



PROYECTO

DATALAKE TO DATAMART

DESARROLLO
DE
APLICACIONES
PARA CIENCIA
DE DATOS

Grado de Ingeniería en
Ciencia de Datos

ESCUELA UNIVERSITARIA DE INFORMÁTICA.
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

ÍNDICE

1.Resumen

2.Recursos utilizados

3.Diseño

4.Conclusiones

5.Líneas futuras

6.Bibliografía

RESUMEN

El programa realizado en esta práctica tiene como principal función la extracción de los datos procedentes del webservice de la AEMET cada hora.

Todo esto realizado en el lenguaje de programación java, mediante diferentes clases con funciones en específico. Un módulo llamado feeder, en el cual la función principal es descargar estos datos del webservice.

Estos datos serán almacenados en un directorio llamado datalake.dir, siendo archivos .events, los cuales tendrán de nombre la fecha correspondiente del día en el que se han recogido los datos.

Los archivos serán leídos a través de un FileReader en el siguiente módulo, llamado datamart. Este módulo se encarga de leer los datos de los archivos generados anteriormente e insertarlos en tablas SQL. Estas tablas serán llamadas Max y Min, en las cuales se almacenan los datos de la estación con la temperatura máxima y mínima respectivamente, junto al día, la hora, lugar y estación. Esto lo haremos todo mediante una conexión con SQL a la base de datos. Creando así el archivo datamart.db, dónde se almacena lo dicho anteriormente.

En el último módulo llamado api, nos encargaremos de coger los datos que nos interesan del archivo datamart.db, que en este caso será la temperatura máxima y temperatura mínima que hubo en el rango de días señalados por el usuario. Esto se señalará mediante una consulta a una API Rest que realizaremos en la clase APIRest, encargada de filtrar y recoger las temperaturas máxima y mínima procedentes entre los días seleccionados, mostrándolo al usuario a través de la siguientes peticiones:

```
GET /v1/places/with-max-temperature?from={date}&to={date}
```

```
GET /v1/places/with-min-temperature?from={date}&to={date}
```

RECURSOS USADOS

En cuanto a los entorno de desarrollo utilizados, se ha usado IntelliJ IDEA para la programación en java.

A parte, también se ha hecho uso del navegador web de Google, para poder comprobar los resultados de la API a través de por ejemplo:

<http://localhost:4567/v1/places/with-max-temperature?from=2023-01-10&to=2023-01-12>

También se ha usado la herramienta de GitHub para poder subir el proyecto a un repositorio al que se puede acceder mediante un link y poder visualizar el contenido del proyecto.

DISEÑO

En cuanto a los patrones de diseño realizado, se ha intentado seguir el MVC (Model View Controller), para poder separar los datos de la aplicación, la interfaz de usuario y la lógica de control en tres componentes diferentes.

A su vez se ha intentado separar el proyecto en diferentes módulos para poder realizar una mejor ejecución y distribución del programa.

CONCLUSIONES

Gracias a como se ha realizado el programa, he logrado entender mucho mejor el funcionamiento de una API, ya que gracias a esta práctica se como realizar peticiones a través de un parámetro en específico de la Query y eso está bastante bien objetivamente hablando.

Si tuviera la posibilidad de tener o más tiempo o repetir el trabajo entero, intentaría hacer el uso de más clase POJO o de soporte para generar así un mejor Clean Code.

LÍNEAS FUTURAS

Para hacer evolucionar este producto se podría crear una interfaz gráfica de cara al usuario, para que este pudiera elegir los días en los que desea visualizar las máximas y mínimas temperaturas, a pesar de que la API lo muestra ya bastante bien a través de un Json.

B I B L I O G R A F Í A

<https://opendata.aemet.es/centrodedescargas/inicio>

<https://www.sqlitetutorial.net/sqlite-select/>

<https://github.com/>

JOEL CLEMENTE LÓPEZ CABRERA

DESDE EL DÍA
1/01/2023
HASTA EL DÍA
13/01/2023

DESARROLLO
DE
APLICACIONES
PARA CIENCIA
DE DATOS

Grado de Ingeniería en
Ciencia de Datos