

GUÍA DE LABORATORIO NO. 2

HANNA KATHERINE ABRIL GÓNGORA

JENNIFER NATALIA BELTRAN

FELIPE ARAUJO

UNIVERSIDAD MANUELA BELTRÁN

ESTRUCTURA DE DATOS

DOCENTE

HUGO ALFONSO ORTIZ BARRERO

BOGOTA DC 24 FEBRERO 2025

1. Preguntas orientadoras

1.1. ¿Cuáles fueron los aprendizajes obtenidos al realizar esta guía?, liste como mínimo 3 aprendizajes y relaciónelos con su futuro que hacer profesional.

- *Comprensión y aplicación de estructuras de datos dinámicas*

A medida que el guía pasó, se desarrollaron ideas más claras de estructuras de datos como listas vinculadas, pistas y cuevas. La capacidad para ejecutar estos marco se obtuvo adecuadamente de acuerdo con los requisitos del problema. Esta información importa mucho en el trabajo porque muchos trabajos necesitan manejar grandes cantidades de datos de manera eficiente. Desarrollar programas para sistemas de administración de datos, juego de realidad virtual o herramientas de inteligencia sintética. Estas— estructuras permiten el almacenamiento y recuperación de datos de manera eficiente.

- *Desarrollo del pensamiento lógico y algorítmico*

La resolución eficaz de problemas requiere diseccionar problemas, romperlos y emplear algoritmos adecuados para lograr el mejor resultado En este campo, la creación de algoritmos óptimos es una habilidad clave crucial para el éxito en la codificación, la seguridad de los sistemas digitales y el análisis de la información La capacidad en el futuro permitirá abordar proyectos de optimización, resolver problemas de rendimiento en bases de datos, y diseñar arquitecturas de software escalables.

- *Manejo de errores y depuración de código*

Se aprendió la importancia de detectar y corregir errores en la implementación de estructuras de datos, lo que involucró depuración de código, pruebas unitarias y validación de resultados. Esta habilidad es esencial en el mundo laboral, ya que cualquier sistema informático está sujeto a fallos y errores de ejecución. Un desarrollador con experiencia en depuración puede mejorar la calidad del código, reducir tiempos de desarrollo y evitar fallos críticos en producción.

1.2. ¿Dónde presento mayor dificultad resolviendo la guía? y como lo resolvieron cuales fueron las estrategias de solución?

Entender la lógica de las estructuras de datos.

Al principio, no estaba completamente claro cómo funcionan con las listas y colas vinculadas y la peculiaridad de trabajar con punteros y referencias en memoria.

Estrategia. al respecto, nos ayudaron los videos en YouTube adicionales y las instrucciones sobre algunos proyectos, Además, antes de resolver problemas más complejos, tomamos y realizamos ejemplos más simples para comprender cómo obtener el resultado final que se necesitaba.

2. Mencione 9 situaciones en la que se utilizan los arreglos. (3 unidimensionales, 3 bidimensionales, 3 multidimensionales)

2.1. Arreglos Unidimensionales (Vectores)

- *Registro de temperaturas diarias:* Se usa un arreglo para almacenar la temperatura de cada día del mes.

```
public class Temperaturas {  
    public static void main(String[] args) {  
        double[] temperaturas = {20.5, 21.3, 19.8, 22.1, 18.5}; // Ejemplo de 5 días  
        for (int i = 0; i < temperaturas.length; i++) {  
            System.out.println("Día " + (i + 1) + ": " + temperaturas[i] + "°C");  
        }  
    }  
}
```

- *Lista de calificaciones de un estudiante:* Un arreglo almacena las calificaciones de un alumno en distintas materias.

```
public class Calificaciones {  
    public static void main(String[] args) {  
        int[] calificaciones = {85, 90, 78, 92, 88}; // Ejemplo de 5 materias  
        for (int i = 0; i < calificaciones.length; i++) {  
            System.out.println("Materia " + (i + 1) + ": " + calificaciones[i]);  
        }  
    }  
}
```

- *Inventario de productos en una tienda:* Se usa un arreglo para guardar los nombres de productos disponibles.

```

public class Inventario {
    public static void main(String[] args) {
        String[] productos = {"Laptop", "Teclado", "Mouse", "Monitor", "Impresora"};
        for (String producto : productos) {
            System.out.println("Producto disponible: " + producto);
        }
    }
}

```

2.2. Arreglos Bidimensionales (Matrices)

- *Tablas de multiplicación:* Se puede usar un arreglo bidimensional para almacenar los valores de una tabla de multiplicación.

```

public class TablaMultiplicacion {
    public static void main(String[] args) {
        int[][] tabla = new int[10][10];
        for (int i = 1; i <= 10; i++) {
            for (int j = 1; j <= 10; j++) {
                tabla[i - 1][j - 1] = i * j;
                System.out.print(tabla[i - 1][j - 1] + "\t");
            }
            System.out.println();
        }
    }
}

```

- *Tablero de ajedrez o sudoku:* Se usa una matriz de 8x8 (ajedrez) o 9x9 (sudoku) para representar la posición de las piezas o números.

```

public class TableroAjedrez {
    public static void main(String[] args) {
        char[][] tablero = new char[8][8];
        for (int i = 0; i < 8; i++) {
            for (int j = 0; j < 8; j++) {
                tablero[i][j] = ((i + j) % 2 == 0) ? 'B' : 'N'; // B = Blanco, N = Negro
                System.out.print(tablero[i][j] + " ");
            }
            System.out.println();
        }
    }
}

```

- *Imagen en escala de grises:* Una imagen en blanco y negro se almacena como una matriz donde cada posición representa un píxel con un valor de intensidad.

```
public class ImagenGrises {
    public static void main(String[] args) {
        int[][] imagen = {
            {0, 50, 100},
            {150, 200, 255},
            {75, 125, 175}
        };
        for (int[] fila : imagen) {
            for (int pixel : fila) {
                System.out.print(pixel + "\t");
            }
            System.out.println();
        }
    }
}
```

2.3. Arreglos Multidimensionales

- *Imágenes en color (RGB):* Se usa un arreglo tridimensional donde cada píxel tiene valores para los colores rojo, verde y azul.

```
public class ImagenRGB {
    public static void main(String[] args) {
        int[][][] imagen = {
            {{255, 0, 0}, {0, 255, 0}, {0, 0, 255}}, // Rojo, Verde, Azul
            {{255, 255, 0}, {0, 255, 255}, {255, 0, 255}} // Amarillo, Cyan, Magenta
        };
        for (int[][] fila : imagen) {
            for (int[] pixel : fila) {
                System.out.print("(" + pixel[0] + "," + pixel[1] + "," + pixel[2] + ") ");
            }
            System.out.println();
        }
    }
}
```

- *Simulación de datos meteorológicos:* Se usa un arreglo tridimensional donde cada dimensión representa temperatura, humedad y presión en distintos puntos geográficos.

```

public class DatosMeteorologicos {
    public static void main(String[] args) {
        double[][][] datos = new double[3][3][3]; // 3 ubicaciones, 3 días, 3 variables (temp, humedad, presión)
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                datos[i][j][0] = 20 + Math.random() * 10; // Temperatura
                datos[i][j][1] = 40 + Math.random() * 60; // Humedad
                datos[i][j][2] = 1000 + Math.random() * 100; // Presión
                System.out.println("Ubicación " + (i + 1) + ", Día " + (j + 1) +
                    " -> Temp: " + datos[i][j][0] + "°C, Hum: " + datos[i][j][1] + "%, Presión: " + datos[i][j][2] + "hPa");
            }
        }
    }
}

```

- *Videojuegos en 3D*: Se usan arreglos tridimensionales o más para representar la posición y propiedades de los objetos en un entorno 3D.

```

public class Mundo3D {
    public static void main(String[] args) {
        int[][][] mundo = new int[5][5][5]; // Espacio 3D de 5x5x5
        mundo[2][2][2] = 1; // Un objeto en el centro
        for (int x = 0; x < 5; x++) {
            for (int y = 0; y < 5; y++) {
                for (int z = 0; z < 5; z++) {
                    System.out.print(mundo[x][y][z] + " ");
                }
                System.out.print(" | ");
            }
            System.out.println();
        }
    }
}

```