Memoria del Proyecto: Acceso a Datos - Ficheros JSON
Nombre del Estudiante: Dos Santos Chabauz, Wendel Edgar
Fecha de Presentación: 17 de Diciembre de 2023

Uso y funcionalidad de este proyecto:

El proyecto se centra en el acceso y procesamiento de datos meteorológicos proporcionados por la API de Meteogalicia.

Formatos Utilizados:

Se utilizan dos formatos clave:

- **JSON (JavaScript Object Notation):** Un formato de intercambio de datos ligero y fácilmente legible que organiza la información en pares clave-valor. Su estructura jerárquica lo hace ideal para representar datos complejos.

- **CSV (Comma-Separated Values):** Un formato simple de almacenamiento de datos en forma de tabla. Cada línea del archivo representa una fila y los valores se separan por comas. Es eficiente para almacenar grandes conjuntos de datos tabulares.

Librerías de Java Utilizadas:

Durante el desarrollo, se han empleado dos librerías fundamentales:

- **Gson:** Una librería de Google que simplifica la conversión entre objetos Java y representaciones JSON. Se utiliza para parsear los datos JSON obtenidos de la API.

- **OpenCSV:** Utilizada para la escritura de datos en formato CSV. Facilita la creación y manipulación de archivos CSV en Java.

- Se han agregado las correspondientes dependencias en el fichero build.gradle:

```
// Gson para parserar el JSON
implementation 'com.google.code.gson:gson:2.8.8'

// Para gestionar los CSV
implementation 'com.opencsv:opencsv:5.6'
```

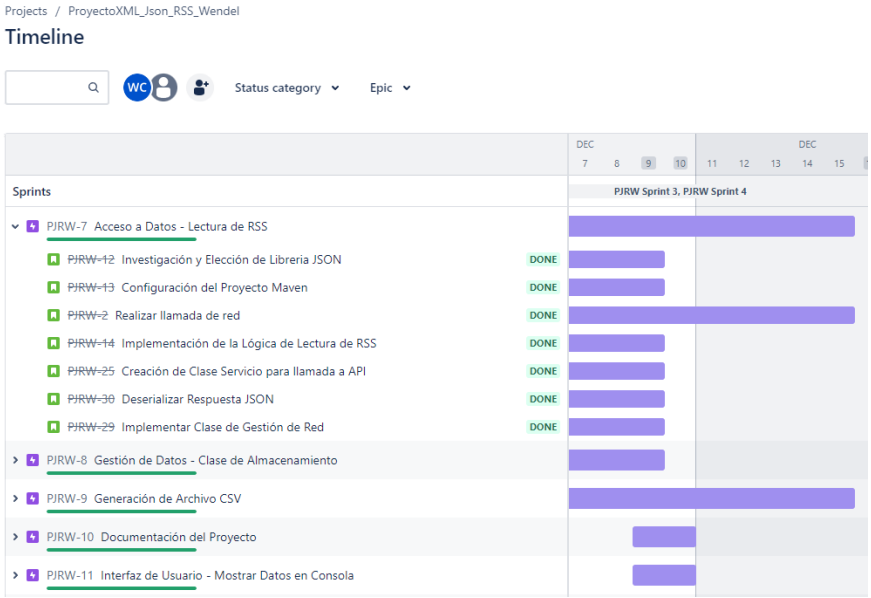
Por qué he elegido éstas librerías?

Gson: Gson, librería de Google para Java, simplifica la conversión entre objetos Java y JSON con su sintaxis clara y eficiente. Diseñada específicamente para Java, ofrece facilidad de uso, integración coherente y rendimiento optimizado al manejar grandes volúmenes de datos JSON.

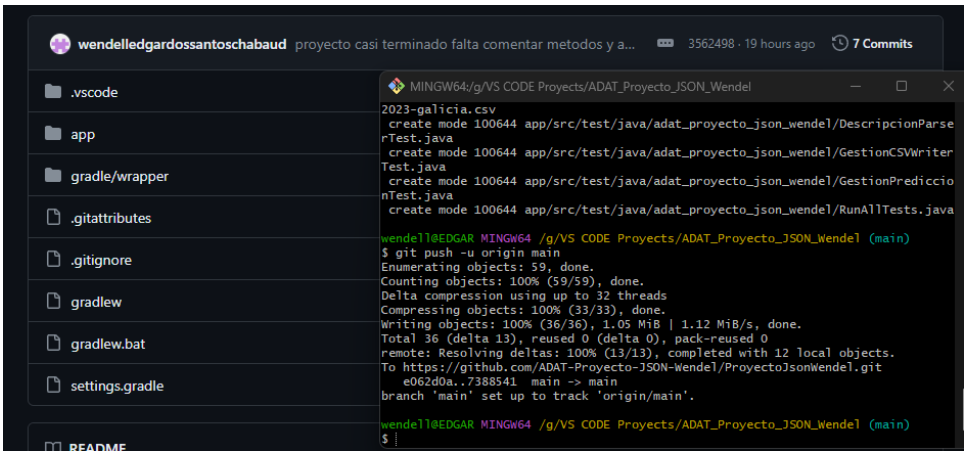
OpenCSV: OpenCSV es una elección acertada para la escritura de datos en formato CSV en Java, proporcionando simplicidad en la manipulación de archivos CSV y herramientas para tratar las peculiaridades del formato. Su integración fluida con Java y capacidad para manejar datos tabulares complejos hacen que sea ideal para proyectos que requieren el procesamiento de información en formato CSV.

Proceso de Desarrollo:

- Se ha empleado la plataforma Jira para su organización en diferentes tareas:



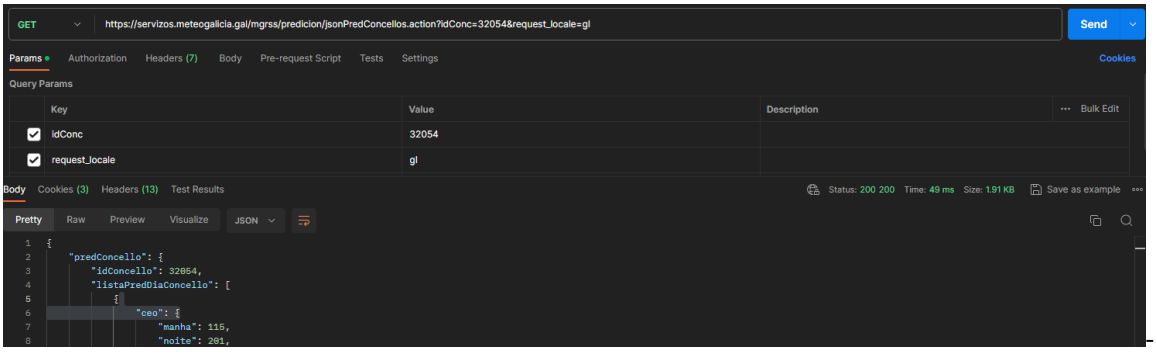
- Para el desarrollo y correcta gestión de un control de veriones he utilizado GIT y GITHUB:



Petición y Parseo del JSON:

Se realiza una petición a la API de Meteogalicia utilizando la URL correspondiente.
Esta API proporciona datos de predicción meteorológica a corto plazo para un concello (municipio) específico en Galicia, España. A través de la URL proporcionada, puedes realizar consultas para obtener información de tallada sobre el tiempo, incluyendo datos como estado del cielo, temperatura, viento, precipitación, etc.

- Ejemplo de petición a la API con POSTMAN:



Podemos observar que nos devuelve un JSON y una HTTP Response con el codigo 200 (OK).

El resultado de la petición, en formato JSON, se parsea con Gson, creando objetos Java que representan la es tructura del JSON.

- Realizamos una petición a una url y lo parseamos utilizando GSON.

```
PrediccionConcello prediccion = gestion.obtenerPrediccion(urlCompleta);
```

```
/**
 * Obtiene la predicción de un concello a partir de una URL.
 *
 * @param url La URL que proporciona la información de la predicción.
 * @return Un objeto PrediccionConcello con la información de la predicción o null si hay un error.
 */
public PrediccionConcello obtenerPrediccion(String url) {
    try {
        // Verificar si la URL es válida
        if (!isValidURL(url)) {
            System.err.println(x:"URL no válida.");
            return null;
        }
        try (InputStreamReader reader = new InputStreamReader(new URL(url).openStream())) {
            Gson gson = new Gson();
            PrediccionWrapper wrapper = gson.fromJson(reader, classOfT:PrediccionWrapper.class);
            return wrapper.getPredConcello();
        }
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }
}
```

Visualización de Datos:

Se implementa la visualización de datos clave por consola, incluyendo diferentes datos que nos ha proporcio nado Meteogalicia.

```
ID do Concello: 15078
Nome do Concello: Santiago de Compostela
Data Predicion: 2023-12-17T00:00:00
Nivel de Aviso: 0
Estado do Ceo pola manha: 101
Estado do Ceo pola tarde: 101
Estado do Ceo pola noite: 201
Probabilidade de Choiva (manha): 5%
Probabilidade de Choiva (tarde): 5%
Probabilidade de Choiva (noite): 5%
Temperatura Maxima: 14Â°C
Temperatura Minima: 5Â°C
Indice Ultravioleta Maximo: 1
Vento (manha): Dato descoÃ±ecido
Vento (tarde): Calma
Vento (noite): Vento frouxo do Norte (N)
-----
```

Clase para Leer Datos del JSON:

- Se crean diferentes clases dedicadas para leer los datos del JSON, implementando métodos de getter y setter para las propiedades más relevantes.

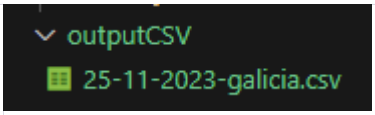
  Ejemplo de una de las clases Model para leer datos del JSON.

```java
public class PrediccionConcello {

    // Identificador único del concello
    private int idConcello;

    // Lista de predicciones diarias para el concello puede haber 3 o 4
    private List<DiaPrediccion> listaPredDiaConcello;

    // Nombre del concello
    private String nome;

    public PrediccionConcello() {
        this.listaPredDiaConcello = new ArrayList<>();
    }

    // Getter y Setter
```

Generación de Fichero .CSV:

- Se genera un archivo CSV que contiene los datos de las 7 ciudades importantes de Galicia. El nombre del archivo es "25-11-2023-galicia.csv".

```
-----
Datos escritos correctamente en el archivo: app\src\main\java\adat_proyecto_json_wendel\outputCSV\25-11-2023-galicia.csv
```

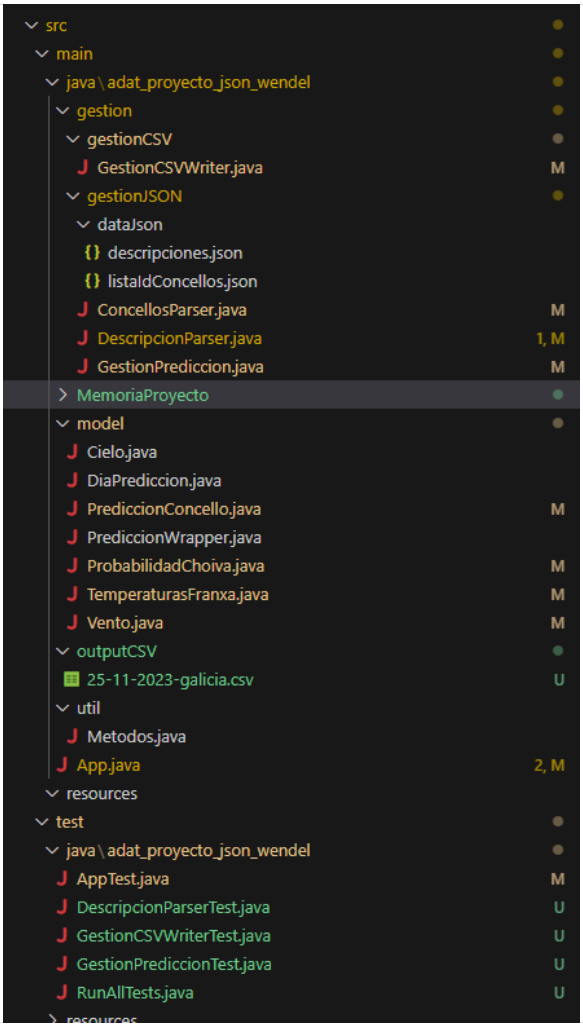



```
"concello","fecha","tMax","tMin","uvMax","estadoCeoManana","direccionVientoManana","intensidadVientoManana"
"Santiago de Compostela","2023-12-17T00:00:00","14","5","1","Despexado","Vento frouxo do Norte (N)","Vento frouxo do Norte (N)"
"Santiago de Compostela","2023-12-18T00:00:00","13","1","1","Despexado","Vento frouxo do Norte (N)","Vento variable"
"Santiago de Compostela","2023-12-19T00:00:00","11","1","1","Choiva débil","Vento frouxo do Sur (S)","Vento frouxo do Norte (N)"
"Santiago de Compostela","2023-12-20T00:00:00","13","6","1","Nubes e claros","Vento moderado do Nordés (NE)","Calma"
"Vigo","2023-12-17T00:00:00","15","5","1","Despexado","Calma","Vento frouxo do Sueste (SE)"
"Vigo","2023-12-18T00:00:00","15","5","1","Despexado","Calma","Vento frouxo do Sueste (SE)"
"Vigo","2023-12-19T00:00:00","14","5","1","Choiva débil","Vento frouxo do Sueste (SE)","Calma"
"Vigo","2023-12-20T00:00:00","16","10","1","Nubes e claros","Vento variable","Vento frouxo do Sueste (SE)"
"Ferrol","2023-12-17T00:00:00","15","2","1","Despexado","Calma","Calma"
"Ferrol","2023-12-18T00:00:00","14","1","1","Nubes e claros","Calma","Calma"
"Ferrol","2023-12-19T00:00:00","13","4","1","Choiva débil","Vento frouxo do Sur (S)","Vento frouxo do Sur (S)"
"Ferrol","2023-12-20T00:00:00","13","9","1","Anubrado 75%","Vento frouxo do Nordés (NE)","Vento variable"
"Lugo","2023-12-17T00:00:00","10","0","1","Despexado","Calma","Calma"
"Lugo","2023-12-18T00:00:00","9","0","1","Despexado","Calma","Calma"
"Lugo","2023-12-19T00:00:00","7","0","1","Orballo","Vento frouxo do Sueste (SE)","Calma"
"Lugo","2023-12-20T00:00:00","9","6","1","Nubes e claros","Vento frouxo do Nordés (NE)","Vento frouxo do Sueste (SE)"
"A Coruña","2023-12-17T00:00:00","16","5","1","Despexado","Vento frouxo do Sur (S)","Vento frouxo do Sur (S)"
"A Coruña","2023-12-18T00:00:00","15","5","1","Nubes e claros","Calma","Calma"
"A Coruña","2023-12-19T00:00:00","14","6","1","Choiva débil","Vento frouxo do Sueste (SE)","Vento frouxo do Sur (S)"
"A Coruña","2023-12-20T00:00:00","14","10","1","Nubes e claros","Vento frouxo do Nordés (NE)","Calma"
"Ourense","2023-12-17T00:00:00","9","2","1","Despexado","Calma","Calma"
"Ourense","2023-12-18T00:00:00","8","3","1","Despexado","Calma","Calma"
"Ourense","2023-12-19T00:00:00","12","4","1","Anubrado 75%","Calma","Calma"
"Ourense","2023-12-20T00:00:00","12","5","1","Nubes e claros","Vento frouxo do Sudoeste (SO)","Calma"
"Pontevedra","2023-12-17T00:00:00","16","1","1","Despexado","Calma","Calma"
"Pontevedra","2023-12-18T00:00:00","16","2","1","Despexado","Calma","Calma"
"Pontevedra","2023-12-19T00:00:00","15","2","1","Choiva débil","Calma","Calma"
"Pontevedra","2023-12-20T00:00:00","16","9","1","Nubes e claros","Vento frouxo do Norte (N)","Calma"
```

Documentación del Código:

ex - El código está organizado en paquetes para una fácil comprensión. Cada clase y método está documentado, plicando su propósito y funcionamiento. Cada método tiene su JavaDoc.

- Organización del código en diferentes paquetes:



- Ejemplo de documentación de un método.

```java
/**
 * Verifica si una URL es válida.
 *
 * @param url La URL a verificar.
 * @return true si la URL es válida, false en caso contrario.
 */
private boolean isValidURL(String url) {
    try {
        new URL(url).toURI();
        return true;
    } catch (MalformedURLException | java.net.URISyntaxException e) {
        return false;
    }
}
```

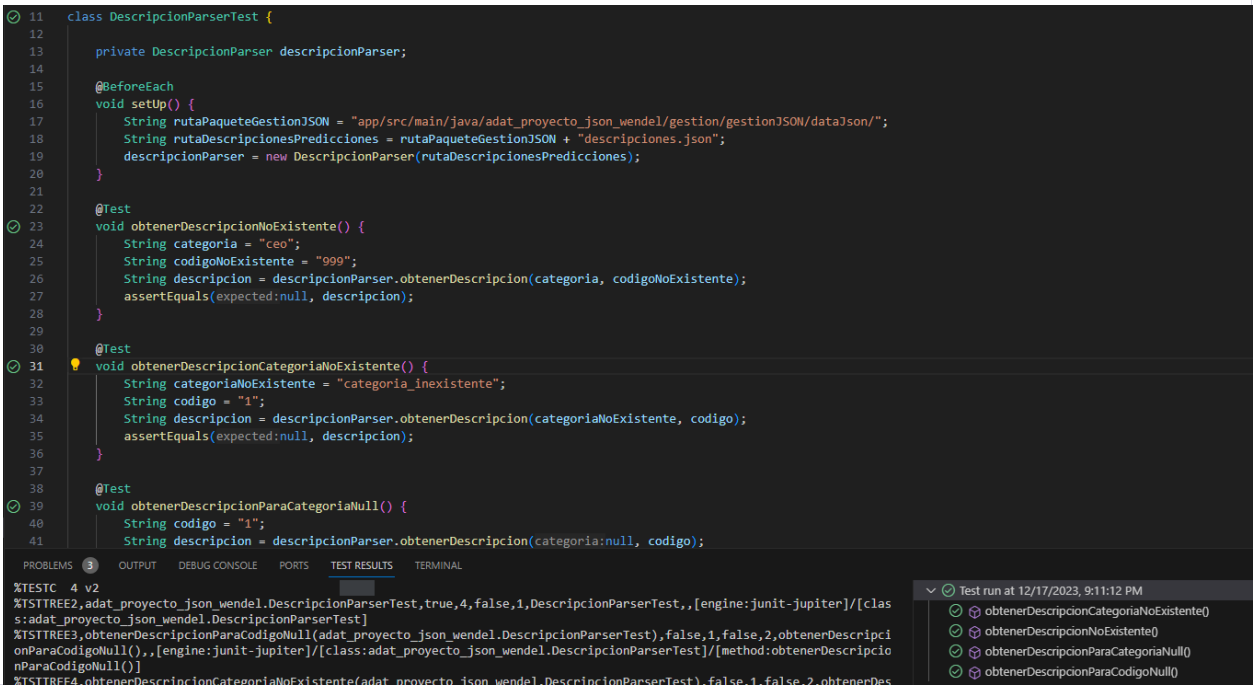
Ejecución del Proyecto y Pruebas:

Pasos para Ejecutar el Proyecto:

Descomprimir Proyecto.
Abrir ruta del Proyecto con VS Code.
Imprescindible tener instalados los addons: Java Extension Pack, Debugger for Java,
Gradle Languaje Support, Java Test Runner, así como el SDK de Java
Ejecutar el proyecto desde el Fichero App.java, también se pueden ejecutar los ficheros de pruebas Unitarias.

Pruebas Realizadas:

- Se realizaron pruebas para verificar la correcta obtención y parseo de los datos JSON.
- Se verificó la generación del archivo CSV con datos precisos y correctamente formateados.
- Se verificaron métodos con pruebas unitarias utilizando JUNIT.



Conclusiones:

El proyecto proporciona una aplicación funcional y documentada para el acceso y procesamiento de datos meteorológicos. La combinación de JSON y CSV permite una representación eficiente y una fácil manipulación de los datos.
El uso de las librerías Gson y OpenCSV simplifica el manejo de datos, garantizando una implementación robusta y eficiente. La aplicación se puede ampliar y mejorar según las necesidades futuras.