

# Estructuras selectivas

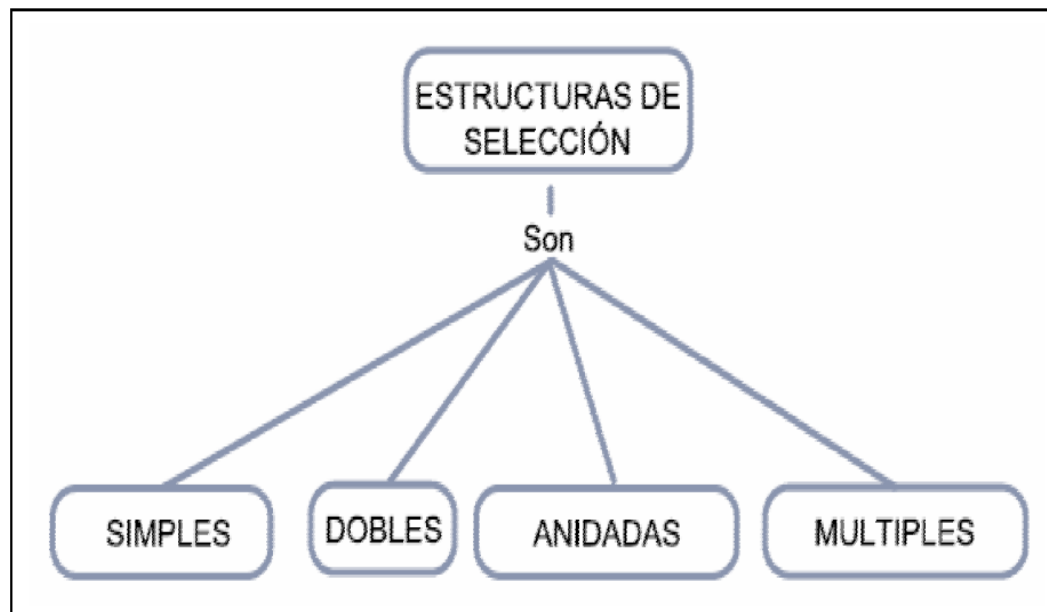
Dr. José Lázaró Martínez Rodríguez

# Estructuras condicionales o de selección

- Diariamente en nuestras vidas cotidianas tomamos decisiones sobre una serie de circunstancias que acontecen, por ejemplo, si queremos ir o no a un determinado lugar, si se dispone o no de dinero para ir de compras, si llueve o no, entre otras.
- Los condicionales permiten determinar si se realiza una determinada acción o no.

# Tipos de E.S.

- Las estructuras de decisión o selección comparan una variable contra otro(s) valor(es), para que de acuerdo al resultado de esta comparación, se siga un curso de Acción dentro del programa.

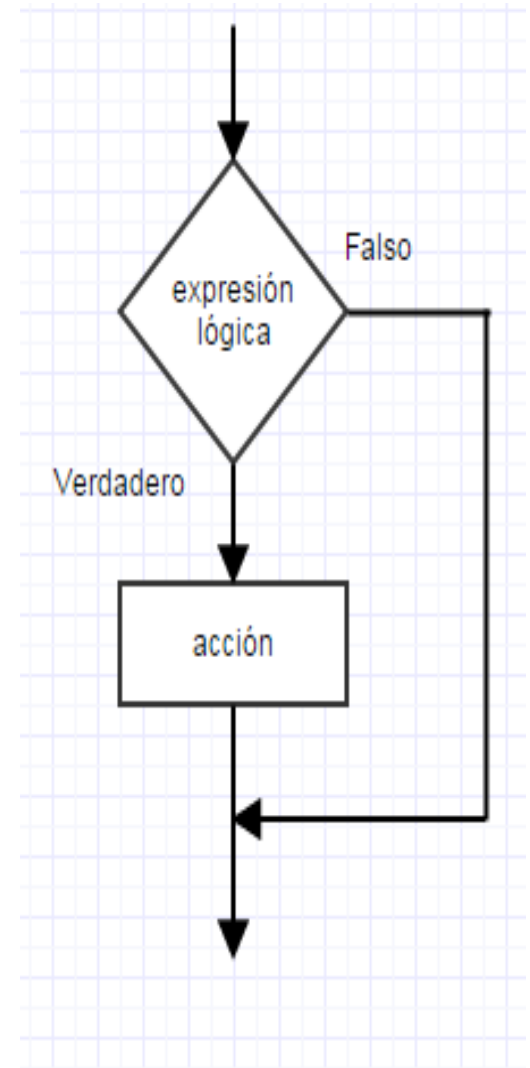


# Estructura condicional simple

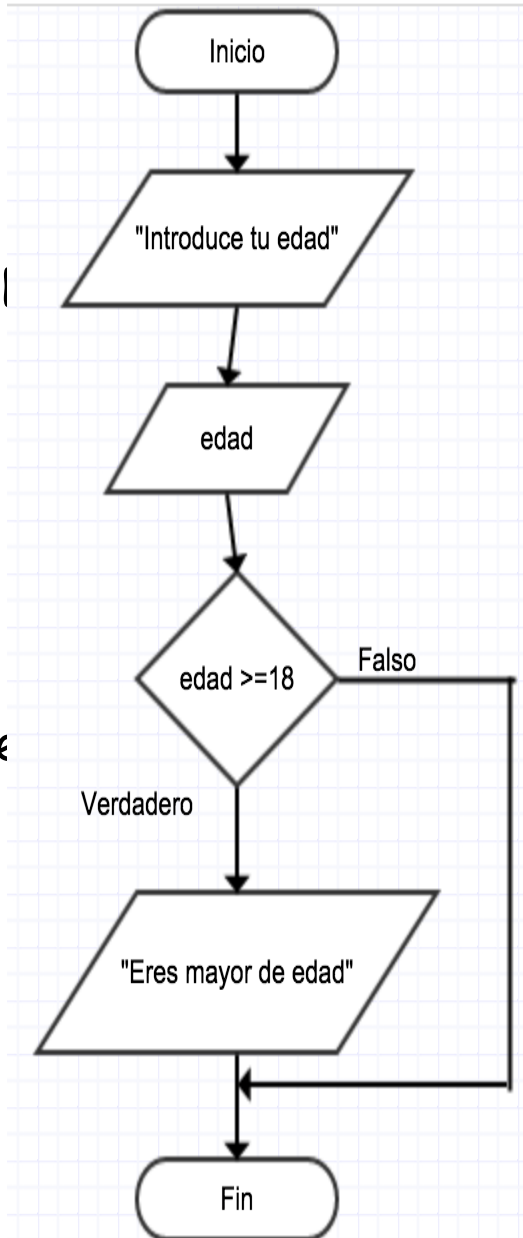
- Las estructuras condicionales simples permiten la toma de decisiones dentro de un programa.
- Dentro de una estructura selectiva, se incluye **una expresión lógica (que devuelve un valor booleano)** que será la condición que se evalúa para definir la ruta que se seguirá dentro del algoritmo.
  - Si la **expresión lógica** (condición) es verdadera, entonces, se ejecutarán todas las sentencias dentro del bloque de la estructura condicional simple.
  - Si la **expresión lógica** es falsa, no se ejecutará ninguna de estas sentencias.

- Representación en diagrama de flujo:

- En pseudocódigo:
- **Si** expresión\_lógica **Entonces**  
    Acción
- **Fin Si**



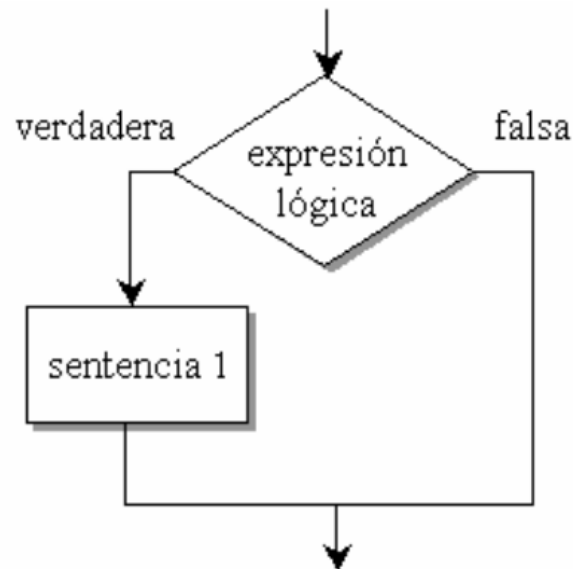
- Ejemplo: Desarrollar un algoritmo en donde se pide la edad del usuario; si es mayor de edad debe mostrarse un mensaje indicándolo.
- Pseudocódigo
- Entrada: **edad**
- Salida: mensaje mostrando si es mayor de edad
- Proceso verificarEdad
  - //Definir edad como Entero
  - 1. Escribir "Introduce tu edad"
  - 2. Leer **edad**
  - 3. **Si**  $\text{edad} \geq 18$  **Entonces**
    - 3.1 Escribir "Eres mayor de edad"
  - 4. Fin Si
- FinProceso



# Estructura condicional

- Sentencia if

```
if (expresionLogica) {  
    sentencia_1;  
}
```



# Ejemplo estructura selectiva

- Estructura simple
- Determine si la calificación de un alumno es aprobatoria
  - Si lo es, imprima “aprobado”
  - De lo contrario no imprima nada
- Imprimir si las ganancias de una persona están entre 500 y 1000
  - $i \in \mathbb{N} \mid i \in [500, 1000]$



# Ejemplo

```
using System;
public class Program
{
    public static void Main()
    {
        int ganancias=501;
        if(ganancias>500){
            Console.WriteLine("Las ganancias estan entre 500 y
1000");
        }
    }
}
```

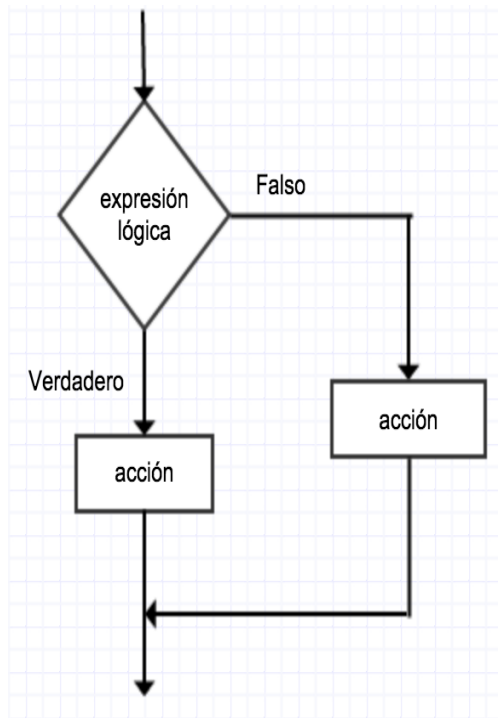
Algún  
problema?

No se evalúa si las  
ganancias son  
menores que 1000

Solución: usar  
operadores  
relacionales y lógicos  
para hacer ambas  
comparaciones

# Estructura condicional doble

- Otra forma de la estructura condicional, es la estructura selectiva doble, en la que se incluye una cláusula opcional **si-no** para establecer un grupo de acciones que se ejecutaran en caso de que la **expresión lógica** sea **falsa**.
- Diagrama de flujo:

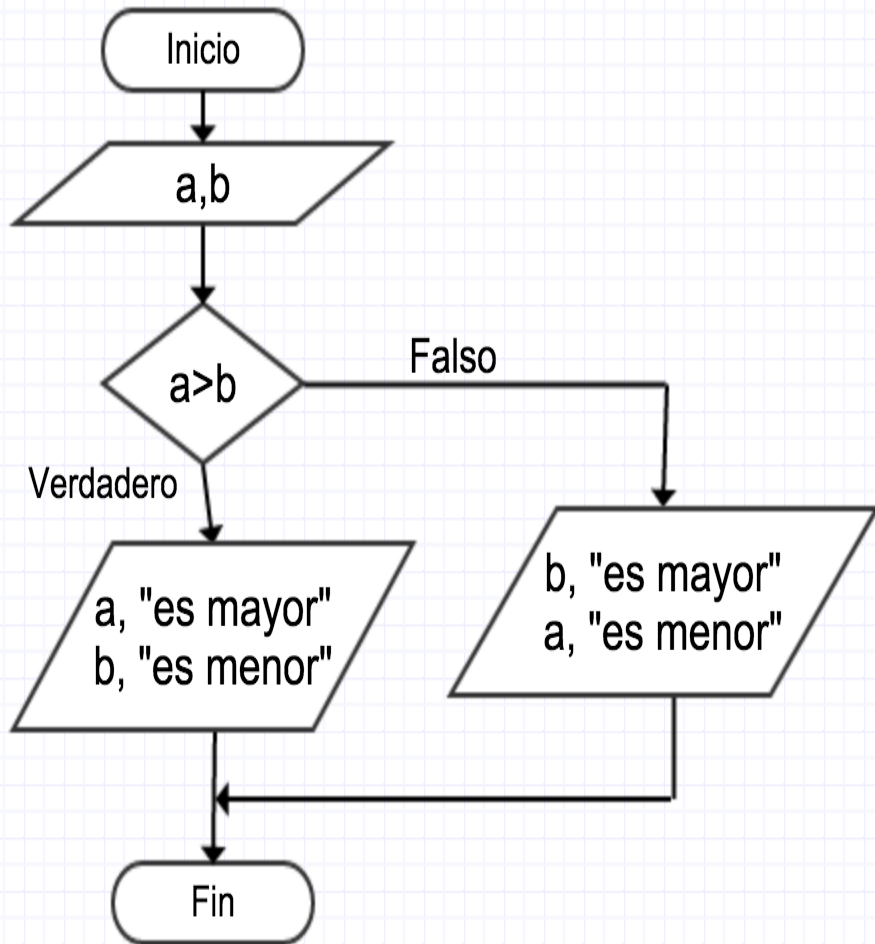


Pseudocódigo

```
Si expresión_lógica Entonces  
    acción  
Sino  
    acción  
Fin Si
```

- Ejemplo: Hacer un algoritmo que dados dos números distintos permita determinar cual es el mayor y cual es el menor de los dos.
- Pseudocódigo:
- Entrada: Números a y b
- Salida: Mensaje indicando el número mayor y el número menor
- Proceso MayorMenor
  - Definir a, b como Entero
  - 1. Leer a,b
  - 2. Si  $a > b$  Entonces
    - 2.1 Escribir a, “es mayor”
    - 2.2 Escribir b, “es menor”
  - 3. Si no
    - 3.1 Escribir b, “es mayor”
    - 3.2 Escribir a, “es menor”
  - 4. Fin Si
- Fin Proceso

# Diagrama de flujo:

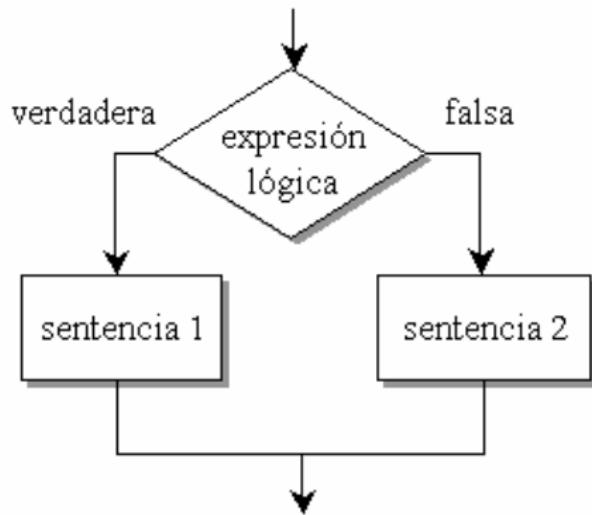


Prueba de escritorio:

a	b	a>b	Salida
5	8	Falso	8, "es mayor" 5, "es menor"
13	10	Verdadero	13, "es mayor" 10, "es menor"
45	2	Verdadero	45, "es mayor" 2, "es menor"

# Estructura condicional doble

- Se considera como una estructura selectiva con “rama”



```
if (expresionLogica) {  
    sentencia_1;  
}  
else {  
    sentencia_2;  
}
```

# Estructura condicional doble

- Comparando dos números

```
using System;

public class Program
{
    public static void Main()
    {
        int a=5
        int b=8;
        if(a<b){
            Console.WriteLine("a es mayor, b es menor");
        }else{
            Console.WriteLine("b es mayor, a es menor");
        }
    }
}
```

Algún  
problema?

Otro  
problema?

Terminar sentencia  
con ;

La expresión no  
corresponde

```
using System;

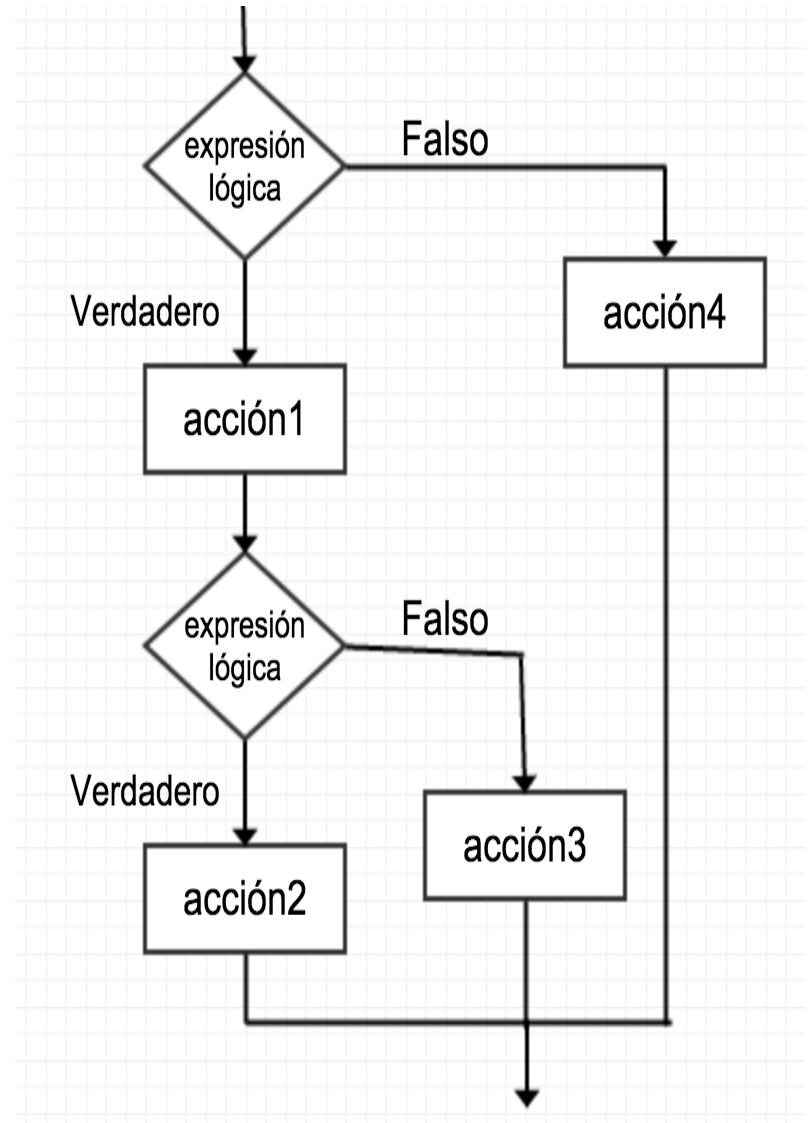
public class Program
{
    public static void Main()
    {
        int a=5;
        int b=8;
        if(a>b){
            Console.WriteLine("a es mayor, b es menor");
        }else{
            Console.WriteLine("b es mayor, a es menor");
        }
    }
}
```

# Estructura condicional anidada

- Las estructuras condicionales pueden **anidarse**, lo que significa que una estructura selectiva puede contener a su vez otra estructura selectiva
- Dentro de cualquiera de las secciones de una estructura condicional, sea *simple* o *doble*, puede incluirse otra estructura condicional que a su vez puede ser *simple* o *doble*
  - y puede contener también dentro de ella, otras estructuras condicionales o selectivas.

- Pseudocódigo  
Diagrama de flujo

```
Si expresión_lógica Entonces  
    acción1  
    Si expresión_lógica Entonces  
        acción2  
    Sino  
        acción3  
    Fin Si  
Sino  
    acción4  
Fin Si
```





- Ejemplo. Diseñar un algoritmo para determinar si un número dado  $N$  es positivo, negativo o cero.
- Pseudocódigo
- Entrada: Número  $n$
- Salida: Mensaje indicando si  $n$  es positivo, negativo o cero

Proceso PosNegCero

1. Leer  $n$

2. **Si**  $n=0$  **Entonces**

2.1 **Escribir**  $n$ , " es cero"

3. **Si no**

3.1 **Si**  $n > 0$  **Entonces**

3.1.1 **Escribir**  $n$ , " es positivo"

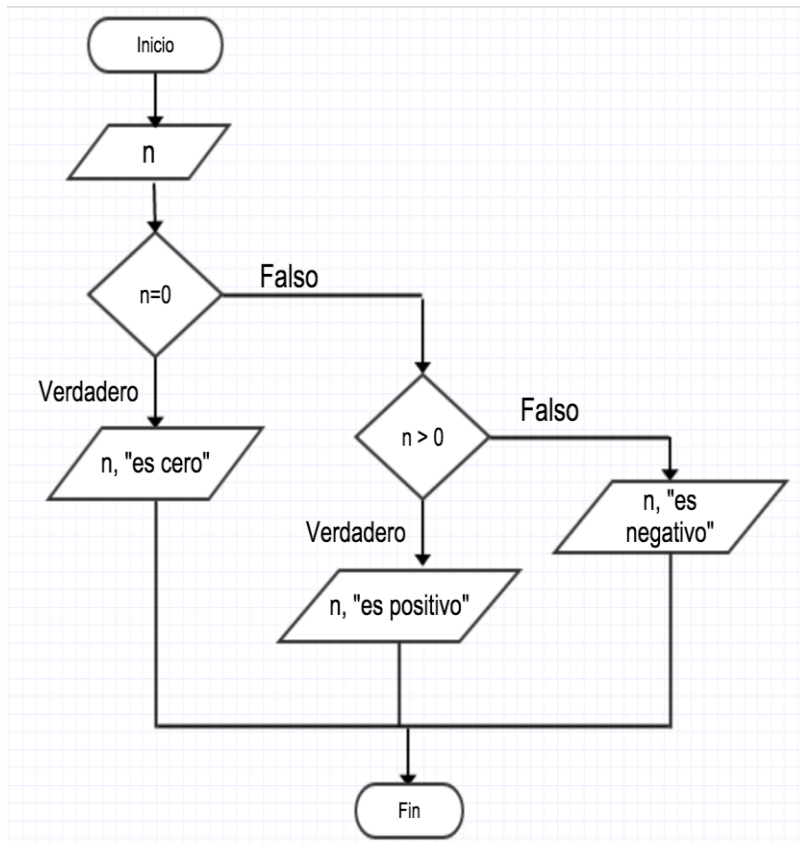
3.2 **Si no**

3.2.1 **Escribir**  $n$ , " es negativo"

3.3 **Fin Si**

4. **Fin Si**

## Diagrama de flujo



## Prueba de escritorio

n	n=0	n>0	Salida
5	Falso	Verdadero	5, "es positivo"
0	Verdadero	-	0, "es cero"
-8	Falso	Falso	-8, "es negativo"



# Ejemplo

- Diseñar un algoritmo para determinar si un número dado N es positivo, negativo o cero.

Algún problema?

Varios:  
Precisión de datos  
Operador lógico  
Bloque mal anidado

```
using System;

public class Program
{
    public static void Main()
    {
        int n=0.1;

        if(n=0){
            Console.WriteLine("n es cero");
        }else{
            if(n>0){
                Console.WriteLine("n es positivo");
            }else{
                Console.WriteLine("n es negativo");
            }
        }
    }
}
```

```
using System;

public class Program
{
    public static void Main()
    {
        int n=0;

        if(n==0){
            Console.WriteLine("n es cero");
        }else{
            if(n>0){
                Console.WriteLine("n es positivo");
            }else{
                Console.WriteLine("n es negativo");
            }
        }
    }
}
```

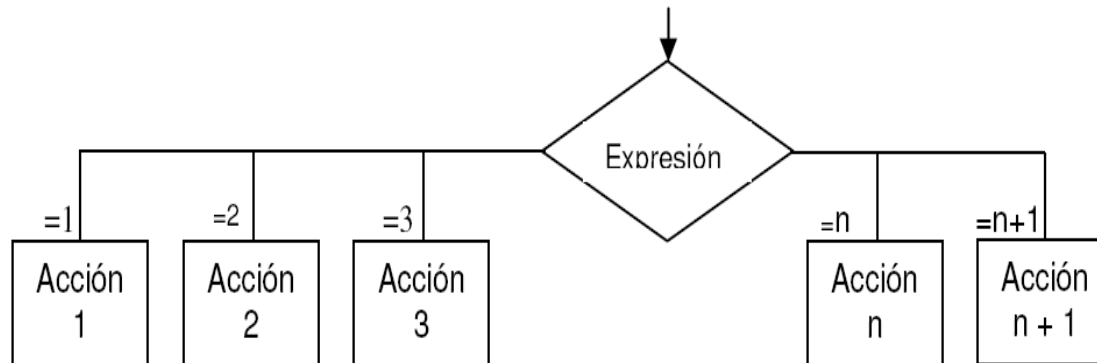
# Ejercicios

- 1. Diseñar un algoritmo para determinar si un número dado  $N$  es par o impar.
  - *(es par, si el residuo de dividir el número por dos 2 es cero, de lo contrario es impar)*
- 2. Diseñar un algoritmo que pida tres calificaciones, determine el promedio e indique si el alumno aprobó o no el curso. La calificación mínima aprobatoria es 6.0
- 3. Diseñar un algoritmo que dados tres números  $A$ ,  $B$  y  $C$ , determinar si la suma de una pareja de ellos es igual al tercer número, si se cumple esta condición imprima un mensaje que diga “Iguales” y en caso contrario imprima “Distintos”.

# Estructura de selección múltiple

- La estructura de selección múltiple **Según\_ Hacer** o casos, permite elegir una entre varias rutas posibles, evaluando para ello una expresión que puede tomar **n** valores distintos: 1, 2, 3, 4, ... , n.

- Diagrama de Flujo



- Pseudocódigo

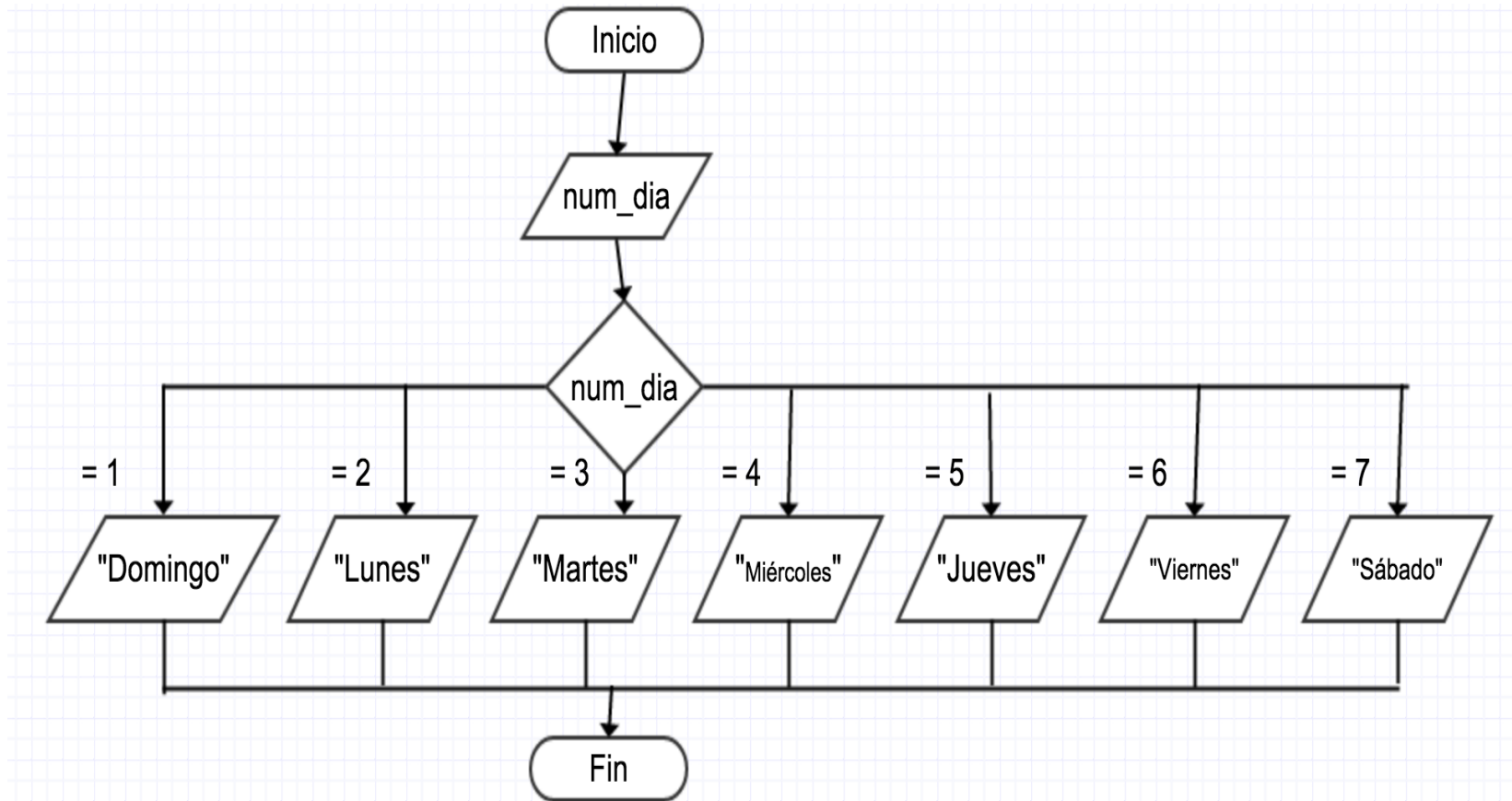
```
Segun variable_numerica Hacer
    opcion_1:
        secuencia_de_acciones_1
    opcion_2:
        secuencia_de_acciones_2
    opcion_3:
        secuencia_de_acciones_3
De Otro Modo:
    secuencia_de_acciones_dom
Fin Segun
```

- Ejemplo: Elabore un algoritmo que lea un valor entre 1 y 7 y despliegue domingo si es 1, lunes si es 2,..., sábado si es 7.
- Análisis
- ¿cuáles son los datos de entrada?
  - Un número entre 1 y 7
- ¿cuáles son los datos de salida?
  - Mensaje indicando si es Domingo, Lunes, Martes, .....  
Sábado
- Proceso:
- Paso 1. Leer el número de día
- Paso 2. Comparar el número de día, si es 1 se deberá mostrar el mensaje "Domingo", si es 2 se deberá mostrar el mensaje "Lunes", si es 3 se deberá mostrar el mensaje "Martes", si es 4 se deberá mostrar el mensaje "Miércoles", si es 5 se deberá mostrar el mensaje "Jueves", si es 6 se deberá mostrar el mensaje "Viernes", si es 7 se deberá mostrar el mensaje "Sábado"
- Paso 3. En cualquier otro caso se deberá indicar que el número está fuera del rango 1-7

- Ejemplo: Elabore un algoritmo que lea un valor entre 1 y 7 y despliegue domingo si el valor es 1, lunes si es 2,..., sábado si es 7.
- Pseudocódigo
- Entrada: número **num\_dia**
- Salida: Mensaje indicando "Domingo", "Lunes", "Martes" ..... ó "Sábado"
- Proceso **díaSemana**
  - Leer **num\_dia**
  - **Segun num\_dia Hacer**
    - 1: Escribir "Domingo"
    - 2: Escribir "Lunes"
    - 3: Escribir "Martes"
    - 4: Escribir "Miércoles"
    - 5: Escribir "Jueves"
    - 6: Escribir "Viernes"
    - 7: Escribir "Sábado"
  - **Fin Segun**
- **FinProceso**



## Diagrama de flujo



# Estructura switch

- La estructura de selección múltiple se conoce como “switch”

```
switch (expresion) {  
    case valor_1: sentencias_1; break;  
    case valor_2: sentencias_2; break;  
    ...  
    case valor_n: sentencias_n; break;  
    [default: sentencias_x;  
}
```

# Ejercicios

- Ejercicio 1. Elabore un algoritmo que lea el número de mes entre 1 y 12 y, que imprima el nombre del mes correspondiente: si es 1 "Enero", si es 2 "Febrero", etc.
- Ejercicio 2. Dados la categoría y el sueldo del trabajador, calcular el aumento correspondiente, teniendo en cuenta la siguiente tabla:

Categoría	Aumento
A	3%
B	5%
C	10%
D	15 %