



UNIVERSIDAD NACIONAL DE TRUJILLO

UNT

# MATRICES

Mg. Marcelino Torres Villanueva



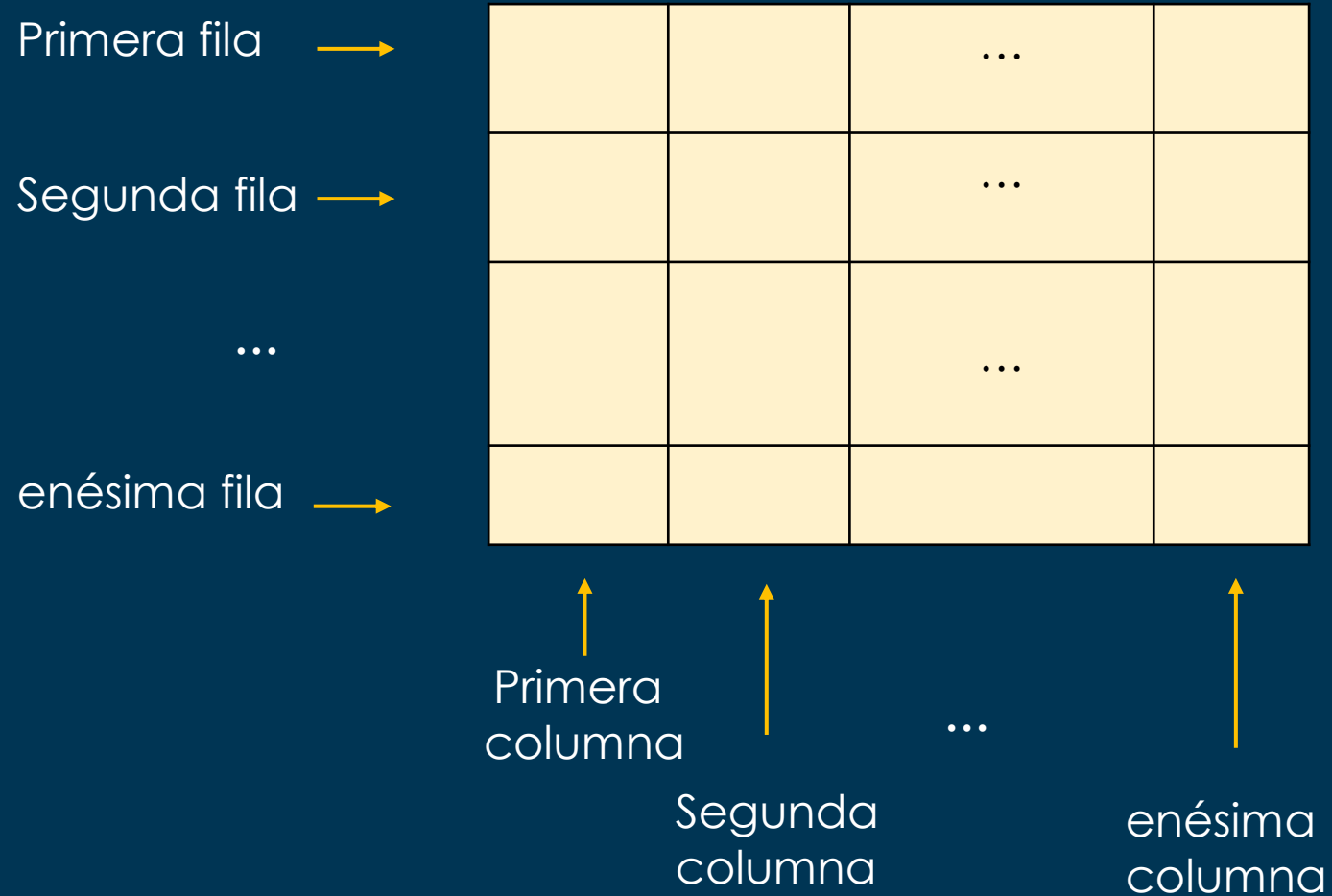
# DEFINICION

Un arreglo bidimensional es un conjunto de  $n$  elementos del mismo tipo almacenados en memoria contigua en una matriz o tabla.

A diferencia de los arreglos unidimensionales que sólo requieren de un subíndice, los arreglos bidimensionales para acceder a cada elemento del arreglo requieren de dos índices o subíndices declarados en dos pares de corchetes, donde el primer corchete se refiere al tamaño de filas y el segundo al tamaño de columnas.



# Arreglo Bidimensional





## Índices y componentes de un arreglo bidimensional

A

A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[0][4]	A[0][5]
8	6	4	1	-2	3
4	3	0	-4	2	4
2	3	5	0	2	1
A[2][0]	A[2][1]	A[2][2]	A[2][3]	A[2][4]	A[2][5]



# DECLARACIÓN DE UNA MATRIZ

**tipo\_de\_dato** **identificador**[numFilas][numColumnas];

```
int notas[filas][columnas];
```

Dónde:

**tipo\_de\_dato:** Es el tipo de datos que contendrá la matriz.

**identificador:** Es el nombre que le damos a la variable matriz y por el cual la referenciaremos en nuestro programa.

**[numFilas][numColumnas]** : Especifica el número de Filas y de columnas que tendrá la matriz.



El espacio que las matrices ocupan en memoria se reserva en el momento de realizar la declaración de los mismos.

Ejemplo:

```
int X[2][5]; // Declaración de una matriz de enteros de 2 filas y 5 columnas
float Y[7][10]; // Declaración de una matriz de reales de 7 filas y 10 columnas
char C[8][3]; // Declaración de una matriz de caracteres de 8 filas y 3 columnas
```



# INICIALIZACIÓN DE UNA MATRIZ

Una matriz se puede inicializar para esto hay que agrupar entre {} cada fila.  
El formato a utilizar sería el siguiente:

```
tipo_de_dato identificador[ filas ][ columnas ] = {  
    { columnas de la fila 1 },  
    { columnas de la fila 2 },  
    ... ,  
    { columnas de la última fila }  
};
```

No debemos olvidar el ';' al final.



Ejemplo:

```
int temperaturas[3][5] = { { 15, 17, 20, 25, 10 }, { 18,  
20, 21, 23, 18 }, { 12, 17, 23, 29, 16 } };
```



**01.** Hacer un programa para generar una matriz de  $f$  filas y  $c$  columnas y calcular el mayor, el menor y el promedio.

#### PSEUDOCODIGO

**ordenMatriz**(entero  $f(R)$  , entero  $c(R)$  )

hacer

escribir "Numero de filas: "

leer  $f$

mientras  $f \leq 0$  o  $f \geq 20$

hacer

escribir "Numero de columnas: "

leer  $c$

mientras  $c \leq 0$  o  $c \geq 20$

**fin\_ordenMatriz**





## PSEUDOCODIGO

**ingresoMatriz(real M[][], entero f, entero c)**

entero i, j

para i  $\leftarrow$  0 hasta f - 1 inc 1 hacer

    para j  $\leftarrow$  0 hasta c - 1 inc 1 hacer

        escribir "M", i, "[", j, "]:"

        leer M[i][j]

    fin\_para

fin\_para

**fin\_ingresoMatriz**



## PSEUDOCODIGO

**reporteMatriz(real M[][], entero f, entero c)**

entero i, j

para i  $\leftarrow$  0 hasta f - 1 inc 1 hacer

para j  $\leftarrow$  0 hasta c - 1 inc 1 hacer

escribir M[i][j], " "

fin\_para

escribir "cambio de linea"

fin\_para

**fin\_reporteMatriz**



## PSEUDOCODIGO

```
real mayor(real M[][], entero f, entero c)
    entero i,j
    real may
    may  $\leftarrow$  M[0][0]
    para i  $\leftarrow$  0 hasta f - 1 inc 1 hacer
        para j  $\leftarrow$  0 hasta c - 1 inc 1 hacer
            si M[i][j] > may entonces
                may  $\leftarrow$  M[i][j]
            fin_si
        fin_para
    fin_para
    retornar may
fin_mayor
```



## PSEUDOCODIGO

```
real menor(real M[][], entero f, entero c)
    entero i,j
    real men
    men  $\leftarrow$  M[0][0]
    para i  $\leftarrow$  0 hasta f - 1 inc 1 hacer
        para j  $\leftarrow$  0 hasta c - 1 inc 1 hacer
            si M[i][j] < men entonces
                men  $\leftarrow$  M[i][j]
            fin_si
        fin_para
    fin_para
    retornar men
fin_menor
```



## PSEUDOCODIGO

**real promedio(real M[][], entero f, entero c)**

entero i,j

real s

para i  $\leftarrow$  0 hasta f - 1 inc 1 hacer

para j  $\leftarrow$  0 hasta c - 1 inc 1 hacer

s  $\leftarrow$  s + M[i][j]

fin\_para

fin\_para

retornar s / (f\*c)

**fin\_promedio**



## PSEUDOCODIGO

### Algoritmo matriz\_01

```
real M[20][20]
entero f,c
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz ingresada"
reporteMatriz(M,f,c)
escribir "El mayor es: ", mayor(M,f,c)
escribir "El menor es: ", menor(M,f,c)
escribir "El promedio: ", promedio(M,f,c)
```

**fin\_algoritmo**



## CODIFICACION

```
#include<iostream>
using namespace std;
#define MAX 10
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
float mayor(float M[][MAX], int f, int c);
float menor(float M[][MAX], int f, int c);
float promedio(float M[][MAX], int f, int c);
int main()
{
    float M[MAX][MAX];
    int f,c;
    ordenMatriz(f,c);
    ingresoMatriz(M,f,c);
    cout<<"Matriz ingresada"<<endl;
    reporteMatriz(M,f,c);
    cout<<"El mayor es : "<<mayor(M,f,c)<<endl;
    cout<<"El menor es : "<<menor(M,f,c)<<endl;
    cout<<"El promedio : "<<promedio(M,f,c)<<endl;

    return 0;
}

void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}
```

```
void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
        {
            cout<<"M["<<i<<"]["<<j<<": ";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX],int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
    {
        for(j=0;j<c;j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}

float mayor(float M[][MAX], int f, int c)
{
    int i,j;
    float may=M[0][0];
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
            if(M[i][j]>may)
                may=M[i][j];
    return may;
}
```



CODIFICACION	SALIDA DE PANTALLA						
<pre>float menor(float M[][MAX], int f, int c) {     int i,j;     float men=M[0][0];     for(i=0;i&lt;f;i++)         for(j=0;j&lt;c;j++)             if(M[i][j]&lt;men)                 men=M[i][j];     return men; }  float promedio(float M[][MAX], int f, int c) {     int i,j;     float s=0;     for(i=0;i&lt;f;i++)         for(j=0;j&lt;c;j++)             s=s+M[i][j];     return s/(f*c); }</pre>	<p>Número de filas : 2 Número de columnas : 3 M[0][0]:12 M[0][1]:23 M[0][2]:6 M[1][0]:5 M[1][1]:9 M[1][2]:12</p> <p>Matriz ingresada</p> <table><tr><td>12</td><td>23</td><td>6</td></tr><tr><td>5</td><td>9</td><td>12</td></tr></table> <p>El mayor es : 23 El menor es : 5 El promedio : 11.1667</p>	12	23	6	5	9	12
12	23	6					
5	9	12					





**02.** Ingresar una matriz de  $f$  filas y  $c$  columnas y calcular la suma de filas y la suma de columnas

#### PSEUDOCODIGO

**sumaDeFilas**(real  $A[] []$ , entero  $f$ , entero  $c$ , real  $sf[]$ )

entero  $i, j$

para  $i \leftarrow 0$  hasta  $f - 1$  inc 1 hacer

$sf[i] \leftarrow 0$

    para  $j \leftarrow 0$  hasta  $c - 1$  inc 1 hacer

$sf[i] \leftarrow sf[i] + A[i][j]$

    fin\_para

fin\_para

**fin\_sumaDeFilas**



## PSEUDOCODIGO

**sumaDeColumnas**(real  $A[] []$ , entero  $f$ , entero  $c$ , real  $sc[]$ )

entero  $i, j$

para  $j \leftarrow 0$  hasta  $c - 1$  inc 1 hacer

$sc[j] \leftarrow 0$

    para  $i \leftarrow 0$  hasta  $f - 1$  inc 1 hacer

$sc[j] \leftarrow sc[j] + A[i][j]$

    fin\_para

fin\_para

**fin\_sumaDeColumnas**



## PSEUDOCODIGO

### Algoritmo matrices\_02

```
real M[20][20],sf[20],sc[20]
entero f, c, i ,j
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz Ingresada"
reporteMatriz(M,f,c)
sumaDeFilas(M,f,c,sf)
para i ←0 hasta f – 1 inc 1 hacer
    escribir "Suma de fila", i, " = ", sf[i]
fin_para

para j ←0 hasta c – 1 inc 1 hacer
    escribir "Suma de columna “, j, " = ", sc[j]
fin_para
```

**fin\_algoritmo**



## CODIFICACION

```
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
void sumaDeFilas(float A[][MAX], int f, int c, float sf[MAX]);
void sumaDeColumnas(float A[][MAX], int f, int c, float sc[MAX]);
int main(int argc, char *argv[])
{
    float M[MAX][MAX], sf[MAX], sc[MAX];
    int f, c, i, j;
    ordenMatriz(f, c);
    ingresoMatriz(M, f, c);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M, f, c);
    sumaDeFilas(M, f, c, sf);
    for(i=0; i<f; i++)
        cout<<"Suma de fila "<<i<<" = "<<sf[i]<<endl;
    sumaDeColumnas(M, f, c, sc);
    for(j=0; j<c; j++)
        cout<<"Suma de columna "<<j<<" = "<<sc[j]<<endl;
    return 0;
}

void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}
```

```
void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i, j;
    for(i=0; i<f; i++)
        for(j=0; j<c; j++)
        {
            cout<<"M["<<i<<"]["<<j<<"]:";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX], int f, int c)
{
    int i, j;
    for(i=0; i<f; i++)
    {
        for(j=0; j<c; j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}

void sumaDeFilas(float A[MAX][MAX], int f, int c, float sf[MAX])
{
    int i, j;
    for(i=0; i<f; i++)
    {
        sf[i]=0;
        for(j=0; j<c; j++)
            sf[i]=sf[i]+A[i][j];
    }
}

void sumaDeColumnas(float A[MAX][MAX], int f, int c, float sc[MAX])
{
    int i, j;
    for(j=0; j<c; j++)
    {
        sc[j]=0;
        for(i=0; i<f; i++)
            sc[j]=sc[j]+A[i][j];
    }
}
```



## SALIDA DE PANTALLA

Número de filas : 3

Número de columnas : 3

M[0][0]:12

M[0][1]:3

M[0][2]:45

M[1][0]:1

M[1][1]:10

M[1][2]:22

M[2][0]:11

M[2][1]:45

M[2][2]:85

Matriz Ingresada

12	3	45
----	---	----

1	10	22
---	----	----

11	45	85
----	----	----

Suma de fila 0 = 60

Suma de fila 1 = 33

Suma de fila 2 = 141

Suma de columna 0 = 24

Suma de columna 1 = 58

Suma de columna 2 = 152



**03.** Ingresar una matriz de  $f$  filas y  $c$  columnas y calcular su matriz transpuesta.

#### PSEUDOCODIGO

```
transpuesta(real M[], entero f, entero c, real T[])
```

```
    entero i, j
```

```
    para i  $\leftarrow$  0 hasta  $f - 1$  inc 1 hacer
```

```
        para j  $\leftarrow$  0 hasta  $c - 1$  inc 1 hacer
```

```
            T[j][i]  $\leftarrow$  M[i][j]
```

```
        fin_para
```

```
    fin_para
```

```
fin_transpuesta
```



## PSEUDOCODIGO

### Algoritmo matrices\_03

```
real M[20][20], T[20][20]
entero f, c
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz Ingresada"
reporteMatriz(M,f,c)
transpuesta(M,f,c,T)
escribir "Matriz Transpuesta"
reporteMatriz(T,c,f)
```

**fin\_algoritmo**



## CODIFICACION

```
#include<iostream>
using namespace std;
#define MAX 10
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
void transpuesta(float M[][MAX], int f, int c, float T[][MAX]);
int main(int argc, char *argv[])
{
    float M[MAX][MAX], T[MAX][MAX];
    int f, c;
    ordenMatriz(f, c);
    ingresoMatriz(M, f, c);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M, f, c);
    transpuesta(M, f, c, T);
    cout<<"Matriz Transpuesta "<<endl;
    reporteMatriz(T, c, f);
}

void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}
```

```
void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i, j;
    for(i=0; i<f; i++)
        for(j=0; j<c; j++)
        {
            cout<<"M["<<i<<"]["<<j<<"]:";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX], int f, int c)
{
    int i, j;
    for(i=0; i<f; i++)
    {
        for(j=0; j<c; j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}

void transpuesta(float M[][MAX], int f, int c, float T[][MAX])
{
    int i, j;
    for(i=0; i<f; i++)
        for(j=0; j<c; j++)
            T[j][i]=M[i][j];
}
```





### SALIDA DE PANTALLA

Número de filas : 3

Número de columnas : 2

M[0][0]:12

M[0][1]:23

M[1][0]:45

M[1][1]:67

M[2][0]:12

M[2][1]:22

Matriz Ingresada

12	23
----	----

45	67
----	----

12	22
----	----

Matriz Transpuesta

12	45	12
----	----	----

23	67	22
----	----	----



#### 04. Ingresar una matriz y reportar el mayor elemento de cada fila

##### PSEUDOCODIGO

**mayorElementoDeCadaFila**(real  $M[][]$ , entero  $f$ , entero  $c$ )

```
    entero i, j
    real may
    para i  $\leftarrow$  0 hasta  $f - 1$  inc 1 hacer
        may  $\leftarrow M[i][0]$ 
        para j  $\leftarrow$  0 hasta  $c - 1$  inc 1 hacer
            si  $M[i][j] > \text{may}$  entonces
                may  $\leftarrow M[i][j]$ 
            fin_si
        fin_para
        escribir "El mayor elemento de la fila ", i,
        " es: ", may
    fin_para
```

**fin\_mayorElementoDeCadaFila**



## PSEUDOCODIGO

### Algoritmo matrices\_04

```
real M[20][20]
entero f, c
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz Ingresada"
reporteMatriz(M,f,c)
mayorElementoDeCadaFila(M,f,c)
```

fin\_algoritmo



## CODIFICACION

```
#include<iostream>
using namespace std;
#define MAX 10
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
void mayorElementoDeCadaFila(float M[][MAX], int f, int c);

int main(int argc, char *argv[])
{
    float M[MAX][MAX];
    int f,c;
    ordenMatriz(f,c);
    ingresoMatriz(M,f,c);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M,f,c);
    mayorElementoDeCadaFila(M,f,c);
    return 0;
}

void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}
```

```
void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
        {
            cout<<"M["<<i<<"]["<<j<<"]:";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX],int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
    {
        for(j=0;j<c;j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}

void mayorElementoDeCadaFila(float M[][MAX], int f, int c)
{
    int i,j;
    float may;
    for(i=0;i<f;i++)
    {
        may=M[i][0];
        for(j=0;j<c;j++)
        {
            if(M[i][j]>may)
                may=M[i][j];
        }
        cout<<"El mayor elemento de la fila "<<i<<" es : "<<may<<endl;
    }
}
```



## SALIDA DE PANTALLA

Número de filas : 4

Número de columnas : 4

M[0][0]:12

M[0][1]:34

M[0][2]:56

M[0][3]:78

M[1][0]:34

M[1][1]:21

M[1][2]:25

M[1][3]:99

M[2][0]:10

M[2][1]:0

M[2][2]:85

M[2][3]:23

M[3][0]:45

M[3][1]:4

M[3][2]:44

M[3][3]:33

Matriz Ingresada

12    34    56    78

34    21    25    99

10    0    85    23

45    4    44    33

El mayor elemento de la fila 0 es : 78

El mayor elemento de la fila 1 es : 99

El mayor elemento de la fila 2 es : 85

El mayor elemento de la fila 3 es : 45



05. Ingresar un número de columna y eliminarla de la matriz.

#### PSEUDOCODIGO

**eliminaColumna**(real  $M[] []$ , entero  $f$ , entero  $c(R)$ , entero  $col$ )

entero  $i, j$

para  $i \leftarrow 0$  hasta  $f - 1$  inc 1 hacer

    para  $j \leftarrow col$  hasta  $c - 2$  inc 1 hacer

$M[i][j] \leftarrow M[i][j+1]$

    fin\_para

fin\_para

$c \leftarrow c - 1$

**fin\_eliminaColumna**



## PSEUDOCODIGO

### Algoritmo matrices\_05

```
real M[20][20]
entero f, c, col
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz Ingresada"
reporteMatriz(M,f,c)
hacer
    escribir "Columna a eliminar: "
    leer col
    mientras col<0 o col>=c

    eliminaColumna(M,f,c,col)
    escribir "Nueva matriz "
    reporteMatriz(M,f,c)
```

fin\_algoritmo



## CODIFICACION

```
#include<iostream>
using namespace std;
#define MAX 10
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
void eliminaColumna(float M[][MAX], int f, int &c, int col);

int main(int argc, char *argv[])
{
    float M[MAX][MAX];
    int f,c,col;
    ordenMatriz(f,c);
    ingresoMatriz(M,f,c);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M,f,c);
    do{
        cout<<"Columna a eliminar : ";
        cin>>col;
    }while(col<0 || col>=c);
    eliminaColumna(M,f,c,col);
    cout<<"Nueva Matriz "<<endl;
    reporteMatriz(M,f,c);
    return 0;
}

void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}
```

```
void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
        {
            cout<<"M["<<i<<"]["<<j<<"]:";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX],int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
    {
        for(j=0;j<c;j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}

void eliminaColumna(float M[][MAX], int f, int &c, int col)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=col;j<c-1;j++)
            M[i][j]=M[i][j+1];
        c=c-1;
}
```





## SALIDA DE PANTALLA

Número de filas : 3

Número de columnas : 3

M[0][0]:12

M[0][1]:34

M[0][2]:76

M[1][0]:44

M[1][1]:5

M[1][2]:6

M[2][0]:92

M[2][1]:9

M[2][2]:100

Matriz Ingresada

12	34	76
----	----	----

44	5	6
----	---	---

92	9	100
----	---	-----

Columna a eliminar : 1

Nueva Matriz

12	76
----	----

44	6
----	---

92	100
----	-----



**06.** Programa para ingresar una matriz de  $f$  filas y  $c$  columnas y luego ingresar el número de fila e insertar una fila en la matriz.

#### PSEUDOCODIGO

**insertaFila**(real  $M[] []$ , entero  $f(R)$ , entero  $c$ , entero  $fil$ )

entero  $i, j$

para  $i \leftarrow f-1$  hasta  $fil$  inc  $-1$  hacer

    para  $j \leftarrow 0$  hasta  $c - 1$  inc  $1$  hacer

$M[i+1][j] \leftarrow M[i][j]$

    fin\_para

fin\_para

escribir "Nueva fila"

para  $j \leftarrow 0$  hasta  $c - 1$  inc  $1$  hacer

    escribir " $M[" , fil, "]" , j, "]:"$

    leer  $M[fil][j]$

fin\_para

$f \leftarrow f + 1$

**fin\_insertaFila**



## PSEUDOCODIGO

### Algoritmo matrices\_06

```
real M[20][20]
entero f, c, fil
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz Ingresada"
reporteMatriz(M,f,c)
hacer
    escribir "fila a insertar: "
    leer fil
    mientras fil<0 o fil>f
        insertaFila(M,f,c,fil)
    escribir "Nueva matriz "
    reporteMatriz(M,f,c)
```

**fin\_algoritmo**



## CODIFICACION

```
#include<iostream>
using namespace std;
#define MAX 10
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
void insertaFila(float M[][MAX], int &f, int c,int fil);

int main(int argc, char *argv[])
{
    float M[MAX][MAX];
    int f,c,fil;
    ordenMatriz(f,c);
    ingresoMatriz(M,f,c);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M,f,c);
    do{
        cout<<"Número de fila a insertar: ";
        cin>>fil;
    }while(fil<0 || fil>f);
    insertaFila(M,f,c,fil);
    cout<<"Nueva Matriz "<<endl;
    reporteMatriz(M,f,c);
    return 0;
}

void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}
```

```
void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
        {
            cout<<"M["<<i<<"]["<<j<<"]:";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX],int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
    {
        for(j=0;j<c;j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}

void insertaFila(float M[][MAX], int &f, int c,int fil)
{
    int i,j;
    for(i=f-1;i>=fil;i--)
        for(j=0;j<c;j++)
            M[i+1][j]=M[i][j];
    cout<<"Nueva Fila"<<endl;
    for(j=0;j<c;j++)
    {
        cout<<"M["<<fil<<"]["<<j<<"]:";
        cin>>M[fil][j];
    }
    f=f+1;
}
```



## SALIDA DE PANTALLA

Número de filas : 3

Número de columnas : 2

M[0][0]:20

M[0][1]:30

M[1][0]:4

M[1][1]:7

M[2][0]:5

M[2][1]:8

Matriz Ingresada

20    30

4     7

5     8

Número de fila a insertar: 1

Nueva Fila

M[1][0]:2

M[1][1]:4

Nueva Matriz

20    30

2     4

4     7

5     8



**07.** Ingresar una matriz de  $f$  filas y  $c$  columnas y luego intercambiar 2 columnas de la Matriz. El número de las columnas a intercambiar debe ingresarse.

#### PSEUDOCODIGO

**intercambiaColumnas**(real  $M[][]$ , entero  $f$ , entero  $c$ , entero  $col1$ , entero  $col2$ )

entero  $i$

real  $temp$

para  $i \leftarrow 0$  hasta  $f - 1$  inc 1 hacer

$temp \leftarrow M[i][col1]$

$M[i][col1] \leftarrow M[i][col2]$

$M[i][col2] \leftarrow temp$

fin\_para

**fin\_intercambiaColumnas**



## PSEUDOCODIGO

### Algoritmo matrices\_07

```
real M[20][20]
entero f, c,col1,col2
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz Ingresada"
reporteMatriz(M,f,c)
hacer
    escribir "Columna1 a intercambiar : "
    leer col1
mientras col1<0 o col1>=c
hacer
    escribir "Columna2 a intercambiar : "
    leer col2
mientras col2<0 o col2>=c
intercambiaColumnas(M,f,c,col1,col2);
escribir "Nueva matriz "
reporteMatriz(M,f,c)
```

**fin\_algoritmo**



## CODIFICACION

```
#include<iostream>
using namespace std;
#define MAX 10
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
void intercambiaColumnas(float M[][MAX], int f, int c,int coll,int col2);
int main(int argc, char *argv[])
{
    float M[MAX][MAX];
    int f,c,coll,col2;
    ordenMatriz(f,c);
    ingresoMatriz(M,f,c);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M,f,c);

    do{
        cout<<"Columnal a intercambiar : ";
        cin>>coll;
    }while(coll<0 || coll>=c);
    do{
        cout<<"Columna2 a intercambiar : ";
        cin>>col2;
    }while(col2<0 || col2>=c);
    intercambiaColumnas(M,f,c,coll,col2);
    cout<<"Nueva Matriz "<<endl;
    reporteMatriz(M,f,c);
    return 0;
}
```





## CODIFICACION

```
void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}

void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
        {
            cout<<"M["<<i<<"]["<<j<<": ";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX],int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
    {
        for(j=0;j<c;j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}
```



## CODIFICACION

```
void intercambiaColumnas(float M[][MAX], int f, int c,int col1,int col2)
{
    int i;
    float temp;
    for(i=0;i<f;i++)
    {
        temp=M[i][col1];
        M[i][col1]=M[i][col2];
        M[i][col2]=temp;
    }
}
```



### SALIDA DE PANTALLA

Número de filas : 3

Número de columnas : 2

M[0][0]:12

M[0][1]:23

M[1][0]:80

M[1][1]:61

M[2][0]:3

M[2][1]:5

Matriz Ingresada

12	23
----	----

80	61
----	----

3	5
---	---

Columna1 a intercambiar : 0

Columna2 a intercambiar : 1

Nueva Matriz

23	12
----	----

61	80
----	----

5	3
---	---



**08.** Programa para ingresar una matriz de  $f$  filas y  $c$  columnas y luego ordenar las filas de la matriz.

#### PSEUDOCODIGO

```
ordenarFilas(real  $M[][]$ , entero  $f$ , entero  $c$ )

    entero  $i, j, k$ 
    real temp
    para  $i \leftarrow f-1$  hasta  $fil$  inc  $-1$  hacer
        para  $j \leftarrow 0$  hasta  $c - 2$  inc  $1$  hacer
            para  $k \leftarrow j+1$  hasta  $c - 1$  inc  $1$  hacer
                si  $M[i][j] > M[i][k]$  entonces
                    temp  $\leftarrow M[i][j]$ 
                     $M[i][j] \leftarrow M[i][k]$ 
                     $M[i][k] \leftarrow temp$ 
            fin_si
        fin_para
    fin_para
fin_ordenarFilas
```



## PSEUDOCODIGO

### Algoritmo matrices\_08

```
real M[20][20]
entero f, c, fil
ordenMatriz(f,c)
ingresoMatriz(M,f,c)
escribir "Matriz Ingresada"
reporteMatriz(M,f,c)
ordenaFilas(M,f,c)
escribir "Matriz ordenada por filas"
reporteMatriz(M,f,c)
```

**fin\_algoritmo**



## CODIFICACION

```
#include<iostream>
using namespace std;
#define MAX 10
void ordenMatriz(int &f, int &c);
void ingresoMatriz(float M[][MAX], int f, int c);
void reporteMatriz(float M[][MAX], int f, int c);
void ordenaFilas(float M[][MAX], int f, int c);
int main(int argc, char *argv[])
{
    float M[MAX][MAX];
    int f,c;
    ordenMatriz(f,c);
    ingresoMatriz(M,f,c);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M,f,c);
    ordenaFilas(M,f,c);
    cout<<"Matriz ordenada por filas"<<endl;
    reporteMatriz(M,f,c);
    return 0;
}

void ordenMatriz(int &f, int &c)
{
    do{
        cout<<"Número de filas : ";
        cin>>f;
    }while(f<=0 || f>MAX);
    do{
        cout<<"Número de columnas : ";
        cin>>c;
    }while(c<=0 || c>MAX);
}
```

```
void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
        {
            cout<<"M["<<i<<"] ["<<j<<": ";
            cin>>M[i][j];
        }
}

void reporteMatriz(float M[][MAX],int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
    {
        for(j=0;j<c;j++)
            cout<<M[i][j]<<"\t";
        cout<<endl;
    }
}

void ordenaFilas(float M[][MAX],int f, int c)
{
    int i,j,k;
    float temp;
    for(i=0;i<f;i++)
    {
        for(j=0;j<c-1;j++)
            for(k=j+1;k<c;k++)
            {
                if(M[i][j]>M[i][k])
                {
                    temp=M[i][j];
                    M[i][j]=M[i][k];
                    M[i][k]=temp;
                }
            }
    }
}
```



## SALIDA DE PANTALLA

Número de filas : 3

Número de columnas : 4

M[0][0]:70

M[0][1]:41

M[0][2]:5

M[0][3]:6

M[1][0]:82

M[1][1]:90

M[1][2]:5

M[1][3]:47

M[2][0]:3

M[2][1]:1

M[2][2]:99

M[2][3]:6

Matriz Ingresada

70	41	5	6
----	----	---	---

82	90	5	47
----	----	---	----

3	1	99	6
---	---	----	---

Matriz ordenada por filas

5	6	41	70
---	---	----	----

5	47	82	90
---	----	----	----

1	3	6	99
---	---	---	----



09. Programa que ingresa el orden de una Matriz cuadrada y generarla y luego hacer lo siguiente:

- a) Calcula la suma de los elementos de la diagonal principal.
- b) Calcula el promedio de los elementos de la diagonal secundaria.

PSEUDOCODIGO	
	<pre>ordenMatriz(entero n(R))      hacer         escribir "Orden de la matriz cuadrada: "         leer n         mientras n&lt;=0 o n &gt; 20      fin_ordenMatriz</pre>





## PSEUDOCODIGO

```
real sumaDiagonalPrincipal(real M[][], entero n)
    entero i, j

    real s ← 0
    para i ← 0 hasta n-1 inc 1 hacer
        para j ← 0 hasta n-1 inc 1 hacer
            si i = j entonces
                s ← s + M[i][j]
            fin_si
        fin_para
    fin_para
    retornar s

fin sumaDiagonalPrincipal
```



## PSEUDOCODIGO

**real promedioDiagonalSecundaria(real M[], entero n)**

entero i, j

real s  $\leftarrow$  0

para i  $\leftarrow$  0 hasta n-1 inc 1 hacer

para j  $\leftarrow$  0 hasta n - 1 inc 1 hacer

si  $i + j = n - 1$  entonces

s  $\leftarrow$  s + M[i][j]

fin\_si

fin\_para

fin\_para

retornar s/n

**fin promedioDiagonalSecundaria**



## PSEUDOCODIGO

### Algoritmo matrices\_09

```
real M[20][20]
entero n
ordenMatriz(n)
ingresoMatriz(M,n,n)
escribir "Matriz ingresada"
reporteMatriz(M,n,n)
escribir "suma diagonal principal:",
sumaDiagonalPrincipal(M,n)
escribir "promedio diagonal secundaria: ",
promedioDiagonalSecundaria(M,n)
```

**fin\_algoritmo**



## CODIFICACION

```
void reporteMatriz(float M[][MAX], int f, int c);
float sumaDiagonalPrincipal(float M[][MAX], int n);
float promedioDiagonalSecundaria(float M[][MAX], int n);

int main(int argc, char *argv[])
{
    float M[MAX][MAX];
    int n;
    ordenMatriz(n);
    ingresoMatriz(M,n,n);
    cout<<"Matriz Ingresada"<<endl;
    reporteMatriz(M,n,n);
    cout<<"Suma de la diagonal principal : "<<sumaDiagonalPrincipal(M,n)<<endl;
    cout<<"Promedio de los elementos de la diagonal secundaria: "<<
    promedioDiagonalSecundaria(M,n)<<endl;
    return 0;
}

void ordenMatriz(int &n)
{
    do{
        cout<<"Orden de la matriz cuadrada: ";
        cin>>n;
    }while(n<=0 || n>MAX);
}

void ingresoMatriz(float M[][MAX], int f, int c)
{
    int i,j;
    for(i=0;i<f;i++)
        for(j=0;j<c;j++)
        {
            cout<<"M["<<i<<"]["<<j<<"]:";
            cin>>M[i][j];
        }
}
```



CODIFICACION	SALIDA DE PANTALLA
<pre>void reporteMatriz(float M[][MAX],int f, int c) {     int i,j;     for(i=0;i&lt;f;i++)     {         for(j=0;j&lt;c;j++)             cout&lt;&lt;M[i][j]&lt;&lt;"\t";         cout&lt;&lt;endl;     } }  float sumaDiagonalPrincipal(float M[][MAX], int n) {     int i,j;     float s=0;     for(i=0;i&lt;n;i++)         for(j=0;j&lt;n;j++)             if(i==j)                 s=s+M[i][j];     return s; }  float promedioDiagonalSecundaria(float M[][MAX], int n) {     int i,j;     float s=0;     for(i=0;i&lt;n;i++)         for(j=0;j&lt;n;j++)             if(i+j==n-1)                 s=s+M[i][j];     return s/n; }</pre>	<p>Orden de la matriz cuadrada: 2 M[0][0]:23 M[0][1]:12 M[1][0]:2 M[1][1]:24 Matriz Ingresada 23    12 2    24 Suma de la diagonal principal : 47 Promedio de los elementos de la diagonal secundaria: 7</p>