

Principales métodos del objeto **String** en Java

Método	Descripción	Devuelve	Ejemplo
<code>length()</code>	Devuelve la longitud del string.	int – Número de caracteres.	<code>"Hola".length()</code> → 4
<code>charAt(int index)</code>	Devuelve el carácter en la posición indicada.	char – Carácter en el índice.	<code>"Hola".charAt(1)</code> → 'o'
<code>substring(int beginIndex)</code>	Devuelve la subcadena desde el índice dado hasta el final.	String – Subcadena.	<code>"Hola".substring(2)</code> → "la"
<code>substring(int beginIndex, int endIndex)</code>	Devuelve la subcadena entre los índices.	String – Subcadena.	<code>"Hola".substring(1, 3)</code> → "ol"
<code>equals(String another)</code>	Compara si dos strings son exactamente iguales.	boolean	<code>"Hola".equals("hola")</code> → false
<code>equalsIgnoreCase(String another)</code>	Compara dos strings ignorando mayúsculas/minúsculas.	boolean	<code>"Hola".equalsIgnoreCase("hola")</code> → true
<code>compareTo(String another)</code>	Compara alfabéticamente.	int – 0 si son iguales, negativo o positivo según orden.	<code>"Hola".compareTo("Adiós")</code> → > 0

<code>contains(CharSequence s)</code>	Verifica si contiene una secuencia.	<code>boolean</code>	<code>"Hola".contains("la") → true</code>
<code>indexOf(String str)</code>	Primer índice donde aparece una subcadena.	<code>int</code> – Posición o -1 si no se encuentra.	<code>"Hola".indexOf("l") → 2</code>
<code>lastIndexOf(String str)</code>	Última aparición de una subcadena.	<code>int</code>	<code>"Hola la".lastIndexOf("la") → 5</code>
<code>toUpperCase()</code>	Convierte a mayúsculas.	<code>String</code>	<code>"hola".toUpperCase() → "HOLA"</code>
<code>toLowerCase()</code>	Convierte a minúsculas.	<code>String</code>	<code>"HOLA".toLowerCase() → "hola"</code>
<code>trim()</code>	Elimina espacios al inicio y al final (espacios clásicos ASCII).	<code>String</code>	<code>" Hola ".trim() → "Hola"</code>
<code>strip()</code>	Elimina espacios al inicio y al final (según Unicode). Es más completo.	<code>String</code>	<code>" Hola ".strip() → "Hola"</code>
<code>replace(char oldChar, char newChar)</code>	Reemplaza caracteres.	<code>String</code>	<code>"hola".replace('a', 'i') → "holi"</code>
<code>replaceAll(String regex, String replacement)</code>	Reemplaza usando expresiones regulares.	<code>String</code>	<code>"a1b2c3".replaceAll("\\d", "") → "abc"</code>
<code>startsWith(String prefix)</code>	Verifica si empieza con el prefijo dado.	<code>boolean</code>	<code>"Hola".startsWith("Ho") → true</code>

<code>endsWith(String suffix)</code>	Verifica si termina con el sufijo dado.	<code>boolean</code>	<code>"Hola".endsWith("la") → true</code>
<code>isEmpty()</code>	Verifica si el string está vacío.	<code>boolean</code>	<code>"".isEmpty() → true</code>
<code>split(String regex)</code>	Divide el string por el patrón dado (regex).	<code>String[]</code> – Arreglo de subcadenas.	<code>"a,b,c".split(",") → ["a", "b", "c"]</code>
<code>matches(String regex)</code>	Verifica si el string cumple con una expresión regular.	<code>boolean</code>	<code>"123".matches("\\d+") → true</code>
<code>toCharArray()</code>	Convierte el string a un arreglo de caracteres.	<code>char[]</code>	<code>"Hi".toCharArray() → ['H', 'i']</code>
<code>join(CharSequence delimiter, CharSequence... elements)</code>	Une strings con un delimitador.	<code>String</code>	<code>String.join("-", "a", "b", "c") → "a-b-c"</code>