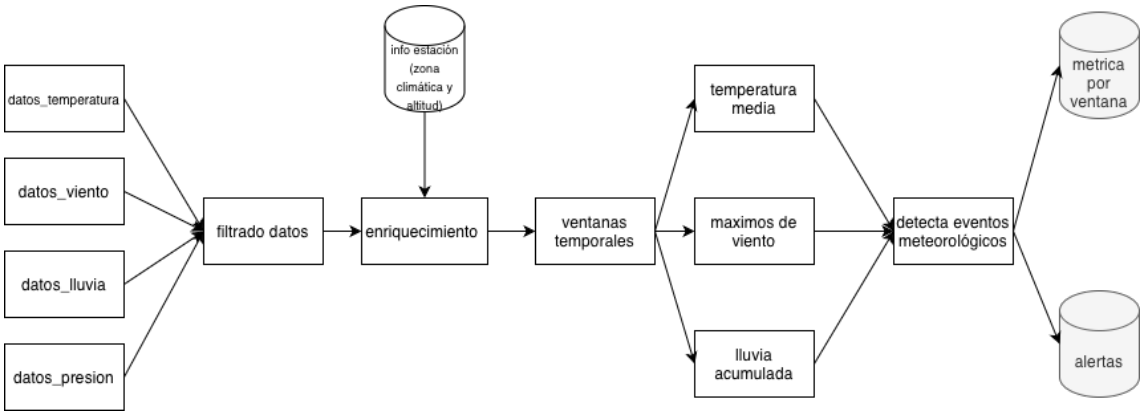
Introducción

El objetivo de este proyecto es diseñar y simular un sistema de detección de eventos meteorológicos en tiempo real utilizando una arquitectura orientada a streaming.

Aunque la implementación se realiza con pandas, el pipeline sigue el mismo modelo conceptual que Spark Structured Streaming, facilitando una futura migración a entornos distribuidos.

Flujo de los datos



EDA

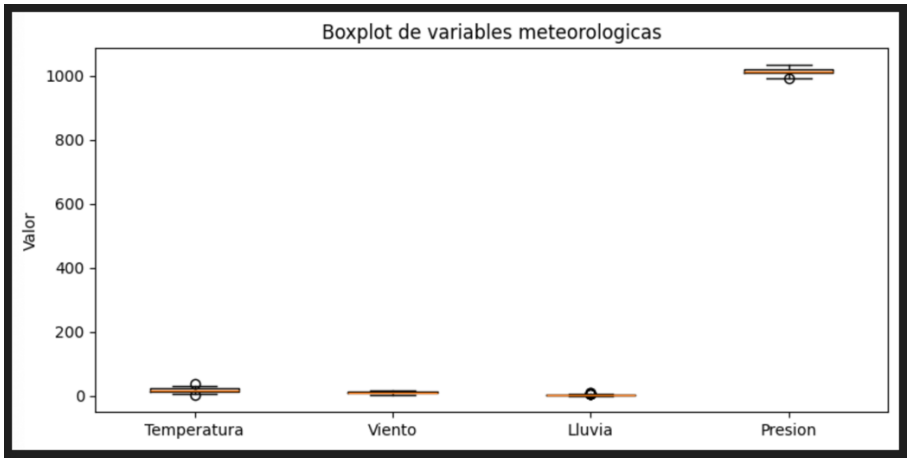
En primer lugar, se ha revisado el dataset con df.info() y se ha comprobado que hay dos variables, timestamp y station_id que son de tipo object, y cuatro variables, temperatura, wind_speed, rainfall y presure que son variables numéricas.

Además, hemos comprobado que no existen valores nulos y también se verificaron duplicados y no se encontraron, lo que indica que los datos están en buen estado para analizar.

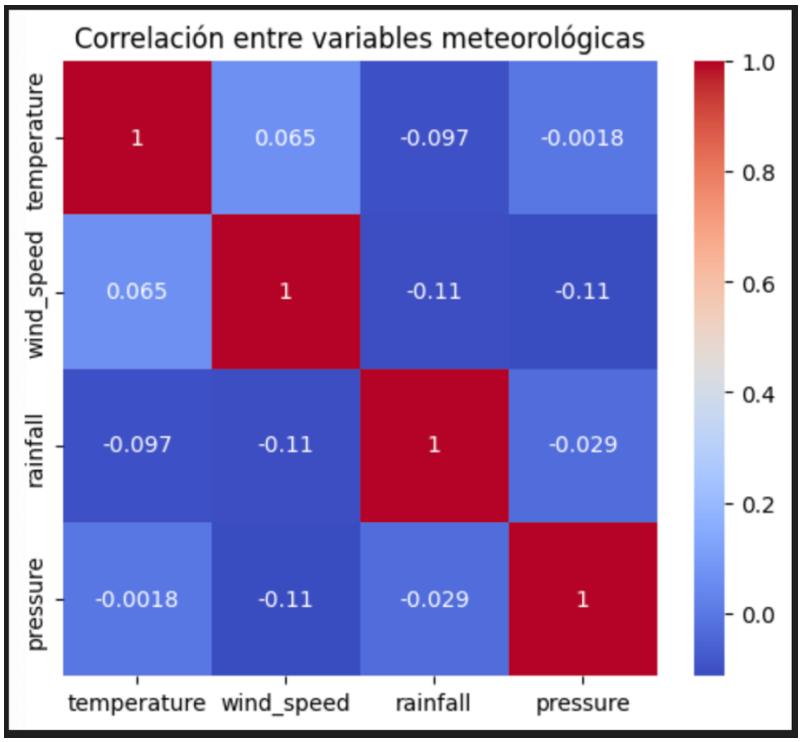
Por otro lado, se han realizado gráficos de series temporales para las variables temperature, wind_speed, rainfall y pressure. Estos gráficos nos muestran cómo cada variable varía a lo largo del tiempo, permitiendo identificar patrones generales, picos o fluctuaciones estacionales.

Por ejemplo, es posible observar ciertos picos de temperatura o aumentos de viento que podrían estar asociados a eventos meteorológicos específicos. Esta visualización nos ayuda a comprender la evolución temporal de las variables y nos sirve como referencia para detectar comportamientos atípicos.

Además, el análisis exploratorio mediante diagramas de caja revela una marcada heterogeneidad en las escalas de medida de las variables meteorológicas. Mientras que la Presión presenta una distribución concentrada en torno a los 1000 hPa con una dispersión mínima, las variables de Temperatura, Viento y Lluvia muestran valores significativamente menores, quedando comprimidas en la base del gráfico. Se pueden ver pocos valores atípicos (outliers), especialmente en la Temperatura y la Presión, lo que sugiere episodios meteorológicos puntuales fuera del rango intercuartílico.



En cuanto a la matriz de correlación, la matriz muestra que casi no existe relación entre las variables meteorológicas analizadas, ya que todos los valores de correlación son muy cercanos a cero. Los colores azules indican que las conexiones son prácticamente inexistentes o ligeramente negativas. Por ejemplo, la relación más 'fuerte' (aunque sigue siendo muy débil) se da entre la lluvia y la velocidad del viento con un -0.11. En conclusión, el comportamiento de una variable no permite predecir el de las demás, lo que indica que funcionan de manera independiente en este conjunto de datos.



Por último, hemos calculado los percentiles clave (25, 50, 75 y 95) para entender cómo se distribuyen los datos meteorológicos. Estos valores nos han permitido identificar, por ejemplo, que la mediana de la temperatura es de 17.32 °C y que el 95% de los días el viento no supera los 15.07 unidades. En general, estos resultados estadísticos confirman lo visto en los gráficos: la Presión mantiene valores altos y estables (sobre los 1013), mientras que la Lluvia muestra valores muy bajos en la mayor parte de los registros, lo que ayuda a entender mejor el comportamiento típico de cada variable.

Simulación de eventos con pandas:

Datos meteorológicos:

Cada registro representa una observación meteorológica:

- timestamp (event time)
- station_id
- temperature
- wind_speed
- rainfall

Las mediciones llegan de forma asíncrona y con frecuencia irregular.

Datos estaciones

Los atributos de las estaciones se almacenan por separado y se incorporan durante el enriquecimiento:

- altitude
- climate_zone

Enriquecimiento de los datos

Las mediciones dinámicas se enriquecen mediante un left join usando station_id, garantizando que no se pierdan observaciones. Las variables estáticas se propagan a nivel de ventana utilizando la agregación first(), al no cambiar en el tiempo.

Ventanas de tiempo

Uno de los elementos fundamentales del sistema es el uso de ventanas temporales para la agregación de datos en un contexto de procesamiento en streaming. Dado que las mediciones meteorológicas llegan de forma continua y asíncrona, resulta necesario agrupar los eventos en intervalos temporales discretos que permitan calcular métricas representativas y detectar eventos relevantes.

En este proyecto se utilizan ventanas temporales tumbling basadas en event time.

El tamaño de la ventana se implementa como un parámetro configurable en la definición, lo que permite evaluar distintos tiempos sin modificar la lógica del pipeline.

Para el estudio comparativo se evaluaron dos configuraciones principales: ventanas de 1 minuto y ventanas de 5 minutos. Las ventanas de 1 minuto ofrecen una mayor resolución temporal y permiten detectar variaciones rápidas en las variables meteorológicas. Sin embargo, al contener un número reducido de observaciones por estación, introducen un mayor nivel de ruido y provocan una sobre-detección de eventos, especialmente en variables altamente volátiles como la velocidad del viento.

Para el caso de las ventanas de un minuto obtenemos 165 ventanas de los datos ya que entran menos dato de la ventana dándonos mucho ruido en los resultados y no siendo tan concluyentes para el análisis

Por el contrario, las ventanas de 5 minutos agrupan un mayor número de observaciones por estación, lo que da lugar a métricas más estables y representativas del fenómeno meteorológico. Este enfoque reduce significativamente el ruido y disminuye la aparición de falsos positivos, proporcionando una base más robusta para la generación de alertas operativas.

Para el caso de las ventanas de 5 min conseguimos obtener menos ventanas pero con un mayor numero de eventos por ventana lo que hace que sea mejor de cara a realizar un posterior análisis

La elección final de ventanas tumbling de 5 minutos se fundamenta en una regla práctica habitual en sistemas de monitorización en tiempo real: el tamaño de la ventana debe ser varias veces superior a la frecuencia de muestreo de los datos para garantizar agregaciones fiables. Dado que la frecuencia de muestreo es de decenas de segundos, las ventanas de 5 minutos ofrecen un análisis posterior mas favorable que las ventanas de 1 min.

En consecuencia, el sistema adopta ventanas tumbling de 5 minutos como configuración final, al proporcionar métricas consistentes, reducir la detección de eventos espurios y alinearse con las prácticas habituales en sistemas de streaming y alertas en producción. Hemos probado a hacerlo con menos tiempo pero únicamente nos salia un evento o dos por ventana lo que no tenia tanto sentido de cara a un posterior análisis.

Para las alertas hemos definido tres situaciones:

- viento fuerte
- lluvia intensa
- ola de calor

Hemos buscado parámetros de cara a poder entender si se ha producido unos de estas situaciones:

- Viento máximo es superior a 80
- Lluvia acumulada es superior a 10
- Calor superior a 35

En caso de que se produzca uno de estas tres situaciones la alerta pasaría a ser de True lo que podría aplicarse a crear una alerta meteorológica.

Futuras mejoras

De cara a mejorar el estudio pensamos que podríamos incluir estas tres siguientes mejoras:

- Adaptación dinámica de los eventos basado en cada cuanto se produce una entrega de datos y podamos realizar un mejor estudio.
- Incorporación de eventos tardíos con watermarks y evitar que los eventos tardíos influyan en el envio de alertas ya que si se entrega un evento de una temperatura de madrugada en las de mediodía podría afectar a nuestro análisis.

- Mejora de las alertas de nuestros datos que permita conocer mejor el estado de la alerta y porque ha saltado, es decir añadir explicabilidad.
- Añadir sliding Windows para detectar los eventos que no se alinean con los limites de las ventanas.