

# Clase String (java.lang.String)

---

## Métodos de la clase String

En todos los prototipos i, j son valores enteros, b es un valor boolean, s, t, s1, s2 son de tipo String, c, c1, c2 es de tipo char.

### Longitud

<b>i = s.length();</b> longitud de la cadena s	<b>public int length()</b>
---	----------------------------

### Comparación (no se usa == ni i=)

<b>i = s.compareTo(t);</b> Comparar con s, devuelve <0 si s < t, 0 si s=t, >0 si s > t	<b>public int compareTo(String s)</b>
---	---------------------------------------

<b>i = s.compareToIgnoreCase(t);</b> Igual que la anterior pero ignorando mayúsculas y minúsculas	<b>public int compareToIgnoreCase(String s)</b>
--	---

<b>b = s.equals(t);</b> true si las dos cadenas son iguales	<b>public boolean equals(Object anObject)</b>
--	---

<b>b = s.equalsIgnoreCase(t);</b> true si las dos cadenas son iguales ignorando mayúsculas y minúsculas	<b>public boolean equalsIgnoreCase(String s)</b>
--	--

<b>b = s.startsWith(t);</b> true si s empieza con t	<b>public boolean startsWith(String prefijo)</b>
--	--

<b>b = s.startsWith(t, i);</b> true si t aparece en s a partir de i	<b>public boolean startsWith(String prefijo, int desplazamiento)</b>
--	--

<b>b = s.endsWith(t);</b> true si s acaba en t	<b>public boolean endsWith(String sufijo)</b>
---	---

### Búsqueda (Todos los indexOf devuelven -1 si la cadena / carácter no se encuentra)

<b>i = s.indexOf(t);</b> devuelve la posición de la primera ocurrencia de t en s	<b>public int indexOf(String s)</b>
---	-------------------------------------

<b>i = s.indexOf(t, i);</b> devuelve la posición de la primera ocurrencia de t en s a partir de i	<b>public int indexOf(String s, int desde)</b>
--	--

**i = s.indexOf(c);** **public int indexOf(int c)**  
devuelve la posición de la primera ocurrencia del carácter c en s

**i = s.indexOf(c, i);** **public int indexOf(int c, int desde)**  
devuelve la posición de la primera ocurrencia del carácter c en s a partir de i

**i = s.lastIndexOf(c);** **public int lastIndexOf(int c)**  
devuelve la posición de la última ocurrencia del carácter c en s

**i = s.lastIndexOf(c, i);** **public int lastIndexOf(int c, int desde)**  
devuelve la posición de la última ocurrencia del carácter c en s antes de i

**i = s.lastIndexOf(t);** **public int lastIndexOf(String s)**  
devuelve la posición de la última ocurrencia de t en s

**i = s.lastIndexOf(t, i);** **public int lastIndexOf(String s, int desde)**  
devuelve la posición de la última ocurrencia de t en s antes de i buscando hacia atrás

**b = s.contains(s);** **public boolean contains(CharSequence s)**  
devuelve true si s contiene a s1

## Obtener partes de una cadena

**c = s.charAt(i);** **public char charAt(int index)**  
carácter de la posición i en s

**s1 = s.substring(i);** **public String substring(int desde)**  
subcadena desde índice i hasta el final de s

**s1 = s.substring(i, j);** **public String substring(int desde, int hasta)**  
subcadena desde índice i hasta j (sin incluir j) de s

## Crear una nueva cadena a partir de la original

**s1 = s.toLowerCase();** **public String toLowerCase()**  
nueva cadena con todos los caracteres en minúscula

**s1 = s.toUpperCase();** **public String toUpperCase()**  
nueva cadena con todos los caracteres en mayúscula

**s1 = s.trim();** **public String trim()**  
nueva cadena con los espacios en blanco eliminados al principio y al final

**s1 = s.replace(c1, c2);** **public String replace(char viejo, char nuevo)**

nueva cadena con todos los caracteres c1 sustituidos por c2

**s1 = s.replace(s1, s2);**

**public String replace(CharSequence vieja,  
CharSequence nueva)**

nueva cadena con todos las subcadenas s1 sustituidas por c2

## Otros métodos

**array = s.split(s2);**

**public String[] split(String patrón)**

array de todas las subcadenas de s delimitadas por s2

**b = s.isEmpty();**

**public boolean isEmpty()**

detecta si s es una cadena vacía devolviendo true si la longitud de s es 0

**s2 = s.concat(s1);**

**public String concat(String s)**

añade s1 al final de s

## Métodos estáticos para convertir a String

**s = String.valueOf(x);**

**public static String valueOf(int i)**

**public static String valueOf(float f)**

convierte x a String, x es de cualquier tipo (primitivo u objeto)

**s = String.format(f, x, ... );**

**public static String format(String formato,  
Object ... args)**

utiliza el formato f para convertir un nº variable de parámetros a String

## Otros métodos estáticos de la clase String

**s = String.join(del, s1, s2, ...);**

**public static String join(CharSequence delimitador,  
CharSequence... elementos)**

devuelve una cadena formada por todas las cadenas elementos unidas por el delimitador

```
String mensaje = String.join("-", "Java", "is", "cool");  
// devuelve "Java-is-cool"
```