

CSV

Los archivos **CSV (Comma-Separated Values)** son un formato popular para almacenar datos en forma de tabla. En Java, se pueden leer y escribir archivos CSV utilizando clases como `FileReader`, `BufferedReader`, `FileWriter`, y bibliotecas externas como **OpenCSV**.

1. Escribir un Archivo CSV (sin librerías externas)

Podemos usar `FileWriter` y `PrintWriter` para escribir datos en un archivo CSV.

```
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class EscribirCSV {
    public static void main(String[] args) {
        String archivo = "datos.csv";
        try (PrintWriter escritor = new PrintWriter(new
FileWriter(archivo))) {
            escritor.println("ID,Nombre,Edad");
            escritor.println("1,Ana,25");
            escritor.println("2,Juan,30");
            escritor.println("3,Pedro,22");
            System.out.println("Archivo CSV creado correctamente.");
        } catch (IOException e) {
            System.out.println("Error al escribir el archivo CSV.");
            e.printStackTrace();
        }
    }
}
```

2. Leer un Archivo CSV (sin librerías externas)

Usamos `BufferedReader` para leer el contenido línea por línea y dividir los datos con `split(",")`.

```
import java.io.BufferedReader;
import java.io.FileReader; import java.io.IOException;

public class LeerCSV {
```

```

    public static void main(String[] args) {
        String archivo = "datos.csv";
        try (BufferedReader lector = new BufferedReader(new
FileReader(archivo))) {
            String linea;
            while ((linea = lector.readLine()) != null) {
                String[] datos = linea.split(",");
                System.out.println("ID: " + datos[0] + ", Nombre: " +
datos[1] + ", Edad: " + datos[2]);
            }
        } catch (IOException e) {
            System.out.println("Error al leer el archivo CSV.");
        }
    }
}

```

3. Manejo de CSV con OpenCSV

Para trabajar con CSV de manera más avanzada, podemos usar la biblioteca **OpenCSV**.

Instalación de OpenCSV (Maven)

Si usas **Maven**, agrega la siguiente dependencia en tu `pom.xml` :

```

<dependency>
    <groupId>com.opencsv</groupId>
    <version>5.7.1</version>
</dependency>

```

4. Escribir CSV con OpenCSV

```

import com.opencsv.CSVWriter;
import java.io.FileWriter;
import java.io.IOException;

public class EscribirCSVOpenCSV {
    public static void main(String[] args) {
        String archivo = "datos_opencsv.csv";
        try (CSVWriter escritor = new CSVWriter(new FileWriter(archivo))) {
            String[] encabezado = {"ID", "Nombre", "Edad"};
            String[] fila1 = {"1", "Ana", "25"};

```

```

        String[] fila2 = {"2", "Juan", "30"};
        escritor.writeNext(encabezado);
        escritor.writeNext(fila1);
        escritor.writeNext(fila2);
        System.out.println("Archivo CSV creado con OpenCSV.");
    } catch (IOException e) {
        System.out.println("Error al escribir el archivo CSV.");
    }
}
}

```

5. Leer CSV con OpenCSV

```

import com.opencsv.CSVReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.List;

public class LeerCSVOpenCSV {
    public static void main(String[] args) {
        String archivo = "datos_opencsv.csv";
        try (CSVReader lector = new CSVReader(new FileReader(archivo))) {
            List<String[]> filas = lector.readAll();
            for (String[] fila : filas) {
                System.out.println("ID: " + fila[0] + ", Nombre: " + fila[1]
+ ", Edad: " + fila[2]);
            }
        } catch (IOException e) {
            System.out.println("Error al leer el archivo CSV.");
            e.printStackTrace();
        }
    }
}

```