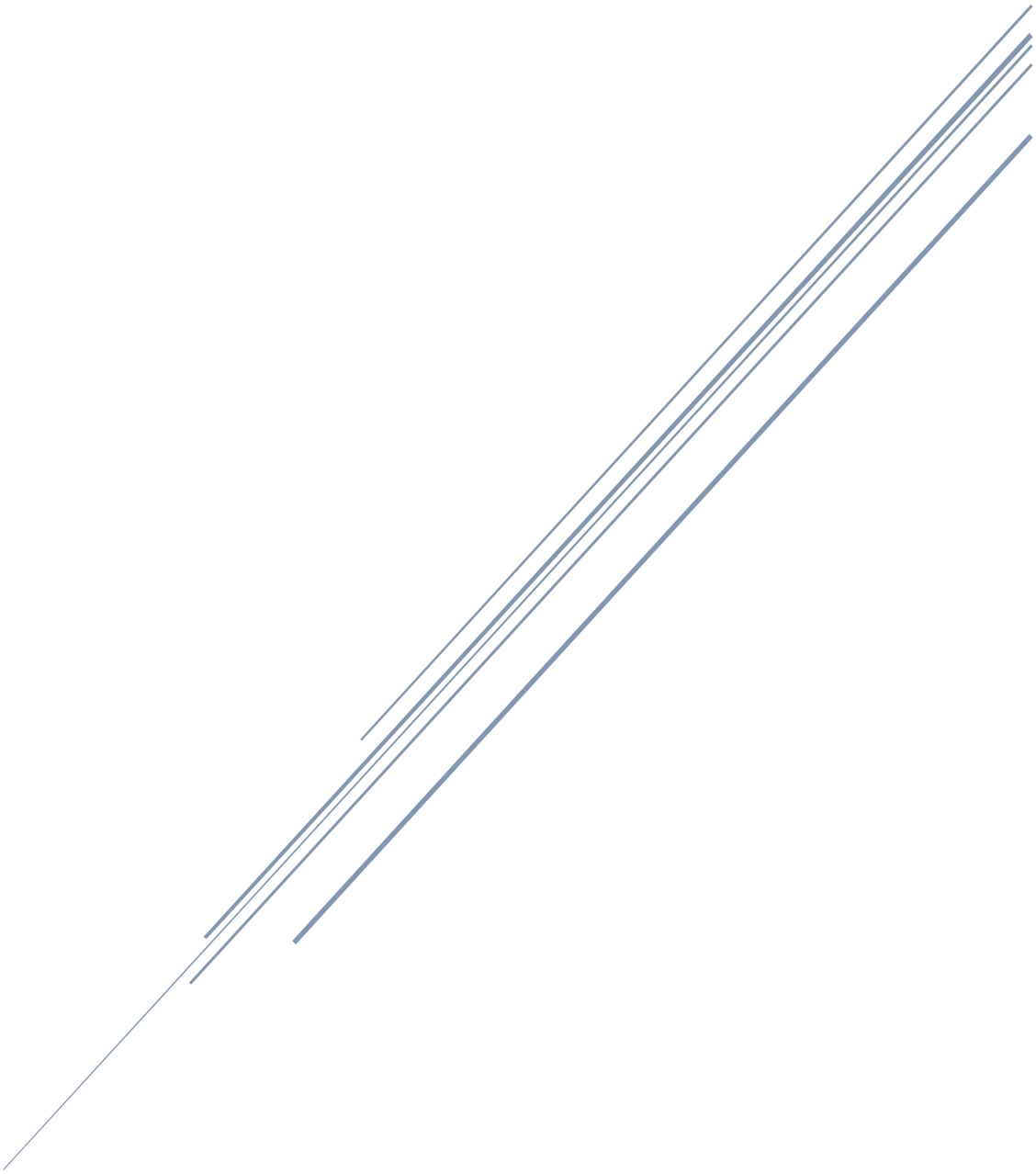
PROGRAMACION

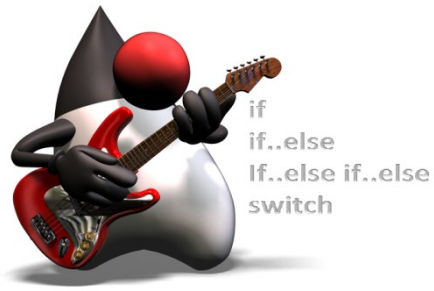
SWITCH



[Escuela]
[Título del curso]

IF.. ELSE IF - CONDICIONAL DE SELECCIÓN EN JAVA.

Otra de las formas posibles que disponemos para implementar las sentencias con if, consiste en unir el else y el if() de la forma siguiente:




```
If (condición) {
        Sentecias 1
} else if (condición) {
        Sentencias 2
} else {
        Sentencias 3
}
```

// Ejemplo if con else if y cláusula final else

```
if (DesplazamientoX == 0 && DesplazamientoY == 1) {
      System.out.println ("Se procede a bajar el personaje 1 posición");
   }
    else if (DesplazamientoX == 1 && DesplazamientoY == 0) {
      System.out.println ("Se procede a mover el personaje 1 posición a la derecha"); }

   else if (DesplazamientoX == -1 && DesplazamientoY == 0) {
      System.out.println ("Se procede a mover el personaje 1 posición a la izquierda");
   }
    else {
      System.out.println ("Los valores no son válidos");
   }
```

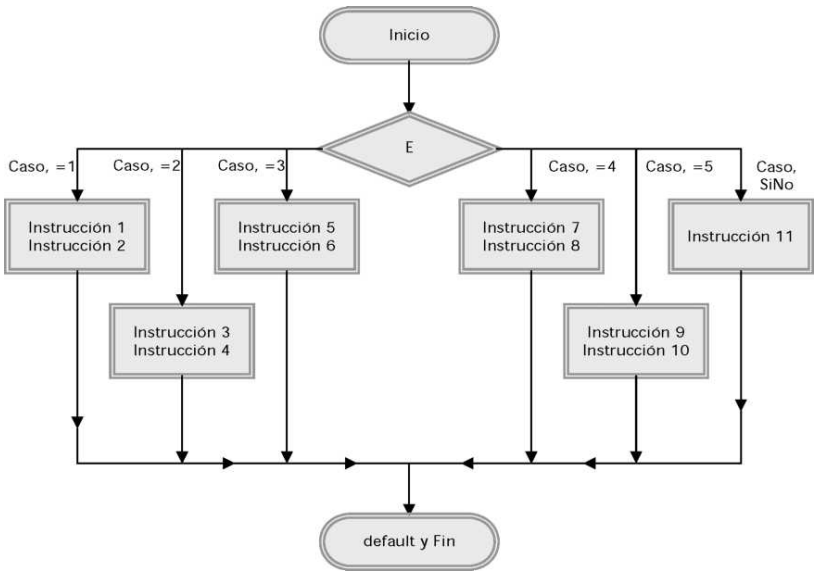
SWITCH - CONDICIONAL DE SELECCIÓN SWITCH EN JAVA.

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.

La sintaxis será:

```
switch (expresión) {
        case valor 1:
        case valor2:
        case valor3:
              instrucciones;
              break;
        case valor4:
              instrucciones;
              break;
        default:
              sentencias;
              break;
}
```

```
switch (expresión) {
        case valor 1:
              instrucciones;
              break;
        case valor2:
              instrucciones;
              break;
        default:
              sentencias;
              break;
}
```



La cláusula *default* es opcional y representa las instrucciones que se ejecutarán en caso de que no se verifique ninguno de los casos evaluados. El último *break* dentro de un *switch* (en *default* si existe esta cláusula, o en el último caso evaluado si no existe *default)* también es opcional, pero lo incluiremos siempre para ser metódicos.

Switch solo se puede utilizar para evaluar ordinales (por ordinal entenderemos en general valores numéricos enteros o datos que se puedan asimilar a valores numéricos enteros). Por tanto no podemos evaluar cadenas (String) usando switch porque el compilador nos devolverá un error de tipo *"found java.lang.String but expected int".* Sí se permite evaluar caracteres y lo que se denominan tipos enumerados, que veremos más adelante.

Switch solo permite evaluar valores concretos de la expresión: no permite evaluar intervalos (pertenencia de la expresión a un intervalo o rango) ni expresiones compuestas. Código de ejemplo:

```
//Ejemplo de método que usa switch
public class dimeSiEdadEsCritica() {
        public static void main(String[] args) {

                //Solicitamos y leemos por teclado la edad

                switch (edad) {
                        case 0:
                          System.out.println ("Acaba de nacer hace poco. No ha cumplido el año"); break;
                        case 18:
                          System.out.println ("Está justo en la mayoría de edad");
                          break;
                         case 65:
                          System.out.println ("Está en la edad de jubilación");
                          break;
                        default:
                          System.out.println ("La edad no es crítica"); break;
                }
```

```
        }
}
```

En algunos casos escribimos varias instrucciones en una línea y en otros una sola instrucción por línea. Ambas posibilidades son válidas. Prueba a escribir, compilar e invocar este método o uno parecido usando *switch.* Para ello crea primero una clase de nombre Persona cuyos atributos sean nombre y edad. Inicializa los atributos a un valor por defecto en el constructor. Crea métodos para definir valor para los atributos (métodos setters) y prueba el método dimeSiEdadEsCritica para comprobar que responde como es de esperar.

EJERCICIO

7. Considera estás desarrollando un programa Java donde necesitas trabajar con objetos de tipo Motor (que representa el motor de una bomba para mover fluidos). Crea un programa TipoMotorBomba.

El programa debe solicitar por pantalla el tipo de motor, dando como valores 0, 1, 2, 3 o 4, y por medio de un condicional switch que devuelva lo siguiente:

a) Si el tipo de motor es 0, mostrar un mensaje por consola indicando "No hay establecido un valor definido para el tipo de bomba".

b) Si el tipo de motor es 1, mostrar un mensaje por consola indicando "La bomba es una bomba de agua".

c) Si el tipo de motor es 2, mostrar un mensaje por consola indicando "La bomba es una bomba de gasolina".

d) Si el tipo de motor es 3, mostrar un mensaje por consola indicando "La bomba es una bomba de hormigón".

e) Si el tipo de motor es 4, mostrar un mensaje por consola indicando "La bomba es una bomba de pasta alimenticia".

f) Si no se cumple ninguno de los valores anteriores mostrar el mensaje "No existe un valor válido para tipo de bomba".

The ? : operator in Java

The value of a variable often depends on whether a particular boolean expression is or is not true and on nothing else. For instance one common operation is setting the value of a variable to the maximum of two quantities. In Java you might write

```
if (a > b) {
  max = a;
}
else {
  max = b;
}
```

Setting a single variable to one of two states based on a single condition is such a common use of if-else that a shortcut has been devised for it, the conditional operator, ?:. Using the conditional operator you can rewrite the above example in a single line like this:

```
max = (a > b) ? a : b;
```

(a > b) ? a : b; is an expression which returns one of two values, a or b. The condition, (a > b), is tested. If it is true the first value, a, is returned. If it is false, the second value, b, is returned. Whichever value is returned is dependent on the conditional test, a > b. The condition can be any expression which returns a boolean value.

Ejercicio 8: Notas (Susp, Aprob, Bien, Not, Sobr)