

# UT5 ejercicio 7

## Constructores

```
// Crea un string builder vacío con capacidad de 16 elementos vacíos
StringBuilder sb = new StringBuilder();

// Crea un string builder inicializado con una secuencia de caracteres
CharSequence cs = "Hello World";StringBuilder sb2 = new StringBuilder(cs);

// Crea un constructor vacío con la capacidad inicial especificada
StringBuilder sb3 = new StringBuilder(2);

// Crea un constructor cuyo valor es inicializado por el string específico
StringBuilder sb4 = new StringBuilder("s");
```

## SetLength()

```
// Establece el largo de la secuencia de caracteres
// Si newLength < length
sb2.setLength(5);
System.out.println(sb2); // imprime "Hello"

//Si newLength > length
sb2.setLength(12);
System.out.println(sb2.length());

// imprime el "mismo" contenido, pero se agregan caracteres vacíos a la secuencia
// teniendo ahora 12 elementos.
```

## EnsureCapacity()

```
// Asegura que la capacidad sea por lo menos igual a lo mínimo especificado
sb2.ensureCapacity(3);
System.out.println(sb2.length());
```

## Append()

```
// Hace un append del argumento al string builder. Los datos son convertidos antes
// que se lleve la operación a cabo.
```

```
sb2.append(55);  
sb2.append("Appending");  
sb2.append(33.5f);  
System.out.println(sb2);
```

## Delete()

```
// Elimina la subsecuencia desde el inicio hasta el final-1 inclusive  
sb2.delete(7, 10);  
System.out.println(sb2);  
  
// Elimina el caracter localizado en el indice especificado.  
sb2.deleteCharAt(3);  
System.out.println(sb2);
```

## Insert()

```
// Inserta el segundo argumento en el string builder  
// El primer argumento indica el indice anterior al que la data va a ser insertada  
// La data se convierte a string antes de que la operación se lleve a cabo.  
sb2.insert(6, "Inserting");  
sb2.insert(3, true);  
sb2.insert(9, 33.45f);  
System.out.println(sb2);
```

## Reverse()

```
// mueve en reversa la secuencia de caracteres en el string builder.  
sb2.reverse();  
System.out.println(sb2);
```