

	UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 1

INFORME DE LABORATORIO

(formato estudiante)

INFORMACIÓN BÁSICA					
ASIGNATURA:	ESTRUCTURA DE DATOS Y ALGORITMOS				
TÍTULO DE LA PRÁCTICA:	<i>FUNDAMENTOS DE PROGRAMACIÓN</i>				
NÚMERO DE PRÁCTICA:	<i>01</i>	AÑO LECTIVO:	2025 – A	NRO. SEMESTRE:	Tercero III
FECHA DE PRESENTACIÓN	<i>17/05/2025</i>	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>a. Problemas Desarrollados</h3> <p>1. Arreglos Unidimensionales:</p> <p>Enunciado: Escribe un programa en Java que permita ingresar las edades de un grupo de personas y determine la edad promedio, la mayor y la menor. El usuario debe poder especificar cuántas edades desea ingresar.</p> <p>Datos de Entrada: Número de edades/personas y Listado de Edades. Datos de Salida: Promedio de edades, mayor edad, menor edad.</p> <pre style="background-color: #2e3436; color: #eeeeec; padding: 10px;">import java.util.Scanner; public class ED1 { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese el número de personas: ");</pre>

```
int n = scanner.nextInt();
int[] edades = new int[n];
System.out.println("Ingrese las edades:");
for (int i = 0; i < n; i++) {
    edades[i] = scanner.nextInt();
}
int suma = 0, mayor = edades[0], menor = edades[0];
for (int edad : edades) {
    suma += edad;
    if (edad > mayor)
        mayor = edad;
    if (edad < menor)
        menor = edad;
}
double promedio = (double) suma / n;
System.out.println("Edad promedio: " + promedio);
System.out.println("Edad mayor: " + mayor);
System.out.println("Edad menor: " + menor);
scanner.close();
}
```

Resumen del Programa:

El programa solicita al usuario el número de personas y sus edades, almacenándolas en un arreglo. Luego calcula la suma para obtener el promedio, y determina la edad mayor y menor del grupo. Finalmente, muestra el promedio, la edad máxima y la mínima.

2. Iteraciones:

Enunciado: Escribe un programa en Java que permita calcular la suma de los primeros N números naturales usando un bucle while. El usuario debe ingresar el valor de N.

Datos de Entrada: cantidad de Número naturales

Datos de Salida: suma de números naturales.

```
import java.util.Scanner;
```

	<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>

```
public class ED2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese un número N: ");
        int n = scanner.nextInt();
        int suma = 0, contador = 1;
        while (contador <= n) {
            suma += contador;
            contador++;
        }
        System.out.println("La suma de los primeros " + n + "
números naturales es: " + suma);
        scanner.close();
    }
}
```

Resumen del Programa:

El programa solicita un número N al usuario y calcula la suma de los primeros N números naturales usando un ciclo `while`. Incrementa un contador desde 1 hasta N acumulando la suma. Finalmente, muestra el resultado de la suma calculada.

3. Invariantes:

Enunciado: Implementa un algoritmo que determine si una lista de números ingresados por el usuario está ordenada de manera ascendente. Debes usar un concepto de invariante dentro del bucle para garantizar que la propiedad de orden se mantiene durante la ejecución.

Datos de Entrada: lista de números con orden ascendente.

Datos de Salida: True o False.

```
import java.util.Scanner;
public class ED3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el número de elementos: ");
        int n = scanner.nextInt();
        int[] numeros = new int[n];
```

```
System.out.println("Ingrese los números:");
for (int i = 0; i < n; i++) {
    numeros[i] = scanner.nextInt();
}
boolean estaOrdenada = true;
for (int i = 1; i < n; i++) {
    if (numeros[i] < numeros[i - 1]) {
        estaOrdenada = false;
        break;
    }
}
System.out.println("¿Está ordenada la lista?: " +
(estaOrdenada ? "Sí" : "No"));
scanner.close();
}
```

Resumen del Programa:

El programa recibe un arreglo de números enteros ingresados por el usuario. Luego verifica si la lista está ordenada en forma no decreciente recorriéndola secuencialmente. Finalmente, imprime si la lista está ordenada o no.

b. Problemas Propuestos

1. Desarrolla un programa en Java que implemente un sistema de gestión de calificaciones de estudiantes. El programa debe permitir al usuario ingresar las calificaciones de N estudiantes y calcular la mediana, moda y desviación estándar.

```
import java.util.*;
public class EP1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingrese el número de estudiantes: ");
        int n = scanner.nextInt();
        double[] notas = new double[n];
```

```
System.out.println("Ingrese las calificaciones:");
for (int i = 0; i < n; i++) {
    notas[i] = scanner.nextDouble();
}

Arrays.sort(notas);
double mediana = calcularMediana(notas);
double moda = calcularModa(notas);
double desviacion = calcularDesviacionEstandar(notas);

System.out.printf("Mediana: %.2f\n", mediana);
System.out.printf("Moda: %.2f\n", moda);
System.out.printf("Desviación estándar: %.2f\n",
desviacion);

    scanner.close();
}
// Calcula la mediana de un arreglo ordenado
public static double calcularMediana(double[] arr) {
    int n = arr.length;
    if (n % 2 == 0) {
        return (arr[n / 2 - 1] + arr[n / 2]) / 2.0;
    } else {
        return arr[n / 2];
    }
}
// Calcula la moda del arreglo
public static double calcularModa(double[] arr) {
    Map<Double, Integer> frecuencia = new HashMap<>();
    for (double num : arr) {
        frecuencia.put(num, frecuencia.getOrDefault(num, 0) +
1);
    }
    double moda = arr[0];
    int maxFreq = 0;
    for (Map.Entry<Double, Integer> entry :
frecuencia.entrySet()) {
        if (entry.getValue() > maxFreq) {
```

```
        maxFreq = entry.getValue();
        moda = entry.getKey();
    }
}
return moda;
}
// Calcula la desviación estándar del arreglo
public static double calcularDesviacionEstandar(double[] arr) {
    int n = arr.length;
    double suma = 0.0;
    for (double num : arr) {
        suma += num;
    }
    double media = suma / n;
    double sumaCuadrados = 0.0;
    for (double num : arr) {
        sumaCuadrados += Math.pow(num - media, 2);
    }
    return Math.sqrt(sumaCuadrados / n);
}
}
```

Resultado del Programa:

Ingrese el número de estudiantes: 5

Ingrese las calificaciones:

3

4

5

20

18

Mediana: 5.00

Moda: 4.00

Desviación estándar: 7.40

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 1>

Resumen del Programa:

El programa solicita las calificaciones de **N** estudiantes y almacena los datos en un arreglo. Luego calcula la mediana ordenando el arreglo, determina la moda mediante conteo de frecuencias y calcula la desviación estándar. Finalmente, imprime estos valores estadísticos para el análisis del grupo.

2. Implementa un programa en Java que encuentre todos los números primos en un rango definido por el usuario utilizando el algoritmo de la Criba de Eratóstenes.

```
import java.util.Scanner;

public class EP2 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese el límite superior del rango para encontrar primos: ");

        int n = scanner.nextInt();

        boolean[] esPrimo = cribaEratostenes(n);

        System.out.println("Números primos hasta " + n + ":");

        for (int i = 2; i <= n; i++) {

            if (esPrimo[i]) {

                System.out.print(i + " ");

            }

        }

        System.out.println();

        scanner.close();

    }

    public static boolean[] cribaEratostenes(int n) {

        boolean[] primo = new boolean[n + 1];

        for (int i = 2; i <= n; i++) {

            primo[i] = true;

        }

        for (int p = 2; p * p <= n; p++) {

            if (primo[p]) {

                for (int multiple = p * p; multiple <= n; multiple += p) {
```

```
        primo[multiple] = false;
    }
}
return primo;
}
```

Resultado del Programa:

Ingresa el límite superior del rango para encontrar primos: 10

Números primos hasta 10:

2 3 5 7

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 1>

Resumen del Programa:

El programa solicita al usuario un límite superior para definir el rango de búsqueda de números primos. Utiliza la Criba de Eratóstenes para marcar números no primos y determinar los primos hasta ese límite. Finalmente, imprime todos los números primos encontrados en el rango.

3. Desarrolla un algoritmo que implemente el Ordenamiento por Inserción, asegurando que en cada paso del bucle el segmento procesado de la lista permanece ordenado (principio de invariante).

```
import java.util.Scanner;
public class EP3 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Ingresa el número de elementos a ordenar:");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Ingresa los elementos:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }
    }
}
```



```
        insercion(arr);

        System.out.println("Lista ordenada:");
        for (int num : arr) {
            System.out.print(num + " ");
        }
        System.out.println();

        scanner.close();
    }

    public static void insercion(int[] arr) {
        for (int i = 1; i < arr.length; i++) {
            int key = arr[i];
            int j = i - 1;
            // Mover elementos mayores que key una posición a la
derecha
            while (j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j--;
            }
            // Insertar key en la posición correcta
            arr[j + 1] = key;
            // Invariante: El subarreglo arr[0..i] está ordenado
        }
    }
}
```

Resultado del Programa:

Ingresa el número de elementos a ordenar: 8

Ingresa los elementos:

5

4

3

7

4

	<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>

2

3

1

Lista ordenada:

1 2 3 3 4 4 5 7

PS C:\Users\Usuario\Desktop\EDA\Laboratorio\Sesión 1>

Resumen del Programa:

El programa recibe una lista de números ingresados por el usuario. Implementa el algoritmo de ordenamiento por inserción que mantiene ordenado el segmento procesado en cada iteración. Finalmente, muestra la lista ordenada.

II. SOLUCIÓN DEL CUESTIONARIO

1. ¿Cuáles fueron las dificultades que encontraste al desarrollar los ejercicios propuestos? por ejemplo, poca documentación, complejidad del lenguaje, etc.



Aunque conozco los conceptos, me tomó tiempo optimizar y estructurar bien los algoritmos para que fueran eficientes y funcionaran correctamente. La mayor dificultad fue balancear la lógica con el rendimiento en cada ejercicio.

III. CONCLUSIONES

La práctica constante es clave para mejorar la eficiencia y comprensión de los algoritmos. Optimizar el código garantiza soluciones más rápidas y robustas en la programación.

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: 11</p>