Ejemplo 1: Operadores Aritméticos

```java
public class OperadoresAritmeticos {
    public static void main(String[] args) {
        // Declaramos variables enteras a y b
        int a = 10;
        int b = 3;

        // Suma
        int suma = a + b; // suma = 10 + 3 = 13

        // Resta
        int resta = a - b; // resta = 10 - 3 = 7

        // Multiplicación
        int multiplicacion = a * b; // multiplicacion = 10 * 3 = 30

        // División
        int division = a / b; // division = 10 / 3 = 3 (división entera)

        // Módulo
        int modulo = a % b; // modulo = 10 % 3 = 1 (resto de la división)

        // Incremento
        a++; // a = a + 1 => a = 11

        // Decremento
        b--; // b = b - 1 => b = 2

        // Imprimir resultados
        System.out.println("Suma: " + suma);
        System.out.println("Resta: " + resta);
        System.out.println("Multiplicación: " + multiplicacion);
        System.out.println("División: " + division);
        System.out.println("Módulo: " + modulo);
        System.out.println("Nuevo valor de a (incrementado): " + a);
        System.out.println("Nuevo valor de b (decrementado): " + b);
    }
}
```

Explicación línea por línea:

1. **Declaración de la clase:**

   ```java
   public class OperadoresAritmeticos {
   ```

   - Se define una clase pública llamada `OperadoresAritmeticos`.

2. **Método principal:**

   ```java
   public static void main(String[] args) {
   ```

   - Punto de entrada del programa donde se ejecutará el código.

3. **Declaración de variables:**

   ```java
   int a = 10;
   int b = 3;
   ```

   - Se declaran dos variables enteras `a` y `b` con valores iniciales de 10 y 3, respectivamente.

4. **Operaciones aritméticas:**
   - **Suma:**

     ```java
     int suma = a + b; // suma = 10 + 3 = 13
     ```

     - Se suma `a` y `b` y se almacena en `suma`.
   - **Resta:**

```
int resta = a - b; // resta = 10 - 3 = 7
```

- Se resta `b` de `a` y se almacena en `resta`.

- **Multiplicación:**

```
int multiplicacion = a * b; // multiplicacion = 10 * 3 = 30
```

- Se multiplica `a` por `b` y se almacena en `multiplicacion`.

- **División:**

```
int division = a / b; // division = 10 / 3 = 3
```

- Se divide `a` entre `b` y se almacena en `division`. Nota: Al ser enteros, el resultado es la parte entera.

- **Módulo:**

```
int modulo = a % b; // modulo = 10 % 3 = 1
```

- Se obtiene el resto de dividir `a` entre `b` y se almacena en `modulo`.

5. **Operadores de incremento y decremento:**

- **Incremento:**

```
a++; // a = a + 1 => a = 11
```

- Se incrementa `a` en 1.

- **Decremento:**

```
b--; // b = b - 1 => b = 2
```

- Se decrementa `b` en 1.

6. **Impresión de resultados:**

```
System.out.println("Suma: " + suma);
System.out.println("Resta: " + resta);
System.out.println("Multiplicación: " + multiplicacion);
System.out.println("División: " + division);
System.out.println("Módulo: " + modulo);
System.out.println("Nuevo valor de a (incrementado): " + a);
System.out.println("Nuevo valor de b (decrementado): " + b);
```

- Se imprimen los resultados de las operaciones anteriores.

Resultado al ejecutar el programa:

```
Suma: 13
Resta: 7
Multiplicación: 30
División: 3
Módulo: 1
Nuevo valor de a (incrementado): 11
Nuevo valor de b (decrementado): 2
```

Ejemplo 2: Operadores Lógicos

```
public class OperadoresLogicos {
    public static void main(String[] args) {
        // Declaramos variables booleanas
        boolean x = true;
        boolean y = false;

        // Operador AND lógico
```

```
        boolean resultadoAnd = x && y; // resultadoAnd = true && false = false

        // Operador OR lógico
        boolean resultadoOr = x || y; // resultadoOr = true || false = true

        // Operador NOT lógico
        boolean resultadoNot = !x; // resultadoNot = !true = false

        // Operador XOR lógico
        boolean resultadoXor = x ^ y; // resultadoXor = true ^ false = true

        // Imprimir resultados
        System.out.println("Resultado AND: " + resultadoAnd);
        System.out.println("Resultado OR: " + resultadoOr);
        System.out.println("Resultado NOT: " + resultadoNot);
        System.out.println("Resultado XOR: " + resultadoXor);
    }
}
```

Explicación línea por línea:

1. **Declaración de la clase y método principal:** Igual que antes.
2. **Declaración de variables booleanas:**

   ```
   boolean x = true;
   boolean y = false;
   ```

   - Se declaran dos variables booleanas `x` y `y`.
3. **Operaciones lógicas:**
   - **AND lógico (`&&`):**

     ```
     boolean resultadoAnd = x && y; // false
     ```

     - El resultado es `true` solo si ambos operandos son `true`.
   - **OR lógico (`||`):**

     ```
     boolean resultadoOr = x || y; // true
     ```

     - El resultado es `true` si al menos uno de los operandos es `true`.
   - **NOT lógico (`!`):**

     ```
     boolean resultadoNot = !x; // false
     ```

     - Invierte el valor lógico de `x`.
   - **XOR lógico (`^`):**

     ```
     boolean resultadoXor = x ^ y; // true
     ```

     - El resultado es `true` si los operandos son diferentes.
4. **Impresión de resultados:** Igual que antes.

Resultado al ejecutar el programa:

```
Resultado AND: false
Resultado OR: true
Resultado NOT: false
Resultado XOR: true
```

Ejemplo 3: Uso Práctico Combinado

```
public class UsoPracticoOperadores {
    public static void main(String[] args) {
```

```
        // Variables de ejemplo
        int edad = 25;
        double salario = 3500.50;
        boolean tieneExperiencia = true;

        // Verificar si cumple con los requisitos para un puesto de trabajo
        boolean esElegible = (edad >= 18 && salario >= 3000) && tieneExperiencia;

        // Imprimir resultado
        System.out.println("¿El candidato es elegible? " + esElegible);
    }
}
```

Explicación línea por línea:

1. **Variables de ejemplo:**

   ```
   int edad = 25;
   double salario = 3500.50;
   boolean tieneExperiencia = true;
   ```

   - Se asignan valores a las variables que podrían ser criterios para un puesto de trabajo.

2. **Evaluación de elegibilidad:**

   ```
   boolean esElegible = (edad >= 18 && salario >= 3000) && tieneExperiencia;
   ```

   - **Primera condición (`edad >= 18`):** Verifica si el candidato es mayor de edad.
   - **Segunda condición (`salario >= 3000`):** Verifica si el salario esperado es de al menos 3000.
   - **Tercera condición (`tieneExperiencia`):** Verifica si el candidato tiene experiencia.
   - **Operadores lógicos:** Se usan `&&` para asegurarse de que todas las condiciones se cumplan.

3. **Impresión del resultado:**

   ```
   System.out.println("¿El candidato es elegible? " + esElegible);
   ```

   - Muestra si el candidato cumple con todos los requisitos.

Resultado al ejecutar el programa:

```
¿El candidato es elegible? true
```

Explicación del resultado:

- Dado que:
  - `edad >= 18` es `true` (25 >= 18)
  - `salario >= 3000` es `true` (3500.50 >= 3000)
  - `tieneExperiencia` es `true`
- Todas las condiciones son `true`, por lo que `esElegible` es `true`.

Resumen

- **Operadores Aritméticos:**
  - `+` Suma
  - `-` Resta
  - `*` Multiplicación
  - `/` División
  - `%` Módulo (resto de la división)
  - `++` Incremento
  - `--` Decremento
- **Operadores Lógicos:**
  - `&&` AND lógico
  - `||` OR lógico

- `!` NOT lógico
- `^` XOR lógico

Estos ejemplos prácticos deberían ayudarte a comprender cómo funcionan los diferentes operadores en Java y cómo pueden aplicarse en situaciones reales.