

## Ejercicio 1

El primer ciclo paralelo C, cuenta con 28 estudiantes, de los cuáles al finalizar el periodo, la Dirección de la carrera de Computación a solicitado las siguientes estadísticas de la materia INTRODUCCIÓN A LA PROGRAMACIÓN en función a los promedios por estudiante, dichos promedios se deben calcular (ponderar) de 3 calificaciones (ACD que representa el 35% de la nota, APE del 35% y la nota del AA con un peso del 30%). En resumen, los requerimientos son los siguientes:

- Registre los nombres de cada estudiante de dicho paralelo.
- Genere aleatoriamente las notas ACD, APE, AA, para cada uno de los 28 estudiantes de 0-10 pts.
- Calcule el promedio de cada uno de los estudiantes del paralelo dada la siguiente ponderación: ACD->35%, APE->35%, y el AA->30%.
- Obtenga el promedio del curso, del paralelo C.
- Liste los nombres de los estudiantes y su nota, que hayan obtenido un promedio por encima del promedio del curso.
- Liste los nombres de los estudiantes y su nota, que hayan obtenido un promedio por debajo del promedio del curso.
- Muestre el estudiante con su calificación, si es el del mayor promedio (el más alto de la clase).
- Muestre el estudiante con su calificación, si es el del menor promedio (el más bajo de la clase).

### Análisis:

Para este ejercicio se debe generar de manera automática las notas entonces usaríamos el comando aleatorio, luego comparar y presentar los datos obtenidos.

### Entrada:

Nombres de los estudiantes.

### Proceso:

- Genera las 3 notas para cada estudiante de manera aleatoria.
- Promedia las 3 para obtener su promedio.
- Suma todos los promedios para obtener el promedio del curso.
- Se compara el promedio de cada estudiante con el promedio del curso para sacar quienes están por encima y quienes por debajo del promedio.
- Se comparan los promedios para saber la mejor y peor nota.

**Salida:**

- Se presenta la nota de los estudiantes, junto a su promedio.
- Se presenta el promedio del curso.
- Se presentan los estudiantes por encima y debajo del promedio.
- Se presenta al estudiante con el mejor y peor promedio, con su nota.

**Pseudocodigo**

## Algoritmo Uno

```
//Declarar

Definir limEst, limNotas Como Entero

Definir mayorNotaNombre, menorNotaNombre como Cadena

Definir mayorNota, menorNota, promedioCurso Como Real

promedioCurso<-0

limEst<-5

limNotas<-3

Dimension nomAband[limEst]

Dimension matNotas[limEst, limNotas]

Dimension arrProm[limEst]

//Generar notas del estudiante

Para i<-1 Hasta limEst Con Paso 1

    Escribir "Ingrese el nombre del estudiante [",i,"]"

    Leer nomAband[i]

    Para j<-1 Hasta limNotas Con Paso 1

        matNotas[i,j]<-Aleatorio(0,9)

    FinPara

FinPara
```

```

//Generar promedios individuales y del curso

Para i<-1 Hasta limEst Con Paso 1

    arrProm[i]<-
(matNotas[i,1]*0.35)+(matNotas[i,2]*0.35)+(matNotas[i,3]*0.3)

    promedioCurso<-promedioCurso+arrProm[i]

FinPara

promedioCurso<-promedioCurso/limEst

//Mostrar Estadísticas Estudiantes

Escribir "Lista de estudiantes y sus notas"

Escribir "| NOMBRES | ACD | APE | AA | PROMEDIO |"

Para i <- 1 Hasta limEst Con Paso 1

    Escribir nomAband[i], " " Sin Saltar

    Para j <- 1 Hasta limNotas Con Paso 1

        Escribir "| ", matNotas[i, j], " " Sin Saltar

    FinPara

    Escribir "| ", arrProm[i] Sin Saltar

    Escribir ""

FinPara

//Mejor y peor nota

mayorNotaNombre <- nomAband[1]

mayorNota <- arrProm[1]

menorNotaNombre <- nomAband[1]

menorNota <- arrProm[1]

Para i <- 1 Hasta limEst Con Paso 1

```

Si arrProm[i] > mayorNota Entonces

    mayorNota <- arrProm[i]

    mayorNotaNombre <- nomAband[i]

FinSi

Si arrProm[i] < menorNota Entonces

    menorNota <- arrProm[i]

    menorNotaNombre <- nomAband[i]

FinSi

FinPara

    //Mostrar Estadísticas Curso

    Escribir "Promedio del paralelo [C]: ", promedioCurso

Escribir "Estudiantes con una nota por encima del promedio: "

Para i <- 1 Hasta limEst Con Paso 1

    Si arrProm[i] >= promedioCurso Entonces

        Escribir nomAband[i], ": ", arrProm[i]

    FinSi

FinPara

Escribir "Estudiantes con una nota por debajo del promedio: "

Para i <- 1 Hasta limEst Con Paso 1

    Si arrProm[i] < promedioCurso Entonces

        Escribir nomAband[i], ": ", arrProm[i]

    FinSi

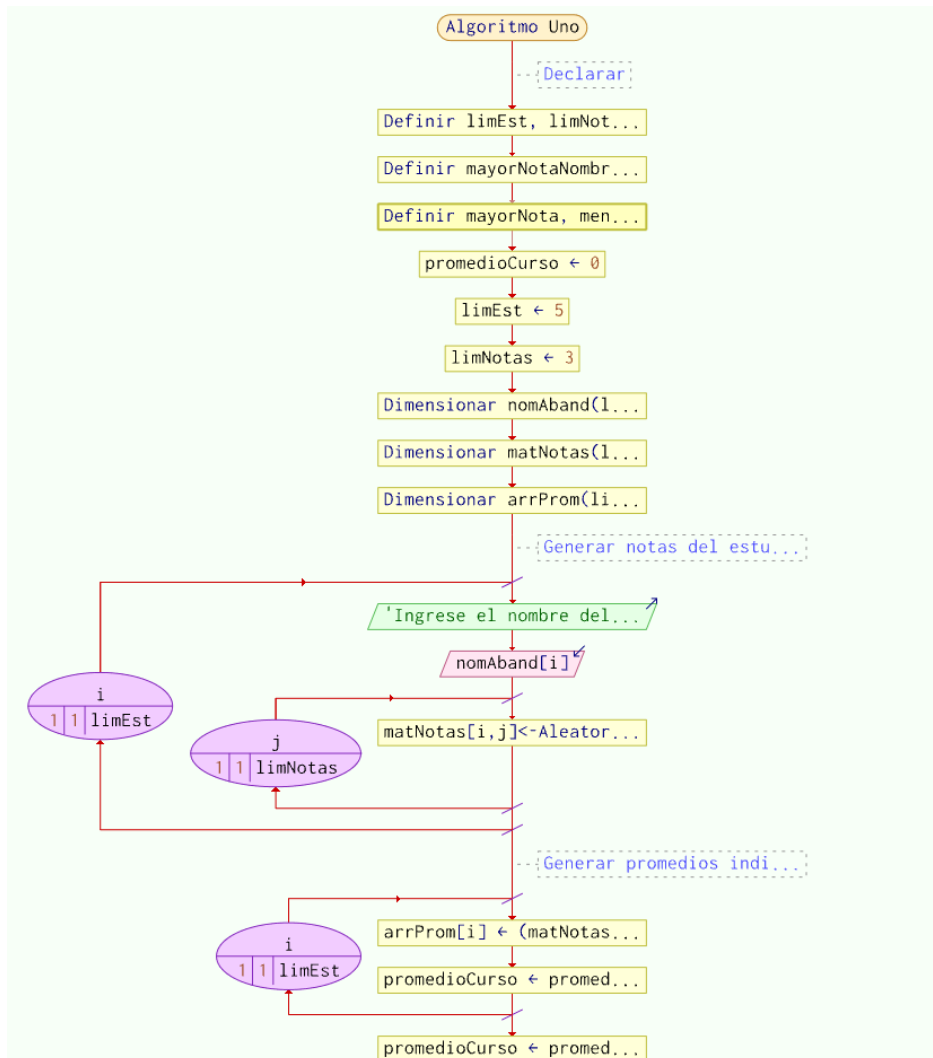
FinPara

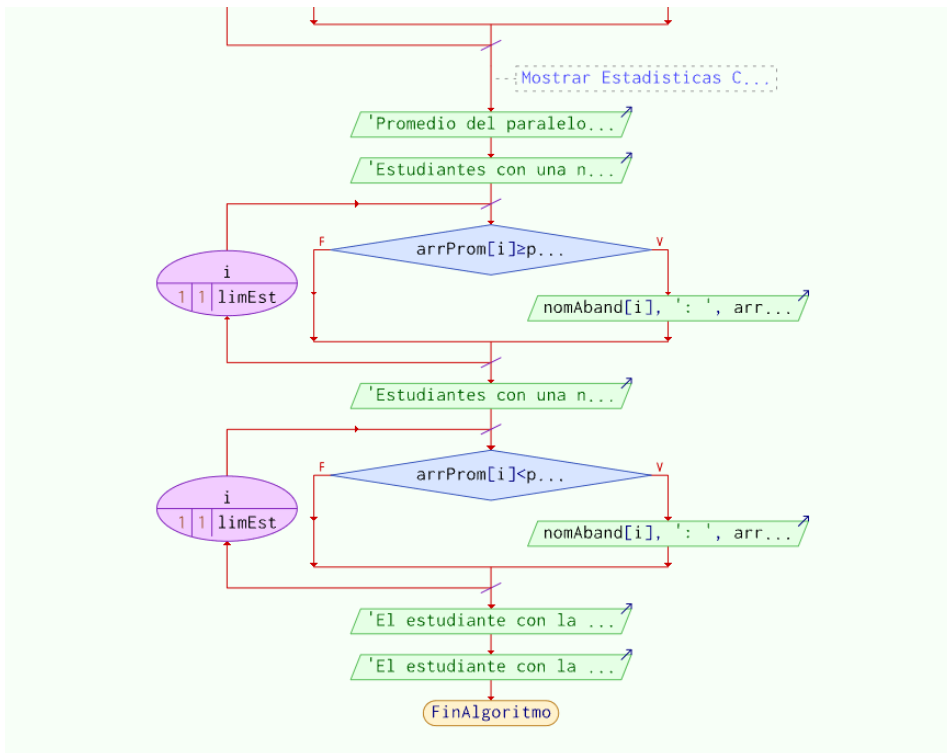
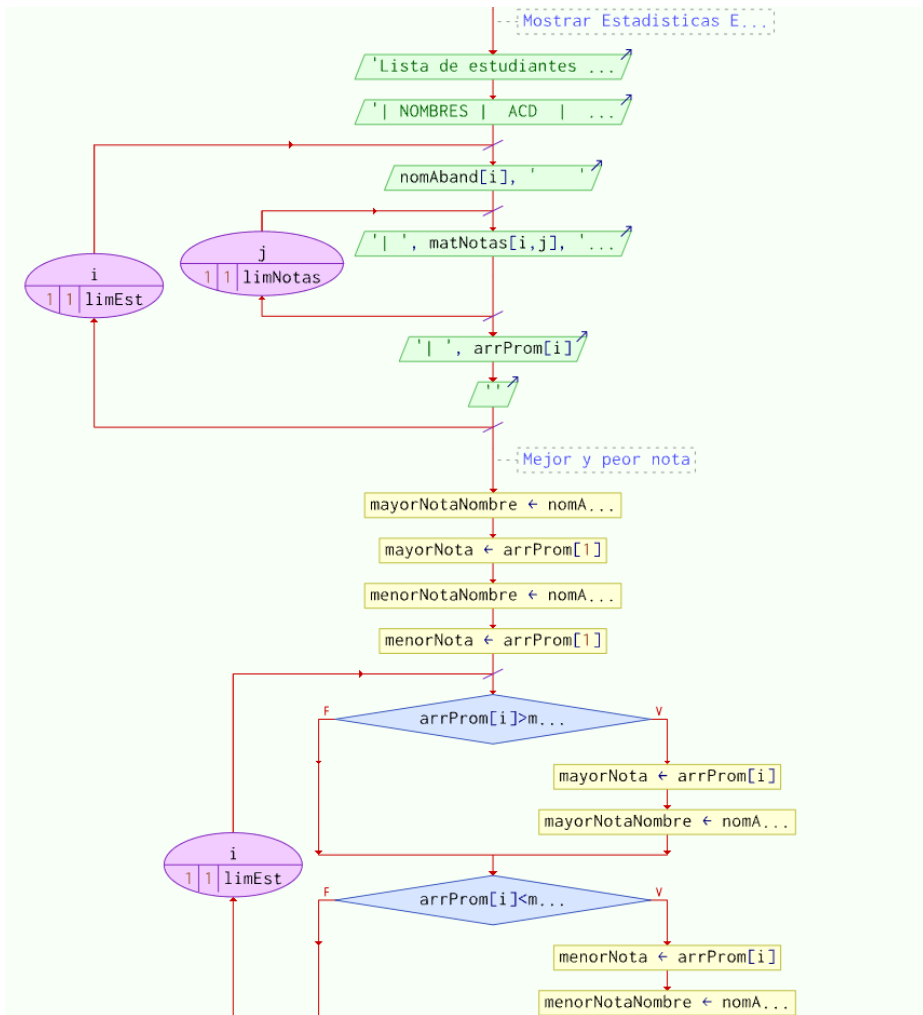
Escribir "El estudiante con la mayor nota es ", mayorNotaNombre, " con una nota de ", mayorNota

Escribir "El estudiante con la menor nota es ", menorNotaNombre, " con una nota de ", menorNota

FinAlgoritmo

## Diagrama de flujo





## Prueba de escritorio

Nombres	ACD	APE	AA	PROMEDIO	Por A/D del promedio	Mejor/Peor
Marcos	4	4	7	4.9	Debajo	
Alicia	9	8	2	6.55	Arriba	
Fernando	8	7	0	5.25	Arriba	
Ana	8	4	9	6.9	Arriba	Mejor
Daniel	4	1	0	1.75	Abajo	Peor
Curso "C"				5.07		

## Ejercicio 2

Crea un programa que gestione el inventario de una tienda. Utiliza una matriz bidimensional para almacenar los productos disponibles en la tienda, con información como nombre, precio y cantidad. El programa debe permitir agregar nuevos productos, actualizar existencias, buscar productos y eliminarlos.

### Análisis:

En este ejercicio se debe agregar opciones para agregar, actualizar y eliminar un producto.

### Entrada:

Nombres productos.

Precio productos.

Cantidad de productos.

### Proceso:

Presentamos las opciones de agregar o buscar un producto.

Al elegir agregar, debemos especificar cuantos queremos agregar.

Pedimos el nombre, precio y cantidad por cada producto.

Luego podemos elegir buscar, encontrara el producto pedido y nos preguntara si queremos actualizar o eliminar.

En este caso actualizar nos permite cambiar las existencias del producto.

Y eliminar borra ese producto de la lista.

**Salida:**

Nos presenta la lista dependiendo la opción que elegimos.

**Pseudocódigo:**

Algoritmo Dos

Definir limProd, limDatos, agregar, inicial como Entero

Definir nomProd, nombreAct, respuesta como Cadena

Definir precCant como Real

Definir mensaje como Logico

limProd <- 1

limDatos <- 2

agregar <- 0

inicial <- 0

mensaje <- Verdadero

Dimension nomProd[28]

Dimension precCant[28, 28]

Repetir

Escribir "Desea agregar productos o buscar producto?  
(Agregar/Buscar/Finalizar)"

Leer respuesta

Si (Mayusculas(respuesta)) = "AGREGAR" Entonces

Escribir "Cuantos productos desea agregar?"

Leer agregar

inicial <- limProd



```

limProd <- limProd + agregar

    Para i <- inicial Hasta limProd-1 Con Paso 1

        Escribir "Ingrese el nombre del producto: "

        Leer nomProd[i]

        Escribir "Ingrese precio y cantidad: "

        Para j <- 1 Hasta limDatos Con Paso 1

            Leer precCant[i, j]

        FinPara

    FinPara

Sino

    Si (Mayusculas(respuesta))="FINALIZAR" Entonces

        mensaje <- Falso

    SiNo

        Si (Mayusculas(respuesta)) = "BUSCAR" Entonces

            Escribir "Ingrese el nombre del producto: "

            Leer nombreAct

            Para i <- 1 Hasta limProd Con Paso 1

                Si (Mayusculas(nombreAct)) =
(Mayusculas(nomProd[i]))Entonces

                    Escribir "Desea actualizar las
existencias o eliminar este producto? (Actualizar/Eliminar)"

                    Leer respuesta

                    Si (Mayusculas(respuesta)) =
"ACTUALIZAR" Entonces

                        Leer precCant[i, 2]

```

```

FinSi
Si (Mayusculas(respuesta)) =
"ELIMINAR" Entonces
    1 Con Paso 1
    nomProd[i + 1]
    limDatos Con Paso 1
    <- precCant[i + 1, j]
    FinPara
    FinPara
    limProd <- limProd - 1
    FinSi
FinSi
FinPara
FinSi
FinSi
FinSi
Si mensaje Entonces
    Para k <- 1 Hasta 1 Con Paso 1
        Escribir "Lista de productos: "
        Escribir "Producto / Precio / Cantidad"
        Para i <- 1 Hasta limProd-1 Con Paso 1
            Escribir nomProd[i] "    "Sin Saltar

```

Para j <- 1 Hasta limDatos Con Paso 1

Escribir "/" , precCant[i, j], " " Sin Saltar

FinPara

Escribir ""

FinPara

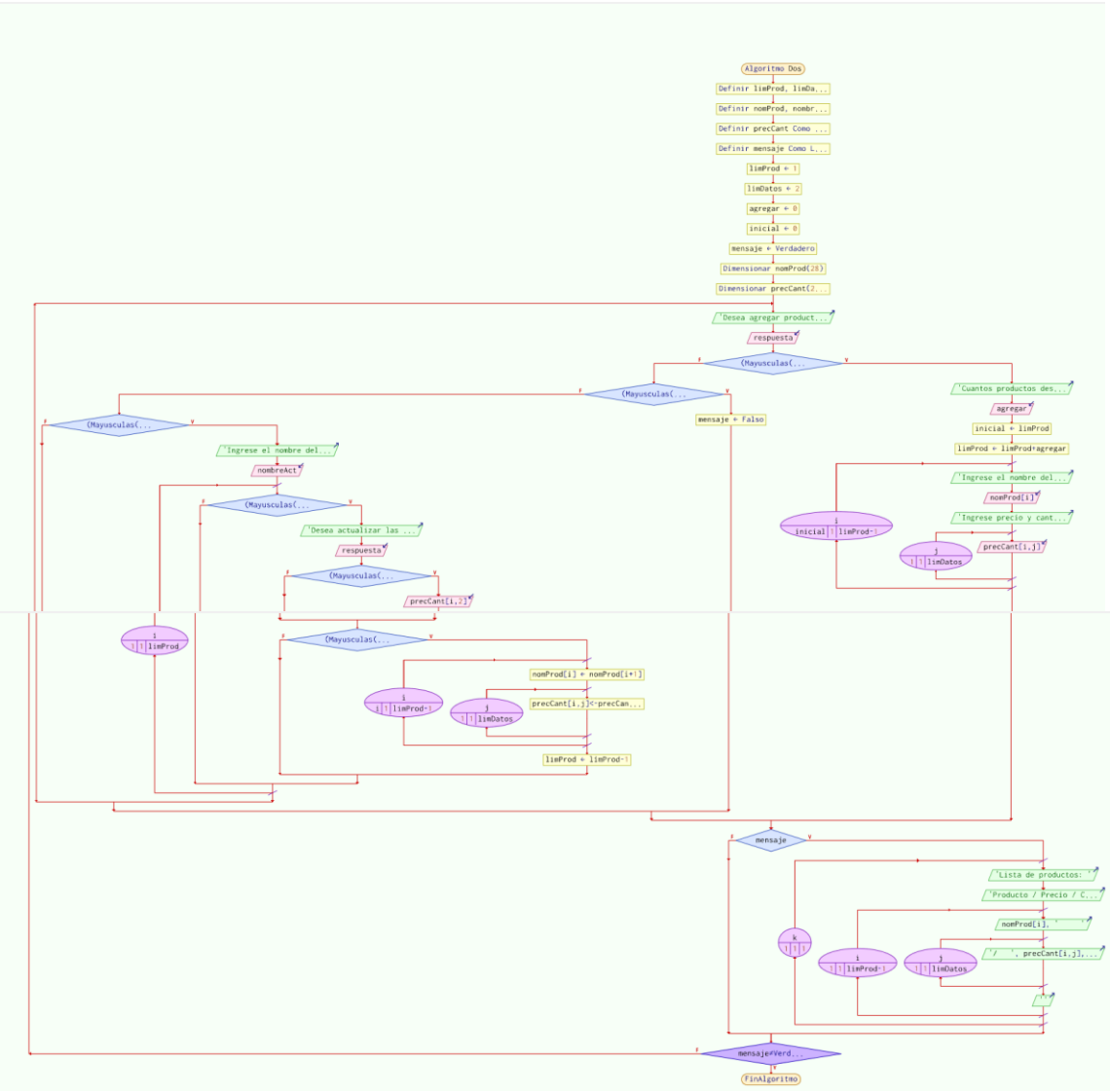
FinPara

FinSi

Hasta Que mensaje<>Verdadero

FinAlgoritmo

Diagrama de flujo:



Prueba de escritorio:

Acción	Cantidad	Productos	Precio	Cantidad	Acción	Acción	Cantidad	Precio
Agregar	2	Azúcar	2	4	Buscar	Actualizar	2	7
		Chocolate	3	2	Buscar	Eliminar	-	-

### Ejercicio 3

Crea un programa que gestione el inventario de una tienda, así como la emisión de facturas. Utiliza una matriz bidimensional para almacenar los productos disponibles en la tienda, con información como nombre, precio y cantidad. El programa debe permitir facturar un producto dado su código, y unidades deseadas. Adicional se debe agregar a la factura al 12% del IVA, y si la compra superar los \$100, se debe aplicar un descuento.

Nota: Considere la alternativa de inexistentes en Stop, para el caso, muestre la alerta respectiva. Crea

#### Análisis

Se utiliza el mismo sistema que en el ejercicio 2 para generar el inventario.

Generar un código aleatorio para cada producto, verificar si el monto total pasa los 100\$ aplicar el descuento y el IVA.

#### Entrada:

Nombre, Precio, Cantidad.

#### Proceso:

Pedir el número de productos que desea ingresar.

Generar un código por cada nombre.

Preguntar si desea ver la lista o hacer una factura.

Pedir código.

Preguntar la cantidad de productos.

#### Salida:

Mostrar la factura de todos los productos agregados.

#### Pseudocódigo:

Algoritmo InventarioFactura03

Definir limProd, limDatos, a, numeroProductos, CantidadProd como Entero

Definir precCant como Real

Definir mensaje como Logico

Definir sumatoria como Real

limProd = 1

limDatos = 2

a = 1

mensaje = Verdadero

sumatoria = 1-1

Dimension codigoProducto[28], nomProd[28], codigoFac[28], CantidadProd[28], precCant[28, 28]

Mostrar "Cuantos productos desea agregar?"

leer limProd

Para i = 1 Hasta limProd Hacer

Mostrar "Ingrese el nombre del producto: "

Leer nomProd[i]

codigoProducto[i] <- aleatorio(10000, 99999)

Escribir codigoProducto[i]

Mostrar "Ingrese precio y cantidad: "

Para j = 1 Hasta limDatos Hacer

Leer precCant[i, j]

Fin Para

Fin Para

respuesta = "Ver"

Repetir

Si (Mayusculas(respuesta) == "VER") Entonces

Para k = 1 Hasta 1 Hacer

Mostrar "Lista de productos: "

Mostrar "Codigo / Producto / Precio / Cantidad "

Para i = 1 Hasta limProd Hacer

```

    Saltar
        Escribir codigoProducto[i] , " / " , nomProd[i] Sin
        Para j = 1 Hasta limDatos Hacer
            Escribir "/" , precCant[i, j] Sin Saltar
        Fin Para
        Escribir " "
    Fin Para
Fin Para

Fin Si

Si mensaje Entonces
    (Ver/Factura/Terminar)"
        Mostrar "Desea ver la lista o generar una factura?"
        leer respuesta

        Si (Mayusculas(respuesta) == "FACTURA") Entonces
            Mostrar "Ingrese el codigo del producto:"
            Leer codigo
            i = 1
            encontrado = Verdadero

            Mientras encontrado
                Si codigo es igual a codigoProducto[i] Entonces
                    encontrado = Falso

                    Si (precCant[i, 2]) > 0 Entonces
                        Mostrar "Ingrese el numero de
                        productos:"
                        leer numeroProductos

```

```

Si numeroProductos <=
    (precCant[i, 2]) Entonces
        precCant[i, 2] =
            (precCant[i, 2])-numeroProductos
        codigoFac[a] <- i
        CantidadProd[a] =
            numeroProductos
        sumatoria = sumatoria +
            (precCant[i, 2] * numeroProductos)
        a = a + 1
    SiNo
        Escribir "Excede la
            cantidad de existencias"
    Fin Si
Sino
    Mostrar "No quedan existencias"
Fin Si
Sino
    i = i + 1
Fin Si
Fin Mientras
Fin Si
Si (Mayusculas(respuesta) == "TERMINAR") Entonces
    mensaje = Falso
Fin Si
Fin Si
Mientras que mensaje
    Mostrar "Factura:"
    Mostrar "Codigo / Producto / Precio / Cantidad / Total"
    k=1

```



Mientras ( $k < a$ )

    Escribir codigoProducto[codigoFac[k]] , " / ", nomProd[codigoFac[k]] , " / "  
    , precCant[codigoFac[k], 1] , " / " , CantidadProd[k] , " / " , (precCant[codigoFac[k], 1] \* CantidadProd[k])

$k = k + 1$

FinMientras

Mostrar "Total: " , sumatoria

Si sumatoria > 100 Entonces

    Mostrar "Descuento 20%: " , (sumatoria \* 0.2)

    sumatoria = sumatoria - (sumatoria \* 0.2)

Sino

    Mostrar "Descuento 20%: - "

Fin Si

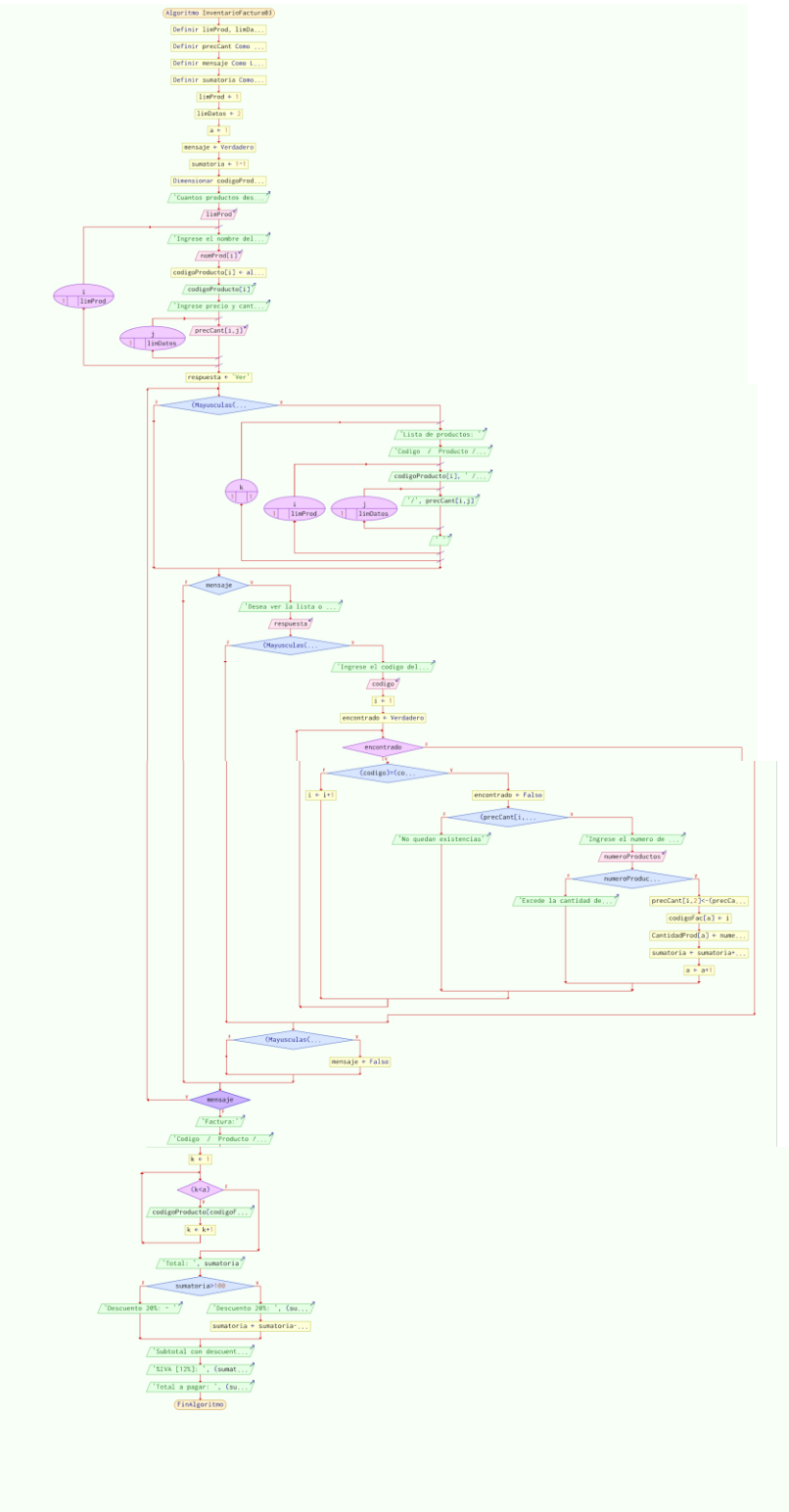
Mostrar "Subtotal con descuento: " , sumatoria

Mostrar "%IVA [12%]: " , (sumatoria \* 0.12)

Mostrar "Total a pagar: " , (sumatoria + (sumatoria \* 0.12))

Fin Algoritmo

# Diagrama de flujo:



## Prueba de escritorio:

Código	Producto	Precio	Cantidad	Cantidad Producto	Total	Descuento%20	IVA%12	Monto Total
12145	Maní	40	5	2	80	-	9.6	89.6
56522	Aceite	50	6	2	180	36	17.28	161.28

## Ejercicio 4

Crea un juego de tres en raya utilizando una matriz bidimensional de 3x3. Permita a dos jugadores marcar sus movimientos alternativamente. El juego debe verificar si alguno de los jugadores ha ganado o si hay un empate.

### Análisis

#### Entrada:

Presenta la matriz 3x3 y con 9 turnos en total para el juego

#### Proceso:

Para i = 1 Hasta 3 Con Paso 1 Hacer

    Para j = 1 Hasta 3 Con Paso 1 Hacer

        tablero[i,j] = " "

    Fin Para

Fin Para

Mientras tiros <= 9 Y No juegoTerminado Hacer

    Para i = 1 Hasta 3 Con Paso 1 Hacer

        Para j = 1 Hasta 3 Con Paso 1 Hacer

            Escribir tablero[i,j] " | " Sin Saltar

        Fin Para

        Escribir " "

    Fin Para

    Escribir "Turno del jugador " , turno , ". Indique la fila (1-3): " Sin

Saltar

leer f

Escribir "Indique la columna (1-3): "

leer c

Si tablero[f,c] = " " Entonces

Si turno = 1 Entonces

tablero[f,c] = "X"

turno = 2

Sino

tablero[f,c] = "O"

turno = 1

Fin Si

tiros = tiros + 1

Para i = 1 Hasta 3 Con Paso 1 Hacer

Si (tablero[i,1] <> " " Y tablero[i,1] = tablero[i,2] Y  
tablero[i,2] = tablero[i,3]) O (tablero[1,i] <> " " Y tablero[1,i] = tablero[2,i] Y  
tablero[2,i] = tablero[3,i]) Entonces

Escribir "El jugador " , turno , " ha ganado"

juegoTerminado = Verdadero

Fin Si

Fin Para

Si (tablero[1,1] <> " " Y tablero[1,1] = tablero[2,2] Y  
tablero[2,2] = tablero[3,3]) O (tablero[1,3] <> " " Y tablero[1,3] = tablero[2,2] Y  
tablero[2,2] = tablero[3,1]) Entonces

Escribir "El jugador " , turno , " ha ganado"

juegoTerminado = Verdadero

```

        Fin Si
    Sino
        Escribir ". Repetir el turno y seleccione una posición que
esté libre."
    Fin Si
    Si tiros = 9 Y No juegoTerminado Entonces
        Escribir "¡Es un empate!"
        juegoTerminado = Verdadero
    Fin Si
Fin Mientras
Para i = 1 Hasta 3 Con Paso 1 Hacer
    Para j = 1 Hasta 3 Con Paso 1 Hacer
        Escribir tablero[i,j] " | " Sin Saltar
    Fin Para
    Escribir " "
Fin Para

```

### **Salida:**

Como resultado va obtener el tablero de juego y cada jugador va tener un turno para darle y si hay un ganador se podrá ver y en caso de empate igual

### **Pseudocodigo**

Inicio

turno = 1

tiros = 1

Dimension tablero[3,3]

juegoTerminado = Falso

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

tablero[i,j] = " "

Fin Para

Fin Para

Mientras tiros <= 9 Y No juegoTerminado Hacer

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir tablero[i,j] " | " Sin Saltar

Fin Para

Escribir " "

Fin Para

Escribir "Turno del jugador " , turno , ". Indique la fila (1-3): " Sin

Saltar

leer f

Escribir "Indique la columna (1-3): "

leer c

Si tablero[f,c] = " " Entonces

Si turno = 1 Entonces

tablero[f,c] = "X"

turno = 2

Sino

tablero[f,c] = "O"

turno = 1

Fin Si

tiros = tiros + 1

Para i = 1 Hasta 3 Con Paso 1 Hacer

Si (tablero[i,1] <> " " Y tablero[i,1] = tablero[i,2] Y  
tablero[i,2] = tablero[i,3]) O (tablero[1,i] <> " " Y tablero[1,i] = tablero[2,i] Y  
tablero[2,i] = tablero[3,i]) Entonces

Escribir "El jugador " , turno , " ha ganado"

juegoTerminado = Verdadero

Fin Si

Fin Para

Si (tablero[1,1] <> " " Y tablero[1,1] = tablero[2,2] Y  
tablero[2,2] = tablero[3,3]) O (tablero[1,3] <> " " Y tablero[1,3] = tablero[2,2] Y  
tablero[2,2] = tablero[3,1]) Entonces

Escribir "El jugador " , turno , " ha ganado"

juegoTerminado = Verdadero

Fin Si

Sino

Escribir ". Repetir el turno y seleccione una posición que  
esté libre."

Fin Si

Si tiros = 9 Y No juegoTerminado Entonces

Escribir "¡Es un empate!"

juegoTerminado = Verdadero

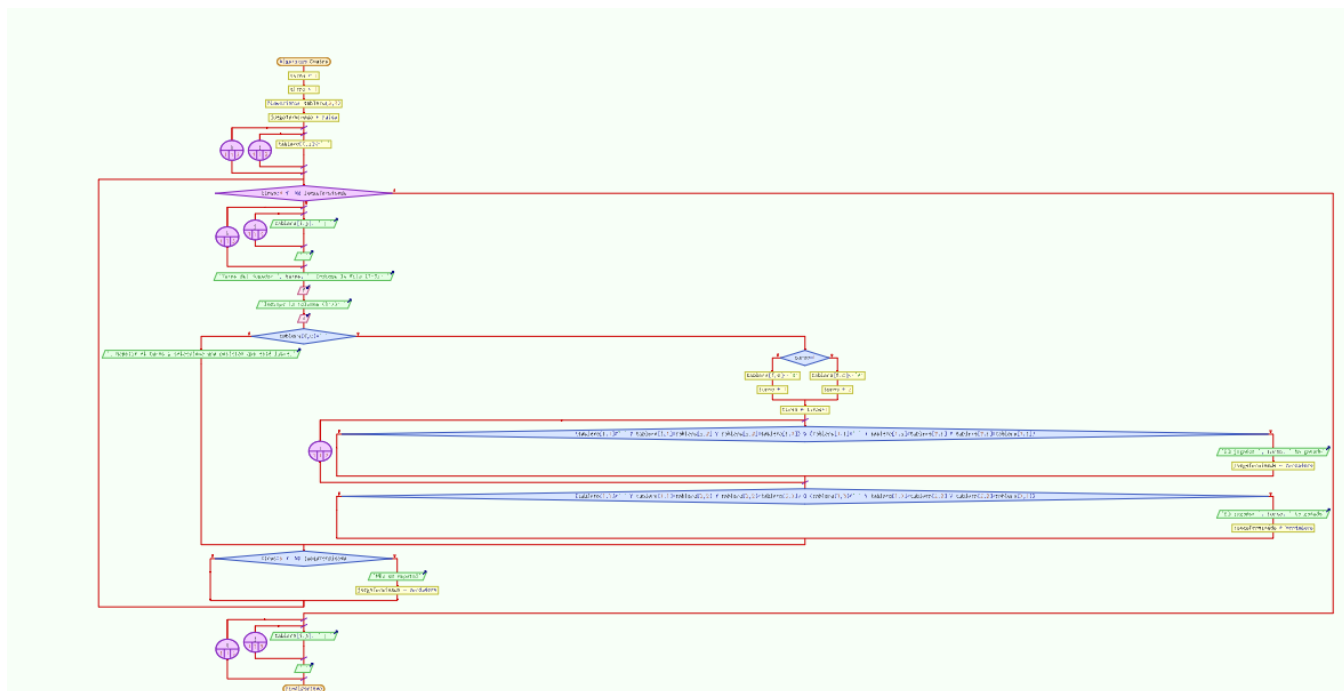
Fin Si

Fin Mientras

Fin Para

Fin

## Diagrama de flujo





## Prueba de escritorio

```
Turno del jugador 2. Indique la fila (1-3): > 2
Indique la columna (1-3):
> 1
X |   |   |
O |   |   |
  |   |   |
Turno del jugador 1. Indique la fila (1-3): > 3
Indique la columna (1-3):
> 1
X |   |   |
O |   |   |
X |   |   |
Turno del jugador 2. Indique la fila (1-3): > 2
Indique la columna (1-3):
> 2
X |   |   |
O | O |   |
X |   |   |
Turno del jugador 1. Indique la fila (1-3): > 3
Indique la columna (1-3):
> 3
X |   |   |
O | O |   |
X |   | X |
Turno del jugador 2. Indique la fila (1-3): > 2
Indique la columna (1-3):
> 3
El jugador 1 ha ganado
X |   |   |
O | O | O |
X |   | X |
*** Ejecución Finalizada. ***
```

## Ejercicio 5

Desarrolla una solución que permita multiplicar dos matrices bidimensionales. El usuario debe ingresar las dimensiones y los elementos de ambas matrices. Realiza la multiplicación y muestra el resultado al final. Nota: Las matrices se deben llenar con número aleatorios únicos enteros positivos o negativos (-9-0, 0-9).

### Análisis

#### Entrada:

Se ingresan dos matrices de manera aleatoria sus valores

#### Proceso:

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz1[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz2[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Escribir "Matriz 1:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz1[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir "Matriz 2:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz2[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir " La multiplicacion es "

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

mmatriz[i, j] = 0

Para k = 1 Hasta 3 Con Paso 1 Hacer

mmatriz[i, j] = mmatriz[i, j] + (matriz1[i, k] \* matriz2[k,  
j])

Fin Para

Escribir mmatriz[i, j], " " Sin Saltar

Fin Para

Escribir ""

Fin Para

### **Salida:**

Mostrara las dos matrices y la multiplicación de ambas

### **Pseudocodigo**

Inicio

Definir i, j Como Entero

Dimension matriz1[3, 3]

Dimension matriz2[3, 3]

Dimension mmatriz[3, 3]

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz1[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz2[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Escribir "Matriz 1:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz1[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir "Matriz 2:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz2[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir " La multiplicacion es "

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

mmatriz[i, j] = 0

Para k = 1 Hasta 3 Con Paso 1 Hacer

mmatriz[i, j] = mmatriz[i, j] + (matriz1[i, k] \* matriz2[k,  
j])

Fin Para

Escribir mmatriz[i, j], " " Sin Saltar

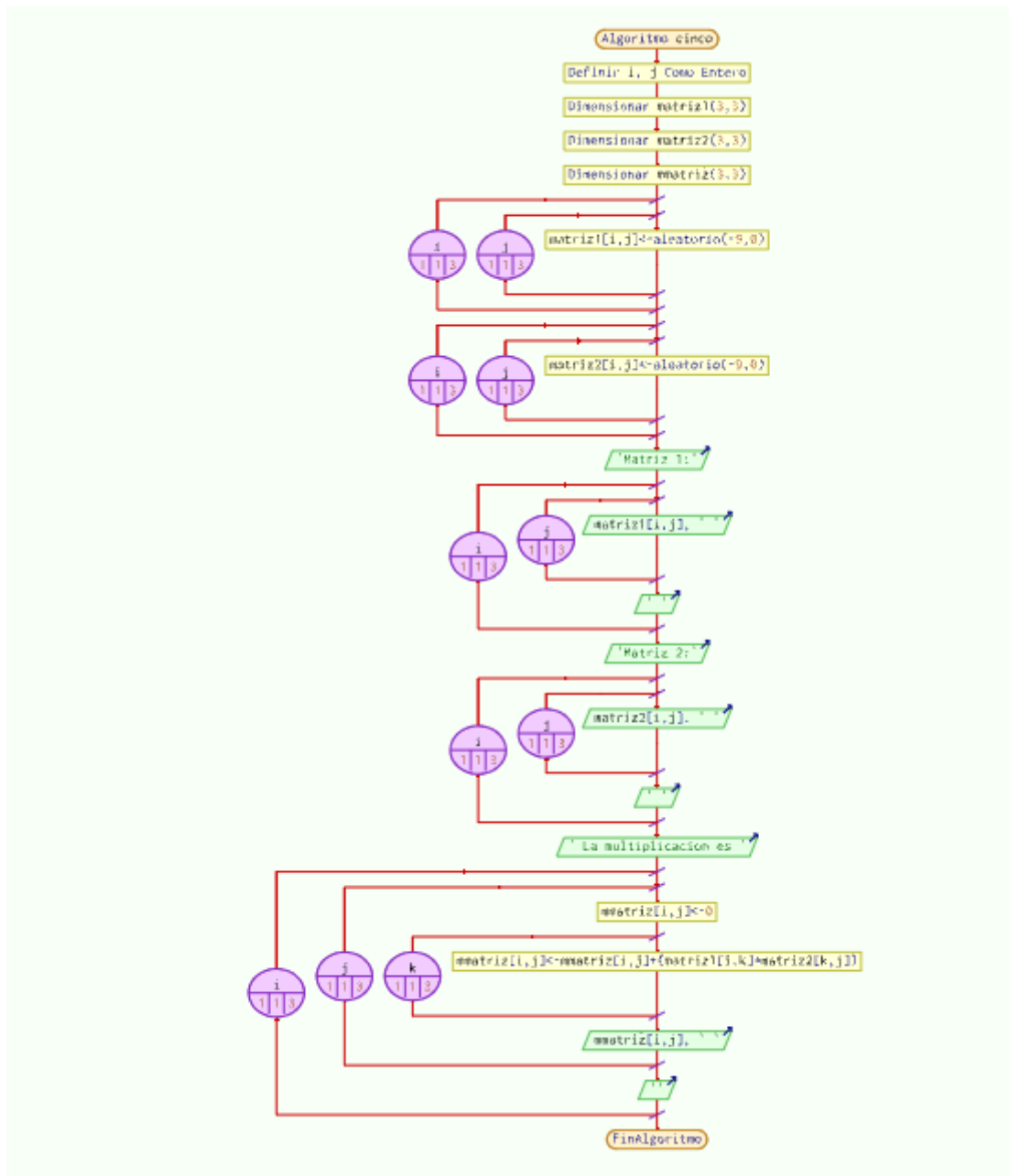
Fin Para

Escribir ""

Fin Para

Fin

## Diagrama de flujo



## Prueba de escritorio

```
Matriz 1:  
-4 -9 -4  
-5 -4 -2  
-5 -7 -7  
Matriz 2:  
-7 -4 -9  
-2 -7 -1  
-9 -2 -9  
La multiplicacion es  
82 87 81  
61 52 67  
112 83 115  
*** Ejecución Finalizada. ***
```

## Ejercicio 6

Desarrolla una solución en pseudocódigo que permita mediante un menú (al estilo calculadora matemática), facilitar las operaciones de suma, resta de dos matrices. Considerar la validación de las reglas para realizar estas operaciones. Nota: Las matrices se deben llenar con número aleatorios únicos enteros positivos o negativos (-9-0, 0-9).

### Análisis

#### Entrada:

Las dos matrices se generaran con valores aleatorios

#### Proceso:

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz1[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz2[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Escribir "Matriz 1:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz1[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir "Matriz 2:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz2[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir "OPERACIONES"

Escribir "1 - SUMA "

Escribir "2 - RESTA "

leer opcion

segun opcion hacer

Caso 1:



Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

$mmatriz[i, j] = matriz1[i, j] + matriz2[i, j]$

Fin Para

Fin Para

Escribir "Resultado de la suma es: "

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir  $mmatriz[i, j]$ , " " Sin Saltar

Fin Para

Escribir " "

Fin Para

Caso 2:

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

$mmatriz[i, j] = matriz1[i, j] - matriz2[i, j]$

Fin Para

Fin Para

Escribir "Resultado de la resta es: "

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir  $mmatriz[i, j]$ , " "

Fin Para

Escribir" "

Fin Para

Fin Segun

### **Salida:**

El usuario escoger la opción para que se realice la operación

### **Pseudocodigo**

Inicio

Definir i, j Como Entero

Dimension matriz1[3, 3]

Dimension matriz2[3, 3]

Dimension mmatriz[3, 3]

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz1[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

matriz2[i, j] = aleatorio(-9, 0)

Fin Para

Fin Para

Escribir "Matriz 1:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz1[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir "Matriz 2:"

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir matriz2[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Escribir "OPERACIONES"

Escribir "1 - SUMA "

Escribir "2 - RESTA "

leer opcion

segun opcion hacer

Caso 1:

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

mmatriz[i, j] = matriz1[i, j] + matriz2[i, j]

Fin Para

Fin Para

Escribir "Resultado de la suma es: "

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir mmatriz[i, j], " " Sin Saltar

Fin Para

Escribir" "

Fin Para

Caso 2:

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

mmatriz[i, j] = matriz1[i, j] - matriz2[i, j]

Fin Para

Fin Para

Escribir "Resultado de la resta es: "

Para i = 1 Hasta 3 Con Paso 1 Hacer

Para j = 1 Hasta 3 Con Paso 1 Hacer

Escribir mmatriz[i, j], " " Sin saltar

Fin Para

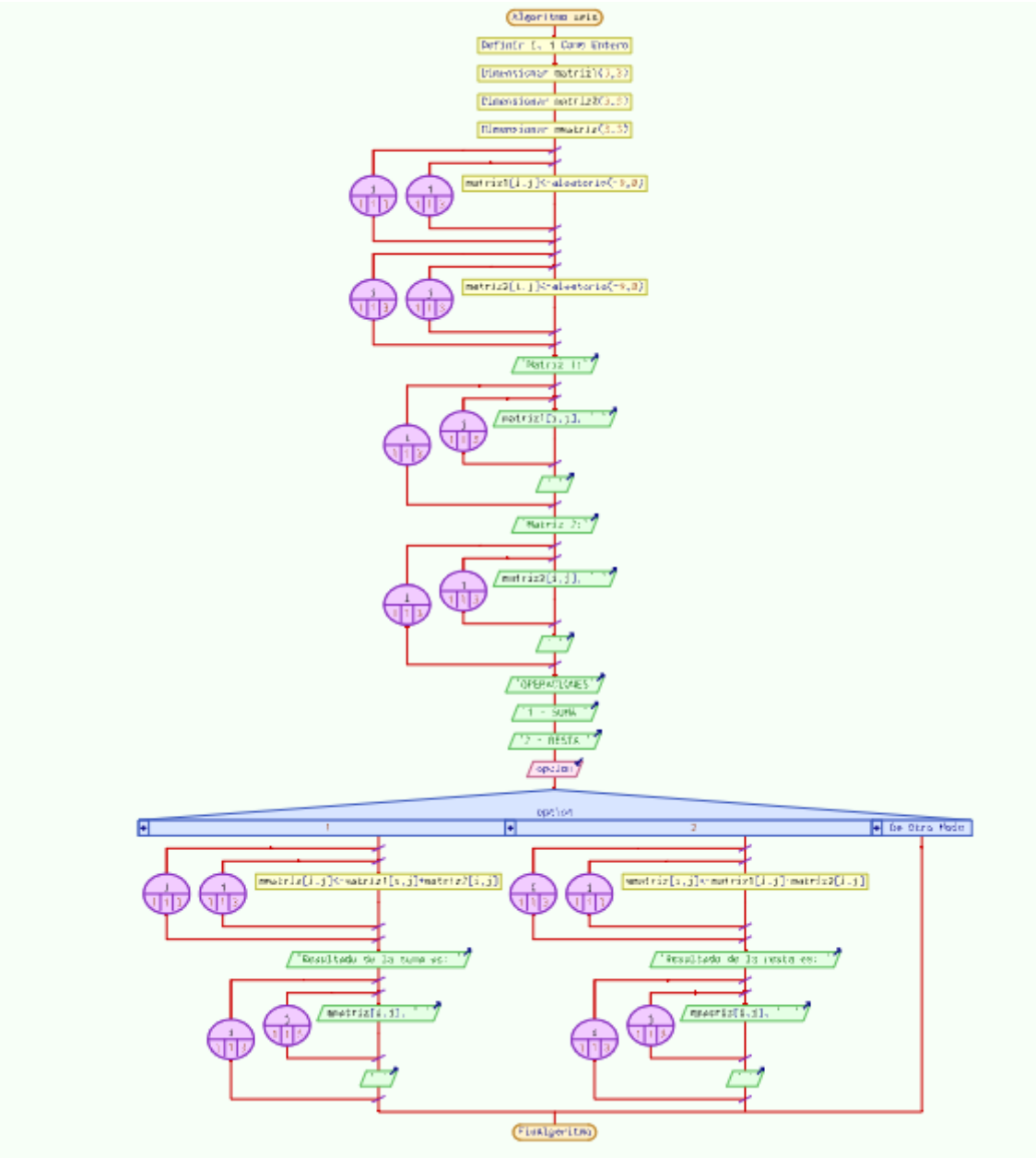
Escribir" "

Fin Para

Fin Segun

Fin

Diagrama de flujo



## Prueba de escritorio

Matriz 1:

-2 -6 0

-4 -2 0

-9 -9 -7

Matriz 2:

-3 -1 -6

-7 -2 -7

-8 -6 -5

OPERACIONES

1 - SUMA

2 - RESTA

> 1

Resultado de la suma es:

-5 -7 -6

-11 -4 -7

-17 -15 -12

Matriz 1:

-9 -4 -4

-4 -8 -2

-7 -2 -5

Matriz 2:

-4 -9 -6

-2 -7 -9

-9 -9 -4

OPERACIONES

1 - SUMA

2 - RESTA

> 2

Resultado de la resta es:

-5 5 2

-2 -1 7

2 7 -1

## Ejercicio 7

Dada una matriz cuadrada y con número aleatorios únicos enteros positivos o negativos (-9-0, 0-9) generados automáticamente, presente los elementos de la diagonal secundaria, los ubicados sobre la diagonal secundaria, y bajo dicha diagonal.

### Análisis

### Entrada:

Se presentara la matriz con los valores aleatorios

### Proceso:

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

matriz[i, j] = aleatorio(-9,0)

Fin Para

Fin Para

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

    Escribir matriz(i , j) , " " Sin Saltar

Fin Para

    Escribir " "

Fin Para

Escribir("Valores de la segunda diagonal:")

Para i = 1 Hasta 5 Con Paso 1 Hacer

    Para j = 1 Hasta 5 Con Paso 1 Hacer

        Si  $i + j = 6$  Entonces

            Escribir matriz(i, j), " " Sin Saltar

        Sino

            Escribir " " Sin Saltar

        Fin Si

    Fin Para

        Escribir ""

Fin Para

Escribir("Valores sobre la segunda diagonal:")

Para i = 1 Hasta 5 Con Paso 1 Hacer

    Para j = 1 Hasta 5 Con Paso 1 Hacer

        Si  $i + j < 6$  Entonces

            Escribir matriz(i, j), " " Sin Saltar

        Sino

            Escribir " " Sin Saltar

```

        Fin Si
    Fin Para
    Escribir" "
Fin Para
Escribir("Valores bajo dicha diagonal: ")
Para i = 1 Hasta 5 Con Paso 1 Hacer
    Para j = 1 Hasta 5 Con Paso 1 Hacer
        Si  $i + j > 6$  Entonces
            Escribir matriz(i, j), " " Sin Saltar
        Sino
            Escribir " " Sin Saltar
        Fin Si
    Fin Para
    Escribir" "
Fin Para

```

### **Salida:**

Se presentara la matriz , seguido con los valores de la diagonal secundaria , los que están sobre la diagonal secundaria y bajo

### **Pseudocodigo**

```

Inicio
    Definir i, j Como Entero
    Dimension matriz[5,5]
    Para i = 1 Hasta 5 Con Paso 1 Hacer

```



Para j = 1 Hasta 5 Con Paso 1 Hacer

matriz[i, j] = aleatorio(-9,0)

Fin Para

Fin Para

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

Escribir matriz(i, j), " " Sin Saltar

Fin Para

Escribir " "

Fin Para

Escribir("Valores de la segunda diagonal:")

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

Si  $i + j = 6$  Entonces

Escribir matriz(i, j), " " Sin Saltar

Sino

Escribir " " Sin Saltar

Fin Si

Fin Para

Escribir ""

Fin Para

Escribir("Valores sobre la segunda diagonal:")

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

Si  $i + j < 6$  Entonces

    Escribir matriz(i, j), " " Sin Saltar

    Sino

        Escribir " " Sin Saltar

    Fin Si

Fin Para

Escribir" "

Fin Para

Escribir("Valores bajo dicha diagonal:.")

Para i = 1 Hasta 5 Con Paso 1 Hacer

    Para j = 1 Hasta 5 Con Paso 1 Hacer

        Si  $i + j > 6$  Entonces

            Escribir matriz(i, j), " " Sin Saltar

        Sino

            Escribir " " Sin Saltar

        Fin Si

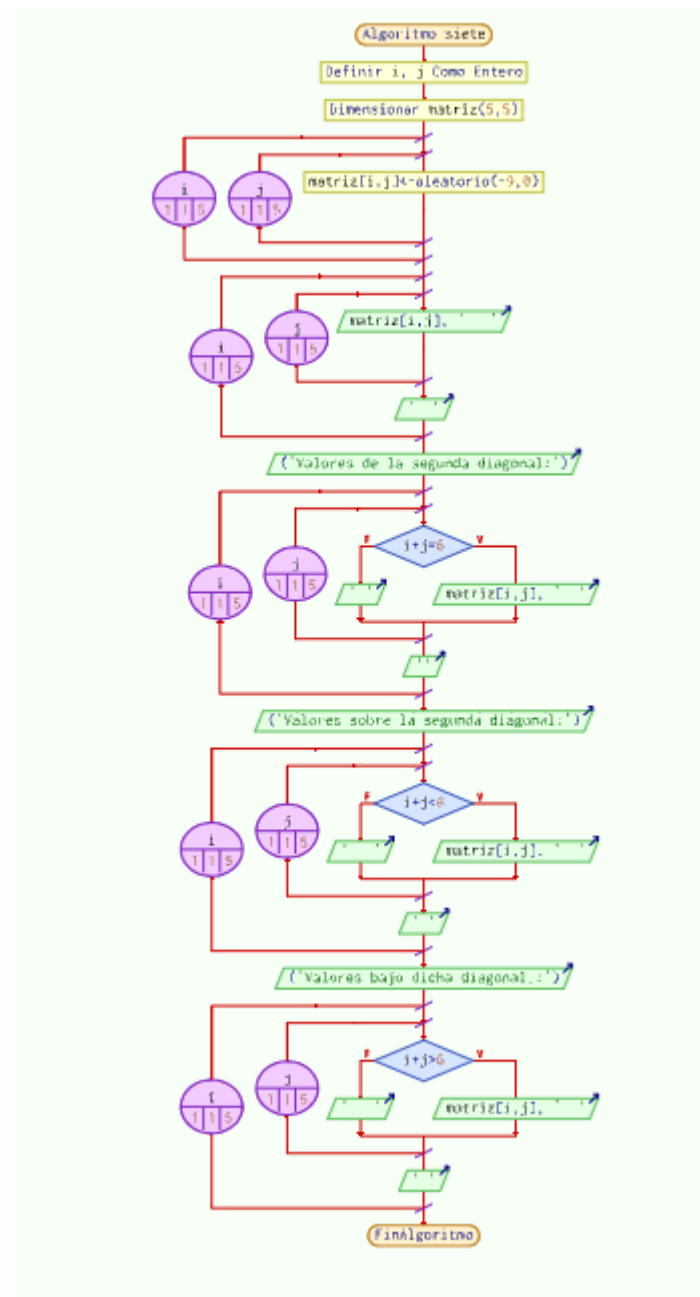
    Fin Para

Escribir" "

Fin Para

Fin

## Diagrama de flujo



## Prueba de escritorio

```
-9  -2  -1  -1  -6
-8  -3  -5  -6  -5
-3  -6  -2  -6  -6
-4  -5  -4  -8  -8
-9  -7  -8  -2  0
```

Valores de la segunda diagonal:

```
-6
-6
-2
-5
-9
```

Valores sobre la segunda diagonal:

```
-9 -2 -1 -1
-8 -3 -5
-3 -6
-4
```

Valores bajo dicha diagonal.:

```
-5
-6 -6
-4 -8 -8
-7 -8 -2 0
```

## Ejercicio 8

Dada una matriz cuadrada y con número aleatorios únicos enteros positivos o negativos (-9-0, 0-9) generados automáticamente, presente los elementos de la diagonal principal, los ubicados sobre la diagonal principal, y bajo dicha principal.

### Análisis:

Debemos generar la diagonal principal los cuales son los valores con i y j iguales.

Para generar los valores arriba de la diagonal debemos encontrar los valores en los que i sea menor a j.

Para generar los valores abajo de la diagonal debemos encontrar los valores en los que i sea mayor a j.

### Entrada:

Limite.

**Proceso:**

Generar los números.

Encontrar los números en los que i y j coincidan para determinar la diagonal principal.

Encontrar los valores arriba de la diagonal encontrando los valores donde i sea menor a j.

Encontrar los valores abajo de la diagonal encontrando los valores donde i sea mayor a j.

**Salida:**

Diagonal.

Números por arriba de la diagonal.

Números por debajo de la diagonal.

**Pseudocódigo**

Inicio

Definir i, j Como Entero

Dimension matriz[5,5]

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

matriz[i, j] = aleatorio(-9,0)

Fin Para

Fin Para

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

Escribir matriz(i, j), " " Sin Saltar

Fin Para

Escribir " "

Fin Para

Escribir("Valores de la diagonal:")

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

Si  $i = j$  Entonces

Escribir matriz(i, j), " " Sin Saltar

Sino

Escribir " " Sin Saltar

Fin Si

Fin Para

Escribir" "

Fin Para

Escribir("Valores sobre la diagonal:")

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

Si  $i < j$  Entonces

Escribir matriz(i, j), " " Sin Saltar

Sino

Escribir " " Sin Saltar

Fin Si

Fin Para

Escribir " "

Fin Para

Escribir("Valores sobrebajo dicha principal.:")

Para i = 1 Hasta 5 Con Paso 1 Hacer

Para j = 1 Hasta 5 Con Paso 1 Hacer

Si  $i > j$  Entonces

Escribir matriz(i, j), " " Sin Saltar

SiNo

Escribir " " Sin Saltar

Fin Si

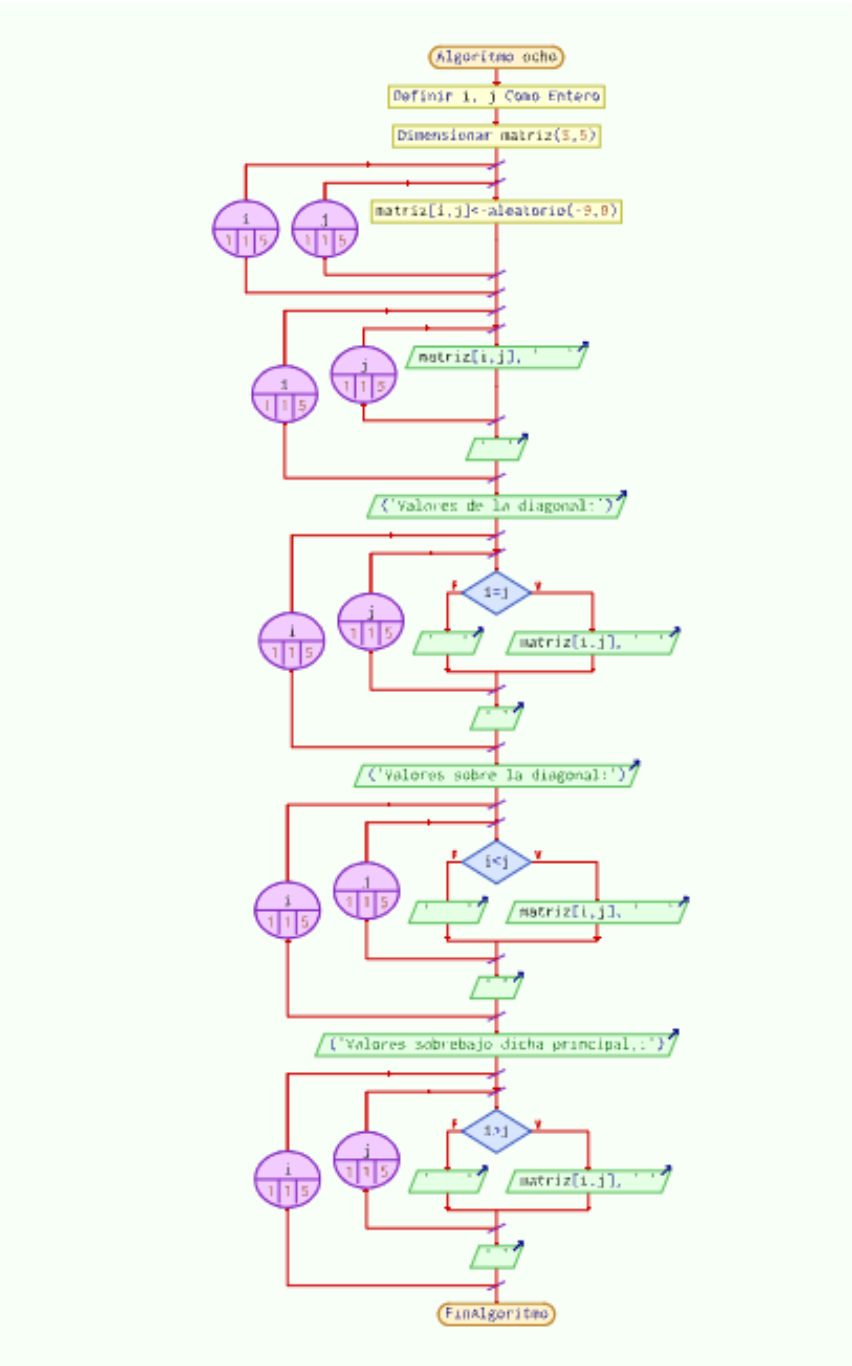
Fin Para

Escribir " "

Fin Para

Fin

Diagrama de flujo



Prueba de escritorio



-5	-8	-4	-1	-4
-4	-9	-2	-3	-1
-8	-1	-9	-8	-5
-7	-7	0	-3	-1
-3	-7	-8	-3	-4

-5				
	-9			
		-9		
			-3	
				-4

	-8	-4	-1	-4
		-2	-3	-1
			-8	-5
				-1

-4				
-8	-1			
-7	-7	0		
-3	-7	-8	-3	

## Ejercicio 9

Con el objetivo de manipular correctamente los ciclos repetitivos anidados, se solicita, dado la altura de la salida, generar por pantalla las siguientes figuras.

Nota: Ud. puede elegir libremente el caracter a mostrar en la figura.

```
*  
* *  
* * *  
* * * *
```

Ejemplo dado un límite de 4.

### Análisis

Según el límite generar la figura.

### Entrada:

Límite.

### Proceso:

Pedir el límite.

Según el límite ir generando asteriscos hasta generar la figura, usando "Para".

### Salida:

Presentar la figura.

### Pseudocódigo

Inicio

Definir i , n, j Como Entero;

Leer n

Para i <- 1 Hasta n Con Paso 1 Hacer

Para j <- 1 Hasta i Con Paso 1 Hacer

Escribir " \* " Sin Saltar

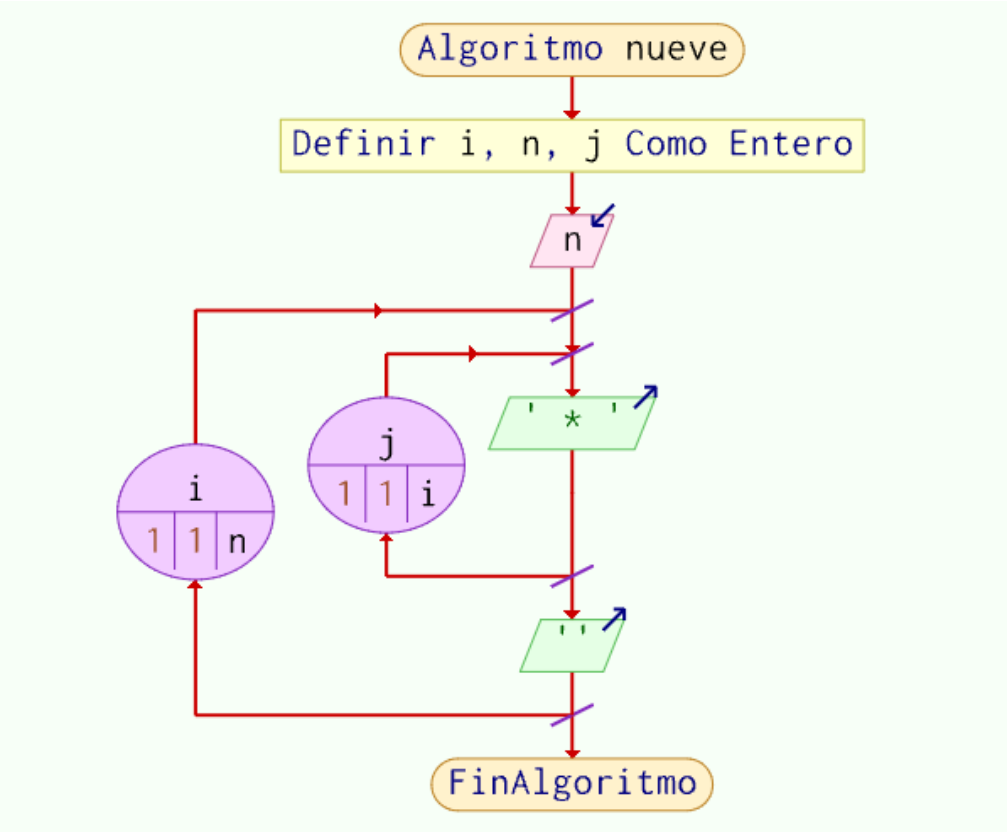
Fin Para

Escribir ""

Fin Para

Fin

Diagrama de flujo



Prueba de escritorio

Limite	3	
*		
*	*	
*	*	*

## Ejercicio 10

Con el objetivo de manipular correctamente los ciclos repetitivos anidados, se solicita, dado la altura de la salida, generar por pantalla las siguientes figuras. Nota: Ud. puede elegir libremente el caracter a mostrar. Nota: Analice cual debería ser el límite correcto (mínimo y máximo) de la doble img para que sea una doble pirámide legible, validar ese dato de entrada.

```
  *
 * *
* * *
* * * *
 * * *
  * *
   *
```

Ejemplo dado un límite de 7.

### Análisis:

Para este problema necesitamos generar un rombo formado por asteriscos dependiendo del limite que se ponga.

### Entrada:

Limite

### Proceso:

Se pide el límite.

Generar usando "Para", para generar espacios y asteriscos formando la figura.

### Salida:

Se presenta la figura.

### Pseudocodigo

Inicio

Definir i , j, k , n Como Entero;

Leer n

Para i = n Hasta 1 Con Paso -1 Hacer

Para j = 1 Hasta i Con Paso 1 Hacer

Escribir " " Sin Saltar

  Fin Para

Para k = i Hasta n Con Paso 1 Hacer

    Escribir "\*" Sin Saltar

  Fin Para

  Escribir ""

Fin Para

n = n - 1

Para i = n Hasta 1 Con Paso -1 Hacer

  Para k = i Hasta n Con Paso 1 Hacer

    Escribir " " Sin Saltar

  Fin Para

  Para j = 1 Hasta i Con Paso 1 Hacer

    Escribir " \*" Sin Saltar

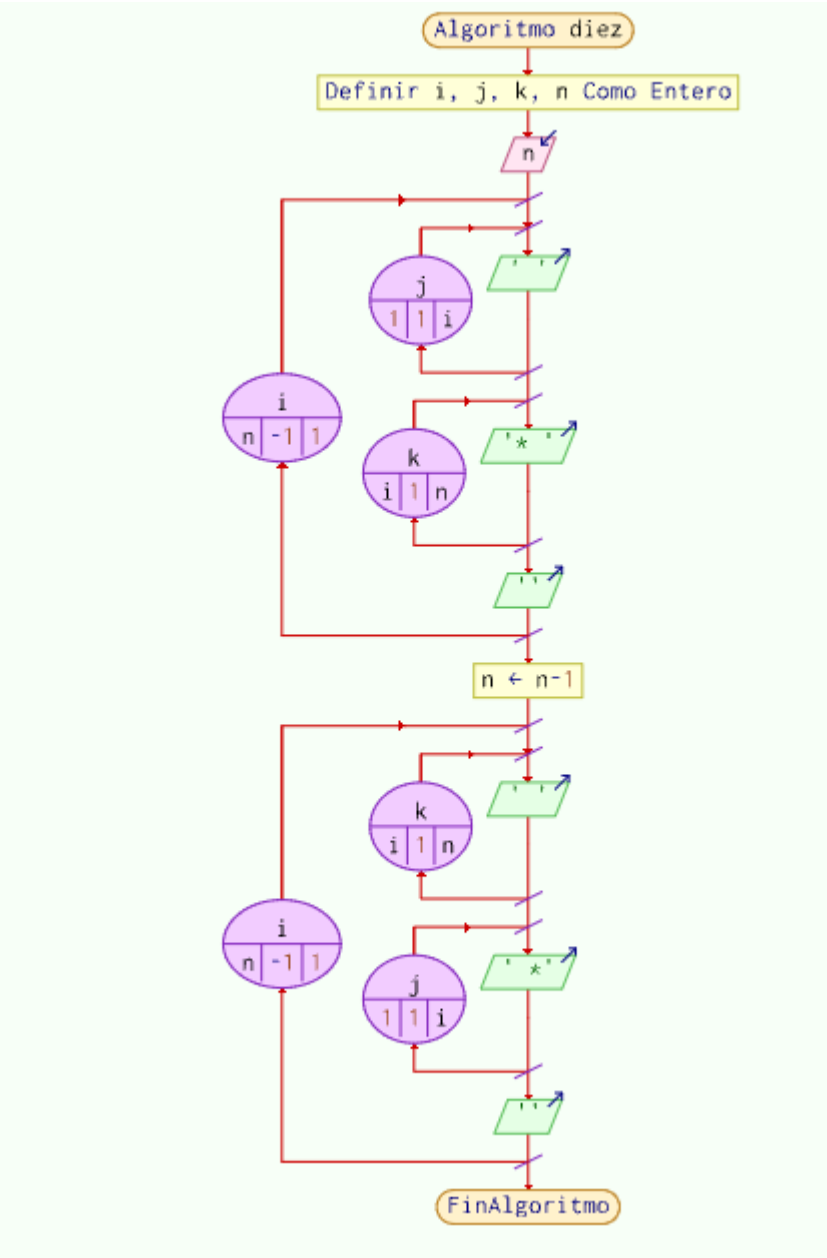
  Fin Para

  Escribir ""

Fin Para

Fin

Diagrama de flujo



**Prueba de escritorio**

Limite	3					
			*			
		*	*	*		
	*	*	*	*	*	
*	*	*	*	*	*	*
	*	*	*	*	*	
		*	*	*		
			*			