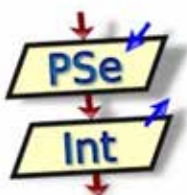


PROGRAMACIÓN DESDE CERO

ARREGLOS CON PSEINT



Universidad de
LA PUNTA



GOBIERNO DE
SAN LUIS





Objetivos de la Guía

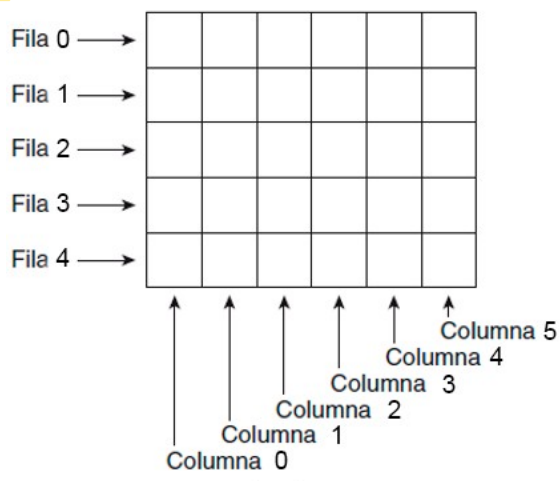
En esta guía aprenderemos a:

- Definir arreglos de acuerdo al tipo de dato que contendrán.
- Dimensionar arreglos.
- Rellenar arreglos.
- Mostrar correctamente arreglos por pantalla.
- Utilizar funciones y subprogramas, y estructuras de control para trabajar con arreglos y matrices.

ARREGLOS BIDIMENSIONALES: MATRICES

¿QUÉ SON LAS MATRICES?

Una matriz se puede considerar como un **vector de vectores**. Una matriz es un **conjunto de elementos, todos del mismo tipo**, en el cual el **orden de los componentes es significativo** y en el que se necesita **especificar dos subíndices** para poder identificar cada elemento del arreglo. Si se visualiza un arreglo unidimensional, se puede considerar como una columna de datos; un arreglo bidimensional o matriz es un **grupo de filas y columnas**:



En una matriz un subíndice no es suficiente para especificar un elemento, se referencian con dos subíndices: el **primer subíndice se refiere a la fila** y el **segundo subíndice se refiere a la columna**. Por lo tanto, una matriz se considera que **tiene dos dimensiones** (una dimensión por cada subíndice) y necesita un valor para cada subíndice para poder identificar un elemento individual. En notación estándar, normalmente el primer subíndice se refiere a la fila del arreglo, mientras que el segundo subíndice se

refiere a la columna.

Declaración

Definir `nombre_matriz` como `Tipo_de_Dato`

`Dimension nombre_matriz(tamañoFila,tamañoColumna)`

Donde `Tipo_De_Dato` se corresponde con cualquiera de los tipos de datos simples vistos previamente: entero, real, cadena, lógico.

La declaración `Dimension` nos sirve para darle el tamaño a nuestra matriz, que recordemos, no puede cambiar una vez declarada. A diferencia de los vectores, vamos a tener que darle un tamaño a las filas y un tamaño a las columnas, separadas por coma.

A la hora de definir el tamaño de una matriz, no es necesario que sean matrices cuadradas. Las matrices cuadradas son, cuando tienen el mismo tamaño tanto para las filas que para las columnas. Pero podemos crear matrices que tengan valores distintos en filas y columnas.

ARREGLOS MULTIDIMENSIONALES

Un arreglo puede ser definido de tres dimensiones, cuatro dimensiones, hasta de n-dimensiones. Los conceptos de rango de subíndices y número de elementos se pueden ampliar directamente desde arreglos de una y dos dimensiones a estos arreglos de orden más alto. En general, un arreglo de n-dimensiones requiere que los valores de los n subíndices puedan ser especificados a fin de identificar un elemento individual del arreglo. Si cada componente de un arreglo tiene n subíndices, el arreglo se dice que es sólo de n-dimensiones. Declaración

Definir `nombre_arreglo` como `Tipo_de_Dato`

`Dimension nombre_arreglo(tamañoDim1,tamañoDim2,..., tamañoDimN)`

ASIGNAR ELEMENTOS A UN ARREGLO

Cuando queremos asignar un elemento a un arreglo bidimensional o matriz, vamos a necesitar pasarle dos subíndices, uno para las filas y otro para las columnas.

```
nombre_matriz[0,0] = 10
```

Y para poder asignar varios elementos a nuestra matriz, usaríamos dos bucles Para anidados, ya que un Para recorrerá las filas (variable i) y otro las columnas (variable j).

```
Para i<-0 Hasta 2 Con Paso 1 Hacer
```

```
    Para j<-0 2 Con Paso 1 Hacer
```

```
        nombre_matriz[i,j] = 10
```

```
    Fin Para
```

```
Fin Para
```

MOSTRAR O TRAER ELEMENTOS DE UN ARREGLO

Cuando queremos mostrar o traer un elemento de un arreglo bidimensional o matriz, vamos a necesitar pasarle dos subíndices, uno para las filas y otro para las columnas.

```
Escribir nombre_matriz[0,0]
```

```
Variable = nombre_matriz[0,0]
```

Ahora, si quisiéramos mostrar todos los elementos de nuestro arreglo bidimensional o matriz, vamos a tener que utilizar dos estructuras Para para traer todos los elementos de nuestra matriz, ya que un Para recorrerá las filas y otro las columnas.

```
Para i<-0 Hasta 2 Con Paso 1 Hacer
  Para j<-0 2 Con Paso 1 Hacer
    Escribir Sin Saltar nombre_matriz[i,j]
  Fin Para
  Escribir " "
Fin Para
```

Nota: este ejemplo funciona con una matriz cuadrada donde el tamaño de las filas sea el mismo que de las columnas.