

4.B. Estructuras de selección.

✓ Hecho

1. Estructuras de selección.

1.1. Estructura if / if-else.

La estructura **if** es una estructura de selección o estructura condicional, en la que se evalúa una expresión lógica o sentencia de decisión y en función del resultado, se ejecuta una sentencia o un bloque de éstas.

La estructura **if** puede presentarse de las siguientes formas:

Estructura if e if-else.				
Estructura if simple.		Estructura if de doble alternativa.		
Sintaxis:	Sintaxis: if (expresión-lógica) { sentencia1; ...; sentenciaN; }	Sintaxis: if (expresión-lógica) sentencia1; ...; sentenciaN; }	Sintaxis:	
			if (expresión-lógica)	
			{	
			sentencia1;	
			...;	
			if (expresión-lógica)	sentenciaN;
			sentencia1;	}
	if (expresión-lógica)	sentencia2;		
			else	else
	sentencia1;	...;	sentencia2;	{
	sentenciaN;		sentencia1;	
			...;	
			sentenciaN;	
	}		}	
Funcionamiento:		Funcionamiento:		
Si la evaluación de la expresión-lógica ofrece un resultado verdadero, se ejecuta la sentencia1 o bien el bloque de sentencias asociado. Si el resultado de dicha evaluación es falso, no se ejecutará ninguna instrucción asociada a la estructura condicional.		Si la evaluación de la expresión-lógica ofrece un resultado verdadero, se ejecutará la primera sentencia o el primer bloque de sentencias. Si, por el contrario, la evaluación de la expresión-lógica ofrece un resultado falso, no se ejecutará la primera sentencia o el primer bloque y sí se ejecutará la segunda sentencia o el segundo bloque.		

Haciendo una interpretación cercana al pseudocódigo tendríamos que si se cumple la condición (expresión lógica), se ejecutará un conjunto de instrucciones y si no se cumple, se ejecutará otro conjunto de instrucciones.

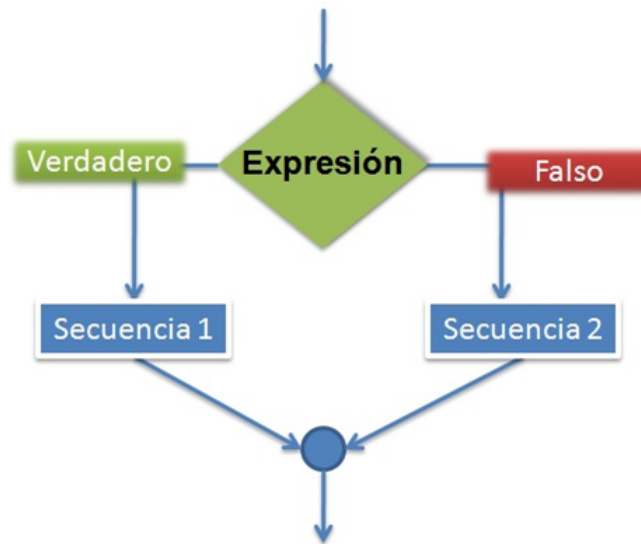


Imagen extraída de curso Programación del MECD.

Hay que tener en cuenta que la cláusula `else` de la sentencia `if` **no es obligatoria**. En algunos casos no necesitaremos utilizarla, pero sí se recomienda cuando es necesario llevar a cabo alguna acción en el caso de que la expresión lógica no se cumpla.

En aquellos casos en los que no existe cláusula `else`, si la expresión lógica es **falsa**, simplemente se continuarán ejecutando las siguientes sentencias que aparezcan bajo la estructura condicional `if`.

Los condicionales `if` e `if-else` pueden anidarse, de tal forma que dentro de un bloque de sentencias puede incluirse otro `if` o `if-else`. El nivel de anidamiento queda a criterio del programador, pero si éste es demasiado profundo podría provocar problemas de eficiencia y legibilidad en el código. En otras ocasiones, un nivel de anidamiento excesivo puede denotar la necesidad de utilización de otras estructuras de selección más adecuadas.

Cuando se utiliza anidamiento de este tipo de estructuras, es necesario poner especial atención en saber a qué `if` está asociada una cláusula `else`. Normalmente, un `else` estará asociado con el `if` inmediatamente superior o más cercano que exista dentro del mismo bloque y que no se encuentre ya asociado a otro `else`.

Debes conocer

Para completar la información que debes saber sobre las estructuras `if` e `if-else`, estudia el siguiente código. En él podrás analizar el código de un programa que realiza el cálculo de la nota de un examen de tipo test. Además de calcular el valor de la nota, se ofrece como salida la calificación no numérica de dicho examen. Para obtenerla, se combinarán las diferentes estructuras condicionales aprendidas hasta ahora.

Presta especial atención a los comentarios incorporados en el código fuente, así como a la forma de combinar las estructuras condicionales y a las expresiones lógicas utilizadas en ellas.

```

package sentencias_condicionales;

/**
 *
 * Ejemplos de utilización de diferentes estructuras
 * condicionales simples, completas y anidamiento de éstas.
 */

public class Sentencias_condicionales {

    /**Vamos a realizar el cálculo de la nota de un examen
     * de tipo test. Para ello, tendremos en cuenta el número
     * total de pregunta, los aciertos y los errores. Dos errores
     * anulan una respuesta correcta.
  
```

*

* Finalmente, se muestra por pantalla la nota obtenida, así

* como su calificación no numérica.

*

* La obtención de la calificación no numérica se ha realizado

* combinando varias estructuras condicionales, mostrando expresiones

* lógicas compuestas, así como anidamiento.

*

*/

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

```
    // Declaración e inicialización de variables
```

```
    int num_aciertos = 12;
```

```
    int num_errores = 3;
```

```
    int num_preguntas = 20;
```

```
    float nota = 0;
```

```
    String calificacion="";
```

```
    //Procesamiento de datos
```

```
    nota = ((num_aciertos - (num_errores/2))*10)/num_preguntas;
```

```
    if (nota < 5)
```

```
    {
```

```
        calificacion="INSUFICIENTE";
```

```
    }
```

```
    else
```

```
    {
```

```
        /* Cada expresión lógica de estos if está compuesta por dos
```

```
        * expresiones lógicas combinadas a través del operador Y o AND
```

```
        * que se representa con el símbolo &&. De tal manera, que para
```

```
        * que la expresión lógica se cumpla (sea verdadera) la variable
```

```
        * nota debe satisfacer ambas condiciones simultáneamente
```

```
        */
```

```
        if (nota >= 5 && nota <6)
```

```
            calificacion="SUFICIENTE";
```

```
        if (nota >= 6 && nota <7)
```

```
            calificacion="BIEN";
```

```
        if (nota >= 7 && nota <9)
```

```
            calificacion="NOTABLE";
```

```
        if (nota >= 9 && nota <=10)
```

```
            calificacion="SOBRESALIENTE";
```

```
}  
  
  
//Salida de información  
  
System.out.println ("La nota obtenida es: " + nota);  
  
System.out.println ("y la calificación obtenida es: " + calificacion);  
  
}  
  
  
}
```

Autoevaluación

¿Cuándo se mostrará por pantalla el mensaje incluido en el siguiente fragmento de código?

```
If (numero % 2 == 0);  
  
System.out.print("El número es par /n");
```

- ☐ Nunca.
- ☒ Siempre.
- ☐ Cuando el resto de la división entre 2 del contenido de la variable numero, sea cero.

[◀ 4.A. Introducción a las estructuras de control.](#)