

# Unidad 2\_ anexo

## ESTRUCTURAS DE PROGRAMACIÓN

1

## Objetivos

- Conocer y saber usar las tres estructuras básicas de la programación estructurada.
- Interpretar y construir el esquema del flujo de control de las distintas estructuras de programación.
- Correcta composición y escritura de las expresiones lógicas que sirven como condiciones.
- Diferenciar entre procesos que se excluyen y procesos dependientes.
- Aprender a construir bucles correctamente.
- Conocimiento y uso adecuado de las variables de control de proceso (VCB).
- Conocer, distinguir, y saber usar las distintas variables de trabajo, contadores, acumuladores, interruptores, así como los datos centinelas utilizados en la construcción de algoritmos.
- Realización de procesos de lectura en número determinado e indeterminado.
- Realizar las pruebas necesarias mediante revisiones de código y traza para detectar errores en las etapas mas tempranas del diseño y asegurar su correcto funcionamiento.
- Construir algoritmos usando la estructura más adecuada en cada caso.
- Saber interpretar algoritmos sencillos ya resueltos.

2

# Contenidos

- Estructura de control Secuencial
- Estructura de control Alternativa
  - SI\_En\_Otro\_Caso
  - SI Anidados
  - SI\_En\_Otro\_Caso\_SI
- Estructura de control Repetitiva
- Bucles Mientras
- Contadores y Acumuladores
- Centinelas
- Flags
- Tipos de bucles
- Subtareas Iterativas
- Como diseñar bucles
- Estructuras de Selección múltiple SEGÚN
- Sintaxis
- Aplicación procesos controlados por MENÚS
- Otras estructuras repetitivas
  - Bucles REPETIR
  - Bucles PARA
- Criterios para la elección de una estructura repetitiva.
- Anexo. Números Aleatorios

3

# Estructura de control secuencial

Cuando un proceso directo, se puede realizar con la información que recibe sin necesidad de desviar los pasos y de una única vez, el proceso es susceptible de ser ejecutado en una operación o, como máximo, en una secuencia de operaciones más sencillas que compongan la acción única total.

Normalmente, en programación, las acciones de los procesos secuenciales se suelen escribir una debajo de la otra acabando, en la mayoría de los lenguajes de programación, cada acción con un signo de punto y coma ‘;’.

```
...  
acción 1;  
acción que sigue a la 1;  
debo ser la acción 3;  
...
```

**Por ejemplo, tengo dos números y quiero obtener su media:**

```
se tienen dos números y se quiere obtener la media  
sumar los dos números guardando el resultado;  
tomar el resultado guardado antes y dividirlo por 2 guardando el resultado  
poner el resultado anterior donde sea conveniente;
```

**Algoritmo para abrir un envase TetraBrick c para diestros**

```
Tomar y mantener el envase con la mano izquierda;  
Aplastar haciendo presión la esquina superior donde viene el dibujito de las tijeras;  
Doblar el cartón hacia un lado por la línea punteada;  
Enderezar de nuevo el cartón;  
Rasgar el cartón por la línea punteada
```

4

## Estructura de control alternativa, condicional, de selección o de bifurcación

**SI (expresión) [entonces]**

**instrucción 1**

**[EN\_OTRO\_CASO o SINO]**

**instrucción 2**

**FINSI**

*Anidar estructuras* es: ejecutar una estructura dentro de otra del mismo tipo.  
Deberemos controlar la legibilidad del código

EJEMPLO: LEER UN NÚMERO ENTERO DE TECLADO  
Y PRESENTAR EN PANTALLA SU VALOR ABSOLUTO.

**INICIO**

```
Leer numero del teclado
Si (numero < 0) entonces
    Valor_absoluto = - numero
en otro caso
    Valor_absoluto = numero
fin
Escribir en pantalla Valor_absoluto
```

**FIN**

**ENTORNO**

entero numero, Valor\_absoluto

**ALGORITMO GENERALIZADO**

**INICIO**

```
LEER(numero)
CALCULAR(numero, Valor_absoluto)
ESCRIBIR(Valor_absoluto)
```

**FIN**

5

## Estructuras repetitivas, cíclicas o iterativas

- Existen muchos procesos en los que **se deben ejecutar** acciones **mientras** no se deje de dar una situación o **hasta** que se alcance un estado determinado.
- Ejemplos: “caminar hasta alcanzar la acera” o “mientras no se alcance la acera”, “sacar monedas mientras no se rebase o iguale la cantidad solicitada”; etc.
- Los ordenadores son capaces de ejecutar bucles, esto es, repetir un juego de acciones hasta que se deje de cumplir o se cumpla una condición predeterminada.

**MIENTRAS (expresión)**

**instrucciones**

**FIN\_MIENTRAS**

**REPETIR**

**instrucciones**

**MIENTRAS(expresión)**

**Deberemos** inicializar, evaluar y actualizar la **VCB** (Variable de Control de Bucle), estos requisitos deben ser tenidos en cuenta a la hora de diseñar cualquier estructura repetitiva.

6

## Tipos de bucles

Presenta en pantalla la suma de los números pares comprendidos entre 0 y 100

### ■ CONTROLADOS POR CONTADOR

```
Entorno
Variables
    entero i = 0, suma = 0
Constantes
    MAX = 100
Inicio
    i = 2
    mientras (i < MAX)
        suma = suma + i
        i = i + 2
    finmientras
    escribir (suma)
fin
```

### ■ CONTROLADOS POR SUCESOS

```
Entorno
Variables
    entero i = 0, suma = 0
    booleana salir = falso
Constantes
    MAX = 100
Inicio
    i = 2
    mientras (salir = = falso)
        suma = suma + i
        i = i + 2
        si (i > = MAX)
            salir = verdadero
        fin si
    finmientras
    escribir (suma)
fin
```

modificar estos códigos utilizando la instrucción repetir-mientras y analizar diferencias

7

## Tipos de bucles

Presenta en pantalla la suma de los números pares introducidos por teclado. El programa acaba cuando se introduzca un número impar.

### ■ REPETIR-MIENTRAS

```
Entorno
Variables
    entero i = 0, suma = 0
Inicio
    repetir
        suma = suma + entero
        leer (entero)
    mientras (entero MOD 2 == 0)
        escribir (suma)
fin
```

### ■ MIENTRAS

```
Entorno
Variables
    entero i = 0, suma = 0
    booleana salir = falso
Inicio
    leer (entero)
    mientras (entero MOD 2 == 0)
        suma = suma + entero
        leer (entero)
    finmientras
    escribir (suma)
fin
```

8

## Datos asociados a las estructuras

### ■ CONTADORES

- Se incrementa o decrementa de forma continua
- Deben ser inicializados
- Ej: i

### ■ ACUMULADORES

- Se incrementa o decrementa de forma variable
- Deben inicializarse, tanto los sumativos (0) como multiplicativos (1)
- Ej: suma y factorial de un número

### ■ DATOS CENTINELAS

- Es un dato que no pertenece al rango de valores permitidos y por esto es por lo que se puede usar para alertarnos de una situación determinada.
- Ej: leer números positivos hasta -1. o introducir nombres de asignaturas hasta "FIN"

### ■ SWITCHES, FLAGS O INDICADORES

- Son variables booleanas que alteran el flujo del programa.
- Ej: adivinar un número entre 0 y 20 en 5 intentos.

9

## Cómo diseñar un bucle

### ■ Invariante de bucle:

- Son las condiciones que deben darse al comienzo de cada iteración del bucle.
- Diseño
  - ¿cuál es la condición de salida del bucle? ¿y la de entrada?
  - ¿cuál debe ser el valor inicial y cómo se actualiza?
  - ¿Qué parte del proceso debe repetirse dentro del bucle?
- Analizar y completar los siguientes ejemplos:

#### Ejemplo 1

```
....  
i = 91  
mientras (i > 90)  
    escribir (i)  
    Leer (i)  
Fin mientras  
....
```

#### Ejemplo 2

```
....  
i = 6, sum = 0  
mientras (i > sum)  
    sum = sum + i  
    i = i + 2  
Fin mientras  
....
```

#### Ejemplo 3

```
....  
error = falso  
i = 0  
mientras (error == falso)  
    leer (sexo)  
    si (sexo == )  
        i = i + 1  
    sino  
        error = verdadero  
    fin si  
fin si  
Fin mientras  
....
```

10

## Otras estructuras repetitivas

### REPETIR

1.

**Repetir**

**Instrucciones**

**Mientras (Expresión)**

2.

**Repetir**

**Instrucciones**

**Hasta (Expresión)**

- Útiles para validar datos de entrada.
- Dependerán del lenguaje.

### PARA

**Para (Expresión1, Mientras Expresión2, Expresión3)**

**Instrucciones**

**Fin\_Para**

para (i = 100, mientras (i > 50), i = i - 4)  
escribir (i)

finpara

11

## Estructura de selección múltiple

- Permiten seleccionar varias acciones alternativas y elegir una en tiempo de ejecución.
- Aplicable a procesos controlados por menú.

**según (expresión) [hacer]**

**para** expresión = constante 1  
instrucciones A

**para** expresión = constante 2  
instrucciones B

....

[en otro caso]  
instrucciones X

**finsegún**

- Después de ejecutar las instrucciones correspondientes, se pasa a la siguiente después de finsegún.

PROGRAMA PRINCIPAL GENERALIZADO

Inicio

Presentar MENU

Elegir y *validar* Opción

Mientras no quiera finalizar (Opcion != 'F')

Obtener números /\*Inicializa y actualiza el proceso\*/

Según (Opcion)

Para Opcion = 'S'

Realizar SUMAR

Para Opcion = 'R'

Realizar RESTAR

Para Opcion = 'M'

Realizar MULTIPLICAR

Para Opcion = 'D'

Validar Numero2 y DIVIDIR

Fin Según

Presentar resultados

Presentar MENU

Elegir y *validar* Opcion

Fin\_mientras

FinPP

12

“Las cosas se aprenden haciéndolas.”

Aristóteles