



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

# » El entorno Java para la programación

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ *Variables en Java*
- ✓ *Tipos de Datos en Java*

# LEARNING PATHWAY

4.3

Start! 🏁

El Entorno Java para la programación

Operadores en Java

Operando

Operadores en Java:

- Aritmeticos
- Unarios
- Relacionales
- Condicionales

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



**Reconocer la implementación y manipulación de los operadores en Java**





# Rompehielo

## **Operando:** ¿Algún voluntario?

Levanten la mano para ser voluntarios de la calculadora humana! La idea es que vayas escribiendo todas las operaciones que pondremos como ejemplo. Serás la pantalla de la calculadora. Las operaciones serán aritméticas y lógicas.

## **Consigna:**

El moderador dirá en voz alta una operación matemática simple, por ejemplo: "2 + 4"

Los participantes deberán indicar en el chat qué operador aritmético corresponde: en este caso, el signo "+".

El voluntario anotará en su "pantalla" la operación con el operador que los participantes indiquen: 2 + 4

**Los participantes deberán identificar el operador correcto para que el voluntario lo añada.**

# › Operadores en Java





# Operadores en Java



## ¿Qué son los operadores?:

En programación un operador representa un símbolo que permite realizar operaciones aritméticas, relacionar elementos o hacer preguntas donde se involucra más de una condición.



Un **operador lleva a cabo operaciones** sobre uno (operador unario), dos (operador binario) o tres (operador ternario) datos u operandos de tipo primitivo devolviendo un valor determinado también de un tipo primitivo. El tipo de valor devuelto tras la evaluación depende del operador y del tipo de los operandos. Por ejemplo, los operadores aritméticos trabajan con operandos numéricos, llevan a cabo operaciones aritméticas básicas y devuelven el valor numérico correspondiente.





# Operadores en Java

## Operadores aritméticos



Los operadores aritméticos en Java son aquellos que nos permiten realizar operaciones matemáticas: **suma**, **resta**, **multiplicación**, **división** y **resto**.

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
-	operador unario de cambio de signo	-4	-4
+	Suma	2.5 + 7.1	9.6
-	Resta	235.6 - 103.5	132.1
*	Producto	1.2 * 1.1	1.32
/	División (tanto entera como real)	0.050 / 0.2 7 / 2	0.25 3
%	Resto de la división entera	20 % 7	6





# Operadores en Java



## Operadores unarios

Los operadores unarios en Java son aquellos que solo requieren un operando para funcionar.



Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
++	Incremento	4++	5
	i++ primero se utiliza la variable y luego se incrementa su valor ++i primero se incrementa el valor de la variable y luego se utiliza	a=5; b=a++; a=5; b=++a;	a vale 6 y b vale 5 a vale 6 y b vale 6
--	decremento	4--	3



# Operadores en Java

## Operadores de relación

Realizan comparaciones entre datos compatibles de tipos primitivos (numéricos, carácter y booleanos) teniendo siempre un resultado booleano. Los operandos booleanos sólo pueden emplear los operadores de igualdad y desigualdad.



Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
<code>==</code>	igual que	<code>7 == 38</code>	false
<code>!=</code>	distinto que	<code>'a' != 'k'</code>	true
<code>&lt;</code>	menor que	<code>'G' &lt; 'B'</code>	false
<code>&gt;</code>	mayor que	<code>'b' &gt; 'a'</code>	true
<code>&lt;=</code>	menor o igual que	<code>7.5 &lt;= 7.38</code>	false
<code>&gt;=</code>	mayor o igual que	<code>38 &gt;= 7</code>	true





# Operadores en Java

## Operadores lógicos



Realizan operaciones sobre datos booleanos y tienen como resultado un valor booleano.

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
!	Negación - NOT (unario)	<code>!false</code> <code>!(5==5)</code>	<code>true</code> <code>false</code>
	Suma lógica – OR (binario)	<code>true   false</code> <code>(5==5)   (5&lt;4)</code>	<code>true</code> <code>true</code>
^	Suma lógica exclusiva – XOR (binario)	<code>true ^ false</code> <code>(5==5)   (5&lt;4)</code>	<code>true</code> <code>true</code>
&	Producto lógico – AND (binario)	<code>true &amp; false</code> <code>(5==5) &amp; (5&lt;4)</code>	<code>false</code> <code>false</code>
	Suma lógica con cortocircuito: si el primer operando es <code>true</code> entonces el segundo se salta y el resultado es <code>true</code>	<code>true    false</code> <code>(5==5)    (5&lt;4)</code>	<code>true</code> <code>true</code>
&&	Producto lógico con cortocircuito: si el primer operando es <code>false</code> entonces el segundo se salta y el resultado es <code>false</code>	<code>false &amp;&amp; true</code> <code>(5==5) &amp;&amp; (5&lt;4)</code>	<code>false</code> <code>false</code>





# Operadores en Java



## Operadores lógicos

Para mejorar el rendimiento de ejecución del código **es recomendable** emplear en las expresiones booleanas el **operador &&** en lugar del operador &. En este caso es conveniente situar la condición más propensa a ser falsa en el término de la izquierda.



Esta técnica puede reducir el tiempo de ejecución del programa.

De forma equivalente es preferible emplear el **operador ||** al operador |. En este caso es conveniente colocar la condición más propensa a ser verdadera en el término de la izquierda.



# LIVE CODING

Ejemplo en vivo

## Practicando operaciones:

*Vamos a realizar diversas operaciones en Java con Eclipse, y vamos a imprimir por pantalla los resultados de dichas operaciones:*

- 1. Desarrollar operaciones aritméticas simples (suma, resta, multiplicación y división).*
- 2. Desarrollar operaciones de incremento, utilizando el operador ++.*
- 3. Desarrollar operaciones lógicas, utilizando && y ||*

 **Tiempo:** 25 minutos





# **Ejercicio N° 1**

# Operando





# Operando

**Vamos a poner en práctica lo aprendido:** 🙌

En la programación, la necesidad de realizar operaciones es constante. Es por esto que te invitamos a practicar los conceptos vistos en tu IDE Eclipse. Debes mostrar todos los resultados por pantalla.



**Consigna:** ✍️

- Pedir al usuario dos números y mostrar su suma
- Leer un número y mostrar su cuadrado elevándolo al exponente 2
- Leer dos palabras y concatenarlas en una frase usando el operador +
- Pedir dos números y comparar si son iguales con == mostrando el resultado

**Tiempo** 🕒: 25 minutos

○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ **Conocer e implementar el uso de operadores en programas Java.**



# #WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 🙌 🙌 🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. Material 1 (Foro)
  - b. *Lectura Módulo 4, Lección 3: páginas 4 - 5*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

# ¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

# Time-out!

🕒 5 min.

