

UnidaddeTrabajo4

EJEMPLO4-5

ESTUDIOS

Ciclo Formativo: **Desarrollo de aplicaciones Web**

Modalidad: **Distancia**

Módulo:

Programación

ORGANIZACIÓN

Departamento: **Informática y Comunicaciones**

**Instituto de Educación Secundaria
“José Planes”**

C/ Maestro Pérez Abadía,
230100Espinardo –
MurciaT.968834605
30010577@murciaeduca.es
www.iesjoseplanes.es

AUTORÍA

JuanPedroMartínezGallar

LICENCIA



Esta obra está bajo una [Licencia
Creative CommonsAtribución-
NoComercial4.0
Internacional](https://creativecommons.org/licenses/by-nc/4.0/)

1 Enunciado

En este ejemplo se explica cómo hacer uso de las clases **Arrays**, **String** y del manejo de **expresiones regulares** en Java.

2 Explicación

A) Clase Arrays:

Hasta ahora hemos trabajado con arrays que no son sino colecciones de elementos del mismo tipo, p. e:

```
int a[] = {0,1,2,3,4};
```

Además, java incluye una clase estática especial denominada **Arrays** que nos permite realizar operaciones interesantes sobre los arrays que nosotros definimos, por ejemplo para ordenar un array no tendríamos que implementar un algoritmo sino invocar al método “sort”.

En el código que aparece más abajo (**PROGRAMA1**) nos aparecen varios ejemplos para manejar esta clase, permitiendo rellenar los elementos de un array con determinados valores (método **fill**), comparar 2 arrays (método **equals**), ordenar un array (método **sort**), buscar un elemento en un array ordenado (método **binarySearch**) y copiar un array en otro (método **arraycopy**).

B) Clase String:

Es la que se emplea para manejar cadenas de texto. Esta clase nos permite realizar operaciones con cadenas de texto tales como comparar 2 cadenas, determinar su orden alfabético, convertir a mayúsculas/minúsculas, reemplazar un texto en su interior...

En el código que aparece más abajo (**PROGRAMA2**) nos aparecen varios ejemplos para manejar esta clase a través de diversos métodos.

C) Expresiones regulares:

Una expresión regular es una forma de definir un patrón de búsqueda dentro de una cadena de caracteres. Se emplea para determinar por ejemplo si una cadena contiene un patrón, por ejemplo sirve para comprobar si una fecha leída tiene el formato dd-mm-aa o si una URL es correcta según el formato `protocolo://máquina/directorio/archivo`

En el código que aparece más abajo (**PROGRAMA3**) nos aparecen varios ejemplos para manejarlas en String a partir de los métodos **matches** y **split**.

3 Programa

PROGRAMA1:



Solución

Aquí puedes usar el **CÓDIGO**.

```
package com.mycompany.proyectoarraystring;

import java.util.Arrays;
import java.util.Scanner;

public class Principal
{
    public static void main(String[] args)
    {

        int miarray[] = {1,2,3,4,5,6,7,8,9,10};
        int miarray2[] = {1,22,3,44,5,56,7,68,9,510};
        int miarray3[] = new int[10];

        for (int i = 0; i < miarray.length; i++)
            System.out.print(miarray[i] + " ");

        System.out.println("");

        Arrays.fill(miarray, -1); //Rellena todo el array con -1

        for (int i = 0; i < miarray.length; i++)
            System.out.print(miarray[i] + " ");

        System.out.println("");

        if (Arrays.equals(miarray,miarray2))
            System.out.println("Los 2 array son iguales");
        else
            System.out.println("Los 2 array NO son iguales");
    }
}
```



Solución

Aquí puedes usar el **CÓDIGO**.

```
Arrays.sort(miarray2); //Ordenación ascendente del array

for (int i = 0; i < miarray2.length; i++)
    System.out.print(miarray2[i] + " ");

int pos,n=7;

pos=Arrays.binarySearch(miarray2, n);

if (pos<0)
    System.out.println("El elemento " + n + " NO está en el
                        array");
else
    System.out.println("El elemento " + n + " está en la
                        posición: " + pos);

System.arraycopy(miarray2, 0, miarray3, 0,miarray2.length);
//Copiar totalmente miarray2 en miarray3

for (int i = 0; i < miarray3.length; i++)
    System.out.print(miarray3[i] + " ");

}

}
```



Solución

Aquí puedes usar el **CÓDIGO**.

```
package com.mycompany.proyectoarraystring;
import java.util.Scanner;
public class Principal
{
    public static void main(String[] args)
    {
        Scanner teclado;
        teclado = new Scanner(System.in);
        System.out.println("Introduzca un texto: ");
        String cadena;
        cadena = teclado.nextLine();
        cadena = "La cadena introducida es: " + " -" + cadena + "-";
        System.out.println(cadena);
        String cadena1 = "varon", cadena2 = "baron",cadena3;
        if (cadena1.equalsIgnoreCase(cadena2))
            System.out.println("Son iguales");
        else
            System.out.println("Son diferentes");

        if (cadena1.compareTo(cadena2)<0)
            System.out.println("La primera cadena va a antes que la
                                segunda");
        else
            System.out.println("La segunda cadena es anterior a la
                                primera");
        String numero = String.valueOf(1234);
        System.out.println("Caracter de la posicion 3 de " +
                            cadena2+ " es: " + cadena2.charAt(3));
        cadena3 = cadena2.toUpperCase();
        System.out.println("Minúsculas: " + cadena2 + " -
                            Mayúsculas: " + cadena3);
    }
}
```



Solución

Aquí puedes usar el **CÓDIGO**.

```
package com.mycompany.proyectoarraystring;
import java.util.Scanner;
public class Principal
{
    public static void main(String[] args)
    {
        String cadena2 = "hoy es jueves";
        if (cadena2.matches(".*jue.*"))
            System.out.println("Sí contiene jue");
        else System.out.println("No contiene jue");

        String str = "blanco-rojo:amarillo.verde_azul";
        String [] cadenas = str.split("[-:._]");
        //extraer como cadenas lo que está entre los símbolos
        for(int i = 0; i<cadenas.length; i++)
            System.out.println(cadenas[i]);

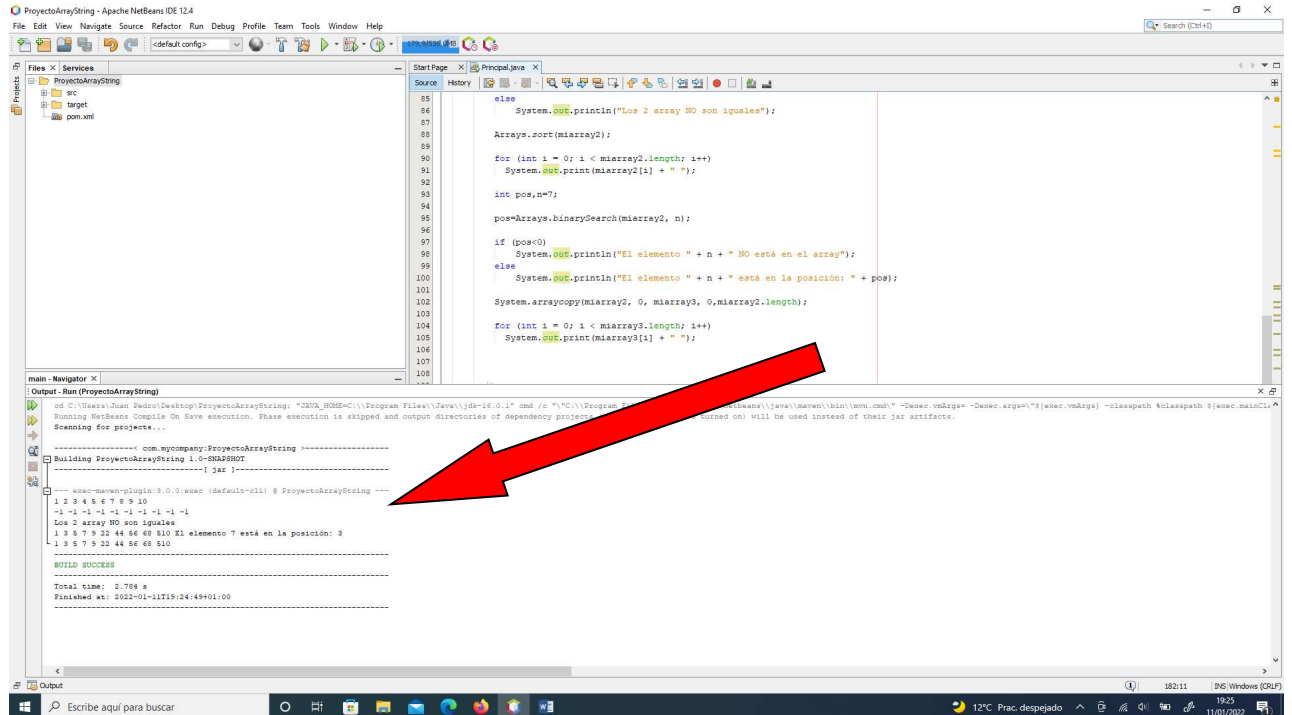
        for (char i = 'a'; i != 'z'; i++)
            System.out.print(i + " ");

        String micadena= "en el cementerio en el que me encuentro
                        hay enterrada mucha gente ";

        int aparicion=micadena.indexOf("en"); // Mostrar apariciones
        while(aparicion>=0)
        {
            System.out.println("La cadena en sí aparece en la
                                posicion: " + aparicion);
            aparicion=micadena.indexOf("en",aparicion+1);
        }
    }
}
```

4 Ejecución

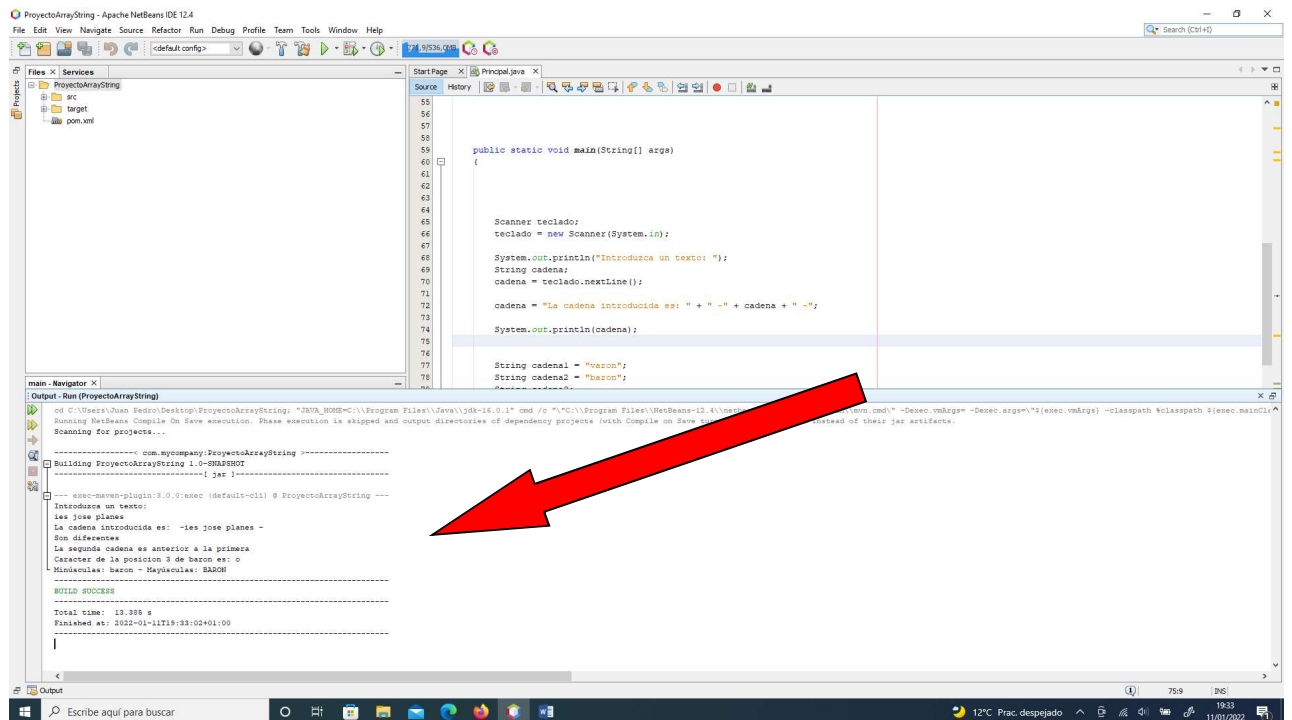
PROGRAMA1



```
1 2 3 4 5 6 7 8 9 10
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Los 2 array NO son iguales
1 3 5 7 9 22 44 56 68 810 El elemento 7 está en la posición: 3
1 3 5 7 9 22 44 56 68 810

BUILD SUCCESS
Total time: 2.784 s
Finished at: 2022-01-11T19:24:49:01:00
```

PROGRAMA2



```
Introduzca un texto:
Introduzca un texto:
La cadena introducida es: -les jose planes -
Son diferentes
La segunda cadena es anterior a la primera
Caracter de la posición 3 de baron es: o
Minúsculas: baron - Mayúsculas: BARON

BUILD SUCCESS
Total time: 13.888 s
Finished at: 2022-01-11T19:33:02:40:00
```


PROGRAMA3

The screenshot shows the Apache NetBeans IDE interface. The main editor displays a Java file named `Principal.java` with the following code:

```
54 }
55
56
57
58
59 public static void main(String[] args)
60 {
61
62
63
64     String cadena2 = "hoy es jueves";
65
66     if (cadena2.matches(".*jue.*"))
67     {
68         System.out.println("Si contiene jue");
69     }
70     else System.out.println("No contiene jue");
71
72     String[] palabras = {"verde", "rojo", "amarillo", "blanco", "negro"};
73     for (int i = 0; i < palabras.length; i++)
74     {
75         System.out.println(palabras[i]);
76     }
77 }
```

The output window at the bottom shows the execution results:

```
----- com.mycompany:ProyectoArrayString -----
Building ProyectoArrayString 1.0-SNAPSHOT
-----[jars]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ ProyectoArrayString ---
Si contiene jue
blanco
rojo
amarillo
verde
-----
a b c d e f g h i j k l m n o p q r s t u v w x y La cadena es si apa
La cadena es si aparece en la posicion: 9
La cadena es si aparece en la posicion: 17
La cadena es si aparece en la posicion: 30
La cadena es si aparece en la posicion: 34
La cadena es si aparece en la posicion: 44
La cadena es si aparece en la posicion: 61
-----
BUILD SUCCESS
Total time: 2.496 s
Finished at: 2022-01-11T19:48:58+01:00
-----
```

A large red arrow points from the output window towards the code in the main editor, specifically highlighting the `cadena2` variable and the `matches` method call.