

Vamos a declarar las siguientes variables del tipo *String*:

```
1 package apuntes;
2
3 public class ManejoStrings {
4
5     static String saludo1 = "Hola Mundo";
6     static String saludo2 = "Hola";
7     static String frase = "Esto es una frase.";
8
9     public static void main (String[] args) {
10
11         saludo2 += " Mundo";
12
13     }
14
15 }
```

A partir de éstas variables os iré explicando cómo funcionan los *Strings*, recordaré por quinta vez que los *Strings* son objetos, aunque no indagaré en el tema, quiero recordaros que todos los datos *no primitivos* son aquellos que empiezan por mayúscula, de momento solo conocemos los *Strings* y los *Doubles*, pero ya veremos cómo crearlos, y luego están los *primitivos*, los que van en minúscula → int, char, boolean, etc.

Vamos a empezar viendo los métodos existentes para los tipo *String*, para usar un método usaremos ésta sintaxis → *string.método(parámetros)*

Es decir, pondremos primero el nombre de nuestro *String*, seguido de un **punto** y el método que sea, y entre paréntesis le pasamos los parámetros que necesite el método, esto será más fácil de ver con unos ejemplos, pero primero veamos los métodos más interesantes:

- *.length()* : *int* → Devuelve la longitud del *String*.
- *.charAt(int n)* : *char* → Devuelve el *char* de la posición *n* (empezando en 0).
- *.equals(String str2)* : *boolean* → Devuelve si dos *Strings* son iguales.
- *.replaceAll(String a, String b)* : *String* → Devuelve un *String* donde se han cambiado los *Strings a* por el *String b*.
- *.substring(int n1, int n2)* : *String* → Devuelve un *String* que es la sub-palabra [n1,n2), esto lo explico mejor más adelante.

Estos son los métodos principales que usaremos, aquí tenéis un link con todos:

<https://www.discoduroderoer.es/metodos-de-string/>

Explicuemos estos métodos de uno en uno:

.length()

Devuelve la longitud de un *String*, este es bastante simple.

```
System.out.println(saludo1.length());
```

>>>

10

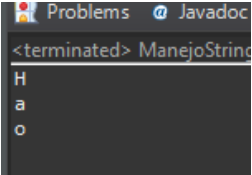
Tiene en cuenta los espacios

.charAt(int n)

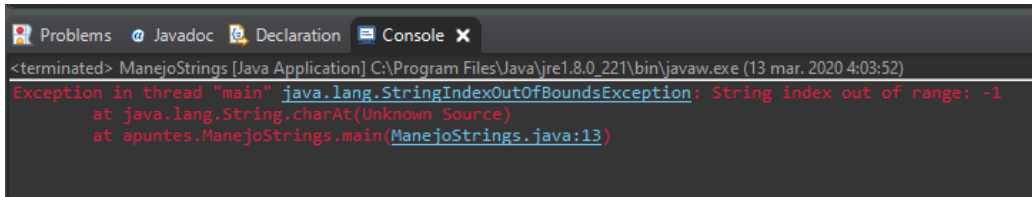
Devuelve en carácter en la posición n , teniendo en cuenta que el primero de todos es el 0:

```
System.out.println(saludo1.charAt(0));
System.out.println(saludo1.charAt(3));
System.out.println(saludo1.charAt(9));
```

>>>



Tened en cuenta que si tratas de imprimir un carácter fuera de rango os dará este error:



```
<terminated> ManejoStrings [Java Application] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (13 mar, 2020 4:03:52)
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: -1
    at java.lang.String.charAt(Unknown Source)
    at apuntes.ManejoStrings.main(ManejoStrings.java:13)
```

Aquí trate de imprimir → `System.out.println(saludo1.charAt(-1));`

.equals(String str2)

Indica si dos *Strings* son iguales, no hay mucho que explicar:

```
System.out.println(saludo1.equals(saludo2));
```

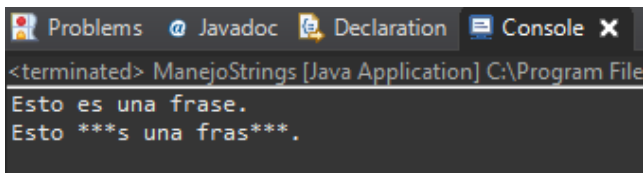
>>> true

.replaceAll(String a, String b)

Cambia todos los *Strings* a por b :

```
System.out.println(frase);
System.out.println(frase.replaceAll("e", "***"));
```

>>>



```
<terminated> ManejoStrings [Java Application] C:\Program File
Esto es una frase.
Esto ***s una fras***.
```

Tened en mente que no los sustituye en el propio *String*, sino que devuelve otro distinto.

.substring(int a, int b)

Este puede ser más complicado de pillar, vamos a ver un ejemplo:

```
System.out.println(frase);
System.out.println(frase.charAt(10));
frase = frase.substring(2,10);
System.out.println(frase);
```

>>>

```
Esto es una frase.
a
to es un
```

Si nos fijamos si introducimos 2 como primer parámetro, sí que te devuelve la frase empezando en el carácter 2, pero si introducimos como segundo parámetro el 10, acaba en el carácter 9, por lo que podemos decir que te devuelve un String desde *a* hasta *b-1*.

Esto se debe seguramente a que tiene el mismo funcionamiento que éste código, como ya he dicho, yo no he creado la clase *String.java*, pero puedo intuirlo:

```
public static void main (String[] args) {
    int[] nums = {0,1,2,3,4,5,6,7,8,9,10};

    int a = StdIn.readInt();
    int b = StdIn.readInt();

    for (int i = a; i < b; i++) {
        System.out.print(nums[i] + " ");
    }
}
```

>>>

```
2
7
2 3 4 5 6
```

Como veis no me incluye el 7.