

# ACTIVIDADES DE LA UNIDAD DE TRABAJO 5 – PRO

**Autor: Derimán Tejera Fumero. 1ºDAW**

**Fecha: 13/12/2022**

## Índice

|                                   |   |
|-----------------------------------|---|
| Actividades de comprobación ..... | 2 |
| Actividades de aplicación .....   | 4 |

## Actividades de comprobación

**5.1. Una tabla puede almacenar datos de distintos tipos, como por ejemplo enteros, booleanos, reales, etcétera:**

- a) Cierto, las tablas siempre pueden almacenar datos de distintos tipos.
- b) Falso, las tablas solo pueden almacenar datos de un único tipo.**
- c) Puede almacenar datos de distintos tipos siempre que estos sean numéricos.
- d) Puede almacenar datos de distintos tipos siempre que la longitud de los datos sea idéntica.

**5.2. En Java, la numeración de los índices que determina la identificación de cada elemento de una tabla comienza en:**

- a) Cero.**
- b) Uno.
- c) Depende del tipo de dato de la tabla.
- d) Es configurable por el usuario.

**5.3 Si en una tabla de 10 elementos utilizamos el elemento con índice 11 (que se encuentra fuera de rango):**

- a) Al salir del rango de la longitud, Java redimensiona la tabla de forma automática.
- b) No es posible y produce un error.**
- c) Las tablas tienen un comportamiento circular y utilizar el índice 11 es idéntico a utilizar el índice 1.
- d) Ninguna de las anteriores respuestas es cierta.

**5.4. ¿Qué método de la clase Arrays permite realizar una búsqueda dicotómica en una tabla?**

- a) Arrays.search().
- b) Arrays.find().
- c) Arrays.binarySearch().**
- d) Cualquiera de los métodos anteriores realiza una búsqueda.

**5.5. Con respecto a las tablas, el operador new:**

- a) Destruye, crea y redimensiona tablas.
- b) Destruye y crea tablas.

**c) Crea tablas.**

d) Destruye las tablas.

**5.6. La forma de invocar al recolector de basura es:**

a) Mediante `System.garbageCollectorO` .

b) Mediante el operador `new`.

c) Mediante `Arrays.garbageCollector {}`.

**d) Ninguna de las anteriores respuestas es correcta.**

**5.7. La forma de conocer la longitud de una tabla `t` es mediante:**

a) `t.size`.

b) `t.elements`.

**c) `t.length`.**

d) `Arrays.size(t)`.

**5.8 La comparación del contenido (los elementos) de dos tablas se realiza utilizando:**

a) `Arrays.compare()`.

b) El operador `==`.

**c) `Arrays.equals()`.**

d) `Arrays.same ()`.

**5.9. ¿Qué condición tiene que cumplir una tabla para que podamos realizar búsquedas dicotómicas en ella?**

**a) Que esté ordenada.**

b) Que esté ordenada y sea una tabla de enteros.

c) Que no esté ordenada.

d) No importa si la tabla esta ordenada, lo realmente importante es que sea de algún tipo numérico.

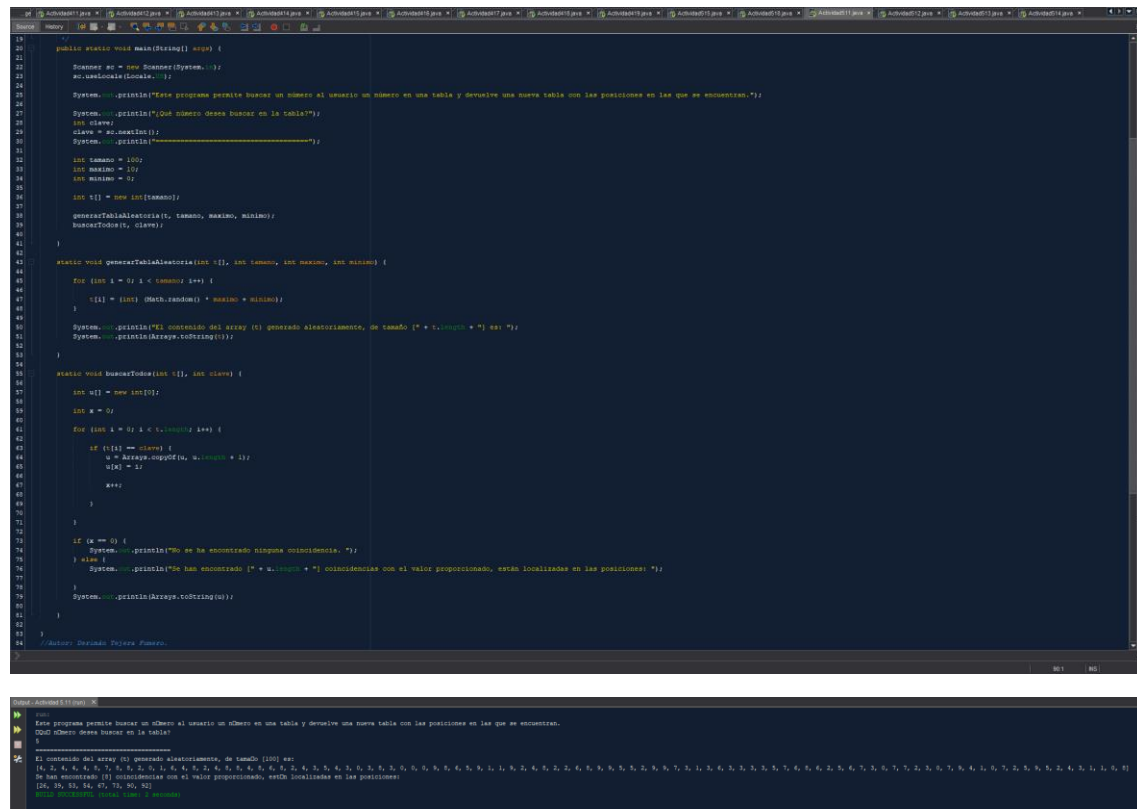
**5.10. ¿Cuál es la principal diferencia entre `Arrays.copyOf ()` y `System.arraycopy()`?**

a) No existe diferencia alguna, ambos métodos son idénticos.

b) `Arrays.copyOf ()` copia mientras `System.arraycopy ()` copia y compara.

d) `Arrays.copyOf()` crea una nueva tabla y copia en ella mientras `System.arraycopy()` solo copia entre tablas ya creadas.

5.11 Realiza la función: `int [ ] buscarTodos (int t [ ], int clave)`, que crea y devuelve una tabla con todos los índices de los elementos donde se encuentra la clave de búsqueda. En el caso de que clave no se encuentre en la tabla `t`, la función devolverá una tabla vacía.



5.12. Escribe la función void desordenar (int t[]), que cambia de forma aleatoria los elementos contenidos en la tabla t. Si la tabla estuviera ordenada, dejaría de estarlo.

```
1 //
2 * Click https://about/systemfilesystem.com/licenses/licenses-default.txt to change this license
3 * Click https://about/systemfilesystem.com/licenses/licenses-default.txt to edit this template
4 //
5 package actividad.pkg1;
6
7 import java.util.Arrays;
8 import java.util.Random;
9
10 /**
11  *
12  * @author Dorian
13  */
14 public class Actividad512 {
15
16     /**
17      *
18      * @param args the command line arguments
19      */
20     public static void main(String[] args) {
21         int t[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
22
23         System.out.println("Array con contenido ordenado:");
24         System.out.println(Arrays.toString(t));
25         System.out.println("-----");
26         desordenar(t);
27     }
28
29     /**
30      *
31      * @param t
32      */
33     static void desordenar(int t[]) {
34         Random aleatorio = new Random();
35
36         for (int i = 0; i < t.length; i++) {
37             int posAleatoria = aleatorio.nextInt(t.length);
38             int temp = t[i];
39             t[i] = t[posAleatoria];
40             t[posAleatoria] = temp;
41         }
42
43         System.out.println("Array con contenido (DES)ordenado:");
44         System.out.println(Arrays.toString(t));
45     }
46 }
47
48 //Autor: Dorian Tejera Fumero.
49
50 //
51 //Escribe la función void desordenar (int t[]), que cambia de forma aleatoria los elementos
52 //contenidos en la tabla t. Si la tabla estuviera ordenada, dejaría de estarlo.
53 //
```

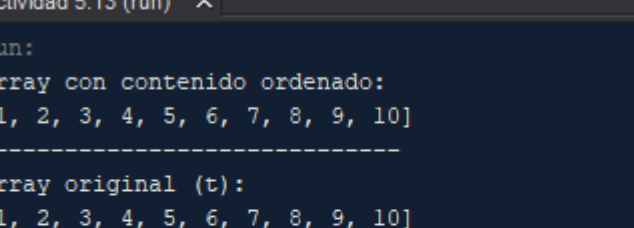
```
Output - Actividad 5.12 (run) X
run:
Array con contenido ordenado:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
-----
Array con contenido (DES)ordenado:
[5, 1, 2, 9, 7, 3, 4, 10, 8, 6]
BUILD SUCCESSFUL (total time: 0 seconds)
```

5.13. Modifica la Actividad de aplicación 5.12 para que la función no modifique la tabla que se pasa como parámetro y, en su lugar, cree y devuelva una copia de la tabla donde se han desordenado los valores de los elementos.

```

1  // Click https://about:blank to change this license
2  // Click https://about:blank to edit this template
3
4  package activited-pkg1.pkg1;
5
6  import java.util.Arrays;
7  import java.util.Random;
8
9  /**
10   *
11   * @author Doreen
12   */
13  public class Activited013 {
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19
20          int v[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
21
22          System.out.println("Array original desordenado: ");
23          System.out.println(Arrays.toString(v));
24          System.out.println("-----");
25
26          desordenar(v);
27
28      }
29
30      static void desordenar(int v[]) {
31
32          int w[] = new int[v.length];
33
34          w = Arrays.copyOf(v, v.length);
35
36          Random aleatorio = new Random();
37
38          for (int i = 0; i < w.length; i++) {
39              int posAleatorio = aleatorio.nextInt(w.length);
40              int temp = w[i];
41              w[i] = w[posAleatorio];
42              w[posAleatorio] = temp;
43          }
44
45          System.out.println("Array original (0): ");
46          System.out.println(Arrays.toString(v));
47
48          System.out.println("Array nuevo (0) desordenado: ");
49          System.out.println(Arrays.toString(w));
50
51      }
52  }
53
54  /**Autor: Doreen Prieta Powers.
55
56   *
57   * Modificó la Actividad de aplicación 0.12 para que la función no modifique la tabla que se pasa como parámetro y, en su lugar,
58   * cree y devuelva una copia de la tabla donde se han desordenado los valores de los elementos.
59   */

```



```
run:
Array con contenido ordenado:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
-----
Array original (t):
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Array nuevo (u) desordenado:
[3, 10, 7, 9, 4, 8, 2, 6, 5, 1]
BUILD SUCCESSFUL (total time: 0 seconds)
```

5.14 El ayuntamiento de tu localidad te ha encargado una aplicación que ayude a realizar encuestas estadísticas para conocer el nivel adquisitivo de los habitantes del municipio. Para ello, tendrás que preguntar el sueldo a cada persona encuestada. A priori, no conoces el número de encuestados. Para finalizar la entrada de datos, introduce un sueldo con valor -1.

Una vez terminada la entrada de datos, muestra la siguiente información:

- Todos los sueldos introducidos ordenados de forma decreciente.
- El sueldo máximo y mínimo.
- La media de los sueldos.

```
23 double salario = 0;
24 double salarioCiudadano[] = new double[10];
25 double salarioMinimo = 1050.00;
26
27 System.out.println("Este programa realiza una encuesta estadística para conocer el nivel adquisitivo de los habitantes de una comunidad.");
28
29 while (salario != -1) {
30     System.out.println("Introduce valor del sueldo del ciudadano [" + salarioCiudadano.length + "]: ");
31     salario = sc.nextDouble();
32     if (salario < salarioMinimo) {
33         System.out.println("El número no es válido, recuerda que el salario mínimo es de " + salarioMinimo + " (euros) el salario del ciudadano [" + salarioCiudadano.length + "] debe ser igual o superior.");
34     } else {
35         salarioCiudadano[salarioCiudadano.length - 1] = salario;
36         salarioCiudadano = Arrays.copyOf(salarioCiudadano, salarioCiudadano.length + 1);
37     }
38 }
39
40 ordenarArrayDecrecientemente(salarioCiudadano);
41
42 System.out.println("Valores de los sueldos:");
43
44 System.out.println(":::::::::::: DATOS ::::::::::::::");
45 System.out.println("Tamaño actual de la base de datos: " + salarioCiudadano.length + " = " + "ciudadano(s).");
46 System.out.println("Datos introducidos hasta ahora son:");
47 System.out.println(Arrays.toString(salarioCiudadano));
48 System.out.println("El sueldo mínimo es: " + salarioCiudadano[0]);
49 System.out.println("El sueldo máximo es: " + salarioCiudadano[salarioCiudadano.length - 1]);
50 mediaSueldos(salarioCiudadano);
51 System.out.println("::::::::::::");
52
53 static void ordenarArrayDecrecientemente(double salario[]) {
54     for (int i = 0; i < salario.length - 1; i++) {
55         int max = i;
56
57         for (int j = i + 1; j < salario.length; j++) {
58             if (salario[j] > salario[max]) {
59                 max = j;
60             }
61         }
62
63         if (i != max) {
64             double aux = salario[i];
65             salario[i] = salario[max];
66             salario[max] = aux;
67         }
68     }
69 }
70
71 static void mediaSueldos(double salarioCiudadano[]) {
72     double total = 0;
73     double media = 0;
74
75     for (int i = 0; i < salarioCiudadano.length - 1; i++) {
76         total = total + salarioCiudadano[i];
77     }
78
79     media = total / (salarioCiudadano.length - 1);
80     System.out.println("La media de los sueldos es: " + media);
81 }
82
83 /****/ Escrito por: Juan Torres.
```

```
Output - Actividad 5.14 (run) X
RUN:
Este programa realiza una encuesta estadística para conocer el nivel adquisitivo de los habitantes de una comunidad.
Introduce valor del sueldo del ciudadano [1]:
1050
Introduce valor del sueldo del ciudadano [2]:
1200
Introduce valor del sueldo del ciudadano [3]:
1450
Introduce valor del sueldo del ciudadano [4]:
1500
Introduce valor del sueldo del ciudadano [5]:
2000
Introduce valor del sueldo del ciudadano [6]:
2200
Introduce valor del sueldo del ciudadano [7]:
1200
Introduce valor del sueldo del ciudadano [8]:
1100
Introduce valor del sueldo del ciudadano [9]:
5200
Introduce valor del sueldo del ciudadano [10]:
-1
El número no es válido, recuerda que el salario mínimo es de 1050.0 (euros) el salario del ciudadano [10] debe ser igual o superior.

:::::::::::: DATOS ::::::::::::::
Tamaño actual de la base de datos: 9 ciudadano(s).
Datos introducidos hasta ahora son:
[5200.0, 2200.0, 2000.0, 1500.0, 1450.0, 1200.0, 1200.0, 1100.0, 1050.0, 0.0]
El sueldo máximo es: 5200.0
El sueldo mínimo es: 1050.0
La media de los sueldos es: 1877.7777777777778
::::::::::::
BUILD SUCCESSFUL (total time: 27 seconds)
```

5.15 Debes desarrollar una aplicación que ayude a gestionar las notas de un centro educativo. Los alumnos se organizan en grupos compuestos por 5 personas. Leer las notas (números enteros) del primer, segundo y tercer trimestre de un grupo. Debes mostrar al final la nota media del grupo en cada trimestre y la media del alumno que se encuentra en una posición dada (que el usuario introduce por teclado).

```

34         System.out.println(" ");
35         System.out.println(" ");
36     }
37     System.out.println(" ");
38     System.out.println(" ");
39 }
40
41
42 static void muestraMediaPorTrimestre(int notas[][], double mediaTrimestre[]) {
43     for (int i = 0; i < 3; i++) {
44         for (int j = 0; j < 5; j++) {
45             int x = 0;
46             mediaTrimestre[i] = mediaTrimestre[i] + notas[i][j];
47             x++;
48         }
49         System.out.println("Media del grupo 1 en el T° " + (i + 1) + ": " + mediaTrimestre[i] / 5);
50     }
51     System.out.println(" ");
52 }
53
54 static void muestraMediaPorAlumno(int notas[][], double mediaTrimestre[]) {
55     Scanner sc = new Scanner(System.in);
56     sc.useLocale(Locale.US);
57
58     int numAlu = 0;
59     double mediaAlumnoDelegacionado = 0;
60
61     while (numAlu != -1) {
62         System.out.println("Introduce el número del alumno que desea consultar:");
63         System.out.println("RECUERDA: Para salir, introduce: -1.");
64         numAlu = sc.nextInt();
65
66         if (numAlu < 0 || numAlu > 4) {
67             System.out.println("El número no es válido, recuerda que en el grupo 1 solo hay 5 alumnos.");
68         } else {
69             for (int i = 0; i < 3; i++) {
70                 mediaAlumnoDelegacionado = (notas[numAlu - 1][i] + notas[numAlu - 1][i+1] + notas[numAlu - 1][i+2]);
71             }
72
73             mediaAlumnoDelegacionado = mediaAlumnoDelegacionado / 3;
74
75             System.out.println(" ");
76             System.out.println("La media del alumno [" + numAlu + "] es: ");
77             System.out.println(mediaAlumnoDelegacionado);
78             System.out.println(" ");
79         }
80     }
81 }
82
83 static void muestraMediaPorAlumno(int notas[][], double mediaTrimestre[]) {
84     String pattern = "%,f%";
85     DecimalFormat decimalFormat = new DecimalFormat(pattern);
86     String formattedDouble = decimalFormat.format(mediaAlumnoDelegacionado);
87     System.out.println(formattedDouble);
88 }
89
90 }
91
92 //Autor: David Torra Fumero

```

```

Output x
Actividad 5.15 (run) x Actividad 5.15 (run) #2 x
RUN:
Tamaño actual del array es: 5
[[7, 2, 8], [6, 2, 8], [4, 2, 8], [7, 7, 8], [5, 5, 8]]

-----
| NOTAS ALUMNOS GRUPO 1: |
-----
      T1 T2 T3
Alumno: 1 -> 7 2 8
Alumno: 2 -> 6 2 8
Alumno: 3 -> 4 2 8
Alumno: 4 -> 7 7 8
Alumno: 5 -> 5 5 8

Media del grupo 1 en el T1: 5.8
Media del grupo 1 en el T2: 3.6
Media del grupo 1 en el T3: 6.0

Introduce el número del alumno que desea consultar:
RECUERDA: Para salir, introduce: -1.
1

La media del alumno [1] es: 5,67

Introduce el número del alumno que desea consultar:
RECUERDA: Para salir, introduce: -1.
5

La media del alumno [5] es: 6

Introduce el número del alumno que desea consultar:
RECUERDA: Para salir, introduce: -1.
4

La media del alumno [4] es: 7,33

Introduce el número del alumno que desea consultar:
RECUERDA: Para salir, introduce: -1.
2

La media del alumno [2] es: 5,33

Introduce el número del alumno que desea consultar:
RECUERDA: Para salir, introduce: -1.
3

La media del alumno [3] es: 4,67

Introduce el número del alumno que desea consultar:
RECUERDA: Para salir, introduce: -1.
-1
El número no es válido, recuerda que en el grupo 1 solo hay 5 alumnos.
BUILD SUCCESSFUL (total time: 15 seconds)

```



5.18 Escribe un programa que solicite los elementos de una matriz de tamaño 4x4. La aplicación debe decidir si la matriz introducida corresponde a una matriz mágica, que es aquella donde la suma de los elementos de cualquier fila o de cualquier columna valen lo mismo.

```

37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
//Autor: Damián Pérez Flores.

```

```

Output
Actividad 5.15 (run) x Actividad 5.18 (run) x
run:
Este programa verifica si una matriz que introduce por teclado el usuario es mágica o no.
Introduce los datos para la matriz...
Posición -> Fila: [1] - Columna [1]:
5
Posición -> Fila: [1] - Columna [2]:
15
Posición -> Fila: [1] - Columna [3]:
16
Posición -> Fila: [1] - Columna [4]:
2
Posición -> Fila: [2] - Columna [1]:
10
Posición -> Fila: [2] - Columna [2]:
8
Posición -> Fila: [2] - Columna [3]:
7
Posición -> Fila: [2] - Columna [4]:
13
Posición -> Fila: [3] - Columna [1]:
6
Posición -> Fila: [3] - Columna [2]:
12
Posición -> Fila: [3] - Columna [3]:
11
Posición -> Fila: [3] - Columna [4]:
9
Posición -> Fila: [4] - Columna [1]:
17
Posición -> Fila: [4] - Columna [2]:
3
Posición -> Fila: [4] - Columna [3]:
4
Posición -> Fila: [4] - Columna [4]:
14

La matriz que introdujiste es esta:
[[5, 15, 16, 2], [10, 8, 7, 13], [6, 12, 11, 9], [17, 3, 4, 14]]

o vista de otro modo:
5 15 16 2
10 8 7 13
6 12 11 9
17 3 4 14
¿La matriz es mágica?

¡FELICIDADES! La matriz es mágica.
BUILD SUCCESSFUL (total time: 18 seconds)

```