

CREACIÓN DE **STRINGs**

- ▶ Java crea un **String** con un literal encerrado entre dos comillas dobles.
- ▶ Estos **String** son inmutables. Todas las operaciones que hagamos con ellos, darán como resultado uno nuevo, no la modificación del anterior (altamente ineficiente).

```
String saludo = "Hola Mundo";
```

CONCATENACIÓN DE **STRINGs**

- ▶ Concatenar es yuxtaponer un **String** a continuación de otro.
- ▶ Se puede hacer con el operador + y el método **concat(...)**.

```
"Cadena " + "concatenada"  
"Mi nombre es ".concat("Pepe")
```

LONGITUD DE UN **STRING**

- ▶ Se trata del número de caracteres que lo conforman.
- ▶ Cuentan también los espacios en blanco, tabuladores, signos de puntuación, ...
- ▶ Método **length()**

```
String saludo = "Hola Mundo!";  
System.out.println(saludo.length()); //Debe imprimir 11
```

```

public static void main(String[] args) {

    //CONCATENAR
    String nombre = "Mi nombre es ".concat("Pepe");
    String nombre2 = "Mi nombre es " + "Pepe";
    System.out.println(nombre);
    System.out.println(nombre2);

    //MAYÚSCULAS-MINÚSCULAS
    String cadena = "En un lugar de La Mancha";
    System.out.println(cadena);
    System.out.println(cadena.toLowerCase());
    System.out.println(cadena.toUpperCase());
}

```

CADENAS FORMATEADAS

- ▶ Nos permiten insertar valores dentro de una cadena a posteriori.
- ▶ Evitan concatenaciones tediosas.
- ▶ Uso del método **format(...)**

`String.format("Hola, soy %s %s y quiero saludarte diciéndote %s",
nombre, apellidos, mensaje);`

```

String nombre = "Pepe";
String apellidos = "Pérez";
String mensaje = "¿qué tal estás?";
String str = String.format("Hola, soy %s %s y quiero saludarte diciéndote %s", nombre, apellidos,
System.out.println(str);

```

Es lo mismo que usar:

```

//System.out.println(str);
System.out.printf("Hola, soy %s %s y quiero saludarte diciéndote %s", nombre, apellidos, mensaje)

```

Para ver los métodos de cada clase, pon en buscador:

Java 8 apidocs String

Para comparar cadenas hay muchos métodos, de los más usados `compareTo`.

Otros:

```

System.out.println(quijote.endsWith("La Mancha"));
System.out.println(quijote.startsWith("En un lugar"));
System.out.println(quijote.contains("lugar"));

```

Si coincide con el final de la cadena, si coincide con el principio o si contiene una palabra.

Devuelven `true` o `false`.