Actividades Vectores + Manejo de Excepciones

Ejercicio 1 - Validar edad ingresada

Consigna:
Solicitar al usuario su edad. Verificar que sea un número entero mayor a 0. Capturar errores si se ingresa texto o valores inválidos.

Resultado esperado:

```
Ingrese su edad: 25
Edad válida: 25

Ingrese su edad: -5
La edad debe ser mayor a cero.

Ingrese su edad: hola
Error: debe ingresar un número entero.
```

Conceptos aplicados:
`Scanner`, `try-catch`, validación de entrada

Solución:

```
import java.util.Scanner;
public class Ej1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Ingrese su edad: ");
            int edad = Integer.parseInt(sc.nextLine());
            if (edad > 0) {
                System.out.println("Edad válida: " + edad);
            } else {
                System.out.println("La edad debe ser mayor a
cero.");
            }
        } catch (Exception e) {
            System.out.println("Error: debe ingresar un número
entero.");
        }
```

```
    }
}
```

Ejercicio 2 - Registro de temperatura semanal

Consigna:
Leer 7 temperaturas (una por día). Guardarlas en un vector. Mostrar el promedio, la máxima y la mínima.

Resultado esperado:

```
Temperatura día 1: 20
...
Temperatura día 7: 22
Promedio: 21.14
Máxima: 23.5
Mínima: 19.0
```

Conceptos aplicados:
Vectores, recorridos, acumuladores, comparaciones

Solución:

```
import java.util.Scanner;
public class Ej2 {
    public static void main(String[] args) {
        double[] temp = new double[7];
        Scanner sc = new Scanner(System.in);
        double suma = 0, max = Double.MIN_VALUE, min =
Double.MAX_VALUE;

        for (int i = 0; i < 7; i++) {
            System.out.print("Temperatura día " + (i + 1) + ": ");
            temp[i] = sc.nextDouble();
            suma += temp[i];
            if (temp[i] > max) max = temp[i];
            if (temp[i] < min) min = temp[i];
        }

        System.out.println("Promedio: " + (suma / 7));
        System.out.println("Máxima: " + max);
        System.out.println("Mínima: " + min);
    }
```

}

Ejercicio 3 - Agenda telefónica simple

Consigna:
Ingresar hasta 5 nombres en una agenda. Mostrar los datos. Indicar si hay espacios libres.

Resultado esperado:

```
Contacto 1: Ana
Contacto 2:
Contacto 3: Luis
Contacto 4:
Contacto 5: Marta
Agenda:
1. Ana
Espacio libre en posición 1
3. Luis
Espacio libre en posición 3
5. Marta
```

Conceptos aplicados:
Vectores de `String`, validación con `isBlank()`

Solución:

```
import java.util.Scanner;
public class Ej3 {
    public static void main(String[] args) {
        String[] agenda = new String[5];
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < agenda.length; i++) {
            System.out.print("Contacto " + (i + 1) + ": ");
            agenda[i] = sc.nextLine();
        }

        System.out.println("Agenda:");
        for (int i = 0; i < agenda.length; i++) {
            if (agenda[i].isBlank()) {
                System.out.println("Espacio libre en posición " +
i);
            } else {
```

```
                System.out.println((i + 1) + ". " + agenda[i]);
            }
        }
    }
}
```

Ejercicio 4 - Encuesta de satisfacción

Consigna:
Registrar en una matriz las respuestas (1 a 5) de 3 clientes a 4 preguntas. Calcular el promedio de cada pregunta.

Resultado esperado:

```
Respuestas:
4 3 5 2
5 4 4 3
3 2 5 5
Promedio Pregunta 1: 4.0
Promedio Pregunta 2: 3.0
Promedio Pregunta 3: 4.67
Promedio Pregunta 4: 3.33
```

Conceptos aplicados:
Matrices, bucles anidados, promedio por columnas

Solución:

```
import java.util.Scanner;
public class Ej4 {
    public static void main(String[] args) {
        int[][] respuestas = new int[3][4];
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 4; j++) {
                System.out.print("Cliente " + (i + 1) + " - Pregunta
" + (j + 1) + ": ");
                respuestas[i][j] = sc.nextInt();
            }
        }

        for (int j = 0; j < 4; j++) {
```

```
            int suma = 0;
            for (int i = 0; i < 3; i++) {
                suma += respuestas[i][j];
            }
            System.out.println("Promedio Pregunta " + (j + 1) + ": "
+ (double) suma / 3);
        }
    }
}
```

Ejercicio 5 - Cálculo de facturación

Consigna:
Cargar hasta 10 importes de ventas. Mostrar el total. Alertar si alguna venta supera $100.000.

Resultado esperado:

```
Importe venta 1: 150000
⚠️ Venta mayor a $100.000 detectada
...
Total facturado: $347000.0
```

Conceptos aplicados:
Vectores, acumuladores, condicionales

Solución:

```
import java.util.Scanner;
public class Ej5 {
    public static void main(String[] args) {
        double[] ventas = new double[10];
        Scanner sc = new Scanner(System.in);
        double total = 0;

        for (int i = 0; i < ventas.length; i++) {
            System.out.print("Importe venta " + (i + 1) + ": ");
            ventas[i] = sc.nextDouble();
            total += ventas[i];
            if (ventas[i] > 100000) {
                System.out.println("⚠️ Venta mayor a $100.000
detectada");
```

```
            }
        }

        System.out.println("Total facturado: $" + total);
    }
}
```

Ejercicio 6 - Control de stock

Consigna:
Simular una matriz 3x3 con cantidades por producto y sucursal. Mostrar totales por sucursal.

Resultado esperado:

```
Sucursal 1 total: 250
Sucursal 2 total: 190
Sucursal 3 total: 310
```

Conceptos aplicados:
Matrices, acumuladores por fila

Solución:

```
import java.util.Scanner;
public class Ej6 {
    public static void main(String[] args) {
        int[][] stock = new int[3][3];
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < 3; i++) {
            System.out.println("Sucursal " + (i + 1));
            for (int j = 0; j < 3; j++) {
                System.out.print("Producto " + (j + 1) + ": ");
                stock[i][j] = sc.nextInt();
            }
        }

        for (int i = 0; i < 3; i++) {
            int total = 0;
            for (int j = 0; j < 3; j++) {
```

```
                total += stock[i][j];
            }
            System.out.println("Total Sucursal " + (i + 1) + ": " +
total);
        }
    }
}
```

Ejercicio 7 - Validación de acceso por índice

Consigna:
Mostrar el valor de un vector por índice. Si es incorrecto, capturar la excepción.

Resultado esperado:

```
Ingrese índice (0-4): 6
Error: índice fuera de rango.
```

Conceptos aplicados:
Vectores, `ArrayIndexOutOfBoundsException`

Solución:

```
import java.util.Scanner;
public class Ej7 {
    public static void main(String[] args) {
        String[] codigos = {"A123", "B234", "C345", "D456", "E567"};
        Scanner sc = new Scanner(System.in);

        try {
            System.out.print("Ingrese índice (0-4): ");
            int i = sc.nextInt();
            System.out.println("Código: " + codigos[i]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: índice fuera de rango.");
        } catch (Exception e) {
            System.out.println("Entrada inválida.");
        }
    }
}
```

Ejercicio 8 - Leer empleados desde archivo CSV

Consigna:
Leer un archivo `empleados.csv` con formato `nombre,edad,sueldo`. Mostrar los datos y sueldo promedio.

Resultado esperado:

```
Juan - Edad: 28 - Sueldo: $95000.0
Ana - Edad: 35 - Sueldo: $105000.0
Promedio sueldos: $100000.0
```

Conceptos aplicados:
Archivos, `split()`, casting, promedio

Solución:

```
import java.io.File;
import java.util.Scanner;

public class Ej8 {
    public static void main(String[] args) {
        double totalSueldos = 0;
        int cantidad = 0;

        try {
            Scanner sc = new Scanner(new File("empleados.csv"));
            while (sc.hasNextLine()) {
                String[] datos = sc.nextLine().split(",");
                String nombre = datos[0];
                int edad = Integer.parseInt(datos[1]);
                double sueldo = Double.parseDouble(datos[2]);
                System.out.println(nombre + " - Edad: " + edad + " -
Sueldo: $" + sueldo);
                totalSueldos += sueldo;
                cantidad++;
            }
            System.out.println("Promedio sueldos: $" + (totalSueldos
/ cantidad));
        } catch (Exception e) {
            System.out.println("Error al leer archivo.");
        }
```

```
    }
}
```

Ejercicio 9 - Guardar respuestas en CSV

Consigna:
Pedir nombre y calificación (1-5) de 3 personas. Guardar en `respuestas.csv`. Luego mostrar el contenido.

Resultado esperado:

```
Nombre: Laura - Calificación: 5
Nombre: Marcos - Calificación: 4
Nombre: Sofía - Calificación: 5
```

Conceptos aplicados:
Archivos, `PrintWriter`, `split()`, `Scanner`

Solución:

```
import java.io.*;
import java.util.Scanner;

public class Ej9 {
    public static void main(String[] args) {
        try (PrintWriter pw = new PrintWriter("respuestas.csv");
             Scanner sc = new Scanner(System.in)) {
            for (int i = 0; i < 3; i++) {
                System.out.print("Nombre: ");
                String nombre = sc.nextLine();
                System.out.print("Calificación (1-5): ");
                int calificacion = Integer.parseInt(sc.nextLine());
                pw.println(nombre + "," + calificacion);
            }
        } catch (IOException e) {
            System.out.println("Error al escribir archivo.");
        }

        try (Scanner lector = new Scanner(new
File("respuestas.csv"))) {
            while (lector.hasNextLine()) {
                String[] datos = lector.nextLine().split(",");
```

```
                System.out.println("Nombre: " + datos[0] + " -
Calificación: " + datos[1]);
            }
        } catch (Exception e) {
            System.out.println("Error al leer archivo.");
        }
    }
}
```

🟥 Ejercicio 10 - Desafío: Analizador de stock

Consigna:
Simular un sistema de stock usando una matriz de 5 productos (filas) y 4 depósitos (columnas). El programa debe:

1. Permitir ingresar la cantidad de cada producto en cada depósito.
2. Calcular y mostrar el **stock total por producto** (suma de cada fila).
3. Calcular y mostrar el **stock total por depósito** (suma de cada columna).
4. Identificar y mostrar qué producto tiene **mayor stock acumulado**.
5. Validar todas las entradas numéricas con `try-catch`.

Resultado esperado:

```
Ingrese stock para Producto 1 en Depósito 1: 50
...
Total por producto:
Producto 1: 180
Producto 2: 150
...
Total por depósito:
Depósito 1: 220
Depósito 2: 190
...
Producto con mayor stock: Producto 1 (180 unidades)
```

Conceptos aplicados:

- Matrices
- Recorrido por filas y columnas
- Acumuladores
- Validación con excepciones (`try-catch`)

Ejercicio 11 - Desafío: Editor de archivo CSV desde consola

Consigna:
Implementar un pequeño sistema interactivo por consola que permita operar con un archivo `personas.csv` con formato:
`nombre,edad,correo`

El programa debe incluir un **menú con opciones**:

1. Mostrar todo el contenido del archivo
2. Agregar una nueva persona
3. Buscar una persona por nombre
4. Salir

Resultado esperado:

```
1. Ver archivo
2. Agregar persona
3. Buscar persona
4. Salir
>> 1
Juan,30,juan@mail.com
Ana,25,ana@mail.com

>> 2
Nombre: Laura
Edad: 28
Correo: laura@mail.com
Persona guardada.

>> 3
```

```
Buscar nombre: Ana
Encontrado: Ana,25,ana@mail.com
```

📚 **Conceptos aplicados:**

- Archivos CSV (`Scanner`, `PrintWriter`)
- `split()`
- Búsqueda por clave
- Menú con bucles y control de flujo
- Validaciones y excepciones