UNIDAD 2

Estructuras condicionales

Contenido

Sentencias de control de flujo
Sentencias condicionales
Operadores de relación
Operadores lógicos o booleanos
Tabla de verdad del operador &&
Tabla de verdad del operador ||
Tabla de verdad del operador !
Sentencia IF
Sentencia IF-ELSE
Operador ternario
Anidación de condicionales
SWITCH

Sentencias de control de flujo

Las sentencias de control de flujo se emplean en los programas para ejecutar sentencias condicionalmente, repetir un conjunto de sentencias o, en general, cambiar el flujo secuencial de ejecución.

Sin ellas, el programa siempre se ejecutará en el mismo orden.

Sentencias condicionales

Las estructuras condicionales nos permiten ejecutar una serie de instrucciones si cumple una determinada condición que nosotros le indiquemos.

Es importante recordar que la condición debe dar un resultado booleano, por lo que lo más normal es usar operadores relacionales y condicionales.

Para poder manipular las estructuras condicionales, debes tener claro los operadores de relación y los operadores lógicos o booleanos.

Operadores de relación

Realizan comparaciones entre datos compatibles de tipos primitivos (numéricos, carácter y booleanos) teniendo siempre un resultado booleano.

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

Operadores lógicos o booleanos

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

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

Tablas de verdad del operador &&

a	b	a && b
falso	falso	falso
cierto	falso	falso
falso	cierto	falso
cierto	cierto	cierto

Tablas de verdad del operador ||

a	b	a || b
falso	falso	falso
cierto	falso	cierto
falso	cierto	cierto
cierto	cierto	cierto

Tablas de verdad del operador !

a	! a
falso	cierto
cierto	falso

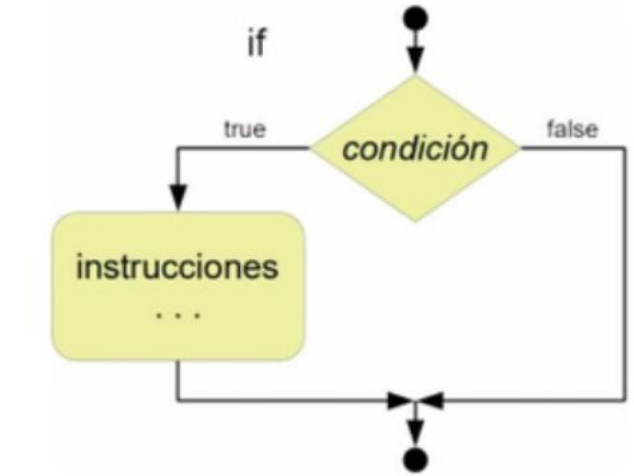
Sentencia IF

Se evalúa una condición y en caso de que se cumpla se ejecuta el contenido entre las llaves {} o en caso de que se omitan se ejecuta el código hasta el primer «;» por lo tanto si no se usan los {} la condición aplica solo a la siguiente instrucción al if.

```
if (condición) {
        sentencias
        ...
}

-------------------------------------------

if (condición) sentencia;
```



Ejemplos

```
// Si la temperatura es mayor que 25
if (temperatura > 25) {
    System.out.println("A la playa!!!");
}

// Si lloviendo es verdadero
if (lloviendo) {
      System.out.println("Coger paraguas");
}

// Si está nevando o lloviendo
if (nevando || lloviendo) {
      System.out.println("Quedarse en casa");
}
```

```
// Si nevado y la temperatura está entre 15 y 30
if (nevado && (temperatura >= 15 &&
temperatura <= 30)) {
      System.out.println("Vamos a esquiar");
}
```

```
/* Si es mayor de edad se le da un premio de
dinero y si es menor, un regalo. */
if (edad>=18) System.out.println(“100€”);
if (edad<18)  System.out.println(“Patinete”);
/* Se evalúa dos veces si es mayor de edad,
tanto si es mayor o igual de 18 como si es menor
de 18 */
```

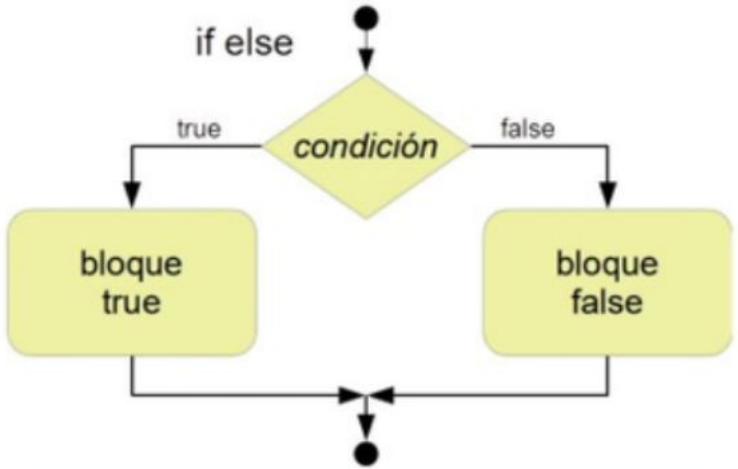
Sentencia IF-ELSE

Se usa else par indicar otras sentencias que se ejecutarán **cuando no se cumpla la condición**.

```
if (condición) {
        sentencia 1
        ...
} else {
        sentencia 2

        ...
}
```

```
if (condición) sentencia1;
else sentencia2;
```



De esta forma se evita evaluar dos veces la condición.

Ejemplos

```
// Si es mayor de edad se le da un premio de dinero y si es menor, un regalo.

if (edad>=18)
        System.out.println(“100€”);
else
        System.out.println(“Patinete”);

// Simple validación de usuario y contraseña para acceder a elementos restringidos.

if ( login.equals(”admin”) && password.equals(“1234”) ) {
        System.out.println(“Usuario válido”);
        System.out.println(“Acceso a área restringida”);
} else {
        System.out.println(“Usuario o contraseña incorrecta”);
}
```

Operador ternario

Es habitual que haya que asignar a una variable un valor u otro dependiendo de una condición.

```
Scanner sc = new Scanner(System.in);
System.out.print("¿Qué edad tienes? ");
int edad = sc.nextInt();

String regalo;
if (edad<18)
    regalo="Patinete";
else
    regalo="100€";

System.out.println("Tu regalo es " + regalo);
```

Esto se puede simplificar usando el operador ternario que es de la forma:

variable = condición ? valor1 : valor2

El ejemplo anterior quedaría:

```
Scanner sc = new Scanner(System.in);
System.out.print("¿Qué edad tienes? ");
int edad = sc.nextInt();
String regalo = edad<18 ? "Patinete" : "100€";
System.out.println("Tu regalo es " + regalo);
```

Ejemplos

// Calcular el máximo de dos números introducidos por teclado

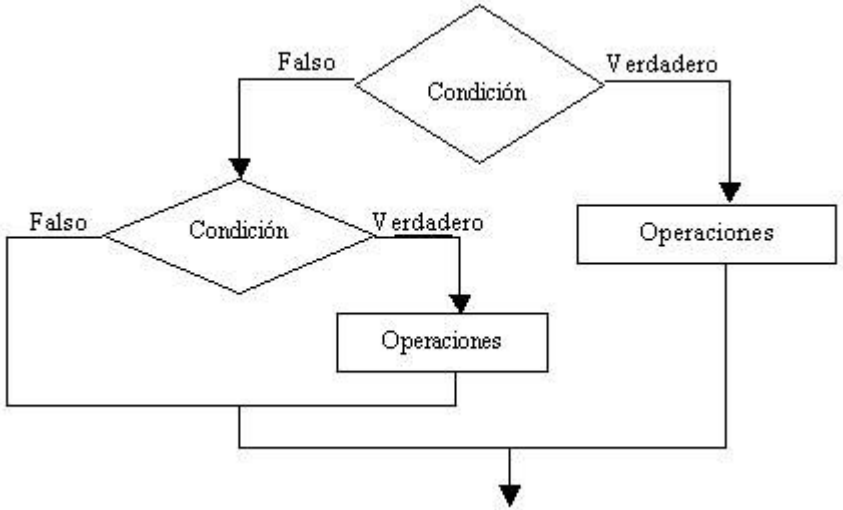
```
Scanner sc = new Scanner(System.in);
int a = sc.nextInt();
int b = sc.nextInt();
int maximo = a > b ? a : b;
System.out.println("El máximo es: " + maximo);
```

Anidación de condicionales

Dentro de un **if** tenemos un bloque de sentencias que se ejecutan si se cumple la condición. También podemos tener un **else** que indica otro bloque de sentencias si no se cumple la condición.

Pues dentro de estos bloques de sentencias podemos insertar también sentencias condicionales. Tanto en el **if** como en el **else**.

Esto se llama **anidación de condicionales** o **IF anidados.**



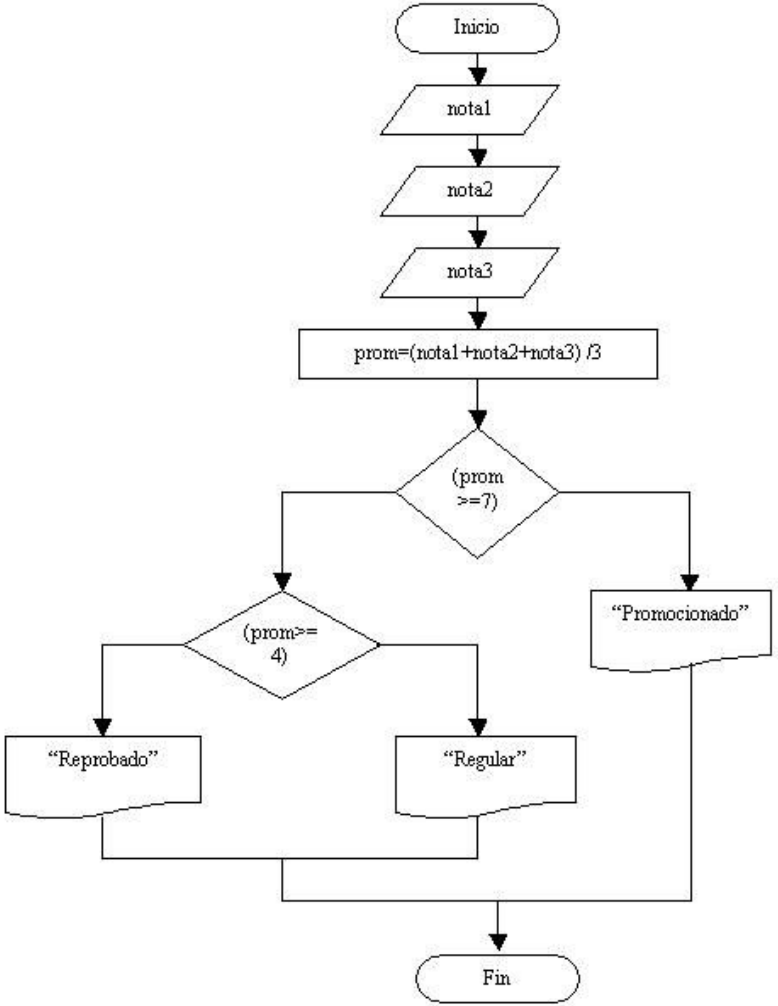
Ejemplo IF anidado

Confeccionar un programa que pida por teclado tres notas de un alumno, calcule el promedio e imprima alguno de estos mensajes:

Si el promedio es >=7 mostrar "Promocionado".

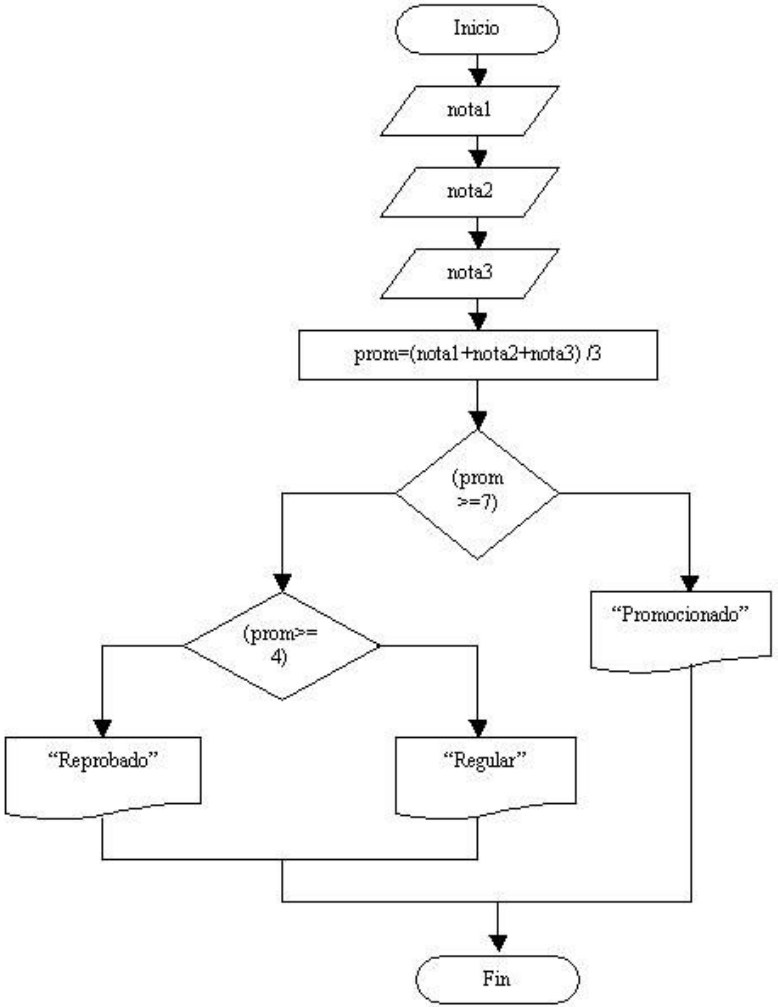
Si el promedio es >=4 y <7 mostrar "Regular".

Si el promedio es <4 mostrar "Reprobado".



Ejemplo IF anidado

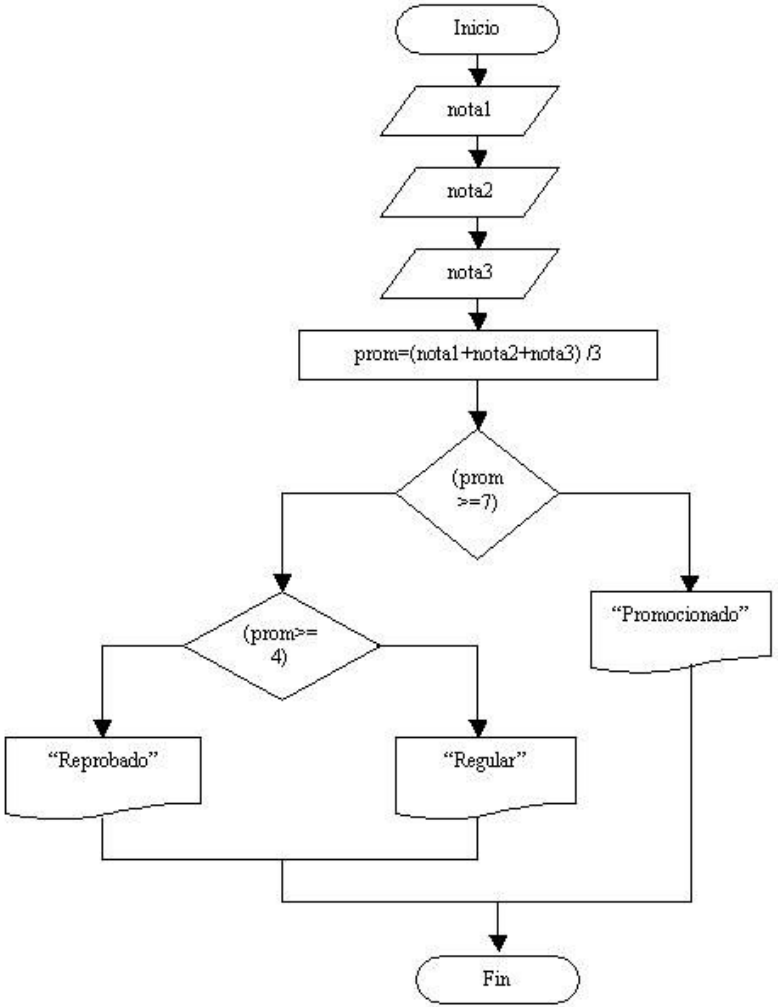
```
...
if (nota>=7) {
    System.out.println("Promocionado");
} else {
    if (nota>=4) {
        System.out.println("Regular");
    } else {
        System.out.println("Reprobado");
    }
}
```



Ejemplo IF anidado

Como cuando usamos una única sentencia, no es necesario usar las llaves, podemos aprovechar esto y cuando el if anidado está en el else, se puede poner de forma abreviada.

```
...
if (nota>=7) {
    System.out.println("Promocionado");
} else if (nota>=4) {
    System.out.println("Regular");
} else {
    System.out.println("Reprobado");
}
```



Ejemplo IF anidado

```
if (a - 2 == 1) {
   System.out.println("Hola ");
} else {
   if (a - 2 == 5) {
      System.out.println("Me ");
   } else {
      if (a - 2 == 8) {
         System.out.println("Alegro ");
      } else {
         if (a - 2 == 9) {
            System.out.println("De ");
         } else {
            if (a - 2 == 11) {
               System.out.println("Conocerte.");
            } else {
               System.out.println("Sin coincidencia");
            }
         }
      }
   }
}
```

SWITCH

La instrucción switch es una forma de expresión de un anidamiento múltiple de instrucciones if ... else.

Su uso no puede considerarse, por tanto, estrictamente necesario, puesto que siempre podrá ser sustituida por el uso de if. No obstante, a veces nos resultará útil al introducir mayor claridad en el código.

Dentro de cada case, se ejecutarán todas las sentencias que haya posterior a él hasta que se encuentre un **break**.

```
switch (expresión) {
   case valor1:
      conjunto instrucción 1
   case valor2:
      conjunto instrucción 2
   ...

   case valorN:
      conjunto instrucción N
   default:
      conjunto instrucción default
}
```

SWITCH

A TENER EN CUENTA

En los cases, utiliza siempre un valor constante (un número, o un texto entre comillas). No utilices otra variable.

Nunca utilices dos cases con el mismo valor.

Aunque la opción default suela aparecer la última en un switch, en realidad puede aparecer en cualquier posición.

Si no está en última posición, la opción default necesita un **break**.

Si está en la última posición, pasaría igual que con cualquier case: como está al final, la ejecución no "vuelve atrás" a ejecutar el código de los otros cases, sino que abandona el switch y sigue adelante. Es por ello que no es necesario especificar un break.

Ejemplo de SWITCH

No hay break, ¿Qué pasa aquí?
¿Qué sale por pantalla?

```
a = 10;
switch (a-2) {
   case 1:
      System.out.print("Hola ");
   case 5:
      System.out.print("Me ");
   case 8:
      System.out.print("Alegro ");
   case 9:
      System.out.print("De ");
   case 11:
      System.out.print("Conocerte. ");
   default:
      System.out.print("Sin coincidencia");
}
```

Ejemplo de SWITCH

Se ejecuta el bloque de instrucciones asociado al segundo case, pero break impide que se ejecuten los siguientes.

```
a = 1;
switch (a*2) {
   case 1:
      System.out.println("Hola");
      break;
   case 2:
      System.out.println("Paco");
   case 3:
      System.out.println("Adiós");
      break;
   default:
      System.out.println("Sin coincidencia");
}
```

Ejemplo de SWITCH

En las últimas versiones de JAVA se ha añadido una nueva forma de utilizar la instrucción SWITCH en la que no es necesario usar break. Si el valor de la expresión coincide con algún case, solamente se ejecutará el bloque de instrucciones asociado a dicho case.

```
switch (nota) {
   case 0,1,2,3,4 -> { //bloque formado por dos instrucciones: entre llaves
      System.out.println("Suspenso.");
      System.out.println("Ánimo...");
   }
   case 5 -> //bloque de una única instrucción: podemos obviar las llaves
      System.out.println("Suficiente.");
   case 6 ->
      System.out.println("Bien.");
   case 7, 8 ->
      System.out.println("Notable");
   case 9, 10 -> {
      System.out.println("Sobresaliente.");
      System.out.println("Enhorabuena");
   }
   default ->
      System.out.println("Nota incorrecta");
}
```