

INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	ESTRUCTURA DE DATOS Y ALGORITMOS				
TÍTULO DE LA PRÁCTICA:	TÉCNICAS Y DISEÑO DE ALGORITMOS				
NÚMERO DE PRÁCTICA:	02	AÑO LECTIVO:	2025 – A	NRO. SEMESTRE:	Tercero III
FECHA DE PRESENTACIÓN	17/05/25	HORA DE PRESENTACIÓN			
INTEGRANTE (s): Aragón Carpio Fredy José				NOTA:	
DOCENTE(s): <ul style="list-style-type: none"> Mg. Ing. Rene Alonso Nieto Valencia. 					

SOLUCIÓN Y RESULTADOS
<p>I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS https://github.com/DARSEN12/Laboratorio_EDA_E.git</p> <h3>α. PROBLEMAS DESARROLLADOS</h3> <p>1. Ejercicio: Implementación de un método recursivo.</p> <div style="display: flex; justify-content: space-between; margin-top: 20px;"> <pre style="background-color: #2e3436; color: #eeeeec; padding: 10px; width: 45%;"> public class Recursividad { void repetir() { repetir(); } public static void main(String[] ar) { Recursividad re = new Recursividad(); re.repetir(); } } </pre> <div style="background-color: #2e3436; color: #eeeeec; padding: 10px; width: 45%;"> // El programa entra en una llamada recursiva sin fin y termina con un error de desbordamiento de pila. </div> </div>

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 2</p>

2. Ejercicio: Implementación de un método recursivo que reciba un parámetro de tipo entero y luego llame en forma recursiva con el valor del parámetro menos 1.

```
public class Recursividad {

    void imprimir(int x) {
        System.out.println(x);
        imprimir(x - 1);
    }

    public static void main(String[] ar) {
        Recursividad re = new Recursividad();
        re.imprimir(5);
    }
}
```

```
// El método imprime números
decrecientes recursivamente sin
detenerse, causando recursión infinita
y error de pila.
```

3. Ejercicio: Implementar un método recursivo que imprima en forma descendente de 5 a 1 de uno en uno.

```
public class Recursividad {

    void imprimir(int x) {
        if (x > 0) {
            System.out.println(x);
            imprimir(x - 1);
        }
    }

    public static void main(String[] ar) {
        Recursividad re = new Recursividad();
        re.imprimir(5);
    }
}
```

```
// Imprime números decrecientes desde
x hasta 1 usando recursión controlada
con caso base.
```

4. Ejercicio: Imprimir los números de 1 a 5 en pantalla utilizando recursividad.

```
public class Recursividad {

    void imprimir(int x) {
        if (x > 0) {
            imprimir(x - 1);
            System.out.println(x);
        }
    }

    public static void main(String[] ar) {
        Recursividad re = new Recursividad();
        re.imprimir(5);
    }
}
```

```
// Imprime los números del 1 al 5 en
orden ascendente usando recursión.
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 3</p>

- 5. Ejercicio: Obtener el factorial de un número. Recordar que el factorial de un número es el resultado que se obtiene de multiplicar dicho número por el anterior y así sucesivamente hasta llegar a uno. Ej. el factorial de 4 es $4 * 3 * 2 * 1$ es decir 24.**

```
public class Recursividad {
    int factorial(int fact) {
        if (fact > 0) {
            int valor = fact * factorial(fact - 1);
            return valor;
        } else
            return 1;
    }

    public static void main(String[] ar) {
        Recursividad re = new Recursividad();
        int f = re.factorial(4);
        System.out.println("El factorial de 4 es " + f);
    }
}
```

```
// Calcula recursivamente el factorial
de un número entero positivo y muestra
el resultado.
```

- 6. Ejercicio: Implementar un método recursivo para ordenar los elementos de un vector**


```
class Recursividad {
    static int[] vec = { 312, 614, 88, 22, 54 };

    void ordenar(int[] v, int cant) {
        if (cant > 1) {
            for (int f = 0; f < cant - 1; f++)
                if (v[f] > v[f + 1]) {
                    int aux = v[f];
                    v[f] = v[f + 1];
                    v[f + 1] = aux;
                }
            ordenar(v, cant - 1);
        }
    }

    void imprimir() {
        for (int f = 0; f < vec.length; f++)
            System.out.print(vec[f] + " ");
        System.out.println("\n");
    }

    public static void main(String[] ar) {
        Recursividad r = new Recursividad();
        r.imprimir();
        r.ordenar(vec, vec.length);
        r.imprimir();
    }
}
```

```
/*
- Implementa el algoritmo burbuja
recursivo para ordenar el arreglo vec.
- En cada llamada, "burbujea" el mayor
elemento hasta el final y luego reduce
el rango.
- Cuando el rango (cant) es 1 o menos,
la recursión termina.
- imprimir() muestra el arreglo antes
y después del ordenamiento.
*/
```

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 4</p>

b. PROBLEMAS PROPUESTOS

1. Invertir vector de enteros, permite ingresar tamaño y captura de valores del arreglo, el método invertirArray calcula y muestra el resultado.

N = 3

A = [1 2 3] -> Asalida=[3 2 1]

```
import java.util.*;
public class EP1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int tamaño;


        System.out.println("Ingrese el tamaño del vector :");
        tamaño = scanner.nextInt();
        int[] vector_int = new int[tamaño];

        System.out.println("Ingrese los valores que tendrá el vector");
        for (int i = 0; i < tamaño; i++) {
            System.out.println("Ingrese valor " + (i + 1) + ":");
            vector_int[i] = scanner.nextInt();
        }

        String array = "[";
        for (int i = 0; i < tamaño; i++) {
            array += vector_int[i];
            if (i != tamaño - 1) array += ", ";
        }
        array += "]";

        int[] vector_int_inv = invertirArray(vector_int);

        String arrayInv = "[";
        for (int i = 0; i < tamaño; i++) {
            arrayInv += vector_int_inv[i];
            if (i != tamaño - 1) arrayInv += ", ";
        }
        arrayInv += "]";
    }
}
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 5</p>

```

        System.out.println("A = " + array + " -> Asalida = " + arrayInv);

        scanner.close();
    }

    public static int[] invertirArray(int[] A) {
        int[] Asalida = new int[A.length];
        for (int i = 0; i < A.length; i++) {
            Asalida[i] = A[A.length - 1 - i];
        }
        return Asalida;
    }
}

```

Resultado del Programa:

Ingrese el tamaño del vector :

4

Ingrese los valores que tendrá el vector

Ingrese valor 1:

1

Ingrese valor 2:

2

Ingrese valor 3:

3

Ingrese valor 4:

4

A = [1, 2, 3, 4] -> Asalida = [4, 3, 2, 1]

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 2>

Resumen del Programa:

Este programa lee un vector de enteros ingresado por el usuario, crea una copia invertida de ese vector y muestra ambos vectores en formato de lista.

2. Rotar a la Izquierda, permite ingresar tamaño y captura de valores del arreglo, el método `rotarIzquierdaArray` calcula y muestra el resultado.

Si $d=2$

$A=[1\ 2\ 3\ 4\ 5] \rightarrow A_{invertido}=[3\ 4\ 5\ 1\ 2]$

```
import java.util.*;
public class EP2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("De qué tamaño es el vector?");
        int tamaño = scanner.nextInt();

        int[] vector_int = new int[tamaño];
        System.out.println("Ingrese los valores del vector:");
        for (int i = 0; i < tamaño; i++) {
            System.out.print("Valor " + (i+1) + ": ");
            vector_int[i] = scanner.nextInt();
        }

        System.out.println("¿Cuántos valores desea rotar a la izquierda?");
        int muestra = scanner.nextInt();

        scanner.close();

        int[] vector_rotado = rotarIzquierdaArray(vector_int, muestra);

        System.out.print("Vector original: [");
        for (int i = 0; i < tamaño; i++) {
            System.out.print(vector_int[i]);
            if (i != tamaño - 1) System.out.print(", ");
        }
        System.out.println("]");

        System.out.print("Vector rotado: [");
        for (int i = 0; i < tamaño; i++) {
            System.out.print(vector_rotado[i]);
            if (i != tamaño - 1) System.out.print(", ");
        }
        System.out.println("]");
    }
}
```

```
}  
  
public static int[] rotarIzquierdaArray(int[] A, int d) {  
    int n = A.length;  
    int[] Ainvertido = new int[n];  
    for (int i = 0; i < n; i++) {  
        Ainvertido[i] = A[(i + d) % n];  
    }  
    return Ainvertido;  
}  
}
```

Resultado del Programa:

De qué tamaño es el vector?

5

Ingrese los valores del vector:

Valor 1: 1

Valor 2: 2

Valor 3: 3

Valor 4: 4

Valor 5: 5

¿Cuántos valores desea rotar a la izquierda?

3

Vector original: [1, 2, 3, 4, 5]

Vector rotado: [4, 5, 1, 2, 3]

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 2>

Resumen del Programa:

Este programa recibe un vector de enteros y un número d, luego rota el vector d posiciones a la izquierda y muestra tanto el vector original como el vector rotado.



3. Triángulo recursivo 1. El método trianguloRecursivo1 calcula y muestra el resultado.

- Si b = 5

- Salida:

*

**

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 8</p>

```
public class EP3 {
    public static void main(String[] args) {
        int b = 10;
        trianguloRecursivo1(b);
    }

    public static void trianguloRecursivo1(int b) {
        trianguloHelper(1, b);
    }

    public static void trianguloHelper(int n, int b) {
        if (n > b) return;



        for (int i = 0; i < n; i++) {
            System.out.print("*");
        }
        System.out.println();

        trianguloHelper(n + 1, b);
    }
}
```

Resultado del Programa

*

**

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 9</p>

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 2>

Resumen del Programa:

Este programa imprime un triángulo de asteriscos recursivamente con líneas de longitud creciente desde 1 hasta **b** (10 en este caso), mostrando una forma triangular alineada a la izquierda.

4. Triangulo recursivo 2. El método trianguloRecursivo2 calcula y muestra el resultado.

- Si **b = 5**
- Salida:


```
*
**
***
****
*****}
```

```
public class EP4 {
    public static void main(String[] args) {
        int b = 10;
        trianguloRecursivo2(b);
    }

    public static void trianguloRecursivo2(int b) {
        trianguloHelper(1, b);
    }

    private static void trianguloHelper(int n, int b) {
        if (n > b) return;
        for (int i = 0; i < b - n; i++) {
            System.out.print(" ");
        }
        for (int i = 0; i < n; i++) {
            System.out.print("*");
        }
        System.out.println();

        trianguloHelper(n + 1, b);
    }
}
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 10</p>

}

Resultado del Programa:

```

*
**
***
****
*****
*****
*****
*****
*****
*****
*****

```

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 2>

Resumen del Programa:

Este programa imprime recursivamente un triángulo de asteriscos alineado a la derecha con líneas que van de 1 a **b** (10), agregando espacios a la izquierda para formar la forma inclinada.

5. Triangulo recursivo 3. El método `trianguloRecursivo3` calcula y muestra el resultado.

- Si $b = 5$
- Salida:

```

*
**
***
****
*****

```

```

public class EP5 {
    public static void main(String[] args) {
        int b = 10;
        piramide2D(1, b);
    }
    public static void piramide2D(int n, int b) {
        if (n > b) return;

        for (int i = 0; i < b - n; i++) {

```

```
public class EP6 {
```

```
public static void main(String[] args) {  
    int b = 10;  
    cuadradoRecursivo(1, b);  
}  
  
public static void cuadradoRecursivo(int n, int b) {  
    if (n > b) return;  
  
    if (n == 1 || n == b) {  
        for (int i = 0; i < b; i++) {  
            System.out.print("*");  
        }  
    } else {  
        System.out.print("*");  
        for (int i = 0; i < b - 2; i++) {  
            System.out.print(" ");  
        }  
        System.out.print("*");  
    }  
    System.out.println();  
  
    cuadradoRecursivo(n + 1, b);  
}  
}
```

Resultado del Programa:

```
*      *  
*      *  
*      *  
*      *  
*      *  
*      *  
*      *  
*      *  
*      *
```

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 2>

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 13</p>

Resumen del Programa:

Este programa imprime recursivamente un cuadrado hueco de asteriscos de tamaño $b \times b$, mostrando líneas completas de asteriscos en la primera y última fila, y líneas con asteriscos sólo en los bordes en las filas intermedias.

II. SOLUCIÓN DEL CUESTIONARIO

1. Dificultades al desarrollar ejercicios

- Al desarrollar los ejercicios, tuve dificultades entendiendo la sintaxis y el control de la recursión, lo que causaba errores de pila. También lo que retrasó el desarrollo. Además, noté que algunos algoritmos recursivos, como el ordenamiento burbuja, eran ineficientes para grandes datos y necesitaban optimización.

2. Diferencias entre algoritmos de secuencialidad, decisión e iteración

- **Secuencialidad:** Ejecuta instrucciones en orden, sin cambios.
- **Decisión:** Ejecuta bloques según condición (if, switch).
- **Iteración:** Repite código mientras se cumpla una condición (for, while).

3. Clases y métodos genéricos

- **Clases genéricas:** Plantillas que funcionan con cualquier tipo de dato.
- **Métodos genéricos:** Métodos que usan tipos variables para mayor reutilización.

III. CONCLUSIONES

La recursión requiere un manejo cuidadoso de casos base para evitar errores y asegurar eficiencia. Además, una correcta configuración del entorno y optimización algorítmica son clave para un desarrollo exitoso.

RETROALIMENTACIÓN GENERAL

REFERENCIAS Y BIBLIOGRAFÍA

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 14</p>