



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

DIVISIÓN SISTEMA UNIVERSIDAD ABIERTA Y EDUCACIÓN A DISTANCIA

**SUAYED**  
Sistema Universidad Abierta y  
Educación a Distancia

## **PROGRAMACION DE DISPOSITIV.MOVILES**

### **UNIDAD 3**

ACTIVIDAD TEORICA

**NOMBRE DEL ALUMNO:**

PAULINA RODRIGUEZ SAMPEDRO

**NOMBRE DEL ASESOR:**

CRISTIAN CARDOSO ARELLANO

**SEXTO SEMESTRE**

GRUPO: 9696

# APRENDER SOBRE VARIABLES, CONSTANTES, CADENAS Y OPERADORES

## Objetivo

- Comprender los conceptos básicos de programación, incluyendo el uso de constantes y variables, la seguridad de tipos, los tipos de datos primitivos (numéricos, enteros, de punto flotante, booleanos y de cadena), así como estructuras como enumeraciones.
- Identificar los diferentes operadores (de asignación, comparación, aritméticos, lógicos y compuestos), con el fin de que más adelante se puedan desarrollar programas sólidos, seguros y eficientes.

## Definición de constantes y variables

Una variable es un espacio en la memoria utilizado para almacenar datos cuyo valor puede cambiar durante la ejecución del programa, mientras que la constante es un valor fijo que no se puede modificar una vez asignado.

## Tipo safety

Es la propiedad de un lenguaje de programación que garantiza que las operaciones se realizan únicamente sobre datos de tipos compatibles. En Java, esto significa que el compilador y la máquina virtual verifican que las variables, métodos y expresiones usen tipos correctos, evitando errores como asignar un número a una cadena o invocar métodos inexistentes en un objeto.

## Tipos numéricos

Java define **tipos primitivos numéricos** que se dividen en dos grandes grupos:

### 1. Enteros (sin decimales):

- byte: 8 bits con rango de -128 a 127
- short: 16 bits con rango de -32,768 a 32,767
- int: 32 bits con rango de -2,147,483,648 a 2,147,483,647
- long: 64 bits con rango de -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807

### 2. De punto flotante (con decimales):

- float: 32 bits con precisión simple
- double: 64 bits con doble precisión.

## **El tipo booleano**

El tipo de dato boolean solo tiene dos valores posibles: true (verdadero) y false (falso).

## **El tipo de cadena (string)**

Una cadena es una secuencia de caracteres, usada para representar texto.

## **Enums**

Un tipo de enumeración es un tipo de dato especial que permite que una variable sea un conjunto de constantes predefinidas. La variable debe ser igual a uno de los valores predefinidos. Algunos ejemplos comunes son las direcciones de la brújula (valores NORTE, SUR, ESTE y OESTE) y los días de la semana.

En el lenguaje de programación Java, se define un tipo de enumeración mediante la enum palabra clave.

## **Operadores**

Los operadores son símbolos especiales que realizan operaciones específicas con uno, dos o tres operandos y devuelven un resultado.

### **El operador de asignación**

Uno de los operadores más comunes es el operador de asignación simple " =", el cual asigna el valor de su derecha al operando de su izquierda.

### **Operadores de comparación**

Son símbolos que se utilizan para comparar dos valores (operandos). El resultado de esta evaluación es siempre un valor de tipo primitivo booleano: true (verdadero) o false (falso). Estos son == (igual a), != (distinto), < (menor que), > (mayor que), <= (menor o igual que), >= (mayor o igual que).

### **Operadores aritméticos**

Realizan operaciones matemáticas básicas como + (sumas), - (restas), \* (multiplicaciones), / (divisiones) y % (módulo o resto).

### **El operador remainder (%)**

En Java es un operador aritmético que se utiliza para obtener el residuo o resto que resulta de dividir un número entre otro, después de realizar la división entera o real.

### **Operadores de asignación compuestos**

Son una forma abreviada de escribir expresiones de asignación en las que se realiza una operación aritmética o lógica y se asigna el resultado a la misma variable de origen. Los operadores aritméticos compuestos son +=(suma y asignación), -=(resta y asignación), \*= (multiplicación y asignación), /= (división y asignación), %=

(residuo y asignación). Los operadores lógicos compuestos son &= (AND de bits y asignación), ^= (XOR de bits y asignación), |= (OR de bits y asignación).

### **El operador condicional ternario**

Java cuenta con el operador condicional (?:), que en ocasiones puede utilizarse en lugar de una instrucción if... else. Éste es el único operador ternario en java que utiliza tres operandos. En conjunto, los operandos y ? : forman una expresión condicional. El primer operando (a la izquierda de ?) es una expresión boolean (es decir, una condición), el segundo operando (entre el ? y : ) es el valor de la expresión condicional si la expresión boolean es verdadera, y el tercer operando (a la derecha : ) es el valor de la expresión condicional si la expresión boolean se evalúa como false.

### **El operador lógico NOT (!)**

Conocido como el operador de complemento lógico. Es un operador unario (solo actúa sobre un operando). Su función es invertir el valor booleano de su operando: si es true lo convierte en false, y viceversa.

### **El operador lógico AND (&&)**

Es un operador binario que devuelve true únicamente si ambos operandos son verdaderos.

### **El operador lógico OR (||)**

Operador binario que devuelve true si al menos uno de los dos operandos es verdadero. Solo devuelve false si ambos son falsos.

## **Conclusión**

El estudio de los conceptos básicos de programación como lo son las constantes y variables, la seguridad de tipos, los tipos de datos primitivos (numéricos, booleanos y de cadena), y los operadores (aritméticos, de comparación, lógicos y de asignación compuesta) constituye la base indispensable para el desarrollo de software confiable y eficiente.

Comprender estos fundamentos permite estructurar programas claros, prevenir errores comunes y aprovechar al máximo las capacidades del lenguaje, garantizando que las operaciones se realicen de manera segura y coherente. Además, estos conocimientos no solo fortalecen las competencias técnicas, sino que también prepara al futuro profesional en informática desenvolverse con solvencia en proyectos académicos y profesionales, consolidando las bases de su formación en informática.

## Referencias

- Programación Java (s.f.). *Variables y tipos de datos*. [http://profesores.fi-b.unam.mx/carlos/java/java\\_basico2\\_6.html](http://profesores.fi-b.unam.mx/carlos/java/java_basico2_6.html)
- McGraw, G., & Felten, E. (1999). *Type safety (Section 10)*. In *Securing Java: Getting down to business with mobile code*. John Wiley & Sons, Inc. <https://www.securingjava.com/chapter-two/chapter-two-10.html>
- Oracle. (n.d.). *Primitive data types*. In *The Java™ Tutorials — Learning the Java Language: Language Basics*. Oracle. <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- Oracle. (n.d.). *Operators*. In *The Java™ Tutorials — Learning the Java Language: Language Basics*. Oracle. <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html>
- Romo, C. (s. f.). *II – Expresiones y operadores*. Universidad Nacional Autónoma de México (FI-B). [https://profesores.fi-b.unam.mx/carlos/java/java\\_basico2\\_5.html](https://profesores.fi-b.unam.mx/carlos/java/java_basico2_5.html)
- IONOS México. (s. f.). *Visión general de los operadores Java más importantes*. IONOS. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/operadores-javascript/>
- Deitel,H.,Deitel,P.(2003) . *Como programar en java*. Pearson educación.