

String

Métodos String: Longitud

- ▶ Para obtener la longitud, número de caracteres que guarda un String se llama a la función miembro **length()**.

```
String str="El primer programa";  
int longitud=str.length();
```

Métodos String: Comienzo

- ▶ Podemos conocer si un String comienza con un determinado prefijo, llamando al método **startsWith**, que devuelve true o false, según que el String comience o no por dicho prefijo

```
String str="El primer programa";  
boolean resultado=str.startsWith("El");
```

Métodos String: Fin

- ▶ Podemos saber si un String finaliza con un conjunto dado de caracteres, mediante la función miembro `endsWith`.

```
String str="El primer programa";  
boolean resultado2=str.endsWith("programa");
```

BÚSQUEDAS POR POSICIÓN

`String cadena="Ejemplo de Operaciones con "`
9

- `int pos=cadena.indexOf('e');`
 - Devuelve la posición de la primera ocurrencia que encuentre del carácter buscado 2
 - En este caso devuelve 2 ya que se encuentra en la segunda posición empezando desde cero
- `int pos=cadena.indexOf('e',10);`
 - En este caso devuelve la posición de la ocurrencia especificando desde donde queremos que empiece a buscar 13

BÚSQUEDAS POR POSICIÓN II

String cadena="Ejemplo de Operaciones con "

- **int pos=cadena.lastIndexOf('e');** 20
 - Devuelva la posición de la última ocurrencia encontrada
- **int pos=cadena.lastIndexOf('e', 25);** 20
 - Devuelve la posición de la última ocurrencia encontrada antes de la posición 25

String cadena="Ejemplo de Operaciones con "

Más métodos...

- ▶ `cadena.concat(" Concatenar")`
 - Devuelve la concatenación (tipo String)
- ▶ `cadena.contains("Ej")`
 - Devuelve un valor boolean
- ▶ `cadena.replace("o","xxx")`
 - Devuelve el String con los valores reemplazados (tipo String)
- ▶ `cadena.substring(3, 11);`
 - Devuelve el substring entre dichas posiciones (tipo String)
- ▶ `cadena.trim()` devuelve la cadena sin espacios

```
String str=" 12 ";    String qs=str.trim();
```

Más métodos...

- ▶ `"comparar".equals("Comparar")`
 - Compara dos cadenas en java y devuelve true si son iguales
 - Recordemos que java es sensible a Minisculas y Mayusculas (Case Sensitive)
 - En este caso devuelve **false**
- ▶ `"comparar".equals("comparar");`
 - En este caso devuelve **true** ya que ambas son iguales
- ▶ `"Comparar".equalsIgnoreCase("comparar")`
 - compara dos cadenas en java ignorando que estén Mayúsculas y Minúsculas en este caso devuelve **true**

Más métodos...

- ▶ **charAt(int pos)**
 - Devuelve caracter del String, llega al método la posición del caracter a extraer.
- ▶ **cadena.toLowerCase()**
 - Devuelve un String con el contenido convertido todo a minúsculas.
- ▶ **cadena.toUpperCase()**
 - Devuelve un String con el contenido convertido todo a mayúsculas.

Lectura de un carácter

```
//importamos el paquete
import java.util.Scanner;
public class MinusculasLECTURA {
    public static void main (String [] args) {
        //Declaración de varibales
        char caracter;
        //creamos objeto
        Scanner sc = new Scanner(System.in);
        System.out.println("Introduce un caracter: ");
        caracter = sc.next().charAt(0);
        System.out.println("Carácter introducido" + caracter );
    } } }
```