

# Developing a datalake

Desarrollo de Aplicaciones



CIENCIA E INGENIERÍA DE DATOS  
ESCUELA DE INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

## Resumen:

El proyecto se divide en 3 directorios, “Feeder”, “Datamart-builder” y “Temperatura-API”, los cuales aunque sean independientes uno de otros, en cierta manera se necesitan ya que para ejecutarlos tenemos que seguir un orden.

El primer directorio que vamos a ejecutar es el “Feeder”, el cual a través de la clase “AemetLoader” va a leer los datos que se encuentran en la API de la Aemet para filtrar y guardar en la clase “Event” los datos recogidos en Gran Canaria. Seguidamente en la clase “FileSystemDatalake” se va a crear una carpeta en la ruta definida en los argumentos y la cual llamaremos como lo hemos definido en estos, en este caso la llamaremos “datalake”, a continuación se crearan ciertos ficheros dentro de esa carpeta que contendrán los datos de la clase “Event” en formato JSON. Además tendrá dos funciones read uno para la leer la hora del último del evento registrado en el día previo y otro para el día actual. Cabe mencionar que a la hora de crear y escribir los eventos en el “datalake” hace falta otra clase llamada “InstantSerialier” la cual se encargara de serializar los time stamps de la clase “Event”. Posteriormente en la clase “Controller” se llamará a la interfaz “Datalake” y a la interfaz “Loader”, las cuales actuaran de puentes entre los métodos de las clases anteriormente mencionadas y la clase “Controller”, añadiéndole además un Timer Task, haciendo que cada hora se vayan incluyendo nuevos datos en los eventos. Por último, en la clase “Main” se llamará a la clase “Controller” para ejecutar todo el programa.

El siguiente directorio que se ejecutará será el “Datamart-builder”, quien tiene la responsabilidad de leer los datos de los eventos anteriormente creados en el “Feeder” y guardarlos en la clase “Event”, este proceso se realizara en el “FileDatalake”. A continuación, en la clase “SqliteWriter” vamos a crear una base de datos la cual tendrá dos tablas una para las temperaturas mínimas y otra para las máximas. Estas tablas contendrán las columnas: “Value”, “Time”, “Date”, “Place” y “Station”; en las que insertaremos los datos correspondientes a las temperaturas máximas y mínimas de los eventos. Hay que mencionar también que tendremos una función la cual borrara todos los elementos de la tabla. Por último, en relación con esta clase, cabe destacar que tiene una clase abstracta llamada “DatabaseWriter”. Por otro lado en la clase “Controller” inicializaremos la interfaz “Datalake”, buscaremos y le pasaremos al “SqliteWriter” los datos de las temperaturas máximas y mínimas, y crearemos un Timer Task el cual al llamar a la clase “Contoller” desde el “Main” nos borrara los datos de las tablas cada hora y nos volverá a escribir los datos actualizados.

Por último, ejecutaremos el directorio “Temperatura-API” en el cual en la clase “SqliteReader” leeremos todos los datos del datamart anteriormente creado, guardando los datos de este en las clases “Event\_Max” y “Event\_Min”, además también inicializaremos los parámetros from y to, los cuales hacen referencia al rango de días, primer día y último día, que se encuentran en el datamart, estos parámetros nos servirán para poder hacer la estructura que deben tener las rutas para pedir estos datos por internet. A continuación en la clase “TemperaturaCommand” mediante los métodos “execute\_max” y “execute\_min” y mediante los parámetros “Request” y “Responde”, que representan la solicitud HTTP y la respuesta HTTP, vamos a obtener una lista de eventos con todos los datos que hemos recogido en el “SqliteReader”, los vamos a pasar a formato JSON y se los vamos a pasar a la clase “SparkWebService” en la cual configuramos el servidor web para escuchar en el puerto 4567 y establecemos los controladores de ruta para las solicitudes HTTP para las rutas establecidas.

#### **ÍNDICE:**

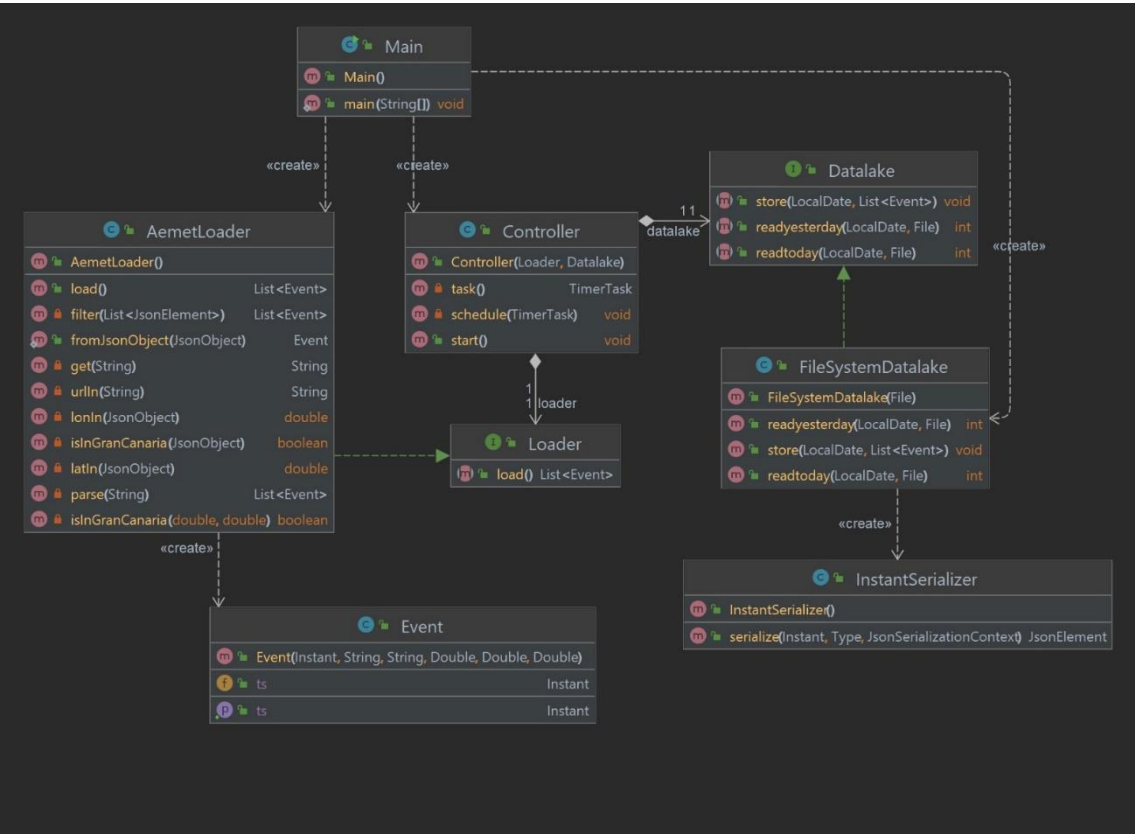
<b>Recursos utilizados.....</b>	<b>2</b>
<b>Diseño.....</b>	<b>3</b>
<b>Conclusiones.....</b>	<b>4</b>
<b>Líneas futuras.....</b>	<b>5</b>
<b>Bibliografía.....</b>	<b>5</b>

#### **Recursos utilizados:**

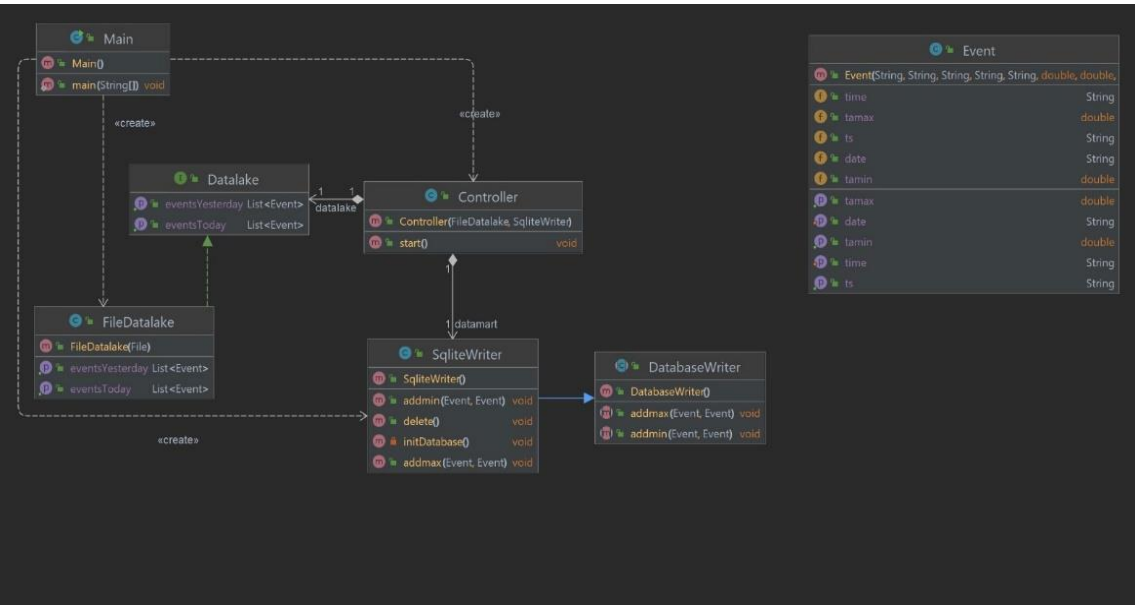
Como entorno de desarrollo he utilizado el programa intellij, como herramienta de control he utilizado Git y por último como herramienta de documentación he utilizado Word.

Diseño:

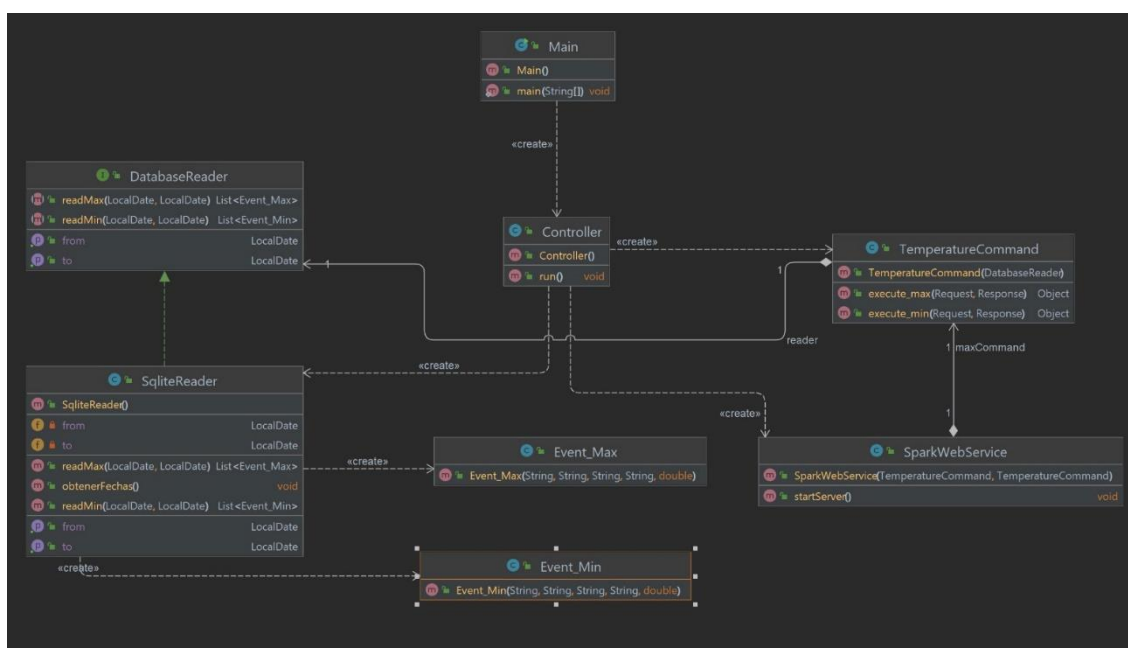
Feeder



Datamart-builder



## Temperature-API



### Conclusiones:

Este trabajo me ha servido, además de para obviamente aprender, para darme cuenta de todo lo que he aprendido este curso y para darme cuenta de que soy capaz de hacer más de lo que pensaba. Cuando empecé con el trabajo no tenía ni idea de cómo empezar, no sabía cómo crear un datalake, no sabía que eran un evento, no sabía cómo crear un datamart ...; por lo que al principio necesite bastante ayuda por parte de los profesores para poder empezar a desarrollarlo. A medida que los profesores me fueron explicando las cosas me di cuenta que realmente no era tan difícil como parecía sino que al no tener idea de que eran exactamente algunas cosas que me pedían no podía ni siquiera empezar, por ejemplo no sabía que con crear un datamart se referían a crear una base de datos o que con el datalake se referían a una carpeta. Una vez que tuve los conceptos básicos claros, y con ayuda de los profesores, pude empezar a desarrollar el directorio “Feeder”. Los otros dos directorios , “Datamart-builder” y “Temperature-API”, al empezar a hacerlos en navidades no pude contar con la ayuda de los maestros por lo que las tuve que desarrollarlos más bien sola, sin embargo aun así creo que el directorio que más me costo desarrollar fue el “Feeder”, ya que en los otros directorios utilizábamos aspectos empleados en anteriores trabajos. Además de lo anteriormente mencionado me gustaría hacer una mención especial a mi compañero Aris Vazdekis Soria con el cual llevo trabajando conjuntamente en el trabajo desde diciembre y sin el cual estoy segura de que no hubiera podido hacer el trabajo.

Este es sin duda el proyecto más difícil que he hecho y en el más que he aprendido. La verdad que siendo sincera mirando hacia atrás no puedo decir que ha sido lo más que me costado del trabajo, ya que en general todo el trabajo me ha costado bastante, pero algunas de las cosas que más me han costado han sido el crear la función “read”, en el “Feeder”, la cual lee

los eventos del “datalake” evitando así que se copien de nuevo los que ya están escritos y permitiendo escribir los eventos que no han sido copiados; otra cosa que también me ha costado bastante en el “Datamart-builder” es el extraer los datos del “datalake” y filtrarlos para que escribiese en el “datamart” solo los datos que pertenecen a las temperaturas máximas y mínimas.

Por último, si volviera a hacer este proyecto lo enfocaría de la misma manera, sin embargo, lo hubiera empezado antes por el hecho de que al haberlo realizado casi todo en navidades no pude pedir tutorías para algunas dudas que me fueron surgiendo, y porque al tener en paralelo que prepararme los exámenes de enero me vi un poco desbordada.

### **Líneas futuras:**

En un futuro para crear una aplicación útil y funcional se podría crear un bróker el cual se encargue de ejecutar al unísono los diferentes directorios, siendo así más fácil la obtención de datos. Al mismo tiempo también se podría filtrar por más lugares aparte de Gran Canaria consiguiendo así un archivo, el cual se podría consultar, de todas las regiones de España, haciendo de esta manera más fácil el consultar los datos de estas y permitiendo consultar datos más concretos.

La forma de desarrollar este proyecto sería utilizando la misma metodología que hemos utilizado en este trabajo, además también se le añadiría un bróker y nuevas clases a todos los directorios para poder filtrar por más regiones los datos y obtenerlos.

### **Bibliografía**

- <https://learn.microsoft.com/es-es/sql/t-sql/lesson-1-creating-database-objects?view=sql-server-ver16>
- <https://www.javadoc.io/doc/com.google.code.gson/gson/2.8.0/com/google/gson/TypeAdapter.html>
- [https://www.youtube.com/watch?v=f-kcvxYZrB4&ab\\_channel=CodinginFlow](https://www.youtube.com/watch?v=f-kcvxYZrB4&ab_channel=CodinginFlow)
- [ChatGPT](#)
- [Apuntes del campus de la API](#)