

Universidad Autónoma de Entre Ríos  
**Facultad de Ciencia y Tecnología**  
Sede: Oro Verde



---

# **FUNDAMENTOS DE PROGRAMACIÓN**

RESUMEN DE CONTENIDOS N°4  
**Estructuras de Control: Iterativas**

## Estructuras de Control: Iterativas

### Introducción

Analizaremos ahora las diferentes formas de plantear, en la construcción de algoritmos, la repetición de la ejecución de una o varias acciones, a través de las estructuras de ITERACIÓN.

Las estructuras de iteración son aquellas que nos permiten ejecutar una acción o un conjunto de acciones varias veces, dependiendo de una condición.

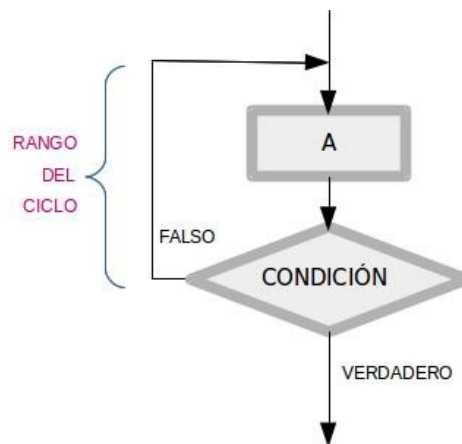
En la solución de problemas computacionales, se presentan casos en los que no se conoce de antemano la cantidad de veces que se quiere repetir un conjunto de acciones, en otros, ese número de repeticiones es conocido. Definiremos estructuras algorítmicas que nos permitan describir ambas situaciones.

Dentro de las estructuras de iteración, veremos, en pseudocódigo, los tres esquemas básicos de la programación estructurada:

- REPETIR
- MIENTRAS
- PARA

### REPETIR

La estructura **Repetir**, se representa como:  
En diagrama de flujo:



donde **condición** puede ser una variable o expresión lógica y **A** representa cualquier acción primitiva o conjunto de estructuras, llamado **rango del ciclo**.

En pseudocódigo la sintaxis equivalente al diagrama es:

```

REPETIR
A
HASTAQUE condición
    
```

donde el principio y el fin de la estructura están dados por las palabra claves **REPETIR** y **HASTAQUE**.

Esta estructura es considerada como una unidad, con un único punto de entrada y un único punto de salida.

Esta estructura plantea la ejecución del bloque **A**, reiteradamente, hasta que se cumpla la condición.

En presencia de este esquema, el procesador deberá:

- ejecutar el bloque **A**
- Observar la **condición**
- Si la condición es **FALSO** ⇒ ejecutar **A**
- Observar nuevamente la **condición**
- Si es **FALSO** , ejecutar **A** y vuelve a evaluar la condición y así sucesivamente hasta que de la observación de la condición se obtenga un valor verdadero
- Si es **VERDADERO** , sale de la estructura
- Si la condición es **VERDADERO** ⇒ finaliza la ejecución de la estructura



**Observación:** no se coloca punto y coma (;):

- después de la palabra clave **REPETIR**.
- después del predicado **HASTAQUE** condición



**Debemos notar que:**

- La condición es evaluada después de cada ejecución del bloque A, por lo tanto éste será ejecutado al menos una vez.
- Entre el conjunto de acciones de A, deberá existir una acción que permita modificar, en determinado momento, su valor de verdad; sino el bloque A se repetirá indefinidamente.

### Ejemplo Práctico 1:

*Una compañía de turismo ha definido una política de promoción de sus empresas para lo cual ha fijado descuentos para sus clientes en función de los viajes anteriores realizados. El valor del descuento es del 20% para aquellos clientes que, considerando los viajes realizados en el último año y el que desean realizar, han recorrido más de 3000 Kms. y del 5% para aquellos que no han alcanzado dicha cifra.*

*Se desea definir un algoritmo que calcule la recaudación diaria de la compañía para sus N clientes. Para ello, por cada cliente se ingresa su nombre y apellido, el total de Kms. recorridos hasta el momento, y los datos del viaje a contratar: precio, destino, Kms. a recorrer. Para cada cliente informar los datos ingresados, el monto del descuento y el monto a pagar. Informar además el total recaudado y el total de descuentos efectuados por la Compañía, con leyendas alusivas.*

*El valor N se ingresa como primer dato.*

### Ambiente:

Variable	Num	Carac.	Lóg	Clase	Significado
N	X			Simple	Cantidad de Clientes
NYA		X		Simple	Nombre y Apellido del Cliente
TACUM	X			Simple	Kms. acumulados hasta la fecha
PRECIO	X			Simple	Precio del viaje
KMS	X			Simple	Kms. a recorrer
DEST		X		Simple	Destino del Viaje
TOTAL	X			Simple	Kms. Tot. incluidos los del viaje actual
DESC	X			Simple	Descuento a realizar
IMP	X			Simple	Importe a Pagar
TOTDESC	X			Simple	Total de Descuentos
TOTIMP	X			Simple	Total recaudado por la Compañía
CONT	X			Simple	Contador de Clientes

### Algoritmo: En Pseudocódigo

#### ALGORITMO TURISMO2

TOTDESC = 0; TOTIMP = 0; CONT = 0;

LEER N;

#### REPETIR

LEER NYA, TACUM, PRECIO, DESTINO, KMS;

TOTAL= TACUM + KMS ;

SI TOTAL > 3000

ENTONCES DESC= PRECIO \* 0.20

SINO DESC= PRECIO \* 0.05

#### FINSI

IMP= PRECIO - DESC;

ESCRIBIR "Cliente: ", NYA, "Acumulados: ", TACUM;

ESCRIBIR "Datos del viaje: \$ ", PRECIO, KMS, "Kms. ", DESTINO;

ESCRIBIR "Descuento: \$ ", DESC, "Importe: \$ ", IMP;

TOTIMP= TOTIMP + IMP;

TOTDESC= TOTDESC + DESC;

CONT= CONT + 1;

HASTAQUE CONT = N

ESCRIBIR "Recaudación de la Cia: \$ ", TOTIMP;

ESCRIBIR "Total de Descuentos: \$ ", TOTDESC;

#### FINALGORITMO



**Observación:** vemos en este ejemplo que conocemos de antemano la cantidad de clientes a procesar a través de la variable N.

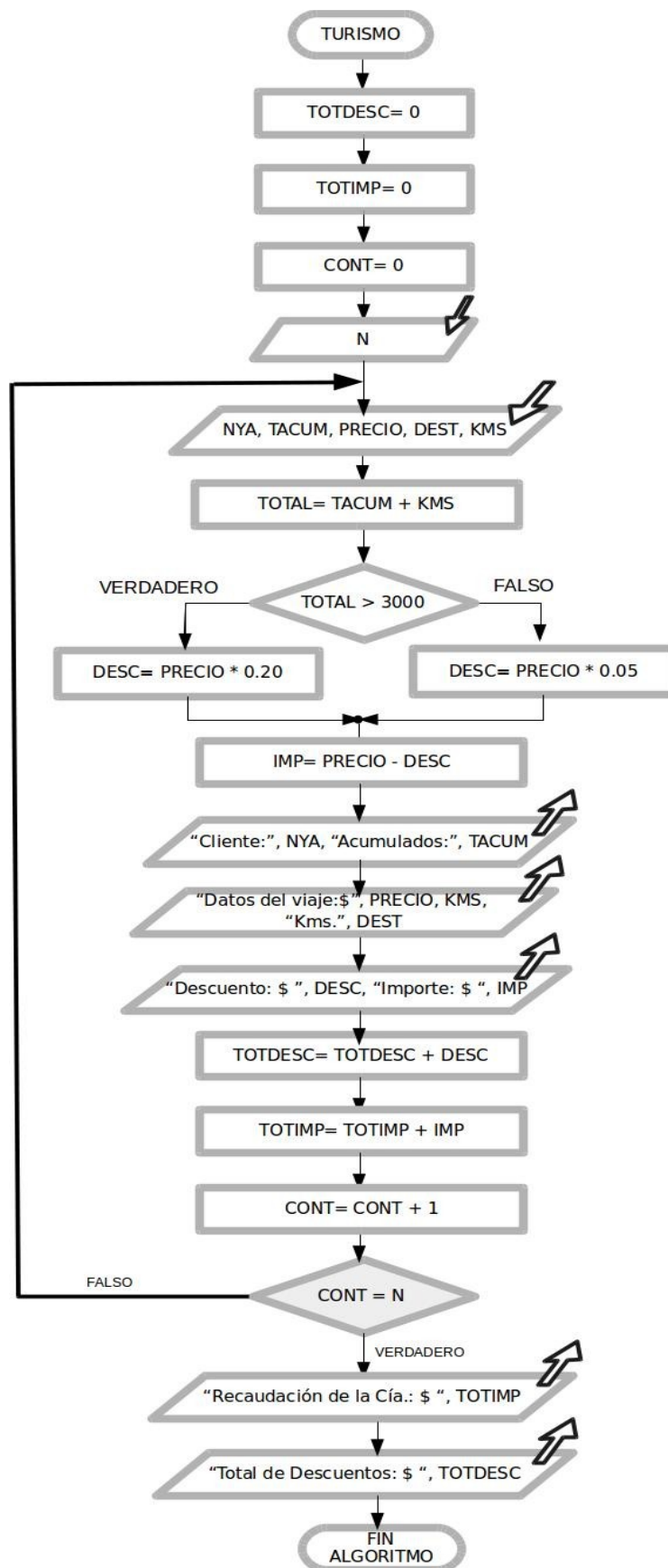
### Seguimiento:

**Datos:** 2, 'Juan Pérez', 3100, 350, 'Misiones', 800  
'José Díaz', 300, 500, 'Mendoza', 956

NYA	TACUM	PRECIO	KMS	DEST	TOTAL	DESC	IMP	TOTDES	TOTIMP	N	CONT
'Juan Pérez'	3100	350	800	'Misiones'	3900	70	289	0	0	2	0
								70	299		1
'José Díaz'	300	500	956	'Mendoza'	1256	25	475	95	774		2

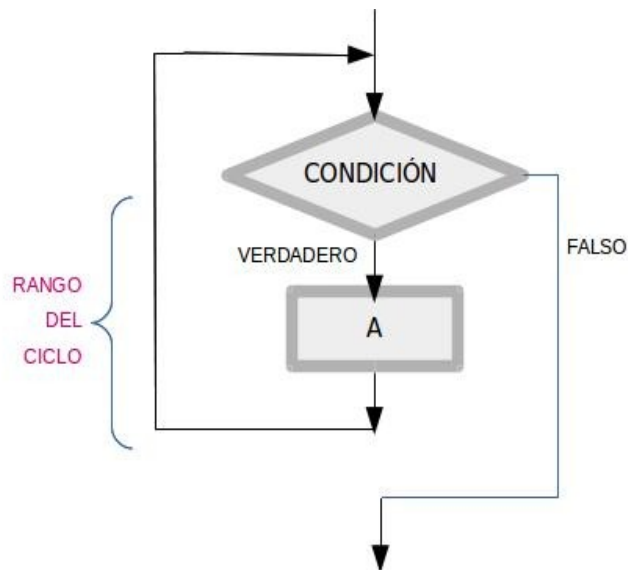
Acciones de escritura
Cliente: Juan Pérez Acumulados: 3100
Datos del viaje: \$350 800 Kms. Misiones
Descuento: \$70 Importe: \$289
Cliente: José Díaz Acumulados: 300
Datos del viaje: \$500 956 Kms. Mendoza
Descuento: \$25 Importe: \$475
Recaudación de la Cia: \$ 774
Total de Descuentos: \$ 95

## Algoritmo: En Diagrama de Flujo



## MIENTRAS

La estructura **Mientras**, se representa como:  
En diagrama de flujo:



donde **condición** puede ser una variable o expresión lógica y **A** representa cualquier acción primitiva o conjunto de estructuras, llamado **rango del ciclo**.  
 Como en las estructuras vistas anteriormente -secuencia y selección- la iteración se considera una unidad, con un sólo punto de entrada y un sólo punto de salida.

En pseudocódigo la sintaxis equivalente al diagrama es:

```
MIENTRAS condición HACER  
A  
FINMIENTRAS
```

Las palabras claves **MIENTRAS** y **FINMIENTRAS** identifican, respectivamente, el principio y fin de la estructura.

Cuando el procesador encuentra esta estructura deberá:

- Observar la condición
- Si la condición es **VERDADERO** ⇒ ejecutar **A**
- Observar nuevamente la condición
- Si es **VERDADERO**, ejecutar **A** y volver a evaluar la condición y así sucesivamente hasta que de la observación de la condición se obtenga un valor **falso**
- Si es **FALSO**, finaliza la ejecución
- Si la condición es **FALSO** ⇒ salir de la estructura



### Debemos notar que:

- El bloque A puede no llegar a ejecutarse, si al principio, la condición es falsa.
- Entre el conjunto de acciones de A, deberá existir una acción que permita modificar, en determinado momento, su valor de verdad; sino el bloque A se repetirá indefinidamente.
- No es necesario conocer de antemano el número de repeticiones, o bien no es necesario expresarlo en forma directa.
- En la estructura Mientras el bloque A pueda ser ejecutado desde 0 a n veces.
- La diferencia entre la estructura Repetir y la estructura Mientras, reside en el punto de evaluación de la condición. Esto hace que en la estructura Mientras el bloque A pueda ser ejecutado desde 0 a n veces, mientras que en el Repetir desde 1 a n veces.

### Ejemplo Práctico 2:

Una compañía de turismo ha definido una política de promoción de sus empresas para lo cual ha fijado descuentos para sus clientes en función de los viajes anteriores realizados. El valor del descuento es del 20% para aquellos clientes que, considerando los viajes realizados en el último año y el que desean realizar, han recorrido más de 3000 Kms. y del 5% para aquellos que no han alcanzado dicha cifra.

Se desea realizar un algoritmo que calcule la recaudación de la compañía en un día. Para ello, por cada cliente se ingresa su nombre y apellido, el total de Kms. recorridos hasta el momento, y los datos del viaje a contratar: precio, destino, Kms. a recorrer. El fin de datos se produce al ingresar como nombre y apellido un valor "ZZZ".

Para cada cliente informar los datos ingresados, el monto del descuento y el monto a pagar. Informar además el total recaudado y el total de descuentos realizados por la Compañía, con leyendas alusivas.

### Ambiente:

Variable	Num	Carac.	Lóg	Clase	Significado
NYA		X		Simple	Nombre y Apellido del Cliente
TACUM	X			Simple	Kms. acumulados hasta la fecha
PRECIO	X			Simple	Precio del viaje
KMS	X			Simple	Kms. a recorrer
DEST		X		Simple	Destino del Viaje
TOTAL	X			Simple	Kms. Tot. incluidos los del viaje actual
DESC	X			Simple	Descuento a realizar
IMP	X			Simple	Importe a Pagar
TOTDESC	X			Simple	Total de Descuentos
TOTIMP	X			Simple	Total recaudado por la Compañía

### Algoritmo: En Pseudocódigo

#### ALGORITMO TURISMO

TOTDESC= 0; TOTIMP= 0;

LEER NYA

**MIENTRAS** NYA <> "ZZZ" **HACER**

**LEER** TACUM, PRECIO, DEST, KMS;



TOTAL= TACUM + KMS

SI TOTAL > 3000

**ENTONCES** DESC= PRECIO \* 0.20;

**SINO** DESC= PRECIO \* 0.05;

**FINSI**

IMP= PRECIO - DESC;

**ESCRIBIR** "Cliente: ", NYA, " Acumulados: ", TACUM;

**ESCRIBIR** "Datos del viaje: \$ ", PRECIO, KMS, "Kms. ", DESTINO;

**ESCRIBIR** "Descuento: \$ ", DESC, "Importe: \$ ", IMP;

TOTIMP= TOTIMP + IMP;

TOTDESC= TOTDESC + DESC;

**LEER** NYA ;

**FIMIENTRAS**

**ESCRIBIR** "Recaudación de la Cia: \$ ", TOTIMP;

**ESCRIBIR** "Total de Descuentos: \$ ", TOTDESC

**FINALGORITMO**

**Seguimiento:**

- a) Datos: "Juan Pérez", 3100, 3500, "Misiones", 800  
"José Díaz", 300, 5000, "Mendoza", 956  
"ZZZ"

NYA	KMREC	PRECIO	KMS	DEST	TOTAL	DESC	IMP	TOTDES	TOTREC
"Juan Pérez"	3100	3500	800	"Misiones"	3900	700	2800	0	0
								700	2800
"José Díaz"	300	5000	956	"Mendoza"	1256	250	4750	950	7550
"ZZZ"									

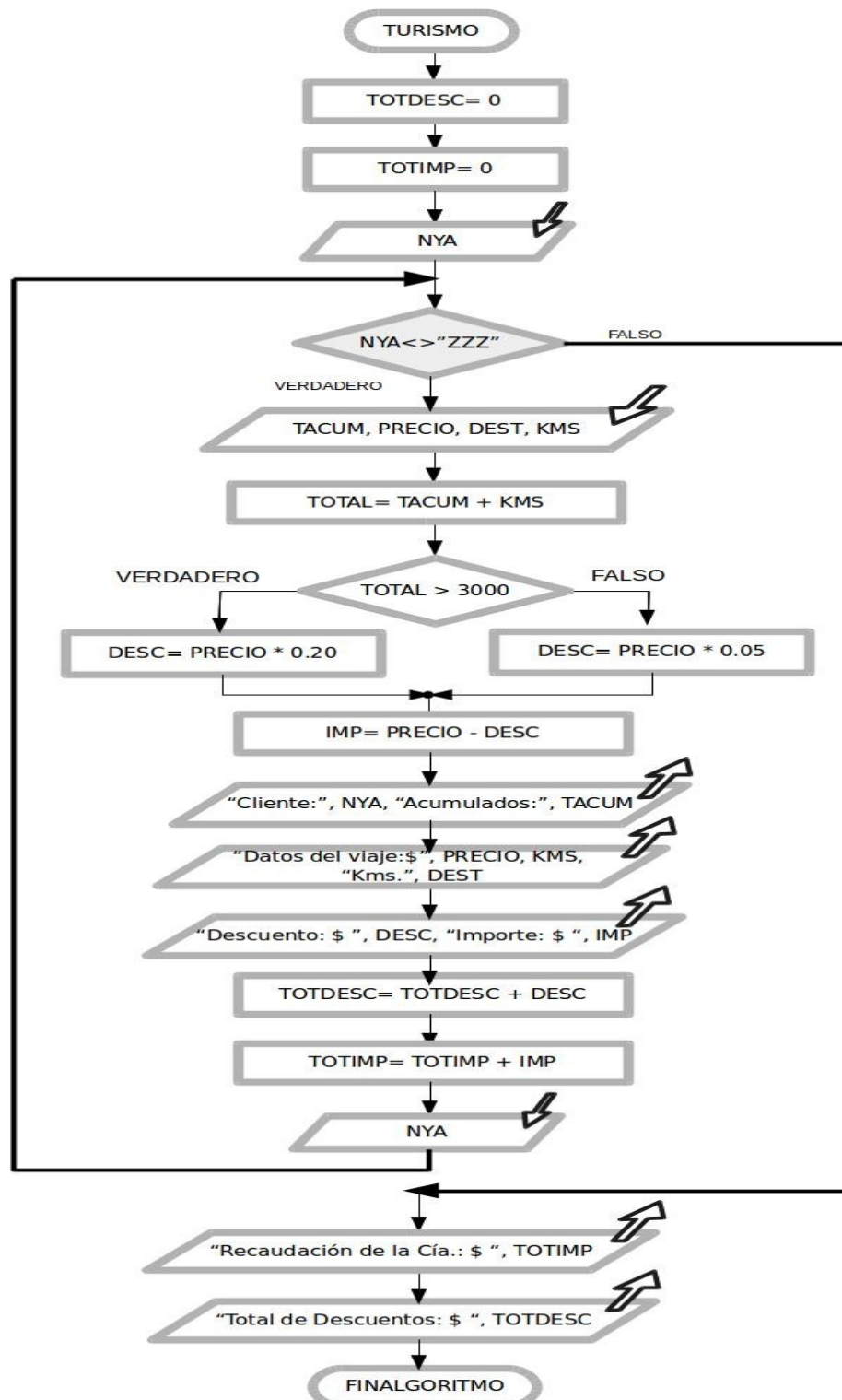
- b) Datos: "ZZZ"

NYA	KMREC	PRECIO	KMS	DEST	TOTAL	DESC	IMP	TOTDES	TOTREC
"ZZZ"								0	0



**Observación:** debemos notar que en este ejemplo se desconoce a priori la cantidad de veces que se repite el proceso. Sin embargo, el mismo algoritmo puede utilizarse para el caso de `ningún` cliente (ejemplo b), para un solo cliente, para 2 (ejemplo a) o para **n** clientes.

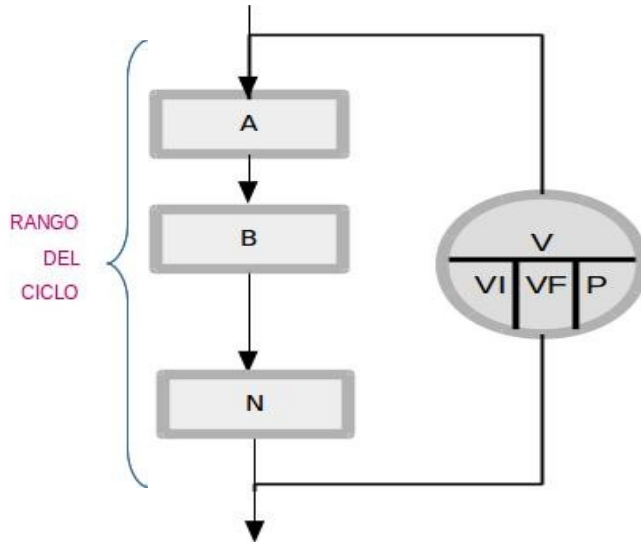
## Algoritmo: En Diagrama de Flujo



## PARA

La estructura de repetición llamada **PARA** nos permite realizar un conjunto de acciones un número determinado de veces.

En diagrama de flujo:



Donde:

**V** : es una variable numérica llamada variable de control

**VI**: es variable o constante numérica, es el valor inicial que toma V

**VF**: variable o constante numérica, es el valor final que toma V

**P** : variable o constante numérica, es el paso, el valor de P puede ser positivo o negativo pero no puede ser cero.

En pseudocódigo la sintaxis equivalente al diagrama es:

```
PARA V DESDE VI HASTA VF CON PASO P HACER
A;
B;
...
N;
FINPARA
```

donde el principio y el fin de la estructura están dados por las palabra claves **PARA** y **FINPARA**.

Tal como en el caso anterior, esta estructura es considerada como una unidad, con un único punto de entrada y un único punto de salida.

A diferencia de las estructuras REPETIR y MIENTRAS, en esta estructura iterativa **el procesador tiene a su cargo la variación de la variable de control** ya que se incrementa su valor **automáticamente** de acuerdo a P, hasta llegar a VF.

El procesador ejecutará las acciones en el siguiente orden:

1. A la variable V le asigna el valor de VI.
2. Ejecuta las acciones previstas (acción 1 hasta acción n).
3. Incrementa la variable V sumándole P.

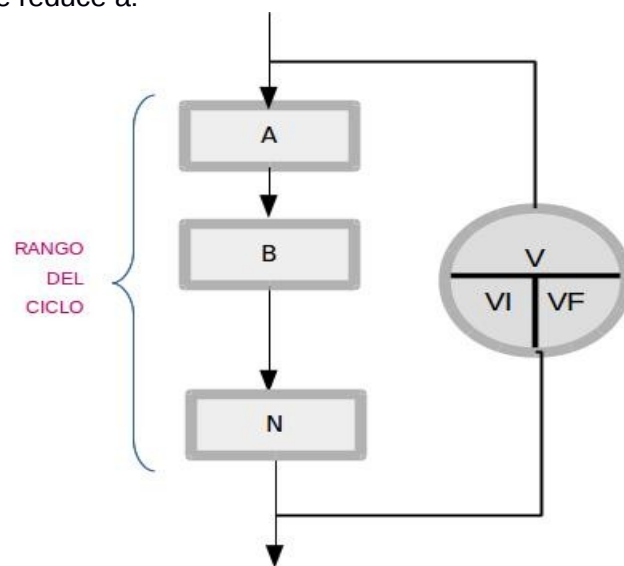
4. Evalúa si el valor de la variable V de control es menor o igual a VF en ese caso continúa a partir de 2.
5. Si la variable V es mayor a VF finaliza la ejecución de la estructura y continúa con la ejecución del algoritmo.



### Observaciones:

- 1- Cuando  $P > 0$  el valor de VF debe ser mayor o igual a VI.
1. Cuando  $P < 0$  el valor de VF debe ser menor o igual a VI
2. Las acciones 1 a n se ejecutan siempre al menos una vez

Cuando  $P = 1$  el formato se reduce a:



**Ejemplo Práctico 3:**

Una compañía de turismo ha definido una política de promoción de sus empresas para lo cual ha fijado descuentos para sus clientes en función de los viajes anteriores realizados. El valor del descuento es del 20% para aquellos clientes que, considerando los viajes realizados en el último año y el que desean realizar, han recorrido más de 3000 Kms. y del 5% para aquellos que no han alcanzado dicha cifra.

Se desea definir un algoritmo que calcule la recaudación diaria de la compañía para sus N clientes. Para ello, por cada cliente se ingresa su nombre y apellido, el total de Kms. recorridos hasta el momento, y los datos del viaje a contratar: precio, destino, Kms. a recorrer. Para cada cliente informar los datos ingresados, el monto del descuento y el monto a pagar. Informar además el total recaudado y el total de descuentos efectuados por la Compañía, con leyendas alusivas.

El valor N se ingresa como primer dato.

**Ambiente y Seguimiento**, ver ejemplo 1 (REPETIR)

**Algoritmo: en Pseudocódigo****ALGORITMO TURISMO2**

TOTDESC = 0;

TOTIMP = 0;

LEER N;

**PARA** CONT **DESDE** 0 **HASTA** N **HACER**

**LEER** NYA, TACUM, PRECIO, DESTINO, KMS;

    TOTAL= TACUM + KMS ;

**SI** TOTAL > 3000

**ENTONCES** DESC= PRECIO \* 0.20

**SINO**       DESC= PRECIO \* 0.05

**FINSI**

    IMP= PRECIO - DESC;

**ESCRIBIR** "Cliente: ", NYA, "Acumulados: ", TACUM;

**ESCRIBIR** "Datos del viaje: \$ ", PRECIO, KMS, "Kms. ", DESTINO;

**ESCRIBIR** "Descuento: \$ ", DESC, "Importe: \$ ", IMP;

    TOTIMP= TOTIMP + IMP;

    TOTDESC= TOTDESC + DESC;

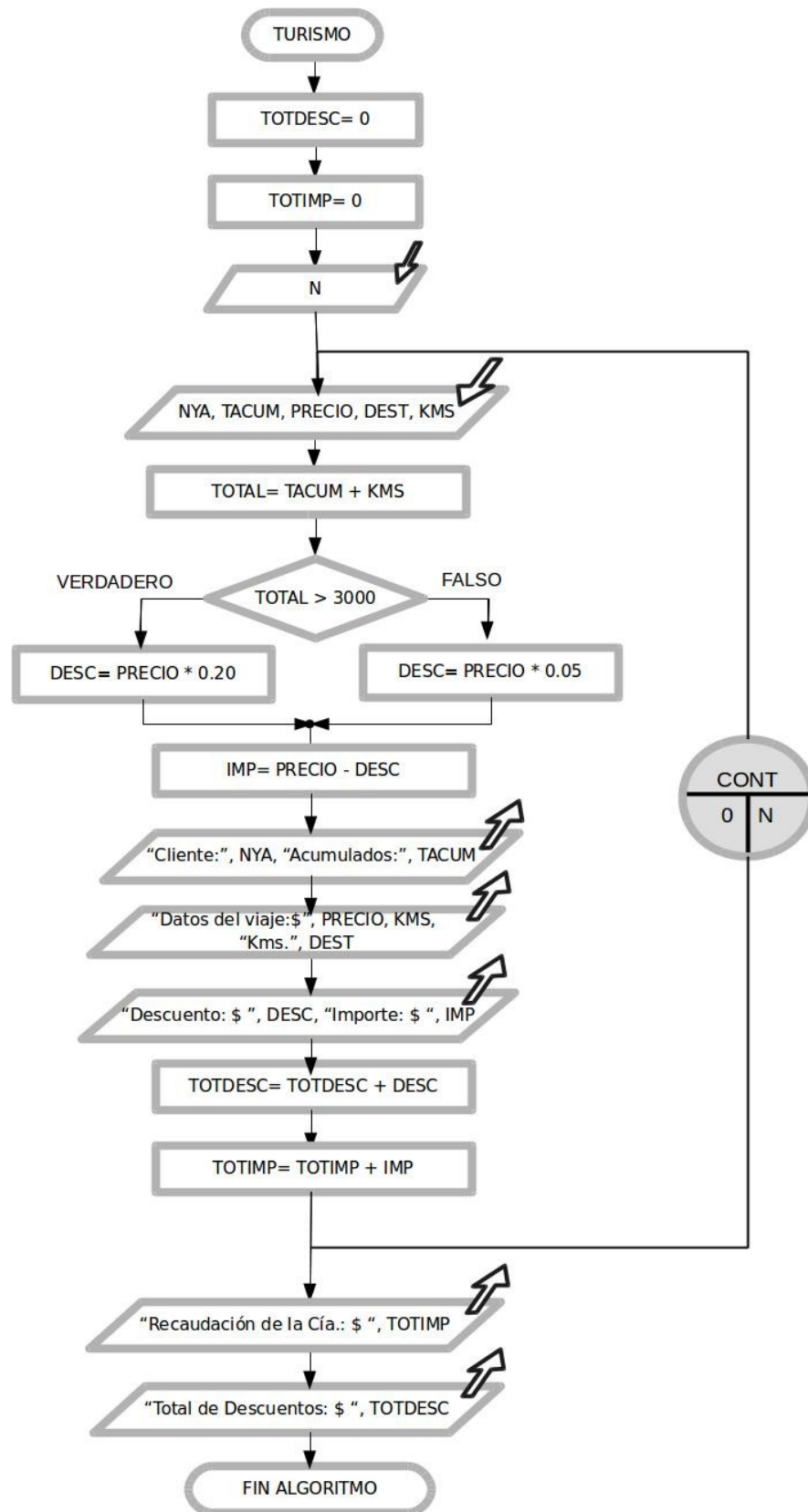
**FINPARA**

**ESCRIBIR** "Recaudación de la Cia: \$ ", TOTIMP;

**ESCRIBIR** "Total de Descuentos: \$ ", TOTDESC;

**FINALGORITMO**

### Algoritmo: en Diagrama de Flujo



## ANIDAMIENTO DE ESTRUCTURAS DE CONTROL

En el diseño de algoritmos, es común la utilización de estructuras lógicas de control complejas, las que se basan en la combinación de estructuras elementales.

Por ejemplo, los **condicionales anidados**, constituyen una estructura de control compleja, dado que es una estructura de decisión dentro de otra estructura de decisión. De la misma manera, los **ciclos anidados**, pues son estructuras de iteración incluidas dentro de otras siguiendo las mismas reglas: la estructura interna debe estar totalmente incluida en la estructura externa.

A continuación, se presentan algunos ejemplos:

Condicionales anidados
<pre> SI condicion 1     ENTONCES A         SI condicion 2             ENTONCES B             SINO                 C; D         FINSI     SINO         SI condicion 3             ENTONCES M             SINO                 R         FINSI     W FINSI </pre>
Condicionales anidados, dentro de un ciclo iterativo
<pre> MIENTRAS condicion 1 HACER     SI condicion 2         ENTONCES             SI condicion 3                 ENTONCES B             FINSI         SINO             A     FINSI FINMIENTRAS </pre>
Ciclos iterativos Para anidados
<pre> Para I desde 1 hasta 10 hacer     A     Para J desde 1 hasta 5 hacer </pre>

B;  
C  
Finpara  
Finpara

## ESTRATEGIAS

En la unidad 1, definimos ESTRATEGIA y ubicamos a la **Definición de una estrategia** como una etapa dentro de la **Resolución de problemas**, que debe realizarse previa a la **Definición de algoritmos**.

Comenzaremos a aplicar estos conceptos y los vistos en esa unidad referidos a diseño descendente, en la resolución de problemas que se nos plantean.

### Ejemplo Práctico 4:

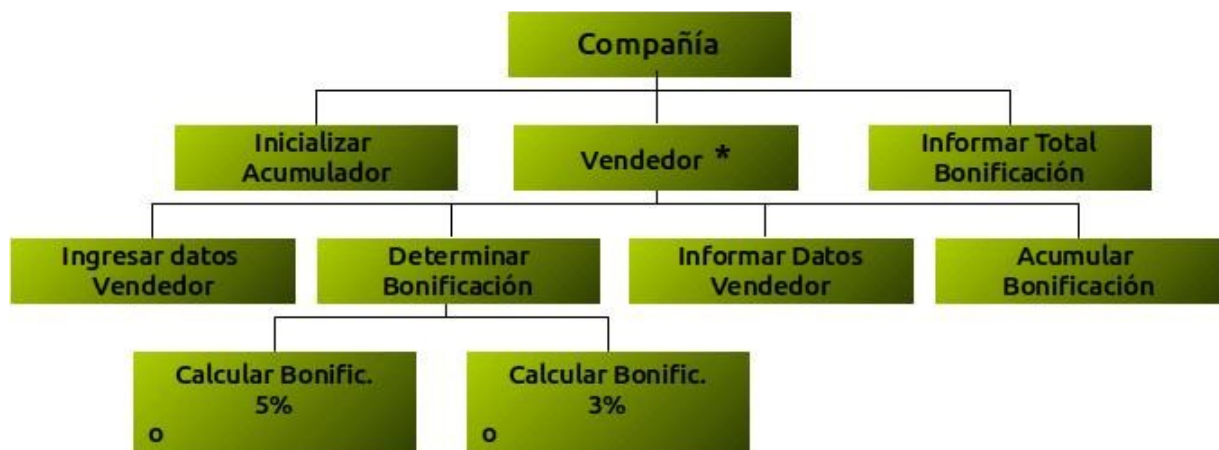
*Una compañía paga a cada uno de sus vendedores una bonificación anual basada en el sueldo del vendedor y en su total de ventas durante el año. El valor de la bonificación es del 3 % del total de las ventas si las ventas son inferiores a 4 veces el sueldo del vendedor, o del 5 % si son mayores o iguales a ese monto.*

*Realizar un algoritmo que ingrese por cada vendedor el nombre y apellido, el sueldo y el total de ventas. El fin de datos se produce al ingresar como nombre y apellido 'ZZZZ'.*

*Para cada empleado informar el nombre y apellido y la bonificación correspondiente, y además informar el total que debe pagar la compañía en concepto de bonificaciones, con leyenda indicativa.*

Para resolver este problema, luego de realizar el análisis del mismo, debemos plantear la estrategia, definiendo las etapas que deben seguirse para arribar a la solución del problema, indicando **QUÉ** debe hacerse y no el **CÓMO**.

En el enunciado planteado una alternativa de Estrategia es:



### Notación:



Representa un proceso iterativo. Debajo de él pueden graficarse los módulos que forman el ciclo.



Representa una alternativa cuya ejecución dependerá de una condición.