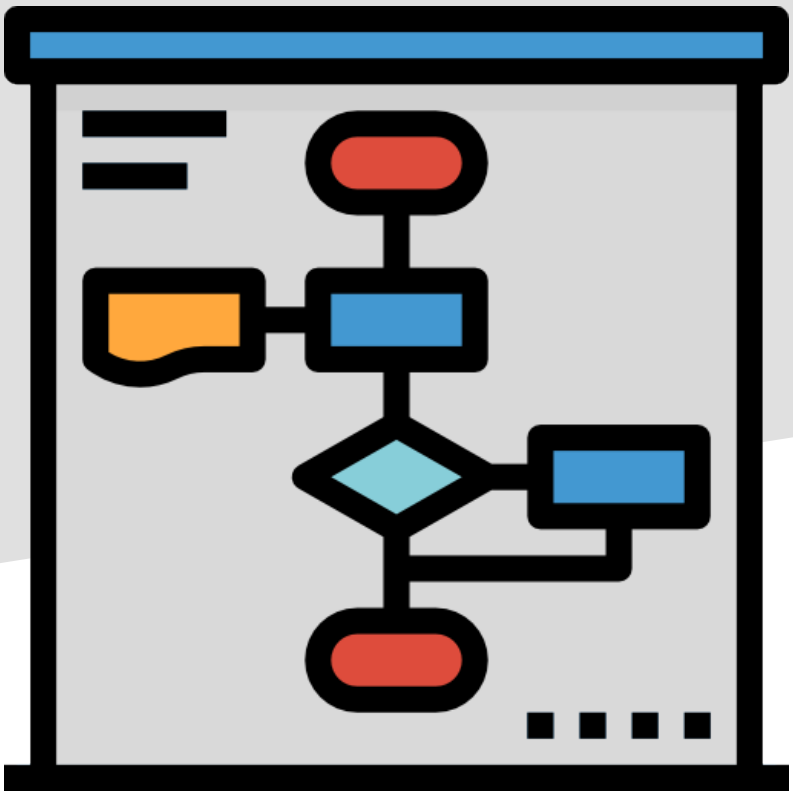
Tema 01. Programación estructurada



Índice

1. Introducción
2. Diagramas de flujo
3. Pseudocódigo
4. Datos
5. Tipos de datos básicos
6. Operadores y expresiones
7. Tipos de instrucciones
8. Contadores, acumuladores e interruptores

1.- Introducción.

- **Programa**: conjunto de instrucciones que dirige el comportamiento del ordenador.

- **Lenguaje de programación**: conjunto de símbolos y caracteres combinados entre sí de acuerdo con una sintaxis definida utilizado para transmitir órdenes o instrucciones al ordenador.

1.- Introducción.

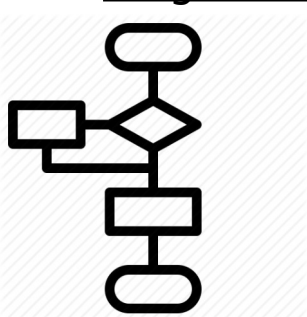
- **Código máquina**: lenguaje utilizado directamente por el procesador (instrucciones en binario).
- **Programa fuente**: escrito en un lenguaje de programación concreto.
- **Programa traductor**: se encarga de traducir el programa fuente en código máquina. Previamente comprobará si la sintaxis es correcta e informará de los errores.

1.- Introducción.

- **Algoritmo**: es un procedimiento a seguir para resolver un problema. Hay que indicar:
  - Acciones que se ejecutan.
  - Orden de las acciones

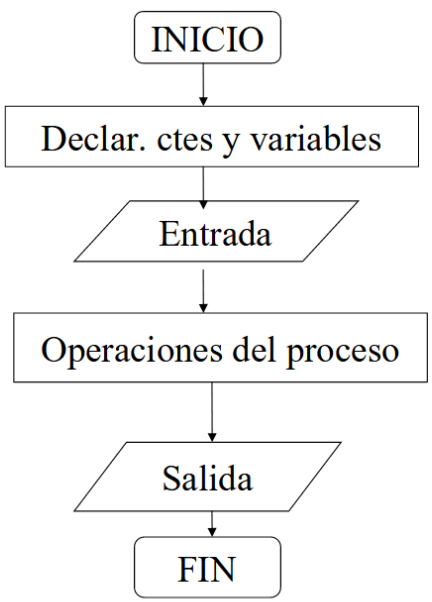
El algoritmo es la “base” para construir el programa. Un algoritmo es correcto cuando dá la solución al problema a tratar.

- **Representación de un algoritmo**:
  - Texto: pseudocódigo.
  - Gráfica: diagrama de flujo.



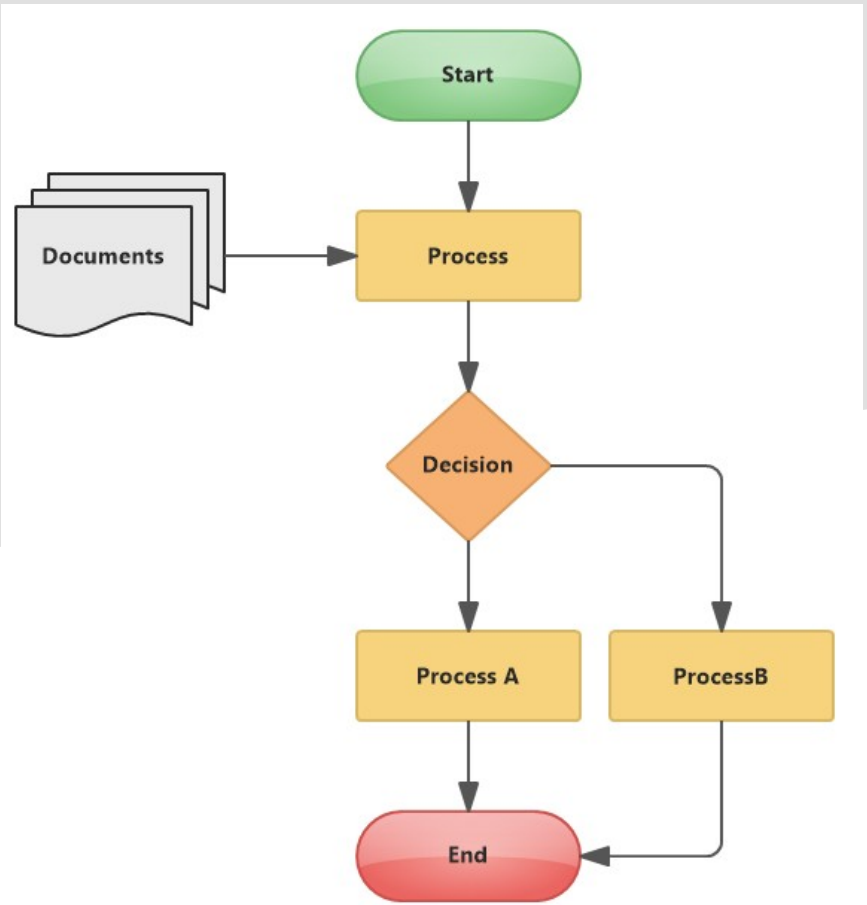
2.- Diagramas de flujo.

- Representación gráfica de un algoritmo. Ayuda a entender los principios de programación de forma gráfica.
- Muestra el orden en que se van ejecutando las instrucciones.
- Consta de:
  - Símbolo de inicio.
  - Secuencia de operaciones
    - Instrucciones de entrada/salida
    - Instrucciones de proceso.
  - Símbolo de fin.
  - Líneas de flujo.



2.- Diagramas de flujo.

- Ejemplo de diagrama de flujo



3.- Pseudocódigo.

Es un lenguaje de programación “informal” que se utiliza para aprender de forma más sencilla los principios de la programación.

```
PROGRAMA MiPrimerPrograma
    CONSTANTES
    VARIABLES
    INICIO
    ...
    FIN
FIN MiPrimerPrograma
```

4.- Datos.

Un dato es la unidad mínima de almacenamiento dentro del sistema. Tiene:

- **Identificador**: es el nombre utilizado en el programa para hacer referencia al dato.
- **Tipo**: establece el rango de valores que puede tomar el dato. Determinará el espacio de memoria reservado para el dato.
- **Valor**: información asociada al dato.

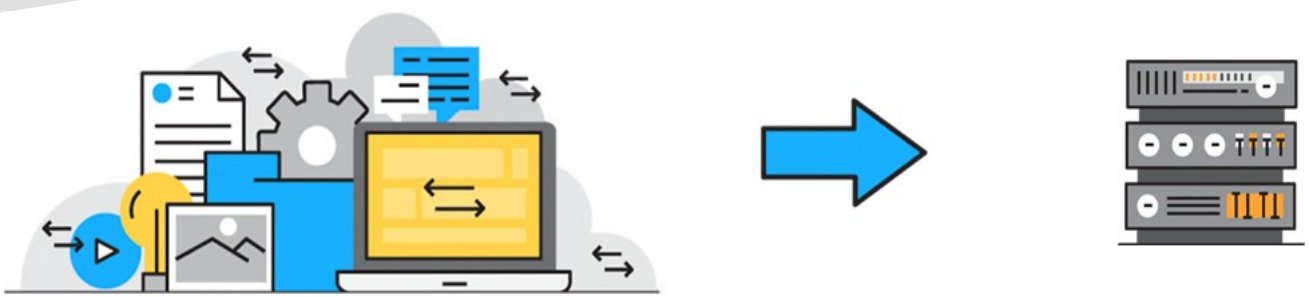
Pueden ser constantes o variables.

4.- Datos.

- **Constantes**:
  - Su valor es fijo durante la ejecución del programa.
  - Su identificador será siempre en mayúsculas.
    - *ID_CTE: TIPO = valor*
- **Variables**:
  - Su valor variará durante la ejecución del programa.
  - Su identificador debe comenzar por una letra en minúscula.
    - *id_var: TIPO*

5.- Tipos de datos básicos.

- **Entero**: conjunto de los números enteros: positivos, negativos, y cero (sin decimales).
- **Real**: conjunto de los números reales (con decimales).
- **Carácter**: cualquier símbolo del código ASCII: letras, números, símbolos...
- **Lógico** (booleano): valores verdadero (V) y false (F).



6.- Operadores y expresiones.

SÍMBOLO	***OPERACIÓN***
-	Signo negativo
( )	Paréntesis, precedencia de operaciones
**+, - , *, /, %, ^, **	Aritméticos: suma, resta, multiplicación, división, módulo, potencia, división entera
<, >, <=, >=, =, <>	Relacionales: mayor, menor, mayor que, menor que, igual, distinto
NOT, AND, OR	Lógicos: no es, y, o

6.- Operadores y expresiones.

- Orden de prioridad de operadores y expresiones

1. () Paréntesis
2. - Signo negativo
3. ^ Potencia y raiz
4. * Producto, / División y % módulo
5. + Suma y – resta
6. <, >, <=, >=, =, <> Relaciones
7. NOT negación
8. AND conjunción
9. OR disyunción

Los que aparezcan en la misma línea tienen la misma prioridad.

En este caso se evaluan de izquierda a derecha.

6.- Operadores y expresiones.

- Operador **OR**

A **OR** B devuelve VERDADERO si A es VERDADERO o B es VERDADERO

A	B	*A OR B*
FALSO	FALSO	**FALSO**
FALSO	VERDADERO	**VERDADERO**
VERDADERO	FALSO	**VERDADERO**
VERDADERO	VERDADERO	**VERDADERO**

6.- Operadores y expresiones.

- Operador **AND**

A **AND** B devuelve VERDADERO si A es VERDADERO y B es VERDADERO

A	B	*A AND B*
FALSO	FALSO	**FALSO**
FALSO	VERDADERO	**FALSO**
VERDADERO	FALSO	**FALSO**
VERDADERO	VERDADERO	**VERDADERO**

6.- Operadores y expresiones.

- Operador **NOT**

NOT A devuelve VERDADERO si A es FALSO

A	***NOT A***
FALSO	**VERDADERO**
VERDADERO	**FALSO**

** Esta operación se realiza sobre un único dato.*

6.- Operadores y expresiones.

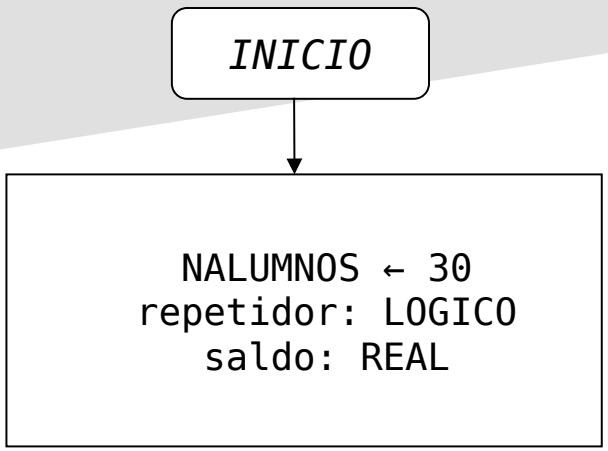
Tipos de expresiones

- **Aritméticas**: sus operandos son numéricos y su resultado es numérico.
  - Ejemplos: *5 + 2*, *5 – 2*
- **Relacionales**: sus operandos son numéricos y su resultado es lógico.
  - Ejemplos: *5 < 2*, *5 > 2*
- **Lógicas**: sus operandos son lógicos y su resultado es lógico.
  - Ejemplos: *2 < 3 AND 10 > 5*, *2 < 3 OR 10 > 10*

7.- Tipos de instrucciones.

Instrucciones de definición de datos

- **Definición de constante**:
  - *ID_CTE = VALOR*
- **Definición de variable**:
  - *idVar1, idVar2: TIPO*
- **Diagrama de flujo**: →



7.- Tipos de instrucciones.

Instrucciones primitivas

- **Instrucción de asignación**:

Almacena en una variable un dato o resultado de una expresión.

Se evalua la expresión y su valor se asigna a la variable

- Sintaxis: *idVariable ← expresión*
- Ejemplos: *edad ← 25*

  saldo ← saldo - 100

7.- Tipos de instrucciones.

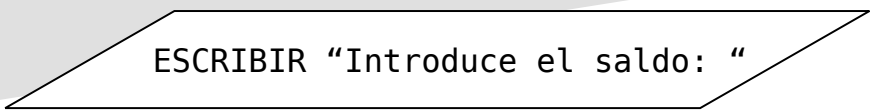
Instrucciones primitivas

- **Instrucciones de salida**:

Escriben variables y expresiones en un dispositivo de salida (pantalla).

- Sintaxis pseudocódigo: *ESCRIBIR expresión*

- Diagrama de flujo: →



- Si se quiere escribir sin salto de línea, se usará la siguiente sintaxis:

ESCRIBIR_SS expresión

7.- Tipos de instrucciones.

Instrucciones primitivas

- **Instrucciones de entrada**:

Recogen un valor de un periférico de entrada (teclado) y lo almacenan en memoria en una variable.

- Sintaxis pseudocódigo: *LEER id_variable*

- Diagrama de flujo: →

7.- Tipos de instrucciones.

Instrucciones de control

Controlan el flujo de ejecución de un programa.

Son de 3 tipos:

- Instrucciones secuenciales.
- Instrucciones condicionales o alternativas:
  - Alternativa simple (*SI…*)
  - Alternativa doble (*SI*… *SI NO*…)
  - Alternativa múltiple (*SEGÚN VALOR*…)
- Instrucciones repetitivas:
  - Instrucción *MIENTRAS*…
  - Instrucción *REPETIR*… *MIENTRAS*
  - Instrucción *PARA*…

7.- Tipos de instrucciones.

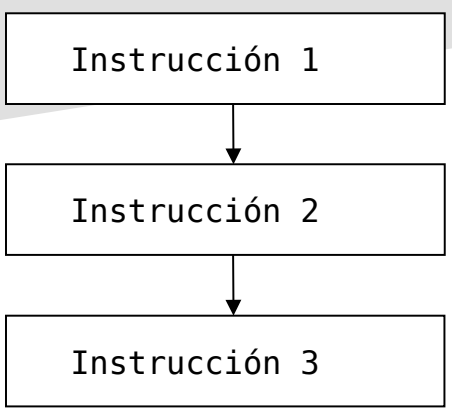
- Instrucciones secuenciales

El orden de ejecución de las instrucciones es de izquierda a derecha y de arriba a abajo. Una detrás de otra.

Pseudocódigo.

```
Instrucción 1
Instrucción 2
Instrucción 3
...
```

Diagrama de flujo.



7.- Tipos de instrucciones.

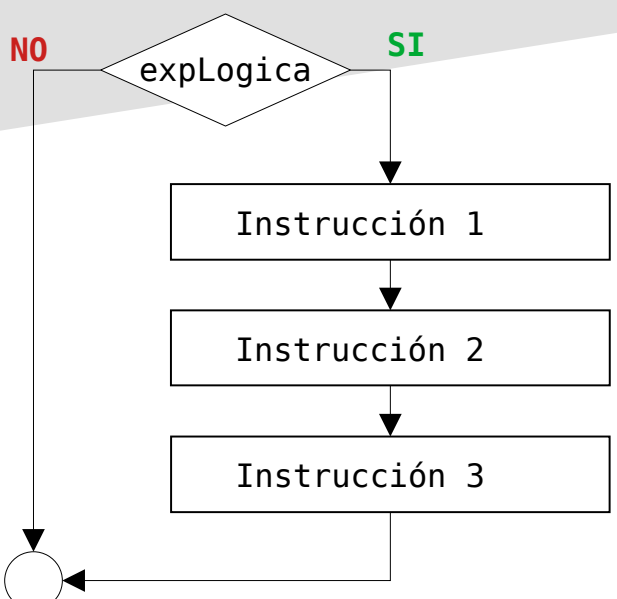
- Instrucciones condicionales**:** **ALTERNATIVA SIMPLE**

Controlan la ejecución de un conjunto de instrucciones en función de que se cumpla o no una condición (expresión lógica) previamente establecida.

Pseudocódigo

```
SI expLogica
  Instrucción 1
  Instrucción 2
  Instrucción 3
  …
FIN SI
```

Diagrama de flujo



7.- Tipos de instrucciones.

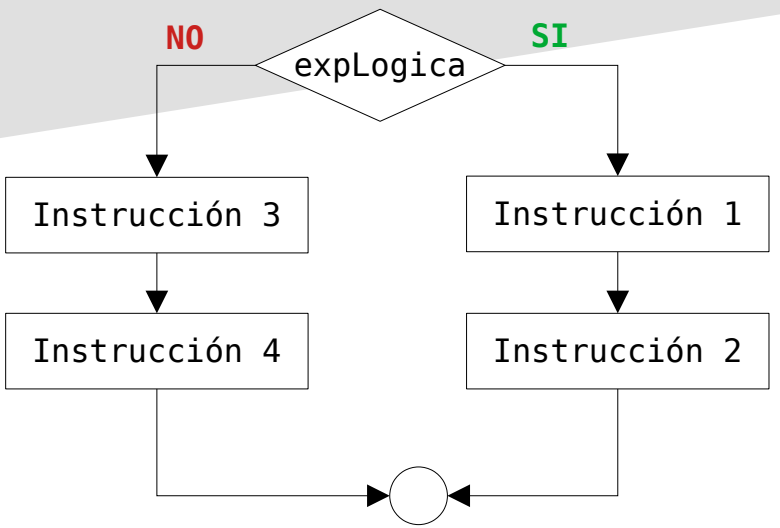
- Instrucciones condicionales**:** **ALTERNATIVA DOBLE**

Controlan la ejecución de un conjunto de instrucciones en función de que se cumpla o no una condición (expresión lógica) previamente establecida.

Pseudocódigo

```
SI expLogica
  Instrucción 1
  Instrucción 2
SI NO
  Instrucción 3
  Instrucción 4
FIN SI
```

Diagrama de flujo



7.- Tipos de instrucciones.

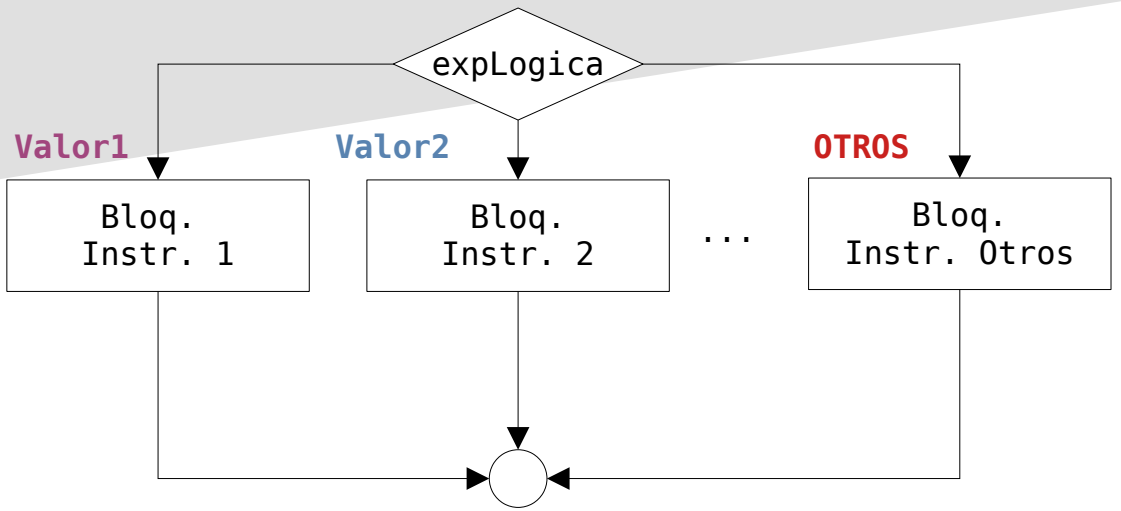
- Instrucciones condicionales**:** **ALTERNATIVA MÚLTIPLE**

Controlan la ejecución de un conjunto de instrucciones en función de que se cumpla o no una condición (expresión lógica) previamente establecida.

Pseudocódigo

```
SEGÚN_VALOR expLogica
  Valor1:
    Bloq. Instr. 1
  Valor2:
    Bloq. Instr. 2
  ...
  OTROS:
    Bloq. Instr. otros
FIN SEGÚN_VALOR
```

Diagrama de flujo



7.- Tipos de instrucciones.

- Instrucciones repetitivas**:** **REPETIR MIENTRAS**

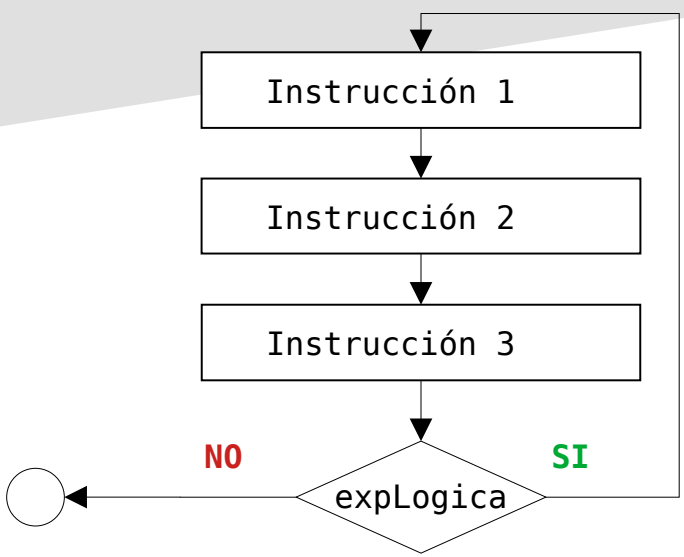
Repiten **de 1 a N veces** un conjunto de acciones mientras se cumpla una determinada condición (expresión lógica).

LA EXPRESIÓN LÓGICA DEBE ACTUALIZARSE PARA NO CREAR UN BUCLE INFINITO

Pseudocódigo

```
REPETIR
  Instrucción 1
  Instrucción 2
  Instrucción 3
  …
MIENTRAS expLogica
```

Diagrama de flujo



7.- Tipos de instrucciones.

- Instrucciones repetitivas**:** **REPETIR**

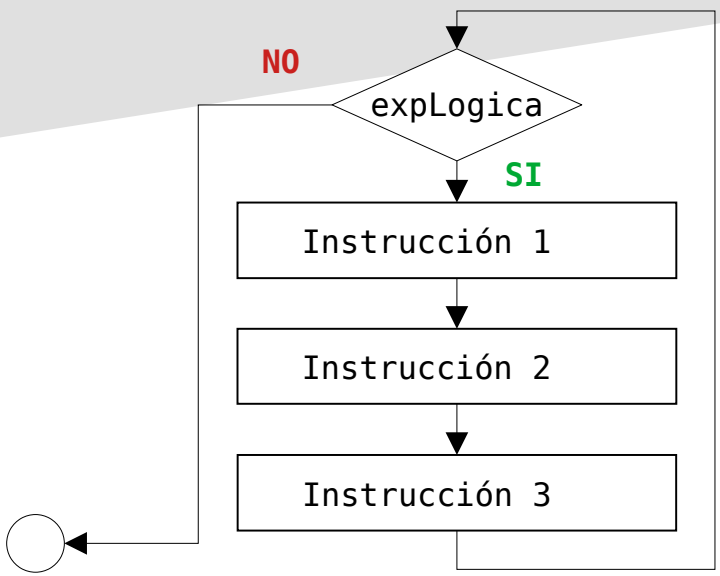
Repiten **de 0 a N veces** un conjunto de acciones mientras se cumpla una determinada condición (expresión lógica).

LA EXPRESIÓN LÓGICA DEBE ACTUALIZARSE PARA NO CREAR UN BUCLE INFINITO

Pseudocódigo

```
MIENTRAS expLogica
  Instrucción 1
  Instrucción 2
  Instrucción 3
  …
FIN MIENTRAS
```

Diagrama de flujo



7.- Tipos de instrucciones.

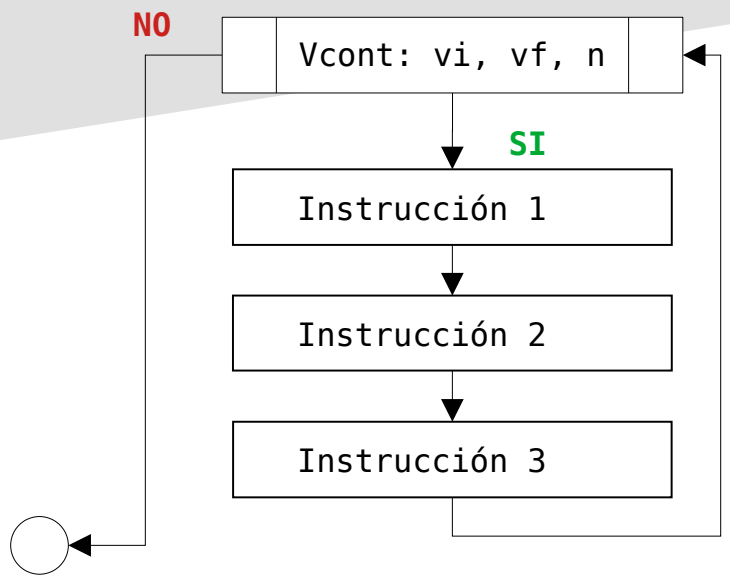
- Instrucciones repetitivas**:** **PARA**

Repiten **un número de veces fijo** un conjunto de acciones. Debe establecerse un contador (inicializado) un valor inicial y un valor final.

Pseudocódigo

```
PARA vcont DE vIni A vFin
  CON INC = num
  Instrucción 1
  Instrucción 2
  Instrucción 3
  …
FIN PARA
```

Diagrama de flujo



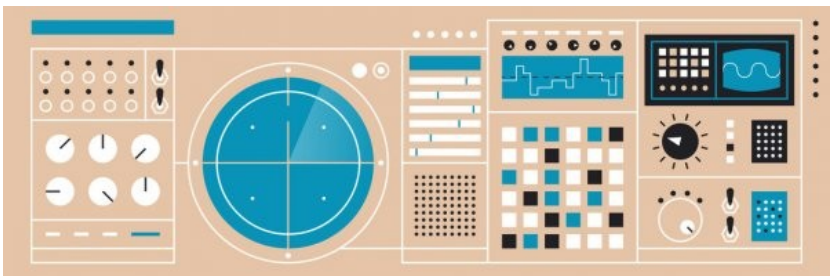
7.- Tipos de instrucciones.

- Instrucciones repetitivas**:** **PARA**
  - ***vcont*** es una variable entera que se usa como contador.
  - ***vi*** es el valor inicial, puede ser una expresión.
  - ***vf*** es el valor final, puede ser una expresión.
  - Si no aparece ***CON INC =*** … se supone que el incremento es 1.
  - El incremento puede ser negativo, entonces el valor inicial debe ser mayor que el valor final.

8.- Contadores, acumuladores e interruptores.

Son variables de programa que realizan una determinada función.

- Contador: variable entera que se incrementa o decrementa en una cantidad fija constante.
- Acumulador: variable numérica que almacena un total acumulado de operaciones con variables.
- Interruptor o flag: variable lógica (booleana) para recordar la ocurrencia de un determinado suceso dentro del programa.



Tema 01. Programación estructurada