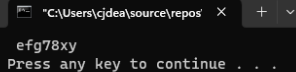
```
8
9      PROGRAM caracteres
10     CHARACTER(8) :: a,b,c
11     a='abcdefghij'
12     b='12345678'
13     c=a(5:7)    !operación substring
14     WRITE(*,*) c
15     b(7:8)=a(2:6)
16     WRITE(*,*) b
17     END PROGRAM caracteres
18
```



```
"C:\Users\cjdea\source\repos" X + v
efg
123456bc
Press any key to continue . . . |
```


Operador Concatenación //

```
9      PROGRAM caracteres
10     CHARACTER(8) :: a,b,c,d
11     a='abcdefghij'
12     b='12345678'
13     d=a(5:7)//b(7:8)//'xy'  !operación de concatenación
14     WRITE(*,*) d
15     END PROGRAM caracteres
16
```



"C:\Users\cjdea\source\repos" × + ▾

efg78xy
Press any key to continue . . .

Operadores Relacionales con Caracteres

```
9      PROGRAM caracteres
10     WRITE(*,*) '1:', 'A'<'B'
11     WRITE(*,*) '2:', 'a'<'B'
12     WRITE(*,*) '3:', 'AAAB'>'AAAA'
13     WRITE(*,*) '4:', 'AAAB'>'AA'
14     WRITE(*,*) '5:', 'AB'>'AAAA'
15     END PROGRAM caracteres
16
17
18
```

"C:\Users\cjdea\source\repos" X + v

```
1: T
2: F
3: T
4: T
5: T
Press any key to continue . . .
```

APPENDIX A: ASCII and EBCDIC Coding Systems

Decimal	Octal	Hex	ASCII Character	EBCDIC Character
55	67	37	7	
56	70	38	8	
57	71	39	9	
58	72	3A	:	
59	73	3B	;	
60	74	3C	<	
61	75	3D	=	
62	76	3E	>	
63	77	3F	?	
64	100	40	@	blank
65	101	41	A	
66	102	42	B	
90	132	5A	Z	!
91	133	5B	[\$
92	134	5C	\	*
93	135	5D])
94	136	5E	^(or ↑)	;
95	137	5F	`	~
96	140	60	`	-
97	141	61	a	/

```
9      PROGRAM caracteres
10     WRITE(*,*) achar(67)
11     WRITE(*,*) iachar('A')
12     WRITE(*,*) len('hello')
13     WRITE(*,*) len_trim('hello  ')
14     WRITE(*,*) trim('hello  ')
15     WRITE(*,*) len(trim('hello  '))
16     END PROGRAM caracteres
```

Output:

```
C
65
5
5
hello
5
Press any key to continue . . .
```

TABLE 4-1

Some common character intrinsic functions

Function name and argument(s)	Comments
ACHAR(ival)	Returns the character corresponding to ival in the ASCII collating sequence
IACHAR(char)	Returns the integer corresponding to char in the ASCII collating sequence
LEN(str1)	Returns length of str1 in characters.
LEN_TRIM(str1)	Returns length of str1, excluding any trailing blanks.
TRIM(str1)	Returns str1 with trailing blanks removed.

- ▶ Escribir un programa que reciba por consola el grado n y las componentes de dos vectores en \mathbb{R}^n , y retorne su producto punto.
- ▶ Escribir un programa que reciba por consola las componentes de dos vectores en \mathbb{R}^3 y retorne su producto cruz.

Nota: tenga en mente todas las buenas prácticas de programación.