

# Introducción a la Programación y Computación 1 Sección E

Ing. MSc. Neftalí Calderón

# Diagrama de flujo

Un Diagrama de Flujo representa la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema.



**Inicio o fin del diagrama**



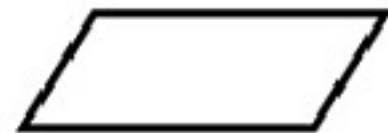
**Acción o proceso**



**Toma de decisión**



**Salida de información**



**Entrada de información**



**Subprograma**



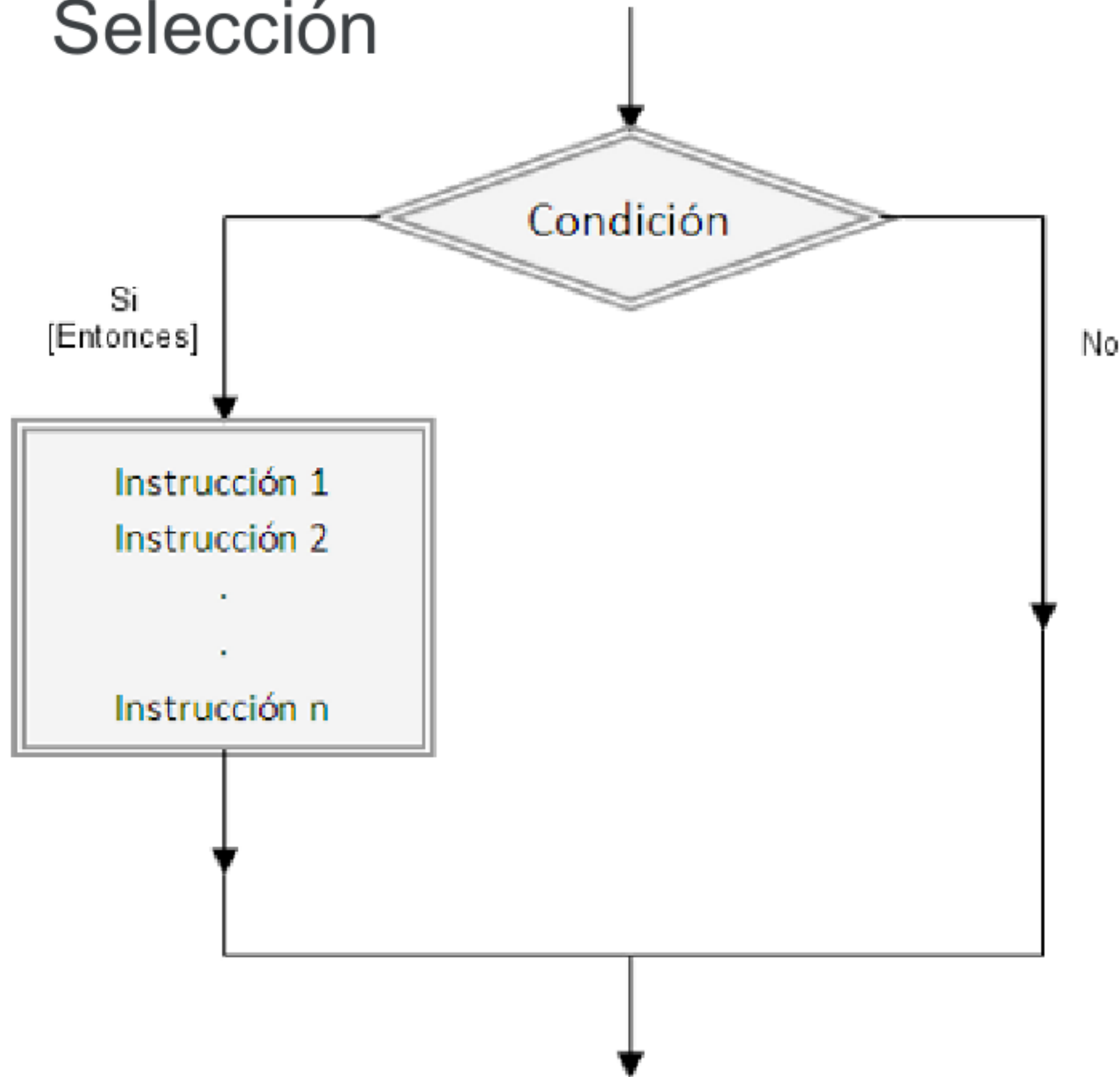
**Conector**

# Estructuras algorítmicas

Las estructuras de operación de programas son un grupo de formas de trabajo, que permiten, mediante la manipulación de variables, realizar ciertos procesos específicos que nos lleven a la solución de problemas



# Selección



```
int nota;  
leer(nota);
```

```
if (nota > 60) {  
    escribir("Aprobado");  
}  
else {  
    escribir("Reprobado");  
}
```

```
instrucción1;  
instrucción2;  
instrucción3;
```

```
boolean flag;  
flag = true;
```

```
if (flag) {  
    instrucción1;  
}  
instrucción2;  
instrucción3;  
instrucción4;
```

```

int zona;
int lab;

leer(zona);
leer(lab);

if (zona > 60) && (lab > 60) {
    escribir("Aprobado");
}
else {
    escribir("Reprobado");
}

```

```

int zona;
int lab;

leer(zona);
leer(lab);

if (zona > 60) {
    if (lab > 60) {
        escribir("Aprobado");
    }
    else {
        escribir("Reprobado");
    }
}
else {
    escribir("Reprobado");
}

```

```

instrucción1;
instrucción2;
.
.
.

```

```

if ( (parcial1 >60) && (parcial2 >60) && (parcial3 >60) ) {
    if ( (lab > 60) || ( (zona > 60) && (lab > 60) ) ) {
        escribir("Aprobado");
    }
    else {
        escribir("Reprobado");
    }
}

```

```

int parcial1;
int parcial2;
int parcial3;
int final;
int lab;

```

```

leer(parcial1);
leer(parcial2);
leer(parcial3);
leer(final);
leer(lab);

```

```

if ( (parcial1 >60) && (parcial2 >60) && (parcial3 >60)
&& (lab >60) ) || ( (zona > 60) && (lab > 60) ) {
    escribir("Aprobado");
}
else {
    escribir("Reprobado");
}

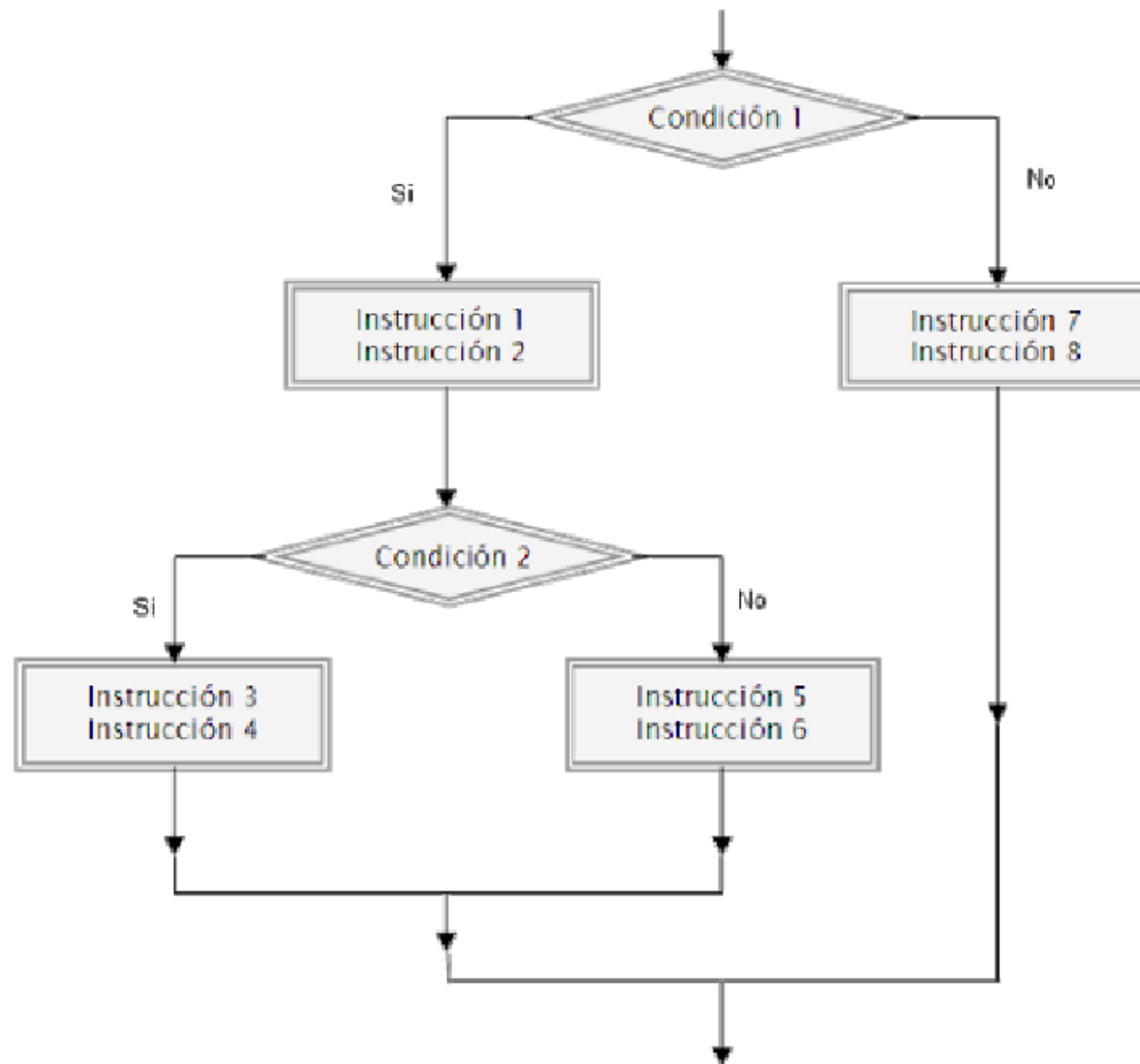
```

and &&  
or ||

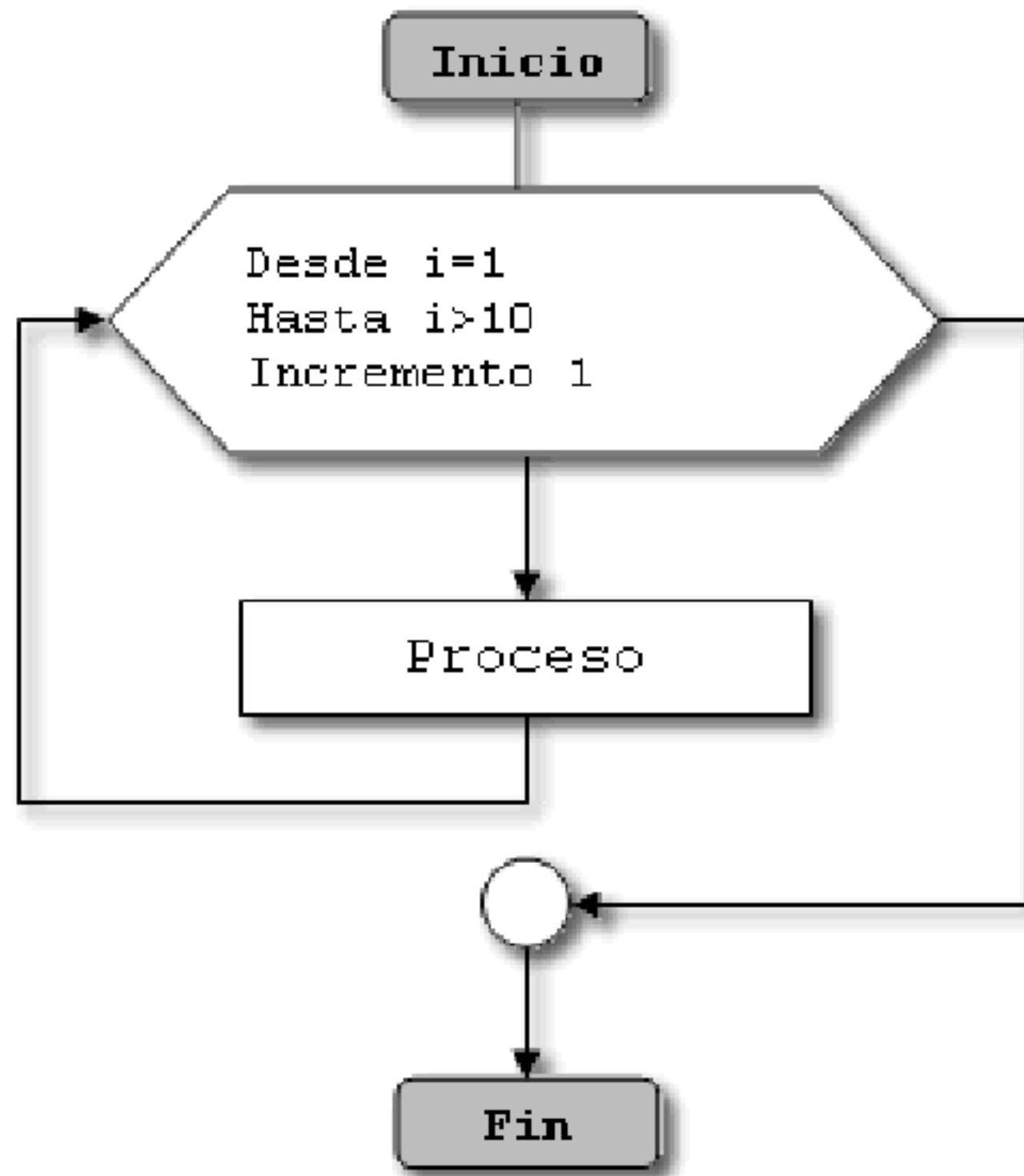
```
int mes;  
  
if (mes == 1) {  
    escribir("Enero");  
}  
else if (mes == 2) {  
    escribir("Febrero");  
}  
else if (mes == 3) {  
    escribir("Marzo");  
}  
...  
...  
....  
else if (mes == 12) {  
    escribir("Diciembre");  
}  
else {  
    escribir("me no válido");  
}  
}
```

```
int mes;  
  
switch(mes) {  
    case 1: escribir("Enero"); break;  
    case 2: escribir("Febrero"); break;  
    case 3: escribir("Marzo"); break;  
    case 4: escribir("Abril"); break;  
    .  
    .  
    .  
    case 12: escribir("Diciembre"); break;  
    else: escribir("mes no válido");  
}
```





# Repetición (for)



```
for (int i=1; i<10; i++) {  
    imprimir(i);  
}  
instrucción1;  
instrucción2;  
instrucción3;
```

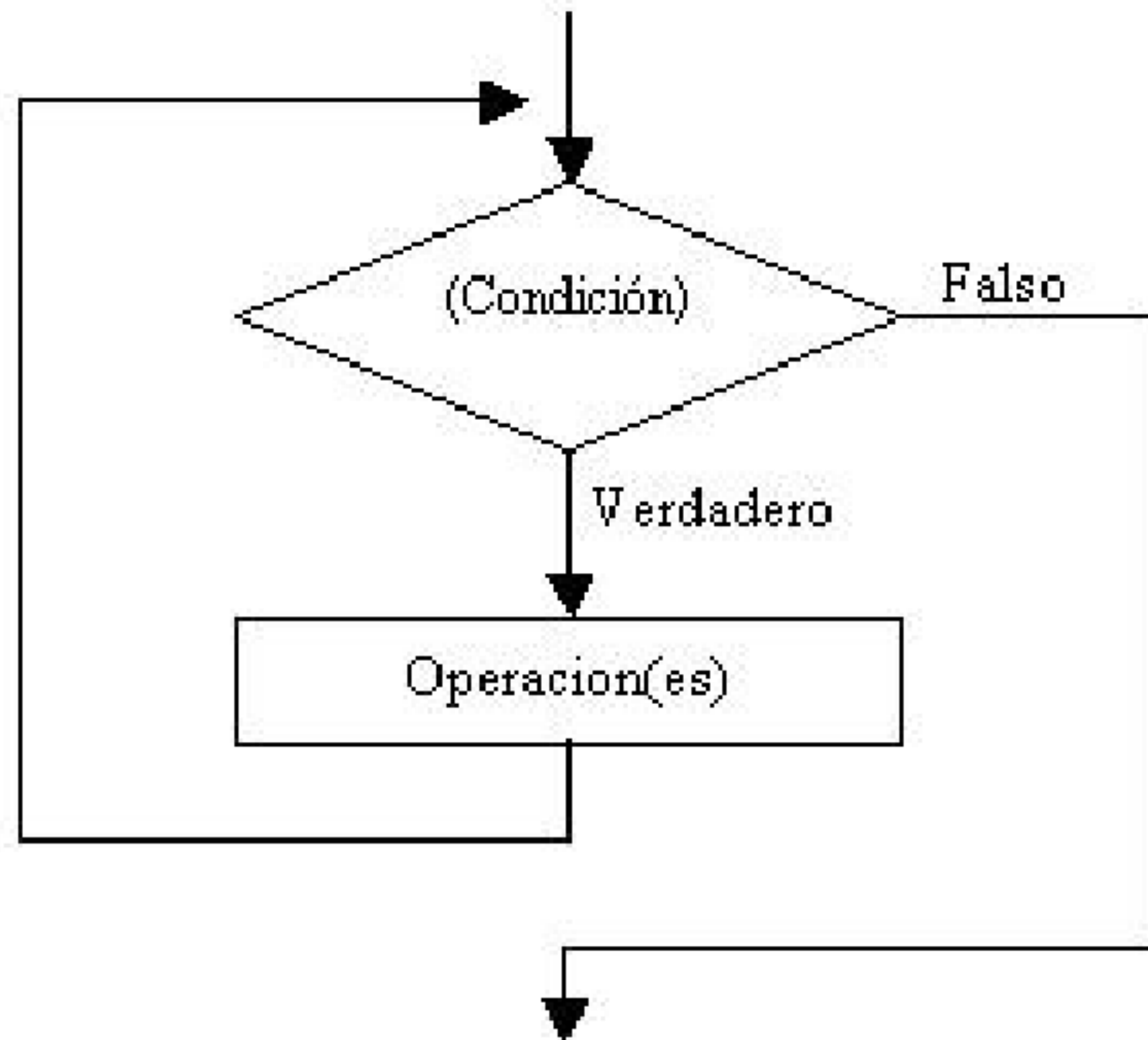
1  
2  
3  
4  
5  
6  
7  
8  
9

```
for (int i=1; i<10; i=i+2) {  
    imprimir(i);  
}  
instrucción1;  
instrucción2;  
instrucción3;
```

```
int j=5;  
for (int i=1; (i<10) && (j>5); i=i+2) {  
    imprimir(i);  
    j = j + i;  
}  
instrucción1;  
instrucción2;  
instrucción3;
```



# While



```
int i = 10;
```

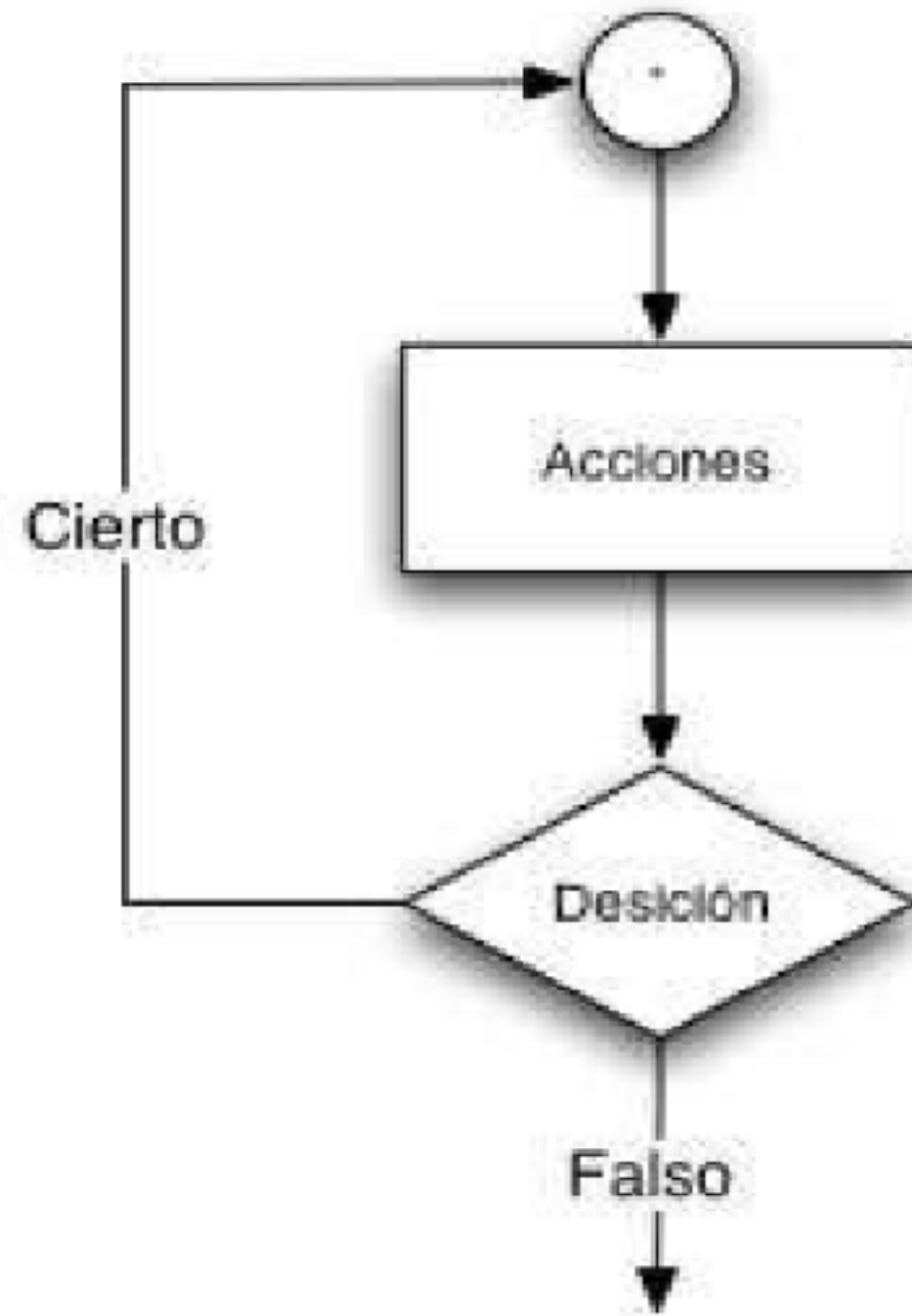
```
while (i < 10) {  
    imprimir(i);  
    i++;  
}
```

```
instrucción1;  
instrucción2;
```

```
boolean flag=true;
```

```
while(flag) {  
    leer(nota);  
    if (nota < 0) {  
        flag = false;  
    }  
}
```

## Do-while



```
int i = 10;
```

```
do {  
    imprimir(i);  
    i++;  
    if () {  
        for () {  
            while () {  
            }  
        }  
    }  
} while (i < 10)
```

```
instrucción1;  
instrucción2;
```