

2.B. Tipos de datos.

3. Trabajo con cadenas.

Ya hemos visto en el apartado de literales que el objeto `String` se corresponde con una secuencia de caracteres entrecomillados, como por ejemplo "hola". Este literal se puede utilizar en Java como si de un tipo de datos primitivo se tratase, y, como caso especial, no necesita la orden `new` para ser creado.

No se trata aquí de que nos adentremos en lo que es una clase u objeto, puesto que lo veremos en unidades posteriores, y trabajaremos mucho sobre ello. Aquí sólo vamos a utilizar determinadas operaciones que podemos realizar con el objeto `String`, y lo verás mucho más claro con ejemplos descriptivos.

Para aplicar una operación a una variable de tipo `String`, escribiremos su nombre seguido de la operación, separados por un punto. Entre las principales operaciones que podemos utilizar para trabajar con cadenas de caracteres están las siguientes:

- **Creación.** Como hemos visto en el apartado de literales, podemos crear una variable de tipo `String` simplemente asignándole una cadena de caracteres encerrada entre comillas dobles.
- **Obtención de longitud.** Si necesitamos saber la longitud de un `String`, utilizaremos el método `length()`.
- **Concatenación.** Se utiliza el operador `+` o el método `concat()` para concatenar cadenas de caracteres.
- **Comparación.** El método `equals()` nos devuelve un valor booleano que indica si las cadenas comparadas son o no iguales. El método `equalsIgnoreCase()` hace lo propio, ignorando las mayúsculas de las cadenas a considerar.
- **Obtención de subcadenas.** Podemos obtener cadenas derivadas de una cadena original con el método `substring()`, al cual le debemos indicar el inicio y el fin de la subcadena a obtener.
- **Cambio a mayúsculas/minúsculas.** Los métodos `toUpperCase()` y `toLowerCase()` devuelven una nueva variable que transforma en mayúsculas o minúsculas, respectivamente, la variable inicial.
- **Valueof.** Utilizaremos este método para convertir un tipo de dato primitivo (`int`, `long`, `float`, etc.) a una variable de tipo `String`.

A continuación varios ejemplos de las distintas operaciones que podemos realizar con cadenas de caracteres o `String` en Java:

```
public class ejemplocadenas {
    public static void main(String[] args)
    {
        String cad1 = "CICLO DAM";
        String cad2 = "ciclo dam";

        System.out.printf("La cadena cad1 es: %s y cad2 es: %s", cad1, cad2 );

        System.out.printf( "\nLongitud de cad1: %d", cad1.length() );

        // concatenación de cadenas (concat o bien operador +)
        System.out.printf( "\nConcatenación: %s", cad1.concat(cad2) );

        //comparación de cadenas
        System.out.printf("\ncad1.equals(cad2) es %b", cad1.equals(cad2) );
        System.out.printf("\ncad1.equalsIgnoreCase(cad2) es %b", cad1.equalsIgnoreCase(cad2) );
        System.out.printf("\ncad1.compareTo(cad2) es %d", cad1.compareTo(cad2) );

        //obtención de subcadenas
        System.out.printf("\ncad1.substring(0,5) es %s", cad1.substring(0,5) );

        //pasar a minúsculas
        System.out.printf("\ncad1.toLowerCase() es %s", cad1.toLowerCase() );

        System.out.println();
    } // fin main
} // fin ejemplocadenas
```

Código anterior copiable:

/**

```

*
* @author FMA
*/

public class ejemplocadenas {

    public static void main(String[] args)

    {

        String cad1 = "CICLO DAM";

        String cad2 = "ciclo dam";

        System.out.printf( "La cadena cad1 es: %s y cad2 es: %s", cad1,cad2 );

        System.out.printf( "\nLongitud de cad1: %d", cad1.length() );


        // concatenación de cadenas (concat o bien operador +)

        System.out.printf( "\nConcatenación: %s", cad1.concat(cad2) );


        //comparación de cadenas

        System.out.printf("\ncad1.equals(cad2) es %b", cad1.equals(cad2) );

        System.out.printf("\ncad1.equalsIgnoreCase(cad2) es %b", cad1.equalsIgnoreCase(cad2) );

        System.out.printf("\ncad1.compareTo(cad2) es %d", cad1.compareTo(cad2) );


        //obtención de subcadenas

        System.out.printf("\ncad1.substring(0,5) es %s", cad1.substring(0,5) );


        //pasar a minúsculas

        System.out.printf("\ncad1.toLowerCase() es %s", cad1.toLowerCase() );


        System.out.println();

    } // fin main

} // fin ejemplocadenas

```

Puedes buscar más información sobre la clase String en la [API de Java](#).

