



# DISEÑO DE ALGORITMOS

## TEMA 2. ESTRUCTURAS DE CONTROL REPETITIVAS CON DIAGRAMAS DE FLUJO Y PSEUDOCÓDIGO

Presenta: Mtro. David Martínez Torres

Universidad Tecnológica de la Mixteca

Instituto de Computación

Oficina No. 37

[dtorres@gs.utm.mx](mailto:dtorres@gs.utm.mx)

# Contenido

1. Estructura de control **Mientras**
2. Estructura de control **Hacer ... Mientras**
3. Estructura de control **Para ... hasta ó Desde ... hasta**

# 1. Estructuras de control repetitivas

Estas estructuras se utilizan en algoritmos donde algunas instrucciones se deben ejecutar un número repetido de veces, llamado **ciclo**.

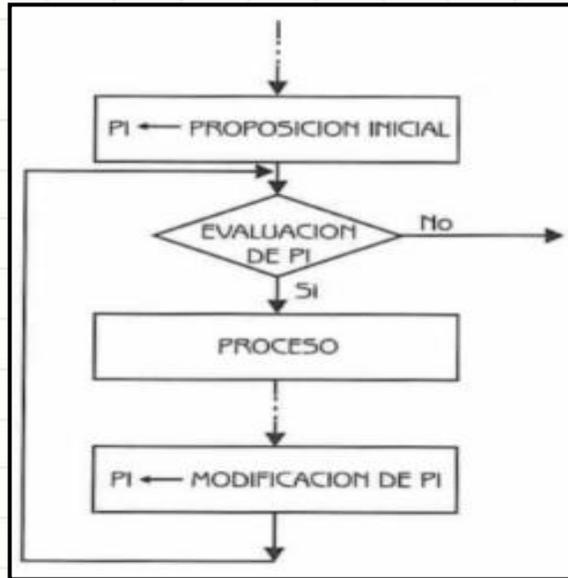
Todo **ciclo** debe terminar de ejecutarse después de un número finito de veces, por tanto, en cada iteración se debe evaluar la(s) condición(es) necesaria(s) para decidir si se debe seguir ejecutando el ciclo ó detenerse.



Las instrucciones son las mismas, los datos que operan en ellas son los que cambian.

La **estructuras repetitivas** se clasifican en: *mientras*, *hacer mientras* y *para*.

# 1. Estructura de control **Mientras**



La **estructura de control mientras** *se recomienda utilizar cuando no se sabe cuantas veces se repetirá el ciclo.* Tal número depende de las proposiciones que están dentro del ciclo.

Algunos ejemplos de problemas: cuando se tiene que obtener el total de una serie de gastos, pero no se sabe exactamente cuántos son; o cuando se tiene que sacar el promedio de calificaciones de un examen, pero tampoco se sabe cuántos alumnos lo aplicaron, el ciclo se repite mientras se tengan calificaciones de alumnos.

# 1. Estructura de control

## Mientras

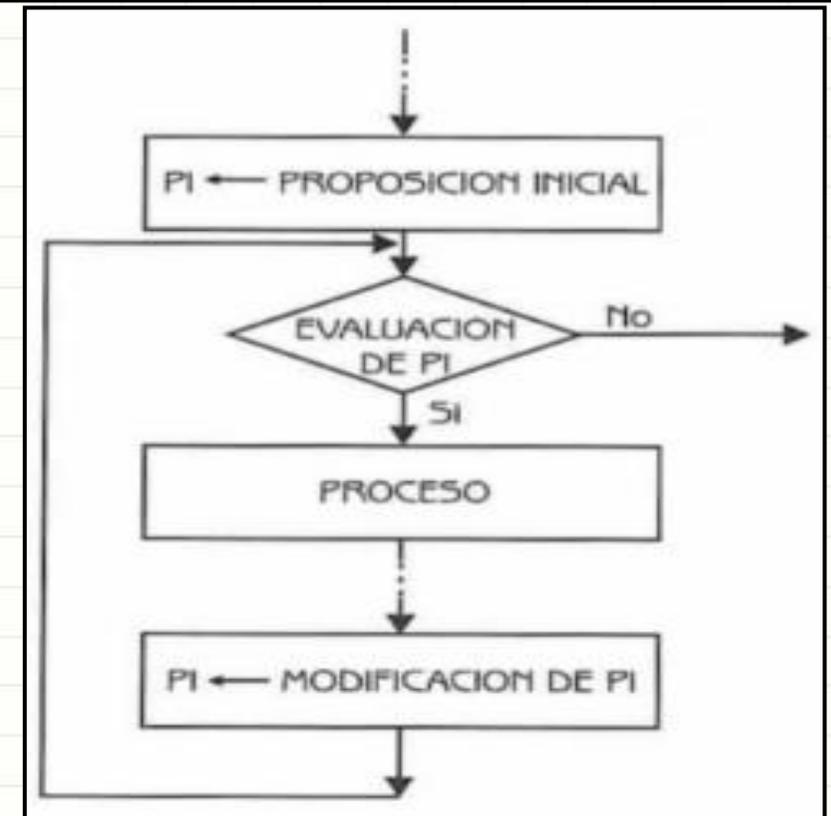
En la estructura **mientras** se distinguen dos partes:

- ❑ **Ciclo:** conjunto de instrucciones que se ejecutarán repetidamente.
- ❑ **Condición de terminación.** La evaluación de esta condición se realiza al inicio del ciclo y mientras la condición es verdadera se ejecuta el ciclo. Así mismo, *el ciclo se ejecuta cero ó más veces.*



## Sintaxis en pseudocódigo

```
vi ← valorInicial
mientras evaluación de vi
    acción 1
    ...
    acción n
vi ← modificación de vi
finMientras
```



## 1. Estructura de control **Mientras**

**Ejemplo 1.1.** Calcular la suma de los gastos que se hicieron en el último viaje, para terminar la entrada de gastos teclear -1.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

# 1. Estructura de control **Mientras**

**Ejemplo 1.1.** Calcular la suma de los gastos que se hicieron en el último viaje, para terminar la entrada de gastos teclear -1.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

Se recomienda el uso de la **estructura de control mientras** cuando *no se conoce la cantidad de entrada de datos*.

## **Pseudocódigo**

**Algoritmo** totalGastos

Definir gasto, suma como real

Escribir 'Programa que suma los gastos de un viaje'

Escribir 'Teclea un gasto [-1 para terminar]: '

Leer gasto

suma ← 0

**Mientras** gasto ≠ -1

..... suma ← suma + gasto

..... Escribir 'Teclea un gasto [-1 para terminar]: '

..... Leer gasto

**FinMientras**

Escribir 'El total de gastos fueron: ', suma

**FinAlgoritmo**

## Algoritmo totalGastos

Definir gasto, suma como real

Escribir 'Programa que suma los gastos de un viaje'

Escribir 'Teclea un gasto [-1 para terminar]: '

Leer gasto

suma ← 0

Mientras gasto ≠ -1

    suma ← suma + gasto

    Escribir 'Teclea un gasto [-1 para terminar]: '

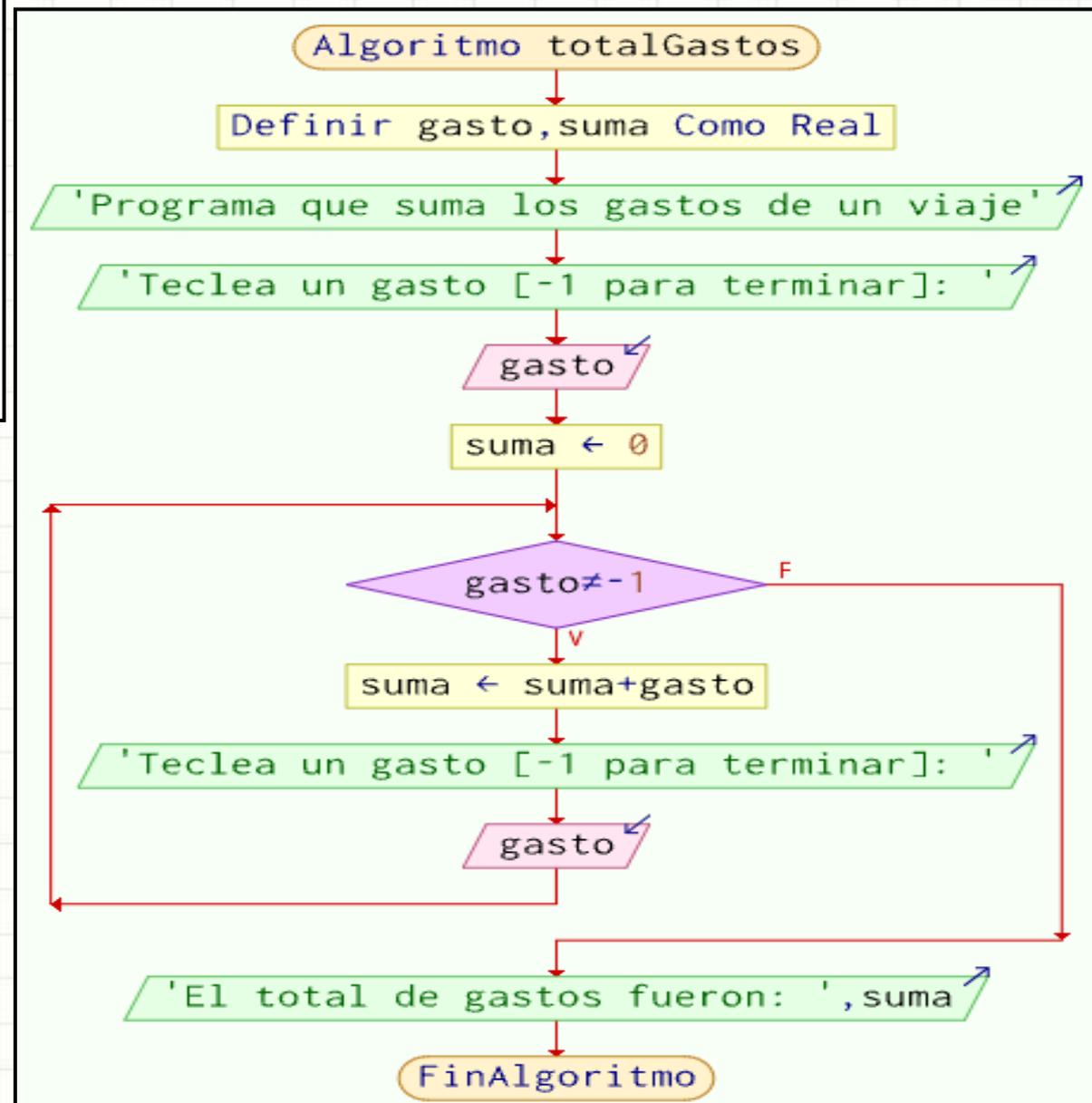
    Leer gasto

FinMientras

Escribir 'El total de gastos fueron: ', suma

FinAlgoritmo

## Pseudocódigo y su respectivo Diagrama de flujo



## Algoritmo totalGastos

Definir gasto, suma como real

Escribir 'Programa que suma los gastos de un viaje'

Escribir 'Teclea un gasto [-1 para terminar]: '

Leer gasto

suma ← 0

Mientras gasto ≠ -1

    suma ← suma + gasto

    Escribir 'Teclea un gasto [-1 para terminar]: '

    Leer gasto

FinMientras

Escribir 'El total de gastos fueron: ', suma

FinAlgoritmo

Fijarse que es importante recordar al usuario el dato que debe teclear para terminar la entrada de datos

Realice una prueba de escritorio con los siguientes gastos: 200, 150, 250 y -1

número de corrida	gasto	suma	gasto < > -1	Salida
1	200	0	v	Programa que suma los gastos de un viaje
	150	200	v	Teclea un gasto [-1 para terminar]: 200
	250	350	v	Teclea un gasto [-1 para terminar]: 150
	-1	600	f	Teclea un gasto [-1 para terminar]: 250
				Teclea un gasto [-1 para terminar]: -1
				El total de gastos fueron: 600

## 1. Estructura de control **Mientras**

**Ejemplo 1.2** Calcular e imprimir el cuadrado y cubo de 10 números naturales positivos introducidos por teclado.

*Realice el análisis del problema, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

**Ejemplo 1.2** Calcular e imprimir el cuadrado y cubo de 10 números naturales positivos introducidos por teclado.

*Realice el análisis del problema, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

## 1. Estructura de control **Mientras**

Uso de la estructura de control **mientras** *cuando se conoce la cantidad de entrada de datos.*

Pseudocódigo versión 1 cuando se introducen los datos correctos. Realice otra versión para que funcione con datos no válidos.

```
Algoritmo cuadradoCuboNumerosNaturales
  Definir i,num,cuadrado,cubo Como Entero
  Escribir "Programa que calcula el cuadrado y cubo de 10 numeros "sin saltar
  Escribir "naturales (enteros positivos)"
  i←1
  Mientras i≤10
  |   Escribir i,". Teclea un numero natural: "
  |   Leer num
  |   cuadrado←num*num
  |   cubo←num↑3
  |   Escribir "su cuadrado y cubo son: ",cuadrado," ",cubo," respectivamente"
  |   i←i+1
  FinMientras
FinAlgoritmo
```

## 1. Estructura de control **mientras**

**Ejercicio 1.3.** Dado por teclado un número entero  $\geq 0$ , calcular el factorial de ese número.

Definición:  $n! = n * n-1 * n-2 * \dots * 1$

Ejemplo  $4! = 4 * 3 * 2 * 1 = 24$

$0! = 1$

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con tres corridas para  $num = 4, 5$  y  $0$ .*

## 2. Estructura de control **Hacer ... mientras**

**Hacer ... mientras**, es una estructura de control parecida al ciclo **Mientras**, la diferencia es que la **condición se coloca al final del ciclo**, y éste último se repite mientras la condición sea verdadera.

Esto significa que las instrucciones del ciclo se ejecutan **al menos una vez**. Ver sintaxis **pseudocódigo** y **diagrama de flujo**.

### Pseudocódigo

...

**VI** ← valorInicial

**hacer**

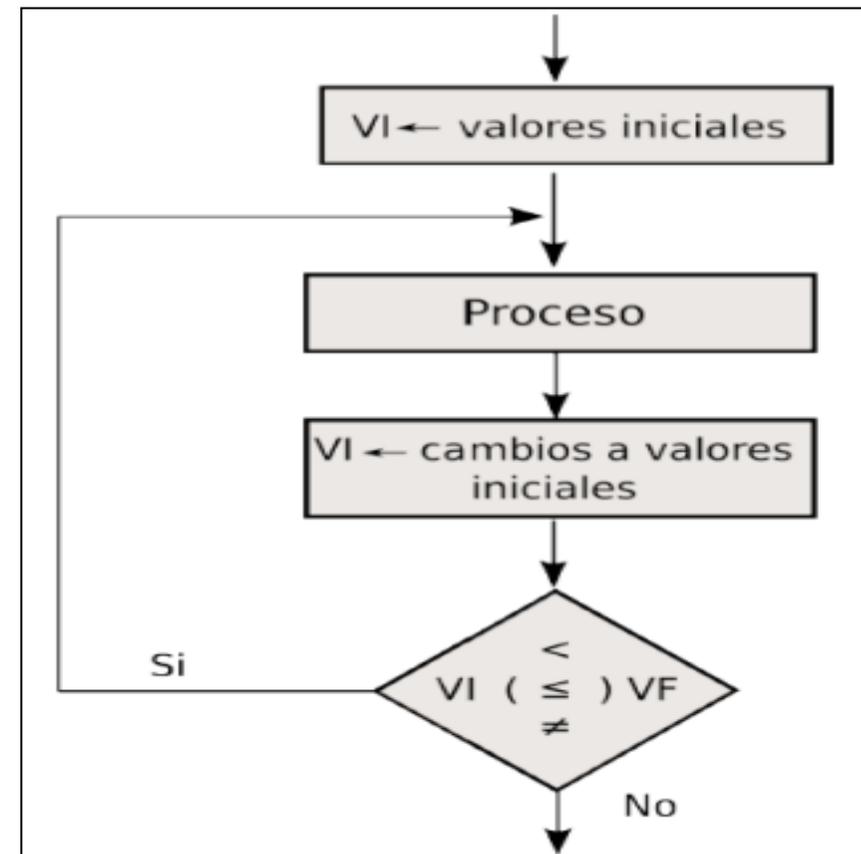
acción 1

...

acción n

**VI** ← modificación de **VI**

**mientras** evaluación de **VI**



## 2. Estructura de control **Hacer ... mientras**

**Ejemplo 2.1.** Dado por teclado 10 números enteros, calcular e imprimir la suma de los números impares y el promedio de los números pares.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

## Estructura de control Hacer ... mientras

**Ejemplo 2.1.** Dado por teclado 10 números enteros, calcular e imprimir la suma de los números impares y el promedio de los números pares.

*Después de realizar el análisis en clases (D. entrada, Proceso y Salida), se presenta una solución en pseudocódigo.*

```
Algoritmo sumaImpares_PromedioPares
```

```
    var i,num,sumaI, sumaP,contP,residuo: Entero
```

```
Inicio
```

```
    i<-1
```

```
    sumaI<-0
```

```
    sumaP<-0
```

```
    contP<-0
```

```
    Imprimir "Programa que dado 10 números enteros, realiza  
    la suma de los impares y el promedio de los pares"
```

```
Hacer
```

```
    Imprimir "Teclea el número ",i,":"
```

```
    Leer num
```

```
    residuo<-num mod 2
```

```
    si residuo=0
```

```
        sumaP<-sumaP+num
```

```
        contP<-contP+1
```

```
    SiNo
```

```
        sumaI<-sumaI+num
```

```
    FinSi
```

```
    i<-i+1
```

```
Mientras i<=10
```

```
    si contP>0
```

```
        prom<-sumaP/contP
```

```
        Imprimir "El promedio de los número pares es: ",prom
```

```
    SiNo
```

```
        Imprimir "No introdujo número pares"
```

```
    FinSi
```

```
    Imprimir "La suma de los números impares es: ",sumaI
```

```
Fin
```

# Estructura de control Hacer ... mientras

## Pseudocódigo y Diagrama de flujo.

Algoritmo sumaImpares\_PromedioPares

```
var i,num,sumaI, sumaP,contP,residuo: Entero
```

Inicio

```
i<-1
```

```
sumaI<-0
```

```
sumaP<-0
```

```
conP<-0
```

```
Imprimir "Programa que dado 10 números enteros, realiza la suma de los impares y el promedio de los pares"
```

Hacer

```
Imprimir "Teclea el número ",i,":"
```

```
Leer num
```

```
residuo<-num mod 2
```

```
si residuo=0
```

```
    sumaP<-sumaP+num
```

```
    contP<-contP+1
```

```
SiNo
```

```
    sumaI<-sumaI+num
```

```
FinSi
```

```
i<-i+1
```

```
Mientras i<=10
```

```
si contP>0
```

```
    prom<-sumaP/contP
```

```
    Imprimir "El promedio de los número pares es: ",prom
```

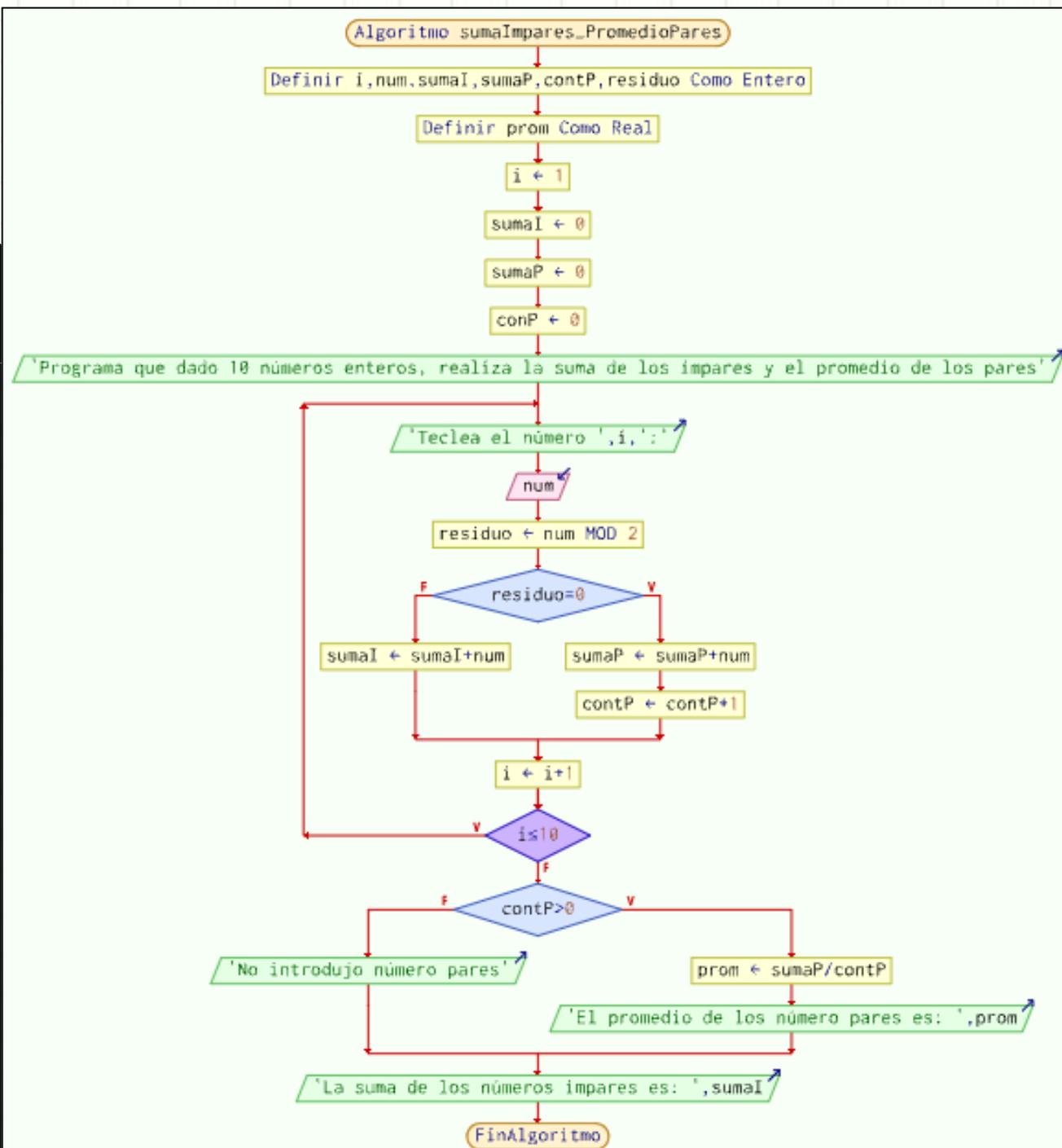
```
SiNo
```

```
    Imprimir "No introdujo número pares"
```

```
FinSi
```

```
Imprimir "La suma de los números impares es: ",sumaI
```

Fin



## 2. Estructura de control **Hacer ... mientras**

**Ejemplo 2.2.** Dado un número entero positivo por teclado, obtener e imprimir el número al revés.

Ejemplo 285 , el número al revés es 582

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

## 2. Estructura de control Hacer ... mientras

**Ejemplo 2.2.** Dado un número entero positivo por teclado, obtener e imprimir el número al revés.

Ejemplo 285 , el número al revés es 582

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

**Mejorar el siguiente pseudocódigo que valide que el número de entrada sea entero positivo**

```
Algoritmo imprimirEnteroAlRevés
var num, residuo: entero
Inicio
    Escribir "Programa que lee un número
        positivo y lo imprime al revés"
    Escribir "Teclea un numero:"
    Leer num
    Hacer
        residuo<-num mod 10
        Escribir residuo
        num <- num / 10
    mientras num>0
Fin
```

## 2. Estructura de control Hacer ... mientras

**Ejemplo 2.3.** Dado por teclado un número  $\geq 0$ , calcular el factorial de ese número.

Definición:  $n! = n * n-1 * n-2 * \dots * 1$

Ejemplo  $4! = 4 * 3 * 2 * 1 = 24$

$0! = 1$

*Realice el análisis, pseudocódigo y prueba de escritorio con tres corridas para  $num = 4, 5$  y  $0$ .*

Se presenta una solución en pseudocódigo con algunas validaciones. Pruebe y mejore agregando dato no válido en caso que la entrada no sea válida.

**Algoritmo** factorial

Definir num, resultado como Entero

Imprimir 'Programa que calcula el factorial de un numero'

Imprimir 'Teclea un numero entero positivo: '

Leer num

resultado  $\leftarrow$  1

si num = 0

..... Imprimir 'El factorial es: ', resultado

FinSi

si num > 0

..... Hacer

..... resultado  $\leftarrow$  resultado \* num

..... num  $\leftarrow$  num - 1

..... Mientras Que num > 1

..... Imprimir 'El factorial es: ', resultado

FinSi

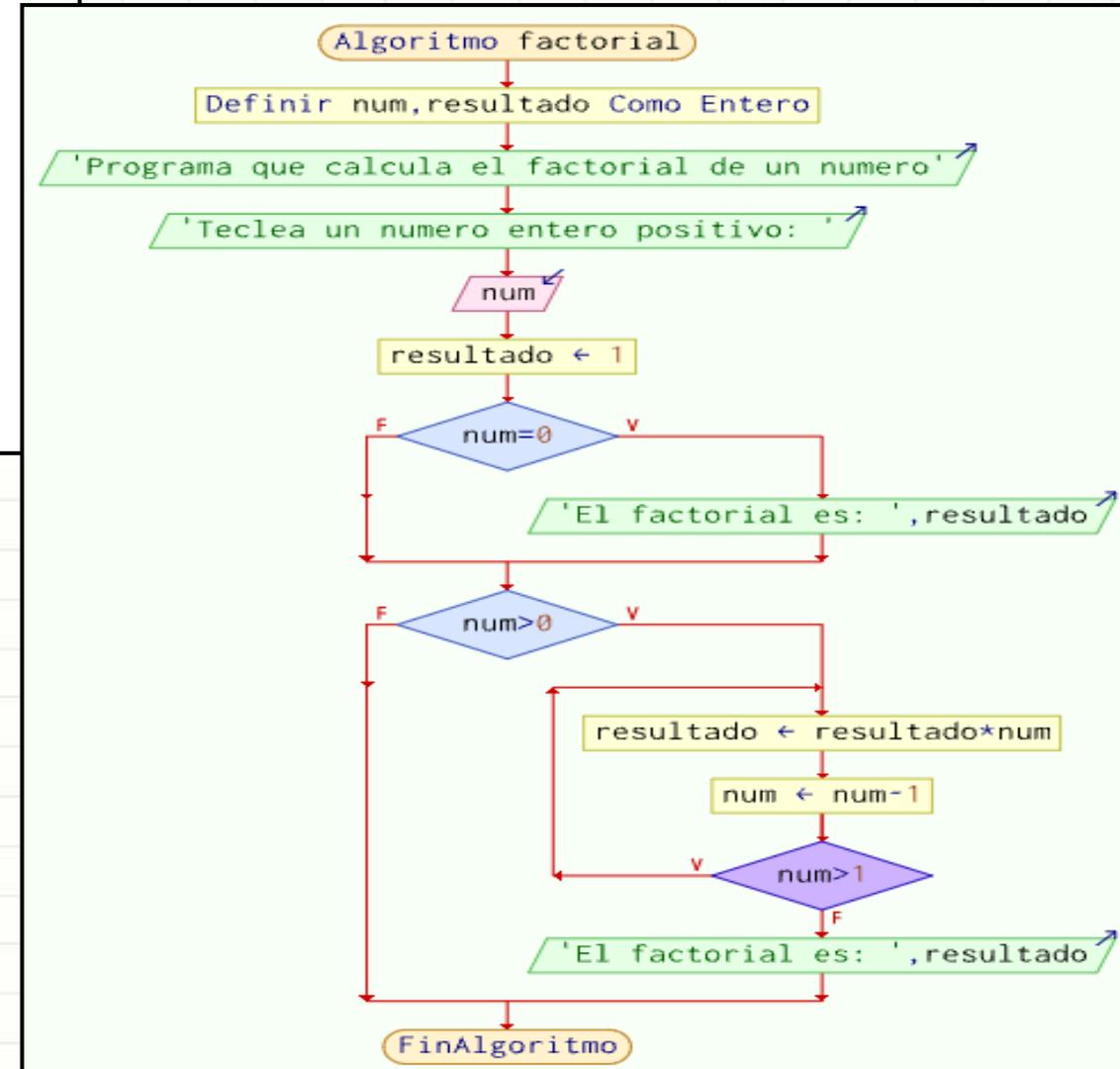
**FinAlgoritmo**

## Algoritmo factorial

```
Definir num,resultado como Entero
Imprimir 'Programa que calcula el factorial de un numero'
Imprimir 'Teclea un numero entero positivo: '
Leer num
resultado←1
si num = 0
    Imprimir 'El factorial es: ',resultado
FinSi
si num>0
    Hacer
        resultado←resultado*num
        num←num-1
    Mientras Que num>1
    Imprimir 'El factorial es: ',resultado
FinSi
FinAlgoritmo
```

Pseudocódigo y su respectivo Diagrama de flujo.

## 2. Estructura de control Hacer ... mientras



## 2. Estructura de control **Hacer ... mientras**

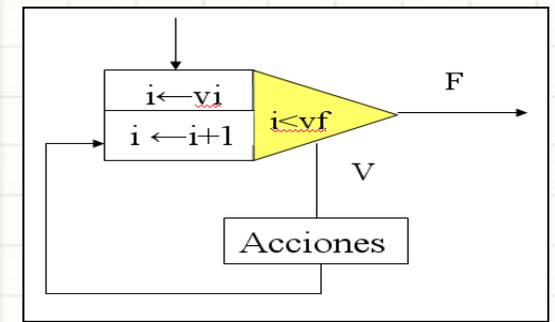
**Ejercicio.** Dado un número entero positivo por teclado, determinar e imprimir si es número primo o no.

**Definición:** Un número primo es aquel que solo es divisible por sí mismo y por la unidad.

Ejemplo de números primos: 1, 2, 3, 5, 7, 11 etc.

*Realizar análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas. Primero resuélvalo usando la estructura de control **mientras** y después con la estructura de control **hacer ... mientras**.*

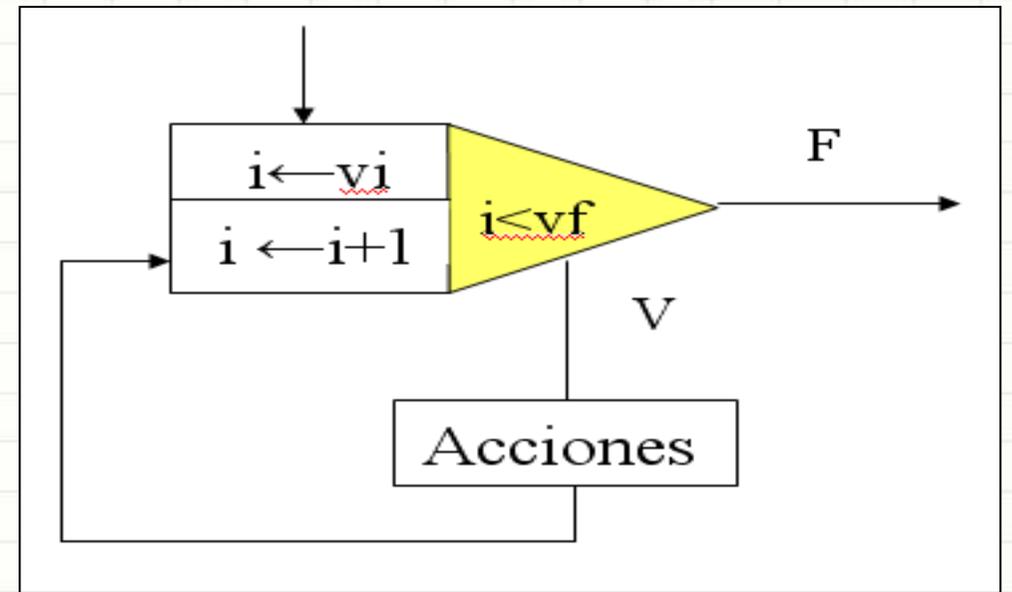
### 3. Estructura de control Para ... hasta



- La estructura de control repetitiva **Para ... hasta**, ejecuta un **ciclo** un número definido de veces.
- Por ejemplo cuando se necesita calcular el promedio de calificaciones de un curso, se deben sumar las N calificaciones de los alumnos y dividir la suma entre N.
- También cuando se necesita calcular la nómina total de la empresa, se tienen que sumar los sueldos de los N empleados de la misma.
- Es decir, se sabe por anticipado cuántas veces tenemos que repetir una determinada operación, acción o tarea. El número de veces se obtiene del planteamiento del problema o de una lectura.

### 3. Estructura de control Para ... hasta

- A continuación se presenta la sintaxis tanto en pseudocódigo como en diagrama de flujo.



#### Sintaxis en pseudocódigo

...

**Para**  $v_i \leftarrow$  valor inicial **hasta**  $v_f$  **incremento o decremento** en 1 o más

acción 1

...

acción n

**finPara**

...

### 3. Estructura de control Para ... hasta

**Ejemplo 3.1.** Dado por teclado el valor de  $n$  trabajadores, teclee posteriormente los sueldos de los  $n$  trabajadores de una empresa y obtenga el total de la nómina a pagar.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

### 3. Estructura de control Para ... hasta

**Ejemplo 3.1.** Dado por teclado el valor de  $n$  trabajadores, teclee posteriormente los sueldos de los  $n$  trabajadores de una empresa y obtenga el total de la nómina a pagar

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

#### **Análisis del problema(para)**

**Datos de entrada:** ej.  $n=4$

ejemplo 1000, 1500, 2000,1200

variable  $n$ : entero

sueldo: real

#### **Proceso**

Variable  $i$ : entero

variable totalNomina  $\leftarrow 0$ : real

**Para  $i \leftarrow 1$  hasta  $n$  incremento 1**

Pedir sueldo

Leer sueldo

totalNomina  $\leftarrow$  totalNomina + sueldo

**FinPara**

#### **Salida**

Imprimir El total de la nómina es ejemplo 5700.

Después de probar esta solución, realice otra versión que valide los sueldos de los trabajadores.

## Análisis del problema(para)

**Datos de entrada:** ej. n=4

ejemplo 1000, 1500, 2000,1200

variable n: entero

sueldo: real

## Proceso

Variable i: entero

variable totalNomina <- 0: real

**Para** i<-1 hasta n **incremento 1**

Pedir sueldo

Leer sueldo

totalNomina<-totalNomina + sueldo

**FinPara**

## Salida

Imprimir El total de la nómina es ejemplo 5700.

```
Algoritmo calcularTotalNomina
```

```
var i,n: entero
```

```
var sueldo, totalNomina: real
```

```
Inicio
```

```
    Escribir "Programa que calcula el total  
    de la nómina de n trabajadores"
```

```
    Escribir "Teclea cuántos sueldos vas a  
    proporcionar:?"
```

```
    Leer n
```

```
    totalNomina<-0
```

```
    Para i<-1 Hasta n Incremento 1
```

```
        Escribir "Teclea el sueldo del  
        trabajador ",i,":"
```

```
        Leer sueldo
```

```
        totalNomina<-totalNomina+sueldo
```

```
    FinPara
```

```
    Escribir "El total de la nómina es:  
    ",totalNomina
```

```
FinAlgoritmo
```

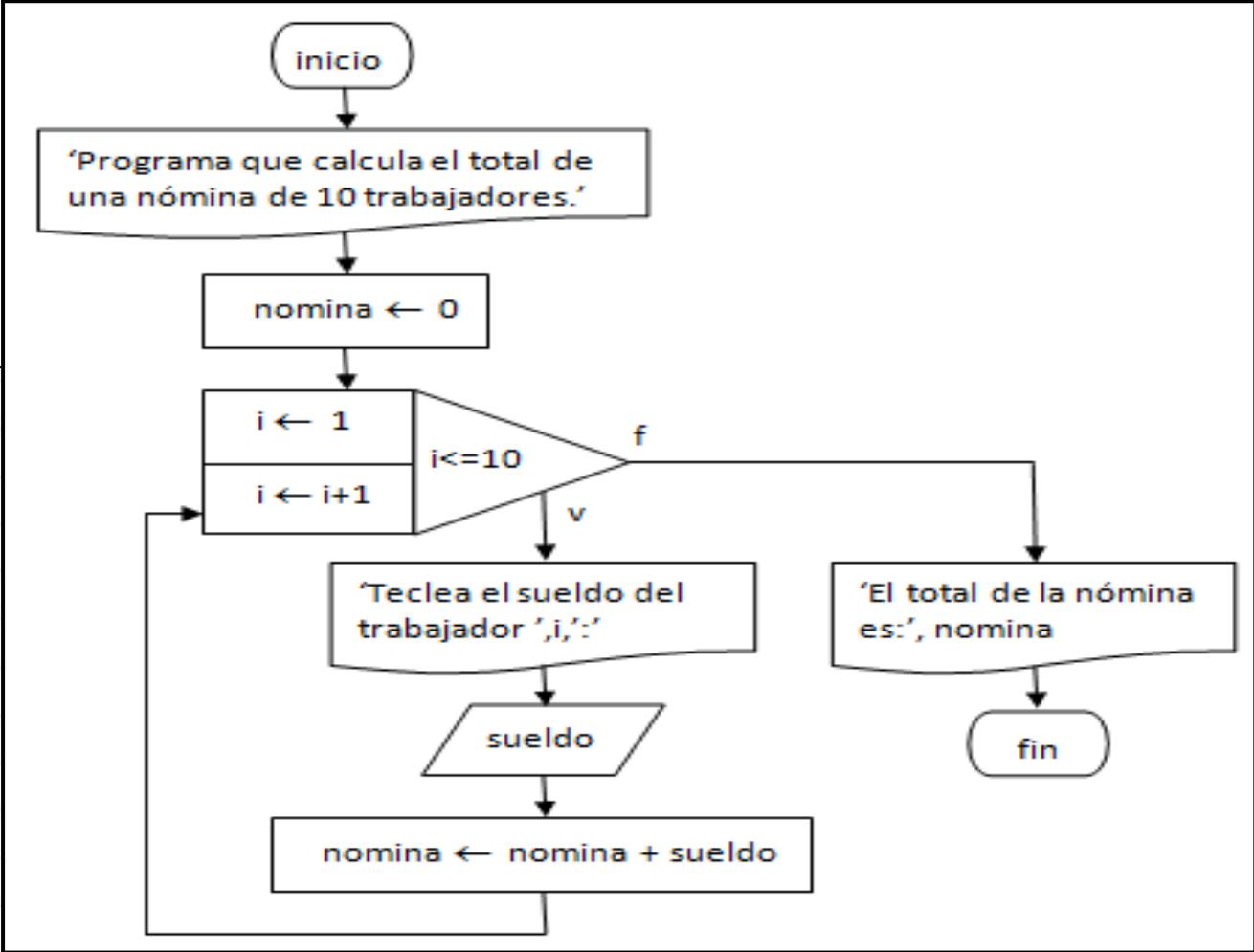
```

Algoritmo calcularTotalNomina
var i,n: entero
var sueldo, totalNomina: real
Inicio
  Escribir "Programa que calcula el total
    de la nómina de n trabajadores"
  Escribir "Teclea cuántos sueldos vas a
    proporcionar:?"
  Leer n
  totalNomina<-0
  Para i<-1 Hasta n Incremento 1
    Escribir "Teclea el sueldo del
      trabajador ",i,":"
    Leer sueldo
    totalNomina<-totalNomina+sueldo
  FinPara
  Escribir "El total de la nómina es:
    ",totalNomina
FinAlgoritmo

```

### 3. Estructura de control Para ... hasta

**Pseudocódigo y su respectivo Diagrama de flujo**



### 3. Estructura de control

#### Para ... hasta

#### Pseudocódigo

Realice una prueba de escritorio con los siguientes datos: 1000, 2500, 3700 y 5000

```
Algoritmo calcularTotalNomina
var i,n: entero
var sueldo, totalNomina: real
Inicio
  Escribir "Programa que calcula el total
  de la nómina de n trabajadores"
  Escribir "Teclea cuántos sueldos vas a
  proporcionar:?"
  Leer n
  totalNomina<-0
  Para i<-1 Hasta n Incremento 1
    Escribir "Teclea el sueldo del
    trabajador ",i,":"
    Leer sueldo
    totalNomina<-totalNomina+sueldo
  FinPara
  Escribir "El total de la nómina es:
  ",totalNomina
FinAlgoritmo
```

número de corrida	nomina	i	i<=4	sueldo	Salida
1	0	1	v	1000	Programa que obtiene ...
	1000	2	v	1500	Teclea el sueldo del trabajador 1: 1000
	2500	3	v	1200	Teclea el sueldo del trabajador 2: 1500
	3700	4	v	1300	Teclea el sueldo del trabajador 3: 1200
	5000	5	f		Teclea el sueldo del trabajador 4: 1300
					El total de la nómina es: 5000

### 3. Estructura de control Para ... hasta

**Ejemplo 3.2.** Dado por teclado las calificaciones válidas de N alumnos, calcular e imprimir el promedio. La validación de las calificaciones las debe realizar con la estructura de control **hacer-mientras**.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

### 3. Estructura de control Para ... hasta

**Ejemplo 3.2.** Dado por teclado las calificaciones válidas de N alumnos, calcular e imprimir el promedio. La validación de las calificaciones las debe realizar con la estructura de control **hacer-mientras**.

*Realice el análisis, pseudocódigo, diagrama de flujo y prueba de escritorio con dos corridas.*

Mejore la siguiente versión pero validando que las N calificaciones sean válidas.

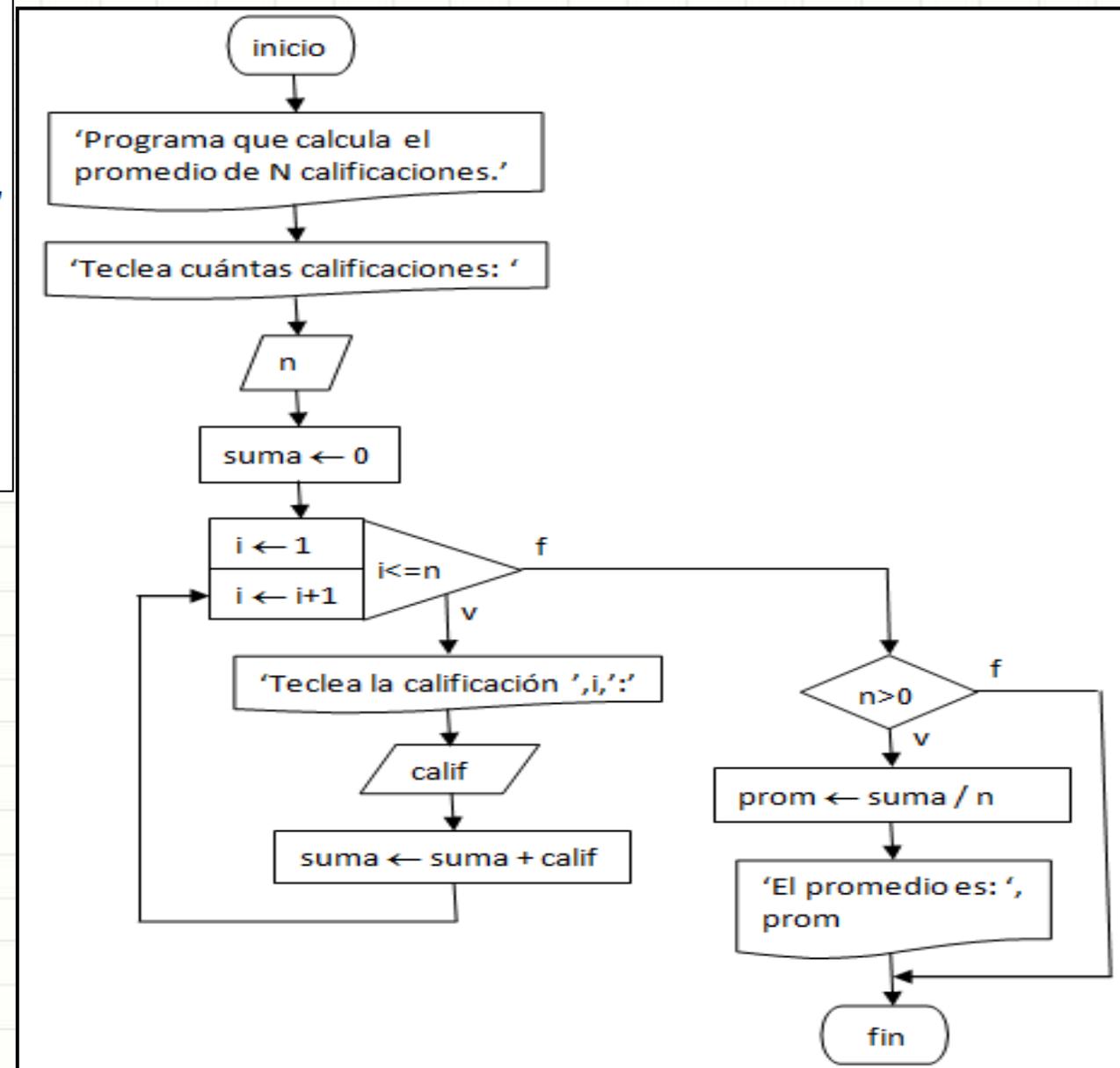
```
Algoritmo promedioCalificaciones
var n, i : entero
var calif, suma: real
Inicio
  Escribir "Programa que calcula el
           promedio de n calificaciones"
  Escribir "Teclea cuantas calificaciones:"
  Leer n
  suma <-0
  Para i<-1 Hasta n incremento 1
    Escribir "Teclea la calificacion ",i,":"
    Leer calif
    suma<-suma+calif
  FinPara
  Si n>0
    prom<-suma/n
    Escribir "El promedio es: ",prom
  FinSi
Fin
```

### Algoritmo promedioCalificaciones

```
var n, i : entero
var calif, suma: real
Inicio
  Escribir "Programa que calcula el
  promedio de n calificaciones"
  Escribir "Teclea cuantas calificaciones:"
  Leer n
  suma <- 0
  Para i <- 1 Hasta n incremento 1
    Escribir "Teclea la calificacion ", i, ":"
    Leer calif
    suma <- suma + calif
  FinPara
  Si n > 0
    prom <- suma / n
    Escribir "El promedio es: ", prom
  FinSi
Fin
```

## Pseudocódigo y su respectivo Diagrama de flujo.

Ahora realice una prueba de escritorio con las siguientes calificaciones: 6.8, -2, 20, 8.2, 9.0,...



## Ejercicios donde utilice principalmente estructuras de control repetitivas

Para cada uno de los siguientes ejercicios realice su análisis, pseudocódigo, diagrama de flujo y una prueba de escritorio con 1 corrida. Así mismo, para cada ejercicio identifique y resuelva con la estructura de control repetitiva más adecuada: **mientras**, **hacer-mientras** o **para ... hasta**

1. Dado por teclado como datos 10 números enteros, calcular e imprimir la suma de los números impares y el promedio de los números pares.
2. Identifique la serie que genera la siguiente lista de números, posteriormente calcule e imprima cada uno de los términos hasta el 50. También debe calcular e imprimir el total de la suma de todos los términos generados.

2, 5, 7, 10, 12, 15, ..., 40

## Ejercicios donde utilice principalmente estructuras de control repetitivas

3. Dado por teclado un número entero N, calcule e imprima el resultado de la sumatoria.

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \pm \frac{1}{N}$$

4. Calcule el aumento de sueldo para un grupo de empleados de una empresa considerando el siguiente criterio:

Si el sueldo es inferior a \$1000 , aumento 15%

Si el sueldo es mayor o igual a \$1000, aumento 12%

La entrada de datos termina cuando el sueldo es negativo.

## Ejercicios donde utilice principalmente estructuras de control repetitivas

5. Dado N números enteros por teclado, calcular lo siguiente:

- a) ¿Cuántos números leídos fueron mayores o igual a cero?
- b) ¿Cuántos números leídos fueron menores a cero?
- c) El promedio de los números positivos incluido el cero

6. Dado N números enteros por teclado, calcular el mayor y el menor de estos números.

## Ejercicios donde utilice principalmente estructuras de control repetitivas

7. Dado por teclado N ventas con los siguientes criterios, imprima cuantas corresponden a cada criterio.

- a) Ventas de \$200 o menos.
- b) Ventas mayores a \$200, pero menores a \$400
- c) Ventas de \$400 o más.

8. Dado por teclado un número entero positivo, calcular los términos de la serie de ULAM, la cual consiste en lo siguiente: Si el número es par, divídalo entre 2; si es impar, multiplíquelo por 3 y agréguele 1. Repetir el proceso hasta obtener el número 1.

Ejemplo: sin numero= 26, se obtendrá: 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

# Ejercicios donde utilice principalmente estructuras de control repetitivas

8. Leer un número entero positivo N y que imprima el siguiente histograma.

Ejemplo, si  $N=5$ , generará el siguiente histograma.

```
*****
```

```
****
```

```
***
```

```
**
```

```
*
```

## Ejercicios donde utilice principalmente estructuras de control repetitivas

9. Dados dos números enteros positivos M y N. Calcular el cociente y el residuo de la división mediante restas sucesivas. El método consiste en restar sucesivamente el divisor del dividendo hasta obtener un resultado menor que el divisor, que será el resto de la división; el número de restas efectuadas será el cociente.

Ejemplo M(dividendo) = 45, N(divisor) = 12

$$45-12 = 33 \quad c=1$$

$$33-12= 21 \quad c=2$$

$$21-12= 9 \quad c=3$$

Por tanto, el cociente es 3 y el residuo es 9.

## Referencias

1. Zapata Ospina, Carlos A. "Fundamentos de programación, Guía de autoenseñanza". Alfaomega Ra-Ma. 2006
2. Cairó Osvaldo. "Metodología de la programación". Alfaomega, 3ª edición. 2005.
3. Joyanes Aguilar, Luis. "Fundamentos de programación: Algoritmos, estructura de datos y objetos". McGraw-Hill, 4ª edición. 2008