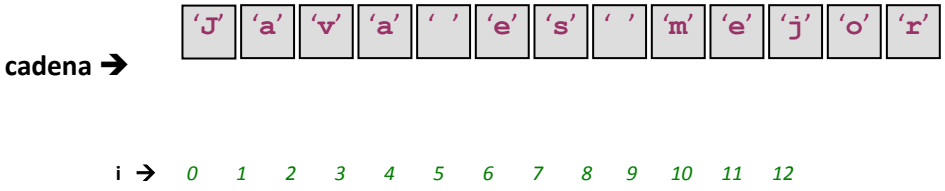
Métodos de la Clase: *String*

Esta Clase dispone de diversos métodos para manipular cadenas.

```
String  cadena = "Java es mejor";
```



METODO	DESCRIPCION

`length()`	Devuelve la longitud de la cadena.
	
	`int  longitud = cadena.length();`
	
	`longitud ← 13`

`charAt(int)`	Devuelve una copia del carácter que encuentre en la posición indicada por el parámetro.
	
	`char  caracter = cadena.charAt(8);`
	
	`caracter ← 'm'`

`equals(String)`	Comprueba si dos cadenas son iguales. En este caso comprueba que el objeto dado como argumento sea de tipo *String* y contenga la misma cadena de caracteres que el objeto actual.
	
	`String  s = "Java";`
	`boolean  x = cadena.equals(s);`
	
	`x ← false`

`equalsIgnoreCase(String)`	Realiza la misma tarea que *equals* pero sin tener en cuenta las mayúsculas o minúsculas.

	`String s = "java Es MeJOR";`
	`boolean x = cadena.equalsIgnoreCase(s);`
	
	`x ← true`

`compareTo(String)`	Devuelve un entero menor que cero si la cadena es alfabéticamente menor que la dada como argumento, cero si las dos cadenas son léxicamente iguales y un entero mayor que cero si la cadena es mayor alfabéticamente.
	
	`String s1 = "Java es lo máximo",`
	`s2 = "Java es mejor",`
	`s3 = "Java es ok";`
	
	`int x = cadena.compareTo(s1),`
	`y = cadena.compareTo(s2),`
	`z = cadena.compareTo(s3);`
	
	`x ← 1 // cadena mayor que s1 alfabéticamente`
	`y ← 0 // cadena contiene lo mismo que s2`
	`z ← -2 // cadena menor que s3 alfabéticamente`

`startsWith(String)`	Comprueba si el comienzo de la cadena actual coincide con la cadena pasada como parámetro.
	
	`String s = "JavaX";`
	`boolean x = cadena.startsWith(s);`
	
	`x ← false`

`endsWith(String)`	Comprueba si el final de la cadena actual coincide con la cadena pasada como parámetro.
	
	`String s = "mejor";`
	`boolean x = cadena.endsWith(s);`
	
	`x ← true`

`indexOf(int)`	Devuelve la posición que por primera vez aparece el carácter (expresado como entero) pasado como parámetro. En caso no exista devuelve -1.
	
	`int i = cadena.indexOf('e');`
	`i ← 5`

`indexOf(int,int)`	Devuelve la posición que por primera vez aparece el carácter (expresado como entero) a partir de la posición especificada como segundo parámetro.
	
	`int i = cadena.indexOf('e',6);`
	`i ← 9`

`indexOf(String)`	Devuelve la posición que por primera vez aparece la cadena pasada como parámetro.
	
	`int i = cadena.indexOf("va");`
	`i ← 2`

`indexOf(String,int)`	Devuelve la posición que por primera vez aparece la cadena pasada como parámetro, pudiendo especificar en un segundo parámetro a partir de dónde buscar.
	
	`int i = cadena.indexOf("ej",5);`
	`i ← 9`

`lastIndexOf(int)`
`lastIndexOf(int,int)`
`lastIndexOf(String)`
`lastIndexOf(String,int)`
	Devuelve la última vez que aparece el carácter (expresado como entero) o cadena pasada como parámetro, pudiendo especificar en un segundo parámetro, a partir de dónde buscar (búsqueda hacia atrás).
	
	`String s = "e";`
	`int i = cadena.lastIndexOf(s);`
	`i ← 9`

`toLowerCase()`	Convierte la cadena a minúsculas.
	
	`String s = "CiberJava – Lima – Perú";`
	`s = s.toLowerCase();`
	`s ←"ciberjava – lima – perú"`

`toUpperCase()`	Convierte la cadena a mayúsculas.
	
	`String s = "CiberJava – Lima – Perú";`
	`s = s.toUpperCase();`
	`s ←"CIBERJAVA – LIMA – PERÚ"`

`trim()`	Elimina espacios al principio y al final de la cadena.
	
	`String s = "  CiberJava  Lima  ";`
	`s = s.trim();`
	`s ←"CiberJava  Lima"`

`substring(int)`	Devuelve una subcadena de la cadena actual, empezando por el primer índice indicado hasta antes del segundo índice (si se especifica) o hasta el final de la cadena.
	
	`String s1 = "Viva el Perú",`
	`s2 = s1.substring(5),`
	`s3 = s1.substring(3,9);`
	`s2 ← "el Perú"`
	`s3 ← "a el P"`
	
	`substring(int,int)`

`replace(char,char)`	Reemplaza todos los caracteres iguales al primer parámetro y los sustituye por el carácter que pasamos en segundo lugar, teniendo en cuenta lo mismo una mayúscula que una minúscula.
	
	`String s = "biba el Perú";`
	`s = s.replace('b','v');`
	`s ←"viva el Perú"`

`toCharArray()`	Convierte la cadena a un vector de caracteres. `char[] arreglo = cadena.toCharArray();`

Métodos estáticos de conversión

La clase *String* dispone de varios métodos para transformar valores de otros tipos de datos a cadena. Todos se llaman valueOf y son estáticos.

`String.valueOf(boolean)`
 	
`String.valueOf(int)`
 	
`String.valueOf(long)`
 	
`String.valueOf(float)`
 	
`String.valueOf(double)`
 	
`String.valueOf(Object)`
 	
`String.valueOf(char[])`
	`double r = 3.1416;`
	
	`String s = String.valueOf(r);`
	
	`s ← "3.1416"`

`String.valueOf(char[],int,int)`	Transforma una subcadena de un arreglo de caracteres, especificando una posición y la longitud.
	
	`char c[]={'C','i','b','e','r','J','a','v','a'};`
	`String s = String.valueOf(c,3,5);`
	
	`s ← "erJav"`