PROYECTO PREDICCIÓN METEOROLOGICA

PROYECTO 3 - BD

JIAJIAO XU, JORDI VIDAL
STUCOM

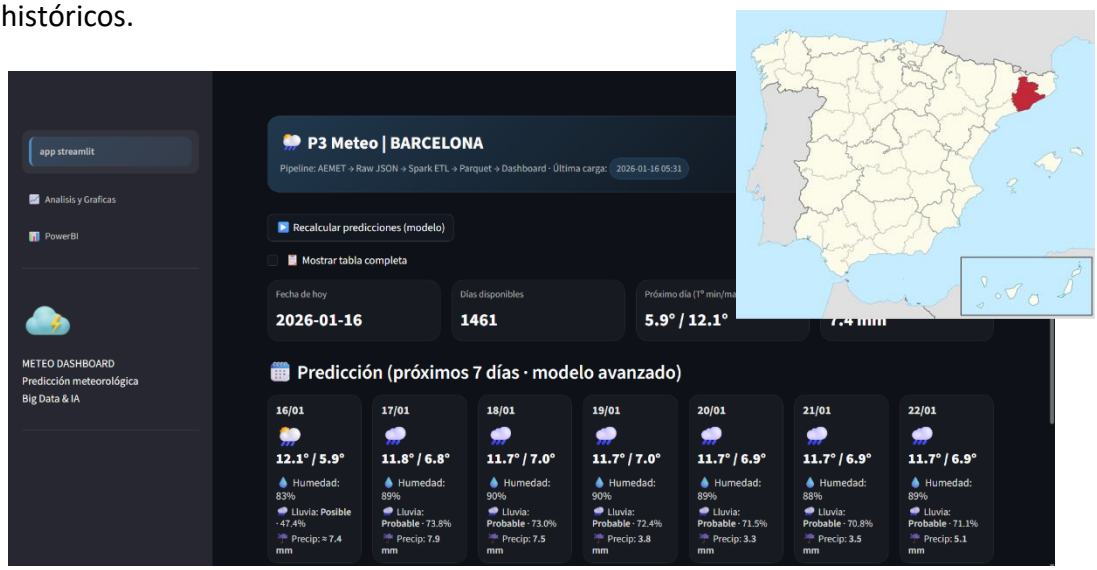
ÍNDICE

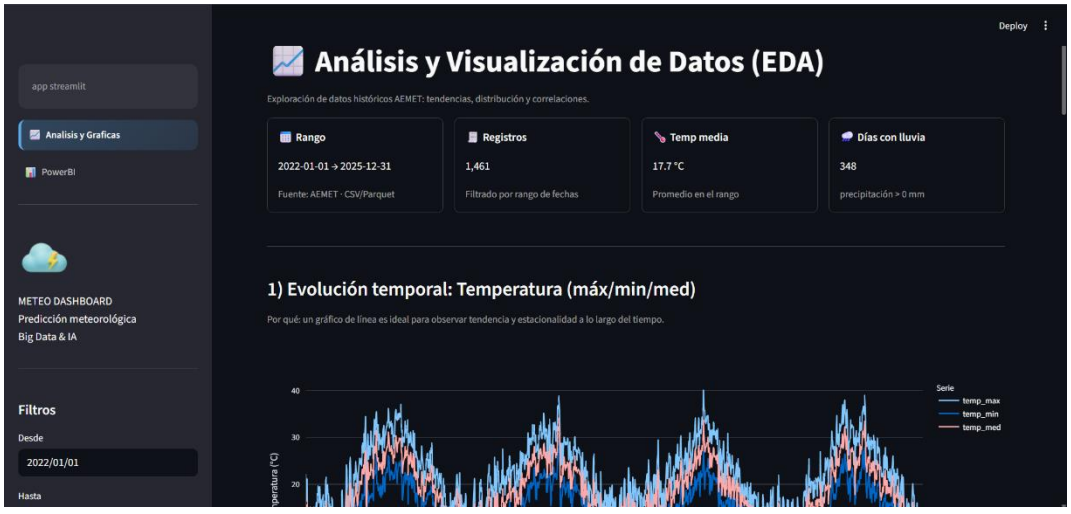
1. Introducción

Este proyecto tiene como objetivo el desarrollo de un **sistema de predicción meteorológica** basado en una arquitectura Big Data, utilizando datos oficiales proporcionados por la Agencia Estatal de Meteorología (AEMET).

El sistema implementa un flujo completo que abarca la **adquisición de datos**, su **procesamiento mediante técnicas ETL**, el **almacenamiento optimizado** y la **visualización interactiva**. Además, incorpora un componente de Machine Learning basado en un enfoque modular: un modelo de clasificación para estimar la probabilidad de lluvia y modelos de regresión para predecir variables continuas (temperatura, humedad y precipitación), incluyendo evaluación mediante backtesting y predicción dinámica a 7 días.

El proyecto se centra en el municipio de **Barcelona**, aunque la arquitectura está diseñada para ser escalable y adaptable a otros municipios o a un mayor volumen de datos históricos.

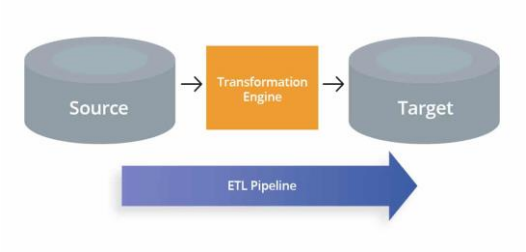


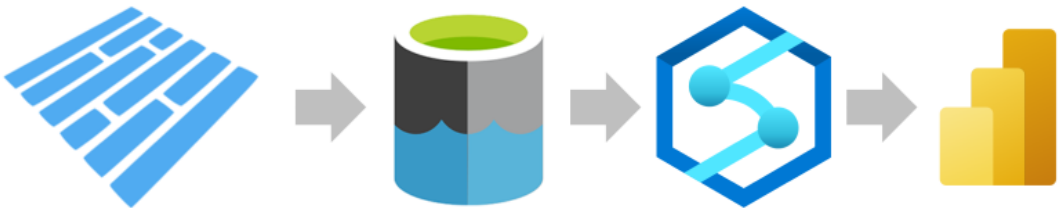


2. Objetivos del proyecto

Los principales objetivos del proyecto son los siguientes:

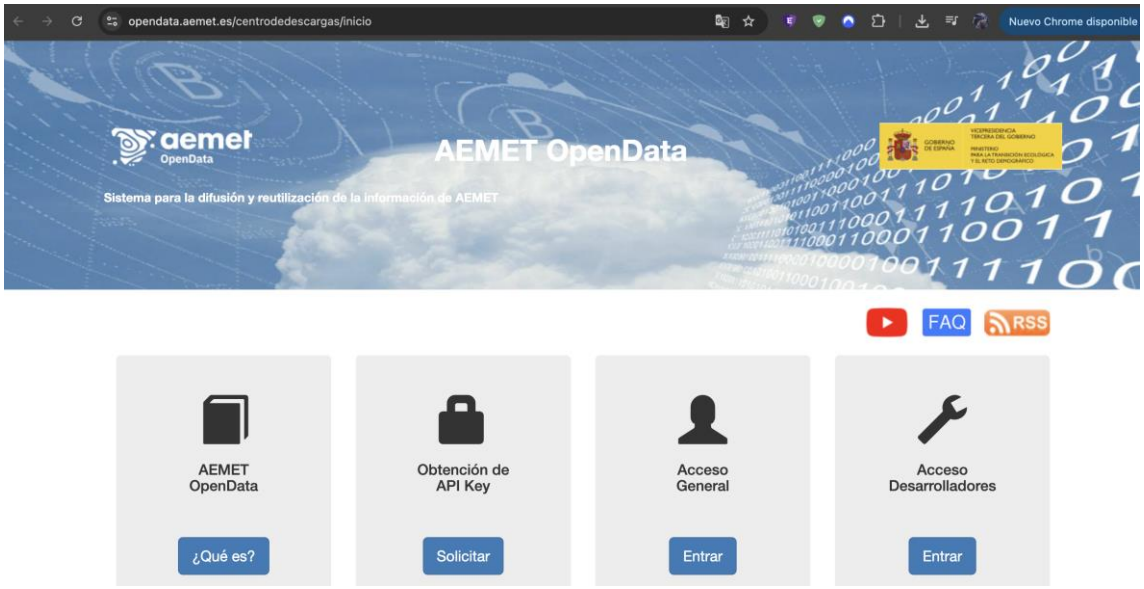
- Diseñar e implementar un **pipeline Big Data completo y funcional**.
- Obtener datos meteorológicos a partir de una **fuente oficial y fiable** mediante una API REST.
- Procesar los datos utilizando **Apache Spark**, resolviendo inconsistencias propias del formato de origen.
- Almacenar la información procesada en formato **Parquet**, optimizado para análisis y escalabilidad.
- Desarrollar un **cuadro de mandos interactivo** que permita explorar los datos mediante filtros, indicadores y gráficos.
- Integrar modelos supervisados para realizar predicciones meteorológicas a corto plazo: clasificación de lluvia (probabilidad) y regresión de variables continuas (temperatura, humedad y precipitación), con evaluación mediante métricas (AUC/MAE) y backtesting.
- Garantizar la **reproducibilidad y automatización** del sistema.





3. Origen y adquisición de datos

Los datos utilizados en el proyecto proceden de la plataforma **AEMET OpenData**, que ofrece acceso público a información meteorológica oficial a través de una API REST.



Plataforma AEMET OpenData utilizada como fuente oficial de datos meteorológicos.

En concreto, se ha utilizado el servicio de **predicción diaria por municipio**, obteniendo datos como:

- Temperatura máxima, mínima y media (tmax, tmin, tmed)
- Humedad relativa media (hrMedia)
- Precipitación diaria (prec)
- Fecha de observación (fecha)
- Estación meteorológica

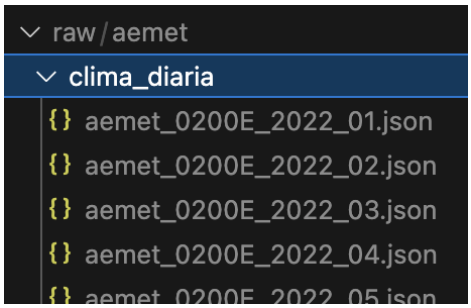
La adquisición de datos se realiza mediante un script en Python que consume la API utilizando una clave de acceso (API Key), almacenando la respuesta original en formato **JSON** dentro de una capa de datos sin procesar (*raw layer*).

```
(.venv) macbookpro@MacBook-Pro-de-MACBOOK P3_Meteo_BigData % python fetch_aemet_barcelona.py

  Request 1: AEMET diaria municipio 08019
  Datos URL: https://opendata.aemet.es/opendata/sh/9f856d46
   OK guardado: data/raw/aemet/municipio_diaria/barcelona_08019_20260108_193402.json
   Registros: 1
(.venv) macbookpro@MacBook-Pro-de-MACBOOK P3_Meteo_BigData %
```

Ejecución del script de adquisición de datos mediante la API de AEMET.

Ejemplo de fichero JSON almacenado en la capa *raw*.



Para acceder a la API de AEMET fue necesario solicitar una clave de acceso (API Key), la cual se gestiona mediante una variable de entorno por motivos de seguridad.

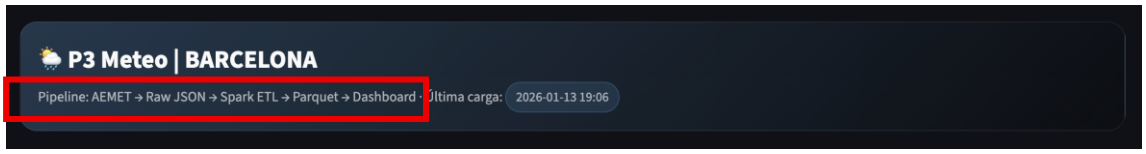
4. Arquitectura y pipeline Big Data

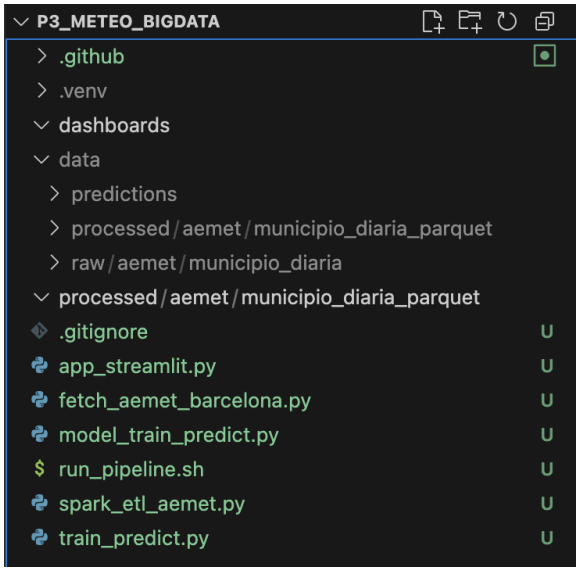
La arquitectura del sistema sigue un enfoque **Big Data por capas**, separando claramente las distintas fases del tratamiento de los datos. Esta separación facilita la escalabilidad, el mantenimiento y la trazabilidad del sistema.

El pipeline implementado consta de las siguientes capas:

- **Capa de adquisición (Ingesta)**
  Obtención de los datos meteorológicos mediante la API REST de AEMET.
- **Capa Raw (datos sin procesar)**
  Almacenamiento de la respuesta original de la API en formato JSON, sin modificaciones.
- **Capa de procesamiento (ETL)**
  Transformación y normalización de los datos utilizando Apache Spark.
- **Capa Processed (datos procesados)**
  Almacenamiento de los datos limpios y estructurados en formato Parquet, particionados por fecha.
- **Capa de visualización**
  Explotación de los datos mediante un cuadro de mandos interactivo desarrollado con Streamlit.

Este diseño permite que el sistema pueda ampliarse fácilmente para incluir nuevos municipios, un mayor histórico de datos o nuevos modelos analíticos sin necesidad de modificar la arquitectura base.





Estructura del proyecto con separación por capas (raw / processed).

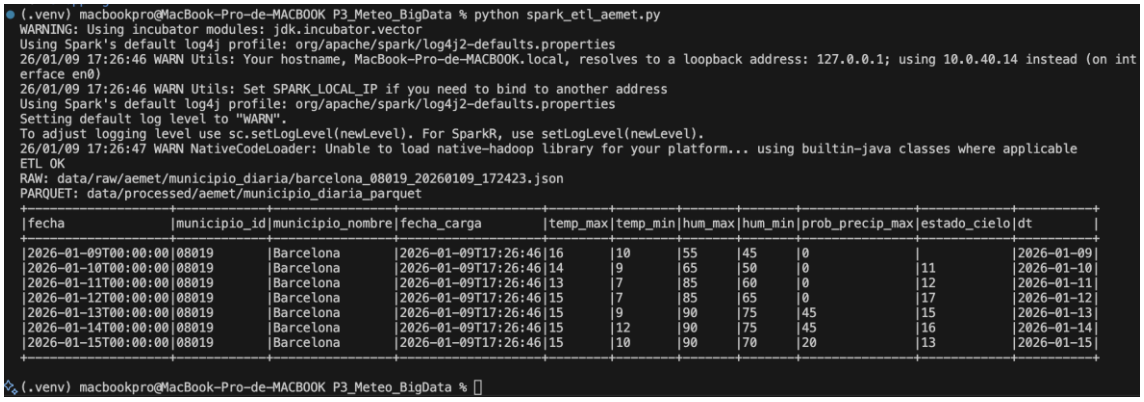
5. Proceso ETL

El proceso ETL (Extract, Transform, Load) se ha implementado utilizando **Apache Spark en modo local**, lo que permite simular un entorno Big Data real incluso con un volumen de datos reducido.

5.1 Extracción (Extract)

La fase de extracción consiste en la lectura de los ficheros JSON almacenados en la capa *raw*. Estos ficheros contienen la información meteorológica diaria devuelta por la API de AEMET para el municipio seleccionado.

Cada ejecución del proceso ETL consume el fichero más reciente disponible, garantizando que el sistema siempre trabaja con la última información obtenida.



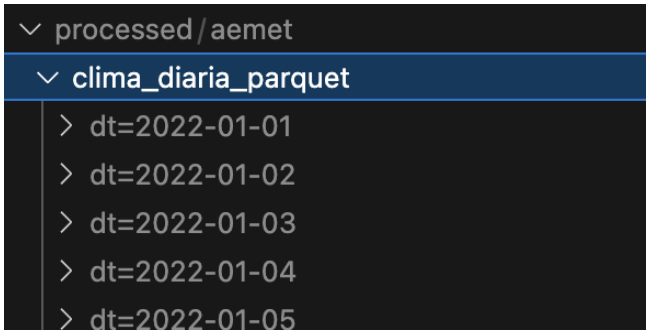
Ejecución del proceso ETL con Apache Spark y generación de datos estructurados.

5.2 Transformación (Transform)

Durante la fase de transformación se llevan a cabo las siguientes operaciones:

- Normalización de estructuras JSON complejas (listas y diccionarios anidados).
- Conversión de tipos de datos heterogéneos a tipos consistentes.
- Extracción de variables relevantes como temperatura máxima y mínima, humedad relativa y probabilidad de precipitación.
- Normalización de valores anómalos, como probabilidades de precipitación superiores al 100%.
- Creación de una columna de fecha (`dt`) para facilitar la partición y el análisis temporal.
- Eliminación de inconsistencias propias del formato original de la API.

El uso de un **esquema explícito** en Spark evita errores de inferencia y garantiza la estabilidad del proceso ante cambios en la estructura de los datos de origen.



Almacenamiento de datos procesados en formato Parquet particionado por fecha.

5.3 Carga (Load)

Una vez transformados, los datos se almacenan en la capa *processed* utilizando el formato **Parquet**, optimizado para análisis y consultas analíticas.

El proceso de carga se realiza en modo *append*, conservando el histórico completo de ejecuciones para mantener la trazabilidad de los datos. Las posibles duplicidades se gestionan posteriormente en la capa de visualización.

NOTA:

Durante el desarrollo del proyecto se detectaron diversas incidencias reales, como discrepancias en el rango temporal de los datos y desfases en el resumen diario.

En particular, el bloque “Resumen (7 días)” mostraba inicialmente los primeros registros del dataset, lo que podía provocar un desfase respecto a la fecha actual.

Esta incidencia se resolvió filtrando los datos a partir de la fecha actual y seleccionando dinámicamente los siete días siguientes disponibles, garantizando así la coherencia temporal del dashboard.

6. Almacenamiento de datos

El sistema de almacenamiento sigue una estructura tipo **Data Lake**, organizada en dos niveles principales:

6.1 Capa Raw

La capa *raw* contiene los datos originales en formato JSON, exactamente como son devueltos por la API de AEMET.

Esta capa permite:

- Auditoría y trazabilidad de los datos.
- Reprocesamiento en caso de errores.
- Comparación entre distintas ejecuciones.

Ejemplo de estructura:

```
data/raw/aemet/municipio_diaria/
data/raw/aemet/clima_diaria/
```

6.2 Capa Processed

La capa *processed* almacena los datos transformados en formato **Parquet**, particionados por fecha (dt). Este enfoque mejora el rendimiento de lectura y facilita el análisis temporal.

Ejemplo de estructura:

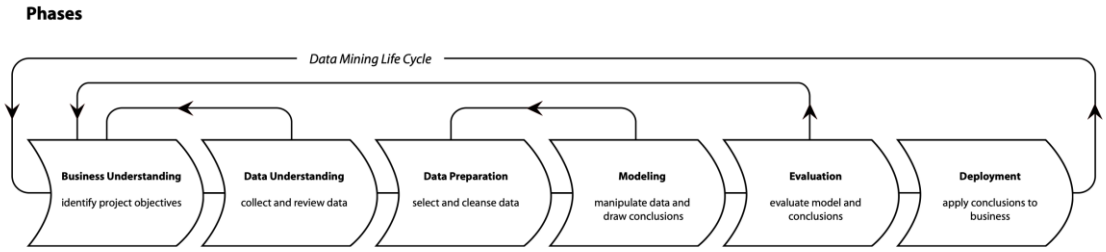
```
data/processed/aemet/municipio_diaria_parquet/
 └── dt=2026-01-08/
 └── dt=2026-01-09/

data/processed/aemet/clima_diaria_parquet/
 └── dt=2022-01-01/
 └── dt=2022-01-02/
```

El uso de Parquet permite una lectura eficiente, compresión automática y compatibilidad con herramientas analíticas y de visualización.

6.3 Metodología de trabajo (CRISP-DM)

El desarrollo del proyecto ha seguido la metodología CRISP-DM (Cross Industry Standard Process for Data Mining), ampliamente utilizada en proyectos de análisis de datos y machine learning. Esta metodología permite estructurar el proceso completo desde la comprensión del problema hasta el despliegue del sistema, garantizando coherencia, reproducibilidad e interpretabilidad de los resultados.



6.3.1. Comprensión del negocio (Business Understanding)

El objetivo principal del proyecto es predecir condiciones meteorológicas a corto plazo en el municipio de Barcelona, con especial énfasis en:

- Temperatura (máxima, mínima y media)
- Probabilidad real de lluvia
- Cantidad estimada de precipitación

El sistema está orientado a usuarios finales, por lo que se priorizan los siguientes aspectos:

- Interpretabilidad de los resultados
- Coherencia temporal de las predicciones
- Evitar falsos positivos en la predicción de lluvia

6.3.2. Comprensión de los datos (Data Understanding)

Los datos utilizados proceden de la plataforma AEMET OpenData, con histórico diario desde el año 2022 hasta la actualidad.

Durante la fase de exploración se identificaron diversas características relevantes:

- Presencia de valores faltantes
- Estructuras JSON heterogéneas
- Un fuerte desbalance entre días con y sin precipitación
- Diferencias entre la probabilidad de precipitación y la ocurrencia real de lluvia

6.3.3. Preparación de los datos (Data Preparation)

En esta fase se llevó a cabo un proceso ETL completo utilizando Apache Spark, incluyendo:

- Limpieza y normalización de los datos
- Conversión a formato Parquet
- Creación de variables derivadas (feature engineering):
- Variables temporales (mes, día del año)
- Variables históricas (lags)
- Ventanas móviles (rolling means)

Asimismo, se realizó una separación clara entre las variables de entrada (features) y las variables objetivo (targets).

6.3.4. Modelado (Modeling)

Se implementó un enfoque basado en dos tipos de modelos supervisados, diferenciados según la naturaleza de la variable a predecir.

- Modelo de lluvia (clasificación)

- Objetivo: predecir la ocurrencia de lluvia
- Salida: probabilidad de lluvia
- Métrica principal: AUC

Un valor del 50% de probabilidad no implica necesariamente un evento de lluvia, por lo que se definieron umbrales interpretables:

- < 40%: No
- 40–60%: Posible
- > 60%: Probable

- Modelos de regresión (temperatura, humedad y precipitación)

- Predicción de valores continuos
- Métrica utilizada: MAE

En el caso de la precipitación, la cantidad solo se estima cuando el modelo de lluvia indica un evento probable.

6.3.5. Evaluación (Evaluation)

Se realizó un proceso de backtesting temporal sobre los últimos 120 días, utilizando métricas específicas según el tipo de modelo:

- Temperatura máxima: MAE ≈ 1.8 °C

- Temperatura mínima: MAE ≈ 1.4 °C
- Humedad media: MAE ≈ 9 %
- Lluvia: AUC ≈ 0.75

El histórico utilizado para el modelado procede del servicio de climatología diaria por estación (observaciones), mientras que el sistema genera predicciones a 7 días mediante modelos supervisados.

6.3.6. Despliegue (Deployment)

Los modelos se integraron en scripts independientes de entrenamiento y predicción, así como en un dashboard interactivo desarrollado con Streamlit.

Las predicciones se almacenan en ficheros CSV y se consumen dinámicamente desde la interfaz, permitiendo actualización automática, reentrenamiento bajo demanda y una visualización clara de los resultados.

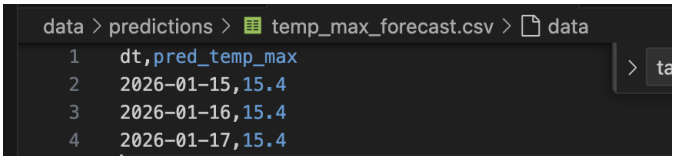
7. Modelo predictivo

En el sistema se ha integrado un componente de Machine Learning con un enfoque modular, diferenciando entre modelos de clasificación (lluvia) y modelos de regresión (temperatura, humedad y precipitación), con el objetivo de generar predicciones interpretables y coherentes a corto plazo, complementando la información proporcionada por la fuente de datos original.

7.1 Selección del modelo

En lugar de un único modelo simple, se ha optado por un enfoque modular basado en modelos supervisados diferenciados, adaptados a la naturaleza de cada variable meteorológica.

Este enfoque permite aplicar técnicas de clasificación para la predicción de lluvia y modelos de regresión para variables continuas, priorizando la interpretabilidad y la coherencia temporal de los resultados frente a la complejidad innecesaria.



Resultado del modelo predictivo con la estimación de la temperatura máxima.

7.2 Proceso de entrenamiento y predicción

El proceso de entrenamiento y predicción incluye las siguientes etapas:

- Ingeniería de características
- Separación temporal entre entrenamiento y validación
- Backtesting sobre datos recientes
- Predicción dinámica (rolling) para horizontes futuros
- Separación clara entre el proceso de entrenamiento y la visualización de resultados

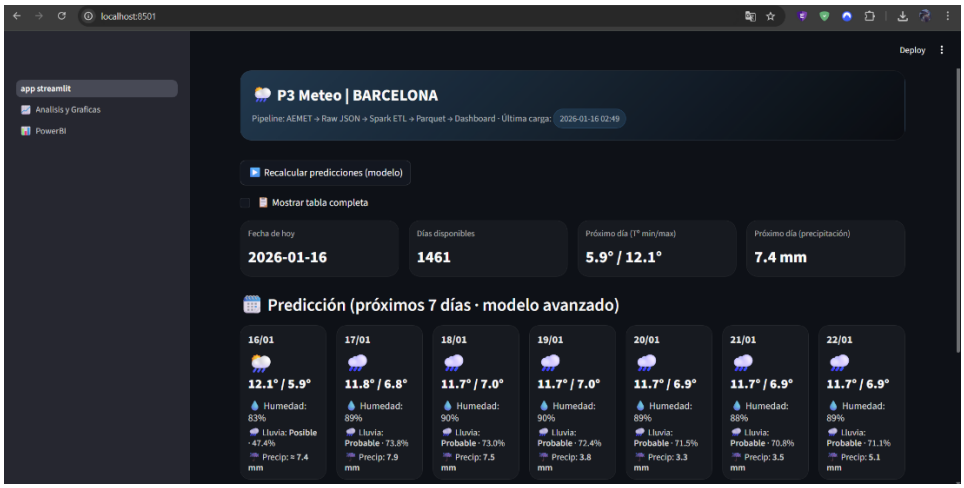
Este enfoque garantiza resultados más realistas y alineados con el comportamiento temporal de los datos meteorológicos.

7.3 Justificación técnica del enfoque

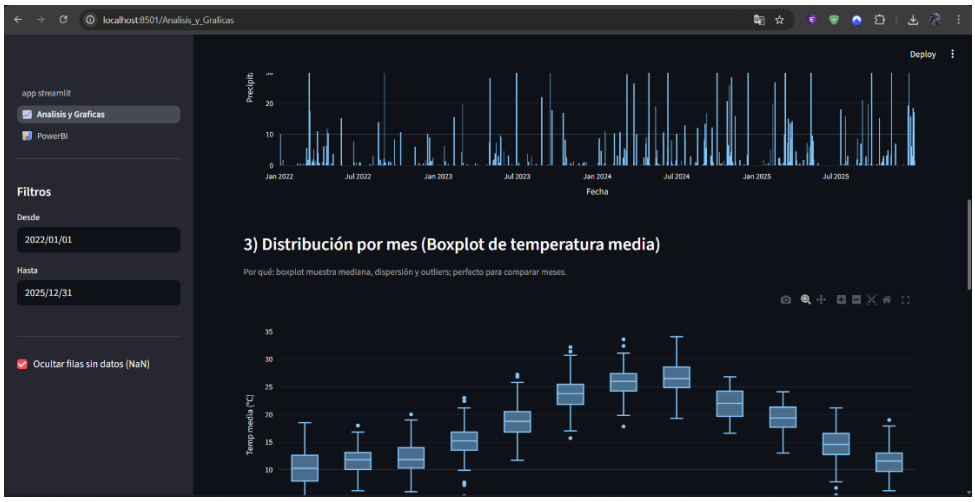
Un valor del 50% de probabilidad no implica necesariamente un evento de lluvia, por lo que se ha diseñado una lógica explícita de interpretación probabilística para evitar resultados engañosos para el usuario final.

8. Cuadro de mandos y visualización

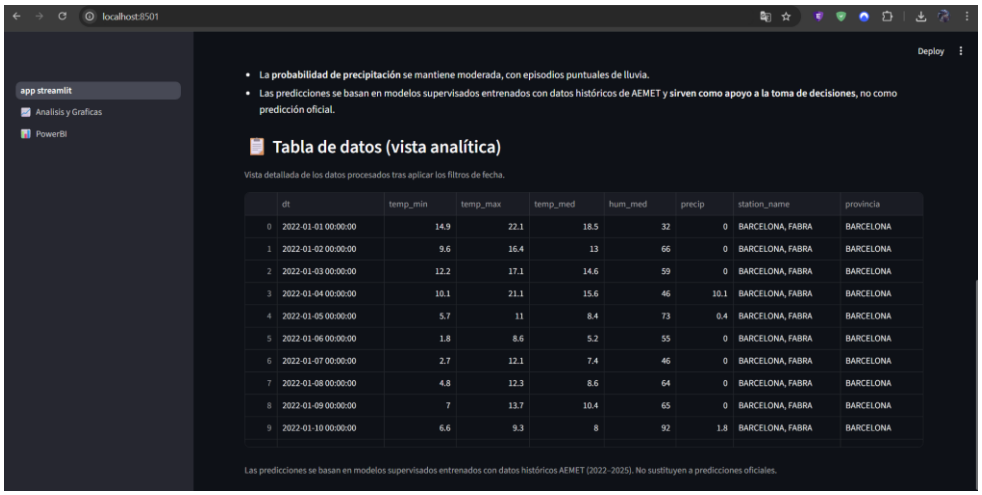
A continuación, se muestran capturas reales del dashboard en ejecución (modo Windows fallback mediante CSV).



Vista general del dashboard (cards y predicción 7 días).



Visualización: temperatura observada y precipitación diaria.



	dt	temp_min	temp_max	temp_med	hum_med	precip	station_name	provincia
0	2022-01-01 00:00:00	14.9	22.1	18.5	32	0	BARCELONA, FABRA	BARCELONA
1	2022-01-02 00:00:00	9.6	16.4	13	66	0	BARCELONA, FABRA	BARCELONA
2	2022-01-03 00:00:00	12.2	17.1	14.6	59	0	BARCELONA, FABRA	BARCELONA
3	2022-01-04 00:00:00	10.1	21.1	15.6	46	10.1	BARCELONA, FABRA	BARCELONA
4	2022-01-05 00:00:00	5.7	11	8.4	73	0.4	BARCELONA, FABRA	BARCELONA
5	2022-01-06 00:00:00	1.8	8.6	5.2	55	0	BARCELONA, FABRA	BARCELONA
6	2022-01-07 00:00:00	2.7	12.1	7.4	46	0	BARCELONA, FABRA	BARCELONA
7	2022-01-08 00:00:00	4.8	12.3	8.6	64	0	BARCELONA, FABRA	BARCELONA
8	2022-01-09 00:00:00	7	13.7	10.4	65	0	BARCELONA, FABRA	BARCELONA
9	2022-01-10 00:00:00	6.6	9.3	8	92	1.8	BARCELONA, FABRA	BARCELONA

Tabla analítica con los datos procesados y filtrados.

Para la explotación de los datos procesados se ha desarrollado un **cuadro de mandos interactivo** utilizando la herramienta **Streamlit**.

El dashboard permite visualizar de forma clara e intuitiva la información meteorológica y los resultados del modelo predictivo.

8.1 Funcionalidades del dashboard

El cuadro de mandos incluye las siguientes funcionalidades:

- **Indicadores clave (KPIs)** con información resumida del municipio y del periodo seleccionado.
- **Tarjetas resumen** con la predicción meteorológica de los próximos siete días.
- **Gráficas de evolución temporal** de la temperatura y la probabilidad de precipitación.
- **Visualización de resultados del modelo predictivo**, tanto en forma de gráfico como de indicadores.
- **Vista analítica opcional**, que permite mostrar la tabla completa de datos procesados para inspección detallada.

Para la predicción de lluvia, el dashboard incorpora una lógica de interpretación probabilística orientada al usuario final. En lugar de mostrar únicamente un porcentaje, la probabilidad se clasifica en rangos (No / Posible / Probable), evitando que valores intermedios como el 50% se interpreten erróneamente como un evento asegurado.

8.2 Gestión de inconsistencias y calidad de datos

Durante el desarrollo del proyecto se identificaron diversas **inconsistencias en los datos de origen**, propias de la estructura heterogénea de la API de AEMET. Para garantizar la calidad de los datos, se implementaron las siguientes medidas:

- Normalización de estructuras complejas (listas y diccionarios).
- Conversión de tipos de datos inconsistentes.
- Limitación de la probabilidad de precipitación al rango [0,100].
- Eliminación de duplicados en la capa de visualización, conservando únicamente la información más reciente por fecha.

Estas medidas aseguran que el dashboard muestre información coherente y fiable, independientemente del número de ejecuciones del proceso ETL.

8.3 Separación entre análisis y visualización

El dashboard diferencia claramente entre:

- **Vista de visualización**, orientada a usuarios finales, con gráficos y tarjetas resumen.
- **Vista analítica**, activable mediante un control específico, destinada a la inspección detallada de los datos procesados.

Esta separación mejora la usabilidad del sistema y permite adaptar la información al perfil del usuario.

El cuadro de mandos muestra por defecto la predicción meteorológica completa disponible, correspondiente a los próximos días proporcionados por la API de AEMET.

Inicialmente se incorporaron filtros temporales manuales (Desde / Hasta), pero se eliminaron tras comprobar que, al tratarse de predicciones a corto plazo y no de un histórico extenso, dichos filtros no aportaban valor analítico y podían inducir a confusión en el usuario final.

9. Casos de uso y pruebas

Con el objetivo de validar el correcto funcionamiento del sistema, se han definido y ejecutado diversos **casos de uso y pruebas funcionales**, cubriendo las distintas fases del pipeline.

9.1 Caso de uso principal

CU-01: Consulta de predicción meteorológica

- **Actor:** Usuario
- **Descripción:** El usuario accede al cuadro de mandos para consultar la predicción meteorológica del municipio de Barcelona.
- **Flujo principal:**
    1. El sistema obtiene los datos procesados desde la capa *processed*.
    2. Se aplican los filtros temporales seleccionados por el usuario.
    3. Se muestran los indicadores, gráficos y tarjetas resumen.
    4. Se visualizan los resultados del modelo predictivo.
- **Resultado esperado:** El usuario obtiene información meteorológica clara, actualizada y coherente.

9.2 Pruebas realizadas

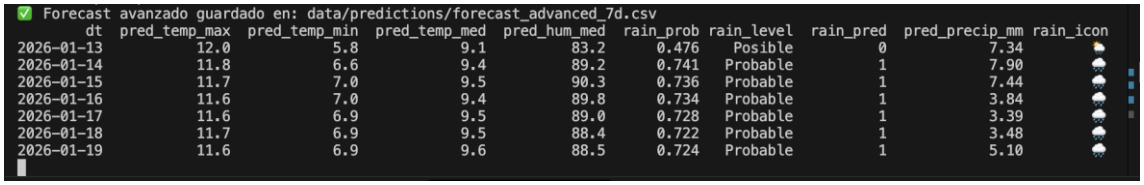
Se han realizado las siguientes pruebas:

- **Prueba de adquisición de datos**
    - Ejecución del script de descarga desde la API de AEMET.
    - Verificación de la correcta generación del fichero JSON en la capa *raw*.
- **Prueba de proceso ETL**
    - Ejecución del proceso Spark ETL.
    - Validación de la transformación correcta de los datos y generación de ficheros Parquet.
- **Prueba de calidad de datos**
    - Comprobación de la normalización de valores anómalos (probabilidad de precipitación).
    - Verificación de la coherencia de tipos de datos.
- **Prueba de visualización**
    - Comprobación de la correcta aplicación de filtros de fecha.
    - Validación de la actualización dinámica de KPIs y gráficos.

Evaluación temporal (backtesting) sobre una ventana reciente (últimos 120 días), utilizando MAE para regresión y AUC para clasificación.

- **Prueba del modelo predictivo**
    - Entrenamiento del modelo con el histórico disponible.
    - Generación y visualización de predicciones futuras.

Todas las pruebas se han ejecutado con resultados satisfactorios, garantizando la estabilidad del sistema.



✅ Forecast avanzado guardado en: data/predictions/forecast_advanced_7d.csv

dt	pred_temp_max	pred_temp_min	pred_temp_med	pred_hum_med	rain_prob	rain_level	rain_pred	pred_precip_mm	rain_icon
2026-01-13	12.0	5.8	9.1	83.2	0.476	Posible	0	7.34	
2026-01-14	11.8	6.6	9.4	89.2	0.741	Probable	1	7.90	
2026-01-15	11.7	7.0	9.5	90.3	0.736	Probable	1	7.44	
2026-01-16	11.6	7.0	9.4	89.8	0.734	Probable	1	3.84	
2026-01-17	11.6	6.9	9.5	89.0	0.728	Probable	1	3.39	
2026-01-18	11.7	6.9	9.5	88.4	0.722	Probable	1	3.48	
2026-01-19	11.6	6.9	9.6	88.5	0.724	Probable	1	5.10	

10. Manual de instalación y ejecución

10.1 Requisitos del sistema

- Python 3.10 o superior
- Java JDK 17
- Entorno virtual Python
- Sistema operativo Windows, macOS o Linux

10.2 Instalación del entorno

1. Crear un entorno virtual:
   ```
   python -m venv .venv
   ```

2. Activar el entorno virtual:

   - Windows:
     ```
     .venv\Scripts\activate
     ```

   - macOS / Linux:
     ```
     source .venv/bin/activate
     ```

3. Instalar dependencias:
   ```
   pip install -r requirements.txt
   ```

10.3 Ejecución del sistema

El sistema puede ejecutarse de forma manual o mediante un script automatizado.

Ejecución manual del pipeline completo:

1. Descarga de datos desde AEMET OpenData:
   ```
   python fetch_aemet_barcelona.py
   ```

2. Proceso ETL y generación de datos estructurados:
   ```
   python spark_etl_aemet.py
   ```

3. Entrenamiento y predicción de modelos supervisados:

   - Modelo de clasificación de lluvia:
     ```
     python rain_train_predict.py
     ```

   - Modelo avanzado con ingeniería de características y predicción dinámica a 7 días:
     ```
     python model_advanced_train_predict.py
     ```

4. Ejecución del cuadro de mandos:
   ```
   streamlit run app_streamlit.py
   ```

Los scripts de entrenamiento generan ficheros CSV en el directorio `data/predictions/`,
los cuales son consumidos dinámicamente por el dashboard para la visualización de resultados.

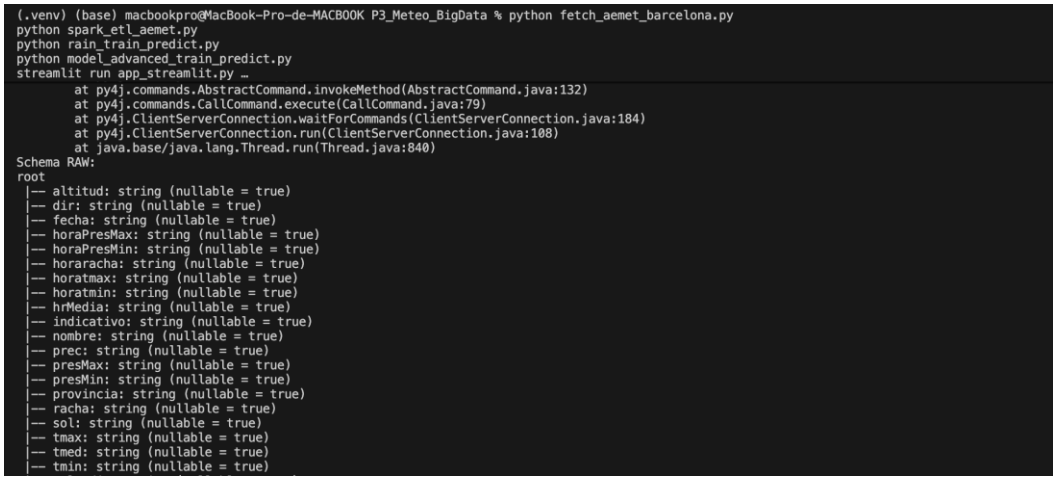
De forma alternativa, el proyecto incluye el script run_pipeline.sh, que permite ejecutar todo el pipeline de forma automatizada, desde la adquisición de datos hasta la actualización del dashboard.

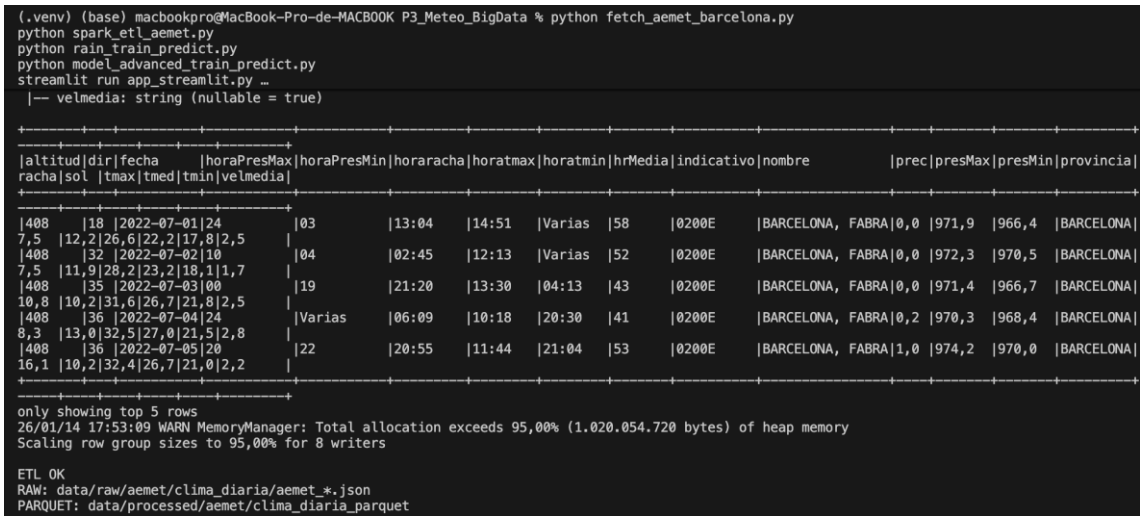
10.4 Estructura del proyecto

```
data/
  ├── raw/
  └── processed/
docs/
scripts/
```

El directorio `data/predictions/` contiene los resultados de los modelos predictivos en formato CSV, utilizados por el dashboard.

Esta estructura permite una clara separación entre datos originales, datos procesados y código del sistema.





```
ETL OK
RAW: data/raw/aemet/clima_diaria/aemet_*.json
PARQUET: data/processed/aemet/clima_diaria_parquet
+----------+----------+
|    min_dt|    max_dt|
+----------+----------+
|2022-01-01|2025-12-31|
+----------+----------+

+----------+----------+--------+--------+--------+------+-------+----------+----------------+---------+
|dt        |fecha     |temp_max|temp_min|temp_med|precip|hum_med|station_id|station_name    |provincia|
+----------+----------+--------+--------+--------+------+-------+----------+----------------+---------+
|2022-07-01|2022-07-01|26.6    |17.8    |22.2    |0.0   |58.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-02|2022-07-02|28.2    |18.1    |23.2    |0.0   |52.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-03|2022-07-03|31.6    |21.8    |26.7    |0.0   |43.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-04|2022-07-04|32.5    |21.5    |27.0    |0.2   |41.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-05|2022-07-05|32.4    |21.0    |26.7    |1.0   |53.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-06|2022-07-06|27.8    |19.0    |23.4    |0.9   |71.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-07|2022-07-07|30.5    |18.4    |24.4    |0.0   |55.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-08|2022-07-08|30.4    |21.7    |26.0    |0.0   |52.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-09|2022-07-09|30.7    |22.5    |26.6    |0.0   |54.0   |0200E     |BARCELONA, FABRA|BARCELONA|
|2022-07-10|2022-07-10|31.8    |22.3    |27.0    |0.0   |42.0   |0200E     |BARCELONA, FABRA|BARCELONA|
+----------+----------+--------+--------+--------+------+-------+----------+----------------+---------+
only showing top 10 rows
Rain rate histórico (lluvia=1): 0.238
```

```
Test últimos 90 días -> Accuracy: 0.656 | ROC-AUC: 0.553
Predicción lluvia guardada en: data/predictions/rain_forecast.csv
        dt  rain_prob  rain_pred rain_level rain_icon
2026-01-14      0.505          1      Media         🌦
2026-01-15      0.506          1      Media         🌦
2026-01-16      0.507          1      Media         🌦
2026-01-17      0.507          1      Media         🌦
2026-01-18      0.508          1      Media         🌦
2026-01-19      0.509          1      Media         🌦
2026-01-20      0.510          1      Media         🌦
Columnas base: ['temp_max', 'temp_min', 'temp_med', 'precip', 'hum_med']
Nº features: 34
Rango datos: 2022-01-08 -> 2025-12-31
Backtest temp_max últimos 120 días -> MAE: 1.82
Backtest temp_min últimos 120 días -> MAE: 1.42
Backtest temp_med últimos 120 días -> MAE: 1.47
Backtest hum_med últimos 120 días -> MAE: 9.13
Backtest lluvia últimos 120 días -> AUC: 0.753 | Rain rate test: 0.342
Backtest precip(mm) (solo días con lluvia) -> MAE: 5.38 mm
```

```
✅ Forecast avanzado guardado en: data/predictions/forecast_advanced_7d.csv
        dt  pred_temp_max  pred_temp_min  pred_temp_med  pred_hum_med  rain_prob rain_level  rain_pred  pred_precip_mm rain_icon
2026-01-14           12.1            5.9            9.1          83.2      0.476     Posible          0            7.35         🌦
2026-01-15           11.8            6.8            9.4          89.2      0.745    Probable          1            7.90         🌧
2026-01-16           11.7            7.0            9.5          90.3      0.742    Probable          1            7.45         🌧
2026-01-17           11.6            7.0            9.5          89.8      0.736    Probable          1            3.83         🌧
2026-01-18           11.7            6.9            9.5          89.0      0.728    Probable          1            3.34         🌧
2026-01-19           11.7            6.9            9.6          88.2      0.722    Probable          1            3.46         🌧
2026-01-20           11.7            6.9            9.6          88.5      0.724    Probable          1            5.10         🌧

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://10.0.40.14:8501

  For better performance, install the Watchdog module:

  $ xcode-select --install
  $ pip install watchdog
```

11. Conclusiones y mejoras futuras

11.1 Conclusiones

En este proyecto se ha desarrollado con éxito un **sistema completo de predicción meteorológica** basado en una arquitectura Big Data.

El sistema integra todas las fases necesarias para el tratamiento de datos:

- Ingesta desde una fuente oficial.
- Procesamiento y normalización mediante Spark.
- Almacenamiento optimizado en formato Parquet.
- Visualización interactiva mediante un cuadro de mandos.
- Integración de modelos supervisados diferenciados (clasificación/regresión) con evaluación mediante métricas y backtesting, proporcionando predicciones interpretables y coherentes a 7 días.

El resultado es un prototipo estable, escalable y fácilmente extensible, que cumple con los objetivos planteados inicialmente.

11.2 Mejoras futuras

Como posibles líneas de mejora se identifican las siguientes:

- Ampliación del histórico de datos para mejorar la calidad de las predicciones.
- Incorporación de nuevos municipios y variables meteorológicas.
- Evaluación de modelos predictivos más avanzados cuando el volumen de datos lo permita.
- Automatización completa mediante tareas programadas.
- Despliegue del sistema en un entorno cloud.