

Programación de Servicios y Procesos

UD 1 – Concurrency

Práctica 1

Obtención de la temperatura máxima y mínima estaciones de Aragón

El concepto **open data** es una filosofía y práctica que persigue que determinados tipos de datos estén disponibles de forma libre para todo el mundo, sin restricciones de derechos de autor, de patentes o de otros mecanismos de control. La agencia estatal de meteorología (aemet) dispone de una plataforma de open data que permite acceder a un conjunto de datos de manera libre y gratuita. Esta implementación incluye una API que permite a nuestros programas conectar de forma automática y solicitar los datos incluidos en la API.

Disponemos de un conjunto de datos diarios de un conjunto de estaciones meteorológicas distribuidas por el territorio de Aragón desde 1990 hasta la actualidad. El archivo EjemploDatos.txt tiene los datos diarios del observatorio de Teruel el 5 de mayo de 2004, el archivo Metadatos.txt muestra información sobre el contenido de los ficheros de datos diarios.

Se requiere realizar un programa que lea la información de los ficheros de datos (Datos Diarios.zip) y muestre la temperatura más alta registrada por cada mes, en qué fecha se produjo, a qué hora se tomó la medida y en qué estación meteorológica. Se requiere la obtención de los mismos datos para la temperatura mínima alcanzada. Finalmente se mostrarán las temperaturas máxima y mínima absoluta para los datos disponibles.

El programa debe permitir dividir la tarea de leer todos los ficheros en distintos hilos de forma que se puedan paralelizar las tareas de procesamiento de los datos. También se pide que se encuentre el número de hilos óptimo para ejecutar esta tarea en el ordenador del aula.

Ayuda.

- Para obtener el listado de ficheros existente en un directorio podemos utilizar código como el siguiente:

```
Path dir = ...;
try (DirectoryStream<Path> stream = Files.newDirectoryStream(dir))
{
    for (Path path: stream) {
        System.out.println(path.getFileName());
    }
} catch (IOException | DirectoryIteratorException x) {
    // IOException can never be thrown by the iteration.
    // In this snippet, it can only be thrown by
    newDirectoryStream.
    System.err.println(x);
}
```

- Para leer completamente un fichero y asignarlo a una variable de tipo String podemos utilizar código como el que tenemos a continuación:

```
String content = new String(Files.readAllBytes(path));
```

- Para la deserialización de datos json podemos utilizar la librería GSON y para utilizar esta librería en nuestros proyectos resulta conveniente utilizar Maven para gestionar el proyecto. Resultará conveniente definir una clase que se ajuste al contenido de los ficheros json. El código para la deserialización sería algo así:

```
Gson gson = new Gson();
MyClass myClass = gson.fromJson(jsonContent, MyClass.class);
```

- Recuerda que, si varios hilos acceden al mismo objeto, en nuestro caso los datos de temperaturas máximas y mínimas registradas por mes, deberemos implementar exclusión mutua para el acceso y modificación de estos datos por parte de los distintos hilos en ejecución.

Recuerda incluir comentarios al código que realizas.

*Una vez completados los programas incluye el **proyecto maven / java completo** en un archivo **zip** y súbelo a Classroom.*