



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊 

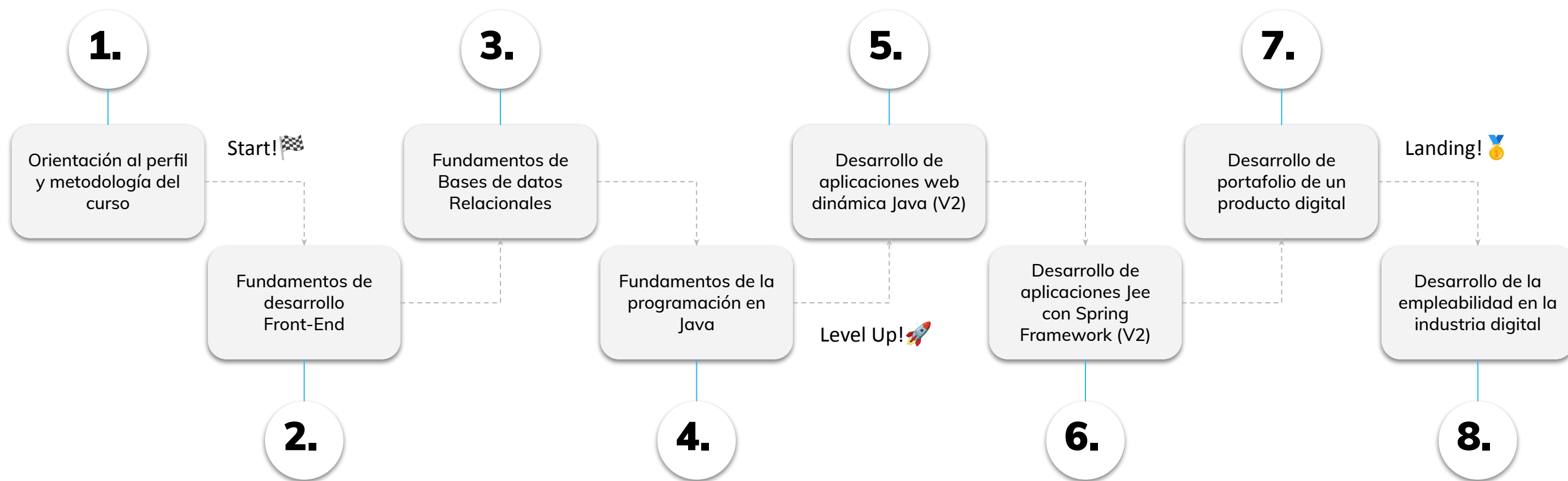
# » Algoritmos

---

**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 :

- ✓ Conceptos y características de los algoritmos
- ✓ Variables
- ✓ Tipos de datos

# LEARNING PATHWAY

4.2

Start! 🏁

## Fundamentos de la programación en Java

Expresiones aritméticas,  
operadores aritméticos y  
relacionales.

Operadores aritméticos

Superliga

Operadores relacionales

Comparando

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



**Comprender el uso de las expresiones aritméticas en un algoritmo.**



**Conocer la utilización de los operadores aritméticos.**



**Entender la implementación de los operadores relacionales.**





# Rompehielo



## ¿Cuándo son necesarias las expresiones aritméticas?:

¿Pueden dar ejemplos de cuándo utilizan operaciones matemáticas en su vida cotidiana? (al cocinar, en las compras, en el trabajo, etc)



## Consigna: Responder en el chat

¿Qué ejemplos de alguna tarea específica se les ocurren donde sea necesario utilizar suma, resta, multiplicación y división todos juntos?



## Paso a paso:

Piensen el ejemplo como un algoritmo! y compartan cómo sería el paso a paso para resolver la tarea.

# › Expresiones aritméticas





# Expresiones aritméticas



**Las expresiones aritméticas son similares a fórmulas matemáticas.**

En el código de la computadora, los operadores y las funciones se ejecutan a través de conjuntos de expresiones que incluyen expresiones aritméticas, de caracteres y booleanas. Estos proporcionan la base para los tipos de trabajo de datos que se realizan dentro de un programa de software.



Existen diferentes tipos de expresiones aritméticas en la sintaxis de programación: enteros o números reales, y números de punto flotante (para identificar y almacenar números complejos que pueden no encajar en un valor entero).





# Operadores aritméticos



## Las operadores aritméticos son:

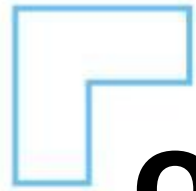
- + Suma
- - Resta
- \* Multiplicación
- / División
- -, \*\*, ^ Exponenciación
- div División entera (cociente)
- mod Módulo (resto de una división)



Los operadores se utilizan de igual forma que en matemáticas. Por consiguiente:

**A x B** se escribe en algoritmo como **A \* B**,   
y **1/4 x C** como **C/4**.

Al igual que en matemáticas, el signo menos juega un doble papel, como resta en **A - B** y como cambio de signo en **- A**.



# Operadores aritméticos



**Operador DIV:  $A \text{ div } B = x$**

Obtiene la parte entera de  $A/B$ .

**Operador MOD:  $A \text{ mod } B = x$**

Obtiene el resto (módulo) de  $A/B$

<u>División</u>	<u>División entera</u>
$\begin{array}{r} 63.0 \overline{) 2} \\ 03 \phantom{00} \\ 10 \phantom{00} \\ \underline{00} \end{array}$	$\begin{array}{r} 63 \overline{) 2} \\ 03 \phantom{00} \\ 1 \phantom{00} \\ \underline{00} \end{array}$
$31.5 \leftarrow \text{Cociente } (/)$	$31 \leftarrow \text{Cociente } (//)$
	$1 \leftarrow \text{Resto } (\%)$





# Operadores aritméticos



## Reglas de prioridad

Las expresiones que tienen dos o más operandos, responden a **reglas matemáticas** que permiten determinar **el orden de las operaciones**, se denominan reglas de prioridad o precedencia.

Las operaciones que están encerradas entre **paréntesis se evalúan primero**. Si existen diferentes paréntesis anidados (interiores unos a otros), las expresiones más internas se evalúan primero.





# Operadores aritméticos



## Reglas de prioridad

Las operaciones aritméticas dentro de una expresión suelen seguir el siguiente orden de prioridad:



- Operador exponencial (, o bien \*\*).
- Operadores \*, /, \,
- Operadores div y mod.
- Operadores +, -.

En caso de coincidir varios operadores de igual prioridad en una expresión o subexpresión encerrada entre paréntesis, el orden de prioridad en este caso es de izquierda a derecha.



# LIVE CODING

Ejemplo en vivo

## Sacando cuentas:

*Vamos a crear un algoritmo para realizar todas las operaciones aritméticas vistas hasta ahora. Para esto será necesario:*

1. **Entrada:** 2 variables numéricas
2. **Proceso:** operaciones aritméticas
3. **Salida:** 1 variable numérica por cada operación.

 **Tiempo: 15 minutos**



# **Ejercicio N° 1**

# Superliga



# Superliga

## ¡A contar puntos! 🙌

Las operaciones aritméticas son utilizadas a menudo en todo tipo de contextos. En este caso, vamos a aplicarlo al contexto de las competiciones deportivas. En salas reducidas, deberán crear un algoritmo que permita resolver el problema propuesto. Luego, compartir con el resto del curso cuáles fueron las partes más difíciles de definir.

### Consigna: 🛠️

Elaborar un algoritmo que permita ingresar el número de partidos ganados, perdidos y empatados por tu equipo favorito en la liga actual. Se debe mostrar el puntaje total, teniendo en cuenta que por cada partido ganado obtendrá 3 puntos, empatado 1 punto y perdido 0 puntos.

**Tiempo** 🕒: 20 minutos

### Paso a paso: ⚙️

Los datos de entrada serán: cantidad de partidos ganados (pg), cantidad de partidos empatados (pe), cantidad de partidos perdidos.

También debes crear variables para calcular el puntaje por partidos ganados (ppg) y puntaje por partidos empatados (ppe). La suma de estas dos daría el puntaje total (ptotal).





# › Operadores Relacionales



# Operadores Relacionales



Los operadores relacionales, también conocidos como operadores de comparación o lógicos, son símbolos que se utilizan en programación **para comparar dos valores** y obtener como **resultado un valor lógico** verdadero o falso.



Estos operadores son muy útiles en programación para comparar valores y expresiones, y para evaluar condiciones que dirigen la lógica de un programa. Se utilizan habitualmente para realizar validaciones.





## Los principales operadores relacionales son:

- **Mayor >** : Compara si el valor de la izquierda es mayor que el de la derecha. Ejemplo:  $7 > 5$  devuelve true.
- **Menor <** : Comprueba si el valor de la izquierda es menor que el de la derecha. Ejemplo:  $3 < 8$  devuelve true.
- **Mayor o igual >=** : Verifica si el valor de la izquierda es mayor o igual al de la derecha. Ejemplo:  $5 \geq 5$  devuelve true.
- **Menor o igual <=** : Revisa si el valor de la izquierda es menor o igual al de la derecha. Ejemplo:  $5 \leq 10$  devuelve true.
- **Igual ==** : Compara si dos valores son iguales. Ejemplo:  $5 == 5$  devuelve true.
- **Distinto !=** : Compara si dos valores son diferentes. Ejemplo:  $5 \neq 7$  devuelve true.



Operadores relacionales	
>	Mayor.
<	Menor.
>=	Mayor o igual.
<=	Menor o igual.
==	Igual.
!=	Distinto.
=	Asignación.



# Operadores Relacionales



## Operador de asignación ( = )

El operador ( = ) es un operador sobrecargado, es decir que dependiendo del contexto realiza distintas operaciones.



En el contexto de la **asignación**, el operador = se utiliza para modificar el valor de las variables, es decir, asignarle un nuevo valor. Ejemplo:  $x = 5$ .

Si lo que queremos es utilizar este operador como sinónimo de **igualdad** y no de asignación, se utiliza el operador dos veces: ( == )



# LIVE CODING

Ejemplo en vivo

## Practicando asignaciones:

*Declararemos cuatro variables de **tipo entero** **A**, **B**, **C** y **D** y le asignaremos un valor diferente a cada una. Vamos a utilizar sólo una variable auxiliar.*

*A continuación, realizar las instrucciones necesarias para que:*

- ***B** tome el valor de **C***
- ***C** tome el valor de **A***
- ***A** tome el valor de **D***
- ***D** tome el valor de **B**.*

**Tiempo: 20 minutos**



## **Ejercicio N° 2**

# Comparando



# Comparando

¡A comparar operaciones! 🙌

Muchas veces es necesario comparar diferentes datos para resolver un problema. En este caso, vamos a evaluar diferentes operaciones y deberemos decidir qué valor lógico (verdadero o falso) responde a cada una.



**Consigna:** ✍️

- Evaluar el valor de verdad de la siguiente expresión:  **$15 \text{ MOD } 2 = 1$**
- Evaluar el valor de verdad de la siguiente expresión:  **$(10 * 10) \leq (10^2)$**

**Recuerda:** ⚙️

- Reglas de prioridad de las expresiones.
- Los resultados sólo pueden ser 2: verdadero o falso.

**Tiempo** 🕒: 10 minutos

○

# ¿Alguna consulta?

+





# RESUMEN

¿Qué logramos en esta clase?

- ✓ Reconocer los operadores aritméticos y su implementación en programación.
- ✓ Comprender los operadores relacionales y los valores lógicos en programación.
- ✓ Identificar las diferencias entre comparación y asignación.



# #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 2: páginas 8 - 11*
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.

