

### Ejercicio 2

### 4 Análisis:

Entrada	Proceso	Salida
Producto que se desea	Generacion de precios y	Inventario con precios y
buscar,ingresar o editar.	total de productos.	disponibilidad

## Seudocódigo:

```
Proceso Ejercicio2
  Definir teclado como Scanner
  Definir nombre como Cadena
  Definir nomProductos como Cadena[6] <- {"HELADOS", "DORITOS",
"DESINFECTANTE", "ALINIOS", "GRANOLA", "YOGURT ECOLAC"}
  Definir limProductos como Entero <- 6
  Definir columnas como Entero <- 2
  Definir matPreCan como Real[6][2]
  // Inicializar matriz con precios y cantidades
  matPreCan[0][0] \leftarrow 1
  matPreCan[1][0] \leftarrow 0.50
  matPreCan[2][0] \leftarrow 2.50
  matPreCan[3][0] \leftarrow 0.50
  matPreCan[4][0] \leftarrow 0.75
  matPreCan[5][0] \leftarrow 1.50
  matPreCan[0][1] \leftarrow 100
  matPreCan[1][1] <- 10
  matPreCan[2][1] \leftarrow 50
  matPreCan[3][1] \leftarrow 50
  matPreCan[4][1] \leftarrow 75
  matPreCan[5][1] \leftarrow 80
  Escribir("¿Qué producto busca?")
  nombre <- Leer(). To Upper()
  Para Cada buscar en nomProductos
     Si buscar. ToUpper(). Comienza Con(nombre)
        Escribir("Sí tenemos este producto, aquí tienes nuestro catálogo:")
        Escribir("Producto | Precio | Cantidad")
        Escribir("======="")
        Para i <- O Hasta limProductos - 1
          Escribir(nomProductos[i], "\t")
          Para j <- O Hasta columnas - 1
             Escribir("| %.2f \t", matPreCan[i][j])
```



```
FinPara
Escribir("")
FinPara
Salir
FinSi
FinPara
FinProceso
```

### ♣ Prueba de escritorio:

# Ejercicio 4

### 4 Análisis:

Entrada	Proceso	Salida
Ingreso de "X" o "O" en	Verificación si se	Ganador
las posiciones de las	encuentran 3 posiciones	Perdedor
matrices.	en posición vertical	Nadie gano
	seguidas	Empate

### Seudocódigo:

```
Proceso JuegoTresEnRaya
Dimension tablero[3][3]
Definir jugadorActual como Caracter
Definir juegoEnCurso como Logico
Definir fila, columna como Entero

// Inicializar tablero
Para i <- 1 Hasta 3
Para j <- 1 Hasta 3
tablero[i][j] <- ' '
FinPara
```



```
FinPara
  jugadorActual <- 'X'
  juegoEnCurso <- Verdadero
  Mientras juegoEnCurso
     MostrarTablero(tablero)
     TomarMovimiento(tablero, jugadorActual)
     juegoEnCurso <- No HaGanado(tablero, jugadorActual) Y No TableroLleno(tablero)
     jugadorActual <- Si jugadorActual == 'X' Entonces 'O' Sino 'X'
  FinMientras
  MostrarTablero(tablero)
  DeterminarGanador(tablero)
FinProceso
Procedimiento Mostrar Tablero (tablero)
  Escribir("----")
  Para i <- 1 Hasta 3
     Escribir("| ")
     Para j <- 1 Hasta 3
       Escribir(tablero[i][j], " | ")
     FinPara
     Escribir()
     Escribir("----")
  FinPara
FinProcedimiento
Procedimiento TomarMovimiento(tablero, jugador)
  Definir fila, columna como Entero
  Repetir
     Escribir("Jugador ", jugador, ", ingrese fila (0-2) y columna (0-2) separadas por
espacio:")
     Leer(fila, columna)
  Hasta fila >= 0 Y fila < 3 Y columna >= 0 Y columna < 3 Y tablero[fila + 1][columna + 1] == '
  tablero[fila + 1][columna + 1] <- jugador
FinProcedimiento
Funcion Logico HaGanado(tablero, jugador)
  Para i <- 1 Hasta 3
     Si (tablero[i][1] == jugador Y tablero[i][2] == jugador Y tablero[i][3] == jugador) O
       (tablero[1][i] == jugador Y tablero[2][i] == jugador Y tablero[3][i] == jugador)
Entonces
```

Devolver Verdadero

```
FinSi
  FinPara
  Si (tablero[1][1] == jugador Y tablero[2][2] == jugador Y tablero[3][3] == jugador) O
    (tablero[1][3] == jugador Y tablero[2][2] == jugador Y tablero[3][1] == jugador)
Entonces
     Devolver Verdadero
  FinSi
  Devolver Falso
FinFuncion
Procedimiento DeterminarGanador(tablero)
  Si HaGanado(tablero, 'X') Entonces
     Escribir("iJugador X ha ganado!")
  Sino Si HaGanado(tablero, 'O') Entonces
     Escribir("iJugador O ha ganado!")
  Sino
     Escribir("iEl juego ha terminado en empate!")
  FinSi
FinProcedimiento
Funcion Logico TableroLleno(tablero)
  Para i <- 1 Hasta 3
     Para j <- 1 Hasta 3
       Si tablero[i][j] == ' ' Entonces
          Devolver Falso
       FinSi
     FinPara
  FinPara
  Devolver Verdadero
FinFuncion
```

### Prueba de escritorio:



#### 4 Análisis:

Entrada	Proceso	Salida
Se elije el proceso que	Generación de los datos	Resultado de la operación
se desea realizar, sea	aleatorios de la matriz.	seleccionada que se
suma o resta.		realice entre las
		matrices.

## ♣ Seudocódigo:

```
Proceso Ejercicio6
  Definir matriz1 como Entero[3][3]
  Definir matriz2 como Entero[3][3]
  Definir matrizResult como Entero[3][3]
  Definir i, j como Entero
  Definir scanner como Scanner
  Definir productoIngre como Cadena
  Definir opcion como Entero
  Escribir("PRIMERA MATRIZ")
  Para i <- 0 Hasta 2
     Para j <- 0 Hasta 2
       matriz1[i][j] <- EnteroAleatorioEntre(-9, 9)</pre>
     FinPara
  FinPara
  MostrarMatriz(matriz1)
  Escribir("SEGUNDA MATRIZ")
  Para i <- 0 Hasta 2
     Para j <- 0 Hasta 2
       matriz2[i][j] <- EnteroAleatorioEntre(-9, 9)
     FinPara
  FinPara
  MostrarMatriz(matriz2)
  Escribir("¿QUÉ DESEA REALIZAR?")
  Escribir("(1) Suma")
  Escribir("(2) Resta")
  opcion <- LeerEntero()
  Seleccionar opcion
     Caso 1:
       Para i <- 0 Hasta 2
          Para j <- 0 Hasta 2
```

```
matrizResult[i][j] <- matriz1[i][j] + matriz2[i][j]
          FinPara
        FinPara
        MostrarMatriz(matrizResult)
     FinCaso
     Caso 2:
        Para i <- 0 Hasta 2
          Para j <- 0 Hasta 2
             matrizResult[i][j] <- matriz1[i][j] - matriz2[i][j]
          FinPara
        FinPara
        MostrarMatriz(matrizResult)
  FinSeleccionar
FinProceso
Funcion Entero Aleatorio Entre (minimo, maximo)
  Devolver Entero Aleatorio Entre (minimo, maximo)
FinFuncion
Procedimiento MostrarMatriz(matriz)
  Para i <- 0 Hasta 2
     Para j <- 0 Hasta 2
       Escribir("[" + matriz[i][j] + "]" + "\t")
     FinPara
     Escribir("")
  FinPara
  Escribir("")
FinProcedimiento
```

### Prueba de escritorio:



[4] [-2] [6] [9] [0] [-4] [3] [-5] [4] [5] [-1] [-3] [5] [-1] [-3] [5] [-1] [-3] [5] [-1] [-3] [6] [7] [4] [0] [-8] [1] [6] [7] [9] [4] [6] [7] [9] [4] [7] [9] [4] [7] [9] [4] [7] [9] [4] [7] [9] [9] [9] [9] [9] [9] [9] [9] [9] [9	PRIMERA	A MATRIZ		PRIMER	RA MATRIZ	
[3] [-5] [4] [5] [-1] [-3]  SEGUNDA MATRIZ SEGUNDA MATRIZ  [0] [7] [4] [0] [-8] [1]  [5] [1] [-8] [-5] [5] [-2]  [4] [0] [0] [-1] [9] [4]  ¿QUE DESEA REALIZAR?  (1) Suma  (2) Resta  1  [-7] [-1] [8] [2] [13] [-9]  [9] [-1] [-2] [14] [-5] [-2]	[-7]	[-8]	[4]	[2]	[5]	[-8]
[3] [-5] [4] [5] [-1] [-3]  SEGUNDA MATRIZ SEGUNDA MATRIZ  [0] [7] [4] [0] [-8] [1]  [5] [1] [-8] [-5] [5] [-2]  [4] [0] [0] [-1] [9] [4]  ¿QUE DESEA REALIZAR? ¿QUE DESEA REALIZAR  (1) Suma (2) Resta (2) Resta  1 2  [-7] [-1] [8] [2] [13] [-9]  [9] [-1] [-2] [14] [-5] [-2]	[4]	[-2]	[6]	[9]	[0]	[-4]
[0] [7] [4] [0] [-8] [1] [5] [1] [5] [1] [-8] [1] [5] [1] [-8] [1] [5] [-2] [4] [0] [0] [-1] [9] [4] [2] [2] [2] [2] [2] [2] [2] [2] [2] [2		[-5]		[5]	[-1]	[-3]
[5] [1] [-8] [-5] [5] [-2 [4] [0] [0] [-1] [9] [4] ¿QUE DESEA REALIZAR? ¿QUE DESEA REALIZAR (1) Suma (1) Suma (2) Resta (2) Resta 1 2 [-7] [-1] [8] [2] [13] [-9 [9] [-1] [-2] [14] [-5] [-2	SEGUNDA MATRIZ		SEGUNDA MATRIZ			
[5] [1] [-8] [-5] [5] [-2] [4] [0] [0] [-1] [9] [4] 2QUE DESEA REALIZAR? 2QUE DESEA REALIZAR (1) Suma (2) Resta (2) Resta 1 2 [-7] [-1] [8] [2] [13] [-9] [9] [-1] [-2] [14] [-5] [-2]	[0]	[7]	[4]	[0]	[-8]	[1]
[4] [0] [0] [-1] [9] [4]  ¿QUE DESEA REALIZAR? (1) Suma (1) Suma (2) Resta (2) Resta 1 2 [-7] [-1] [8] [2] [13] [-9]  [9] [-1] [-2] [14] [-5] [-2		[1]	[-8]	[-5]	[5]	[-2]
¿QUE DESEA REALIZAR? (1) Suma (2) Resta (3) Resta (4) Resta (5) Resta (6) Resta (7) [-1] [-1] [-2] [-2] [-3] [-3] [-9] (9) [-1] [-2] [-2] [-3] [-3] [-3]		[0]	[0]	[-1]	[9]	[4]
(1) Suma (1) Suma (2) Resta (2) Resta 1 2 [-7] [-1] [8] [2] [13] [-9] [9] [-1] [-2] [14] [-5] [-2]			LIZAR?	ZQUE I	ESEA REA	LIZAR?
(2) Resta  1 2  [-7] [-1] [8] [2] [13] [-9]  [9] [-1] [-2] [14] [-5] [-2	_			(1) Sum	na	
1 2 [-7] [-1] [8] [2] [13] [-9] [9] [-1] [-2] [14] [-5] [-2]				(2) Res	sta	
[9] [-1] [-2] [14] [-5] [-2				2		
[9] [-1] [-2] [14] [-5] [-2	[-7]	[-1]	[8]	[2]	[13]	[-9]
[6] [10] [7	-			[14]	[-5]	[-2]
	[7]	[-5]	[4]	[6]	[-10]	[-7]

# Ejercicio 8

## **4** Análisis:

Entrada	Proceso	Salida
Se ingresa el limite de la matriz cuadrada.	Generación de los datos aleatorios de la matriz.	Se presenta la diagonal principal de la matriz. Se presentan los datos sobre la matriz. Se presentan los datos debajo la matriz.

# ♣ Seudocódigo:

Proceso Ejercicio8

Definir limMt como Entero <- 3

Definir mt como Entero[limMt][limMt]

Definir i, j, d como Entero

Funcion EnteroAleatorioEntre(minimo, maximo)

Devolver AleatorioEntre(minimo, maximo)

FinFuncion

```
Para i <- O Hasta limMt - 1
  Para j <- 0 Hasta limMt - 1
     mt[i][j] <- EnteroAleatorioEntre(-9, 9)
  FinPara
FinPara
// Imprimir la matriz completa
Escribir("Matriz:")
Para i <- O Hasta limMt - 1
  Para j <- 0 Hasta limMt - 1
     Escribir("[" + mt[i][j] + "]" + "\t")
  FinPara
  Escribir("")
FinPara
Escribir("")
// Imprimir la diagonal principal
Escribir("La diagonal principal es:")
Para d <- 0 Hasta limMt - 1
  Escribir("[" + mt[d][d] + "]")
FinPara
Escribir("")
// Imprimir los números por encima de la diagonal principal
Escribir("Numeros por encima de la diagonal principal son:")
Escribir("[" + mt[0][1] + "]")
Escribir("[" + mt[0][2] + "]")
Escribir("[" + mt[1][2] + "]")
Escribir("")
// Imprimir los números por debajo de la diagonal principal
Escribir("Numeros por debajo de la diagonal principal son:")
Escribir("[" + mt[1][0] + "]")
Escribir("[" + mt[2][0] + "]")
Escribir("[" + mt[2][1] + "]")
Escribir("")
```

FinProceso

♣ Prueba de escritorio:



```
[4] [5] [-3]
[0] [9] [9]
[-3] [-4] [5]

La diagonal principal es:
[4]
[9]
[5]

Numeros por encima de la diagonal principal son:
[5][-3][9]

Numeros por debajo de la diagonal principal son:
[0][-3][-4]
```

```
[8] [9] [0]
[4] [5] [-6]
[7] [4] [-8]

La diagonal principal es:
[8]
[5]
[-8]

Numeros por encima de la diagonal principal son:
[9][0][-6]

Numeros por debajo de la diagonal principal son:
[4][7][4]
```

## Ejercicio 10

### 4 Análisis:

<b>♣</b> Entrada	Proceso	Salida
Se ingresa la altura	Generacion de la figura	Se presenta la figura de
deseada de la figura	utilizando dos "for" para	acuerdo a la altura
	las filas y columnas.	ingresada anteriormente.

## ♣ Seudocódigo:

```
Inicio
       altura = 4
     ImprimirPatronTriangular(altura)
   Para i desde O hasta altura - 1 hacer
      Para j desde O hasta altura - i - 2 hacer
          Imprimir " "
      Fin Para
      Para j desde O hasta i hacer
          Imprimir "* "
       Fin Para
      Imprimir nueva línea
   Fin Para
   Para i desde altura - 2 hasta 0 hacer
     Para j desde O hasta altura - i - 2 hacer
       Imprimir " "
     Fin Para
     Para j desde O hasta i hacer
       Imprimir "* "
     Fin Para
     Imprimir nueva línea
   Fin Para
Fin Procedimiento
```



# ♣ Prueba de escritorio:

\*
\* \* \*
\* \* \*
\* \* \* \*