

Actividades Java y a Maven

Objetivos

- Usar la clase `Scanner` para lectura desde teclado y archivos.
 - Procesar archivos de texto plano separados por coma.
 - Comprender y aplicar arrays en Java.
 - Usar el método `split()` para separar cadenas.
 - Aplicar conversiones de tipos (casting) básicas.
-

1. Clase `Scanner`

Enunciado:

Se desea escribir un programa en Java que pida al usuario ingresar su nombre y su edad utilizando el teclado, y luego muestre un saludo personalizado con esa información.

Crear un programa que:

1. Solicite al usuario su nombre (como texto).
2. Solicite su edad (como número entero).
3. Muestre un mensaje de saludo en pantalla con los datos ingresados.

Salida esperada:

Ingrese su nombre: Laura

Ingrese su edad: 22

Hola Laura, tenés 22 años.

Conceptos:

- Usar la clase `Scanner` para leer los datos desde `System.in`.
- Para leer el nombre, utilizar `nextLine()`.
- Para la edad, usá `nextInt()`.
- No olvides cerrar el Scanner al final.

Solución

```
import java.util.Scanner;

public class EntradaTeclado {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese su nombre: ");
        String nombre = sc.nextLine();

        System.out.print("Ingrese su edad: ");
        int edad = sc.nextInt();

        System.out.println("Hola " + nombre + ", tenés " + edad + " años.");
        sc.close();
    }
}
```

2. Introducción a Arrays y método `split()`

Enunciado:

Se tiene una cadena de texto con datos de una persona, separados por comas: nombre, edad y altura.

Escribir un programa que:

1. Divida la cadena usando la coma como separador.
2. Guarde cada parte en una variable del tipo correspondiente.

3. Muestre en consola los datos con etiquetas, uno por línea.

Salida esperada:

Nombre: Juan
Edad: 25
Altura: 1.75

Conceptos:

- Arrays: estructuras para almacenar múltiples valores del mismo tipo.
- `split()`: método de `String` para dividir texto en partes.
- Casting con `Integer.parseInt()`, `Double.parseDouble()`.
- Almacenamiento en `String[]`.

Solución:

```
public class ArraysSplit {  
    public static void main(String[] args) {  
        String datos = "Juan,25,1.75";  
        String[] partes = datos.split(",");  
  
        String nombre = partes[0];  
        int edad = Integer.parseInt(partes[1]);  
        double altura = Double.parseDouble(partes[2]);  
  
        System.out.println("Nombre: " + nombre);  
        System.out.println("Edad: " + edad);  
        System.out.println("Altura: " + altura);  
    }  
}
```

3. Lectura de archivos CSV

Enunciado:

Escribir un programa que:

1. Lea el archivo `datos.csv` línea por línea.
2. Separe cada línea utilizando la coma como separador.
3. Convierta los datos al tipo correspondiente (`String`, `int`, `double`).
4. Muestre en consola un mensaje con el formato:

`[Nombre] tiene [Edad] años y mide [Altura] m.`

Ejemplo de archivo `datos.csv`:

```
Ana,23,1.60
Luis,30,1.80
María,28,1.65
```

Solución:

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

public class LeerArchivoCSV {
    public static void main(String[] args) {
        try {
            File archivo = new File("datos.csv");
            Scanner lector = new Scanner(archivo);

            while (lector.hasNextLine()) {
                String linea = lector.nextLine();
                String[] datos = linea.split(",");
                String nombre = datos[0];
                int edad = Integer.parseInt(datos[1]);
                double altura = Double.parseDouble(datos[2]);
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}
```

```

        System.out.println(nombre + " tiene " + edad + "
años y mide " + altura + " m.");
    }
    lector.close();
} catch (FileNotFoundException e) {
    System.out.println("Archivo no encontrado.");
}
}
}

```

4. Registro y consulta de personas

Enunciado:

Desarrollar un programa que permita al usuario:

1. Cargar nuevos datos de personas desde el teclado (nombre, edad y altura).
2. Guardar los datos ingresados en un archivo `personas.csv`.
3. Leer ese archivo línea por línea.
4. Procesar cada línea separándola con `split(",")`.
5. Mostrar un listado completo con el siguiente formato:

Nombre: Ana - Edad: 23 - Altura: 1.60 m

6. Luego, mostrar cuántas personas tienen más de 25 años y calcular la altura promedio.

Conceptos:

- `Scanner` para entrada por teclado y lectura de archivos.
- Arrays (`String[]`) con `split()`.
- `Integer.parseInt()` y `Double.parseDouble()` para casting.

- Archivos con `File` y `PrintWriter`.
-

Solución:

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

public class PersonasApp {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        File archivo = new File("personas.csv");

        // 1. Ingreso de datos por teclado
        try (PrintWriter writer = new PrintWriter(new
FileWriter(archivo, true))) {
            System.out.print("¿Cuántas personas desea ingresar?: ");
            int cantidad = sc.nextInt();
            sc.nextLine(); // Limpia el buffer

            for (int i = 0; i < cantidad; i++) {
                System.out.print("Nombre: ");
                String nombre = sc.nextLine();

                System.out.print("Edad: ");
                int edad = sc.nextInt();

                System.out.print("Altura: ");
                double altura = sc.nextDouble();
                sc.nextLine(); // Limpia el buffer

                writer.println(nombre + "," + edad + "," + altura);
            }
        } catch (IOException e) {
            System.out.println("Error al escribir el archivo.");
        }

        // 2. Lectura y procesamiento del archivo
        int mayores25 = 0;
        double sumaAlturas = 0;
```

```

int total = 0;

try (Scanner lector = new Scanner(archivo)) {
    System.out.println("\nListado de personas:");

    while (lector.hasNextLine()) {
        String linea = lector.nextLine();
        String[] datos = linea.split(",");

        String nombre = datos[0];
        int edad = Integer.parseInt(datos[1]);
        double altura = Double.parseDouble(datos[2]);

        System.out.println("Nombre: " + nombre + " - Edad: "
+ edad + " - Altura: " + altura + " m");

        if (edad > 25) mayores25++;
        sumaAlturas += altura;
        total++;
    }

    if (total > 0) {
        System.out.println("\nPersonas mayores de 25 años: "
+ mayores25);
        System.out.printf("Altura promedio: %.2f m\n",
sumaAlturas / total);
    }
} catch (FileNotFoundException e) {
    System.out.println("Archivo no encontrado.");
}

sc.close();
}
}

```