

Practica Acceso Datos 2

Daniel Castellote y Alejandro López

Buenas te explicare en principio como hemos planteado el proyecto y la resolución a la que hemos llegado:

Primeramente hemos creado clases POJO de temperatura y contaminantes donde guardaremos futuramente los datos cargados de los XML. En ellas están indicados todos los atributos que queremos sacar del CSV dado.

Posteriormente creamos un Lector CSV el cual abrirá un buffer de lectura que leerá todas las líneas de los CSV(reemplazando espacios vacíos por 0 para que no salte error de valor nulo) e ira creando objetos medición y dentro de medición creando elementos que serán cargados por los datos que vamos leyendo. Gracias a ello tendremos nuestros XML creados perfectamente.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <mediciones>
3   <medicion>
4     <provincia>28</provincia>
5     <municipio>102</municipio>
6     <estacion>1</estacion>
7     <magnitud>81</magnitud>
8     <puntoMuestreo>28102001_81_89</puntoMuestreo>
9     <anio>2021</anio>
10    <mes>9</mes>
11    <dia>1</dia>
12    <h1>76,0</h1>
13    <v1>V</v1>
14    <h2>86,0</h2>
15    <v2>V</v2>
16    <h3>71,0</h3>
17    <v3>V</v3>
18    <h4>6,0</h4>
19    <v4>V</v4>
20    <h5>78,0</h5>
21    <v5>V</v5>
22    <h6>63,0</h6>
23    <v6>V</v6>
24    <h7>27,0</h7>
25    <v7>V</v7>
26    <h8>20,0</h8>
```

El siguiente paso será volver a leer los archivos XML y posteriormente cargaremos los datos en los atributos de las clases POJO y crearemos una lista con la que podremos filtrar a partir de ahí.

```
public List<Temperatura> getTemperatura() {
    // Tomamos el elemento raiz y obtenemos sus hijos
    Element root = (Element) this.data.getRootElement();
    List<Element> listaMediciones = root.getChildren( cname: "medicion");

    List<Temperatura> listatemperaturas = new ArrayList<>();

    // Por cada elemento
    listaMediciones.forEach(elemento -> {

        Temperatura temp = new Temperatura();
        temp.setProvincia(Integer.parseInt(elemento.getChildText( cname: "provincia")));
        temp.setMunicipio(Integer.parseInt(elemento.getChildText( cname: "municipio")));
        temp.setEstacion(Integer.parseInt(elemento.getChildText( cname: "estacion")));
        temp.setMagnitud(Integer.parseInt(elemento.getChildText( cname: "magnitud")));
        temp.setPuntoMuestreo(elemento.getChildText( cname: "puntoMuestreo"));
        temp.setAño(Integer.parseInt(elemento.getChildText( cname: "anio")));
        temp.setMes(Integer.parseInt(elemento.getChildText( cname: "mes")));
        temp.setDia(Integer.parseInt(elemento.getChildText( cname: "dia")));
        temp.setH1(Integer.parseInt(elemento.getChildText( cname: "h1")));
        temp.setV1(elemento.getChildText( cname: "v1"));
        temp.setH2(Integer.parseInt(elemento.getChildText( cname: "h2")));
        temp.setV2(elemento.getChildText( cname: "v2"));
        temp.setH3(Integer.parseInt(elemento.getChildText( cname: "h3")));
        temp.setV3(elemento.getChildText( cname: "v3"));
        temp.setH4(Integer.parseInt(elemento.getChildText( cname: "h4")));
        temp.setV4(elemento.getChildText( cname: "v4"));
        temp.setH5(Integer.parseInt(elemento.getChildText( cname: "h5")));
        temp.setV5(elemento.getChildText( cname: "v5"));
        temp.setH6(Integer.parseInt(elemento.getChildText( cname: "h6")));
        temp.setV6(elemento.getChildText( cname: "v6"));
        temp.setH7(Integer.parseInt(elemento.getChildText( cname: "h7")));
        temp.setV7(elemento.getChildText( cname: "v7"));
        temp.setH8(Integer.parseInt(elemento.getChildText( cname: "h8")));
        temp.setV8(elemento.getChildText( cname: "v8"));
        listatemperaturas.add(temp);
    });

    return listatemperaturas;
}
```

A partir de ahí hemos creado una clase nueva llamada “Filtrar” que creara nuevas listas que serán cargadas y a partir de ahí las leemos para comprobar que se imprimen y están bien cargadas, con estas listas podremos filtrar máximos, mínimos y medias.

A partir de aquí no hemos sabido cómo sacar los datos filtrados de una sola medición es decir no podíamos sacar medias del conjunto de valores h de una sola medición, lo mismo ocurre con los datos de validación de hora.

Sin embargo decidimos intentar otras cosas, hemos creado un mapeador para el código de los municipios que indique el nombre de cada uno de los municipios.

Nuestra idea de proyecto fue querer fusionar los XML en uno llamado Medicion.xml que utilizara los atributos comunes de los dos XML y que luego sacara tanto los datos de contaminación como los de temperatura para la misma ciudad.

Para finalizar queríamos hacerte saber que hemos intentado hasta el final poder acabar esta práctica y nos hemos frustrado bastante al ver que no sabíamos seguir, nuestra actitud en esta asignatura es no rendirnos por lo que preferimos mandarte todo lo que podamos y sepamos hacer intentándolo hasta el final porque sabemos que tenemos un ritmo igual algo más lento. Con lo que queremos que veas que por lo menos lo estamos intentando, para conseguir llegar a tu ritmo y poder sacar esta asignatura y el grado adelante.