

Tema 3. Estructuras de Control

Objetivos

- ❑ Estudiar las estructuras básicas de control (secuenciales, condicionales y repetitivas)
- ❑ Aprender a diseñar algoritmos utilizando el pseudocódigo y organigramas (diagramas de flujo) como notación para describirlos
- ❑ Reconocer la importancia de diseñar algoritmos como paso previo a la resolución de problemas mediante la programación
- ❑ Resolver los primeros problemas a través de técnicas de programación básicas

Tema 3. Estructuras de control

1. Estructuras secuenciales
 1. Asignación
 2. Lectura / Entrada
 3. Escritura / Salida
 4. Ejemplos
2. Estructuras condicionales
 1. Simple
 2. Anidada
 3. Múltiple
 4. Ejemplos
3. Estructuras repetitivas
 1. Bucle MIENTRAS
 2. Bucle REPETIR-HASTA
 3. Bucle PARA
 4. Bucles infinitos y bucles anidados
 5. Ejemplos

Tipos de estructuras de control

- Estructuras secuenciales
 - Sentencias que se realizan una a continuación de otra en el orden en el que están escritas
- Estructuras Condicionales
 - Sentencias que dependiendo de la evaluación de una expresión lógica (una condición), se realiza una opción u otra
- Estructuras Repetitivas o Iterativas (Bucles o Ciclos)
 - Sentencias que permiten la repetición de un grupo de acciones
 - Las acciones incluidas en un bucle se pueden repetir un número fijo de veces o bien un número variable determinado por la evaluación de una expresión lógica (una condición)

Estructuras secuenciales - Asignación

- Permite asignar un valor determinado a una variable

var ← exp



- El valor de la variable "var" será el resultado de evaluar la expresión "exp"
- El tipo de la variable implicada en la operación de asignación debe ser compatible con el tipo del resultado de evaluar la expresión
- La asignación es una operación **destructiva**
- Ejemplos:

```
a ← 3.0
b ← (2.0*c+4)/8
c ← (d+2)<7
menor_ej1 ← (ej1<ej2) AND (ej1<ej3)
hipotenusa ← SQRT(cat1*cat1+cat2*cat2)
```

Ejemplos

```
A ← 3
B ← 5
C ← 2
A ← B
B ← C
C ← A*B
A ← B
```

```
A = 2
B = 2
C = 10
```

```
A ← 30
B ← 10
C ← 7
A ← A DIV C
B ← A*3 DIV C+2
C ← A DIV B DIV C
```

```
A = 4
B = 3
C = 0
```

```
A ← 10
B ← 5*A
C ← B*2
A ← C MOD 3
B ← C DIV 3
C ← A+2*2
```

```
A = 1
B = 33
C = 5
```

```
A ← 13
B ← 15
C ← 12
A ← (C-B)*(B-A)
B ← A
C ← A*B
```

```
A = -6
B = -6
C = 36
```

```
A ← 8
B ← A*(3-A)
C ← A-B
A ← B MOD A
B ← -C-(B-A)
C ← A*B
```

```
A = 0
B = -8
C = 0
```

Ejemplo

- ¿En cuáles de los siguientes pares es importante el orden de los enunciados? Es decir, si se modifica el orden ¿cambian los resultados finales? (Suponemos que $X \neq Y \neq Z$)

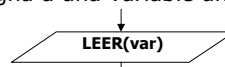
a) $X \leftarrow Y$ b) $X \leftarrow Z$
 $Y \leftarrow Z$ $X \leftarrow Y$

c) $X \leftarrow Y$ d) $Z \leftarrow Y$
 $Z \leftarrow X$ $X \leftarrow Y$

Estructuras secuenciales – Lectura/Entrada

- Mediante la operación de entrada se asigna a una variable un valor dado desde el exterior

LEER(var)



- Se utiliza para que el usuario pueda introducir los datos requeridos por un algoritmo
- El tipo de dato suministrado desde el exterior debe ser compatible con el tipo de la variable "var"
- Tras la operación de lectura, la variable "var" contiene el valor dado por el usuario a través de un dispositivo de entrada

LEER(cateto1)

- La operación de lectura es destructiva
- Es posible leer los valores de varias variables en una misma operación de entrada (separando con comas)

LEER(nota1, nota2)

es equivalente a las siguientes dos sentencias

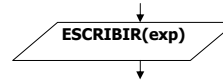
LEER(nota1)

LEER(nota2)

Estructuras secuenciales – Escritura/Salida

- Mediante la operación de salida se transfiere el valor de una expresión a un dispositivo de salida

ESCRIBIR(exp)



- Se utiliza para mostrar al usuario los resultados obtenidos
- Ejemplos:

```
ESCRIBIR(hipotenusa)
ESCRIBIR(b+3)
```

- Se suele utilizar también para mostrar mensajes informativos al usuario que estarán delimitados por comillas

```
ESCRIBIR("Hola")
```

- También es posible, combinar expresiones y cadenas de caracteres encerradas entre comillas en una misma operación de salida siempre que se separen por comas

```
ESCRIBIR("La media de las notas es ",media)
```

Estructura básica de un algoritmo

ALGORITMO nombre.

Entrada

Salida

Proceso (Descripción)

CONSTANTES

```
id_cte1 = valor1
```

```
.....
```

```
id_cteN = valorN
```

VARIABLES

```
id_var1 : tipo1
```

```
.....
```

```
id_varM : tipoM
```

1. [Inicio]

```
.....
```

Sentencias

```
.....
```

F. [Fin]

ALGORITMO triangulo.

Entrada: cat1 y cat2

Salida: hipotenusa

Proceso: Calcula la hipotenusa de un triángulo rectángulo tomando como entrada el valor de los dos catetos

VARIABLES

```
cat1, cat2 : REAL
```

```
hipotenusa : REAL
```

1. [Entrada de datos]

```
ESCRIBIR('Introduzca cateto 1: ')
LEER(cat1)
```

```
ESCRIBIR('Introduzca cateto 2: ')
LEER(cat2)
```

2. [Cálculo de la hipotenusa]

```
hipotenusa ← SQRT(cat1*cat1+cat2*cat2)
```

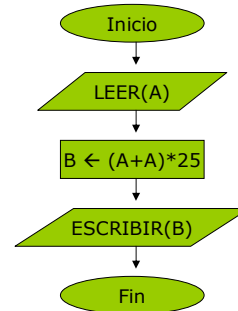
3. [Mostrar el resultado]

```
ESCRIBIR('La hipotenusa es',hipotenusa)
```

4. [Fin]

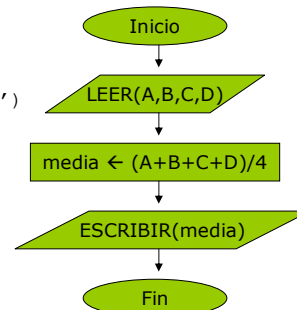
Escribir un algoritmo para leer un valor entero, doblarlo, multiplicarlo por 25 y visualizar el resultado

```
ALGORITMO operacionesbasicas.  
VARIABLES  
  A,B: ENTERO  
1.[Entrada de un valor entero]  
  LEER(A)  
2.[Realizar las operaciones]  
  B ← (A+A)*25  
3.[Mostrar el resultado]  
  ESCRIBIR('El resultado es ',B)  
4.[Fin]
```



Diseñar un algoritmo que lea cuatro calificaciones de un alumno y calcule la calificación promedio

```
ALGORITMO media_calificacion.  
VARIABLES  
  A,B,C,D,media: REAL  
1. [Entrada de datos]  
  ESCRIBIR('Introduzca las 4 calificaciones')  
  LEER(A,B,C,D)  
2. [Calcular el promedio]  
  media ← (A+B+C+D)/4  
3. [Mostrar el resultado]  
  ESCRIBIR('La media es ',media)  
4. [Fin]
```



Realizar un algoritmo para pasar de grados Centígrados (°C) a grados Fahrenheit (°F) sabiendo que $^{\circ}\text{F} = 9/5 \text{ }^{\circ}\text{C} + 32$

ALGORITMO grados.

VARIABLES

C,F: REAL

- [Entrada de los grados centígrados]
ESCRIBIR('Introduzca los grados centígrados')
LEER(C)
- [Calcular los grados Fahrenheit]
 $F \leftarrow 9/5 * C + 32$
- [Mostrar el resultado]
ESCRIBIR(C, ' grados centígrados son ', F, ' grados Fahrenheit')
- [Fin]

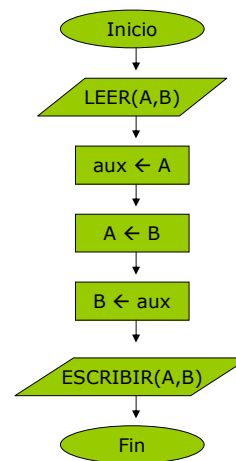
Realizar un algoritmo para intercambiar los valores de dos variables

ALGORITMO intercambio_var.

VARIABLES

A,B,aux: ENTERO

- [Entrada del valor de las variables]
LEER(A,B)
- [Intercambiar las variables]
 $\text{aux} \leftarrow A$
 $A \leftarrow B$
 $B \leftarrow \text{aux}$
- [Mostrar el resultado]
ESCRIBIR('El valor de A es ', A)
ESCRIBIR('El valor de B es ', B)
- [Fin]



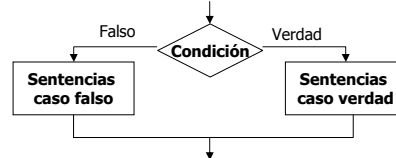
Seminario 1

- Estructuras secuenciales
- Jueves 11 de octubre de 8'30 a 9'30 (grupo A), y de 15'30 a 16'30 (grupo B)
- Relación de ejercicios del tema 3
 - Ejercicios 9, 10, 15 y 16

Estructuras condicionales

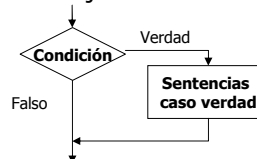
- En una estructura condicional se evalúa una expresión lógica y dependiendo del resultado se ejecutan unas sentencias u otras

SI condición ENTONCES
Sentencias en caso de verdad
SI_NO
Sentencias en caso de falso
FIN_SI



- Si la condición resulta verdadera sólo se realizan las operaciones correspondientes al ENTONCES y si la condición resulta falsa sólo se realizan las del SI_NO. En ningún caso se ejecutarán ambos grupos de sentencias

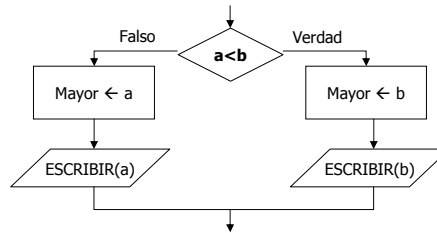
□ La parte SI_NO es opcional
SI condición ENTONCES
Sentencias en caso de verdad
FIN_SI



Estructuras condicionales

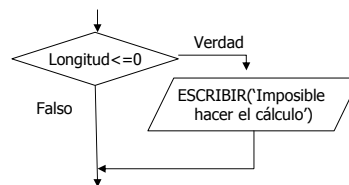
□ SI-ENTONCES-SINO

```
SI (a<b) ENTONCES
  Mayor ← b
  ESCRIBIR('El valor mayor es b',b)
SI_NO
  Mayor ← a
  ESCRIBIR('El valor mayor es a',a)
FIN_SI
```



□ SI-ENTONCES

```
SI (longitud<=0) ENTONCES
  ESCRIBIR('Imposible hacer el cálculo.
  La longitud debe ser un número
  positivo')
FIN_SI
```



Condiciones

□ Una condición o expresión booleana puede ser:

- Una variable booleana

TEST ← T

- Una expresión aritmética seguida de un operador relacional, seguido de otra expresión aritmética

TEST ← 3<5

- Una expresión booleana, seguida de un operador booleano, seguido de una expresión booleana

TEST ← A AND NOT B OR C

□ Se pueden utilizar condiciones tan complejas como se quiera siempre que estén bien formadas

Estructuras condicionales anidadas

- Se pueden anidar varias sentencias condicionales para formar condiciones complejas

```

SI cond1 ENTONCES
  SI cond2 ENTONCES
    sentencias1
  SI_NO
    sentencias2
  FIN_SI
SI_NO
  SI cond3 ENTONCES
    sentencias3
  SI_NO
    sentencias4
  FIN_SI
FIN_SI
    
```

Encontrar el mayor de 3 números

```

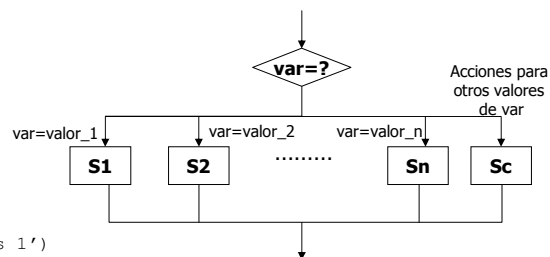
SI (a<b) ENTONCES
  SI (b<c) ENTONCES
    Mayor ← c
  SI_NO
    Mayor ← b
  FIN_SI
SI_NO
  SI (a<c) ENTONCES
    Mayor ← c
  SI_NO
    Mayor ← a
  FIN_SI
FIN_SI
    
```

Estructuras condicionales múltiples

- La sentencia multicondicional CASO se puede ver como un estructura condicional simple, pero en lugar de haber dos alternativas, ahora hay múltiples alternativas

```

CASO var
  Valor_1: S1
  Valor_2: S2
  ...
  Valor_n: Sn
EN OTRO CASO Sc
FIN_CASO
    
```



```

CASO num
  1: ESCRIBIR('El número es 1')
  2: ESCRIBIR('El número es 2')
  3,4: ESCRIBIR('El número es 3 o 4')
EN OTRO CASO ESCRIBIR('Entrada no válida')
FIN_CASO
    
```

Ejemplo. Determinar el mayor de 4 números (con condiciones simples)

```
SI (A<B) ENTONCES
  SI (B<C) ENTONCES
    SI (C<D) ENTONCES
      mayor ← D
    SI_NO
      mayor ← C
  FIN_SI
SI_NO
  SI (B<D) ENTONCES
    mayor ← D
  SI_NO
    mayor ← B
  FIN_SI
FIN_SI
SI_NO
...
SI (A<C) ENTONCES
  SI (C<D) ENTONCES
    mayor ← D
  SI_NO
    mayor ← C
  FIN_SI
SI_NO
  SI (A<D) ENTONCES
    mayor ← D
  SI_NO
    mayor ← A
  FIN_SI
FIN_SI
SI_NO
FIN_SI
```

Ejemplo. Determinar el mayor de 4 números (con condiciones compuestas)

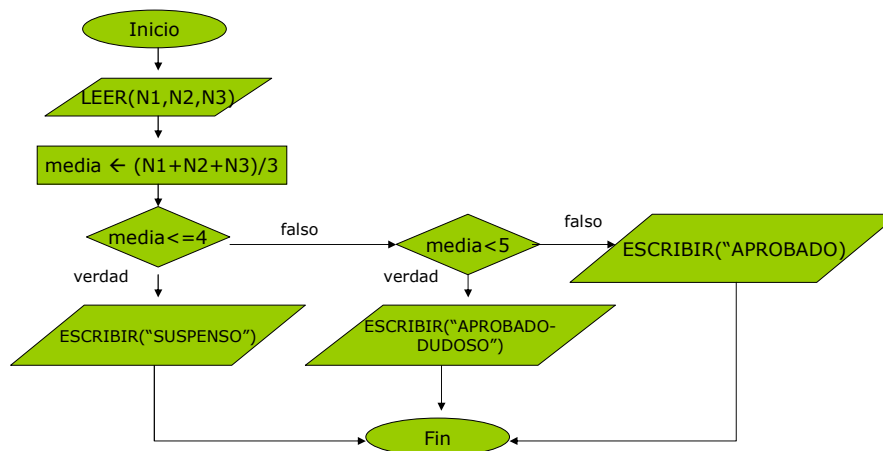
```
SI (A>B) AND (A>C) AND (A>D) ENTONCES
  mayor ← A
FIN_SI
SI (B>A) AND (B>C) AND (B>D) ENTONCES
  mayor ← B
FIN_SI
SI (C>A) AND (C>B) AND (C>D) ENTONCES
  mayor ← C
FIN_SI
SI (D>A) AND (D>B) AND (D>C) ENTONCES
  mayor ← D
FIN_SI
```

Diseñar un algoritmo que lea tres puntuaciones de un alumno y calcule la media. Si la media del alumno no supera los 4 puntos el alumno está suspenso y el algoritmo deberá imprimir un mensaje indicándolo. En caso de que el alumno tenga una media superior o igual a 4 puntos pero inferior a 5, el alumno estará 'aprobado pero dudoso'. Si el alumno supera los 5 puntos el mensaje a imprimir será 'Aprobado'

```

ALGORITMO notas.
VARIABLES
  N1,N2,N3,media: REAL
1. [Entrada de datos]
  ESCRIBIR('Introduzca las 3 notas: ')
  LEER(N1,N2,N3)
2. [Calcular la media]
  media ← (N1+N2+N3)/3
3.[Comprobar la calificación final]
  SI (media<=4) ENTONCES
    ESCRIBIR('SUSPENSO')
  SI_NO
    SI (media<5) ENTONCES
      ESCRIBIR('APROBADO PERO DUDOSO')
    SI_NO
      ESCRIBIR('APROBADO')
  FIN_SI
  FIN_SI
4.[Fin]
  
```

Diseñar un algoritmo que lea tres puntuaciones de un alumno y calcule la media. Si la media del alumno no supera los 4 puntos el alumno está suspenso y el algoritmo deberá imprimir un mensaje indicándolo. En caso de que el alumno tenga una media superior o igual a 4 puntos pero inferior a 5, el alumno estará 'aprobado pero dudoso'. Si el alumno supera los 5 puntos el mensaje a imprimir será 'Aprobado'



Seminario 2

- Estructuras condicionales
- Jueves 25 de octubre de 8'30 a 9'30 (grupo A) y de 15'30 a 16'30 (grupo B)
- Relación de ejercicios del tema 3
 - Ejercicios 18, 26 y 27

Estructuras repetitivas

- **Ámbito o Cuerpo del Bucle:** sentencias que se repiten durante la ejecución del bucle
- **Tipos de bucles:**
 - Bucles controlados por contador
 - El cuerpo del bucle se repite un número fijo de iteraciones dependiendo de una variable de control
 - Bucles controlados por condición
 - El número de veces que se ejecuta el bucle depende de una condición
- **Estructuras repetitivas**
 - Bucle MIENTRAS
 - Bucle REPETIR-HASTA
 - Bucle PARA

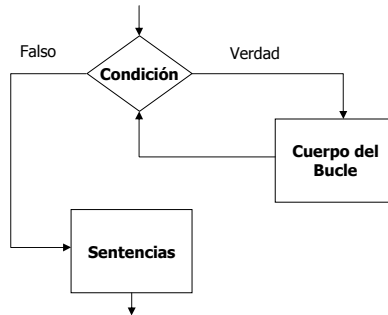
Bucle MIENTRAS

MIENTRAS condición HACER

Sentencias del bucle (cuerpo del bucle)

FIN_MIENTRAS

- El cuerpo del bucle se repite **mientras** la condición sea verdadera



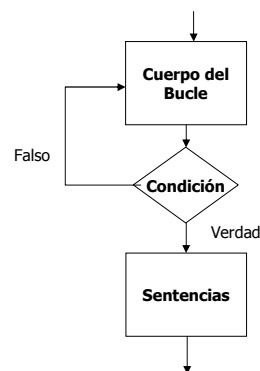
Bucle REPETIR-HASTA

REPETIR

Cuerpo del bucle

HASTA condición

- El cuerpo del bucle se repite **hasta** que la condición sea verdadera (es decir, mientras la condición sea falsa)
- El cuerpo del bucle siempre se ejecuta al menos una vez



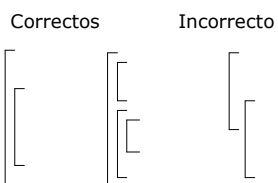
Bucles infinitos

- ❑ La condición de terminación nunca se cumple
- ❑ Debe haber alguna sentencia dentro del cuerpo del bucle que haga que el valor de la condición que controla el bucle varíe

```
i ← 1
MIENTRAS (i>0) HACER
  i ← i+1
  i ← i-1
FIN_MIENTRAS
ESCRIBIR('El valor de i es ',i)
```

Bucles anidados

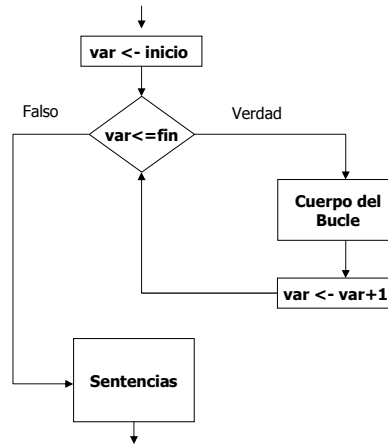
- ❑ El cuerpo de un bucle puede incluir cualquier tipo de sentencias, incluidas otras sentencias repetitivas
- ❑ La estructura interna debe estar completamente dentro de la estructura externa, no pudiendo existir solapamientos



Bucle PARA

PARA var=inicio HASTA fin HACER
Cuerpo del bucle
FIN_PARA

- El cuerpo del bucle se repite un **número fijo** de iteraciones
- Variable de control
 - Controla el número de iteraciones
 - Se le asignan automáticamente valores sucesivos
 - Al comienzo del bucle se debe especificar su valor inicial y su valor final
 - Debe ser de tipo entero, carácter o enumerado y su valor NO puede ser modificado dentro del bucle
 - Cuando el bucle termina, su valor es indeterminado



Equivalencia PARA - MIENTRAS

PARA var=inicio HASTA fin HACER
Sentencias
FIN_PARA



var ← inicio
MIENTRAS (var<=fin) HACER
Sentencias
var ← var + 1
FIN_MIENTRAS

Ejemplo: Factorial

```
ALGORITMO factorial_para.  
Entrada: N  
Salida: fact  
Proceso: calcula el factorial  
de N usando un bucle PARA  
VARIABLES  
N, i, fact: ENTERO  
1. [Entrada del número N]  
LEER(N)  
2. [Calcular el factorial]  
fact ← 1  
PARA i=1 HASTA N HACER  
    fact ← fact * i  
FIN_PARA  
3. [Mostrar el resultado]  
ESCRIBIR('Factorial de ',N,'es',fact)  
4. [Fin]
```

```
ALGORITMO factorial_mientras.  
Entrada: N  
Salida: fact  
Proceso: calcula el factorial de N  
usando un bucle MIENTRAS  
VARIABLES  
N, i, fact: ENTERO  
1. [Entrada del número N]  
LEER(N)  
2. [Calcular el factorial]  
fact ← 1  
i ← 1  
MIENTRAS (i<=N) HACER  
    fact ← fact * i  
    i ← i + 1  
FIN_MIENTRAS  
3. [Mostrar el resultado]  
ESCRIBIR('Factorial de ',N,'es',fact)  
4. [Fin]
```

Validación de datos de entrada

```
ALGORITMO validacion_repetir.  
Entrada: nota  
Salida: nota  
Proceso: Validación de los datos  
de entrada con un bucle REPETIR-  
HASTA  
VARIABLES  
nota: REAL  
1. [Entrada de datos]  
REPETIR  
    ESCRIBIR('Introduzca la nota: ')  
    LEER(nota)  
HASTA (nota>=0.0) AND (nota<=10.0)  
2. [Salida de datos]  
ESCRIBIR('La nota es ', nota)  
3. [Fin]
```

```
ALGORITMO validacion_mientras.  
Entrada: nota  
Salida: nota  
Proceso: Validación de los datos de  
entrada con un bucle MIENTRAS  
VARIABLES  
nota: REAL  
1. [Entrada de datos]  
ESCRIBIR('Introduzca la nota: ')  
LEER(nota)  
MIENTRAS (nota<0.0) OR (nota >10.0) HACER  
    ESCRIBIR('La nota debe ser un n° real  
entre 0 y 10. Inténtelo de nuevo: ')  
    LEER(nota)  
FIN_MIENTRAS  
2. [Salida de datos]  
ESCRIBIR('La nota es ', nota)  
3. [Fin]
```

Fin de ejecución controlada por el usuario

```
ALGORITMO repeticion.  
  Entrada: respuesta  
  Salida:  
  Proceso: repite la ejecución de un algoritmo hasta que el usuario pulse  
           la letra 'n'  
VARIABLES  
  respuesta: CARACTER  
1. [Inicio]  
  REPETIR  
  
    Desarrollo del algoritmo  
  
    ESCRIBIR('Desea volver a ejecutar el programa (S/N)? ')  
    LEER(respuesta)  
    HASTA (respuesta='N') OR (respuesta='n')  
F. [Fin]
```

Indicar en cada uno de los siguientes segmentos de algoritmos cuándo el bucle termina y cuándo no lo hace. En este último caso señálense las razones. Supóngase que todas las variables son enteras

```
a) contador ← 0  
   total ← 0  
   MIENTRAS contador >= 0 HACER  
     total ← total + 2  
   FIN_MIENTRAS  
  
b) contador ← 0  
   total ← 0  
   MIENTRAS contador <= 0 HACER  
     total ← total + 2  
     contador ← contador + 2  
   FIN_MIENTRAS  
  
c) contador ← 0  
   total ← 0  
   MIENTRAS contador <= 10 HACER  
     total ← total + 2  
     contador ← contador + 1  
   FIN_MIENTRAS
```

Indíquese el valor que se imprime para VAR. Supóngase que todas las variables son enteras

- a) var ← 0
PARA num=1 HASTA 10 HACER
 var ← var + 1
FIN_PARA
ESCRIBIR (var)
- b) var ← 0
PARA num=4 HASTA 16 HACER
 var ← var + 1
FIN_PARA
ESCRIBIR (var)
- c) var ← 0
PARA num=1 HASTA 5 HACER
 PARA mult=1 HASTA 8 HACER
 var ← var + 1
 FIN_PARA
FIN_PARA
ESCRIBIR (var)

Indíquese el valor que se imprime para cada uno de los siguientes segmentos de algoritmos

```
PARA i=1 HASTA 10 HACER  
    ESCRIBIR (i)  
FIN_PARA
```

```
PARA i=1 HASTA 10 HACER  
    ESCRIBIR (i+i)  
FIN_PARA
```

```
PARA i=1 HASTA 10 HACER  
    ESCRIBIR (2*i)  
FIN_PARA
```

```
PARA i=1 HASTA 10 HACER  
    ESCRIBIR (3*i)  
FIN_PARA
```

Indíquese el valor que se imprime para cada uno de los siguientes segmentos de algoritmos

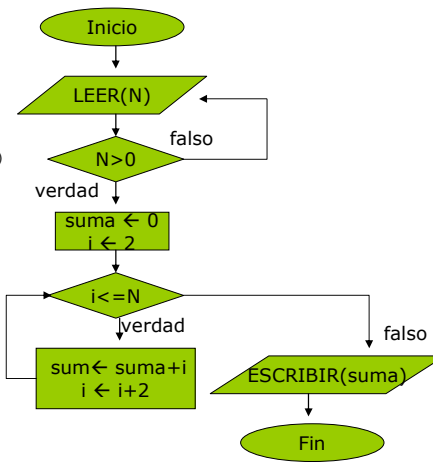
```
LEER(n)
PARA i=1 HASTA 10 HACER
    ESCRIBIR (n*i)
FIN_PARA
```

```
S ← 0
PARA i=1 HASTA 10 HACER
    S ← i
FIN_PARA
ESCRIBIR (S)
```

```
S ← 0
PARA i=1 HASTA 10 HACER
    S ← S +1
FIN_PARA
ESCRIBIR (S)
```

Diseñar un algoritmo para calcular la suma de los N primeros pares

```
ALGORITMO suma_pares.
VARIABLES
    i,n,suma:ENTERO
1.[Entrada de datos]
    REPETIR
        ESCRIBIR('Introduzca un entero:')
        LEER(N)
    HASTA (N>0)
2.[Sumar los n primeros pares]
    suma ← 0
    i ← 2
    MIENTRAS (i<=2*n) HACER
        suma ← suma + i
        i ← i+2
    FIN_MIENTRAS
    ESCRIBIR('La suma es',suma)
3.[Fin]
```



Diseñar un algoritmo para calcular las N primeras potencias de 2

```
ALGORITMO potencias_2.
VARIABLES
  I,N,POT:ENTERO
1.[Entrada de datos]
  REPETIR
    ESCRIBIR('Introduzca un número entero:')
    LEER(N)
  HASTA (N>=0)
2.[Mostrar las primeras N potencias de 2]
  POT ← 1
  PARA I=0 HASTA N HACER
    ESCRIBIR('2 elevado a ',I,' es ',POT)
    POT ← POT*2
  FIN_PARA
3.[Fin]

¿Cómo cambiaría el algoritmo si se desea calcular Xy?
```

Diseñar un algoritmo que tome cada hora la temperatura exterior, leyéndola durante un periodo de 24 horas. Encontrar la temperatura media para el día, y la temperatura más alta y más baja

```
ALGORITMO temperatura.
VARIABLES
  media,temp,max,min:REAL
  i: ENTERO
1.[Inicializaciones]
  media ← 0
  max ← -50
  min ← 50
2.[Leer datos y calcular resultados]
  PARA i=1 HASTA 24 HACER
    REPETIR
      ESCRIBIR('Introduzca la temperatura:')
      LEER(temp)
    HASTA (temp>=50.0) AND (temp<50.0)
    media ← media + temp
    SI (temp>max) ENTONCES
      max ← temp
    FIN_SI
    SI (temp<min) ENTONCES
      min ← temp
    FIN_SI
  FIN_PARA
  media ← media / 24
3.[Mostrar resultados]
  ESCRIBIR('La media es ',media)
  ESCRIBIR('La máxima es ',max)
  ESCRIBIR('La mínima es ',min)
4.[Fin]
```

¿Cómo cambiaría el algoritmo si la lectura no tuviera un nº fijo de datos, y en su lugar, hubiera que leer datos hasta introducir el valor -100?

Realizar un algoritmo que lea desde teclado valores positivos y los vaya contando hasta que se introduzca un cero para terminar

```
ALGORITMO leer_positivos.  
VARIABLES  
  cont,num:ENTERO  
1.[Entrada de datos]  
  cont ← 0  
  REPETIR  
2.[Comprobar que el número es válido]  
  REPETIR  
    ESCRIBIR('Introduzca un n° positivo (cero para terminar):')  
    LEER(num)  
  HASTA (num>=0)  
3.[Incrementar contador si el n° es positivo]  
  SI (num<>0) ENTONCES  
    cont ← cont +1  
  FIN_SI  
4.[Terminar cuando se introduce 0]  
  HASTA (num=0)  
  ESCRIBIR('Se han leído ',cont,' números positivos')  
5.[Fin]
```


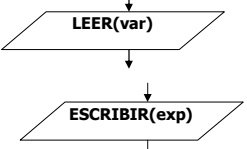
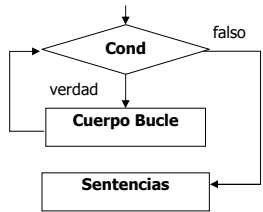
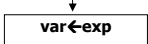
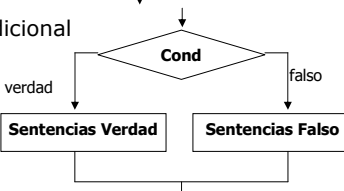
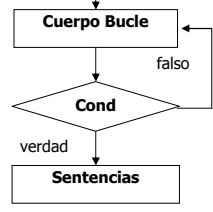
Seminario 3

- Estructuras repetitivas
- Jueves 8 de noviembre de 8'30 a 9'30 (grupo A) y de 15'30 a 16'30 (grupo B)
- Relación de ejercicios del tema 3
 - Ejercicios 37, 40, 41 y 43

Resumen Pseudocódigo

- | | |
|---|--|
| <ul style="list-style-type: none"> □ Estructuras secuenciales
Asignación: $var \leftarrow exp$
Entrada: LEER (var)
Salida: ESCRIBIR (exp) □ Estructura Condicional
SI condición ENTONCES
Sentencias en caso de verdad
SI_NO
Sentencias en caso de falso
FIN_SI | <ul style="list-style-type: none"> □ Estructuras Repetitivas
PARA var=inicio HASTA fin HACER
Cuerpo del bucle
FIN_PARA MIENTRAS condición HACER
Cuerpo del bucle
FIN_MIENTRAS REPETIR
Cuerpo del bucle
HASTA condición |
|---|--|

Resumen Diagramas de flujo

- | | |
|--|---|
| <ul style="list-style-type: none"> □ Inicio/Fin
 □ Entrada/Salida
 □ Bucle MIENTRAS
 | <ul style="list-style-type: none"> □ Secuencial
 □ Condicional
 □ Bucle REPETIR
 |
|--|---|

Resumen Codificación en C

□ Estructuras secuenciales

```
Asignación:  var = exp;
Entrada:     scanf("form",var);
Salida:      printf(exp);
```

□ Estructura Condicional

■ Condicional simple

```
if (condición) {
    Sentencias en caso de verdad
} else {
    Sentencias en caso de falso
}
```

■ Condicional múltiple

```
switch(var){
    case v1:
        sentencias
        break;
    ...
    case vn:
        sentencias
        break;
    default: sentencias
}
```

□ Estructuras Repetitivas

```
for( inicialización; condición;
    incremento){
    Cuerpo del bucle
}
```

```
while (condición) {
    Cuerpo del bucle
}
```

```
do{
    Cuerpo del bucle
}while (condición);
```