

# EJERCICIOS DE JAVA

## Ejercicio 1. Declaración, definición y uso de la clase Matriz.

1. Escribir una clase llamada `Matriz`, en el fichero `Matriz.java`, que implemente las estructuras de datos y métodos necesarios para manejar matrices cuadradas de números enteros, cuya capacidad puede definir el usuario.

Debe incluir los siguientes métodos:

- `void asignarDatos(int[] vector)`, rellena el contenido de la matriz por filas con los datos pasados desde un vector unidimensional.
  - `void suma(Matriz m)`, suma los valores de la propia clase con los valores de una matriz pasada como argumento.
  - `void producto(Matriz m)`, multiplica los valores de la propia clase con los valores de una matriz pasada como argumento.
  - `void traspuesta()`, calcula la traspuesta de la matriz.
  - `void mostrar()`, muestra el contenido de la matriz.
2. Escribir un programa Java orientado a objetos, llamado `MainMatriz.java`, que realice las siguientes operaciones:
    - Crear dos matrices de tamaño 3x3.
    - Rellenar la primera con los datos: 3, 2, 1, 1, 2, 3, 2, 3, 1.
    - Rellenar la segunda con los datos: 1, 1, 2, 2, 1, 1, 1, 2, 1.
    - Mostrar el contenido de ambas matrices.
    - Multiplicar la primera por la segunda y mostrar el resultado.
    - Sumar la primera por la segunda y mostrar el resultado.
    - Calcular la traspuesta de la primera matriz y mostrar el resultado.

### 1.1

```
class Matriz
{
    // Atributos de la clase
    private int tamanno ;
    private int[][] datos ;

    // Constructor
    public Matriz ( )
    {
        tamanno = 3 ;
        datos = new int [tamanno][tamanno] ;
    } // Matriz
```

```

public Matriz (int tam)
{
    tamanno = tam ;
    datos = new int [tam][tam] ;
} // Matriz

// Metodos
public int getTamanno ( )
{
    return tamanno;
} // getTamanno

public void asignarDatos ( int[] d )
{
    int cont = 0 ;

    if ( d.length < tamanno*tamanno )
    {
        System.out.println ( "No hay datos suficientes" ) ;
    }
    for ( int i = 0 ; i < tamanno ; i++ )
    {
        for ( int j = 0 ; j < tamanno ; j++ )
        {
            datos[i][j] = d[cont] ;
            cont++ ;
        }
    }
} // asignarDatos

public int getDatos ( int i , int j )
{
    if ( i < 0 || i >= tamanno || j < 0 || j >= tamanno )
    {
        System.out.println ( "No hay datos suficientes" ) ;
        return -1 ;
    }
    else
    {
        return datos[i][j] ;
    }
} // getDatos

public void suma ( Matriz m )
{
    if ( m.getTamanno() != tamanno )
    {
        System.out.println ( "Tamanos no compatibles" ) ;
    }
    else
    {
        for ( int i = 0 ; i < tamanno ; i++ )
        {
            for ( int j = 0 ; j < tamanno ; j++ )
            {
                datos[i][j] += m.getDatos( i , j ) ;
            }
        }
    }
}

```

```

public void producto ( Matriz m )
{
    int aux ;
    int[][] datosAux = new int[tamanno][tamanno] ;

    if ( m.getTamanno() != tamanno )
    {
        System.out.println ( "Tamanos no compatibles" ) ;
    }
    else
    {
        for ( int i = 0 ; i < tamanno ; i++ )
        {
            for ( int j = 0 ; j < tamanno ; j++ )
            {
                datosAux[i][j] = datos[i][j] ;
            }
        }

        for ( int i = 0 ; i < tamanno ; i++ )
        {
            for ( int j = 0 ; j < tamanno ; j++ )
            {
                aux = 0 ;
                for ( int k = 0 ; k < tamanno ; k++ )
                {
                    aux += datosAux[i][k] * m.getDato(k,j) ;
                }
                datos[i][j] = aux ;
            }
        }
    }
} // producto

```

```

public void traspuesta ( )
{
    int aux ;

    for ( int i = 0 ; i < tamanno ; i++ )
    {
        for ( int j = 0 ; j < tamanno ; j++ )
        {
            if ( i < j )
            {
                aux = datos[i][j] ;
                datos[i][j] = datos[j][i] ;
                datos[j][i] = aux ;
            }
        }
    }
}

```

```

public void mostrar ( )
{
    for ( int i = 0 ; i < tamanno ; i++ )
    {
        for ( int j = 0 ; j < tamanno ; j++ )
        {
            System.out.print( datos[i][j] + "\t" ) ;
        }
        System.out.println ( ) ;
    }
    System.out.println();
} // mostrar

} // class

```

## 1.2

```

class PpalMatrices
{
    public static void main ( String args[] )
    {
        Matriz m1 = new Matriz ( 3 ) ;
        Matriz m2 = new Matriz ( 3 ) ;
        int[] v1 = { 3 , 2 , 1 , 1 , 2 , 3 , 2 , 3 , 1 } ;
        int[] v2 = { 1 , 1 , 2 , 2 , 1 , 1 , 1 , 2 , 1 } ;
        int[] v3 = { 1 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 1 } ;
        m1.asignarDatos ( v1 ) ;
        m2.asignarDatos ( v2 ) ;
        m1.mostrar ( ) ;
        m2.mostrar ( ) ;
        m1.producto(m2);
        m1.mostrar ( ) ;
        m1.suma ( m2 ) ;
        m1.mostrar ( ) ;
        m1.traspuesta ( ) ;
        m1.mostrar ( ) ;
    }
}

```