

## El Pseudo-código

Las características de este pseudolenguaje fueron propuestas en 2001 por el responsable de la asignatura Fundamentos de Programación de la carrera de Ingeniería Informática de la FICH-UNL. Las premisas son:

- Sintaxis sencilla
- Manejo de las estructuras básicas de control
- Solo 3 tipos de datos básicos: numérico, carácter /cadenas de caracteres y lógico (verdadero-falso).
- Estructuras de datos: arreglos

Todo algoritmo en pseudocódigo tiene la siguiente estructura general:

```
Proceso SinTitulo  
  accion 1;  
  accion 1;  
  .  
  .  
  .  
  accion n;  
FinProceso
```

Comienza con la palabra clave **Proceso** seguida del nombre del programa, luego le sigue una secuencia de instrucciones y finaliza con la palabra **FinProceso**.

Una secuencia de instrucciones es una lista de una o más instrucciones, cada una terminada en punto y coma.

Las acciones incluyen operaciones de entrada y salida, asignaciones de variables, condicionales si-entonces o de selección múltiple y/o lazos mientras, repetir o para.

### Variables

Una variable en un algoritmo computacional es una posición de memoria donde se puede almacenar información. Por ejemplo, si un programa debe obtener el área de un triángulo, seguramente la base del mismo y su altura se carguen en memoria en dos variables para poder realizar el cálculo. El resultado, probablemente también se asigne en una variable luego del cálculo para luego informarlo al usuario. Como su nombre lo indica, el valor almacenado en una variable puede ir variando a medida que el programa avanza. En un pseudocódigo el concepto es similar. Una variable representa un lugar donde guardar cierta información.

En un algoritmo o programa se hace referencia a una variable mediante un **identificador** (el nombre de la variable). Un identificador debe comenzar con letras, y puede contener solo letras, números y el guion bajo. No puede contener ni espacios ni operadores, ni coincidir con una palabra reservada o función del lenguaje, para no generar ambigüedad. Ejemplos de identificadores válidos son: **A, B, C, Lado1, Total, Nombre\_y\_Apellido, DireccionCorreo, etc.** En la mayoría de los lenguajes reales los nombres de variables no pueden contener acentos, ni diéresis, ni eñes. En PSeInt, esto se permite si se activa la Sintaxis Flexible (ver Opciones del PSeudocódigo).

En algunos lenguajes se puede guardar cualquier información en cualquier variable, mientras en otros las variables solo pueden guardar cierto tipo de información. En PSeInt las variables tienen un tipo de dato asociado, por lo que durante la ejecución del algoritmo una variable deberá guardar datos siempre del mismo tipo. Por ejemplo, si una variable se utiliza para guardar números, no puede utilizarse luego para guardar texto. Este tipo se puede declarar explícitamente con la palabra clave **Definir**, o se puede dejar que el intérprete intente deducirlo a partir de los datos que se guardan en la misma y la forma en que se la utiliza en el algoritmo. Si utiliza el perfil de lenguaje por defecto (Flexible), la definición explícita es opcional, pero se puede configurar el lenguaje para que la misma sea obligatoria.

Hay dos formas de crear una variable y/o asignarle un valor: **la lectura y la asignación**. Si se lee o asigna un valor en una variable que no existe, esta se crea. Si la variable ya existía, esta toma el nuevo valor, perdiendo el viejo. Por esto se dice que la asignación y la lectura son acciones destructivas (aunque se debe notar que en la asignación pueden intervenir más de una variable, y solo se destruye el contenido previo de la que se encuentra a la izquierda del signo de asignación). Una vez inicializada, la variable puede utilizarse en cualquier expresión (para realizar un cálculo en una asignación, para mostrar en pantalla, como condición en una estructura de control, etc.)

## **Tipos de Datos**

- **Tipos Simples: Numérico, Lógico, Carácter.**
- **Estructuras de Datos: Arreglos.**

## **Definición explícita de variables.**

Los tipos de datos simples se determinan automáticamente cuando se crean las variables. Las dos acciones que pueden crear una variable son la lectura (LEER) y la asignación(<-). Por ejemplo, la asignación "A<-0;" está indicando implícitamente que la variable A será una variable numérica. Una vez determinado el tipo de dato, deberá permanecer constante durante toda la ejecución del proceso; en caso contrario el proceso será interrumpido.

Se puede definir el tipo de una variable antes de utilizarla. Esta definición puede ser obligatoria u opcional dependiendo de la configuración del lenguaje.

Los arreglos son estructuras homogéneas (es decir, que todos sus elementos son del mismo tipo simple de dato). El tipo de sus elementos se determina cuando se utiliza alguno de ellos de igual forma que para las variables de tipos simples.

Pueden ser mono o multidimensionales, pero deben ser dimensionados antes de ser utilizados por primera vez mediante la directiva **DIMENSION** (ejemplo: "DIMENSION A[10,20];").

Los identificadores, o nombres de variables, deben constar sólo de letras, números y/o guion bajo (\_), comenzando siempre con una letra.

## Tipos de Datos Simples

**Existen tres tipos de datos básicos:**

**Numérico:** números, tanto enteros como reales. Para separar decimales se utiliza el punto. **Ejemplos:** 12 23 0 -2.3 3.14

**Lógico:** solo puede tomar dos valores: **VERDADERO o FALSO.**

**Carácter:** caracteres o cadenas de caracteres encerrados entre comillas (pueden ser dobles o simples). **Ejemplos** 'hola', "hola mundo" ,'123', 'FALSO', 'etc'

Opcionalmente, se puede declarar una variable numérica como entera con la instrucción **DEFINIR**. En este caso, todo valor no entero que se lea o asigne a la misma será truncado.

## Operadores y Funciones

Este pseudolenguaje dispone de un conjunto básico de operadores y funciones que pueden ser utilizados para la construcción de expresiones más o menos complejas.

Las siguientes tablas exhiben la totalidad de los operadores de este lenguaje reducido:

<i>Operador</i>	<i>Significado</i>	<i>Ejemplo</i>
<i>Relacionales</i>		
>	Mayor que	3>2
<	Menor que	'ABC'<'abc'

=	Igual que	4=3
<=	Menor o igual que	'a'<='b'
>=	Mayor o igual que	4>=5
<i>Logicos</i>		
& ó Y	Conjunción (y).	(7>4) & (2=1) //falso
ó O	Disyunción (o).	(1=1   2=1) //verdadero
~ ó NO	Negación (no).	~(2<5) //falso
<i>Algebraicos</i>		
+	Suma	total <- cant1 + cant2
-	Resta	stock <- disp - venta
*	Multiplicación	area <- base * altura
/	División	porc <- 100 * parte / total
^	Potenciación	sup <- 3.41 * radio ^ 2
% ó MOD	Módulo (resto de la división entera)	resto <- num MOD div

La jerarquía de los operadores matemáticos es igual a la del álgebra, aunque puede alterarse mediante el uso de paréntesis.

A continuación se listan las funciones integradas disponibles:

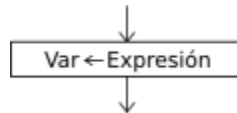
<i>Función</i>	<i>Significado</i>
RC(X)	Raíz Cuadrada de X
ABS(X)	Valor Absoluto de X
LN(X)	Logaritmo Natural de X
EXP(X)	Función Exponencial de X
SEN(X)	Seno de X
COS(X)	Coseno de X
ATAN(X)	Arcotangente de X
TRUNC(X)	Parte entera de X
REDON(X)	Entero más cercano a X
AZAR(X)	Entero aleatorio entre 0 y X-1

### **Algunas Observaciones**

- Se pueden introducir comentarios luego de una instrucción, o en líneas separadas, mediante el uso de la doble barra ( //). Todo lo que precede a //, hasta el fin de la línea, no será tomado en cuenta al interpretar el algoritmo.
- Nótese que no puede haber instrucciones fuera del programa, aunque si comentarios.
- Las estructuras no secuenciales pueden anidarse. Es decir, pueden contener otras adentro, pero la estructura contenida debe comenzar y finalizar dentro de la contenedora.
- Los identificadores, o nombres de variables, deben constar sólo de letras y números, comenzando siempre con una letra, y no pueden ser palabras reservadas (como para, mientras, y, no, etc...)
- Las constantes de tipo carácter se escriben entre comillas ( " ).
- En las constantes numéricas, el punto ( . ) es el separador decimal.
- Las constantes lógicas son *Verdadero* y *Falso*.

## ESTRUCTURAS DE CONTROL SECUENCIAL

### Asignación

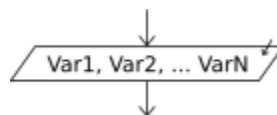


La instrucción de asignación permite almacenar un valor en una variable.

<variable> <- <expresión> ;

Al ejecutarse la asignación, primero se evalúa la **expresión** de la derecha y luego se asigna el resultado a la **variable** de la izquierda. El tipo de la variable y el de la expresión deben coincidir.

### Entradas



La instrucción **Leer** permite ingresar información desde el ambiente.

Leer <variable> , <variable2> , ... ,<variableN> ;

Esta instrucción lee  $N$  valores desde el ambiente (en este caso el teclado) y los asigna a las  $N$  *variables* mencionadas. Pueden incluirse una o más variables, por lo tanto el comando leerá uno o más valores.

## Salidas



La instrucción **Escribir** permite mostrar valores al ambiente.

Escribir <expr1> , <expr2> , ... , <exprN> ;

Esta instrucción imprime al ambiente (en este caso en la pantalla) los valores obtenidos de evaluar  $N$  *expresiones*. Dado que puede incluir una o más expresiones, mostrará uno o más valores.

**Ejemplo1).**- Suponga que un individuo desea invertir su capital en un banco y desea saber cuánto dinero ganara después de un mes si el banco paga a razón de 2% mensual.

## Pseudocódigo:

Proceso CapitalInvertido

    Imprimir "Introduce el Capital a Invertir:";

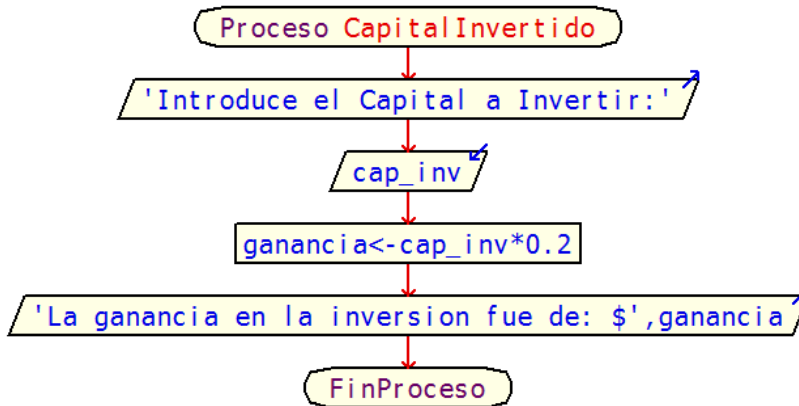
    Leer cap\_inv;

    ganancia=cap\_inv\*0.2;

    Imprimir "La ganancia en la inversion fue de: \$",ganancia;

FinProceso

## Diagrama de Flujo



### Ejecución del Pseudocodigo:

```

PSeInt - Ejecutando proceso CAPITALINVERTIDO
*** Ejecución Iniciada. ***
Introduce el Capital a Invertir:
> 4500
La ganancia en la inversion fue de: $900
*** Ejecución Finalizada. ***
  
```

**Ejemplo2).**- Un vendedor recibe un sueldo base más un 10% extra por comisión de sus ventas, el vendedor desea saber cuánto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

### Pseudocodigo:

Proceso ComisionesVendedor

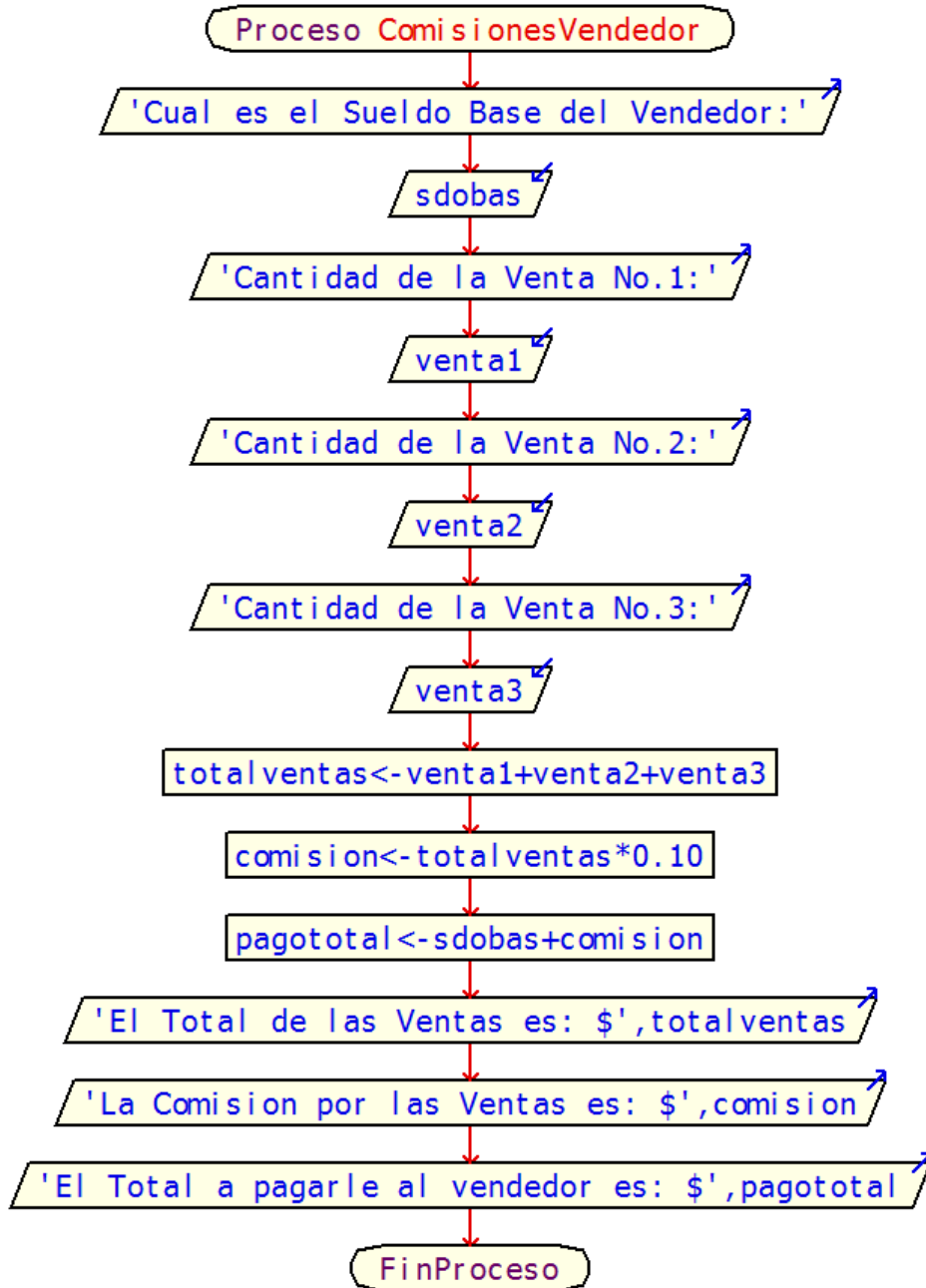
- Imprimir "Cual es el Sueldo Base del Vendedor:";
- Leer sdoBas;
- Imprimir "Cantidad de la Venta No.1:";
- Leer venta1;
- Imprimir "Cantidad de la Venta No.2:";
- Leer venta2;
- Imprimir "Cantidad de la Venta No.3:";

```

Leer venta3;
totalventas=venta1+venta2+venta3;
comision=totalventas*0.10;
pagototal=sdobas+comisión;
Imprimir "El Total de las Ventas es: $",totalventas;
Imprimir "La Comision por las Ventas es: $",comision;
Imprimir "El Total a pagarle al vendedor es: $",pagototal;
FinProceso

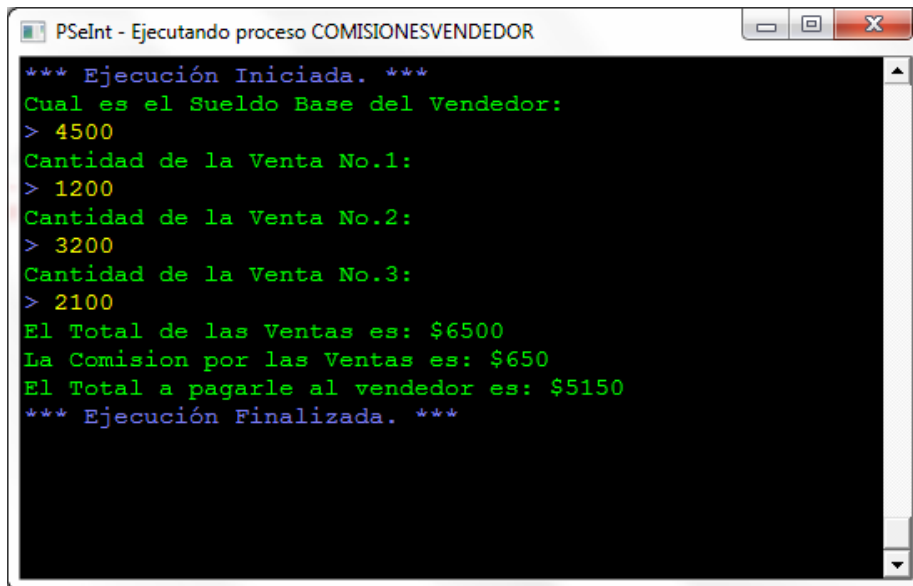
```

**Diagrama de Flujo:**





## Ejecución del Pseudocódigo:



```
PSeInt - Ejecutando proceso COMISIONESVENDEDOR
*** Ejecución Iniciada. ***
Cual es el Sueldo Base del Vendedor:
> 4500
Cantidad de la Venta No.1:
> 1200
Cantidad de la Venta No.2:
> 3200
Cantidad de la Venta No.3:
> 2100
El Total de las Ventas es: $6500
La Comision por las Ventas es: $650
El Total a pagarle al vendedor es: $5150
*** Ejecución Finalizada. ***
```

**Ejemplo3).**- Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra.

## Pseudocódigo:

Proceso DescuentoCompra

Imprimir "Total de la Compra:";

Leer totalcompra;

descuento=totalcompra\*0.15;

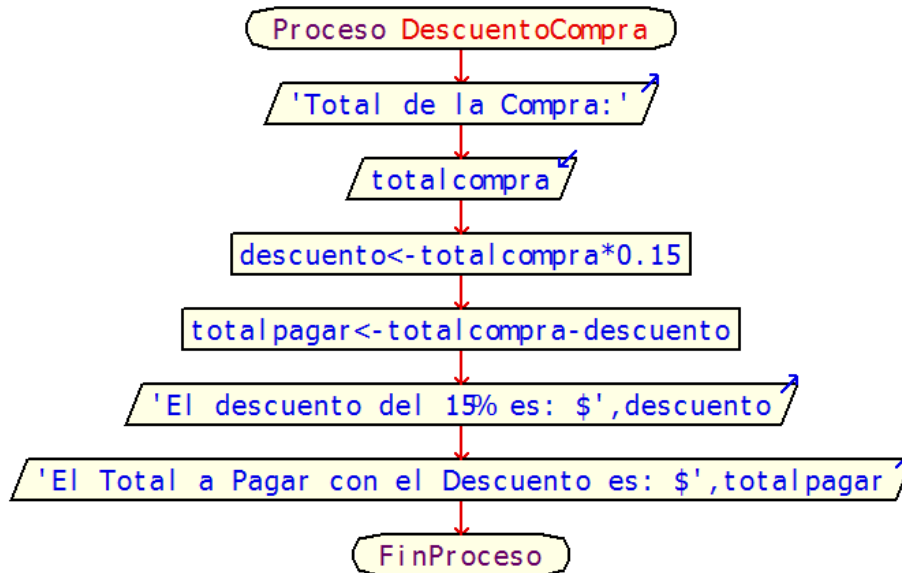
totalpagar=totalcompra-descuento;

Imprimir "El descuento del 15% es: \$",descuento;

Imprimir "El Total a Pagar con el Descuento es: \$",totalpagar;

FinProceso

## Diagrama de Flujo:



### Ejecución del Pseudocódigo:

```

PSeInt - Ejecutando proceso DESCUENTOCOMPRA
*** Ejecución Iniciada. ***
Total de la Compra:
> 12350
El descuento del 15% es: $1852.5
El Total a Pagar con el Descuento es: $10497.5
*** Ejecución Finalizada. ***
  
```

**Ejemplo4).**- Un alumno desea saber cuál será su calificación final en la materia de Algoritmos. Dicha calificación se compone de los siguientes porcentajes:

55% del promedio de sus tres calificaciones parciales.

30% de la calificación del examen final.

15% de la calificación de un trabajo final.

## **Pseudocodigo:**

Proceso CalificacionFinal

Imprimir "Cual es la Calificacion del 1er. Examen Parcial:";

Leer calpar1;

Imprimir "Cual es la Calificacion del 2do. Examen Parcial:";

Leer calpar2;

Imprimir "Cual es la Calificacion del 3er. Examen Parcial:";

Leer calpar3;

Imprimir "Cual es la Calificacion del Examen Final:";

Leer calexafin;

Imprimir "Cual es la Calificacion del Trabajo Final:";

Leer caltrafin;

$promcalpar=(calpar1+calpar2+calpar3)/3;$

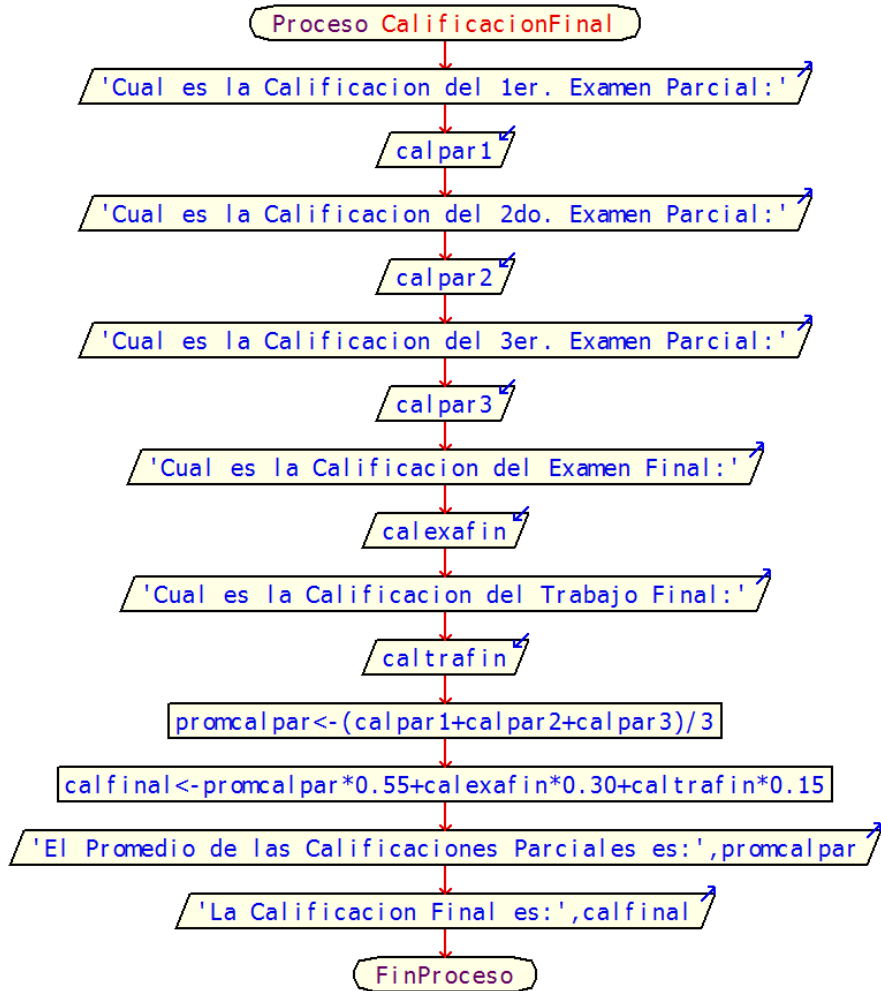
$calfinal=promcalpar*0.55+calexafin*0.30+caltrafin*0.15;$

Imprimir "El Promedio de las Calificaciones Parciales es:",promcalpar;

Imprimir "La Calificacion Final es:",calfinal;

FinProceso

## **Diagrama de Flujo:**



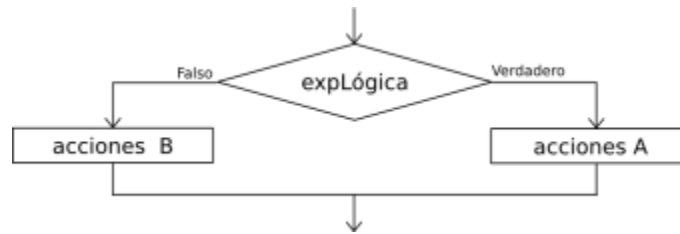
### Ejecución del Pseudocodigo:

```

*** Ejecución Iniciada. ***
Cual es la Calificacion del 1er. Examen Parcial:
> 90
Cual es la Calificacion del 2do. Examen Parcial:
> 87
Cual es la Calificacion del 3er. Examen Parcial:
> 78
Cual es la Calificacion del Examen Final:
> 86
Cual es la Calificacion del Trabajo Final:
> 95
El Promedio de las Calificaciones Parciales es:85
La Calificacion Final es:86.8
*** Ejecución Finalizada. ***
  
```

## Estructuras de Control Selectivas:

### Condiciona Si-Entonces



La secuencia de instrucciones ejecutadas por la instrucción **Si-Entonces-Sino** depende del valor de una condición lógica.

```
Si <condición>
  Entonces
    <instrucciones>
  Sino
    <instrucciones>
FinSi
```

Al ejecutarse esta instrucción, se evalúa la **condición** y se ejecutan las instrucciones que correspondan: las instrucciones que le siguen al **Entonces** si la condición es verdadera, o las instrucciones que le siguen al **Sino** si la condición es falsa. La condición debe ser una expresión lógica, que al ser evaluada retorna **Verdadero** o **Falso**.

La cláusula **Entonces** debe aparecer siempre, pero la cláusula **Sino** puede no estar. En ese caso, si la condición es falsa no se ejecuta ninguna instrucción y la ejecución del programa continúa con la instrucción siguiente.

Ejemplo1).- Un hombre desea saber cuánto dinero se genera por concepto de intereses sobre la cantidad que tiene en inversión en el banco si el banco paga 2% al mes. El decidirá reinvertir los intereses siempre y cuando estos excedan a \$700 y desea saber la cantidad de dinero que tendrá finalmente en su cuenta.

Pseudocódigo:

Proceso InvierteInteres

  imprimir "Cuanto es la Cantidad a Invertir: \$"

  Leer cantinv

  interes=cantinv\*0.02

  Imprimir "Los Intereses Ganados son:",interes

  si interes>700 Entonces

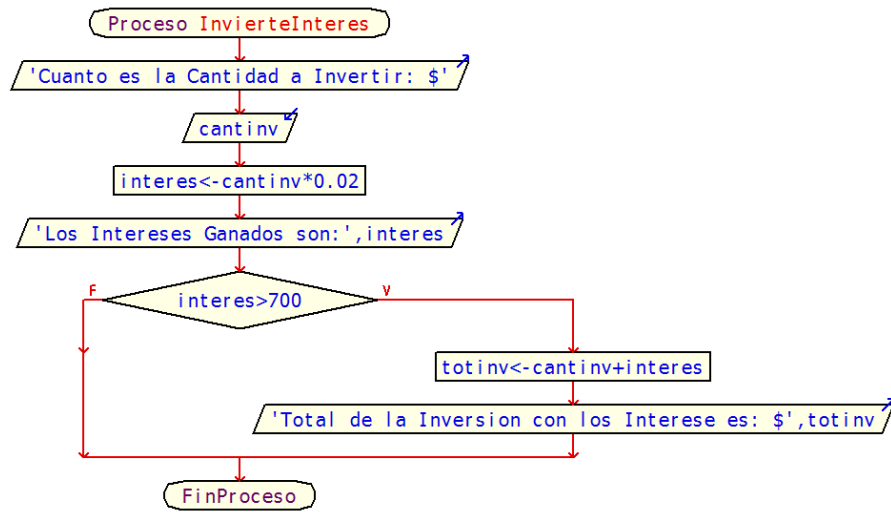
    totinv=cantinv+interes

  Imprimir "Total de la Inversion con los Interese es: \$",totinv

  FinSi

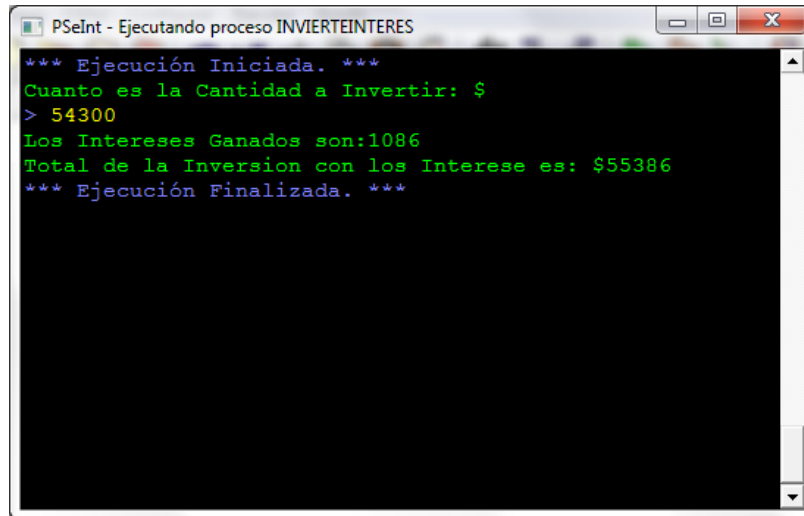
FinProceso

Diagrama de Flujo:



Ejecución del Pseudocodigo:

```
PSeInt - Ejecutando proceso INVIERTEINTERES
*** Ejecución Iniciada. ***
Cuanto es la Cantidad a Invertir: $
> 22500
Los Intereses Ganados son:450
*** Ejecución Finalizada. ***
```



```
PSeInt - Ejecutando proceso INVIERTEINTERES
*** Ejecución Iniciada. ***
Cuanto es la Cantidad a Invertir: $
> 54300
Los Intereses Ganados son:1086
Total de la Inversion con los Interese es: $55386
*** Ejecución Finalizada. ***
```

Ejemplo2).- Encontrar el Mayor de 3 números diferentes proporcionados como datos de entrada

### Pseudocodigo:

Proceso Mayor3Numeros

Imprimir "Introduce el 1er. Numero:"

Leer n1

Imprimir "Introduce el 2do. Numero:"

Leer n2

Imprimir "Introduce el 3er. Numero:"

Leer n3

si  $n1 > n2$  Entonces

si  $n1 > n3$  Entonces

Imprimir "El numero Mayor es:",n1

Sino

Imprimir "El numero Mayor es:",n3

FinSi

Sino

si  $n2 > n3$  Entonces

Imprimir "El numero Mayor es:",n2

Sino

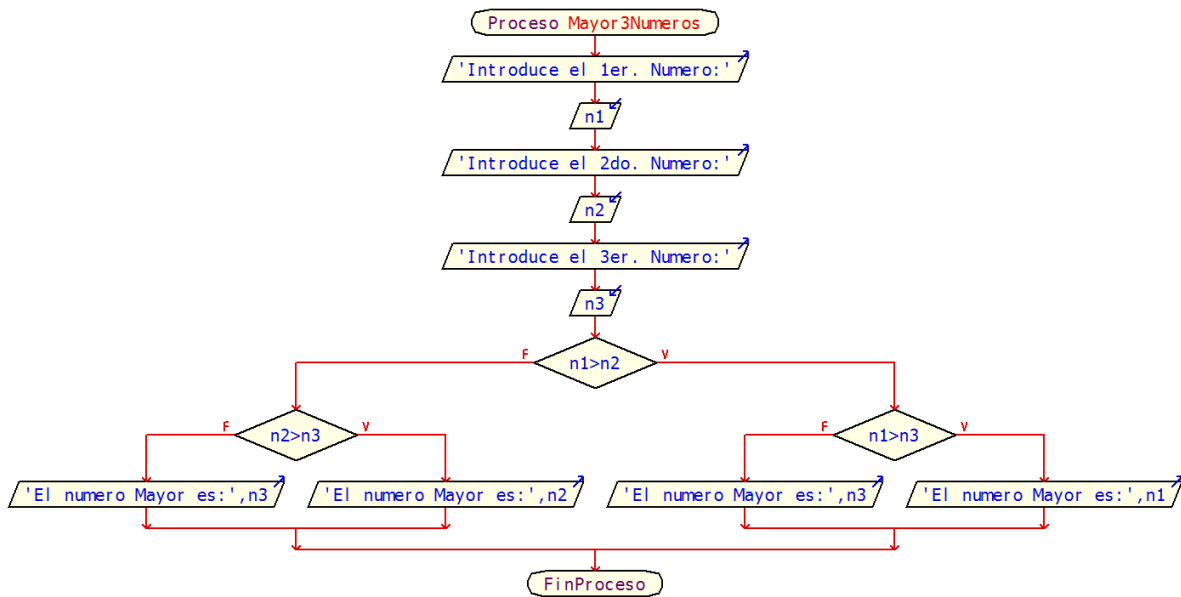
Imprimir "El numero Mayor es:",n3

FinSi

FinSi

FinProceso

### Diagrama de Flujo:



### Ejecución del Pseudocodigo:

```

*** Ejecución Iniciada. ***
Introduce el 1er. Numero:
> 89
Introduce el 2do. Numero:
> 12
Introduce el 3er. Numero:
> 72
El numero Mayor es:89
*** Ejecución Finalizada. ***
  
```

Ejemplo3).- Se desea saber cuánto tendrá que pagar una persona por la compra de un artículo en base a las siguientes condiciones. Si compra menos de 5 del mismo artículo se le hará un descuento del 10% sobre el total de su compra. Si compra 5 o más, pero menos de 10 se le hace un 15% de descuento. Si compra más de 10 pero menos de 15 se le hace un 20% de descuento y en caso contrario se le hará un 25% de descuento sobre su compra.

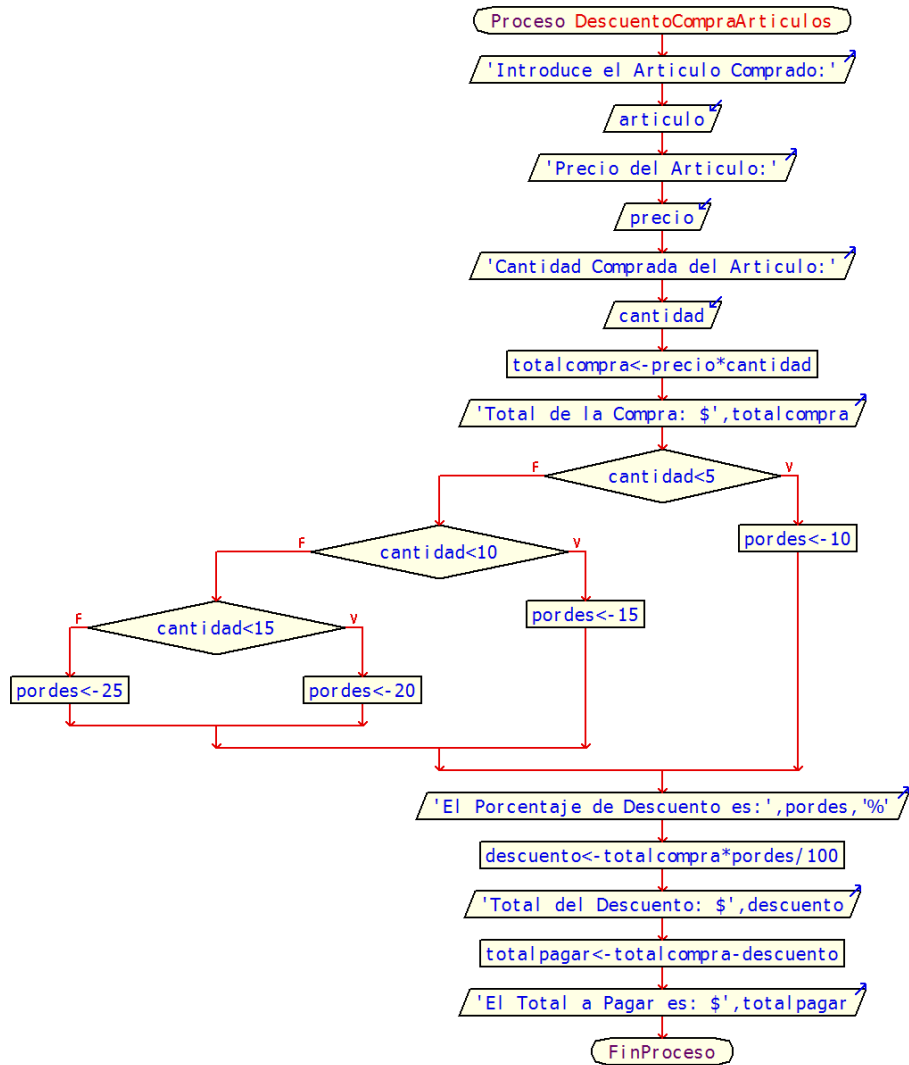
Pseudocodigo:

Proceso DescuentoCompraArticulos



```
Imprimir "Introduce el Articulo Comprado:"
Leer articulo
Imprimir "Precio del Articulo:"
Leer precio
Imprimir "Cantidad Comprada del Articulo:"
Leer cantidad
totalcompra=precio*cantidad
Imprimir "Total de la Compra: $",totalcompra
si cantidad<5 Entonces
    pordes=10
Sino
    si cantidad<10 Entonces
        pordes=15
    Sino
        si cantidad<15 Entonces
            pordes=20
        Sino
            pordes=25
        FinSi
    FinSi
FinSi
FinSi
Imprimir "El Porcentaje de Descuento es:",pordes,"% "
descuento=totalcompra*pordes/100
Imprimir "Total del Descuento: $",descuento
totalpagar=totalcompra-descuento
Imprimir "El Total a Pagar es: $",totalpagar
FinProceso
```

**Diagrama de Flujo:**

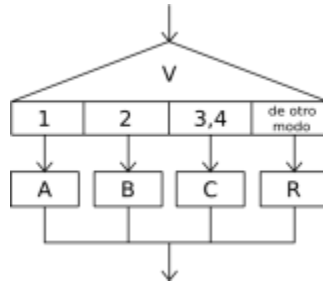


### Ejecución del Pseudocodigo:

```

PSeInt - Ejecutando proceso DESCUENTOCOMPRAARTICULOS
*** Ejecución Iniciada. ***
Introduce el Artículo Comprado:
> Computadoras
Precio del Artículo:
> 9550
Cantidad Comprada del Artículo:
> 10
Total de la Compra: $95500
El Porcentaje de Descuento es:20%
Total del Descuento: $19100
El Total a Pagar es: $76400
*** Ejecución Finalizada. ***
  
```

## Selección Múltiple



La secuencia de instrucciones ejecutada por una instrucción **Según** depende del valor de una variable numérica.

```
Segun <variable> Hacer
  <número1>: <instrucciones>
  <número2>,<número3>: <instrucciones>
  <...>
  De Otro Modo: <instrucciones>
FinSegun
```

Esta instrucción permite ejecutar opcionalmente varias acciones posibles, dependiendo del valor almacenado en una variable de tipo numérico. Al ejecutarse, se evalúa el contenido de la variable y se ejecuta la secuencia de instrucciones asociada con dicho valor.

Cada opción está formada por uno o más números separados por comas, dos puntos y una secuencia de instrucciones. Si una opción incluye varios números, la secuencia de instrucciones asociada se debe ejecutar cuando el valor de la variable es uno de esos números.

Opcionalmente, se puede agregar una opción final, denominada **De Otro Modo**, cuya secuencia de instrucciones asociada se ejecutará sólo si el valor almacenado en la variable no coincide con ninguna de las opciones anteriores.

Ejemplo1).- Hacer un pseudocódigo que permita calcular el área de cualquiera de las siguientes figuras geométricas:

1. Cuadrado
2. Rectángulo
3. Triangulo
4. Circulo

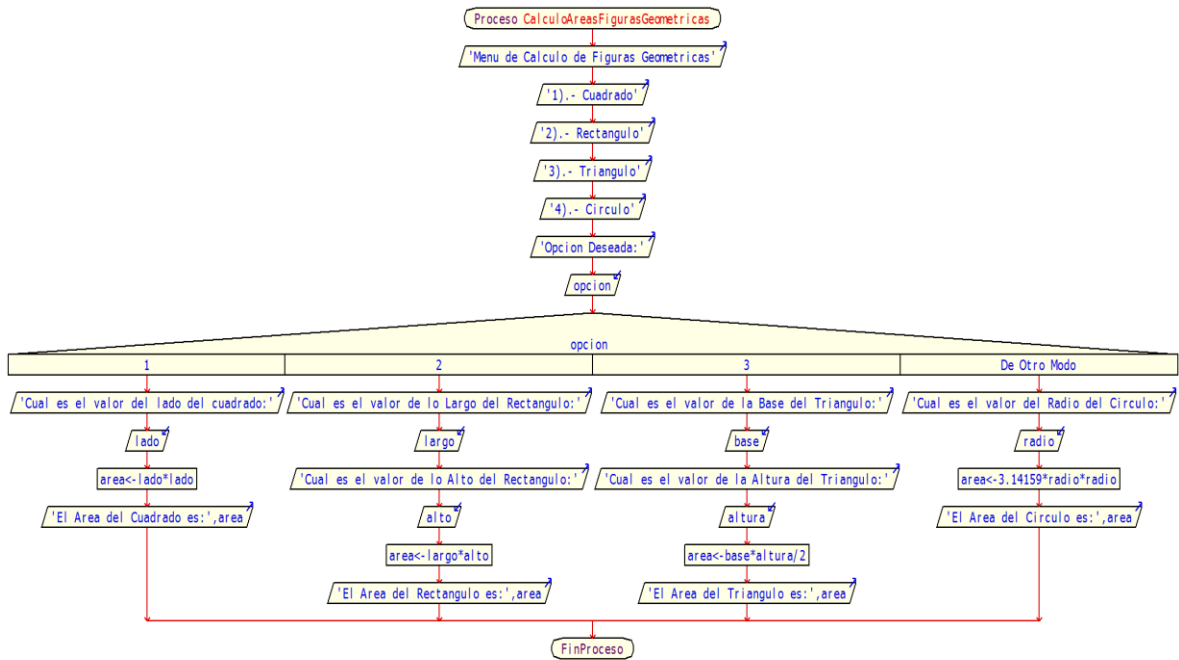
### Pseudocódigo:

```
Proceso CalculoAreasFigurasGeometricas
  Imprimir "Menu de Calculo de Figuras Geometricas"
```

Imprimir "1).- Cuadrado"  
Imprimir "2).- Rectangulo"  
Imprimir "3).- Triangulo"  
Imprimir "4).- Circulo"  
Imprimir "Opcion Deseada:"  
Leer opcion  
Segun opcion Hacer  
1:  
    Imprimir "Cual es el valor del lado del cuadrado:"  
    Leer lado  
     $area=lado*lado$   
    Imprimir "El Area del Cuadrado es:",area  
2:  
    Imprimir "Cual es el valor de lo Largo del Rectangulo:"  
    Leer largo  
    Imprimir "Cual es el valor de lo Alto del Rectangulo:"  
    Leer alto  
     $area=largo*alto$   
    Imprimir "El Area del Rectangulo es:",area  
3:  
    Imprimir "Cual es el valor de la Base del Triangulo:"  
    Leer base  
    Imprimir "Cual es el valor de la Altura del Triangulo:"  
    Leer altura  
     $area=base*altura/2$   
    Imprimir "El Area del Triangulo es:",area  
De Otro Modo:  
    Imprimir "Cual es el valor del Radio del Circulo:"  
    Leer radio  
     $area=3.14159*radio*radio$   
    Imprimir "El Area del Circulo es:",area  
Fin Segun

FinProceso

**Diagrama de Flujo:**

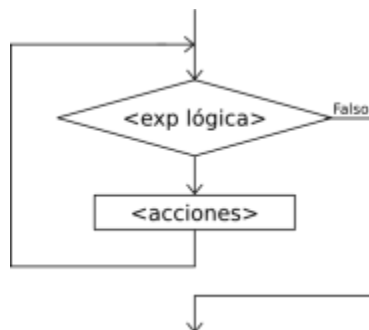


### Ejecución del Pseudocodigo:

```

PSeInt - Ejecutando proceso CALCULOAREASFIGURASGEOMETRICAS
*** Ejecución Iniciada. ***
Menu de Calculo de Figuras Geometricas
1).- Cuadrado
2).- Rectangulo
3).- Triangulo
4).- Circulo
Opcion Deseada:
> 3
Cual es el valor de la Base del Triangulo:
> 12
Cual es el valor de la Altura del Triangulo:
> 5
El Area del Triangulo es:30
*** Ejecución Finalizada. ***
  
```

### Estructura de Control del Ciclo Mientras-hacer



La instrucción **Mientras** ejecuta una secuencia de instrucciones mientras una condición sea verdadera.

```
Mientras <condición> Hacer
  <instrucciones>
FinMientras
```

Al ejecutarse esta instrucción, la condición es evaluada. Si la condición resulta verdadera, se ejecuta una vez la secuencia de instrucciones que forman el cuerpo del ciclo. Al finalizar la ejecución del cuerpo del ciclo se vuelve a evaluar la condición y, si es verdadera, la ejecución se repite. Estos pasos se repiten mientras la condición sea verdadera.

Note que las instrucciones del cuerpo del ciclo pueden no ejecutarse nunca, si al evaluar por primera vez la condición resulta ser falsa.

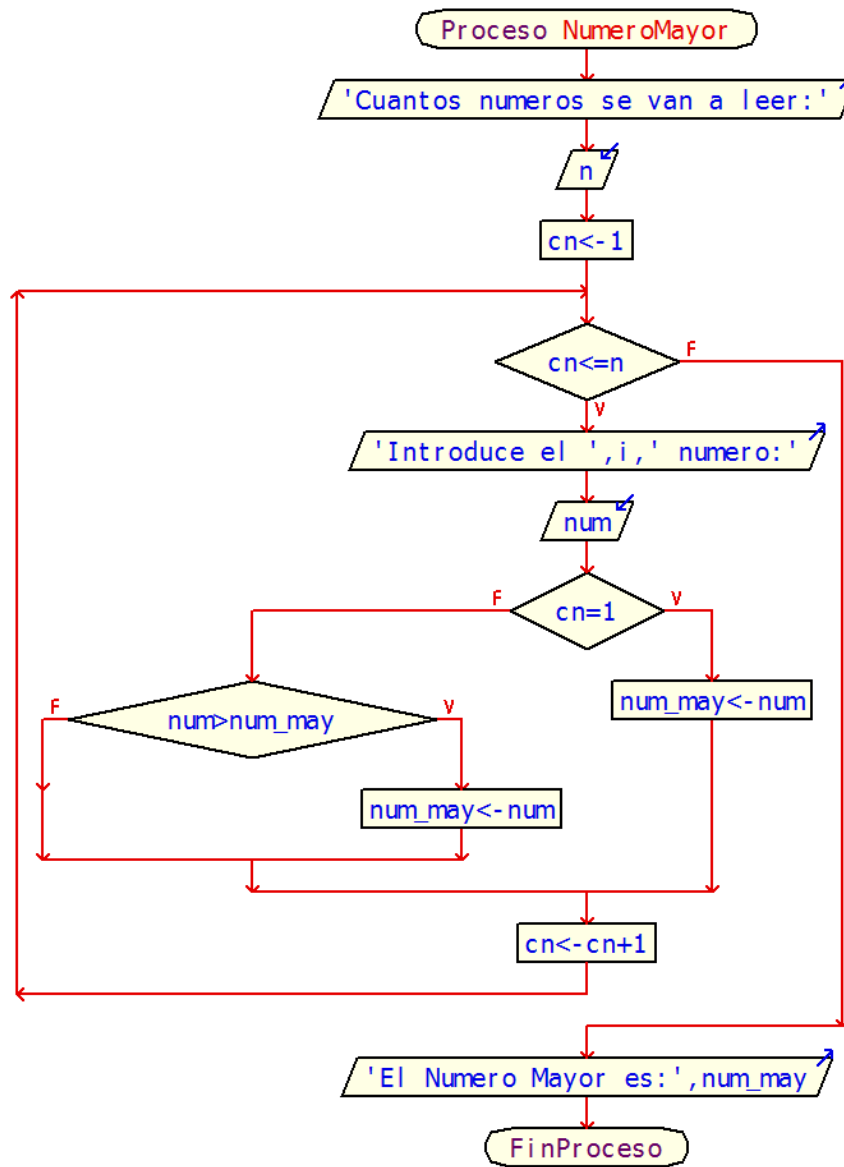
Si la condición siempre es verdadera, al ejecutar esta instrucción se produce un ciclo infinito. A fin de evitarlo, las instrucciones del cuerpo del ciclo deben contener alguna instrucción que modifique la o las variables involucradas en la condición, de modo que ésta sea falsificada en algún momento y así finalice la ejecución del ciclo.

Ejemplo1).- Encontrar el número mayor de una serie de números proporcionados como datos de entrada.

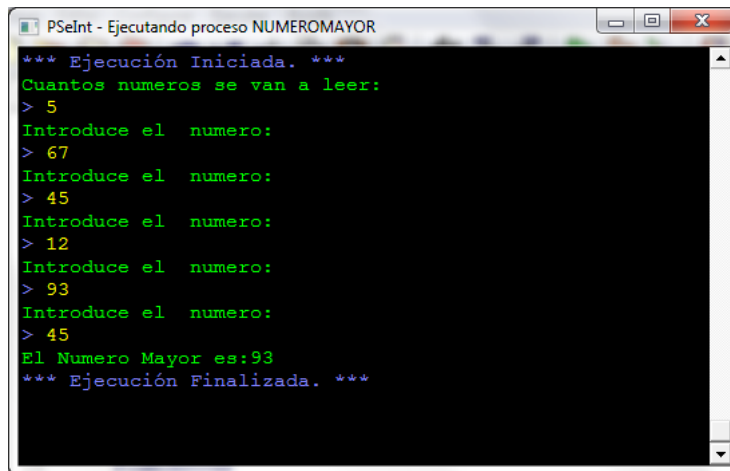
### **Pseudocódigo:**

```
Proceso NumeroMayor
  Imprimir "Cuantos numeros se van a leer:"
  Leer n
  cn=1
  Mientras cn<=n hacer
    Imprimir "Introduce el ",i," numero:"
    Leer num
    si cn=1 Entonces
      num_may=num
    Sino
      si num>num_may Entonces
        num_may=num
      FinSi
    FinSi
    cn=cn+1
  FinMientras
  Imprimir "El Numero Mayor es:",num_may
FinProceso
```

## Diagrama de Flujo:



## Ejecución del Pseudocodigo:



```
*** Ejecución Iniciada. ***
Cuantos numeros se van a leer:
> 5
Introduce el numero:
> 67
Introduce el numero:
> 45
Introduce el numero:
> 12
Introduce el numero:
> 93
Introduce el numero:
> 45
El Numero Mayor es:93
*** Ejecución Finalizada. ***
```

Ejemplo2).- Determinar el porcentaje de números pares e impares de una serie de números proporcionados como datos de entrada

### Pseudocodigo:

Proceso PorcentajeParesImpares

Imprimir "Introduce la cantidad de numeros a leer:"

Leer n

cn=1

cpar=0

cimpar=0

Mientras cn<=n Hacer

Imprimir "Introduce el ",i," numero:"

Leer num

si num%2=0 Entonces

cpar=cpar+1

Sino

cimpar=cimpar+1

FinSi

cn=cn+1

FinMientras

porpar=cpar/n\*100

porimpar=cimpar/n\*100

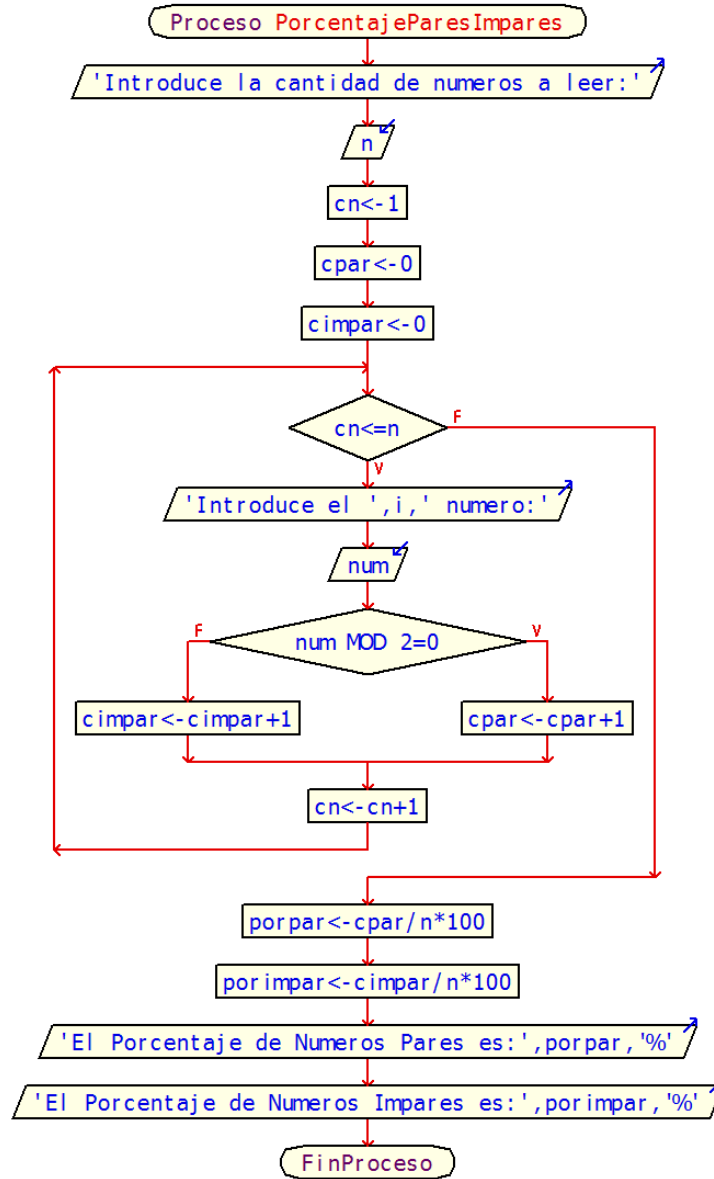
Imprimir "El Porcentaje de Numeros Pares es:",porpar,"%"

Imprimir "El Porcentaje de Numeros Impares es:",porimpar,"%"

FinProceso

Diagrama de Flujo:



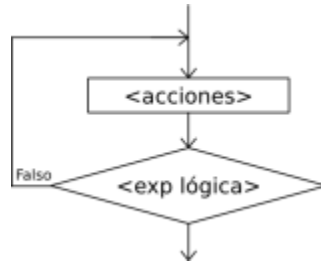


### Ejecución del Pseudocódigo:

```

PSeInt - Ejecutando proceso PORCENTAJEPARESIMPARES
*** Ejecución Iniciada. ***
Introduce la cantidad de numeros a leer:
> 5
Introduce el numero:
> 98
Introduce el numero:
> 12
Introduce el numero:
> 43
Introduce el numero:
> 53
Introduce el numero:
> 68
El Porcentaje de Numeros Pares es:60%
El Porcentaje de Numeros Impares es:40%
*** Ejecución Finalizada. ***
  
```

## Estructura de Control de Ciclos Repetir-hasta que



La instrucción **Repetir-Hasta Que** ejecuta una secuencia de instrucciones hasta que la condición sea verdadera.

```
Repetir
  <instrucciones>
Hasta Que <condición>
```

Al ejecutarse esta instrucción, la secuencia de instrucciones que forma el cuerpo del ciclo se ejecuta una vez y luego se evalúa la condición. Si la condición es falsa, el cuerpo del ciclo se ejecuta nuevamente y se vuelve a evaluar la condición.

Esto se repite hasta que la condición sea verdadera.

Note que, dado que la condición se evalúa al final, las instrucciones del cuerpo del ciclo serán ejecutadas al menos una vez.

Además, a fin de evitar ciclos infinitos, el cuerpo del ciclo debe contener alguna instrucción que modifique la o las variables involucradas en la condición de modo que en algún momento la condición sea verdadera y se finalice la ejecución del ciclo.

Ejemplo1).- Obtener el promedio de una serie de números proporcionados como datos de entrada

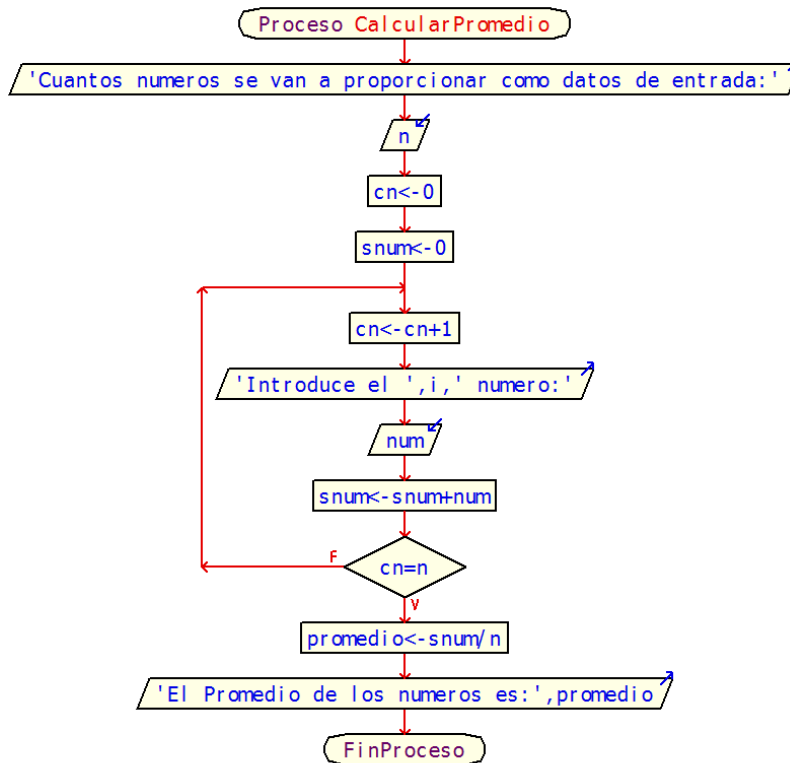
### Pseudocódigo:

Proceso CalcularPromedio

```
  Escribir 'Cuantos numeros se van a proporcionar como datos de entrada:'
  Leer n
  cn<-0
  snum<-0
  Repetir
    cn<-cn+1
    Escribir 'Introduce el ',i,' numero:'
    Leer num
    snum<-snum+num
  Hasta Que cn=n
```

```
promedio<-snum/n
Escribir 'El Promedio de los numeros es:',promedio
FinProceso
```

### Diagrama de Flujo:



### Ejecución del Pseudocodigo:

```
PSeInt - Ejecutando proceso CALCULARPROMEDIO
*** Ejecución Iniciada. ***
Cuantos numeros se van a proporcionar como datos de entrada:
> 5
Introduce el numero:
> 67
Introduce el numero:
> 34
Introduce el numero:
> 12
Introduce el numero:
> 44
Introduce el numero:
> 23
El Promedio de los numeros es:36
*** Ejecución Finalizada. ***
```

Ejemplo2).- Una compañía de seguros tiene contratados a n vendedores. Cada uno hace tres ventas a la semana. Su política de pagos es que un vendedor recibe un sueldo base, y un 10% extra por comisiones de sus ventas. El gerente de su compañía desea saber cuánto dinero obtendrá en la semana cada vendedor por

concepto de comisiones por las tres ventas realizadas, y cuanto tomando en cuenta su sueldo base y sus comisiones.

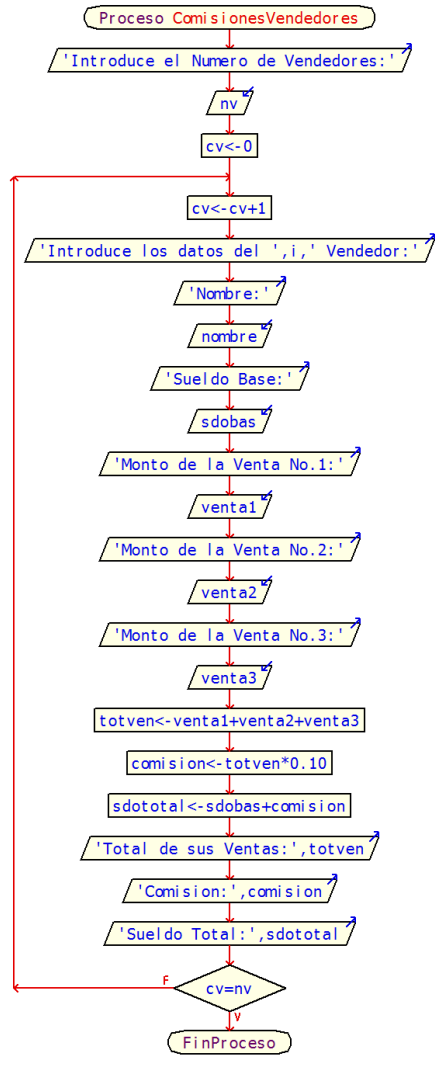
### **Pseudocodigo:**

Proceso ComisionesVendedores

```
Imprimir "Introduce el Numero de Vendedores:"
Leer nv
cv=0
Hacer
    cv=cv+1
    Imprimir "Introduce los datos del ",i," Vendedor:"
    Imprimir "Nombre:"
    leer nombre
    Imprimir "Sueldo Base:"
    Leer sdobas
    Imprimir "Monto de la Venta No.1:"
    leer venta1
    Imprimir "Monto de la Venta No.2:"
    leer venta2
    Imprimir "Monto de la Venta No.3:"
    leer venta3
    totven=venta1+venta2+venta3
    comision=totven*0.10
    sdototal=sdobas+comision
    Imprimir "Total de sus Ventas:",totven
    Imprimir "Comision:",comision
    Imprimir "Sueldo Total:",sdototal
Hasta Que cv=nv
```

FinProceso

### **Diagrama de Flujo:**

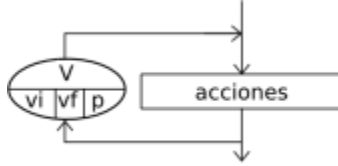


### Ejecución del Pseudocodigo:

```

PSeInt - Ejecutando proceso COMISIONESVENDEDORES
*** Ejecución Iniciada. ***
Introduce el Numero de Vendedores:
> 2
Introduce los datos del Vendedor:
Nombre:
> Felipe de Jesus
Sueldo Base:
> 4500
Monto de la Venta No.1:
> 3200
Monto de la Venta No.2:
> 2500
Monto de la Venta No.3:
> 1250
Total de sus Ventas:6950
Comision:695
Sueldo Total:5195
Introduce los datos del Vendedor:
Nombre:
> |
  
```

## Estructura de Control de Ciclos Para



La instrucción **Para** ejecuta una secuencia de instrucciones un número determinado de veces.

```
Para <variable> <- <inicial> Hasta <final> ( Con Paso <paso> ) Hacer
  <instrucciones>
FinPara
```

Al ingresar al bloque, la variable **<variable>** recibe el valor **<inicial>** y se ejecuta la secuencia de instrucciones que forma el cuerpo del ciclo. Luego se incrementa la variable **<variable>** en **<paso>** unidades y se evalúa si el valor almacenado en **<variable>** superó al valor **<final>**. Si esto es falso se repite hasta que **<variable>** supere a **<final>**. Si se omite la cláusula **Con Paso<paso>**, la variable **<variable>** se incrementará en 1.

Ejemplo1).- Leer 50 calificaciones de un grupo de alumnos. Calcule y escriba el porcentaje de reprobados. Tomando en cuenta que la calificación mínima aprobatoria es de 70

### Pseudocodigo:

Proceso PorcentajeDeReprobados

Imprimir "No. de Alumnos:"

Leer na

crepro=0

para i=1 hasta na Hacer

Imprimir "Introduce la Calificacion del ",i," alumno:"

Leer calificacion

si calificacion<70 Entonces

crepro=crepro+1

FinSi

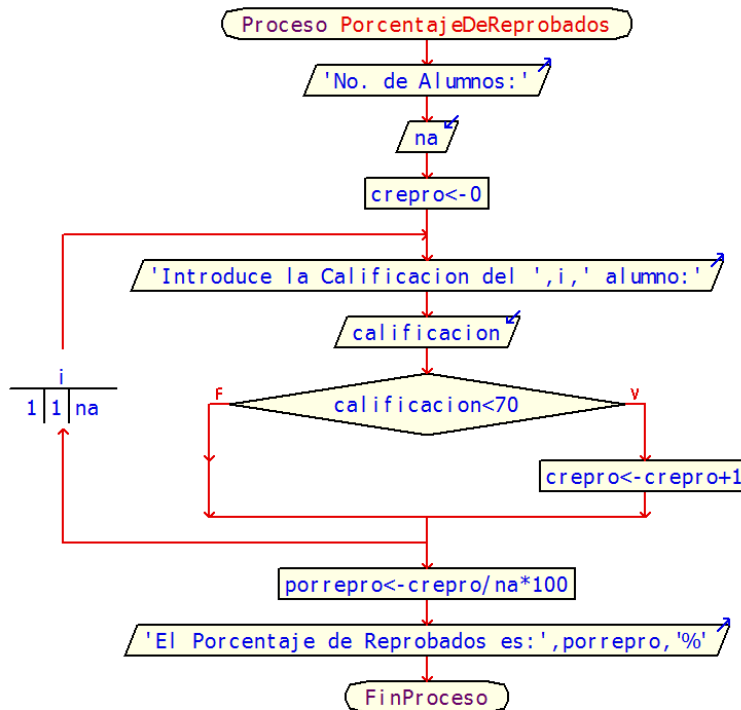
FinPara

porrepro=crepro/na\*100

Imprimir "El Porcentaje de Reprobados es:",porrepro,"%"

FinProceso

Diagrama de Flujo:



Ejecución del Pseudocodigo:

```
PSeInt - Ejecutando proceso PORCENTAJEDEREPROBADOS
*** Ejecución Iniciada. ***
No. de Alumnos:
> 5
Introduce la Calificación del 1 alumno:
> 90
Introduce la Calificación del 2 alumno:
> 65
Introduce la Calificación del 3 alumno:
> 72
Introduce la Calificación del 4 alumno:
> 60
Introduce la Calificación del 5 alumno:
> 87
El Porcentaje de Rebados es:40%
*** Ejecución Finalizada. ***
```

The screenshot shows a terminal window titled 'PSeInt - Ejecutando proceso PORCENTAJEDEREPROBADOS'. The output shows the execution of the pseudocode. It starts with '\*\*\* Ejecución Iniciada. \*\*\*'. The user is prompted for the number of students, entering '5'. Then, for each student, the user is prompted to enter their grade. The grades entered are 90, 65, 72, 60, and 87. Finally, the program outputs 'El Porcentaje de Rebados es:40%' and ends with '\*\*\* Ejecución Finalizada. \*\*\*'.

Ejemplo2).- Leer los n votos otorgados a los 3 candidatos a gobernador e imprimir el número del candidato ganador y su cantidad de votos.

### **Pseudocodigo:**

Proceso CandidatoGanador

    Imprimir "Introduce la cantidad de votos:"

    Leer nv

    cvc1=0

    cvc2=0

    cvc3=0

    para cv=1 hasta nv Hacer

        Imprimir "Introduce el valor del voto No.",cv,":"

        Leer voto

        si voto=1 Entonces

            cvc1=cvc1+1

        Sino

        si voto=2 Entonces

            cvc2=cvc2+1

        Sino

        cvc3=cvc3+1

        FinSi

    FinSi

    FinPara

    imprimir "Total de Votos del Candidato 1:",cvc1

    Imprimir "Total de Votos del Candidato 2:",cvc2

    Imprimir "Total de Votos del Candidato 3:",cvc3

    si cvc1>cvc2 Entonces

        si cvc1>cvc3 Entonces

            Imprimir "El Ganador de las Votaciones es el Candidato 1"

        Sino

            Imprimir "El Ganador de las Votaciones es el Candidato 3"

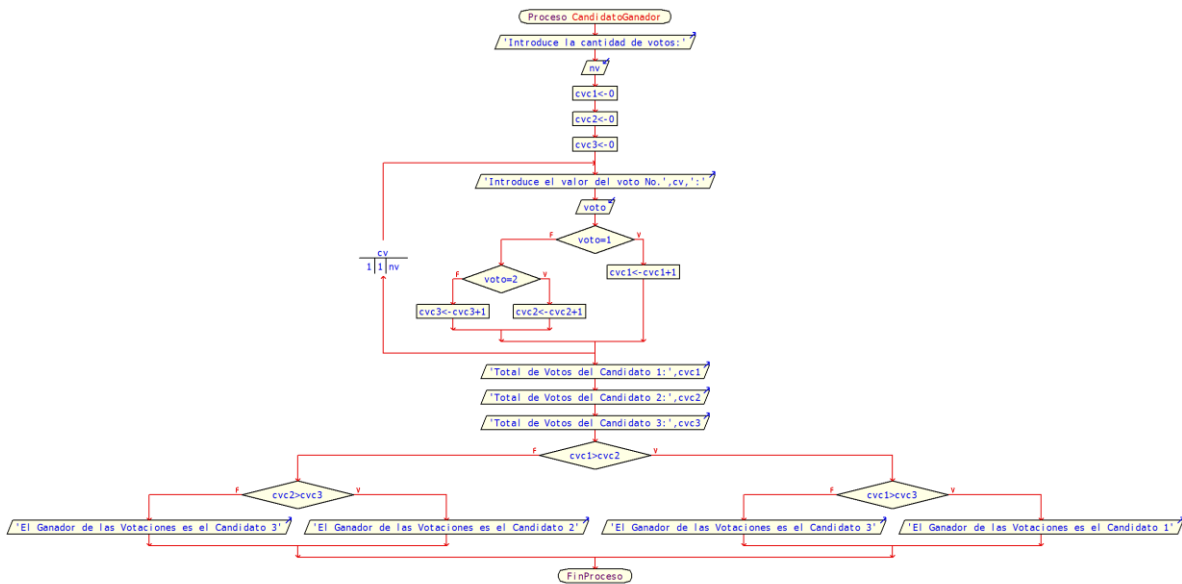


```

FinSi
Sino
si cvc2>cvc3 Entonces
    Imprimir "El Ganador de las Votaciones es el Candidato 2"
Sino
    Imprimir "El Ganador de las Votaciones es el Candidato 3"
FinSi
FinSi
FinProceso

```

**Diagrama de Flujo:**



**Ejecucion del Pseudocodigo:**

```
PSeInt - Ejecutando proceso CANDIDATOGANADOR
Introduce la cantidad de votos:
> 5
Introduce el valor del voto No.1:
> 1
Introduce el valor del voto No.2:
> 2
Introduce el valor del voto No.3:
> 1
Introduce el valor del voto No.4:
> 1
Introduce el valor del voto No.5:
> 3
Total de Votos del Candidato 1:3
Total de Votos del Candidato 2:1
Total de Votos del Candidato 3:1
El Ganador de las Votaciones es el Candidato 1
*** Ejecución Finalizada. ***
```

## Funciones/Subprocesos en PSeInt

Si el perfil de lenguaje seleccionado lo permite, se pueden declarar nuevas funciones o subprocessos en un algoritmo en PSeudoCódigo. La sintaxis para ello es la siguiente:

```
SubProceso variable_de_retorno <- nombre_de_la_funcion ( argumento_1,
argumento_2, ... )
```

```
    accion 1;
```

```
    accion 1;
```

```
    .
```

```
    .
```

```
    .
```

```
    accion n;
```

```
FinSubproceso
```

Comienza con la palabra clave **SubProceso** (o **Función**, son equivalentes) seguida de la **variable de retorno**, el **signo de asignación**, el **nombre del subprocesso**, y finalmente, la **lista de argumentos** entre paréntesis.

Existen variantes para esta estructura. Si la función no retorna ningún valor, pueden omitirse el identificador `variable_de_retorno` y el signo de asignación, es decir, colocar directamente el nombre y los argumentos a continuación de la palabra clave `SubProceso`. Si el subprocesso no recibe ningún valor pueden colocarse los paréntesis vacíos u omitirse, finalizando la primer línea con el nombre del subprocesso. Las reglas para los nombres de subprocessos, variables

de retorno y argumentos son las mismas que para cualquier identificador en pseudocódigo.

Además, opcionalmente pueden agregarse las **palabras claves Por Valor o Por Referencia** para indicar el tipo de paso de parámetro de cada argumento. Si no se indica, los **arreglos se pasan por referencia, las demás expresiones por valor**. El paso de parámetros por referencia implica que si la función modifica el argumento, se modificará en realidad la variable que se utilizó en la llamada, mientras que el paso de parámetros por valor implica que la función opera con una copia de la variable (o el resultado de la expresión) que se utilizó en la llamada, por lo que las modificaciones que aplique la función no se verán reflejadas fuera de la misma.

Para **invocar a la función** se debe utilizar su nombre y entre paréntesis los parámetros, que podrán ser expresiones sólo si el tipo de paso de parámetros es por referencia. Una llamada puede ser en sí una instrucción, pero si la función retorna algún valor, también puede utilizarse como operando dentro de una expresión

Ejemplo1).- Hacer una función que permita calcular el factorial de un número  $n$  pasado como parámetro y que posteriormente se utilice esa misma función para poder calcular combinaciones y permutaciones.

### **Pseudocódigo:**

```
SubProceso combina <- Combinaciones ( n,r )
```

```
    combina=factorial(n)/(factorial(r)*factorial(n-r))
```

```
Fin SubProceso
```

```
SubProceso permuta <- Permutaciones ( n,r )
```

```
    permuta=factorial(n)/factorial(n-r)
```

```
Fin SubProceso
```

```
SubProceso fact <- factorial ( n )
```

```
    fact=1
```

```
    cn=1
```

```
    Mientras cn<=n Hacer
```

fact=fact\*cn

cn=cn+1

Fin Mientras

Fin SubProceso

Proceso CalculoFactorial

Imprimir "Numero a calcular el factorial:"

Leer n

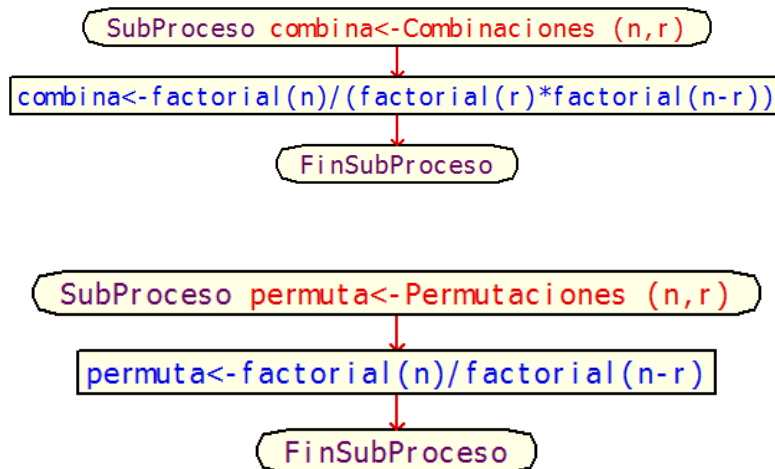
Imprimir "El Factorial de ",n," es: ",factorial(n)

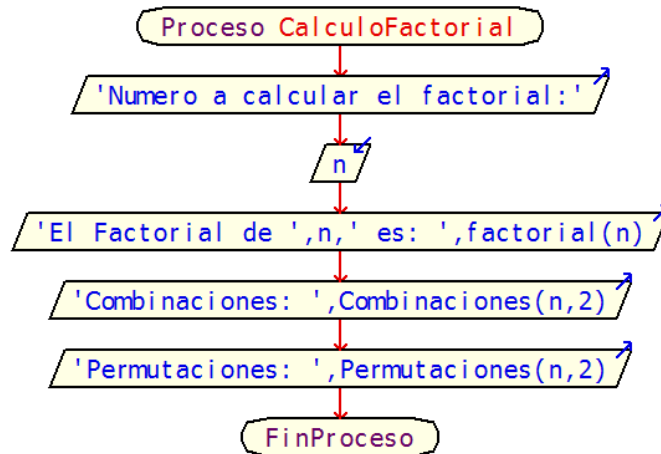
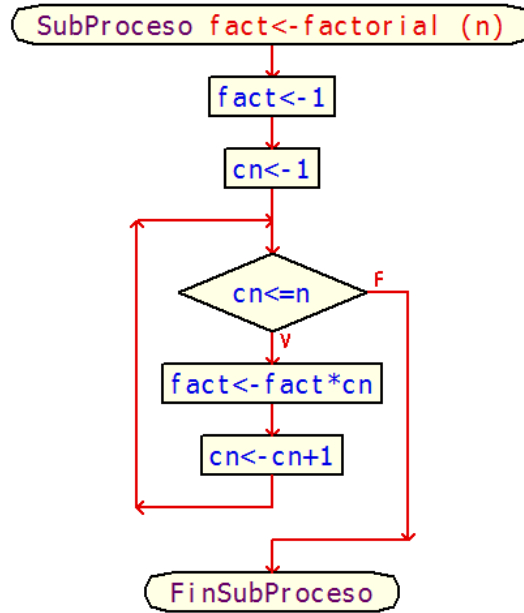
Imprimir "Combinaciones: ",Combinaciones(n,2)

Imprimir "Permutaciones: ",Permutaciones(n,2)

FinProceso

**Diagramas de Flujo de los Subprocesos:**





**Ejecución del Pseudocódigo:**

```

    PSeInt - Ejecutando proceso CALCULOFACTORIAL
    *** Ejecución Iniciada. ***
    Numero a calcular el factorial:
    > 5
    El Factorial de 5 es: 120
    Combinaciones: 10
    Permutaciones: 20
    *** Ejecución Finalizada. ***
  
```

## Arreglos

La instrucción **Dimensión** permite definir un arreglo, indicando sus dimensiones.

```
Dimension <identificador> (<max1>,....,<maxN>);
```

Esta instrucción define un arreglo con el nombre indicado en **<identificador>** y **N** dimensiones. Los N parámetros indican la cantidad de dimensiones y el valor máximo de cada una de ellas. La cantidad de dimensiones puede ser una o más, y la máxima cantidad de elementos debe ser una expresión numérica positiva. Se pueden definir más de un arreglo en una misma instrucción, separándolos con una coma (,).

```
Dimension <ident1> (<max11>,....,<max1N>),...., <identM> (<maxM1>,....,<maxMN>)
```

Es importante notar que es necesario definir un arreglo antes de utilizarlo.

Ejemplo1).- Se desea obtener el promedio de una serie de números almacenados en un arreglo unidimensional o vector y que además calcule la cantidad de esos números que sean mayores al promedio.

### Pseudocódigo:

```
SubProceso imprimeResultados ( prom,cmay )
    Imprimir "El Promedio de los numeros es:",prom
    Imprimir cmay," Numeros son Mayores al Promedio"
Fin SubProceso
```

```
SubProceso cmay <- calculaMayores ( a,n,prom )
    cmay=0
    Para i<-1 Hasta n
        Si a[i]>prom Entonces
            cmay=cmay+1
        Fin Si
    Fin Para
Fin SubProceso
```

```
SubProceso prom <- calculaPromedio ( a,n )
    snum=0
    Para i<-1 Hasta n
        snum=snum+a[i]
    Fin Para
    prom=snum/n
Fin SubProceso
```

```
SubProceso llenarArreglo ( a,n )
```

```

    Para i<-1 Hasta n
        Imprimir "Introduce el ",i," numero:"
        Leer a[i]
    Fin Para
Fin SubProceso

```

```

SubProceso llamaMetodos ()
    Imprimir "Tamaño del Arreglo:"
    Leer n
    Dimension a[n]
    llenarArreglo(a,n)
    prom=calculaPromedio(a,n)
    cmay=calculaMayores(a,n,prom)
    imprimeResultados(prom,cmay)
Fin SubProceso

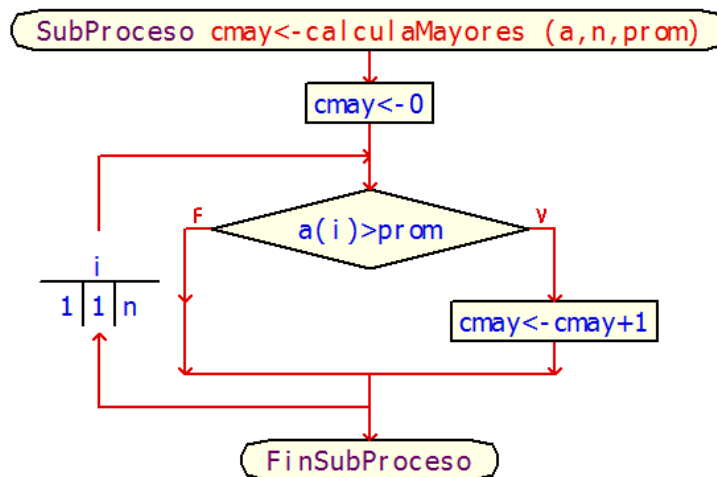
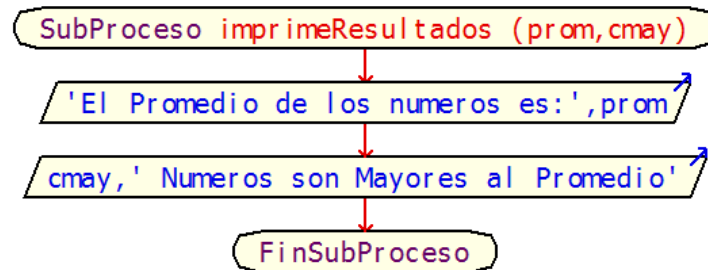
```

```

Proceso Principal
    llamaMetodos()
FinProceso

```

### Diagramas de Flujo de los Subprocesos:



SubProceso `prom<-calculaPromedio (a,n)`

`snum<-0`

`i`  
`1 | 1 | n`

`snum<-snum+a(i)`

`prom<-snum/n`

FinSubProceso

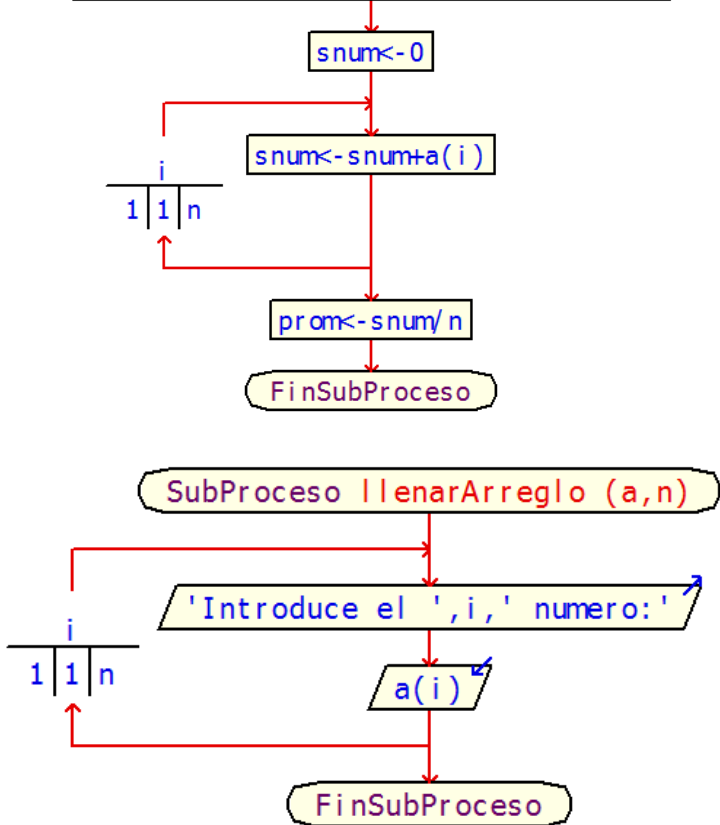
SubProceso `llenarArreglo (a,n)`

`i`  
`1 | 1 | n`

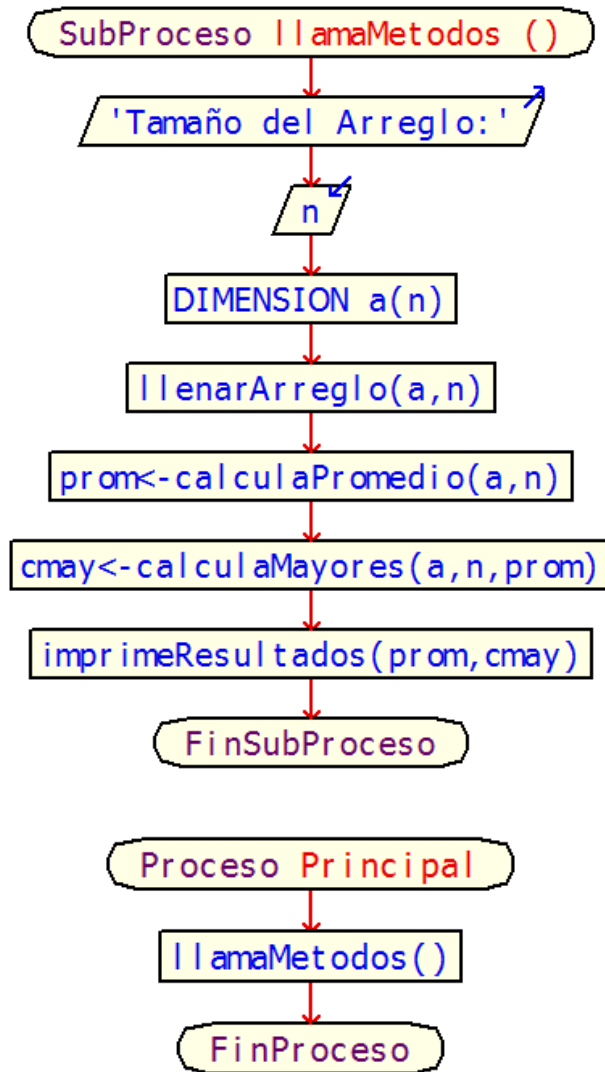
'Introduce el ', `i`, ' numero:'

`a(i)`

FinSubProceso







### Ejecución del Pseudocodigo:

```

PSeInt - Ejecutando proceso PRINCIPAL
*** Ejecución Iniciada. ***
Tamaño del Arreglo:
> 5
Introduce el 1 numero:
> 67
Introduce el 2 numero:
> 23
Introduce el 3 numero:
> 56
Introduce el 4 numero:
> 34
Introduce el 5 numero:
> 12
El Promedio de los numeros es:38.4
2 Numeros son Mayores al Promedio
*** Ejecución Finalizada. ***
  
```

Ejemplo2).- Se tiene una matriz de n número de renglones por x n número de columnas que almacenan números enteros y se desea invertir el contenido de cada una de las filas; es decir el contenido de la 1ra fila, por el contenido de la última fila, el contenido de la segunda fila por el contenido de la fila anterior a la última y así sucesivamente.

**Pseudocodigo:**

SubProceso imprimeMatrizResultante(a,nr,nc)

```
para i=1 hasta nr Hacer
    para j=1 hasta nc Hacer
        Imprimir a[i,j], " " sin saltar
    FinPara
    Imprimir ""
FinPara
```

Fin SubProceso

SubProceso invierteRenglones(a,nr,nc)

```
pos=nr
para i=1 hasta nr/2 Hacer
    para j=i hasta nc Hacer
        aux=a[i,j]
        a[i,j]=a[pos,j]
        a[pos,j]=aux
    FinPara
    pos=pos-1
FinPara
```

Fin SubProceso

SubProceso leerDatosMatriz(a,nr,nc)

    para i=1 hasta nr Hacer

        para j=1 hasta nc Hacer

            Imprimir "Introduce un numero para la ",i," fila y ",j," columna:"

            Leer a[i,j]

        FinPara

    FinPara

Fin SubProceso

SubProceso ProcesoPrincipal()

    Imprimir "No. de Renglones:"

    Leer nr

    Imprimir "No. de Columnas:"

    Leer nc

    Dimension a[nr,nc]

    leerDatosMatriz(a,nr,nc)

    invierteRenglones(a,nr,nc)

    imprimeMatrizResultante(a,nr,nc)

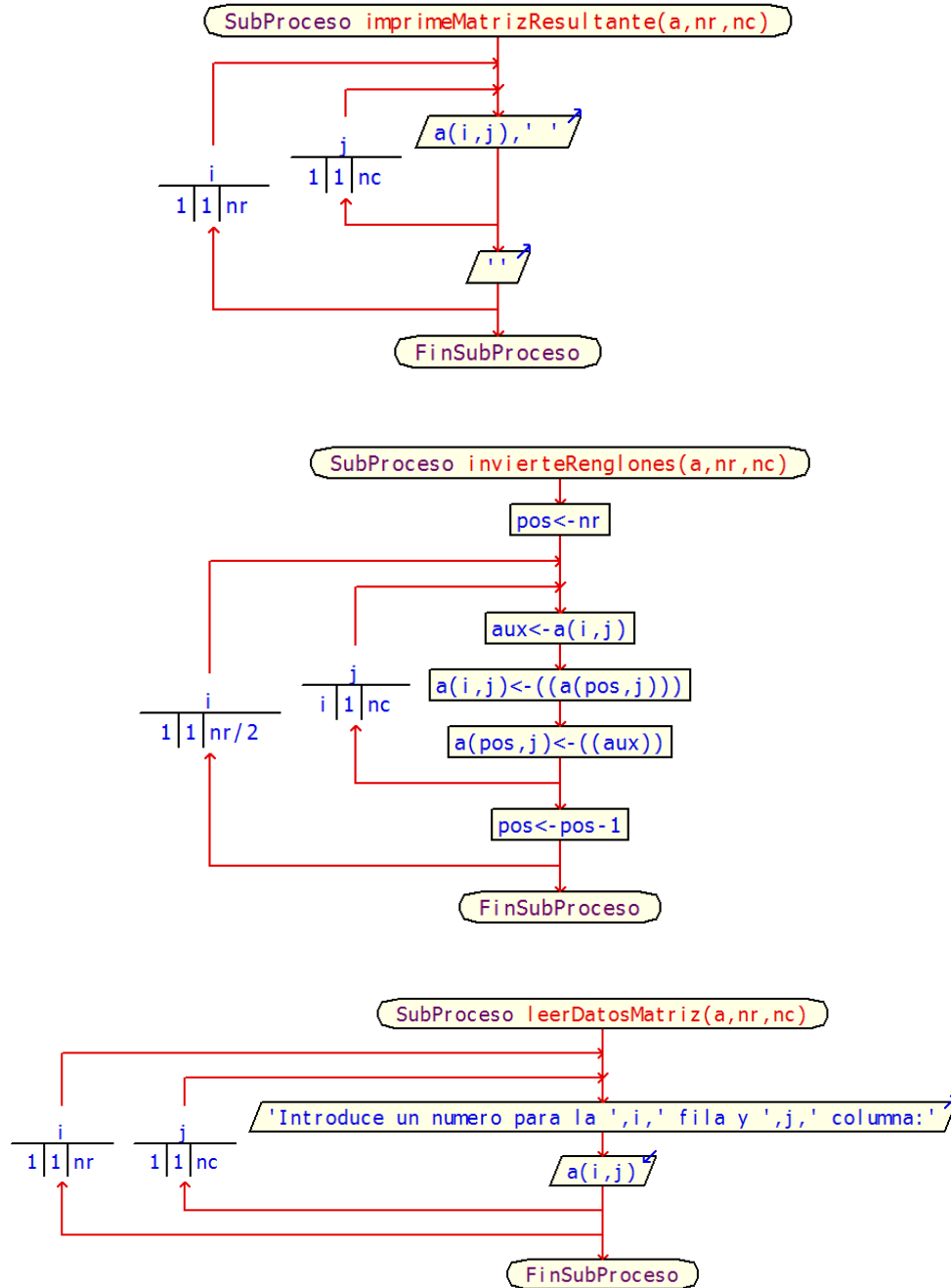
Fin SubProceso

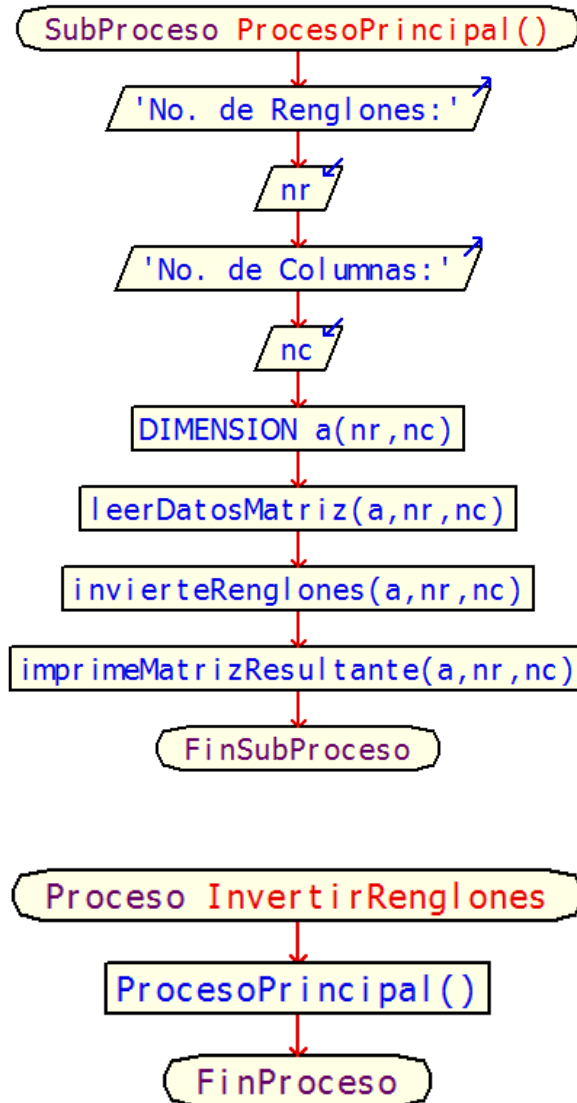
Proceso InvertirRenglones

    ProcesoPrincipal()

FinProceso

## Diagramas de Flujo de los Subprocesos:





### Ejecución del Pseudocodigo:

```

PSeInt - Ejecutando proceso INVERTIRREGLONES
*** Ejecución Iniciada. ***
No. de Renglones:
> 2
No. de Columnas:
> 2
Introduce un numero para la 1 fila y 1 columna:
> 56
Introduce un numero para la 1 fila y 2 columna:
> 12
Introduce un numero para la 2 fila y 1 columna:
> 89
Introduce un numero para la 2 fila y 2 columna:
> 34
89 34
56 12
*** Ejecución Finalizada. ***
  
```

Ejemplo3).- Se tiene almacenado en una matriz las calificaciones de m alumnos que están cursando n materias. Se desea obtener la siguiente información:

- El promedio obtenido por cada uno de los alumnos
- El promedio obtenido por cada una de las materias

**Pseudocodigo:**

SubProceso promedioPorMateria(materias,calificaciones,promat,na,nm)

Imprimir "Promedio de Calificaciones por Materia:"

para j=1 hasta nm Hacer

Imprimir materias[j], " -> " Sin Saltar

scal=0

para i=1 hasta na Hacer

Imprimir calificaciones[i,j], " ," Sin Saltar

scal=scal+calificaciones[i,j]

FinPara

promat[j]=scal/na

Imprimir promat[j]

FinPara

Fin SubProceso

SubProceso promedioPorAlumno(alumnos,calificaciones,proalum,na,nm)

Imprimir "Promedio de Calificaciones por Alumno:"

para i=1 hasta na Hacer

Imprimir alumnos[i], " -> " sin saltar

scal=0

para j=1 hasta nm Hacer

Imprimir calificaciones[i,j], " ," sin saltar

```

        scal=scal+calificaciones[i,j]
    FinPara
    proalum[i]=scal/nm
    Imprimir proalum[i]
FinPara
Fin SubProceso

SubProceso capturaCalificaciones(alumnos,materias,calificaciones,na,nm)
    para i=1 hasta na Hacer
        Imprimir "Introduce las Calificaciones para ",alumnos[i],":"
        para j=1 hasta nm Hacer
            imprimir materias[j],":"
            leer calificaciones[i,j]
        FinPara
        Borrar Pantalla
    FinPara
Fin SubProceso

SubProceso capturaMaterias(materias,nm)
    para i=1 hasta nm Hacer
        Imprimir "Introduce el Nombre de la ",i," Materia:"
        Leer materias[i]
    FinPara
Fin SubProceso

```

SubProceso capturaAlumnos(alumnos,na)

para i=1 hasta na Hacer

Imprimir "Introdue el Nombre del ",i," Alumno:"

Leer alumnos[i]

FinPara

Fin SubProceso

SubProceso principiaPrograma()

Imprimir "No. de Alumnos:"

Leer na

Dimension alumnos[na]

capturaAlumnos(alumnos,na)

Borrar Pantalla

Imprimir "No. de Materias:"

Leer nm

Dimension materias[nm]

capturaMaterias(materias,nm)

Borrar Pantalla

Dimension calificaciones[na,nm]

capturaCalificaciones(alumnos,materias,calificaciones,na,nm)

Dimension proalum[na],promat[nm]

promedioPorAlumno(alumnos,calificaciones,proalum,na,nm)

imprimir "Presiona una tecla para continuar:"

esperar tecla



Borrar pantalla

promedioPorMateria(materias,calificaciones,promat,na,nm)

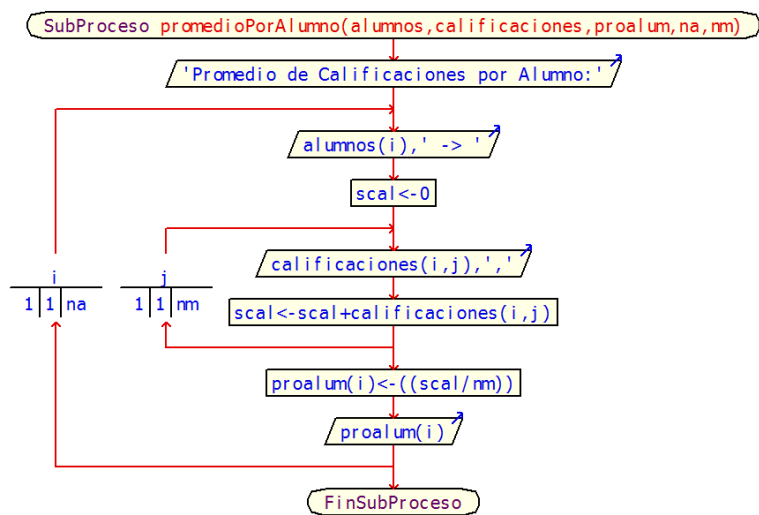
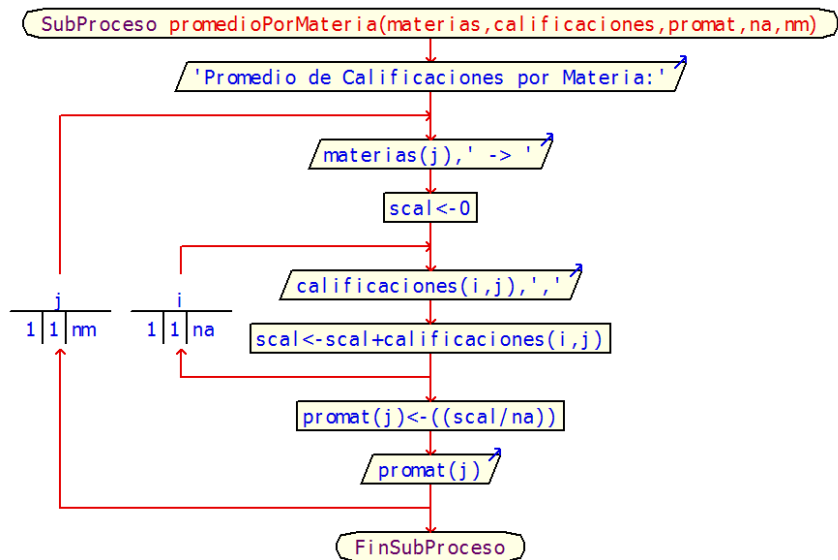
Fin SubProceso

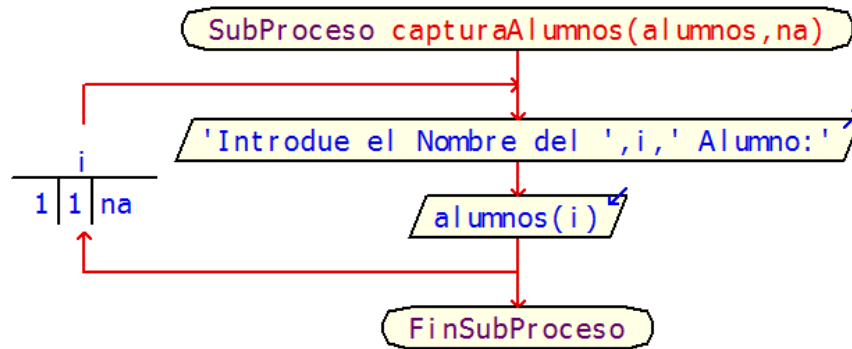
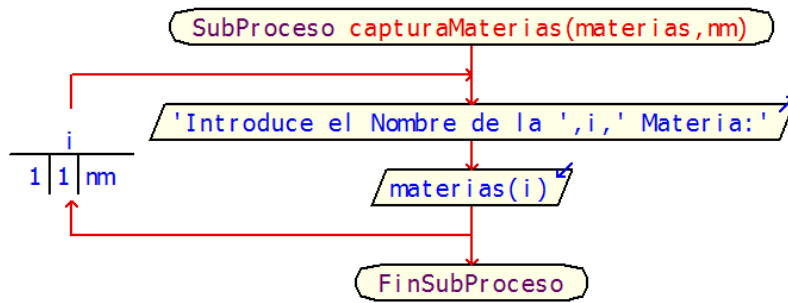
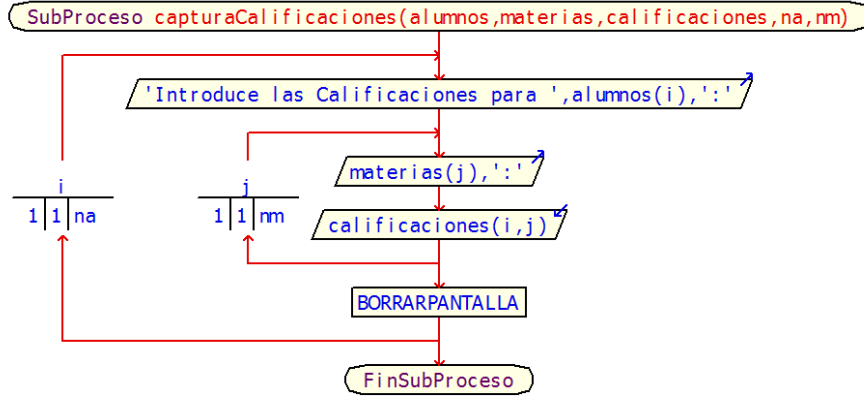
Proceso AlumnosMaterias

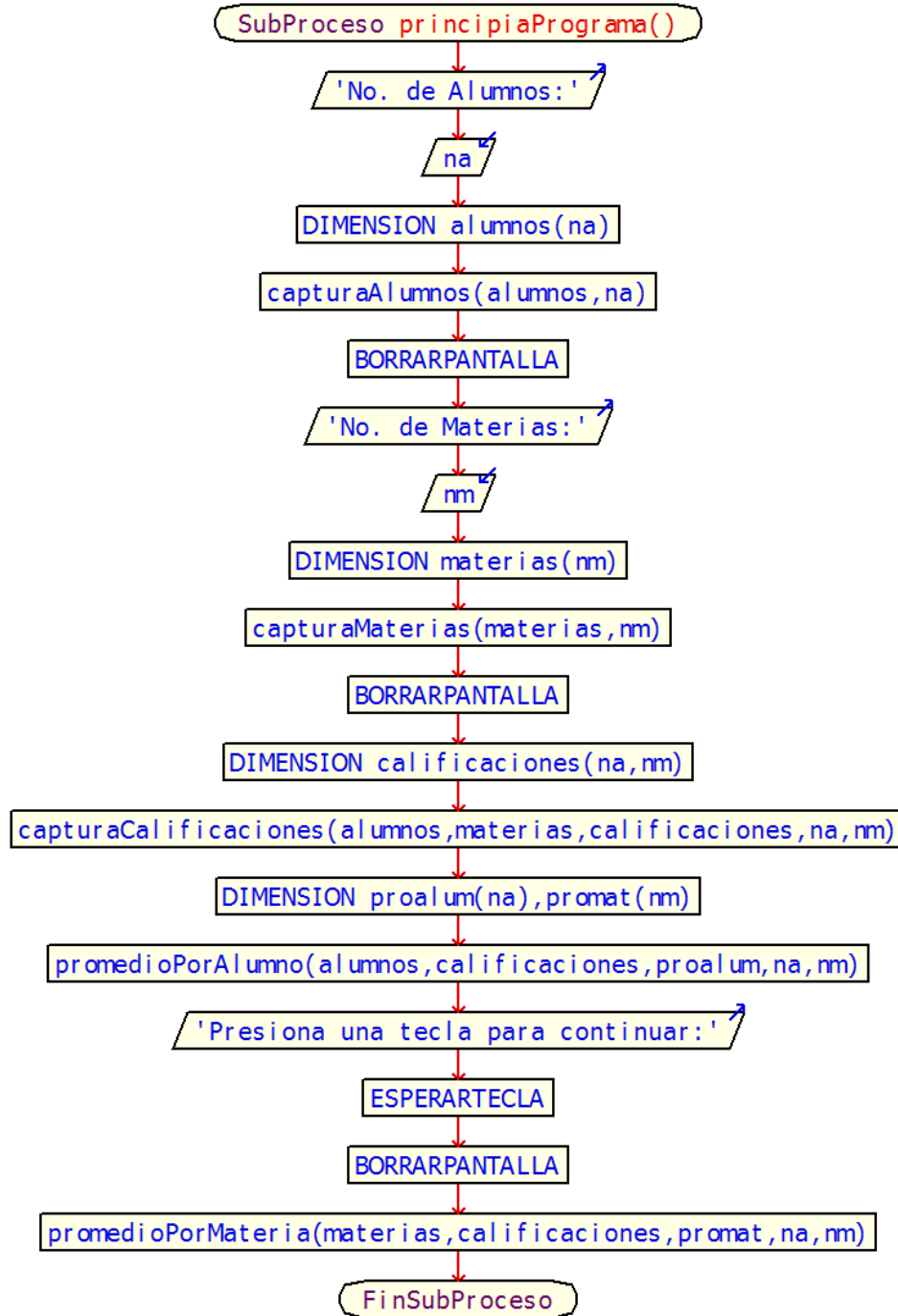
principiaPrograma

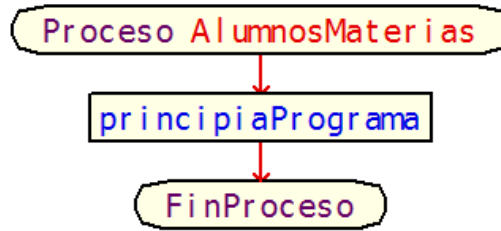
FinProceso

### Diagramas de Flujo de los Subprocesos:









### Ejecucion del Pseudocodigo:

```
PSeInt - Ejecutando proceso ALUMNOSMATERIAS
*** Ejecución Iniciada. ***
No. de Alumnos:
> 3
Introduce el Nombre del 1 Alumno:
> Felipe de Jesus
Introduce el Nombre del 2 Alumno:
> Juan Jose
Introduce el Nombre del 3 Alumno:
> Alberto Carlos
```

```
PSeInt - Ejecutando proceso ALUMNOSMATERIAS
No. de Materias:
> 3
Introduce el Nombre de la 1 Materia:
> Quimica
Introduce el Nombre de la 2 Materia:
> Fisica
Introduce el Nombre de la 3 Materia:
> Matematicas
```

```
PSInt - Ejecutando proceso ALUMNOSMATERIAS
Introduce las Calificaciones para Felipe de Jesus:
Quimica:
> 90
Fisica:
> 78
Matematicas:
> 92
```

```
PSInt - Ejecutando proceso ALUMNOSMATERIAS
Introduce las Calificaciones para Juan Jose:
Quimica:
> 78
Fisica:
> 77
Matematicas:
> 79
```

```
PSInt - Ejecutando proceso ALUMNOSMATERIAS
Introduce las Calificaciones para Alberto Carlos:
Quimica:
> 93
Fisica:
> 98
Matematicas:
> 87
```

```
PSeInt - Ejecutando proceso ALUMNOSMATERIAS
Promedio de Calificaciones por Alumno:
Felipe de Jesus -> 90,78,92,86.6666666667
Juan Jose -> 78,77,79,78
Alberto Carlos -> 93,98,87,92.6666666667
Presiona una tecla para continuar:
```

```
PSeInt - Ejecutando proceso ALUMNOSMATERIAS
Promedio de Calificaciones por Materia:
Quimica -> 90,78,93,87
Fisica -> 78,77,98,84.3333333333
Matematicas -> 92,79,87,86
*** Ejecución Finalizada. ***
```