



# Curso de Java Standard



Ing. Octavio Robleto



octavio.robleto@gmail.com



<https://octaviorobleto.com>



# Introducción

Para manejar y/o manipular información dentro de un sistema necesitamos almacenarla en algún lugar mientras se procesan y así terminaran en una base de datos o un archivo que persista en el tiempo para después volverla a recuperar.

Para este propósito los lenguajes de programación nos ofrecen las Variables (Pueden cambiar su valor a los largo del tiempo) y Constantes (No cambian su valor una vez asignado) que se almacenaran de forma temporal en la memoria de nuestra computadora.

Por ello debemos asignarle un identificador “Nombre” a ese espacio de memoria que guardara nuestro dato.



# Identificador

Se puede decir que gran tiempo que dedicamos a programar es para darle un nombre a nuestras variables y constantes, porque aunque le podemos asignar cualquier identificador es recomendable darles un nombre significativo y que describa para que va a ser usado en nuestro software.

Hay ciertas reglas que debemos tomar en cuenta para asignarle los nombres:

- No deben contener espacios.
- Los únicos caracteres permitidos son el guion bajo \_ y el \$
- Pueden contener números y letras.
- Si posee números, estos no deben ir al inicio del identificador.

Ejemplos Variables: numero1, edad, nombre, \$contador, etc.

Ejemplos Constantes: IMPUESTO, VERSION, etc.

También es recomendable cuando necesitamos utilizar nombres compuestos usar las convenciones.

Para variables compuestas Lower Camel Case o Snake Case: razonSocial, nombre\_apellido.

Para constantes Upper Case con Snake Case: DIAS\_SEMANA, URL\_BUSCADOR, etc.



www.123456789.com

# Sintaxis

Recordemos que la sintaxis son las reglas que nos da el lenguaje para trabajar con él. La forma con la que se declaran las variables y Constantes en Java son:

## Sin darle un valor inicial:

1. Palabra reservada final (solo para constantes)
2. Tipo de dato.
3. Identificador.
4. Punto y Coma.

```
//Constantes
final tipoDato IDENTIFICADOR;

//Variables
tipoDato identificador;
```

## Con un valor inicial:

1. Palabra reservada final (solo para constantes)
2. Tipo de dato.
3. Identificador.
4. Valor.
5. Punto y Coma.

```
//Constantes
final tipoDato IDENTIFICADOR = valor;

//Variables
tipoDato identificador = valor;
```

# Tipos de datos

**Primitivos:** Podemos decir que son los elementales y fundamentales dentro de Java,, estos datos se pueden dividir en numéricos enteros, numéricos con punto flotante, booleanos y de carácter.

**Objetos:** son tipos de estructuras o datos especiales, que comprenderemos mejor mas adelante, por lo pronto conoceremos el mas usado de todos los objetos en Java, el **String**.

	Tipo	Definición	Espacio en Memoria	Valores
Primitivos	byte	Entero	1 byte	-128 a 127
	short	Entero	2 bytes	-32768 a 32767
	int	Entero	4 bytes	$2 \cdot 10^9$
	long	Entero	8 bytes	Muy grande
	float	Decimal simple	4 bytes	Muy grande
	double	Decimal doble	8 bytes	Muy grande
	char	Carácter simple	2 bytes	[caracteres simples]
	boolean	Verdadero y Falso	1 byte	[true/false]
Objetos	String	Cadena de caracteres	Depende de la cantidad de caracteres	Cadena de Caracteres

# Tipos de datos

## Como les asignamos valores?

**Numéricos Enteros:** literal del numero.

**Numéricos Decimales:** literal del numero con el punto como separador de decimales, algo particular que pasa con los **float** es que necesitan la letra “f” al final del número, ya que para Java todos los datos con decimales son de tipo **double** y así los podremos diferenciar.

**Booleanos:** literales true o false.

**Carácter:** carácter encerrado entre comillas simples.

**Cadena de Caracteres:** cadena encerrada entre comillas.

```
//Numeros

byte miByte = 5;

short miShort = 12596;

int miInt = 10;

long miLong = 1000000000;

float miFloat = 10.58f;

double miDouble = 10.58;

//Booleanos

boolean miBoolean = false;

//Caracteres

char miChar = 'P';

//Cadena de caracteres

String miString = "Hola Mundo";
```

# Secuencias de Escape

Suministran un mecanismo que permite expresar códigos de caracteres que de otro modo no se permitirían en los String.

Secuencia	Descripción
\n	Salto de línea. Sitúa el cursor al principio de la línea siguiente
\b	Retroceso. Mueve el cursor un carácter atrás en la línea actual.
\t	Tabulador horizontal. Mueve el cursor hacia adelante una distancia determinada por el tabulador.
\r	Ir al principio de la línea. Mueve el cursor al principio de la línea actual.
\f	Nueva página. Mueve el cursor al principio de la siguiente página.
\"	Comillas. Permite mostrar por pantalla el caracter comillas dobles.
\'	Comilla simple. Permite mostrar por pantalla el carácter comilla simple.
\\	Barra inversa.

```
System.out.println("Primera linea \n Segunda linea \t despues del tabulado...");
```