

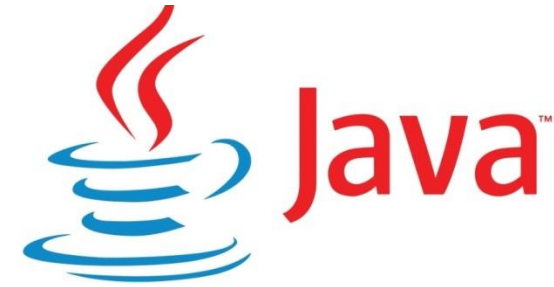
# Java 8 SE Fundamentals





# Manipulating and Formatting the Data in Your Program

- ✓ Using the String Class
- ✓ Using the Java API Docs
- ✓ Using the StringBuilder Class
- ✓ More about primitive data types
- ✓ The remaining numeric operators
- ✓ Promoting and casting variables



## Usando la clase String

La clase String representa cadenas de caracteres.

En Java SE 8 las cadenas son constantes, sus valores no se pueden cambiar después de crearlos.

**Otra sintáxis: `String cadena = new String("Hola");`**



## Ejercicios.

1. Crear un programa en Java que compare dos cadenas y verifique si son iguales, por ejemplo, la cadena de un password.
2. Crear un programa en Java que transforme una cadena en minúsculas a mayúsculas.
3. Crear un programa en Java que reemplace la letra 'z' por la letra 'a' en la siguiente cadena: "Jzvz procesz czdeczs".



## Usando la clase **StringBuilder**

La clase **StringBuilder** es similar a la clase **String**.

La principal diferencia es que en Java SE 8 la clase **String** es inmutable, en cambio la clase **StringBuilder** nos permite construir textos más complejos de forma variable.



## Ejercicios.

1. Crear un programa en Java que agregue, inserte y elimine, texto en un objeto de la clase `StringBuilder`, por ejemplo, en un texto que contenga la información de una persona.
2. Crear la clase `ReciboCFE`, que use la clase `StringBuilder` para generar un reporte de consumo eléctrico según las siguientes reglas:

```
si el consumo < 800kWh <- tarifa1=1.2  
si el 800<= consumo <= 1450kWh tarifa2=1.5  
si el consuma>1450kWh tarifa3=1.8
```



# Tipo de datos primitivos

Diferencia entre tipos primitivos y objetos.



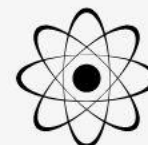
## No poseen atributos

Los tipos de datos no poseen atributo alguno, estos solo guardan un valor en alguna dirección en memoria.



## No poseen métodos

Los tipos de datos no poseen métodos, de hecho estos solo son utilizados por los métodos en partes del programa.



## Son indivisibles, atómicos

Con baja latencia hay menos retrasos; da una reacción genuina en tiempo real.



# Tipo de datos numéricos

Entre los tipos numéricos tenemos: **enteros** y de **punto flotante**

**byte**

**int**

**float**

**double**

$-2^7$  to  $2^7-1$  (-128 to 127)

$-2^{31}$  to  $2^{31}-1$  (-2147483648 to 2147483647)

32 bits, IEEE 754

64 bits, IEEE 754

8 bits, con signo

32 bits, con signo

Negative range:  $-3.4028235E+38$  to  $-1.4E-45$

Positive range:  $1.4E-45$  to  $3.4028235E+38$

**short**

**long**

16 bits, con signo

64 bits, con signo

$-2^{15}$  to  $2^{15}-1$  (-32768 to 32767)

$-2^{63}$  to  $2^{63}-1$

Negative range:  $-1.7976931348623157E+308$  to  $-4.9E-324$

Positive range:  $4.9E-324$  to  $1.7976931348623157E+308$

(i.e., -9223372036854775808 to 9223372036854775807)







# Tipo de datos no numéricos

Entre los no numéricos, tenemos los de carácter y los booleanos.

## boolean

Un tipo de datos booleano declara una variable con el valor **true** (verdadero) o **false** (falso)



## A

## char

Un tipo de datos de carácter representa un único carácter.

## "String"

Una cadena es una secuencia de caracteres. No es primitivo, pero ...

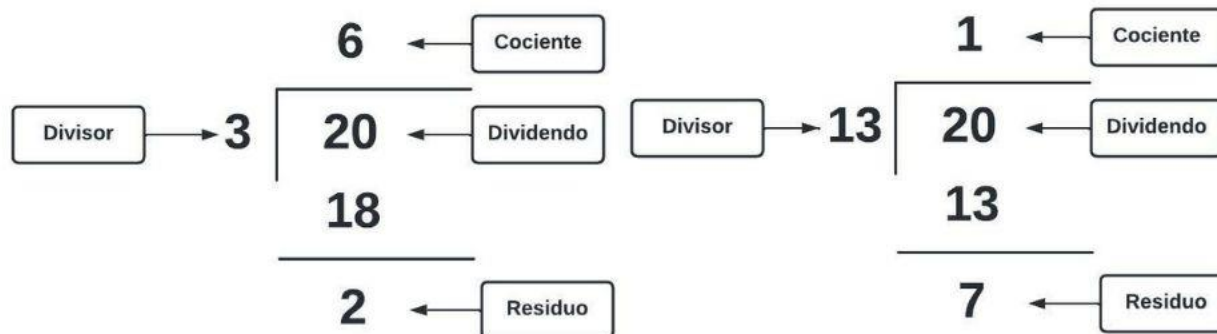




# Operadores aritméticos

Los operadores aritmeticos se utilizan para modificar variables.

Simbolo	Nombre	Ejemplo	Resultado
+	Adición	$34 + 1$	35
-	Sustracción	$34.0 - 0.1$	33.9
*	Multiplicación	$300 * 30$	9000
/	División	$1.0 / 2.0$	0.5
%	residuo o modulo	$20 \% 3$	2



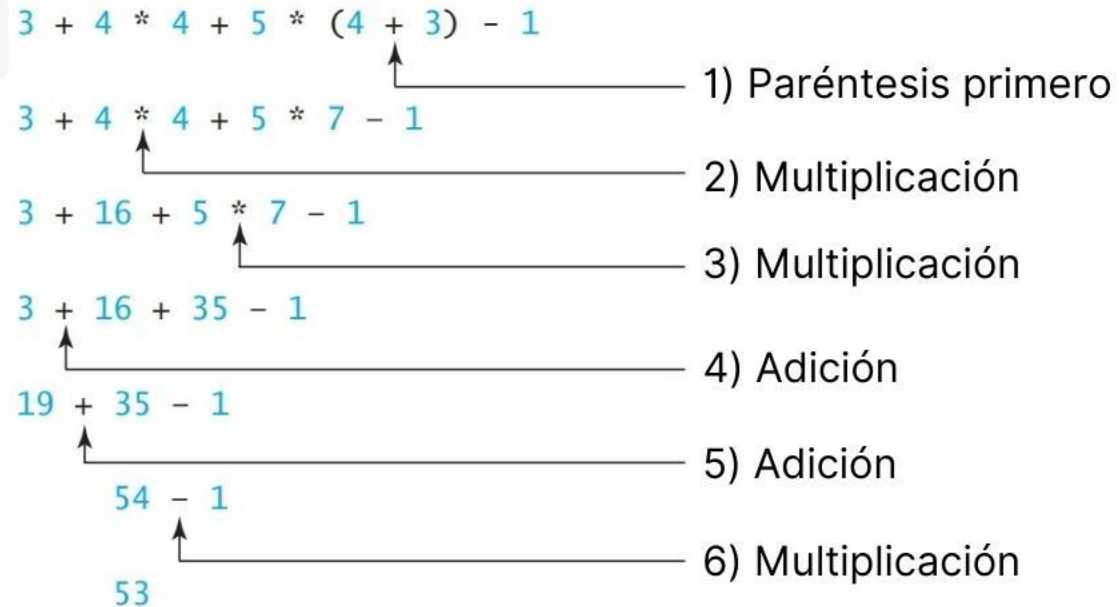


# Jerarquía de los operadores aritméticos

Los operadores aritmeticos se utilizan para modificar variables.

## Constantes

- La jerarquía funciona tal y como es en la aritmética.
- Primero los paréntesis, pudiendo estos estar anidados.
- Después multiplicación, división y resto, modulo o residuo. Si una expresión contiene varios operadores de multiplicación, división y modulo, se aplican de izquierda a derecha.
- Por ultimo adición y sustracción. Si una expresión contiene varios operadores de suma y resta, se aplican de izquierda a derecha.





## Ejercicio.

1. Crear una clase Numero en Java que reciba un dato numérico como String en su constructor y lo transforme a entero para transformarlo a número romano el número debe ser un número entre 0 y 3999, guardar el número en un objeto de la clase StringBuilder para imprimirlo con el método transformarARomano().