



**... Vectores**

- `String[ ] nombres`
- `Nombres = new String[4]`
- `Nombres[3] = "Antonio"`

# Vectores - Vectores Paralelos

- Dos o más arreglos que utilizan el mismo subíndice para acceder a elementos de distintos arreglos, se denominan arreglos paralelos. Estos arreglos pueden procesarse simultáneamente.

- Ejemplo:

Se tienen dos arreglos. El primero contiene nombres de personas y el segundo contiene la zona a la fecha sobre 65 puntos. Obtener a partir de estos un tercer arreglo que contenga lo que le hace falta para ganar la clase. *La zona máxima es de 65 puntos.*

Programa Raspadito

BEGIN

MAIN

BEGIN

// declaracion

string[] nombres = new string[5]

double[] zonas = new double[5]

Double[] final = new double[5]

For (int i=0; i<5; i++)

    write("Ingrese nombre ")

    read(nombres[i])

    write("Ingrese zona")

    read(zonas[i])

Endfor

// calculo

For (int j=0; j<5; j++)

    final[j] = 65 - zonas[j]

Endfor

// mostrar

For (int k=0; k < 5 ; k++)

    write("A "+nombres[k]+ " le falta "+final[k]+ " para ganar raspadito (65)")

endfor

END

END

# Matrices



Introducción a la Programación

# Matrices

- Resolvamos el siguiente problema:

Un instituto desea controlar los resultados de los alumnos en las distintas asignaturas de la facultad de Ingeniería. El programa debe ingresar las calificaciones de los alumnos y **visualizar en pantalla la media de notas por alumno y la media de notas por asignatura.**

Las asignaturas están codificadas de 1 a 6 y hay 30 alumnos.

# Matrices

- Es posible resolver este problema con lo visto hasta ahora?
- Para realizar el anterior programa , debemos trabajar con una tabla (o matriz o arreglo bidimensional)
- Un arreglo bidimensional (tabla o matríz) es un arreglo con 2 índices.
  - Fila
  - columnna

# Matrices

- Un arreglo bidimensional (tabla o matriz) es un arreglo con 2 índices.
  - Fila
  - Columna
- Al igual que los vectores deben ser de un solo tipo de dato
- Es una estructura de datos que se identifica con un nombre (identificador)
- Debe especificarse los tamaños de cada dimensión
  - Cantidad de filas
  - Cantidad de columnas
- Los índices de filas y columnas inician en 0



# Matrices

Mi\_Matriz

Nombre de  
la variable

9	5	6	2	4	8	3
9	5	6	2	4	8	3
<b>8</b>	5	6	2	4	8	3

Posición : **2,0**  
**Posicion: fila, columna**  
Contenido : `Mi_matriz[2,0] = 8`

# Matrices - Declaraciones

- Sintaxis declaración

```
tipo_dato [,] nombre_arreglo
```

- Sintaxis de inicialización

```
Nombre_arreglo = new tipo_dato[filas, columnas]
```

- Ejemplo: matriz de tipo double de 3 filas y 4 columnas

```
double [,] numeros  
Numeros = new double[3,4]  
Double[,]  
Notas = new double[30, 6]
```

# Matrices – Acceso a un elemento

- Al igual que el vector, para acceder a un elemento es necesario indicar la posición exacta, es decir la fila y la columna del elemento.
- Sintaxis

```
nombre_matriz [fila,columna]
```

- Ejemplo: mostrar el contenido de la primera fila y la primera columna.

```
Console.WriteLine(mi_matriz[0,0])
```

- Ejemplo: asignar valor a la fila 0 y columna 4

```
mi_matriz[0,4] = 100
```

# Matrices - Operaciones

- Recorrer una matriz y mostrar sus elementos

```
int fila, col;
For (fila = 0; fila < cant_filas; fila++){
    For (col = 0; col < cant_cols; col++){
        Console.WriteLine(mi_matriz[fila,col]);
    }
}
// donde cant_cols y cant_filas son las dimensiones de
//la matriz
```

El recorrido de las matrices puede realizarse tanto por filas como por columnas, de acuerdo al problema a resolver.

0,0  
0,1  
0,2  
0,3  
0,4  
0,5  
1,0  
1,1  
1,2  
1,3  
1,4  
1,5

# Matrices - Operaciones

- Recorrer una matriz y asignar un valor de 100 a cada elemento.

```
int fila, col;
For (fila = 0; fila < cant_filas; fila++){
    For (col = 0; col < cant_cols; col++){
        mi_matriz[fila,col] = 0;
    }
}
// donde cant_cols y cant_filas son las dimensiones de
//la matriz
```

El recorrido de las matrices puede realizarse tanto por filas como por columnas, de acuerdo al problema a resolver.

# Matrices - Ejemplos resueltos

Un instituto desea controlar los resultados de los alumnos en las distintas asignaturas de la facultad de Ingeniería. El programa debe ingresar las calificaciones de los alumnos y visualizar en pantalla la media de notas por alumno

Las asignaturas están codificadas de 1 a 6 y hay 30 alumnos.

1	2	3	4	5	6
86	87	84	76	77	70
32	33	30	99	100	93
67	68	65	87	88	81
87	88	85	98	99	92
98	99	96	83	84	77
57	58	55	92	93	86

# Matrices - Ejemplos resueltos

```
ProgramaNotas
```

```
BEGIN
```

```
const int max_filas = 30
```

```
const int max_cols = 6
```

```
// declaracion
```

```
double[,] notas
```

```
double[] promAlumnos
```

```
double[] promMateria
```

```
    MAIN()
```

```
    BEGIN
```

```
    int i,j
```

```
    double suma, prom
```

```
    notas = new double[max_filas, max_cols]
```

```
    promAlumnos = new double[max_filas]
```

```
    promMateria = new double[max_cols]
```

```
    suma = 0
```

```
    // ingreso de datos
```

```
    for (i = 0; i < max_filas; i++)
```

```
    ,
```

```
        for (j = 0; j < max_cols; j++)
```

```
            write("ingrese nota no. "+j+ para alumno no. "+i)
```

```
            read(notas[i,j])
```

```
    endfor
```

# Matrices - Ejemplos resueltos

```
...
    // inicializar
    for (i = 0; i < max_filas; i++)
        promAlumnos[i] = 0
    endfor
    // calcular promedio por alumno
    for (i = 0; i < max_filas; i++)
        for (j = 0; j < max_cols; j++)
            promAlumnos[i] = promAlumnos[i]+notas[i,j]
        endfor
        promAlumnos[i] = promAlumnos[i] / max_cols
        write("Promedio del alumno es " + promAlumnos[i])
    endfor

END

END
```

## Ejercicio

la media de notas por asignatura.