

4. Sentencia condicional (if y switch)

Una sentencia condicional permite al programa bifurcar el flujo de ejecución de instrucciones dependiendo del valor de una expresión.

4.1 Sentencia if

La sentencia `if` permite la ejecución de una serie de instrucciones en función del resultado de una expresión lógica. El resultado de evaluar una expresión lógica es siempre verdadero (`true`) o falso (`false`). Es muy simple, en lenguaje natural sería algo como "si esta condición es verdadera entonces haz esto, sino haz esto otro".

El formato de la sentencia `if` es el siguiente:

```
if (condición) {  
  
    instrucciones a ejecutar si la condición es verdadera  
  
} else {  
  
    instrucciones a ejecutar si la condición es falsa  
  
}
```

A continuación se muestra un ejemplo del uso de la sentencia `if`.

```
/**  
 * Sentencia if  
 *  
 * @author Luis J. Sánchez  
 */  
public class ComparacionCadena {  
    public static void main(String[] args) {  
        String miFruta = "naranja";  
  
        if ("naranja".equals(miFruta)) {  
            System.out.println("iguales");  
        } else {  
            System.out.println("distintas");  
        }  
    }  
}
```

```

    }
}

```

¿Podrías adivinar cuál es el resultado del programa? Efectivamente, se muestra la palabra “iguales” ya que la comparación `"naranja".equals(miFruta)` devuelve `true` y, por tanto, se ejecuta el primer bloque de código, o sea, `System.out.println("iguales");`.

Veamos ahora un ejemplo algo más interesante. Se trata de una especie de juego al estilo de “La ruleta de la fortuna” en el que el usuario debe responder a la pregunta que le hace el ordenador.

```

/**
 * Sentencia if
 *
 * @author Luis J. Sánchez
 */
public class SentenciaIf01 {
    public static void main(String[] args) {
        System.out.print("¿Cuál es la capital de Kiribati? ");
        String respuesta = System.console().readLine();

        if (respuesta.equals("Tarawa")) {
            System.out.println("¡La respuesta es correcta!");
        } else {
            System.out.println("Lo siento, la respuesta es incorrecta.");
        }
    }
}

```

En el programa se le pregunta al usuario cuál es la capital de Kiriwati. La respuesta introducida por el usuario se almacena en la variable `respuesta`. A continuación viene la sentencia condicional `if`.

```

if (respuesta.equals("Tarawa"))

```

Llegado a este punto, el programa evalúa la expresión `respuesta.equals("Tarawa")`. Observa que para comparar dos cadenas de caracteres se utiliza `equals()`. Imaginemos que el usuario ha introducido por teclado `Madrid`; entonces la expresión `"Madrid".equals("Tarawa")` daría como resultado `false` (falso).

Si la expresión hubiera dado como resultado `true` (verdadero), se ejecutaría la línea

```

System.out.println("¡La respuesta es correcta!");

```

pero no es el caso, el resultado de la expresión ha sido `false` (falso), todo el mundo sabe que la capital de Kiriwati no es Madrid, por tanto se ejecutaría la línea

```
System.out.println("Lo siento, la respuesta es incorrecta.");
```

Vamos a ver otro ejemplo, esta vez con números. El usuario introducirá un número por teclado y el programa dirá si se trata de un número positivo o negativo.

```
/**
 * Sentencia if
 *
 * @author Luis J. Sánchez
 */
public class SentenciaIf02 {
    public static void main(String[] args) {
        System.out.print("Por favor, introduce un número entero: ");
        String linea = System.console().readLine();
        int x = Integer.parseInt( linea );

        if (x < 0) {
            System.out.println("El número introducido es negativo.");
        } else {
            System.out.println("El número introducido es positivo.");
        }
    }
}
```

El siguiente bloque de código

```
if (x < 0)
    System.out.println("El número introducido es negativo.");
else
    System.out.println("El número introducido es positivo.");
```

compilaría y funcionaría sin problemas - fíjate que hemos quitado las llaves - ya que antes y después del `else` hay una sola sentencia y en estos casos no es obligatorio poner llaves. Sin embargo, nosotros siempre usaremos llaves, es una exigencia del estándar de Google al que nos ceñimos en este manual.



Llaves egipcias (egyptian brackets)

Fíjate en la manera de colocar las llaves dentro del código de un programa en Java. La llave de apertura de bloque se coloca justo al final de la línea y la llave de cierre va justo al principio de la línea. Se llaman **llaves egipcias**¹ por la similitud entre las llaves y las manos de los egipcios que aparecen en los papiros.

¹<http://blog.codinghorror.com/new-programming-jargon/>

4.2 Operadores de comparación

En el ejemplo anterior, usamos el operador `<` en la comparación `if (x < 0)` para saber si la variable `x` es menor que cero. Hay más operadores de comparación, en la tabla 4.2.1 se muestran todos.

Tabla 4.2.1. Operadores de comparación.

OPERADOR	NOMBRE	EJEMPLO	DESCRIPCIÓN
<code>==</code>	igual	<code>a == b</code>	a es igual a b
<code>!=</code>	distinto	<code>a != b</code>	a es distinto de b
<code><</code>	menor que	<code>a < b</code>	a es menor que b
<code>></code>	mayor que	<code>a > b</code>	a es mayor que b
<code><=</code>	menor o igual que	<code>a <= b</code>	a es menor o igual que b
<code>>=</code>	mayor o igual que	<code>a >= b</code>	a es mayor o igual que b

El siguiente ejemplo muestra el uso de uno de estos operadores, concretamente de `>=` (mayor o igual). El usuario introduce una nota; si esta nota es **mayor o igual a 5** se le mostrará un mensaje diciendo que ha aprobado y en caso de que no se cumpla la condición se mostrará un mensaje diciendo que está suspenso.

```
/**
 * Sentencia if
 *
 * @author Luis J. Sánchez
 */
public class SentenciaIf03 {
    public static void main(String[] args) {
        System.out.print("¿Qué nota has sacado en el último examen? ");
        String line = System.console().readLine();
        double nota = Double.parseDouble( line );

        if (nota >= 5) {
            System.out.println("¡Enhorabuena!, ihas aprobado!");
        } else {
            System.out.println("Lo siento, has suspendido.");
        }
    }
}
```

4.3 Operadores lógicos

Los operadores de comparación se pueden combinar con los operadores lógicos. Por ejemplo, si queremos saber si la variable `a` es mayor que `b` y además es menor que `c`, escribiríamos `if ((a > b) && (a < c))`. En la siguiente tabla se muestran los operadores lógicos de Java:

Tabla 4.3.1. Operadores lógicos.

OPERADOR	NOMBRE	EJEMPLO	DEVUELVE VERDADERO CUANDO...
<code>&&</code>	y	<code>(7 > 2) && (2 < 4)</code>	las dos condiciones son verdaderas
<code> </code>	o	<code>(7 > 2) (2 < 4)</code>	al menos una de las condiciones es verdadera
<code>!</code>	no	<code>!(7 > 2)</code>	la condición es falsa

Vamos a ver cómo funcionan los operadores lógicos con un ejemplo. Mediante `if ((n < 1) || (n > 100))` se pueden detectar los números que no están en el rango de 1 a 100; literalmente sería “si `n` es menor que 1 o `n` es mayor que 100”.

```
/**
 * Operadores lógicos
 *
 * @author Luis J. Sánchez
 */
public class OperadoresLogicos01 {
    public static void main(String[] args) {
        System.out.println("Adivina el número que estoy pensando.");
        System.out.print("Introduce un número entre el 1 y el 100: ");
        String linea = System.console().readLine();
        int n = Integer.parseInt( linea );

        if ((n < 1) || (n > 100)) {
            System.out.println("El número introducido debe estar en el intervalo 1 - 100.");
            System.out.print("Tienes otra oportunidad, introduce un número: ");
            linea = System.console().readLine();
            n = Integer.parseInt( linea );
        }

        if (n == 24) {
            System.out.println("¡Enhorabuena!, ¡has acertado!");
        } else {
            System.out.println("Lo siento, ese no es el número que estoy pensando.");
        }
    }
}
```

```

    }
}
}

```

El comportamiento de los operadores lógicos se muestra en la siguiente tabla de verdad (tabla 4.3.2) donde “V” significa “verdadero” y “F” significa “falso”.

Tabla 4.3.2. Tabla de verdad de los operadores lógicos.

A	B	A && B	A B	!A	!B
V	V	V	V	F	F
V	F	F	V	F	V
F	V	F	V	V	F
F	F	F	F	V	V

En el siguiente programa puedes ver el uso de operadores lógicos combinado con operadores relacionales (operadores de comparación). Intenta adivinar cuál será el resultado mirando el código.

```

/**
 * Operadores lógicos y relacionales
 *
 * @author Luis J. Sánchez
 */
public class OperadoresLogicos02 {
    public static void main(String[] args) {
        boolean a = true;
        boolean b = false;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("a && b = " + (a && b));
        System.out.println("a || b = " + (a || b));
        System.out.println("!a = " + !a);
        System.out.println("a || (6 > 10) = " + (a || (6 > 10)));
        System.out.println("((4 <= 4) || false) && (!a) = " + (((4 <= 4) || false) && (!a)));
    }
}

```

4.4 Sentencia switch (selección múltiple)

A veces es necesario comparar el valor de una variable con una serie de valores concretos. La selección múltiple es muy parecida (aunque no es exactamente igual) a una secuencia de varias sentencias if.

El formato de switch es el que se muestra a continuación. En lenguaje natural sería algo así como “Si variable vale valor1 entonces entra por case valor1:, Si variable vale valor2 entonces entra por case valor2:,... Si variable no vale ninguno de los valores que hay en los distintos case entonces entra por default:.

```
switch(variable) {  
    case valor1:  
        sentencias  
        break;  
  
    case valor2:  
        sentencias  
        break;  
    .  
    .  
    .  
  
    default:  
        sentencias  
}
```

A continuación tienes un ejemplo completo en Java. Se pide al usuario un número de mes y el programa da el nombre del mes que corresponde a ese número.

```
/**  
 * Sentencia múltiple (switch)  
 *  
 * @author Luis José Sánchez  
 */  
public class SentenciaSwitch {  
    public static void main(String[] args) {  
  
        System.out.print("Por favor, introduzca un numero de mes: ");  
        int mes = Integer.parseInt(System.console().readLine());  
  
        String nombreDelMes;  
  
        switch (mes) {  
            case 1:
```

```
        nombreDelMes = "enero";
        break;
    case 2:
        nombreDelMes = "febrero";
        break;
    case 3:
        nombreDelMes = "marzo";
        break;
    case 4:
        nombreDelMes = "abril";
        break;
    case 5:
        nombreDelMes = "mayo";
        break;
    case 6:
        nombreDelMes = "junio";
        break;
    case 7:
        nombreDelMes = "julio";
        break;
    case 8:
        nombreDelMes = "agosto";
        break;
    case 9:
        nombreDelMes = "septiembre";
        break;
    case 10:
        nombreDelMes = "octubre";
        break;
    case 11:
        nombreDelMes = "noviembre";
        break;
    case 12:
        nombreDelMes = "diciembre";
        break;
    default:
        nombreDelMes = "no existe ese mes";
}

System.out.println("Mes " + mes + ": " + nombreDelMes);
}
}
```

Observa que es necesario introducir un `break` después de la asignación de la variable `nombreDelMes`. En caso de no encontrarse el `break`, el programa continúa la ejecución en la línea siguiente.

El bloque que corresponde al `default` se ejecuta cuando la variable no coincide con ninguno de los valores de los `case`. Escribiremos siempre el `default` al final de la sentencia `switch` aunque no sea necesario.

La sentencia `switch` se utiliza con frecuencia para crear menús.

```
/**
 * Ejemplo de un menú con switch
 *
 * @author Luis José Sánchez
 */
public class MenuConSwitch {
    public static void main(String[] args) {

        System.out.println(" CÁLCULO DE ÁREAS");
        System.out.println(" -----");
        System.out.println(" 1. Cuadrado");
        System.out.println(" 2. Rectángulo");
        System.out.println(" 3. Triángulo");
        System.out.print("\n Elija una opción (1-3): ");

        int opcion = Integer.parseInt(System.console().readLine());

        double lado;
        double base;
        double altura;

        switch (opcion) {
            case 1:
                System.out.print("\nIntroduzca el lado del cuadrado en cm: ");
                lado = Double.parseDouble(System.console().readLine());
                System.out.println("\nEl área del cuadrado es " + (lado * lado) + " cm2");
                break;

            case 2:
                System.out.print("\nIntroduzca la base del rectángulo en cm: ");
                base = Double.parseDouble(System.console().readLine());
                System.out.print("Introduzca la altura del rectángulo en cm: ");
                altura = Double.parseDouble(System.console().readLine());
                System.out.println("El área del rectángulo es " + (base * altura) + " cm2");
                break;

            case 3:
                System.out.print("\nIntroduzca la base del triángulo en cm: ");
                base = Double.parseDouble(System.console().readLine());
                System.out.print("Introduzca la altura del triángulo en cm: ");
                altura = Double.parseDouble(System.console().readLine());
```

```
        System.out.println("El área del triángulo es " + ((base * altura) / 2) + " cm2");
        break;

    default:
        System.out.print("\nLo siento, la opción elegida no es correcta.");
    }
}
}
```