



## Laboratorio de Estructuras de Datos

### Práctica 7. Arreglos multidimensionales

Unidad Temática: 1. Introducción a las estructuras de datos y estructuras fundamentales

📄 👤 Profesor: Dr. Aldonso Becerra Sánchez

#### Índice

1	Objetivo de la tarea	1
2	Tiempo aproximado de realización	1
3	Fecha de entrega	1
4	Fecha de entrega con extensión y penalización	1
5	Introducción	1
6	Actividades a realizar	1
6.1	Actividad Inicial	1
6.2	Actividad 1	1
6.3	Actividad 2	1
6.4	Actividad 3	2
6.5	Actividad 4	2
6.6	Actividad 5	2
6.7	Actividad 6	2
6.8	Actividad 7	2
7	Contáctame	2
	References	2

#### 1. Objetivo de la tarea

Comprender el uso de los arreglos multidimensionales numéricos para la definición de una base API funcional para su futura aplicación en entornos reales.

#### 2. Tiempo aproximado de realización

🕒 5 horas.

#### 3. Fecha de entrega

📅 13 de septiembre de 2024.

#### 4. Fecha de entrega con extensión y penalización

📅 14 de septiembre agosto de 2024.

#### 5. Introducción

La facilidad que los arreglos multidimensionales tienen para permitir guardar datos en forma de columnas/renglones, los hace pertinentes para la resolución de muchos problemas donde se requiere esta situación. El único detalle con esta cuestión es que es poco

flexible el número de elementos que podemos manipular, ya que se requiere conocer a priori la cantidad de elementos a guardar [1], [2], [3].

#### 6. Actividades a realizar

##### 6.1. Actividad Inicial

Lea primero toda la práctica ⚠️. No inicie a programar sin leer todo cuidadosamente primero. Recuerde que debe generar el reporte en formato IDC con todos sus componentes.

##### 6.2. Actividad 1

Primero genere la Introducción 📄.

##### 6.3. Actividad 2

#### Información importante

Esta actividad debe entrar en la parte de *Desarrollo* 📄.

Defina un TDA llamado “ArregloListaInfoEstatica2Numeros” que herede de “ArregloListaInfoEstatica2Numeros”, de modo que este tipo de matriz pueda realizar lo siguiente:

- Valide que todos los valores pasados a cada método de esta clase sean estrictamente heredados de “Number”, tal como se hizo en ArregloListaInfoEstaticaNumeros. Para esto deberá sobrescribir e invocar los métodos de la super clase de tal manera que se cumpla este cometido también en esos métodos. Guarde los valores como double de manera predeterminada.
- Constructor: (int filas, int columnas).** Que llame a inicializar con valores en cero de manera predeterminada.
- Constructor: (int filas, int columnas, Object valor).** Que llame a inicializar con valores indicados por contenido.
- Multiplicar un escalar por la matriz: boolean porEscalar(Number escalar).** Debe validar las dimensiones y el procedimiento de acuerdo a la teoría de matrices.
- Multiplicar un escalar para cada elemento de la matriz: boolean porEscalares(ArregloListaInfoEstaticaNumeros escalares).** Multiplica el elemento de una posición del arreglo de números (cada posición contiene un escalar) por un solo elemento colocado en la posición correspondiente en la matriz, recorriéndola en orden natural (renglones, columnas). Quiere

49 decir que el escalar de la posición 0 del arreglo se multiplica  
50 por el elemento 0,0 de la matriz, el escalar de la posición 1 del  
51 arreglo se multiplica por el elemento 0,1 de la matriz, etc.

52 6. Sumar un escalar a la matriz: `boolean sumarEscalar(Number`  
53 `escalar)`. Debe validar las dimensiones y el procedimiento de  
54 acuerdo a la teoría de matrices.

55 7. **Sumar un escalar para cada elemento de la matriz: `boolean`**  
56 **`sumarEscalares(ArregloListaInfoEstaticaNumeros`**  
57 **`escalares)`**. Suma un elemento contenido en una posición del  
58 arreglo de números (cada posición contiene un escalar) más un  
59 solo elemento colocado en la posición correspondiente en la  
60 matriz, recorriéndola en orden natural (renglones, columnas).  
61 Quiere decir que el escalar de la posición 0 del arreglo se suma al  
62 elemento 0,0 de la matriz, el escalar de la posición 1 del arreglo  
63 se suma al elemento 0,1 de la matriz, etc.

64 8. **Multiplicar por otra matriz: `boolean multiplicar`**  
65 **`(ArregloListaInfoEstatica2Numeros matriz2)`**. Debe  
66 validar las dimensiones y el procedimiento de acuerdo a la  
67 teoría de matrices.

68 9. **Sumar con otra matriz: `boolean sumar`**  
69 **`(ArregloListaInfoEstatica2Numeros matriz2)`**.  
70 Debe validar las dimensiones y el procedimiento de acuerdo a la  
71 teoría de matrices.

72 10. **Elevar a una potencia la matriz (elemento por elemen-**  
73 **`to): boolean aplicarPotencia(Number escalar)`**. Aplicar la  
74 potencia elemento por elemento.

75 11. **Aplicar el logaritmo (elemento por elemento) a la matriz:**  
76 **`boolean aplicarLogaritmo(TipoLogaritmo tipoLogaritmo)`**.  
77 Valide que si existe un valor en la matriz, no se pue-  
78 de aplicar el logaritmo. El proceso es elemento por elemento.  
79 Defina un enumerado llamado `TipoLogaritmo`, que indica que  
80 es `NATURAL`(base e), `BASE10` o `BASE2`. Este tipo será pasado  
81 como argumento al método.

82 12. **Generar una matriz diagonal con el valor proporcionado:**  
83 **`boolean matrizDiagonal(Number contenido)`**.

84 13. **Determinar si la matriz es una matriz triangular superior:**  
85 **`boolean esDiagonalSuperior()`**.

86 14. **Determinar si la matriz es una matriz triangular inferior:**  
87 **`boolean esDiagonalInferior()`**.

88 15. **Determinar la potencia de una matriz. `boolean potencia`**  
89 **`(int exponente)`**. Aplicar esta operación dado por  $A^2 =$   
90  $AA, A^3 = AAA$ , y así en en general.

91 16. Redimensionar (sumando) una matriz por columnas, a la mitad  
92 (si la matriz no tiene un número par de columnas, la del centro  
93 debe pasar intacta, solo debe acumular las demás), de tal forma  
94 que **`boolean doblarReglones()`**: 4 6 4 3  
95 1 2 2 4  
96 5 3 2 1  
97 0 8 7 3

98  
99 Quedaría así:  
100 10 7  
101 3 6  
102 8 3  
103 8 10  
104

17. Redimensionar (sumando) una matriz por renglones, a la mitad  
(si la matriz no tiene un número par de renglones, el del centro  
debe pasar intacto, solo debe acumular los demás), de tal forma  
que **`boolean doblarColumnas()`**: 4 6 4 3  
1 2 2 4  
5 3 2 1  
0 8 7 3

Quedaría así:  
5 8 6 7  
5 11 9 4


#### 6.4. Actividad 3

Pruebe el funcionamiento del programa de la actividad 2 con todo y  
sus capturas de pantalla.

#### 6.5. Actividad 4

Realice la sección de Código agregado  (diagrama de clases UML).





#### 6.6. Actividad 5

Realice la sección de Pre-evaluación  (use los lineamientos esta-  
blecidos).

#### 6.7. Actividad 6

Finalmente haga las Conclusiones .


#### 6.8. Actividad 7


Subir los entregables (pdf  y zip  con código )  a Moodle.


### 7. Contáctame

Puedes contactarme a través de los siguientes medios.

 <https://moodle.ingsoftware.uaz.edu.mx/>

 a7donso@gmail.com

 Cubículo

 Salón CC2-IS

### Referencias

- [1] O. Cairo y S. Guardati, *Estructura de datos*. McGraw-Hill.
- [2] L. Joyanes Aguilar, *Fundamentos de programación, algoritmos u estructura de datos*. McGraw-Hill.
- [3] M. A. Weiss, *Estructura de datos en Java*. Addison Wesley.