APOYA

Introducción a la Programación

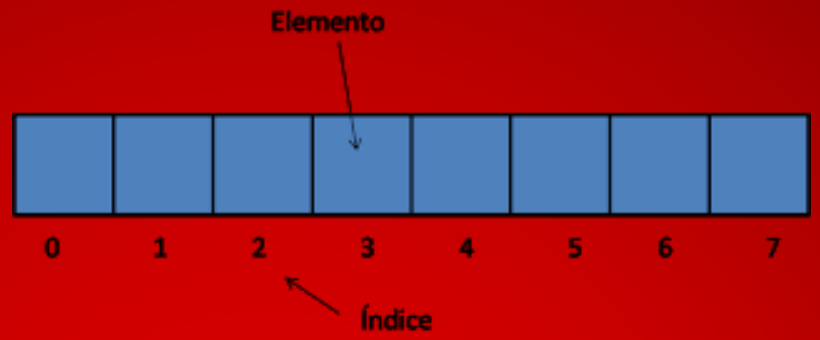
Arreglos: Matrices

Comisión "B"

Año 2020

Profesor: Ing. Gabriel Guismin

ARREGLOS



- **Un arreglo es un conjunto finito, ordenado y homogéneo de elementos.**

- ***Ordenado*** significa que se puede identificar al primer elemento, al segundo, al tercero y así sucesivamente.

- ***Finito***, significa que la cantidad de esos elementos será conocida y fija.

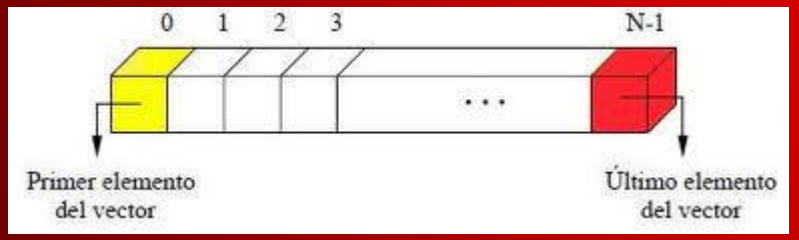
- ***Homogéneo***, indica que todos los elementos son del mismo tipo de datos.

ARREGLOS

- Existen dos tipos de arreglos muy utilizados:

los unidimensionales (llamados **vectores**) y;

los bidimensionales (llamados **matrices**)



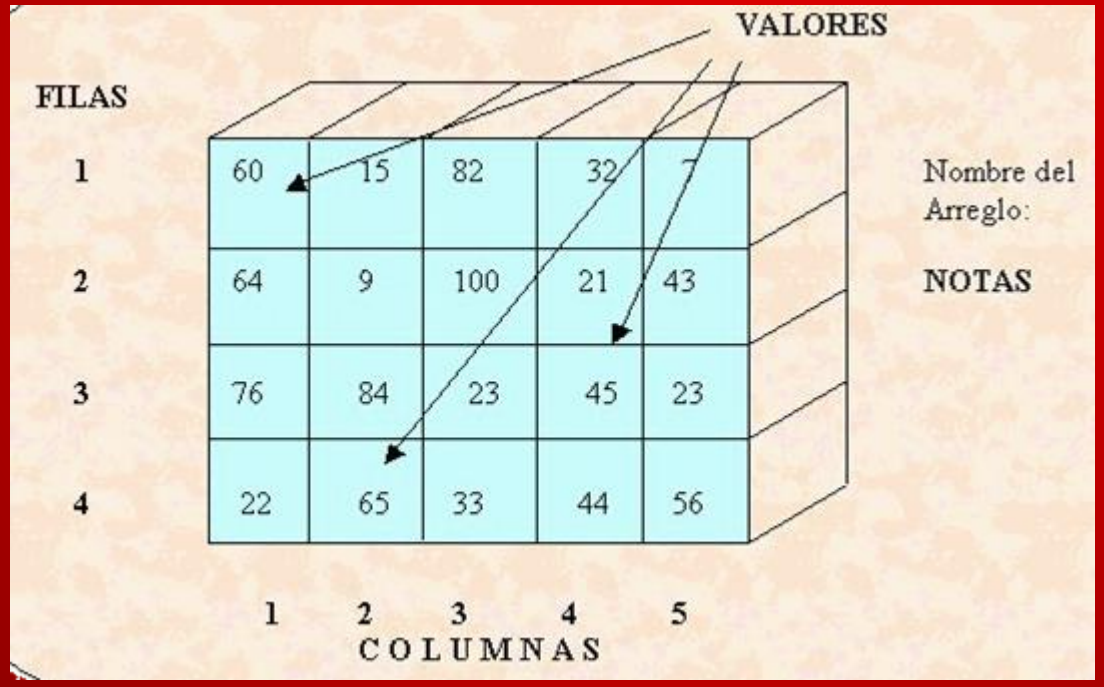
Declaración de un arreglo unidimensional - VECTOR

Declaración de un arreglo bidimensional - MATRIZ

Ing. Gabriel Guismin

Matrices

Una matriz es un tipo de arreglos de dos dimensiones. Podría decirse que la matriz es un arreglo de arreglos.



Ing. Gabriel Guismin

Declaración de una matriz

```
<tipo de datos> nombre[filas, columnas]
```

Declaración de una matriz

- Por ejemplo, para una matriz llamada datos con cuatro filas y seis columnas de celdas de tipo real, la declaración será:

```
real datos[4, 6]
```

Declaración de una matriz con 4 (cuatro) filas y 6 (seis) columnas

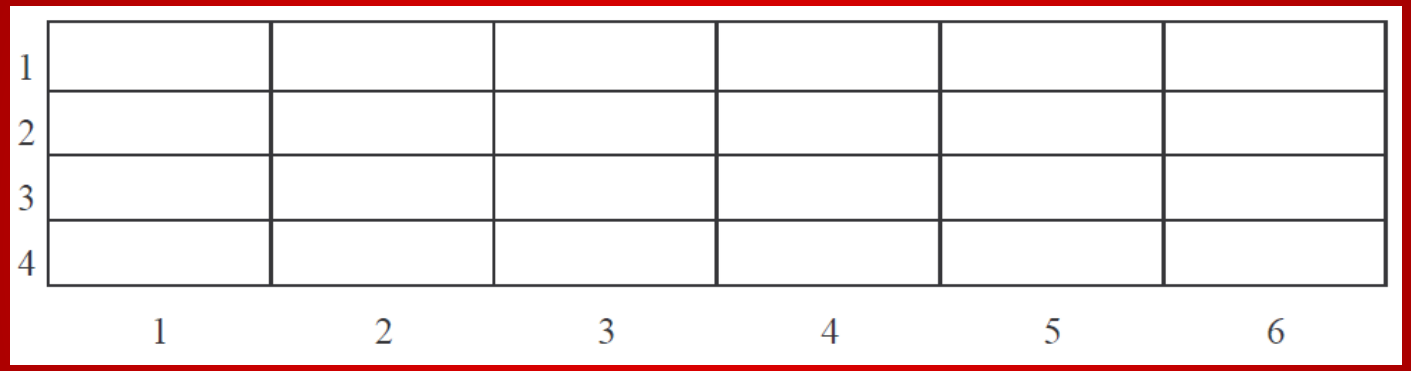
```
Dimension matriz[4,6]
```

Declaración de una matriz con 4 (cuatro) filas y 6 (seis) columnas - (En PSeInt)

Ing. Gabriel Guismin

Matrices: Asignación

Declarando una matriz de 4 (cuatro) filas y 6 (seis) columnas, llamada **datos [4,6]**



La asignación de valores a las celdas de una matriz se realiza de la siguiente manera:

datos[1,3] = 10
datos[4,5] = 3,141592
datos[3,1] = 4567,89

Matrices: Asignación

Luego de estas asignaciones, el valor 10 se almacena en la tercera celda de la primera fila; el valor 3,141592 en la quinta celda de la cuarta fila y el valor 4567,89 en la primer celda de la tercera fila. Gráficamente:

1	?	?	10	?	?	?
2	?	?	?	?	?	?
3	4567,89	?	?	?	?	?
4	?	?	?	?	3,141592	?
	1	2	3	4	5	6

Matrices: Lectura y Escritura

La **lectura** de datos en la matriz y la **escritura** de los datos almacenados se pueden realizar directamente indicando las celdas involucradas, al igual que en los vectores.

```
Leer matriz[1,1]
Leer matriz[2,3]

Escribir matriz[1,1]
Escribir matriz[2,3]
```

Ejemplo de Lectura y Escritura en posiciones indicadas de una matriz

Ing. Gabriel Guismin

Matrices: Acceso secuencial (recorrido)

Las matrices cuentan con dos dimensiones, por lo tanto, el algoritmo de recorrido debe permitir acceder a todas las columnas en todas las filas. Esto requiere de estructuras repetitivas anidadas.

```
para i ←1 hasta 6
    leer (datos[1,i])
finpara
```

Recorrido: De esta forma, se leen todas las columnas de la fila "1"

```
para i ←1 hasta 6
    leer (datos[2,i])
finpara
```

Recorrido: De esta forma, se leen todas las columnas de la fila "2"

Ing. Gabriel Guismin

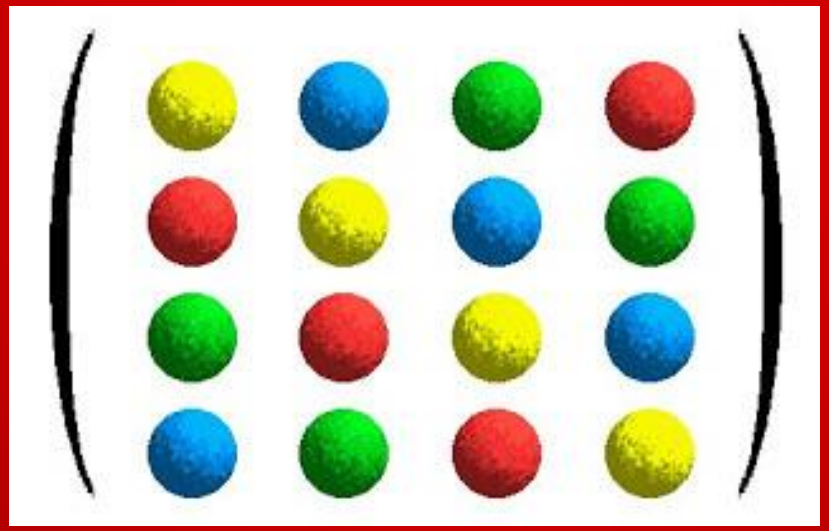
Matrices: Acceso secuencial (recorrido)

```
para i ← 1 hasta 4
    para j ← 1 hasta 6
        leer (datos[i, j])
    finpara
finpara
```

Recorrido: De esta forma, se leen todas las filas y columnas de la matriz

Matrices: Ejemplo práctico (1)

Problema: Se necesita un algoritmo para mostrar la suma de los datos almacenados en dos matrices. Las matrices poseen 3 filas y 2 columnas.



Ing. Gabriel Guismin

Matrices: Ejemplo práctico (2)

Problema: Cargar una matriz de 5 x 7 con números aleatorios en un rango de 1 al 10 e informar cuántas veces aparece el número “3” en la matriz.

73735	45963	78134	63873
02965	58303	90708	20025
98859	23851	27965	62394
33666	62570	64775	78428
81666	26440	20422	05720
15838	47174	76866	14330
89793	34378	08730	56522
78155	22466	81978	57323
16381	66207	11698	99314
75002	80827	53867	37797
99982	27601	62686	44711
84543	87442	50033	14021
77757	54043	46176	42391
80871	32792	87989	72248
30500	28220	12444	71840

Ing. Gabriel Guismin

Para conocer: en JAVA

```java
int matriz[][]=new int[3][3];
```

Ejemplo: Declaración e inicialización de una matriz en Java

```java
//i = filas y j = columnas
for(int i=0;i<matriz.length;i++){
    for(int j=0;j<matriz[0].length;j++){
        matriz[i][j]=(i*matriz.length)+(j+1);
        System.out.print(matriz[i][j]+" ");

    }
    System.out.println("");
}
```

Ejemplo de recorrido de una matriz en Java

Ing. Gabriel Guismin