



UNIVERSIDAD DE ALMERÍA

Grado en Ingeniería Informática

Introducción a la Programación

2021-2022



Tema 2. Introducción a los Arrays

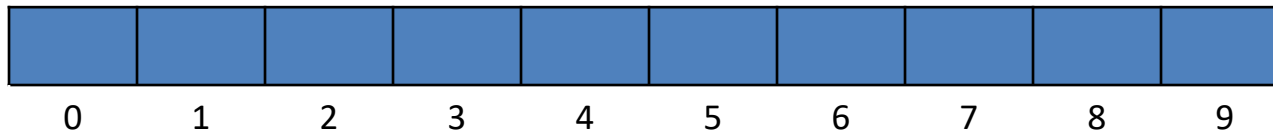
Ejercicios

- Ejercicios Arrays unidimensionales : 1,2 y 3
- Ejercicios Arrays multidimensionales: 4-10

Ejercicios Arrays

Ejercicio 1. Implemente los siguientes métodos que reciben un array de enteros como parámetro y (1) lo rellenan con el valor de la posición, (2) con un valor pasado como parámetro o (3) con el valor de potencias de dos.

```
public static void completarArrayPosicion(int [] array)  
public static void completarArrayValor(int [] array, int valor)  
public static void completarArrayPotenciasDos(int [] array)
```



Ejercicios Arrays

```
main(){  
    int [] array = new int[10];  
    completarArrayPosicion(array);  
    mostrarArraySinIndices(array);  
}
```

Ejercicio 1. Implemente los siguientes métodos que reciben un array de enteros como parámetro y (1) lo rellenan con el valor de la posición, (2) con un valor pasado como parámetro o (3) con el valor de potencias de dos.

```
public static void completarArrayPosicion(int [] array)
```

Salida:

0	1	2	3	4	5	6	7	8	9
i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Ejercicios Arrays

```
main(){  
    int [] array = new int[10];  
    completarArrayValor(array, 5);  
    mostrarArraySinIndices(array);  
}
```

Ejercicio 1. Implemente los siguientes métodos que reciben un array de enteros como parámetro y (1) lo rellenan con el valor de la posición, (2) con un valor pasado como parámetro o (3) con el valor de potencias de dos.

```
public static void completarArrayValor(int [] array, int valor)
```

valor = 5

Salida:

5	5	5	5	5	5	5	5	5	5
i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Ejercicios Arrays

```
main(){  
    int [] array = new int[10];  
    completarArrayPotenciasDos(array);  
    mostrarArraySinIndices(array);  
}
```

Ejercicio 1. Implemente los siguientes métodos que reciben un array de enteros como parámetro y (1) lo rellenan con el valor de la posición, (2) con un valor pasado como parámetro o (3) con el valor de potencias de 2.

```
public static void completarArrayPotenciasDos(int [] array)
```

2^i

Salida:

1	2	4	8	16	32	64	128	256	512
i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Ejercicios Arrays

```
main(){  
    double [] array = {2.0,5.3,8.5,9.7,2.5,5.1,6.0,9.0,1.0,2.3};  
    double valor=2.5;  
    int pos= buscarEnArray (array, valor);  
    System.out.println("El elemento " + valor + " esta en la  
                        posicion " +pos);  
}
```

Ejercicio 2. Implemente un método que reciba como parámetro un array de reales y compruebe si un valor real también dado como parámetro está contenido en el array. El método devolverá un entero que indique la posición en la que está el valor a encontrar o -1 en caso de no estar contenido en el array.

```
public static int buscarEnArray( double [] array, double valor)
```

valor = 2.5

-> devuelve 4

2.0	5.3	8.5	9.7	2.5	5.1	6.0	9.0	1.0	2.3
i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Ejercicios Arrays

```
main(){  
    double [] array = {2.0,5.3,8.5,9.7,2.5,5.1,6.0,9.0,1.0,2.3};  
    double valor=3.0;  
    int pos= buscarEnArray (array, valor);  
    System.out.println("El elemento " + valor + " esta en la  
                        posicion " +pos);  
}
```

Ejercicio 2. Implemente un método que reciba como parámetro un array de reales y compruebe si un valor real también dado como parámetro está contenido en el array. El método devolverá un entero que indique la posición en la que está el valor a encontrar o -1 en caso de no estar contenido en el array.

```
public static int buscarEnArray( double [] array, double valor)
```

valor = 3.0

-> devuelve -1

2.0	5.3	8.5	9.7	2.5	5.1	6.0	9.0	1.0	2.3
i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Ejercicios Arrays

Ejercicio 3. Implemente un método que reciba un array de booleanos y que devuelva el número de valores que están a true. Para procesar el array de booleanos se debe hacer, recorriendo desde el final hasta el principio.

Implemente otro método, que dado un valor entero como parámetro, devuelva un array de booleanos con tamaño ese valor dado como parámetro y rellenado con las posiciones pares a true y las impares a false.

```
public static int numeroDeVerdaderos (boolean [] array)  
public static boolean [] arrayBooleanos (int tamano)
```

Ejercicios Arrays

```
main(){  
    boolean [] array = arrayBooleanos (10);  
}
```

Ejercicio 3.

Implemente un método, que dado un valor entero como parámetro, devuelva un array de booleanos con tamaño ese valor dado como parámetro y rellenado con las posiciones pares a true y las impares a false.

```
public static boolean [] arrayBooleanos (int tamano)
```

tamano = 10

-> devuelve este array:



true	false	true	false	true	false	true	false	true	false
i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Ejercicios Arrays

```
main(){  
    boolean [] array = arrayBooleanos (10);  
    int num= numeroDeVerdaderos (array);  
}
```

Ejercicio 3. Implemente un método que reciba un array de booleanos y que devuelva el número de valores que están a true. Para procesar el array de booleanos se debe hacer, recorriendo desde el final hasta el principio.

```
public static int numeroDeVerdaderos (boolean [] array)
```

-> devuelve 5

true	false	true	true	false	true	false	false	false	true
i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9

Ejercicios Arrays

Ejercicio 4. Implemente dos métodos que reciben una matriz de enteros como parámetro y la rellenan por filas o por columnas con número consecutivos empezando desde el 0.

```
public static void completarMatrizPorFilas (int [][] matriz)
```

```
public static void completarMatrizPorColumnas (int [][] matriz)
```

Ejercicios Arrays

```
main(){  
    int [][] matriz = new int[3][4];  
    completarMatrizPorFilas(matriz);  
    mostrarMatrizEnteros(matriz);  
}
```

Ejercicio 4. Implemente dos métodos que reciben una matriz de enteros como parámetro y la rellenan por filas o por columnas con número consecutivos empezando desde el 0.

public static void completarMatrizPorFilas (int [][] matriz)

Salida:

0	1	2	3
4	5	6	7
8	9	10	11

Ejercicios Arrays

```
main(){  
    int [][] matriz = new int[3][4];  
    completarMatrizPorColumnas(matriz);  
    mostrarMatrizEnteros(matriz);  
}
```

Ejercicio 4. Implemente dos métodos que reciben una matriz de enteros como parámetro y la rellenan por filas o por columnas con número consecutivos empezando desde el 0.

public static void completarMatrizPorColumnas (int [][] matriz)

Salida:

0	3	6	9
1	4	7	10
2	5	8	11

Ejercicios Arrays

```
main(){  
    double [][] matrizE5 = { {0.0, 3.0, 6.0, 1.0},  
                               {1.0, 2.0, 0.0, 1.0},  
                               {1.0, 3.0, 6.0, 1.1},  
                               };  
    double sum = obtenerSumaElementosMatriz(matrizE5);  
    System.out.println("La suma es " + sum);  
}
```

Ejercicio 5. Implemente un método que reciba como parámetro una matriz de reales y devuelva la suma de todos sus elementos.

```
public static double obtenerSumaElementosMatriz ( double [][] matriz )
```

0.0	3.0	6.0	1.0
1.0	2.0	0.0	1.0
1.0	3.0	6.0	1.1

-> devuelve 25.1

Ejercicios Arrays

```
main(){  
    //Tomamos matrizE5 del ejercicio 5  
  
    double [][] matrizMul = matrizMultiplicada(matrizE5, 2.0);  
}
```

Ejercicio 6. Implemente un método que reciba como parámetro una matriz de reales y un factor de multiplicación, y devuelva una **nueva matriz** cuyos elementos serán los contenidos en la matriz recibida como parámetro multiplicados por el factor dado.

```
public static double [][] matrizMultiplicada(double [][] matriz, double valor)
```

Entrada:

0.0	3.0	1.0	1.0
1.0	2.0	0.0	1.0
1.0	3.0	2.0	1.1

valor=2.0

Ejercicios Arrays

```
main(){  
    //Tomamos matrizE5 del ejercicio 5  
  
    double [][] matrizMul = matrizMultiplicada(matrizE5, 2.0);  
    mostrarMatrizReales(matrizMul);  
}
```

Ejercicio 6. Implemente un método que reciba como parámetro una matriz de reales y un factor de multiplicación, y devuelva una **nueva matriz** cuyos elementos serán los contenidos en la matriz recibida como parámetro multiplicados por el factor dado.

```
public static double [][] matrizMultiplicada(double [][] matriz, double valor)
```

0.0	3.0	1.0	1.0
1.0	2.0	0.0	1.0
1.0	3.0	2.0	1.1

devuelve:

0.0	6.0	2.0	2.0
2.0	4.0	0.0	2.0
2.0	6.0	4.0	2.2

Ejercicios Arrays

```
main(){  
    mostrarMatrizIrregular();  
}
```

Ejercicio 7. Implemente un método que cree una matriz irregular y la muestre en la consola.

```
public static void mostrarMatrizIrregular()
```

1	2	3	4	5
1	2	3	4	
1	2	3		
1	2			
1				

Ejercicios Arrays

```
main(){  
  
    int [][] matrizIden = generarMatrizIdentidad( 5);  
    mostrarMatrizEnteros(matrizIden);  
}
```

Ejercicio 8. Implemente un método que reciba como parámetro un valor entero que sea el número de las dos dimensiones de la matriz (cuadrada). El método creará y devolverá la correspondiente matriz cuadrada, rellenando sus posiciones para que construya la matriz identidad (elemento neutro del producto de matrices).

public static int[][] generarMatrizIdentidad(int dimension)

dimension=5 ->

	j=0	j=1	j=2	j=3	j=4
i=0	1	0	0	0	0
i=1	0	1	0	0	0
i=2	0	0	1	0	0
i=3	0	0	0	1	0
i=4	0	0	0	0	1

Ejercicios Arrays

Ejercicio 9. Implemente un método que reciba como parámetro una matriz cuadrada de enteros y devuelva una matriz (de enteros) con los valores de las diagonales principales de dicha matriz.

```
public static int[][] obtenerDiagonales( int [][] matriz)
```

entrada: matriz 5x5

	j=0	j=1	j=2	j=3	j=4
i=0	1	2	3	4	5
i=1	0	1	0	5	0
i=2	1	0	1	0	0
i=3	0	5	3	1	0
i=4	2	0	6	4	1

Ejercicios Arrays

```
main(){
    int [][] matriz = new int[5][5];
    completarMatrizPorColumnas(matriz);

    int [][] matrizDiag = obtenerDiagonales(matriz);
    mostrarMatrizEnteros(matrizDiag);
}
```

Ejercicio 9. Implemente un método que reciba como parámetro una matriz cuadrada de enteros y devuelva una matriz (de enteros) con los valores de las diagonales principales de dicha matriz.

public static int[][] obtenerDiagonales(int [][] matriz)

entrada: matriz 5x5

	j=0	j=1	j=2	j=3	j=4
i=0	1	2	3	4	5
i=1	0	1	0	5	0
i=2	1	0	1	0	0
i=3	0	5	3	1	0
i=4	2	0	6	4	1

salida: nuevamatriz 2x5

1	1	1	1	1
5	5	1	5	2

Ejercicios Arrays

Ejercicio 10. Implemente de otra forma el ejemplo 6 de matrices visto en clase de teoría en el que se muestra **qué fila** de la matriz pasada como parámetro tiene el mayor valor de suma. Utilice **Integer.MIN_VALUE** que es una constante con el menor valor que puede tomar un entero.

Implementar también otro método en Java, siguiendo este último esquema (Integer.MIN_VALUE), que muestre **qué columna** de la matriz tiene el mayor valor de suma.

```
public static void filaConMayorSumaOtroMetodo(int [][] matriz)
public static void columnaConMayorSuma(int [][] matriz)
```

Ejercicios Arrays

Ejercicio 10. Implemente de otra forma el ejemplo 6 de matrices visto en clase de teoría en el que se muestra qué fila de la matriz pasada como parámetro tiene el mayor valor de suma. Utilice `Integer.MIN_VALUE` que es una constante con el menor valor que puede tomar un entero.

```
public static void filaConMayorSumaOtroMetodo(int [][] matriz)
```

La fila 2 tiene la máxima suma de 30

i=0	0	3	6	10
i=1	1	4	7	10
i=2	8	10	1	11

Ejercicios Arrays

Ejercicio 10.

Implementar también otro método en Java, siguiendo este último esquema (Integer.MIN_VALUE), que muestre **qué columna** de la matriz tiene el mayor valor de suma.

```
public static void columnaConMayorSuma(int [][] matriz)
```

La columna 3 tiene la máxima suma de 31

i=0	0	3	6	10
i=1	1	4	7	10
i=2	8	10	1	11
	j=0	j=1	j=2	j=3

Tema2 - Ejemplo 6. Obtener qué fila tiene el mayor valor de suma de todos sus elementos.

```
public static void filaConMayorSuma(int [][] matriz){
```

```
int maxSumaFila = 0
```

```
int indiceMaxSumaFila = 0;
```

```
for (int columna = 0; columna < matriz[0].length; columna++) {  
    maxSumaFila += matriz[0][columna];  
}
```

Primer máximo es la suma
de la primera fila

```
for (int fila = 1; fila < matriz.length; fila++) {
```

```
    int sumaFilaActual = 0;
```

```
    for (int columna = 0; columna < matriz[fila].length; columna++) {
```

```
        sumaFilaActual += matriz[fila][columna];
```

```
    }
```

```
    if (sumaFilaActual > maxSumaFila) {
```

```
        maxSumaFila = sumaFilaActual;        indiceMaxSumaFila = fila;
```

```
    }
```

```
}
```

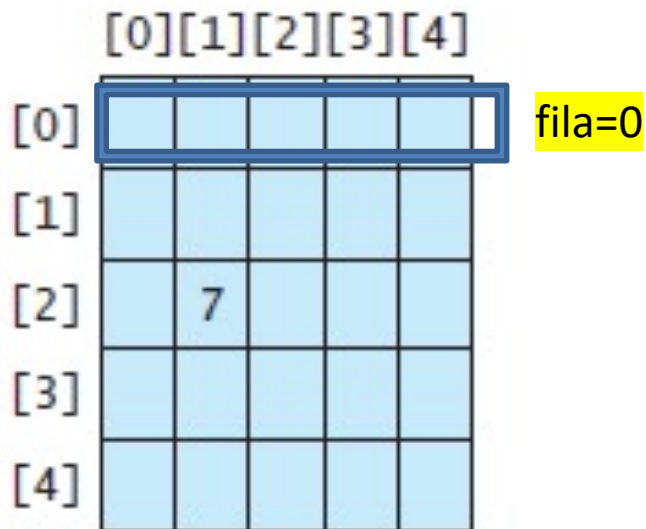
```
System.out.println("Fila " + indiceMaxSumaFila + " tiene la maxima suma de " + maxSumaFila);
```

```
}
```

Tema2

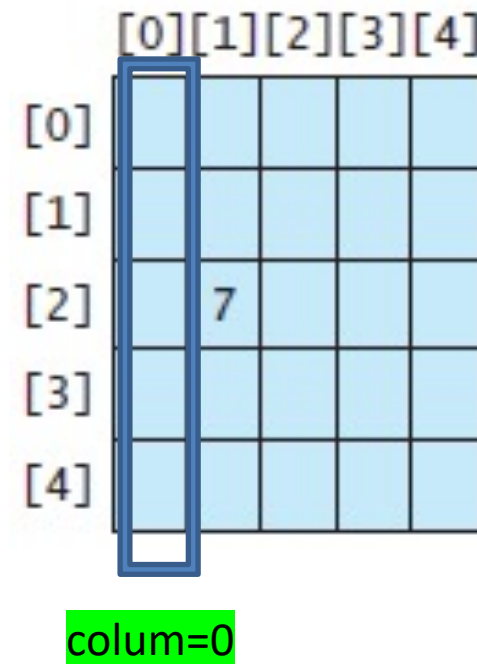
Procesamiento por filas

```
for (int fila = 0; fila < matriz.length; fila++)
    for (int column = 0; column < matriz[fila].length ; column++)
```



Procesamiento por columnas

```
for (int column = 0; column < matriz[0].length; column++)
    for (int fila = 0; fila < matriz.length; fila++)
```



¡MUCHAS GRACIAS!



UNIVERSIDAD DE ALMERÍA

