# Fundamentos de Programación Científica

## Arreglos

( Bidimensionales - Matrices )

Delio Augusto Aristizábal Martínez

Docente

2017-2

M a t r i c e s

# Definición

Al igual que los vectores, una matriz es una colección estructurada, finita y ordenada de datos homogéneos (mismo tipo) en *dos dimensiones*(filas y columnas) que comparten un mismo nombre.

Tiene un tipo de acceso directo, mediante el nombre del arreglo y el uso de dos índices simultáneos; que nos indica la posición relativa del elemento dentro del arreglo.

En C++ y en la mayoría de los lenguajes de programación la primera fila y la primera columna de una matriz se identifican con el índice=0 con incrementos de uno.

En Matlab ambos se identifican desde el índice=1.

# Tipos de matrices:

## Matriz Fila:

Es la matriz que tiene una sola fila y ***M*** columnas.

| | | |
|---|---|---|
| 10 | -4 | 8 |

## Matriz Columna:

Es la matriz que tiene una sola columna y ***N*** filas.

| |
|---|
| 15 |
| 21 |
| -32 |

## Matriz Rectangular:

Es la matriz que tiene distinto número de filas y columnas.

| | |
|---|---|
| 10 | -8 |
| 4 | 0 |
| -16 | 14 |
| 6 | 6 |

## Matriz Cuadrada:

Es la matriz en la cual la cantidad de filas es igual a la cantidad de columnas.

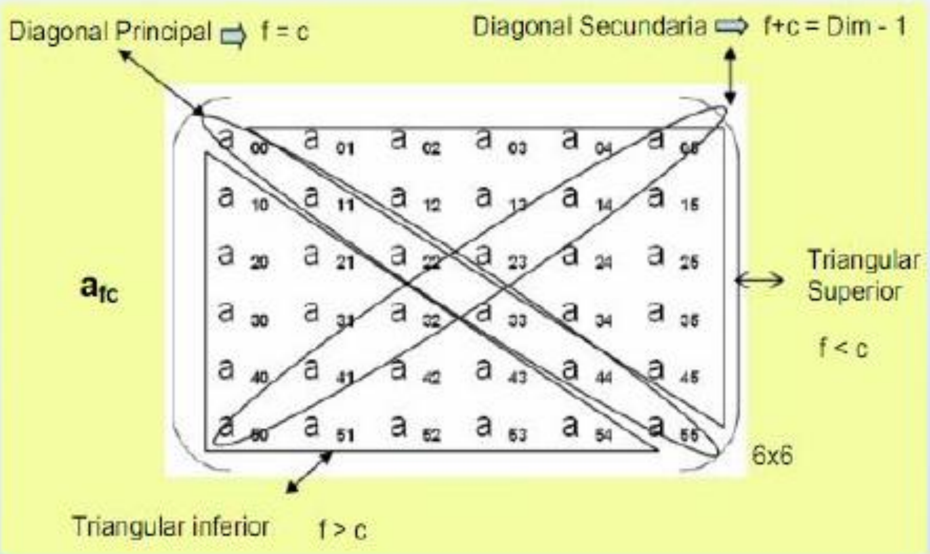
| | | |
|---|---|---|
| 10 | -4 | 10 |
| 25 | 0 | 14 |
| 32 | 14 | 5 |

# Tipos de Matrices

Las matrices cuadradas tienen algunas características importantes asociadas a la ubicación de sus elementos en el conjunto. Se puede hablar de:

- ✓ Diagonal Principal
- ✓ Diagonal Secundaria
- ✓ Triangular Superior
- ✓ Triangular Inferior

Por mencionar algunas.




## Triangular Superior:

Es la matriz donde los elementos debajo diagonal de la principal son nulos. (Matriz ***A***)

$$A = \begin{pmatrix} 4 & 1 & 2 \\ 0 & -3 & 1 \\ 0 & 0 & 8 \end{pmatrix} \text{ triangular superior } B = \begin{pmatrix} -2 & 0 & 0 \\ 1 & 2 & 0 \\ -3 & 4 & 5 \end{pmatrix} \text{ triangular inferior}$$

## Triangular Inferior:

Es la matriz donde los elementos encima de la diagonal principal son nulos. (Matriz ***B***)

# Tipos de matrices:

## Matriz diagonal:

Es la matriz cuadrada donde todos los elementos fuera de la diagonal son cero.

| | | |
|---|---|---|
| 10 | 0 | 0 |
| 0 | 4 | 0 |
| 0 | 0 | 5 |

## Matriz Escalar:

Es la matriz cuadrada en el que todos los términos de la diagonal son iguales.

| | | |
|---|---|---|
| 10 | 0 | 0 |
| 0 | 10 | 0 |
| 0 | 0 | 10 |

## Matriz unidad o matriz identidad: **eye (n)** :

Es la matriz cuadrada escalar cuya diagonal principal elementos son 1.

| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

## Matriz Transpuesta:

Es la matriz en la cual las filas de una matriz son las columnas de la otra. La matriz transpuesta se denota con un apóstrofe. Ejm: A’

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad A' = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

## Matriz Simétrica:

Es la matriz cuadrada cuya transpuesta es igual a la misma matriz.

| | | |
|---|---|---|
| 10 | -4 | 10 |
| -4 | 0 | 14 |
| 10 | 14 | 5 |

# Declaración

***La sintaxis para declarar una matriz en C++ es la siguiente:***

Tipodato ***NombreMatriz***[***filas***][***columnas***];

**Tipodato**: Tipo de dato de los elementos del arreglo (int, float, etc.)
**NombreMatriz**: identificador que representa la colección de elementos.
**filas**: constante entera positiva que representa la cantidad de filas.
**columnas**: constante entera positiva que representa la cantidad de columnas.

```
int     myMatriz1[10][5];
float   myMatriz2[5][10];
string  myMatriz3[15][15];
bool    myMatriz4[1000][3];
```

```
int myMatriz1[2][2] = { {1,2},{3,4} };
```

# Operaciones Básicas

$$\text{int Datos[3][2];} \Longrightarrow$$

| | 0 | 1 |
|---|---|---|
| 0 | 7 | 8 |
| 1 | 0 | 20 |
| 2 | 9 | 7 |

## Llenar

$Datos[0][1] = 8;$ $Datos[1][0] = 0;$

$Datos[1][1] = -2;$ $Datos[2][0] = 9;$ $Datos[1][1] = 20;$

$Datos[2][1] = 0;$ $Datos[0][0] = 7;$ $Datos[2][1] = 7;$

## Mostrar

$cout << Datos[1][1]; \Longrightarrow$ 20

## Recuperar

$X = Datos[0][1]; \Longrightarrow$ X = 8

$X = Datos[1][1]/Datos[0][1]; \Longrightarrow$ X = 2,5

Cada elemento es referenciado por la posición que ocupa dentro del arreglo por el índice de la fila y el índice de la columna; y siempre son correlativos.

# Otras declaraciones

```
/* En el siguiente ejemplo se asignan ceros hasta completar las filas */
 double m[][3] = {1.1, 1.2, 1.3, 1.4, 2.1, 2.2, 2.3};

          fil = (sizeof(m)/sizeof(m[0]));
          col = (sizeof(m[0])/sizeof(m[0][0]));
          for ( int i = 0; i < fil; i++)
          {
             for (int j = 0; j < col; j++)
                cout << m[i][j] << "\t";
             cout << endl;
          }
```

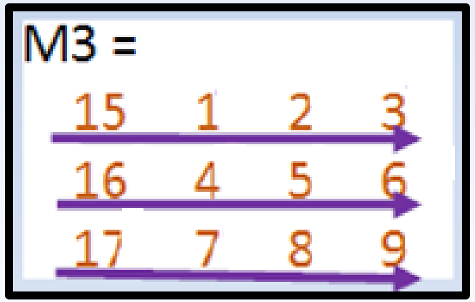
```
/* Las siguientes inicializaciones son INCORRECTAS */
double x[][] = {1.1, 1.2, 1.3, 1.4};
double y[2][] = {1.1, 1.2, 1.3, 1.4};
```

# Recorrido de un arreglo

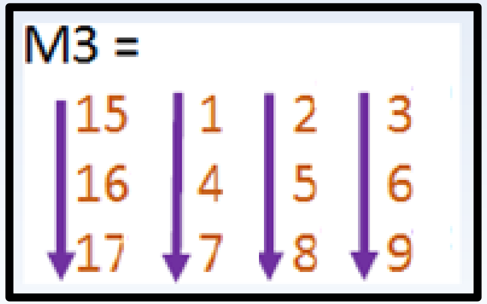
```
int M3[3][4] = { {15,1, 2, 3},{16, 4, 5, 6},{17, 7, 8, 9} };

int fil = (sizeof(M3)/sizeof(M3[0]));
int col = (sizeof(M3[0])/sizeof(M3[0][0]));
```

```
//Recorrido por filas
 for ( int i = 0; i < fil; i++)
   {
        for (int j = 0; j < col; j++)
             cout << M3[i][j] << "\t";
        cout << endl;
   }
```




```
//Recorrido por columnas
 for (int j = 0; j < col; j++)
   {
        for ( int i = 0; i < fil; i++ )
             cout << M3[i][j] << "\t";
        cout << endl;
   }
```



# Operaciones Básicas

## Suma de dos matrices:

Consiste en sumar ***A***+***B*** de igual tamaño, es decir ***A***(i,j) +***B***(i,j) resultando una nueva matriz ***C***, donde cada posición

***C***(i,j) ← ***A***(i,j) + ***B***(i,j)

$$A = \begin{vmatrix} 1 & 3 \\ 5 & 7 \end{vmatrix} \quad B = \begin{vmatrix} 5 & 7 \\ 4 & 8 \end{vmatrix}$$

$$C = \begin{vmatrix} 6 & 10 \\ 9 & 15 \end{vmatrix}$$

## Resta de dos matrices:

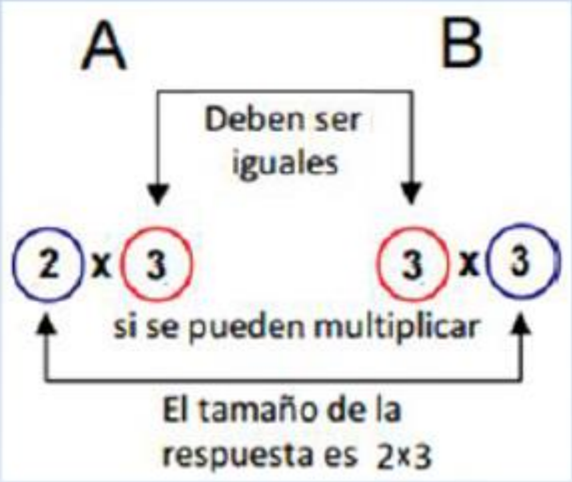
Consiste en restar ***A***-***B*** de igual tamaño, es decir ***A***(i,j) - ***B***(i,j) resultando una nueva matriz ***C***, donde cada posición

***C***(i,j) ← ***A***(i,j) - ***B***(i,j)

$$C = \begin{vmatrix} -4 & -4 \\ 1 & -1 \end{vmatrix}$$

# Operaciones Básicas

## Multiplicar dos Matrices




$$AB=\begin{bmatrix}6 & -2 & 8\\ 1 & 4 & 5\end{bmatrix}\cdot\begin{bmatrix}1 & 3\\ 5 & 0\\ 2 & 7\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}6 & -2 & 8\end{bmatrix}\begin{bmatrix}1\\5\\3\end{bmatrix} & \begin{bmatrix}6 & -2 & 8\end{bmatrix}\begin{bmatrix}3\\0\\7\end{bmatrix}\\ \begin{bmatrix}1 & 4 & 5\end{bmatrix}\begin{bmatrix}1\\5\\3\end{bmatrix} & \begin{bmatrix}1 & 4 & 5\end{bmatrix}\begin{bmatrix}3\\0\\7\end{bmatrix}\end{bmatrix}$$

$$=\begin{bmatrix}6-10+16 & 18+0+56\\ 1+20+15 & 3+0+35\end{bmatrix}=\begin{bmatrix}12 & 74\\ 36 & 38\end{bmatrix}$$

# Ejercicios

En la matriz cuadrada (*Datos*) de NxN con valores reales realizar los siguientes procedimientos:
a) Llenar y mostrar la matriz.
b) Mostrar los elementos de la diagonal Principal y Secundaria así como también la suma de cada una.
C) Llenar dos vectores (Sfilas) y Scolumnas) con las sumas de cada fila y cada columna, mostrarlas.

**Escriba un programa que almacene en una matriz el número de personas que ingresan a una sala de cine X durante cada uno de los días del mes. La matriz debe constar de 5 filas para cada una de las semanas y 7 columnas cada uno de los días de la semana. El programa debe calcular el promedio de personas que ingresan a la sala, promedio de personas por semana y promedio de personas por día, todas en el mes. Nota: Tener en cuanta que el mes puede empezar en cualquier día de la primera semana del mes, los espacios no ocupados deben de estar en cero y no deben ser tenidos en cuenta para los promedios.**

## Tres en raya.

Para jugar al 3 en raya se utiliza una matriz 3x3. Las casillas ocupadas por el primer jugador tienen un 1, las ocupadas por el segundo tienen un 2 y las que están libres un 0. Cada jugador tiene cuatro fichas que ubican en el tablero a su estrategia. No se puede colocar una ficha si esta ocupada

a) Escribir una función que indique si el jugador 1 tiene "3 en raya" en horizontal.
b) Escribir una función que indique si el jugador 1 tiene "3 en raya" en vertical.
c) Escribir una función que indique si el jugador 1 tiene "3 en raya" en cualquiera de las dos diagonales del tablero.
d) Escribir una función que genere (Limpie) el tablero.

Utilizando las funciones anteriores, escribir una función que indique si el jugador 1 tienen 3 en raya (en cualquier posición).

Un profesor quiere analizar los resultados de los exámenes que propone a sus alumnos. Cada alumno se examina 3 veces en un curso. Y en cada ocasión realiza dos tipos de examen: un test y uno de problemas. Por tanto, en total son 6 exámenes por alumno.

a) Calcular la media por cada tipo de examen en cada convocatoria. La función debe devolver un vector con las 6 medias.
b) Calcular la media por alumno. También se devolverá un vector con todas las medias.
c) Escribir una función que indique el número de alumnos cuya nota media es superior a la media de todos los exámenes para todos los alumnos.