



Clase String

int length()

Retorna la cantidad de caracteres de la cadena.

```
int largo = "Abracadabra".length(); // largo ← 11
```

boolean equals(String cadena), equalsIgnoreCase(String cadena)

El método `equals` Compara dos strings, teniendo en cuenta la diferencia entre mayúsculas y minúsculas. El método `equalsIgnoreCase` los sin tener en cuenta la diferencia entre mayúsculas y minúsculas.

```
boolean a = "Abracadabra".equals("abracadabra"); // a ← false  
boolean b = "Abracadabra".equalsIgnoreCase("abracadabra"); // b ← true
```

boolean compareTo(String cadena)

El método `compareTo` Compara dos strings lexicográficamente, teniendo en cuenta la diferencia entre mayúsculas y minúsculas. Retorna un número mayor que cero si la cadena es mayor que la palabra y menor que cero si es menor. El método `compareToIgnoreCase` no distingue entre mayúsculas y minúsculas.

```
int dif = "Casa".compareTo("Camion"); // dif ← 6 (dif entre la "s" y la "m")
```

String substring(int indComienzo), substring(int indComienzo int indFin)

Retorna un substring que comienza en la posición `indComienzo` y termina en (`indFin - 1`). Si no se especifica el `indFin` el retorna el substring desde la posición `indComienzo`.

```
String sub = "Camionero".substring(5); // sub ← nero  
String sub = "Camionero".substring(3, 6); // sub ← ion
```

char charAt(int indice)

Retorna el carácter ubicado en la posición `indice` del String.

```
char caracter = "Palabra".charAt(3); // caracter ← 'a'
```

String concat(String cadena)

Retorna un nuevo String (no modificado) que resulta de unir el string que recibe el mensaje el la `cadena`.

```
String union = "Abra".concat("dacadabra"); // union ← "Abradacadabra"
```

boolean replace(char c1, char c2)

Retorna un nuevo String donde se han reemplazado `c1` por `c2` en la cadena que recibe el mensaje.

```
String nueva = "Abracadabra".replace('a', 'e'); // nueva ← "Abrecedebre"
```

boolean indexOf(char), indexOf(char, int), indexOf(String), indexOf(String, int)

Retorna el índice de la primera aparición del carácter (o String) a partir del índice especificado.

```
int pos = "Abracadabra".indexOf("ra", 3); // pos ← 9
```

boolean lastIndexOf(char), lastIndexOf(char, int), ...

Igual que `indexOf`, pero retorna la última aparición del carácter o cadena a partir del índice especificado..

```
int pos = "Abracadabra".lastIndexOf("ra"); // pos ← 9
```