



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/>

September 5, 2023

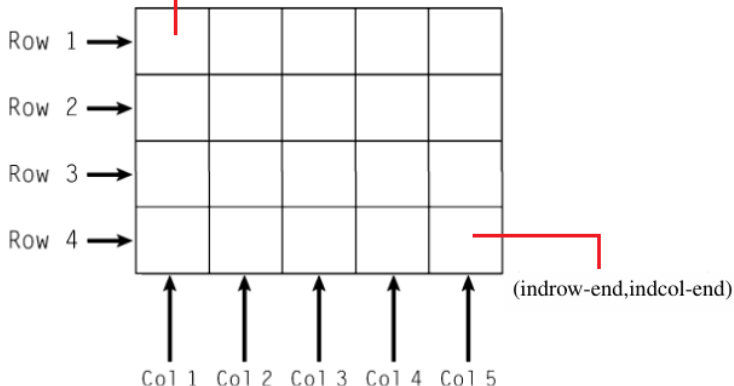
Arreglos de Rango 2 o Bidimensionales

Declaración:

TYPE, DIMENSION(n-row,m-columns)

TYPE, DIMENSION(indrow-ini:indrow-end,indcol-ini:indcol-end)

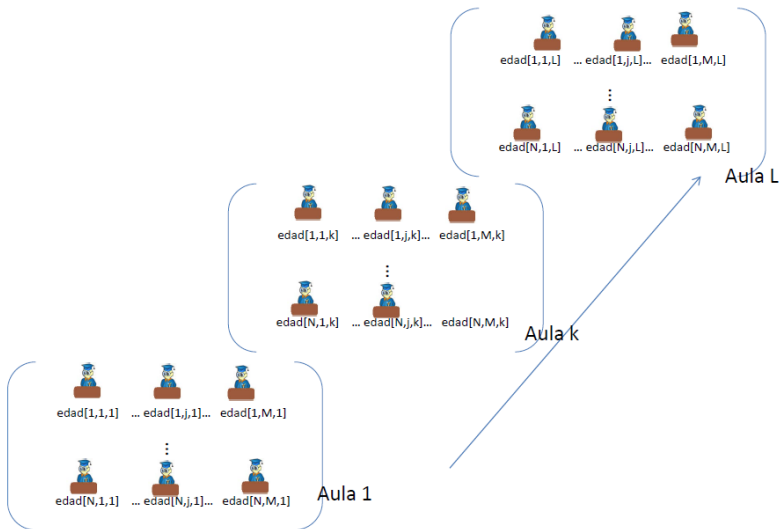
(indrow-ini,indcol-ini)



Arreglos Multidimensionales



UNIVERSIDAD
NACIONAL
DE COLOMBIA



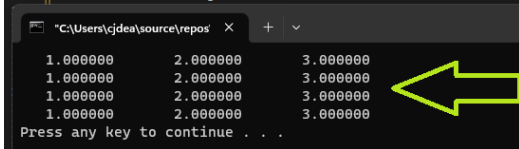
1. Instrucción **READ**

```
9      PROGRAM arreglos_2D
10     IMPLICIT NONE
11     REAL, DIMENSION(10,30)::tabla_multiplicar
12     INTEGER::i,j
13
14     multiplicando : DO i=1,10,1
15
16         multiplicador: DO j=1,10,1
17
18             tabla_multiplicar(i,j)=i*j
19
20         ENDDO multiplicador
21
22     ENDDO multiplicando
23
24     END PROGRAM arreglos_2D
```

2. Instrucciones de Declaración:

RESHAPE(array_index1,array_index2), modifica la forma de un arreglo sin cambiar el número de elementos.

```
9      PROGRAM arreglos_2D
10     IMPLICIT NONE
11     REAL, DIMENSION(4,3)::array
12     REAL, DIMENSION(12)::datos=(/1,1,1,1,2,2,2,2,3,3,3,3/)
13     INTEGER::i,j
14     array=RESHAPE(datos,(/4,3/))
15
16     DO i=1,4,1
17     WRITE(*,*)(array(i,j),j=1,3,1) ! IMPRESION CON DO IMPLICITO
18     ENDDO
19
20     END PROGRAM arreglos_2D
```



| | | |
|----------|----------|----------|
| 1.000000 | 2.000000 | 3.000000 |
| 1.000000 | 2.000000 | 3.000000 |
| 1.000000 | 2.000000 | 3.000000 |
| 1.000000 | 2.000000 | 3.000000 |

Press any key to continue . . .

El cambio de forma mapea los datos en orden descendente por columnas

Dos arreglos se pueden utilizar en operaciones aritméticas conjuntas (elemento a elemento) si son conformables (si tienen la misma forma o uno de ellos es un escalar).

Operaciones con Subconjuntos:

```
9      PROGRAM arreglos_2D
10     IMPLICIT NONE
11     REAL, DIMENSION(4,3)::array
12     REAL, DIMENSION(12)::datos=(/3,4,5,11,0,-3,6,7,8,9,0,3/)
13     REAL, DIMENSION(4)::vector_columna
14     REAL, DIMENSION(3)::vector_fila
15     REAL, DIMENSION(2,2)::matriz
16     INTEGER::i,j
```

Operaciones Conjuntas con Arreglos y Subconjuntos



UNIVERSIDAD
NACIONAL
DE COLOMBIA

```
20 array=RESHAPE(datos,(/4,3/))
21 vector_columna=array(:,2)
22 vector_fila=array(2,:)
23 matriz=array(3:4,2:3)
24
25
26 DO i=1,4,1
27 WRITE(*,*)(array(i,j),j=1,3,1) ! IMPRESION CON DO IMPLICITO
28 ENDDO
29
30 DO i=1,4,1
31 WRITE(*,*) vector_columna(i)
32 ENDDO
33
34 WRITE(*,*) (vector_fila(i),i=1,3,1) ! IMPRESION CON DO IMPLICITO
35
36 DO i=1,2,1
37 WRITE(*,*) (matriz(i,j),j=1,2,1)
38 ENDDO
39
40 END PROGRAM arreglos_2D
```

Output of the program:

| | | |
|----------|--------------|--------------|
| 3.000000 | 0.000000E+00 | 8.000000 |
| 4.000000 | -3.000000 | 9.000000 |
| 5.000000 | 6.000000 | 0.000000E+00 |
| 11.00000 | 7.000000 | 3.000000 |

Output of the program:

| |
|--------------|
| 0.000000E+00 |
| -3.000000 |
| 6.000000 |
| 7.000000 |

Output of the program:

| | | |
|----------|-----------|----------|
| 4.000000 | -3.000000 | 9.000000 |
|----------|-----------|----------|

Output of the program:

| | |
|----------|--------------|
| 6.000000 | 0.000000E+00 |
| 7.000000 | 3.000000 |

Funciones Intrínsecas

1. Funciones intrínsecas elementales: usadas para argumentos escalares y arreglos. Ejemplo: SIN, COS, TAN, EXP, LOG, LOG10, MOD, SQRT.

```
9      PROGRAM arreglos_2D
10     IMPLICIT NONE
11     REAL, DIMENSION(4,3)::array
12     REAL, DIMENSION(12)::datos=(/3,4,5,11,0,-3,6,7,8,9,0,3/)
13     INTEGER::i,j
14
15     array=RESHAPE(datos,(/4,3/))
16     array=ABS(array)
17
18
19     DO i=1,4,1
20     WRITE(*,*)(array(i,j),j=1,3,1) ! IMPRESION CON DO IMPLICITO
21     ENDDO
22
23     END PROGRAM arreglos_2D
```

```
*C:\Users\cjdea\source\repos  X  +  v
3.000000      0.000000E+00      8.000000
4.000000      3.000000      9.000000
5.000000      6.000000      0.000000E+00
11.00000      7.000000      3.000000
Press any key to continue . . .
```


2. Funciones intrínsecas de petición: son funciones cuyo valor depende de las propiedades del objeto que es investigado (Ver Tabla 8.1).

```
9      PROGRAM arreglos_2D
10     IMPLICIT NONE
11     REAL, DIMENSION(4,3)::array
12     REAL, DIMENSION(12)::datos=(/3,4,5,11,0,-3,6,7,8,9,0,3/)
13     INTEGER::i,j
14
15     array=RESHAPE(datos,(/4,3/))
16
17     WRITE(*,*) "shape=", SHAPE(array)
18     WRITE(*,*) "size=", SIZE(array)
19     WRITE(*,*) "lower bounds=", LBOUND(array)
20     WRITE(*,*) "upper bounds=", UBOUND(array)
21
22     END PROGRAM arreglos_2D
```

| "C:\Users\cjdea\source\repos" | | |
|---------------------------------|----|---|
| shape= | 4 | 3 |
| size= | 12 | |
| lower bounds= | 1 | 1 |
| upper bounds= | 4 | 3 |
| Press any key to continue . . . | | |

3. Funciones intrínsecas transformacionales: tienen uno o más arreglos con valores como argumentos. El resultado puede ser un arreglo con forma diferente (ver tabla 8.2).

Ejemplo:

- ▶ *`DOT_PRODUCT(VECTOR_A, VECTOR_B)`*
- ▶ *`MATMUL(MATRIZ_A, MATRIZ_B)`*
- ▶ *`RESHAPE(ARREGLO_FUENTE, ARREGLO1D_FORMA)`*

Cuando se declara el tamaño de un arreglo en la zona de declaraciones del programa unidad, decimos que la asignación de memoria es estática. Pero, ¿cómo modificar su tamaño?

Soluciones { *Modificar tamaño y recompilar*
Colocar tamaños grandes
Diseñar programas de asignación de memoria dinámica

Un Arreglo dinámico contiene el atributo **ALLOCATABLE**:

TYPE, **ALLOCATABLE**, DIMENSION(:,:) :: array_name



arreglo con
forma diferida



Rango esta definido
pero no su tamaño

La determinación de la forma del arreglo se pospone hasta que la memoria del arreglo sea asignada a través de la instrucción **ALLOCATE**.

- ▶ **ALLOCATE**(array1,array2,...,STAT=status) asignación satisfactoria → STAT=0
- ▶ **DEALLOCATE**(array1,array2,...,STAT=status) : Liberar memoria con perdida de datos

Arreglos Dinámicos



UNIVERSIDAD
NACIONAL
DE COLOMBIA

```
9      PROGRAM arreglos_2D
10     IMPLICIT NONE
11     REAL, ALLOCATABLE, DIMENSION(:,)::matriz1,matriz2,matrizsuma
12     REAL::input
13     INTEGER::i,j,m,status
14
15     WRITE(*,*) "Introduzca m : "
16     READ(*,*) m
17     ALLOCATE(matriz1(m,m),matriz2(m,m),matrizsuma(m,m),STAT=status)
18
19     DO i=1,m
20     WRITE(*,*) "Introduzca la fila",i,"de matriz 1"
21     READ(*,*) (matriz1(i,j),j=1,m) !LECTURA CON DO IMPLICITO
22     WRITE(*,*) "Introduzca la fila",i,"de matriz 2"
23     READ(*,*) (matriz2(i,j),j=1,m) !LECTURA CON DO IMPLICITO
24     ENDDO
25
26     matrizsuma=matriz1+matriz2
27     DO i=1,m
28     WRITE(*,*) (matrizsuma(i,j),j=1,m)
29     ENDDO
30
31     END PROGRAM arreglos_2D
```

Construir un programa que solicite a un usuario por consola el rango de dos matrices, sus entradas y retorne su producto.