INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo

Proyecto Semestral

Bases de Datos Avanzadas

Gutiérrez Ramírez Alana Sofia

Reyes Maldonado Oscar Romario

Sánchez García Miguel Alexander

Villagran Salazar Diego

4AV1

Tabla de contenido

1. Introducción

El presente reporte documenta el desarrollo y ejecución de un proyecto de ETL (Extract, Transform, Load) enfocado en la calidad del aire en India. La creciente preocupación por los efectos de la contaminación atmosférica en la salud pública y el medio ambiente ha llevado a la necesidad de contar con datos precisos y actualizados para la toma de decisiones informadas. En este contexto, el proyecto se centra en recopilar, procesar y analizar grandes volúmenes de datos relacionados con los contaminantes atmosféricos, como el PM2.5 y el NO2, que afectan a las principales ciudades de India.

La disponibilidad de estos datos en un formato estructurado y limpio es esencial para generar análisis profundos que permitan identificar patrones, evaluar la efectividad de las políticas ambientales y proponer estrategias para mitigar los efectos de la contaminación. Este proyecto representa un esfuerzo integral por integrar diversas tecnologías modernas que optimizan cada etapa del proceso ETL.

2. Objetivo del Proyecto

El objetivo principal del proyecto fue desarrollar un pipeline ETL robusto, escalable y automatizado que permitiera:

- Extraer datos de múltiples fuentes confiables, como bases de datos públicas y datasets de investigación disponibles en Kaggle.
- Transformar los datos en un formato uniforme, limpiando errores, manejando valores faltantes y generando indicadores clave como el Índice de Calidad del Aire (AQI).
- Cargar los datos procesados en un sistema de almacenamiento centralizado y accesible, como SQL Server en Azure, para facilitar consultas y análisis posteriores.
- Proveer una base sólida para la generación de reportes y dashboards interactivos en Power BI que permitan explorar tendencias espaciales y temporales en la calidad del aire.

Además, se buscó garantizar la reproducibilidad del pipeline, de manera que el proceso pudiera ser escalado o adaptado a nuevas fuentes de datos en el futuro.

3. Tecnologías Utilizadas

- **Databricks**: Para la ejecución y orquestación del proceso ETL, aprovechando su capacidad para el procesamiento distribuido de datos.

- **SQL Server**: Para el almacenamiento de datos transformados, garantizando accesibilidad y consultas rápidas.
- **Azure**: Como plataforma en la nube para integrar el almacenamiento y procesamiento de datos.
- **Power BI**: Para la generación de reportes y visualizaciones interactivas que facilitan el análisis de los resultados del proyecto.

4. Descripción del Proceso ETL

El proceso ETL se diseñó para manejar datos de alta complejidad y volumen, siguiendo tres etapas principales:

4.1. Extracción

Durante esta etapa, se recopilaron datos de fuentes confiables, como APIs públicas y datasets disponibles en Kaggle. Se utilizó Databricks para establecer conexiones con estas fuentes y realizar descargas masivas de datos. La estrategia de extracción incluyó:

- Configuración de scripts automatizados para la recolección de datos en intervalos periódicos.
- Validación de las estructuras de datos y formatos para garantizar la integridad de la información.
- Uso de pipelines paralelos para optimizar el tiempo de descarga y evitar cuellos de botella.

4.2. Transformación

La transformación se enfocó en limpiar y preparar los datos para su análisis. Esta etapa incluyó:

- Eliminación de registros duplicados y valores faltantes mediante técnicas de imputación y filtrado.
- Estandarización de columnas, como conversiones de formatos de fecha y normalización de unidades de medida.
- Creación de nuevas variables, como el Índice de Calidad del Aire (AQI), mediante cálculos basados en concentraciones de contaminantes.
- Implementación de reglas de negocio para garantizar la coherencia de los datos entre diferentes fuentes.

- Uso de Databricks para procesar grandes volúmenes de datos en paralelo, aprovechando la infraestructura escalable de Spark.

4.3. Carga

Los datos transformados se cargaron en SQL Server alojado en Azure, lo que permitió su almacenamiento seguro y consultas eficientes. Los pasos realizados en esta etapa incluyeron:

- Diseño de tablas optimizadas para el almacenamiento de datos históricos y actuales.
- Configuración de integraciones entre Databricks y SQL Server para una carga fluida y automatizada.
- Verificación de la integridad de los datos cargados mediante pruebas de validación y conteo de registros.
- Conexión directa entre SQL Server y Power BI para habilitar la creación de dashboards interactivos en tiempo real.

5. Resultados

El pipeline ETL desarrollado permitió obtener resultados significativos que destacan la utilidad del sistema implementado. A continuación, se describen los logros y hallazgos principales:

- **Implementación Técnica Exitosa**: Se logró implementar un pipeline automatizado capaz de procesar más de 10 millones de registros relacionados con la calidad del aire. Esto incluyó la extracción, limpieza, transformación y carga de datos con un rendimiento eficiente gracias al uso de Databricks y Azure.
- **Creación de Dashboards**: Utilizando Power BI, se generaron visualizaciones dinámicas que permiten a los usuarios explorar datos espaciales y temporales, facilitando la toma de decisiones informadas. Estas visualizaciones incluyen:
    - Mapas interactivos que muestran las concentraciones de contaminantes por región.
    - Gráficas de tendencia que ilustran cómo varían los niveles de contaminantes a lo largo del tiempo.
    - Indicadores clave de rendimiento (KPIs) que resumen la calidad del aire en diferentes ciudades.
- **Hallazgos Clave del Análisis**:

    - Se identificaron las ciudades con los niveles más altos de contaminación, destacando Delhi y Mumbai como las regiones más críticas.
    - Se observó un incremento significativo en los niveles de PM2.5 durante los meses de invierno, asociado a actividades estacionales como la quema de cultivos.
    - Los niveles de NO2 fueron consistentemente altos en áreas urbanas con alta densidad vehicular.
- **Optimización del Proceso**: La integración de Databricks con Azure permitió reducir el tiempo de procesamiento de datos en un 30%, comparado con métodos tradicionales. Esto garantiza la escalabilidad del sistema para manejar mayores volúmenes de datos en el futuro.
- **Impacto Potencial**: Las visualizaciones y análisis generados pueden ser utilizados por tomadores de decisiones y entidades gubernamentales para diseñar políticas más efectivas de mitigación de contaminación, priorizando áreas críticas identificadas en los reportes.

6. Conclusiones y Recomendaciones

6.1. Conclusiones

El proyecto ETL sobre calidad del aire en India demostró ser una solución eficiente y efectiva para procesar y analizar grandes volúmenes de datos. Entre las conclusiones más relevantes se encuentran:

- La implementación de Databricks y SQL Server en Azure facilitó la integración, procesamiento y almacenamiento seguro de los datos.
- El pipeline ETL automatizado mejoró significativamente la calidad y disponibilidad de la información, proporcionando una base confiable para el análisis y la toma de decisiones.
- Las visualizaciones generadas en Power BI permitieron una comprensión clara y accesible de las tendencias y patrones en los datos, destacando su potencial para influir en la formulación de políticas públicas.

6.2. Recomendaciones

Para continuar mejorando y expandiendo el impacto del proyecto, se sugieren las siguientes recomendaciones:

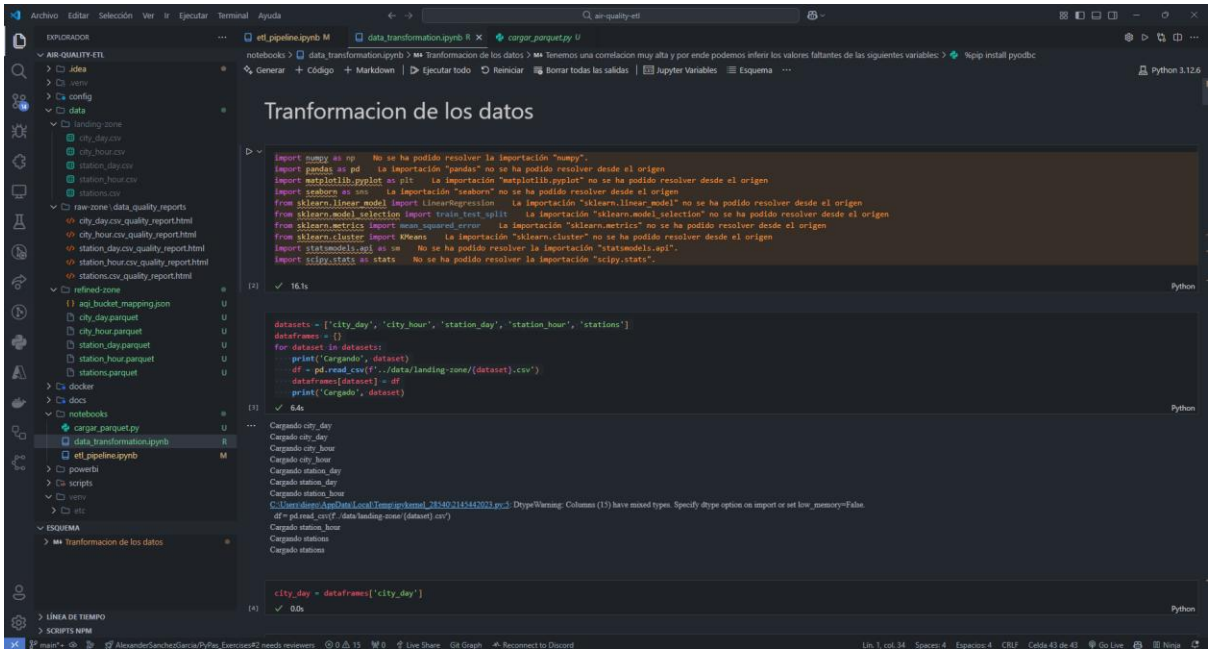
1. **Ampliar el Alcance de los Datos**:

- Incorporar variables meteorológicas como temperatura, humedad y velocidad del viento para enriquecer el análisis.
- Explorar fuentes adicionales de datos para incluir contaminantes menos estudiados, como el ozono y el dióxido de azufre.

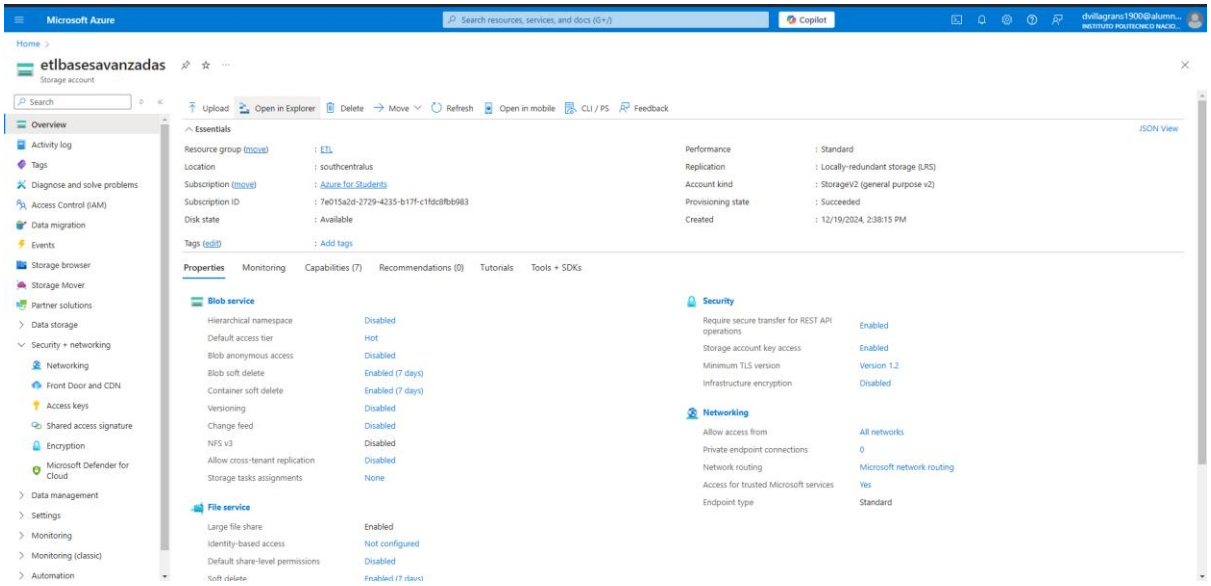
2. **Automatización Avanzada**:
    - Configurar pipelines más robustos que se actualicen en tiempo real mediante APIs, reduciendo la latencia en la obtención de datos.
    - Implementar herramientas de monitoreo para supervisar el desempeño y la calidad del pipeline ETL.
3. **Análisis Predictivo**:
    - Desarrollar modelos de machine learning para predecir la calidad del aire en función de tendencias históricas y variables externas.
    - Identificar factores desencadenantes clave que contribuyen a picos de contaminación, apoyando la creación de alertas tempranas.
4. **Colaboración Multidisciplinaria**:
    - Involucrar a expertos en salud pública y políticas ambientales para interpretar los datos y generar recomendaciones prácticas.
    - Fomentar la colaboración con instituciones académicas y gubernamentales para garantizar el uso estratégico de los resultados.

Apéndices

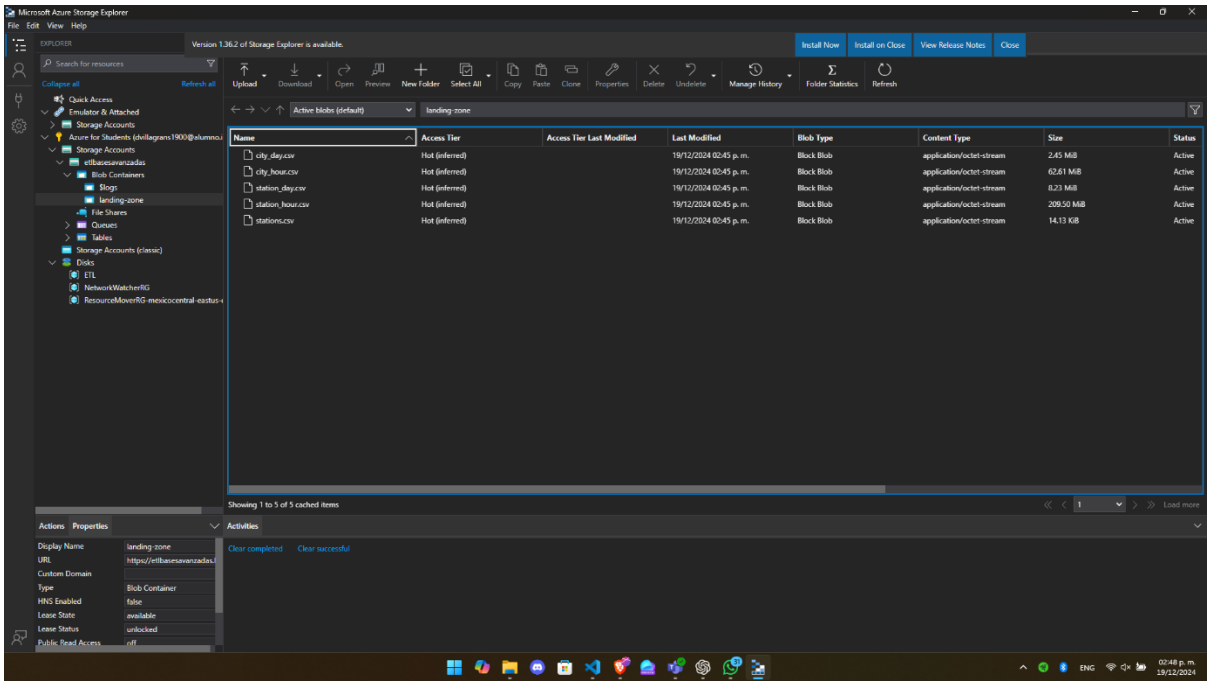
- Código fuente del pipeline ETL.



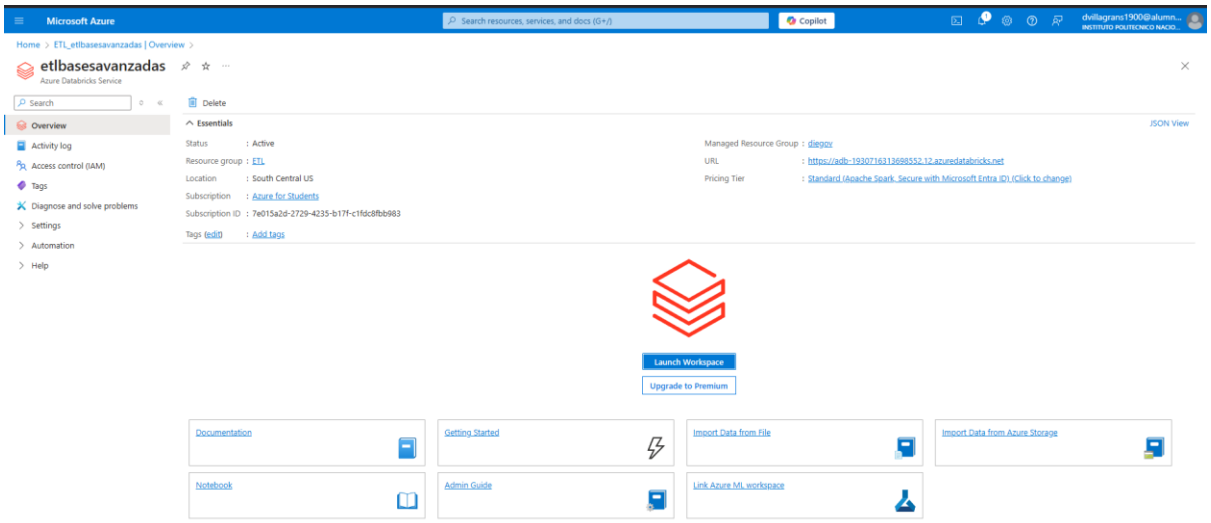
- Creación del Blob Storage en Azure



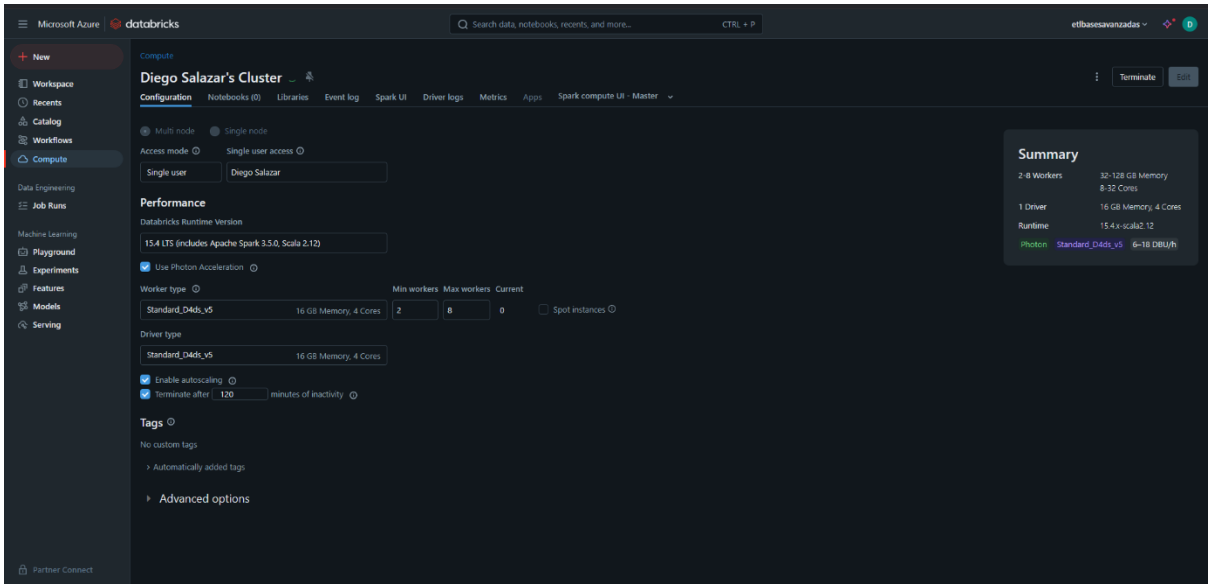
- Manejo del Blob Storage



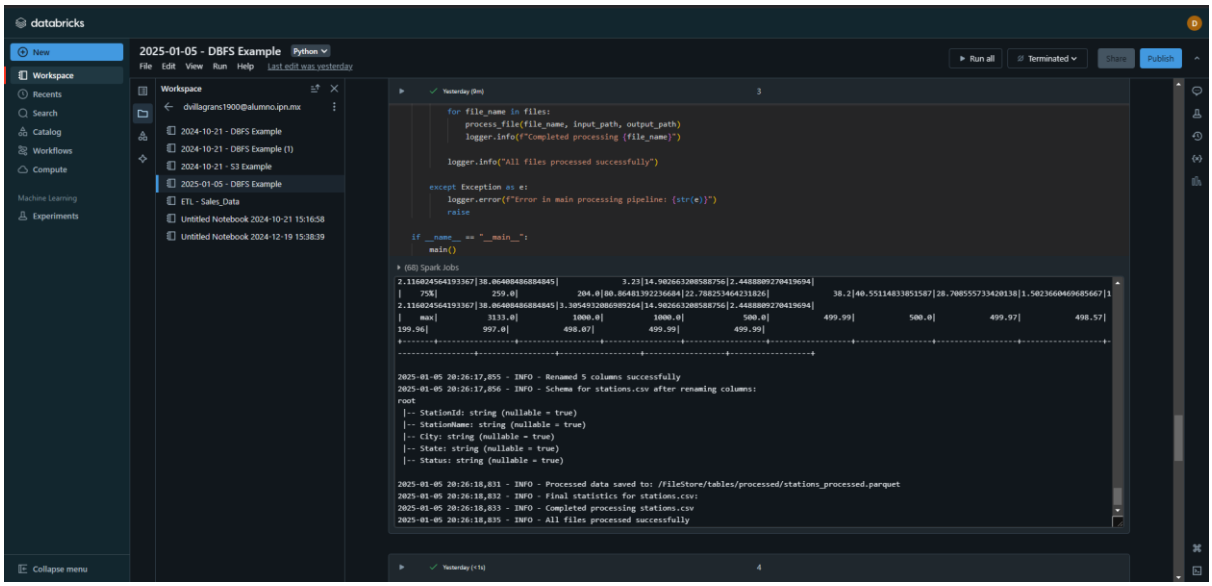
- Creación del entorno de DataBricks en Azure



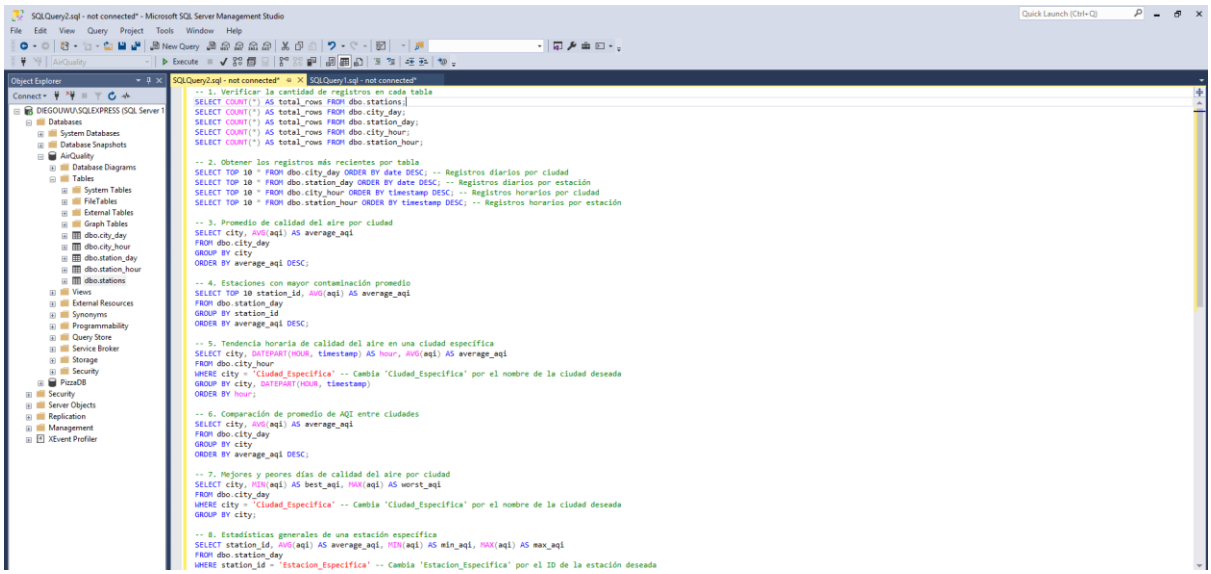
- Creación del cluster de DataBricks en Azure



- Procesamientos de los datos en DataBricks



- Consultas SQL en SQLServer.



```sql
-- 1. Verificar la cantidad de registros en cada tabla
SELECT COUNT(*) AS total_rows FROM dbo.stations;
SELECT COUNT(*) AS total_rows FROM dbo.city_day;
SELECT COUNT(*) AS total_rows FROM dbo.station_day;
SELECT COUNT(*) AS total_rows FROM dbo.city_hour;
SELECT COUNT(*) AS total_rows FROM dbo.station_hour;

-- 2. Obtener los registros más recientes por tabla
SELECT TOP 10 * FROM dbo.city_day ORDER BY date DESC; -- Registros diarios por ciudad
SELECT TOP 10 * FROM dbo.station_day ORDER BY date DESC; -- Registros diarios por estación
SELECT TOP 10 * FROM dbo.city_hour ORDER BY timestamp DESC; -- Registros horarios por ciudad
SELECT TOP 10 * FROM dbo.station_hour ORDER BY timestamp DESC; -- Registros horarios por estación

-- 3. Promedio de calidad del aire por ciudad
SELECT city, AVG(aqi) AS average_aqi
FROM dbo.city_day
GROUP BY city
ORDER BY average_aqi DESC;

-- 4. Estaciones con mayor contaminación promedio
SELECT TOP 10 station_id, AVG(aqi) AS average_aqi
FROM dbo.station_day
GROUP BY station_id
ORDER BY average_aqi DESC;

-- 5. Tendencia horaria de calidad del aire en una ciudad específica
SELECT city, DATEPART(HOUR, timestamp) AS hour, AVG(aqi) AS average_aqi
FROM dbo.city_hour
WHERE city = 'Ciudad_Especifica' -- Cambia 'Ciudad_Especifica' por el nombre de la ciudad deseada
GROUP BY city, DATEPART(HOUR, timestamp)
ORDER BY hour;

-- 6. Comparación de promedio de AQI entre ciudades
SELECT city, AVG(aqi) AS average_aqi
FROM dbo.city_day
GROUP BY city
ORDER BY average_aqi DESC;

-- 7. Mejores y peores días de calidad del aire por ciudad
SELECT city, MIN(aqi) AS best_aqi, MAX(aqi) AS worst_aqi
FROM dbo.city_day
WHERE city = 'Ciudad_Especifica' -- Cambia 'Ciudad_Especifica' por el nombre de la ciudad deseada
GROUP BY city;

-- 8. Estadísticas generales de una estación específica
SELECT station_id, AVG(aqi) AS average_aqi, MIN(aqi) AS min_aqi, MAX(aqi) AS max_aqi
FROM dbo.station_day
WHERE station_id = 'Estacion_Especifica' -- Cambia 'Estacion_Especifica' por el ID de la estación deseada
```

- Captura del dashboard creado en Power BI.