



Agencia Estatal de Meteorología

## Datalake to Datamart

DEVELOPING A DATALAKE

| DESARROLLO DE APLICACIONES PARA CIENCIA DE DATOS |  
| 2º | GCID | ESCUELA DE INGENIRÍA INFORMÁTICA | UPGC |

# Resumen

Este trabajo se centra en el uso de Java para realizar tres funciones. La primera sería descargar cada hora los datos de todas las estaciones meteorológicas de Gran Canaria desde el webservice de AEMET y generar un *datalake* con estos datos. La segunda sería construir un *datamart* a partir de los datos del *datalake* con las temperaturas máximas y mínimas de cada día. La tercera y última función, sería la creación de una *API Rest* que sea capaz de realizar consultas en base a los datos del *datamart*. Para realizar estas funciones, se ha hecho uso principalmente de *JBDC*, *Gson* y *Spark*.

En cuanto al funcionamiento del programa, este está dividido en tres módulos:

-El módulo “feeder”, contiene las clases: -“AemetDownloader” que se encarga de acceder al *web service* de Aemet y descargar todos los datos de las estaciones meteorológicas de España. -“Filter” que se encarga de filtrar todas las mediciones que no pertenezcan a Gran Canaria, al igual que aquellos datos que no queramos. -“FileWriter” que se encarga de guardar los datos de cada día en nuestro *datalake*. Finalmente, en el “Main” tenemos que introducir nuestra *Api Key* y la dirección donde queremos crear el *datalake*. Además, está controlado por un *TimerTask*, que se encarga de coger los datos cada hora, para añadir nuevas mediciones.

-El módulo “datarmart-builder”, contiene las clases: -“Measurements” que maneja los objetos que contienen las mediciones meteorológicas, además, contiene varias funciones que nos permite obtener objetos *Measurements* con solo mediciones máximas o mínimas. -“SqliteWeatherDatabase” que se encarga de crear y manejar nuestro “*datamart.db*” junto a la clase “*DMLTranslator*”. Además contiene dos interfaces llamadas “*WeatherDatabase*” y “*WeatherSource*”. Finalmente, en el “Main” debes colocar la dirección de tu *datalake* para que recoja los datos y los actualice en el *datamart*. Igualmente, está controlado por un *TimerTask*, que se asegura que todos los días a las 23:55, se suban las medidas de las temperaturas máximas y mínimas del día.

-El módulo “temperature-api” contiene dos clases: -“*SQLite*” que se encarga de sacar los datos de nuestro *datamart* en función a la petición de la API. -“Main” que se encarga de recibir las peticiones GET. Adicionalmente, la carpeta “test”, contiene una clase llamada “*TestApi*” para comprobar el correcto funcionamiento de la API, aunque también se puede hacer uso de *Talend API Tester*.

# Índice

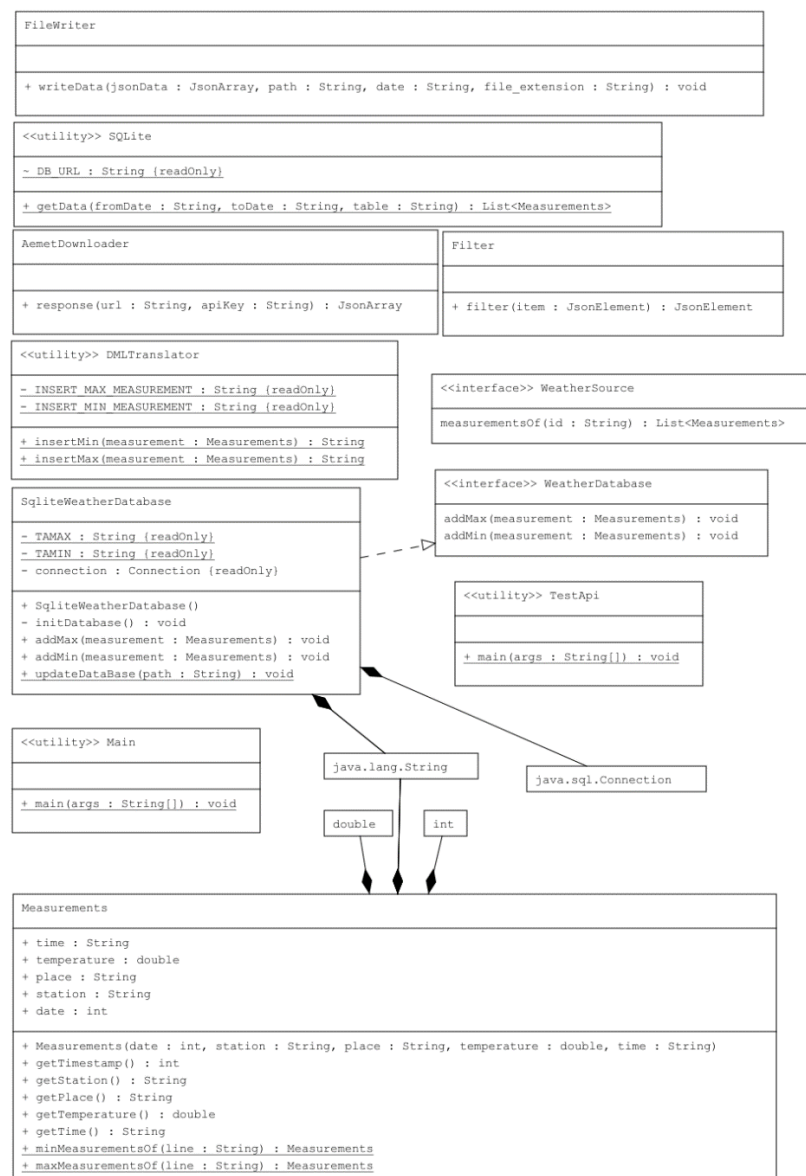
· Resumen.....	1
· Recursos Utilizados.....	3
· Diseño.....	3
· Conclusiones.....	4
· Líneas Futuras.....	4
· Bibliografía.....	5

# Recursos Utilizados

Para la realización de este trabajo se ha usado principalmente el entorno de desarrollo de *IntelliJ IDEA Community Edition 2022.2.1* junto a *Talend API Tester*, mientras que para su documentación se ha usado el software de *Microsoft Word 2019*. Finalmete para subir el programa se ha hecho uso de *GitHub*.

## Diseño

Para este proyecto se ha tenido en cuenta el principio de responsabilidad única por clase, en el que cada clase debe solo ocuparse de una funcionalidad. También empleamos la separación por módulos. Además, se ha hecho uso del principio de inversión de la dependencia al usar interfaces.



# Conclusiones

En este trabajo he aprendido mucho más acerca de la creación de *web services* y *API Rest*. Además de la creación de *datalakes*, el uso de *SQLite* para crear *datamarts* y lo útiles que pueden llegar a ser para almacenar datos concretos sin necesidad de realizar una búsqueda exhaustiva para acceder a ellos, también he aprendido el uso de la clase de java *TimerTask* que nos permite establecer un temporizador a nuestro programa.

# Líneas Futuras

El proyecto creo que podría mejorarse si se crearan un mayor número de *datamarts*, así como de métodos para tener una mayor base datos estructurada de manera específica y acceder a la información más rápido. Para su comercialización se podría crear una interfaz gráfica para el fácil acceso a los datos, todo esto en una página web en la que se haría uso de anuncios, para obtener ganancias, pues no veo justificable pagar solo por obtener los datos que de por sí ya están gratis en la web de Aemet.

# Bibliografía

- <https://www.sqlitetutorial.net/>
- <https://www.javatpoint.com/sql-tutorial>
- <https://stackoverflow.com/>
- <https://www.baeldung.com/java-timer-and-timertask>
- <https://opendata.aemet.es/centrodedescargas/inicio>

Autor: Gabriel Felipe Bernal Pinto

Fecha: 13/01/2023

Version: 1.0