

Fundamentos de la Programación

¿Qué es un arreglo?

Recuerda que **un array o arreglo es un conjunto finito y ordenado de elementos homogéneos**. La característica de ser "**ordenado**" implica que cada elemento, desde el primero hasta el último, puede ser identificado. Además, los elementos de un arreglo son **homogéneos**, lo que significa que son del mismo tipo de datos.

Otra característica fundamental de los arreglos es su incapacidad de ser redimensionados. Al declararlos, se establece tanto el tipo de dato que contendrán como su capacidad de almacenamiento, es decir, cuántos "espacios" tienen disponibles para guardar información. Esta limitación los hace **finitos** y les impide alterar su tamaño una vez definidos.

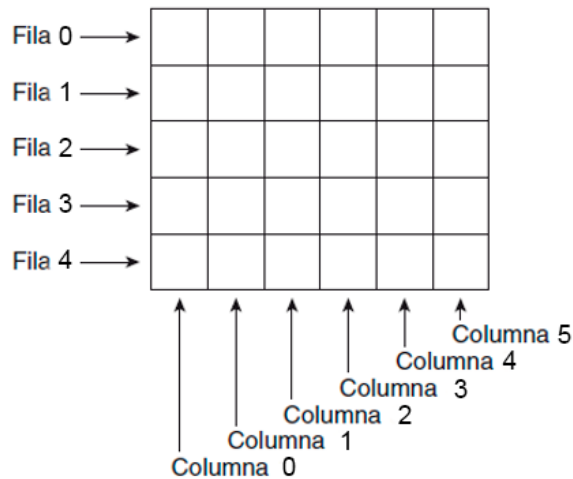
Los arreglos pueden tener múltiples dimensiones, y entre los más comunes se encuentran **las matrices, que son arreglos bidimensionales**.

¿Qué es una matriz?

Una matriz puede concebirse como **una colección de vectores organizados en filas y columnas**. Está compuesta por elementos del mismo tipo, donde el orden de dichos elementos es crucial y se requiere la especificación de dos subíndices para identificar cada elemento en la estructura.

Cuando se observa una matriz unidimensional, se puede asimilar a una sola columna de datos. Por otro lado, una matriz bidimensional, o simplemente matriz, se compone de filas y columnas:

En una matriz, un único subíndice no basta para referirse a un elemento específico; **se utilizan dos subíndices: uno para la fila y otro para la columna**. Por consiguiente, se considera que una matriz posee dos dimensiones, una por cada subíndice, y se requiere un valor para cada uno de ellos para identificar un elemento de manera individual. En la notación estándar, generalmente el primer subíndice indica la fila del arreglo, mientras que el segundo subíndice denota la columna.



En una matriz, la combinación de un par de subíndices es necesaria para referirse a un elemento específico. El primer subíndice indica la posición de la fila, mientras que el segundo subíndice indica la posición de la columna. En una matriz de tamaño NxM, los subíndices para las filas van desde 0 hasta N-1, y para las columnas van desde 0 hasta M-1.

Declaración

```
Definir nombre_matriz como Tipo_de_Dato  
Dimension nombre_matriz(tamañoFila,tamañoColumna)
```

La declaración 'Dimension' se utiliza para asignar tamaño a una matriz de cualquier tipo de dato simple, como entero, real, cadena o lógico. Este tamaño debe ser un par de números enteros o variables enteras; no puede contener números con decimales. La dimensión especificada determina cuántos elementos puede almacenar la matriz en cada dimensión. Es importante tener en cuenta que una vez declarada, la dimensión de la matriz no puede cambiar.

ASIGNAR ELEMENTOS A UNA MATRIZ

Cuando desees ingresar un elemento en tu matriz, selecciona los subíndices donde desees almacenarlo. Luego, invoca tu matriz por su nombre, seguido de paréntesis con los subíndices deseados separados por comas. Utiliza el signo igual (que es el operador de asignación) seguido del elemento que desees guardar. Recuerda que el elemento a guardar debe coincidir con el tipo de dato de tu matriz; por ejemplo, si tu matriz es de tipo entero, solo puedes guardar números enteros. Además, asegúrate de que los subíndices sean válidos según las dimensiones de tu matriz.

Formas de asignar valores a las matrices en PSeInt:

Existen diversas formas de asignar valores a las matrices en PSeInt. Una de ellas es la asignación manual, donde los valores se ingresan uno por uno en cada posición de la matriz. Otra forma es recorrer la matriz utilizando bucles anidados y asignar valores según la necesidad, como de uno en uno o utilizando fórmulas matemáticas para calcular los valores.

Ejemplo: Asignación Manual:

```
matriz[0,0] = 1
matriz[0,1] = 2
matriz[0,2] = 3
matriz[1,0] = 4
matriz[1,1] = 5
matriz[1,2] = 6
matriz[2,0] = 7
matriz[2,1] = 8
matriz[2,2] = 9
```

En este ejemplo, los valores se asignan manualmente a cada posición de la matriz.

Ejemplo: Recorriendo la Matriz:

```
Para fila = 0 hasta 2 hacer
  Para columna = 0 hasta 2 hacer
    matriz[fila,columna] = AZAR(101) // Crea un numero al azar menor a 101
  FinPara
FinPara
```

En este caso, se utilizan bucles anidados para recorrer la matriz y asignar valores de manera dinámica, asignando números al azar en cada posición, menores a 100.

Mostrar o acceder a elementos de un arreglo:

Cuando necesitas mostrar o acceder a un elemento específico de tu arreglo, bidimensional simplemente escribe el nombre del arreglo seguido de corchetes o paréntesis con los 2 índices del elemento que deseas obtener.

Por ejemplo:

```
Escribir nombre_Matriz[0,0]
```

Estas líneas te permiten acceder al primer elemento de la matriz `nombre_Matriz`.

Para mostrar todos los elementos de nuestra matriz bidimensional, necesitamos utilizar dos estructuras de bucle 'Para'. Uno de ellos recorrerá las filas, mientras que el otro recorrerá las columnas. Por lo tanto, para una matriz de tamaño 3x3, necesitaremos dos bucles 'Para' que vayan desde 0 hasta 2.

```
Para fila ← 0 hasta 2 hacer
  Para columna ← 0 hasta 2 hacer
    Escribir matriz[fila,columna], sin saltar " "
  FinPara
  Escribir ""
FinPara
```

Este ejemplo recorre una matriz cuadrada, donde el número de filas es igual al número de columnas.

Uso en Subprogramas

Los arreglos pueden ser pasados como parámetros a un subprograma (función o procedimiento) de la misma manera que las variables escalares. Sin embargo, **es importante tener en cuenta que los arreglos**, a diferencia de los tipos de datos simples, **se pasan siempre como parámetros "Por Referencia"**. Esto significa que, al utilizar arreglos en nuestros subprogramas, generalmente los modificamos, rellenamos o mostramos.

Ejemplo de Función:

```
Funcion variable_de_retorno ← Nombre (matriz por referencia)
  Definir variable_de_retorno como Tipo de Dato
  <acciones>
Fin Funcion
```

En este ejemplo, la función Nombre recibe un arreglo como parámetro por referencia. Puedes realizar operaciones sobre este arreglo dentro de la función y devolver un valor al final.

Ejemplo de Procedimiento:

```
SubProceso Nombre (matriz por referencia)  
    <acciones>  
FinSubProceso
```

En este ejemplo, el procedimiento Nombre recibe un arreglo como parámetro por referencia. Puedes realizar cualquier operación sobre este arreglo dentro del procedimiento.