

# Guía práctica de estudio 11: Arreglos multidimensionales

---



---

***Elaborado por:***

Ing. Jorge A. Solano Gálvez  
Guadalupe Lizeth Parrales Romay

***Revisado por:***

M.C. Edgar E. García Cano

***Autorizado por:***

M.C. Alejandro Velázquez Mena

# Guía práctica de estudio 11:

## Arreglos multidimensionales

### Objetivo:

Elaborar programas en lenguaje FORTRAN para resolver problemas que requieran agrupar conjuntos de datos del mismo tipo en arreglos multidimensionales.

### Actividades:

- Crear arreglos multidimensionales.

### Introducción

Un arreglo es un conjunto de datos contiguos del mismo tipo con un tamaño fijo, definido al momento de crearse. Los arreglos pueden ser unidimensionales (como se vio en la práctica anterior) o multidimensionales y se utilizan para hacer más eficiente el código de un programa.

### Licencia GPL de GNU

El software presente en esta guía práctica es libre bajo la licencia GPL de GNU, es decir, se puede modificar y distribuir mientras se mantenga la licencia GPL.

```
/*
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 *
 * Author: Jorge A. Solano
 */
```

## Arreglos multidimensionales

Lenguaje FORTRAN permite crear arreglos de varias dimensiones con la siguiente sintaxis:

```
tipoDato nombre (tamaño1, tamaño2,...,tamaño7)
```

o con la siguiente sintaxis:

```
tipoDato nombre  
dimension nombre (tamaño1, tamaño2,...,tamaño7)  
tipoDato DIMENSION (d1[, d2...]) :: nombre
```

Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo por dimensión. La palabra reservada DIMENSION permite definir el tamaño máximo de las diferentes dimensiones.

Los tipos de dato que puede tolerar un arreglo multidimensional son: entero, real, carácter o complejo.

De manera práctica se puede considerar que la primera dimensión corresponde a los renglones, la segunda a las columnas, la tercera al plano, y así sucesivamente.

Por defecto, los arreglos inician en la posición 1 y, por tanto, las dimensiones se recorren de la posición 1 a la posición d1 para la primera dimensión, de la posición 1 a la posición d2 en la segunda dimensión, y así sucesivamente. Sin embargo, es posible indicarle al compilador de FORTRAN donde inician las dimensiones de la siguiente manera:

```
tipoDato DIMENSION (inicio_d1: d1, inicio_d2:d2, ...)
```

En este caso, las dimensiones se recorren desde la posición inicio\_d1 hasta la posición d1 para la primera dimensión, de la posición inicio\_d2 hasta la posición d2 para la segunda dimensión y así sucesivamente.

### Código (arreglos multidimensionales)

```
program arregloBidimensional

c Este programa genera un arreglo bidimensional de
c 3 renglones y 3 columnas y accede a cada elemento
c del arreglo a través de un ciclo do

integer i,j, matriz(3,3)

matriz(1,1) = 1
matriz(1,2) = 2
matriz(1,3) = 3
matriz(2,1) = 4
matriz(2,2) = 5
matriz(2,3) = 6
matriz(3,1) = 7
matriz(3,2) = 8
matriz(3,3) = 9

write (*,*) 'Imprimir matriz'

do i = 1, 3, 1
  do j = 1, 3, 1
    write (*,*) matriz(i,j)
  enddo
enddo

stop
end
```

### Código (arreglos multidimensionales)

```
program restaDeMatrices

c Este programa genera las matrices (A y B) y después
c las resta generando una tercera matriz C

integer i,j,cont
integer, dimension (-4:-1,3) :: matrizA, matrizB
integer matrizC(-4:-1,3)

c se genera la matriz A
cont = 1
do i = -4, -1, 1
  do j = 1, 3, 1
    matrizA(i,j) = cont
    cont = cont + 1
  enddo
  write (*,*) ''
enddo
```

```

c se genera la matriz B
  do i = -4, -1, 1
    do j = 1, 3, 1
      cont = cont - 1
      matrizB(i,j) = cont
    enddo
    write (*,*) ''
  enddo

c se imprimen las matrices generadas
write (*,*) 'Matriz A          Matriz B'
do i = -4, -1, 1
  do j = 1, 3, 1
    write(*,*) matrizA(i,j), ' ', matrizB(i,j)
  enddo
  write (*,*) ''
enddo

c se genera e imprime la matriz C
write (*,*) 'Matriz C = Matriz A - Matriz B'
do i = -4, -1, 1
  do j = 1, 3, 1
    matriz(i,j) = matrizA(i,j) - matrizB(i,j)
    write (*,*) matrizC(i,j)
  enddo
  write (*,*) ''
enddo

stop
end

```

## Bibliografía

- Oracle (2010). Fortran 77 Lenguaje Reference. Consulta: Julio de 2015. Disponible en: <http://docs.oracle.com/cd/E19957-01/805-4939/>
- Stanford University (1995). Fortran 77 Tutorial. Consulta: Julio de 2015. Disponible en: [http://web.stanford.edu/class/me200c/tutorial\\_77/](http://web.stanford.edu/class/me200c/tutorial_77/)