

Programación I

Estructuras de control de Flujo: IF y Ternarios

Operadores relacionales y booleanos

Operadores relacionales y booleanos

Las estructuras de control de flujo (condicionales y bucles) hacen uso de operadores relacionales (comparación) y booleanos (lógicos) para construir las condiciones a cumplir.

En las siguientes tablas pueden verse ambos grupos de operadores:

Operador	Significado	Ejemplo	Resultado
==	Igual a	5 == 5	TRUE
!=	No igual a	5 != 5	FALSE
>	Mayor que	10 > 5	TRUE
<	Menor que	5 < 10	TRUE
>=	Mayor o igual	5 >= 5	TRUE
<=	Menor o igual	5 <= 10	TRUE
&&	Y lógico	true && true	TRUE
	O lógico	true false	TRUE
!	No lógico	!true	FALSE

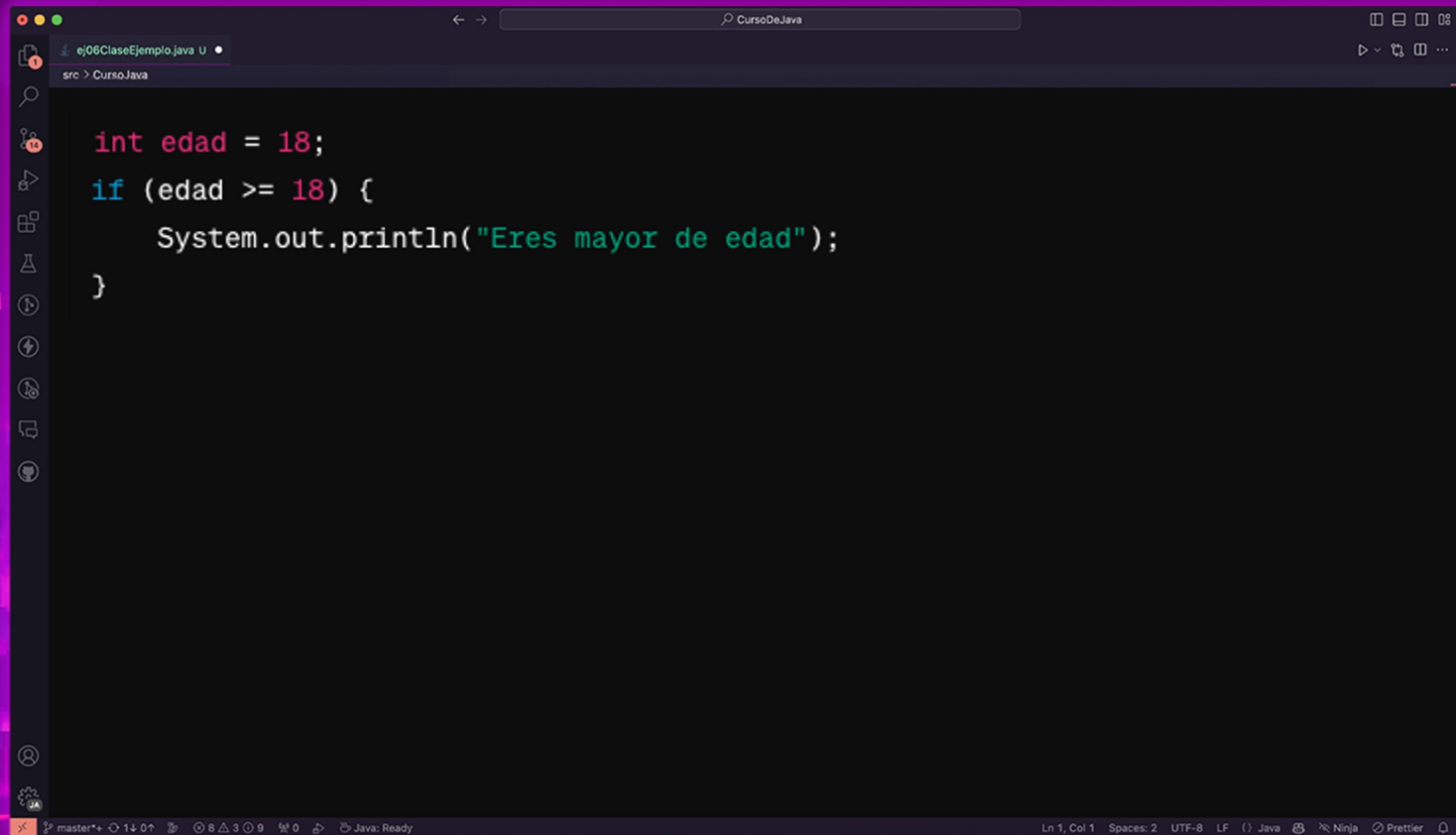
Condicional IF

Condicional IF

El condicional if en Java se utiliza para ejecutar un bloque de código si una condición especificada es verdadera.

```
if (condición) {  
    // Ejecutar si la condición  
    //es verdadera  
}
```

Condicional IF



The image shows a screenshot of an IDE window titled "CursoDeJava". The code editor displays a Java program snippet. The code defines an integer variable "edad" with the value 18, and then uses an "if" statement to check if "edad" is greater than or equal to 18. If the condition is true, it prints the message "Eres mayor de edad" to the console. The IDE interface includes a sidebar with various icons on the left, a top toolbar with window and search icons, and a bottom status bar showing the current file path, line and column numbers, encoding, and other settings.

```
int edad = 18;
if (edad >= 18) {
    System.out.println("Eres mayor de edad");
}
```

Condicional IF-ELSE

El condicional if-else en Java se utiliza para ejecutar un bloque de código si la condición especificada es verdadera, y otro bloque de código si la condición es falsa.

```
if (condición) {  
    // Ejecutar si la condición es verdadera  
} else {  
    // Ejecutar si la condición es falsa  
}
```

Condicional IF-ELSE

```
ej06ClaseEjemplo.java U
src > CursoJava

int edad = 16;
if (edad >= 18) {
    System.out.println("Eres mayor de edad");
} else {
    System.out.println("Eres menor de edad");
}
```

master*+ 14:07 8 3 9 0 Java: Ready Ln 1, Col 1 Spaces: 2 UTF-8 LF () Java Ninja Prettier

En este ejemplo, si la variable edad es mayor o igual a 18, se imprimirá "Eres mayor de edad"; de lo contrario, se imprimirá "Eres menor de edad".

Condicional IF-ELSE anidados

Los condicionales if-else anidados en Java permiten verificar múltiples condiciones.

```
if (condición1) {  
    // Ejecutar si la condición1 es verdadera  
} else if (condición2) {  
    // Ejecutar si la condición2 es verdadera  
} else {  
    // Si ninguna de las anteriores es verdadera  
}
```

Condicional IF-ELSE anidados

```
ej06ClaseEjemplo.java U
src > CursoJava

int edad = 21;
if (edad < 18) {
    System.out.println("Eres menor de edad");
} else if (edad >= 18 && edad < 21) {
    System.out.println("Eres mayor de edad pero aún no puedes beber alcohol");
} else {
    System.out.println("Eres mayor de edad y puedes beber alcohol");
}
```

En este ejemplo, si la variable edad es menor que 18, se imprimirá "Eres menor de edad"; si es mayor o igual a 18 pero menor que 21, se imprimirá "Eres mayor de edad pero aún no puedes beber alcohol"; de lo contrario, se imprimirá "Eres mayor de edad y puedes beber alcohol".

Condicional switch

El condicional switch en Java es una estructura que permite evaluar condiciones encadenadas o anidadas. Todo lo que se puede hacer con un switch se puede hacer también con un if...else. Por tanto, el uso de switch no es estrictamente necesario a la hora de programar. Sin embargo, en determinados escenarios resultará más sencillo utilizar un switch que un if...else por claridad y simplicidad en la sintaxis.

Sintaxis de un Condicional switch

```
switch (expresión) {  
    case valor1:  
        // Si la expresión es igual a valor1  
        break;  
    case valor2:  
        // Si la expresión es igual a valor2  
        break;  
    // Puedes tener tantos 'case' como necesites  
    default:  
        // Si la expresión no coincide con ningún 'case'  
}  
}
```

Condicional switch

```
ej06ClaseEjemplo.java U
src > CursoDeJava

int opcion = 2;
String mensaje;

switch (opcion) {
    case 1:
        mensaje = "Opción 1 seleccionada";
        break;
    case 2:
        mensaje = "Opción 2 seleccionada";
        break;
    case 3:
        mensaje = "Opción 3 seleccionada";
        break;
    default:
        mensaje = "Opción no reconocida";
}

System.out.println(mensaje);
```

En este caso, dependiendo del valor de la variable `opcion`, se asignará un mensaje correspondiente. Si `opcion` es igual a 2, se imprimirá "Opción 2 seleccionada". Si `opcion` no coincide con ninguno de los casos especificados (1, 2, o 3), se ejecutará el caso default y se imprimirá "Opción no reconocida".

A tener en cuenta con el condicional switch

1. El valor a evaluar solo puede ser un char, byte, short, int, String o un enum.
2. En los “case” no se permiten operadores relacionales. Solo se puede evaluar igualdad.
3. La instrucción break es opcional y se utilizará en función de cómo se quiera que funcione el switch. En caso de no utilizarse, el flujo de ejecución entra en “fall through” lo que implica que en caso de encontrar un case que cumpla la condición, el flujo de ejecución ejecutaría ese case y los que pudiera haber a continuación.

Operador ternario

El operador ternario es común a muchos lenguajes de programación, entre ellos Java. Sin embargo, no es muy utilizado a pesar de su sencilla sintaxis.

Es un operador útil cuando las condiciones a evaluar son sencillas convirtiendo el código en estos casos en sencillo y elegante.

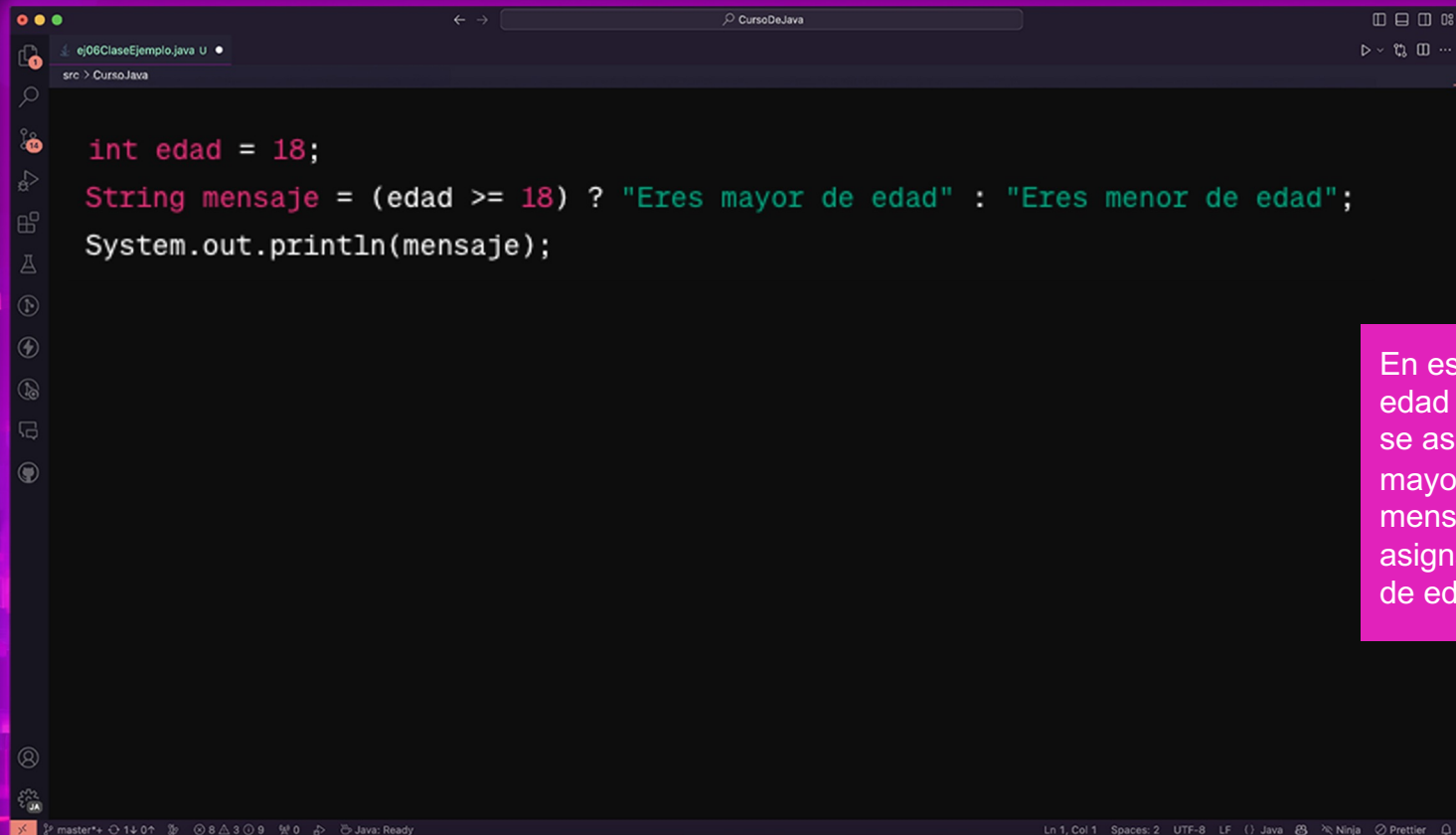
Sintaxis del operador ternario

El operador ternario recibe tres argumentos:

- La condición es una expresión booleana que se evalúa.
- Si la condición es verdadera, se asigna el valor de valorSiVerdadero a la variable.
- Si la condición es falsa, se asigna el valor de valorSiFalso.

variable = (condición) ? valorSiVerdadero : valorSiFalso;

Sintaxis del operador ternario

A screenshot of a code editor window titled 'CursoDeJava'. The editor shows three lines of Java code: 'int edad = 18;', 'String mensaje = (edad >= 18) ? "Eres mayor de edad" : "Eres menor de edad";', and 'System.out.println(mensaje);'. The code is color-coded: keywords in blue, strings in red, and identifiers in green. The IDE interface includes a sidebar with icons for Explorer, Search, Run and Debug, Source Control, Extensions, Testing, Remote Explorer, and Settings. The status bar at the bottom shows 'master*+ 14 0', '8 3 0 0', 'Java: Ready', 'Ln 1, Col 1', 'Spaces: 2', 'UTF-8', 'LF', 'Java', 'Ninja', and 'Prettier'.

En este ejemplo, si la variable edad es mayor o igual a 18, se asignará el valor "Eres mayor de edad" a la variable mensaje; de lo contrario, se asignará el valor "Eres menor de edad".