



# Algoritmia y Programación

Facultad de Ingeniería

**Arreglos Bidimensionales**

Universidad Autónoma de Occidente  
[www.uao.edu.co](http://www.uao.edu.co) Cali, 26 de Octubre de 2015



## OBJETIVO DE APRENDIZAJE

Que el estudiante conozca sobre la creación y uso de los arreglos bidimensionales, también llamados **matrices**, y cómo su utilización es necesaria cuando se van a manejar conjuntos numerosos de datos.



## EJERCICIO

Construir un aplicativo que almacene las 4 notas de corte, de cada uno de los 20 estudiantes de un grupo de Algoritmia y Programación.

Posteriormente, el aplicativo debe permitir el promedio aritmético de cada estudiante.

### Apuesta:

Almacenar todas las notas en una sola estructura.



# PROPUESTA

Nombre del Arreglo : Datos

		Notas (Segundo Subíndice)			
Estudiantes (Primer Subíndice)		Columna0	Columna1	Columna2	Columna3
	Fila0	Datos <sub>0,0</sub>	Datos <sub>0,1</sub>	Datos <sub>0,2</sub>	Datos <sub>0,3</sub>
	Fila1	Datos <sub>1,0</sub>	Datos <sub>1,1</sub>	Datos <sub>1,2</sub>	Datos <sub>1,3</sub>
	....				
	Fila18	Datos <sub>18,0</sub>	Datos <sub>18,1</sub>	Datos <sub>18,2</sub>	Datos <sub>18,3</sub>
	Fila19	Datos <sub>19,0</sub>	Datos <sub>19,1</sub>	Datos <sub>19,2</sub>	Datos <sub>19,3</sub>

Datos[i, j] hace referencia a la Nota en la fila i y columna j.

**Ejemplo:** Datos[3, 1] hace referencia a la segunda Nota(Columna 1) del cuarto Alumno (Fila 3)



## Definición

Un arreglo bidimensional (ó matriz) es un tipo especial de variable que permite almacenar más de un valor al mismo tiempo.

La cantidad de información esta determinada por un valor finito.

Es necesario definir el numero de filas y el de columnas.





## CARACTERISTICAS

Estructura de almacenamiento estática (se fija su capacidad máxima), que permite referenciar datos con un mismo nombre, siendo posible acceder a ellos por el índice de la fila y el índice de la columna.

Es homogénea; es decir, contiene solo un tipo de dato.

Su aplicación se enmarca al uso masivo de datos, mediante la utilización de los ciclos.



Universidad Autónoma de Occidente - Cali

# Estrategia Pedagógica

Aprender Haciendo

Manos a la obra



## RETO

Construir en C# la matriz para almacenar los Datos del  
Aplicativo solicitado:

Almacenar las 4 notas de corte, de cada uno de los  
20 estudiantes de un grupo de Algoritmia y  
Programación.





## RECORDANDO LA SOLUCIÓN:

Nombre del Arreglo : Datos

		Notas (Segundo Subíndice)			
Estudiantes (Primer Subíndice)		Columna0	Columna1	Columna2	Columna3
	Fila0	Datos <sub>0,0</sub>	Datos <sub>0,1</sub>	Datos <sub>0,2</sub>	Datos <sub>0,3</sub>
	Fila1	Datos <sub>1,0</sub>	Datos <sub>1,1</sub>	Datos <sub>1,2</sub>	Datos <sub>1,3</sub>
	....				
	Fila18	Datos <sub>18,0</sub>	Datos <sub>18,1</sub>	Datos <sub>18,2</sub>	Datos <sub>18,3</sub>
	Fila19	Datos <sub>19,0</sub>	Datos <sub>19,1</sub>	Datos <sub>19,2</sub>	Datos <sub>19,3</sub>

Número de Estudiantes = Número de Filas.

Número de Notas = Número de Columnas.

Número Total de Datos = Número de Filas \* Número de Columnas.



# ARREGLOS BIDIMENSIONALES EN C#

Declaración, creación y dimensionamiento:

```
Tipo [ , ] nombreMatriz = new Tipo [nFilas, nColumnas];
```

Declarar, crear y dimensionar el arreglo  
bidimensional visto antes:

```
double [ , ] Datos = new double [20, 4];
```



# INICIALIZACIÓN DE ARREGLOS BIDIMENSIONALES:

Se los inicializa con un conjunto de conjuntos:

Tipo `[ , ]` nombreMatriz = { {.....},{...},{.....} } ;

**Nota:** cada elemento es, a su vez, otro conjunto.

Ejemplo: Para guardar los datos de 4 estudiantes que tiene 2 notas, cada uno →

`double[ , ]` notas = { {3.4, 2.5}, {3.8, 4.3},  
{4.1, 4.6}, {3.9, 4.0} } ;



# RECORRIDOS EN ARREGLOS BIDIMENSIONALES

## A NIVEL DE ANÁLISIS:

Procesar para cada fila ( $\text{Contador filas} < \text{Número de filas}$ )

Procesar para cada columna ( $\text{Contador cols} < \text{Número de cols}$ )

Procesamiento solicitado

Repetir para la siguiente columna

Repetir para la siguiente fila



# RECORRIDOS EN ARREGLOS BIDIMENSIONALES

PARA IMPLEMENTACIÓN EN C#, TENER EN CUENTA:

- Número de filas: `nombreMatriz.GetLength( 0 )`
- Número de columnas: `nombreMatriz.GetLength( 1 )`
- Número total de datos: `nombreMatriz.Length`

Siendo **nombreMatriz** el nombre de la matriz.





# LLENAR Ó LEER UN ARREGLO BIDIMENSIONAL

// Declaración, creación y dimensionamiento de la matriz de notas:

```
double[ , ] datos = new double[ numFilas, numColumnas ];
```

// Lectura de las notas del ejercicio base:

```
for( int contFila = 0; contFila < datos.GetLength(0); contFila ++ ) {  
    for( int contColumna = 0; contColumna < datos.GetLength(1); contColumna ++ ) {  
        datos[contFila, contColumna] = double.Parse( Interaction.InputBox(  
            "Digite el elemento actual:" ) );  
    }  
}
```



## REPORTE DE UN ARREGLO BIDIMENSIONAL

```
public String generarReporte( double[ , ] matriz ) {  
    String reporte = "" ;  
    for( int contFila = 0; contFila < matriz.GetLength(0); contFila ++ ) {  
        for( int contColumna = 0; contColumna < matriz.GetLength(1); contColumna ++ ) {  
            reporte += matriz[contFila, contColumna] + "\\t" ; // Tabule.  
        }  
        reporte += "\\n" ; // Cambio de fila: reporte en la siguiente línea.  
    }  
    return reporte ;  
}
```



## REPORTE DE PROMEDIOS

```
public String reportarPromedios( double[ , ] notas ) {  
    String reporte = "" ;  
    double sumaEstudiante ;  
    for( int contFila = 0; contFila < notas.GetLength(0); contFila ++ ) {  
        sumaEstudiante = 0.0;  
        for( int contColumna = 0; contColumna < notas.GetLength(1); contColumna ++ ) {  
            sumaEstudiante += notas[contFila, contColumna];  
        }  
        reporte += "Promedio " + (contFila + 1) + " = " + sumaEstudiante / notas.GetLength(1)  
            + "\n" ;  
    } return reporte; }
```