



UNIVERSIDAD AUTÓNOMA DE CAMPECHE

**NOMBRE DEL PROFESOR:** Ing. Héctor Manuel Quej Cosgaya

**NOMBRE DE LA PRÁCTICA:** Arreglos

**PRÁCTICA NÚM.** [ 2 ]

<b>LABORATORIO:</b>	Centro de Ingeniería Computacional
<b>MATERIA:</b>	Lenguaje de Programación I
<b>UNIDAD:</b>	Subcompetencia II
<b>TIEMPO:</b>	2 horas

**OBJETIVO:**

Comprender el concepto, funcionamiento y operaciones básicas con arreglos, así como reconocer la utilidad que brindan.

**MARCO TEÓRICO:**

Los arreglos son estructuras fundamentales de cualquier programa, ya que permiten manipular grandes cantidades de datos de una manera ordenada y eficiente. En la práctica, se procede a ejemplificar los principales pasos que hay que realizar al trabajar con arreglos. Se evita el uso de estructuras iterativas, pues aún no han sido cubiertas en el curso.

**LISTA DE MATERIALES:**

- Java SDK
- Bloc de notas / Editor SciTE

**EQUIPO DE LABORATORIO:**

- Computadora Personal

**DESARROLLO DE LA PRÁCTICA:**

Primera parte: **Declaración y construcción de arreglos.**

1. Abre el **Bloc de Notas**.
2. Escribe el siguiente código fuente en el **Bloc de Notas**. ¡Recuerda copiarlo **exactamente** igual!



## UNIVERSIDAD AUTÓNOMA DE CAMPECHE

```
public class Practica2 {
    public static void main(String[] args) {
        // Primera parte : Declaración y construcción de arreglos
        int[] arreglo;
        int tamañoArreglo = 5;
        arreglo = new int[tamañoArreglo];

        System.out.println("Los " + arreglo.length +
            " elementos del arreglo: ");
        System.out.println(arreglo[1]);
        System.out.println(arreglo[2]);
        System.out.println(arreglo[3]);
        System.out.println(arreglo[4]);
        System.out.println(arreglo[5]);
    }
}
```

3. **Guarda** el archivo como '**Practica2.java**'. Asegúrate que sea en una ubicación a la que puedas acceder fácilmente.
4. **Compila y ejecuta** el programa. ¿Qué sucede?

Siempre recuerda que los **índices** de los arreglos empiezan a contarse a partir del **cero**.

5. Resta uno a cada índice del arreglo (el número entre corchetes) en las sentencias de impresión. **Compila** y **ejecuta** el programa de nuevo.

**Segunda parte: Inicialización automática**

6. Añade el siguiente código después de la última instrucción que escribiste en el paso 2:

```
// Segunda parte: Inicialización automática
int[] calificaciones = { 10, 8, 6, 9, 9, 7};
System.out.println("Los " + calificaciones.length +
    " elementos de calificaciones: ");
System.out.println("Elemento 1: " + calificaciones[0]);
System.out.println("Elemento 2: " + calificaciones[1]);
System.out.println("Elemento 3: " + calificaciones[2]);
System.out.println("Elemento 4: " + calificaciones[3]);
System.out.println("Elemento 5: " + calificaciones[4]);
System.out.println("Elemento 6: " + calificaciones[5]);
```

7. **Compila y ejecuta** el programa.

**UNIVERSIDAD AUTÓNOMA DE CAMPECHE**

Observa que no indicamos el tamaño del arreglo, sino que el compilador lo deduce de manera automática a partir de la cantidad de elementos que le indicamos. Igual nota que el último elemento de un arreglo siempre se encuentra en la posición (**tamaño del arreglo – 1**).

**Tercera parte: Arreglos multidimensionales.**

8. Añade el siguiente código después de la última instrucción que escribiste en el paso 6:

```
// Tercera parte: Arreglos multidimensionales
int[][] matriz = new int[2][2];
matriz[0][0] = 11;
matriz[0][1] = 12;
matriz[1][0] = 21;
matriz[1][1] = 22;

System.out.println("Una matriz de 2 x 2:");
System.out.println("[ " + matriz[0][0] + " " + matriz[0][1] + " ]");
System.out.println("[ " + matriz[1][0] + " " + matriz[1][1] + " ]");
```

9. **Compila y ejecuta** el programa.

Como son arreglos, las matrices también pueden **inicializarse** de forma automática. Veamos un ejemplo:

10. Agrega el siguiente código después de la última instrucción que escribiste en el paso 8:

```
int[][] noMatriz = { {1, 2, 3} , {4, 5} , {6, 7, 8, 9} };
System.out.println("Una no-matriz inicializada automáticamente: ");
System.out.println("[ " + noMatriz[0][0] + " " + noMatriz[0][1] +
    " " + noMatriz[0][2] + " ]");
System.out.println("[ " + noMatriz[1][0] + " " + noMatriz[1][1] +
    " ]");
System.out.println("[ " + noMatriz[2][0] + " " + noMatriz[2][1] +
    " " + noMatriz[2][2] + " " + noMatriz[2][3] + " ]");
```

11. **Compila y ejecuta** el programa.

**Fin de la Práctica.**

**RETROALIMENTACIÓN:**

- Utilizando lo aprendido en las prácticas 1 y 2, desarrolla un programa que cree dos matrices de 3 x 3, las llene con valores introducidos por el usuario, e imprima su suma.



FACULTAD DE INGENIERÍA

**FORMATO**  
**PRÁCTICAS DE LABORATORIO**

---

**UNIVERSIDAD AUTÓNOMA DE CAMPECHE**

<b>RECOMENDACIONES ADICIONALES:</b>
-------------------------------------

- |  |
|--|
| <ul style="list-style-type: none"><li>• Leer el capítulo 10 del Dean (Arreglos y Listas de Arreglos)</li></ul> |
|--|

<b>BIBLIOGRAFÍA:</b>
----------------------

- |   |
|---|
| <ul style="list-style-type: none"><li>• Dean, J. S., &amp; Dean, R. H. (2009). Introducción a la programación con Java. México: Mc Graw Hill.</li><li>• Apuntes del profesor.</li></ul> |
|---|