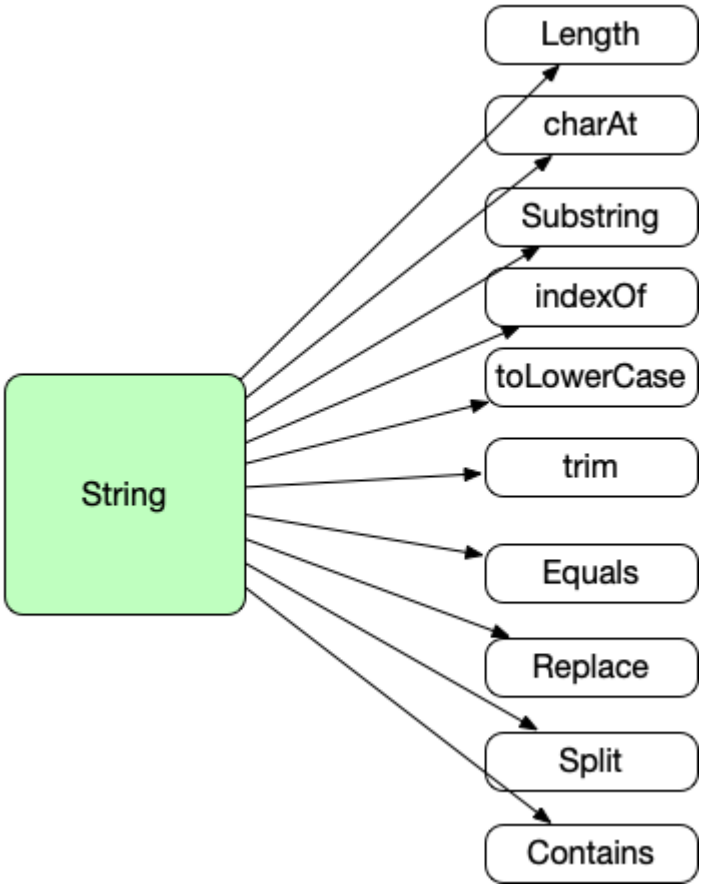
Tabla de Contenidos

En Java, la clase String es una de las más utilizadas y fundamentales. Los objetos de tipo String son inmutables, lo que significa que no se pueden cambiar después de ser creados. Aunque esto pueda parecer una limitación, Java proporciona una amplia gama de métodos para manipular cadenas, creando nuevas instancias sin modificar la original. A continuación, exploraremos algunos de los métodos más importantes de la clase String y cómo utilizarlos.



1. length()

Este método devuelve la longitud de la cadena, es decir, el número de caracteres que contiene. Es útil para saber cuántos caracteres tiene una cadena.

```
String texto = "ArquitecturaJava";
int longitud = texto.length();
System.out.println("Longitud: " + longitud);  // Salida: Longitud: 16
```

2. charAt(int indice)

Devuelve el carácter en una posición específica dentro de la cadena. El índice comienza en 0, por lo que el primer carácter está en la posición 0.

```
String texto = "ArquitecturaJava";
char caracter = texto.charAt(5);
```

```
System.out.println("Caracter en la posición 5: " + caracter);  //
Salida: c
```

3. substring(int inicioIndice)

Este método devuelve una nueva cadena que es una subcadena de la original, comenzando en el índice especificado y hasta el final de la cadena.

```
String subcadena = texto.substring(0, 11);
System.out.println(subcadena);  // Salida: Arquitectura
```

4. indexOf(String cadena)

Devuelve el índice dentro de la cadena donde comienza la primera aparición de la subcadena especificada. Si la subcadena no se encuentra, devuelve -1.

```
String texto = "ArquitecturaJava";
int indice = texto.indexOf("Java");
System.out.println(indice);  // Salida: 12
```

5. toLowerCase() y toUpperCase()

Estos métodos convierten una cadena a minúsculas o mayúsculas, respectivamente.

```
String texto = "ArquitecturaJava";
String minusculas = texto.toLowerCase();
String mayusculas = texto.toUpperCase();
System.out.println(minusculas);  // Salida: arquitecturajava
System.out.println(mayusculas);  // Salida: ARQUITECTURAJAVA
```

6. trim()

Elimina los espacios en blanco al principio y al final de la cadena, pero no afecta a los espacios entre palabras.

```
String texto = "  ArquitecturaJava  ";
String textoRecortado = texto.trim();
System.out.println("[" + textoRecortado + "]");  // Salida:
[ArquitecturaJava]
```

7. equals(Object objeto) y equalsIgnoreCase(String otroString)

El método equals() compara el contenido de dos cadenas para verificar si son exactamente iguales. La versión equalsIgnoreCase() ignora las mayúsculas y minúsculas durante la comparación.

```
String texto1 = "ArquitecturaJava";
String texto2 = "arquitecturajava";

boolean sonIguales = texto1.equals(texto2);
boolean sonIgualesIgnorandoCaso = texto1.equalsIgnoreCase(texto2);

System.out.println(sonIguales);  // Salida: false
System.out.println(sonIgualesIgnorandoCaso);  // Salida: true
```

8. replace(CharSequence origen, CharSequence nuevoTexto)

Reemplaza todas las apariciones de un carácter o secuencia de caracteres dentro de la cadena por otra secuencia especificada.

```
String texto = "ArquitecturaJava";
String nuevoTexto = texto.replace("Java", "Spring");
System.out.println(nuevoTexto);  // Salida: ArquitecturaSpring
```

9. split(String regex)

Divide la cadena en un arreglo de cadenas utilizando un delimitador basado en una expresión regular (regex). Es útil para separar palabras o elementos dentro de una cadena.

```
String texto = "Java,Spring,Hibernate";
String[] partes = texto.split(",");
for (String parte : partes) {
    System.out.println(parte);
}
// Salida:
// Java
// Spring
// Hibernate
```

10. contains(CharSequence s)

Este método devuelve true si la cadena contiene la secuencia de caracteres especificada, y false en caso contrario.

```
String texto = "ArquitecturaJava";
boolean contiene = texto.contains("Java");
System.out.println(contiene);  // Salida: true
```

Java String Conclusiones

Estos son los métodos mas comunes de la clase String y el como utilizarlos

- Java Generics (II) uso de WildCard
- Angular ngFor la directiva y sus opciones
- Java String Pool , un concepto importante
- Java Stream String y Java 8
- Java new String y la creación de objetos