Tema 1: Programación Estructurada

Indice

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 ordenes o instrucciones al ordenador.

2.Definiendo conceptos

- **Código máquina**: Es el 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 a código máquina.

  Previamente comprobara si la sintaxis es correcta e informará de los errores

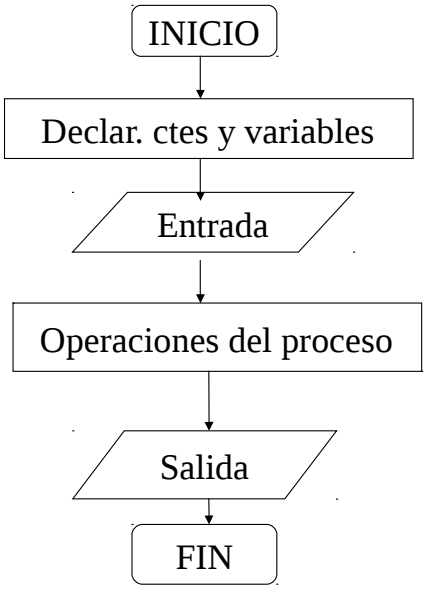
2. Definiendo conceptos

- **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 da la solución al problema a tratar
- Representación de una algoritmo
  - Gráfica: **Diagrama de flujo**
  - Texto: **Pseudocódigo**

3. Diagramas de flujo

- Representación gráfica de un algoritmo. Ayuda a entender los principios de la 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

Ejemplo de diagrama de flujo



4. 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
```

5. Datos. Constantes y variables

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 un rango de valores que puede tomar un dato. Determinará el espacio de memoria reservado para el dato.
- Valor

Pueden ser **constantes** o **variables**

5. Datos. Constantes y variables

- Constantes
  - Su valor es fijo durante la ejecución del programa
  - Su identificador será siempre en mayúsculas
  - IDE_CTE= valor
- Variables
  - Su valor variará durante la ejecución del programa
  - ide_var: TIPO

6. Tipos de datos básicos

- ENTERO: Conjunto de los números enteros, positivos, negativos y cero (sin decimales)
- REAL: Conjuntos de los números reales (con decimales)
- CARÁCTER: Cualquier carácter del código ASCII, letras, números, símbolos, etc.
- LOGICO: Valores **V**erdadero (V) y **F**also (F)

7. Operadores y Expresiones

Signo negativo	-
Parentesis	( )
Aritméticos	Suma(+), resta(-), multiplicación(*), división(/) div o \ División entera mod o % Resto división ^ Potencia, RAIZ() Raíz cuadrada
Relacionales	< > >= <= = <>
Lógicos	NOT, AND, OR

Prioridad

De mayor a menor esta es la prioridad de los operadores:

- Paréntesis ( )
- Signo -
- Potencia ^, Raíz RAIZ()
- Producto(*), división(/), división entera (\ ó div), resto de división (% ó mod)
- Suma (+) y resta (-)
- Relacionales (<, <=, >, >=, = ,<>)
- Negación (NOT)
- Conjunción (AND)
- Disyunción (OR)

Los que aparecen en la misma línea tienen la misma prioridad. En este caso se evalúan de izquierda a derecha

Operador OR

a OR b devuelve VERDADERO si a es VERDADERO o b es VERDADERO

a	b	a OR b
F	F	F
F	V	V
V	F	V
V	V	V

Operador AND

a AND b devuelve VERDADERO si a es VERDADERO y b es VERDADERO

a	b	a AND b
F	F	F
F	V	F
V	F	F
V	V	V

Operador NOT

NOT a

devuelve VERDADERO si a es FALSO

a	NOT a
F	V
V	F

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 AND 10<10

Prioridad

De mayor a menor esta es la prioridad de los operadores:

- Paréntesis ( )
- Signo -
- Potencia ^, Raíz RAIZ()
- Producto(*), división(/), división entera (\ ó div), resto de división (% ó mod)
- Suma (+) y resta (-)
- Relacionales (<, <=, >, >=, = ,<>)
- Negación (NOT)
- Conjunción (AND)
- Disyunción (OR)

Los que aparecen en la misma línea tienen la misma prioridad. En este caso se evalúan de izquierda a derecha

8. Tipos de instrucciones

8.1. Instrucciones de definición de datos

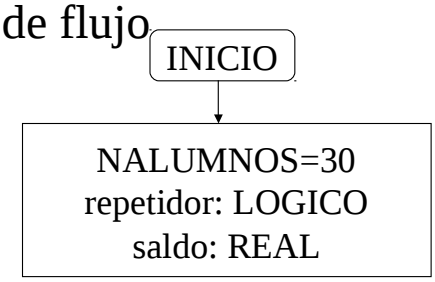
- Definición de constante

  IDE_CTE= VALOR

- Definición de variable

  ideVar1, ideVar2: TIPO

- Diagrama de flujo



8.2. Instrucciones primitivas

Instrucción de asignación

- Para almacenar en una variable un dato o resultado de una expresión
- Se evalúa la expresión y su valor se asigna a la variable. Sintaxis:

  id_variable ← expresion

- Ejemplos:

  edad ← 15

  saldo ← saldo - 100

8.2. Instrucciones primitivas

Instrucciones de Salida

- Escriben variables y expresiones en un dispositivo de salida (pantalla)
- Sintaxis Pseudocódigo:

ESCRIBIR expresion

- Diagrama de flujo

8.2. Instrucciones primitivas

Instrucciones de Salida

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

ESCRIBIR_SS expresion

8.2. 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

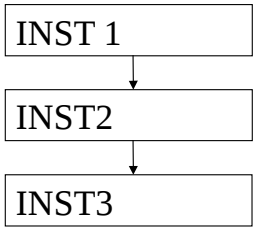
- Diagrama de flujo

8.3. Instrucciones de control

- Controlan el flujo de ejecución de un programa
- 3 tipos:
  - Instrucciones secuenciales
  - Instrucciones condicionales o alternativas
    - Alternativa simple (SI)
    - Alternativa doble (SI… SI NO…)
    - Alternativa múltiple (SEGUN_VALOR)
  - Instrucciones repetitivas
    - Instrucción MIENTRAS
    - Instrucción REPETIR… MIENTRAS
    - Instrucción PARA

Instrucciones secuenciales

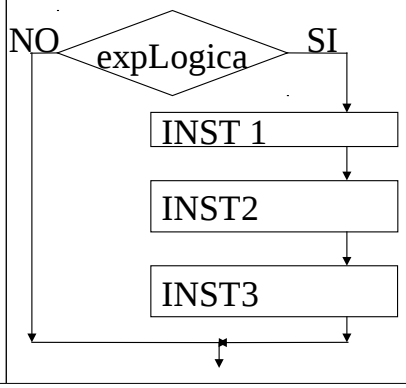
El orden en que se ejecutan las instrucciones es una detrás de otra

Pseudocódigo	Diagrama de Flujo
INST1 INST2 INST3 …	

Instrucciones condicionales

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

ALTERNATIVA SIMPLE

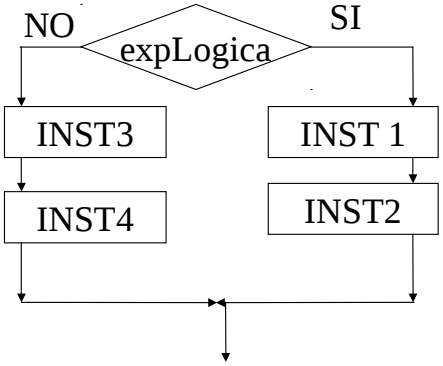
Pseudocódigo	Diagrama de Flujo
SI expLogica **INST1** **INST2** **INST3** **…** **FIN SI**	

Alternativa doble

Psedocódigo

```
SI  expLogica
      INST1
      INST2
SI NO
     INST3
     INST4
FIN SI
```

Diagrama de Flujo



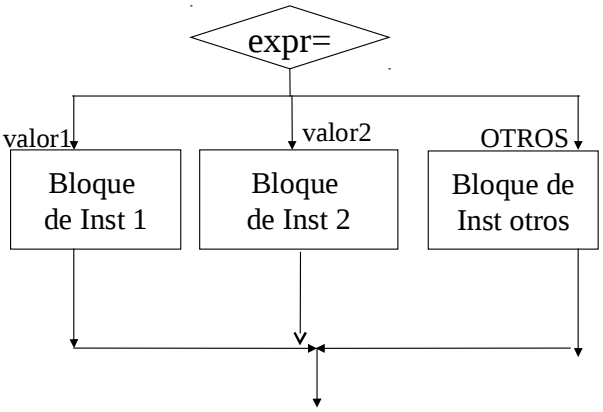
Alternativa múltiple

Pseudocódigo

```
SEGUN_VALOR expr
   valor1:
      Bloque de Inst 1
   valor1:
      Bloque de Inst 2
   ....
   OTROS:
      Bloque de Inst otros

FIN SEGUN_VALOR
```

Diagrama de Flujo



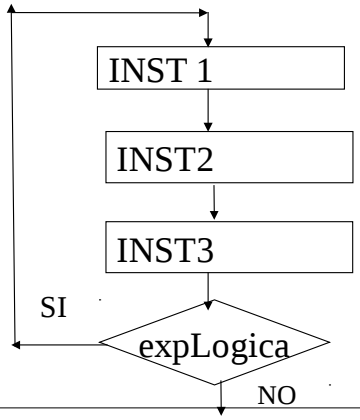
Instrucciones repetitivas

REPETIR MIENTRAS: Repetir de 1 a n veces. Permite que un conjunto de acciones se ejecuten más de una vez en función de que se cumpla una determinada condición (expresión lógica)

Pseudocódigo

```
REPETIR
   INST1
   INST2
   INST3
   …
MIENTRAS expLogica
```

Diagrama de Flujo



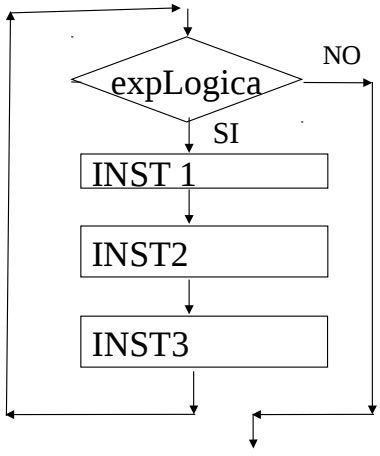
Instrucciones repetitivas

MIENTRAS: Repetir de 0 a n veces

Pseudocódigo

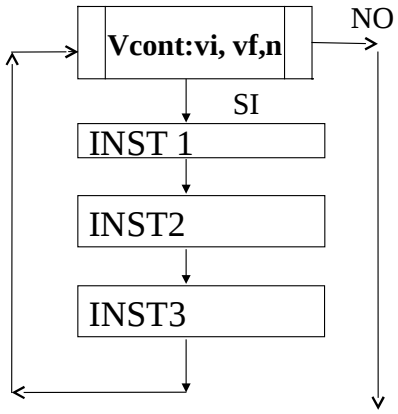
```
MIENTRAS  expLogica
   INST1
   INST2
   INST3
   …
FIN MIENTRAS
```

Diagrama de Flujo



Instrucciones repetitivas

PARA: Repetir de un número de veces fijado

Pseudocódigo	Diagrama de Flujo
PARA vcont **DE** vi **A** vf **CON INC**= num INST1 INST2 INST3 … **FIN PARA**	

Instrucciones repetitivas

PARA: Repetir de un número de veces fijado

- vcont es una variable entera que 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

9. 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 para recordar la ocurrencia de un determinado suceso dentro del programa