

Manipular cadenas con StringBuilder

Fundamentos StringBuilder

- Representa cadenas de caracteres mutables, se pueden modificar a través de los métodos de la clase
- Los métodos operan sobre el propio objeto
- Se crean:

```
StringBuilder sb=new StringBuilder("cadena mutable");
```

- No admite concatenación entre ellos, pero si con String (llamada a toString()), devolviendo un nuevo String:

```
String s="hola "+sb;  
System.out.println(s); //muestra: hola cadena mutable
```

Métodos StringBuilder (I)

- **StringBuilder append(tipo dato).** Método sobrecargado para añadir cualquier tipo Java a la cadena. Modifica la cadena original y devuelve una referencia al propio objeto

```
StringBuilder sb=new StringBuilder("mutable");  
StringBuilder sb2=sb.append(" nueva");  
System.out.println(sb); //llamada a toString(), muestra: mutable nueva  
System.out.println(sb==sb2); //muestra true, apuntan al mismo objeto
```

- **StringBuilder insert(int pos, tipo dato).** Inserta un dato dentro de la cadena en la posición indicada

```
StringBuilder sb=new StringBuilder("es texto");  
sb.insert(3,200)  
System.out.println(sb); //muestra: es 200texto
```

Métodos StringBuilder (II)

- **StringBuilder replace(int a, int b, String s).** Reemplaza los caracteres que se encuentran situados entre las posiciones a y b-1 por la cadena que se indica como tercer parámetro

```
StringBuilder sb=new StringBuilder("primera cadena");  
sb.replace(0,7,"segunda");  
System.out.println(sb); //muestra: segunda cadena
```

- **StringBuilder delete(int a, int b).** Elimina de la cadena los caracteres situados entre las posiciones a y b-1

```
StringBuilder sb=new StringBuilder("cadena nueva");  
sb.delete(3,6);  
System.out.println(sb); //muestra: cad nueva
```

- **StringBuilder reverse().** Invierte la cadena

```
StringBuilder sb=new StringBuilder("hola");  
sb.reverse();  
System.out.println(sb); //muestra: aloh
```

Método compareTo

- Nuevo en Java 11 para poder comparar dos `StringBuilder`
- `int compareTo(StringBuilder another)`. Devuelve 1 si la cadena pasada como parámetro es menor (lexicográficamente) que la principal, -1 si es mayor y 0 si son iguales:

```
StringBuilder sb=new StringBuilder("cadena nueva");  
StringBuilder sb2=new StringBuilder("otra cadena nueva");  
//muestra: sb menor que sb2  
System.out.println(sb.compareTo(sb2)<0?"sb menor que sb2":"sb mayor o igual que sb2");
```

Métodos comunes con String

- `char charAt(int pos)`
- `int indexOf(String s)`
- `int length()`
- `String substring(int a, int b)`

Antonio Martín Sierra

Método equals

➤ NO está sobrescrito

```
StringBuilder s1=new StringBuilder("hello");  
StringBuilder s2=new StringBuilder("hello");  
System.out.println(s1.equals(s2)); //false
```

Antonio Martín Sierra