

Escuela de ingeniería informática, Universidad de Las Palmas de Gran  
Canaria

# Proyecto Datalake to Datamart

Desarrollo de Aplicaciones para Ciencia de Datos.

Javier Franco González

Grado en Ciencia e ingeniería de datos, 2º curso.

## Resumen

En este proyecto se han implementado tres módulos. El primero de ellos obtiene datos meteorológicos de Aemet y va guardando estos datos en una carpeta, creando un fichero para cada día. El segundo fichero crea un Datamart mediante una base de datos Sqlite, en la cual se crean dos tablas donde se almacenan las temperaturas mínimas y máximas de cada día, obtenidas desde el Datalake. El último módulo consiste en un programa API REST que realiza consultas de los sitios con mayor y menor temperatura en un rango de días. Estos datos se obtendrían del Datamart.

Respecto al contenido del primer módulo encontramos varias clases. La clase Aemet Accesor cuenta con una única función, que accede a la API de Aemet y devuelve un jsonarray con datos meteorológicos. Para acceder a estos datos se ayuda de la clase TrustAllTrustManager. Por otra parte tenemos la clase Aemet Controller, cuya funcionalidad es filtrar los datos recogidos de Gran Canaria en la última hora de entre todos los datos obtenidos anteriormente. También se ha creado una clase para representar un evento meteorológico. Por último una clase Datalake con un método principal que va creando y actualizando ficheros cada día con la información de dichos eventos. La clase Main invoca a este método cada hora con la ayuda de un TimerTask.

En el segundo módulo encontramos tan sólo dos clases además del Main. La clase Database incorpora métodos para crear una base de datos Sqlite, así como la creación de las tablas y el añadir o eliminar elementos de dichas tablas. También existe la clase Datamart Controller, que posee funciones para obtener el evento que cuenta con la máxima y con la mínima temperatura de los eventos almacenados en el Datalake. Una vez obtenido este evento, también incluye funciones que añaden dicha información a la base de datos. Cada hora, también con la ayuda de un TimerTask, el Main, usando los métodos de dichas clases encuentra la temperatura máxima y mínima de cada día, actualizando con ellas la base de datos.

En el último módulo es incluso más simple que el segundo. La clase Datamart Accesor posee dos funciones que pasadas una fecha, acceden a la base de datos y extraen el lugar donde se registró la máxima/mínima temperatura en esa fecha. Por otro lado, la clase API Controller contiene los métodos que realizan las consultas de los lugares con la mayor o menor temperatura en un rango de días, métodos que se ayudan de los anteriormente mencionados. Estos sitios se almacenarán en una ArrayList. En la clase Main es donde el usuario debe hacer las peticiones para obtener los lugares más fríos o cálidos. Podemos encontrar en el mismo Main un pequeño ejemplo que como funcionaría este programa.

## **Índice:**

1. Recursos utilizados (Pág 3)
2. Diseño (Pág 3)
3. Conclusiones (Pág 3)
4. Líneas futuras (Pág 4)
5. Bibliografía (Pág 4)

## **Recursos utilizados**

El entorno de programación que se ha utilizado es IntelliJ. El SDK del proyecto es Oracle JDK versión 11. También me he documentado mediante numerosas páginas web que pueden encontrarse en la bibliografía de esta memoria.

## **Diseño**

El diseño se ha centrado en presentar un código con nombres de variables que te ayudan a comprenderlo. También intentar que los métodos y las propias clases no sean demasiado extensos. Además separar las distintas funcionalidades del código en varios módulos, y crear clases para repartirse las tareas dentro de cada módulo. Se han creado las instrucciones necesarias para que el Main de cada módulo presente tan solo unas pocas líneas.

## **Conclusiones**

A pesar de que al inicio no sabía plantear esta actividad, he conseguido realizarla aprendiendo muchas cosas que podrán serme útiles en el futuro. Considero que es una actividad que tiene una funcionalidad real y se asemeja a lo que podría ser un trabajo real de un ingeniero de datos. He aprendido más sobre bases de datos, manejo de ficheros, etc. A pesar de esto no estoy seguro de si he sabido implementar una arquitectura concreta. Obtuve más problemas extrayendo los datos de Aemet, sin embargo no tuve mayores complicaciones con el resto del código. Aún sin saber hacer algunas cosas pude buscar información o ejemplos que me ayudaron.

## **Líneas futuras**

El programa resulta útil de por sí para consultar la información del tiempo en Gran Canaria. Aun así podría ampliarse mucho más obteniendo una mayor cantidad de datos y de otros lugares además de Gran Canaria. También se podrían presentar los datos de una manera más vistosa que en un fichero o una base de datos, para que el cliente pudiera entender mejor la información. Se podrían implementar muchas más peticiones en la API. Los datos meteorológicos son de interés general, por lo que una gran cantidad de personas estarían interesados en usar este programa.

## **Bibliografía**

<https://www.sqlitetutorial.net/sqlite-java/sqlite-jdbc-driver/>

<https://www.sqlitetutorial.net/sqlite-create-table/>

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

<https://mvnrepository.com/>

<https://stackoverflow.com/>

<https://opendata.aemet.es/centrodedescargas/AEMETApi?>

<https://chat.openai.com/chat>

<https://picodotdev.github.io/blog-bitix/2021/09/escapar-simbolos-especiales-en-una-expresion-regular-en-java/>

<https://howtodoinjava.com/java/date-time/dates-between-two-dates/>

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

Javier Franco González, 10/1/2023.