

## ACTIVIDADES ESTRUCTURAS DE ALMACENAMIENTO: MATRICES

### Actividad 1

Crear un programa que implemente operaciones sobre matrices

- `void leerMatriz(int [][]m);`  
una función que sea los valores de una matriz
- `void mostrarMatriz(int [][]m);`  
una función que muestre los valores de una matriz por filas
- `void sumaFilas(int [][]m);`  
una función que muestre las sumas de los valores de una matriz por filas
- `void sumaColumnas (int [][]m);`  
una función que muestre las sumas de los valores de una matriz por columnas
- `int [][] intercambia(int [][]m);`  
una función que retorna una nueva matriz a partir de la matriz parámetro intercambiando filas por columnas y viceversa  
 $m[i][j] = \text{retorno}[j][i]$
- `boolean esSelectiva(int [][]m)`  
determinar si una matriz es selectiva. Una matriz es selectiva, si es cuadrada y todos los elementos de su diagonal son 1
- `boolean tienePuntoSilla(int [][]m)`

función que determine si una matriz tiene un punto silla. Un punto silla, es una posición(i,j) que contiene el valor más grande de su fila que es a la vez el valor más pequeño de su columna

```
Digite Elemento [1][0] :13
Digite Elemento [1][1] :15
Digite Elemento [1][2] :17

Digite Elemento [2][0] :4
Digite Elemento [2][1] :9
Digite Elemento [2][2] :32

      MATRIZ
-----
8      1      26
13     15     17
4       9     32
En la matriz el Punto de Silla es Fila1, Columna 2 = 17
17 es mayor en la fila 1, ya su vez es menor en la Columna 2
Presione una tecla para continuar . . . ■
```

## Actividad 2

Crear un programa con las siguiente funciones

- `int ** crearMatriz()`

función que lea la dimension de una matriz cuadrada. Genera los valores de la matriz a partir de la formula  $m[i][j]=i+j$

- `void mostrarMatriz(int m)**`

función que muestra el contenido de una matriz por filas

- `void trianguloSuperior(int **m)`

función que muestre el triangulo superior de una matriz

- `void trianguloInferior(int **m)`

función que muestre el triangulo inferior de una matriz

### Actividad 3

Implementar un programa que simule un **reloj de arena**. Para lo cual seguiremos los siguientes pasos

- crear una matriz cuadrada de orden impar ( la dim de la matriz es impar)
- rellenar la matriz con valores aleatorios entre 1 y 9
- mostrar la matriz con **todos** sus datos
- mostrar la matriz en forma de reloj de arena

ejemplo M de orden 11

MATRIZ COMPLETA										
0	1	2	3	4	5	6	7	8	9	10
2	3	4	5	6	7	8	9	10	11	12
4	5	6	7	8	9	10	11	12	13	14
6	7	8	9	10	11	12	13	14	15	16
8	9	10	11	12	13	14	15	16	17	18
10	11	12	13	14	15	16	17	18	19	20
12	13	14	15	16	17	18	19	20	21	22
14	15	16	17	18	19	20	21	22	23	24
16	17	18	19	20	21	22	23	24	25	26
18	19	20	21	22	23	24	25	26	27	28
20	21	22	23	24	25	26	27	28	29	30

RELOJ DE ARENA										
0	1	2	3	4	5	6	7	8	9	10
	3	4	5	6	7	8	9	10	11	
		6	7	8	9	10	11	12		
			9	10	11	12	13	14		
					12	13	14			
						15				
						16	17	18		
						17	18	19	20	21
						18	19	20	21	22
						19	20	21	22	23
						20	21	22	23	24
						21	22	23	24	25
						22	23	24	25	26
						23	24	25	26	27
						24	25	26	27	28
						25	26	27	28	29
						26	27	28	29	30