

¿Usar if-else o switch-case?

 ¿Qué diferencias existen?

Tanto if-else como switch-case permiten tomar decisiones en Java, pero tienen diferencias clave en su uso y restricciones:

1. Expresiones Permitidas

- if-else: Admite condiciones más flexibles, como comparaciones (>, <, !=), expresiones booleanas y combinaciones lógicas (&&, ||).
- switch-case: Solo evalúa igualdad (==) con valores discretos como int, char, String, enum y byte (desde Java 7 también String). No admite comparaciones numéricas ni expresiones booleanas.

2. Uso en Rango de Valores

- if-else: Útil cuando se necesita evaluar rangos de valores (ej. if (x >= 10 && x <= 20)).
- switch-case: Mejor cuando hay que comparar valores específicos (case 10: case 20).

3. Legibilidad y Organización

- if-else: Puede volverse más complejo y difícil de leer cuando hay muchas condiciones anidadas.
- switch-case: Es más limpio y estructurado cuando se manejan múltiples casos distintos de un mismo valor.

4. Ejecución y Eficiencia

- if-else: Se evalúa en orden de arriba hacia abajo hasta encontrar una condición verdadera. Puede ser más lento si hay muchas condiciones.
- switch-case: Puede ser más eficiente, ya que en algunos casos se compila como una tabla de búsqueda en lugar de evaluar condiciones secuenciales.

5. Uso de default y else

- if-else: Se usa else para manejar todos los demás casos.
- switch-case: Se usa default para capturar casos no especificados.

¿Cuándo usar cada uno?

Usa if-else cuando:

- Se necesiten **comparaciones complejas** o rangos (>, <, !=, &&, ||).
- Se trabaje con valores **booleanos**.

Usa switch-case cuando:

- Se compare un mismo valor contra **opciones fijas**.
- Se manejen muchos casos discretos y se quiera mejorar la **legibilidad**.

En general, si se puede usar **switch**, suele ser más claro y eficiente, pero **if-else** es más versátil.