

# Proyecto Analizador de Datos Abiertos

## Objetivo

Desarrollar una aplicación Java que permita el parseo de datos en formatos CSV, JSON y XML, utilizando conjuntos de datos reales del **portal datos.gob.es**.



## Descripción

Crear una aplicación en Java utilizando IntelliJ que sea capaz de leer y analizar archivos en diferentes formatos. La aplicación permitirá al usuario seleccionar un archivo, parsearlo y mostrar información relevante sobre los datos contenidos.

## Funcionalidades

1. Selección y lectura de archivos:
  - Permitir al usuario especificar la ruta de un archivo local (CSV, JSON o XML).
  - Detectar automáticamente el formato del archivo.
2. Parseo de datos:
  - Implementar métodos de parseo específicos para cada formato: CSV: Utilizar
3. Análisis básico de datos:
  - Contar el número total de registros.
  - Identificar y listar los campos o atributos presentes.
  - Para datos numéricos, calcular mínimo, máximo y promedio.
4. Visualización de resultados:
  - Mostrar un resumen de la información parseada.
  - Presentar los primeros 5 registros como muestra .

## Estructura del proyecto

```
import com.google.gson.Gson;
import com.google.gson.JsonObject;
import org.w3c.dom.Document;
import org.w3c.dom.Element;
import org.w3c.dom.NodeList;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class AnalizadorDatosAbiertos {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Introduce la ruta del archivo a analizar:");
        String rutaArchivo = scanner.nextLine();

        if (rutaArchivo.endsWith(".csv")) {
            List<String[]> datosCSV = parsearCSV(rutaArchivo);
            mostrarResumenCSV(datosCSV);
        } else if (rutaArchivo.endsWith(".json")) {
            JsonObject datosJSON = parsearJSON(rutaArchivo);
            mostrarResumenJSON(datosJSON);
        } else if (rutaArchivo.endsWith(".xml")) {
            Document datosXML = parsearXML(rutaArchivo);
            mostrarResumenXML(datosXML);
        } else {
            System.out.println("Formato de archivo no soportado.");
        }
    }

    public static List<String[]> parsearCSV(String rutaArchivo) {
        List<String[]> registros = new ArrayList<>();
        try (BufferedReader br = new BufferedReader(new
FileReader(rutaArchivo))) {
            String linea;
            while ((linea = br.readLine()) != null) {
                String[] valores = linea.split(",");
                registros.add(valores);
            }
        }
    }
}
```

```

        } catch (Exception e) {
            System.out.println("Error al leer el archivo CSV: " +
e.getMessage());
        }
        return registros;
    }

    public static JsonObject parsearJSON(String rutaArchivo) {
        JsonObject jsonObject = null;
        try (FileReader reader = new FileReader(rutaArchivo)) {
            jsonObject = new Gson().fromJson(reader, JsonObject.class);
        } catch (Exception e) {
            System.out.println("Error al leer el archivo JSON: " +
e.getMessage());
        }
        return jsonObject;
    }

    public static Document parsearXML(String rutaArchivo) {
        Document doc = null;
        try {
            DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
            DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();
            doc = dBuilder.parse(rutaArchivo);
            doc.getDocumentElement().normalize();
        } catch (Exception e) {
            System.out.println("Error al leer el archivo XML: " +
e.getMessage());
        }
        return doc;
    }

    public static void mostrarResumenCSV(List<String[]> datos) {
        if (datos.isEmpty()) {
            System.out.println("No se encontraron datos.");
            return;
        }

        System.out.println("Resumen del archivo CSV:");
        System.out.println("Número total de filas: " + datos.size());
        System.out.println("Número de columnas: " + datos.get(0).length);

        System.out.println("\nPrimeros 5 registros:");
        for (int i = 0; i < Math.min(5, datos.size()); i++) {
            System.out.println(String.join(" | ", datos.get(i)));
        }
    }

```

```
    }  
}  
  
public static void mostrarResumenJSON(JsonObject datos) {  
    // Implementar lógica para mostrar resumen de datos JSON  
}  
  
public static void mostrarResumenXML(Document datos) {  
    // Implementar lógica para mostrar resumen de datos XML  
}  
}
```

## Instrucciones para el desarrollo

1. Crear un nuevo proyecto Java en IntelliJ.
2. Copiar el código proporcionado en la clase principal.
3. Añadir las dependencias necesarias (Gson para JSON, si se usa).
4. Descargar archivos de ejemplo en CSV, JSON y XML del portal [datos.gob.es](https://datos.gob.es).
5. Implementar los métodos `mostrarResumenJSON` y `mostrarResumenXML`.
6. Ejecutar el programa y probar con diferentes archivos.

## Evaluación

El proyecto se evaluará considerando:

- Correcta implementación del parseo para cada formato.
- Manejo adecuado de excepciones.
- Calidad y claridad del código.
- Funcionalidad de resumen y presentación de datos.
- Uso correcto de las bibliotecas y métodos de parseo .