INSTITUTO POLITECNICO NACIONAL.

Escuela Superior de Cómputo.

Licenciatura en Ciencia de Datos.

Bases de Datos Avanzadas.

Grupo: 4AV1

Profesor: García Floriano Andrés.

Alumnos:

Cano Portugal Israel Daniel Arturo.

Gallegos Sánchez Celso Alberto.

García Venegas Manuel.

Liceaga Cardoso Ángel David.

Montero Marín Andrea Jaqueline.

Proyecto Semestral:
Designing an ETL Process for Air Quality Data in India.

Introducción.

En este proyecto se aborda el diseño e implementación de un proceso ETL (Extract, Transform, Load) para procesar datos relacionados con la calidad del aire en India. Los datos seleccionados serán extraídos, transformados y cargados en un sistema estructurado que permita su análisis eficiente y su integración en herramientas de visualización.

El propósito principal de este proyecto es demostrar el flujo completo de un proceso ETL, optimizando los datos para su almacenamiento y análisis, y proporcionando información valiosa sobre los niveles de contaminación en distintas regiones, estaciones y horarios.

El trabajo incluye:

- Extracción de los datos desde archivos CSV proporcionados.
- Perfilamiento y análisis de calidad de los datos para detectar valores faltantes y otras inconsistencias.
- Transformación de los datos al formato Parquet, garantizando un almacenamiento más eficiente.
- Carga de los datos a una base de datos SQL para su integración y análisis avanzado.
- Visualización de los resultados mediante dashboards interactivos en Power BI.

Objetivos.

El objetivo principal del proyecto es diseñar e implementar un proceso ETL eficiente para procesar datos de calidad del aire en India, optimizando su preparación para análisis posteriores. Para lograrlo, se establecen los siguientes objetivos específicos:

1. **Extraer y organizar los datos:**
   - Descargar los datasets originales y organizarlos en una estructura de carpetas adecuada.
   - Identificar las características principales de los datos crudos.

2. **Evaluar la calidad de los datos:**
   - Aplicar técnicas de perfilamiento para analizar y documentar inconsistencias.
   - Generar informes que reflejen el estado inicial de los datos.

3. **Transformar los datos para almacenamiento eficiente:**
   - Aplicar reglas de negocio para manejar valores nulos y otras inconsistencias.
   - Convertir los datos al formato Parquet para optimizar su uso en sistemas de almacenamiento y análisis.

4. **Carga de Datos:**
   - Integrar los datos procesados en una base de datos SQL para su análisis estructurado.

5. **Visualización:**
   - Crear dashboards en Power BI que representen los datos procesados de manera visual y accesible

Estructura del Notebook.

El notebook está organizado en las siguientes secciones:

1. **Extracción de Datos:**
   - Los datos se descargan y organizan en una estructura de carpetas inicial (landing-zone).

2. **Perfilamiento de Datos:**
   - Se realiza un análisis inicial de los datos para detectar valores faltantes, distribuciones y otras características relevantes. Los resultados se documentan en informes generados automáticamente.

3. **Transformación de Datos:**
   - Se aplican reglas para limpiar y transformar los datos. Además, se convierten los archivos CSV originales al formato Parquet para almacenamiento eficiente.

Descripción de las etapas del proceso ETL.

1. **Extracción de Datos**

Se implementó un script en Python para automatizar la organización inicial de los archivos. El código utiliza las siguientes librerías:

- os y pathlib para la manipulación de directorios y archivos.

El script realiza las siguientes acciones:

1. Creación de la carpeta landing-zone:

```
//landing_zone = Path("/content/landing-zone")

//landing_zone.mkdir(exist_ok=True)
```

2. Organización de los archivos CSV: Se verifica la existencia de cada archivo y se mueve a la carpeta landing-zone.

```
//uploaded_files = ["stations.csv", "station_hour.csv", "station_day.csv", "city_hour.csv", "city_day.csv"]

for file_name in uploaded_files:
    source_path = root_folder / file_name
    destination_path = landing_zone / file_name
    if destination_path.exists():
```

```
        print(f"El archivo ya existe en landing-zone: {file_name}")
    elif source_path.exists():
        os.rename(source_path, destination_path)
        print(f"Archivo movido a landing-zone: {file_name}")
    else:
        print(f"Archivo no encontrado: {file_name}")
```

2. **Perfilamiento y Análisis de Calidad de Datos**

Para esta etapa, se utilizó la librería ydata-profiling (anteriormente conocida como pandas-profiling) para generar reportes de calidad automáticos en formato HTML.

Las acciones incluyen:

1. Creación de directorios para los reportes:

```
//raw_zone = Path("/content/raw-zone")
data_quality_reports  =  raw_zone/"data_quality_reports"
raw_zone.mkdir(parents = True,exist_ok = True)
data_quality_reports.mkdir(parents = True, exist_ok = True)
```

2. Generación de reportes de calidad: Cada archivo CSV en la carpeta landing-zone se analiza y se genera un reporte.

```
//for file in landing_zone.glob("*.csv"):
df = pd.read_csv(file)
if not df.empty:
        report = ProfileReport(
        df,
        title=f"Reporte de Calidad para {file.name}",
        explorative=True
)
        report_path  =  data_quality_reports  /  f"{file.stem}_quality_report.html"
        report.to_file(report_path)
        print(f"Reporte de calidad generado: {report_path}")
else:
        print(f"El archivo {file.name} está vacío.")
```

Flujo de Datos.

1. **Landing-Zone:**
   - Almacena los archivos CSV originales descargados.

2. **Raw-Zone:**
   - Almacena los datos transformados y reportes generados.

3. **Refined-Zone:**
   - Contendrá los datos procesados listos para la carga en una base de datos SQL.

Desafíos y Soluciones.

1. **Valores Faltantes:**
   - Desafío: Gran cantidad de valores nulos en algunos datasets.
   - Solución: Diseñar reglas de negocio para reemplazo eficiente (por ejemplo, media o mediana).

2. **Tamaño de los Datos:**
   - Desafío: Manejo de grandes volúmenes de datos.
   - Solución: Uso de herramientas como ydata-profiling y almacenamiento en formato Parquet.

3. **Organización de Directorios:**
   - Desafío: Creación y gestión de una estructura de carpetas adecuada.
   - Solución: Automatización mediante scripts en Python.

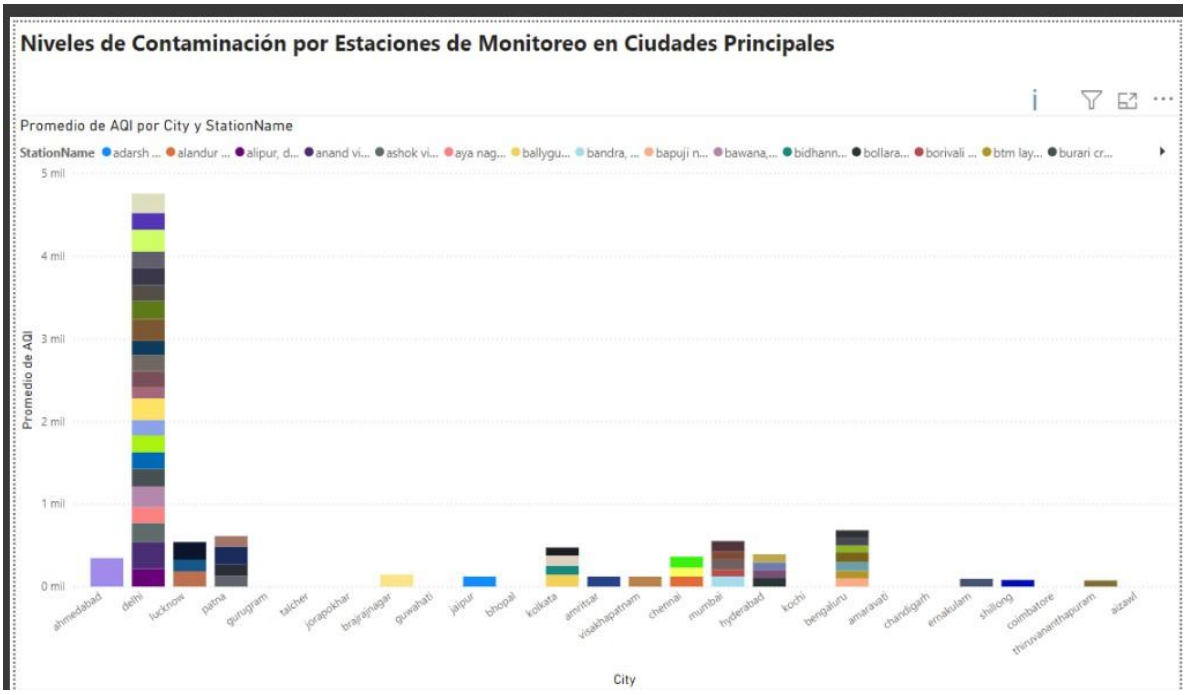
Estos reportes incluyen:

- Estadísticas descriptivas.
- Análisis de valores faltantes.
- Distribuciones y outliers.

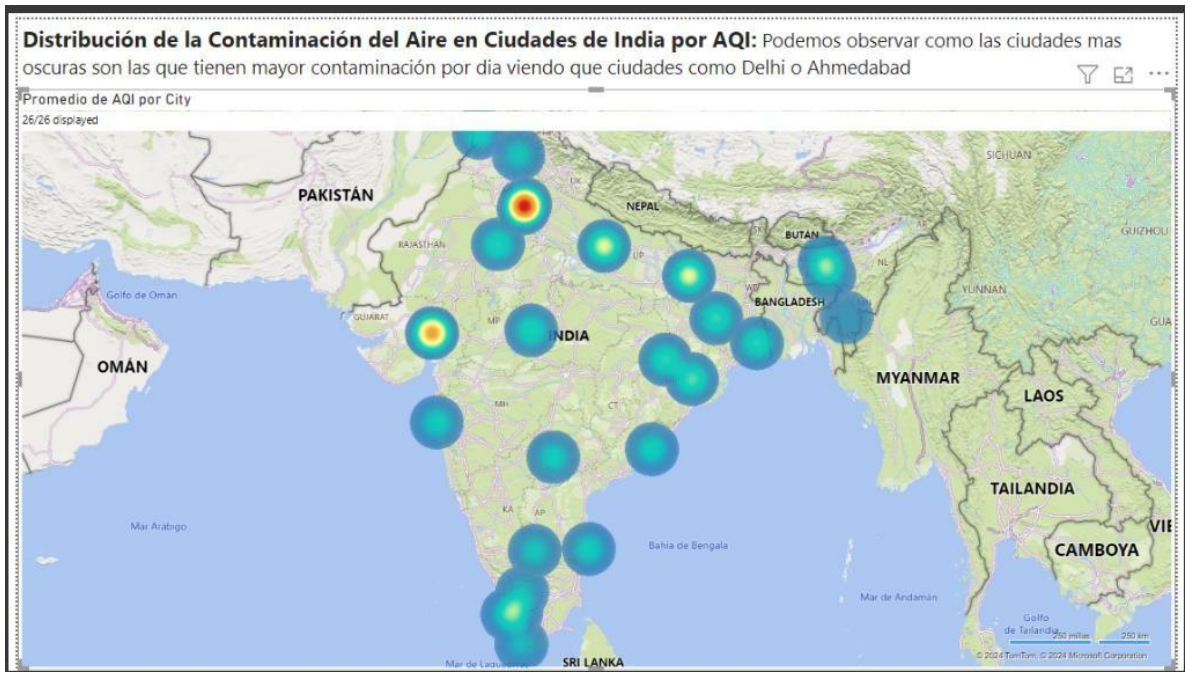
Resultados.

- Los archivos CSV fueron correctamente organizados en la carpeta landing-zone.
- Se generaron reportes de calidad detallados en formato HTML, almacenados en raw-zone/data_quality_reports. Esto facilitó la identificación y solución de problemas en los datos antes de proceder a las siguientes etapas del proceso ETL.

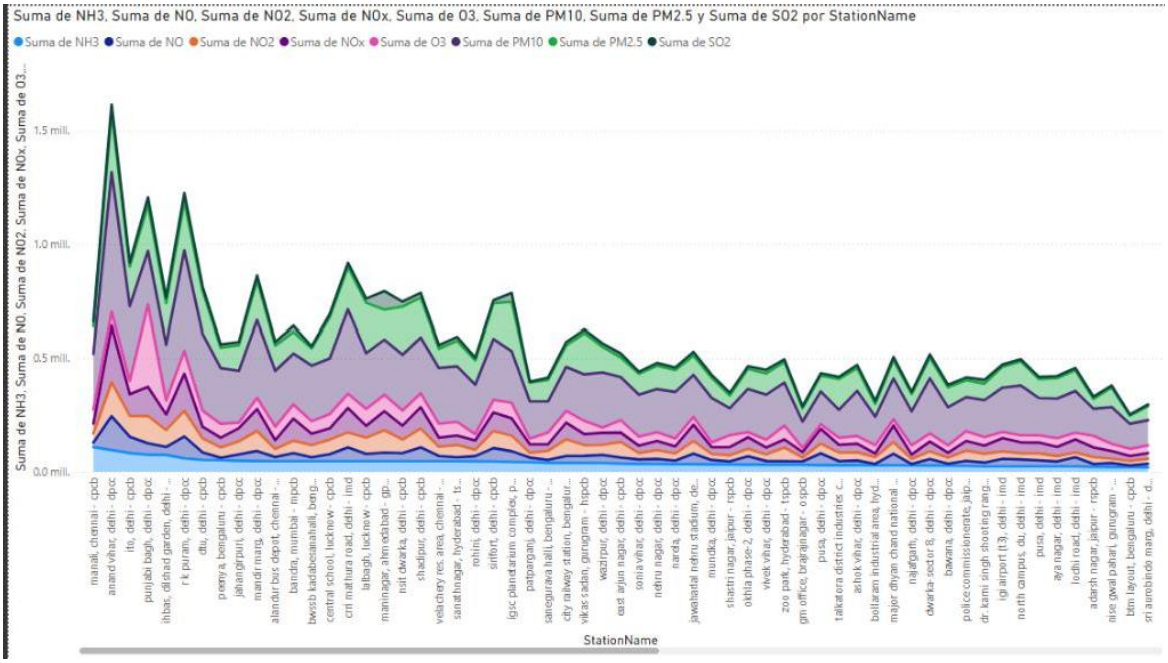
Distribución de la Contaminación del Aire en Ciudades de India por AQI: Mirando el mapa y los datos, podemos ver que las ciudades de India tienen diferentes niveles de contaminación. Ahmedabad es la ciudad más contaminada con un nivel de 339.86, seguida por Delhi con 258.78. La mayoría de las ciudades muy contaminadas están en el norte de India (se ven como puntos rojos y naranjas en el mapa). Por otro lado, las ciudades del sur y las que están cerca del mar tienen menos contaminación, como Aizawl que tiene el nivel más bajo con 36.24, y Shillong con 75.54. Es interesante ver cómo las ciudades grandes e industriales suelen tener más contaminación, mientras que las ciudades más pequeñas o costeras tienen un aire más limpio.



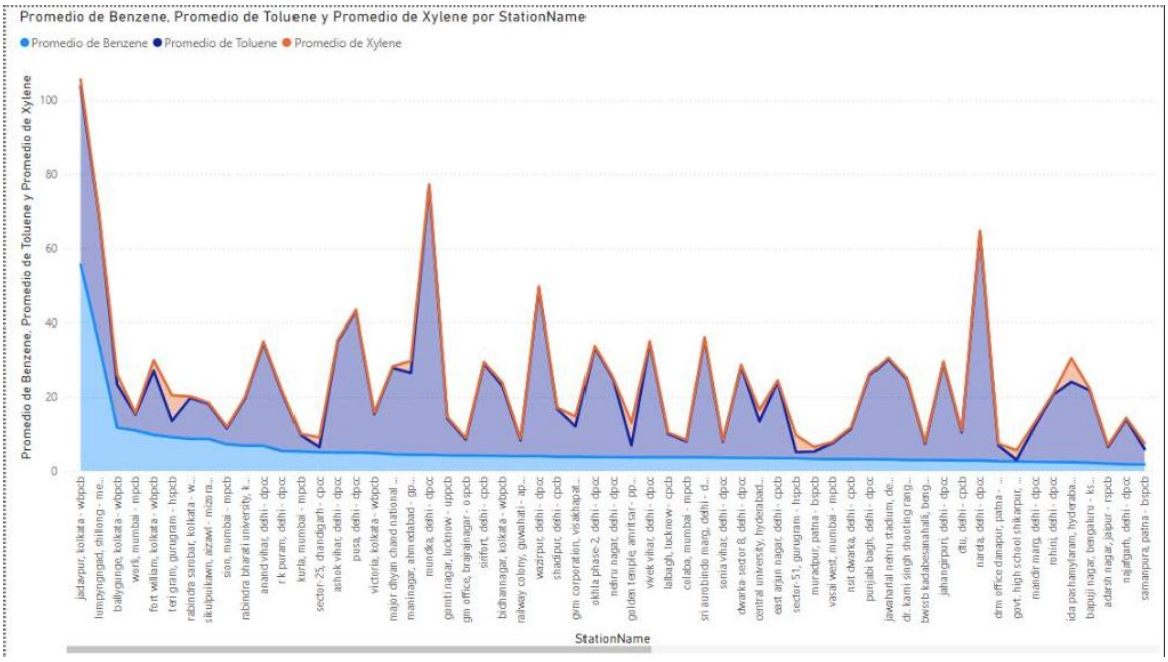
Niveles de Contaminación por Estaciones de Monitoreo en Ciudades Principales: Observando la nueva gráfica, podemos ver un análisis más detallado de las estaciones de medición de contaminación en cada ciudad de India. El gráfico de barras apiladas muestra diferentes estaciones de monitoreo (StationName) dentro de cada ciudad, donde cada color representa una estación diferente. Delhi destaca por tener múltiples estaciones con lecturas altas, como Anand Vihar (316.90), Ashok Vihar (237.76) y Aya Nagar (188.98), lo que explica por qué tiene un promedio general tan alto. En otras ciudades como Jaipur, vemos lecturas más bajas como la estación Adarsh Nagar con 121.21. Chennai muestra una lectura de 120.32 en la estación de Alandur Bus Depot. Esta visualización nos ayuda a entender que la contaminación no solo varía entre ciudades, sino también entre diferentes zonas dentro de una misma ciudad.



Concentración de Contaminantes Principales por Estación de Monitoreo (NH3, NOx, PM10, O3, SO2): En general, podemos concluir que la contaminación del aire en las ciudades de India varía mucho según la ubicación de las estaciones de monitoreo, donde la estación Manali en Chennai y Anand Vihar en Delhi muestran los niveles más preocupantes de contaminantes. El amoníaco (NH3), las partículas gruesas (PM10) y los óxidos de nitrógeno (NOx) son los contaminantes más abundantes en todas las estaciones, siendo especialmente altos en las zonas industriales y urbanas densamente pobladas. Esta información sugiere que las principales fuentes de contaminación podrían estar relacionadas con la actividad industrial, el tráfico vehicular y la densidad poblacional, lo que indica dónde podrían enfocarse las medidas de control de la contaminación.



Niveles de Compuestos BTX (Benceno, Tolueno, Xileno) por Estación: En este gráfico se muestra la concentración de tres contaminantes químicos: Benceno, Tolueno y Xileno (conocidos como BTX) en diferentes estaciones de monitoreo. Lo más destacable es que hay algunos picos muy marcados donde estos tres contaminantes alcanzan niveles muy altos, llegando a más de 100 unidades en la primera estación y con otros picos importantes de alrededor de 60-80 unidades en varias estaciones. Los tres contaminantes suelen seguir patrones similares, subiendo y bajando juntos, lo que sugiere que probablemente provienen de las mismas fuentes de contaminación, como podrían ser emisiones industriales o vehiculares. La mayoría de las estaciones mantienen niveles más bajos, entre 10 y 30 unidades, pero los picos ocasionales son preocupantes desde el punto de vista de la calidad del aire.



Resultados.

- Automatización y Eficiencia en el Proceso ETL

El diseño e implementación del proceso ETL permitió transformar datos desorganizados en información estructurada y lista para su análisis. Las etapas de extracción, transformación y carga se realizaron de forma eficiente gracias a la automatización con scripts en Python, optimizando el manejo de directorios y la conversión de formatos de datos como Parquet.

- Calidad de los Datos y Resolución de Problemas

La aplicación de técnicas de perfilamiento de datos permitió identificar inconsistencias como valores faltantes, lo que llevó a la implementación de reglas de negocio para su corrección. Esto resultó en una mejora significativa en la calidad de los datos procesados, habilitando un análisis más confiable.

- Optimización del Almacenamiento y Análisis

La conversión de datos al formato Parquet y su integración en una base de datos SQL redujeron el tiempo de procesamiento y facilitaron el análisis avanzado. Estas medidas son clave para manejar grandes volúmenes de información de manera eficiente.

- Impacto Visual y Comunicativo

Los dashboards interactivos desarrollados en Power BI proporcionaron una representación clara y accesible de los resultados del análisis, lo que mejora la comprensión de la calidad del aire en India y permite a los usuarios finales tomar decisiones informadas.

- Superación de Desafíos

Los principales desafíos, como el manejo de valores nulos y grandes volúmenes de datos, fueron abordados exitosamente mediante estrategias técnicas y herramientas adecuadas. Esto refuerza la capacidad del equipo para gestionar proyectos de ciencia de datos complejos.

- Aplicabilidad del Proyecto

Este proyecto establece una base para futuros análisis relacionados con la calidad del aire y otras áreas de investigación que requieran procesamiento de datos masivos. Además, proporciona un modelo replicable para implementar procesos ETL en otros contextos.

Referencias.

- Amazon Web Services, "¿Qué es ETL?," [en línea]. Disponible: https://aws.amazon.com/es/what-is/etl/. [Accedido: 03-dic-2025].

- Microsoft, "Consulta y uso del formato Parquet en Azure Databricks," [en línea]. Disponible: https://learn.microsoft.com/es-es/azure/databricks/query/formats/parquet. [Accedido: 08-dic-2025].

- YData, "ydata-profiling Documentation," [en línea]. Disponible: https://docs.profiling.ydata.ai/latest/. [Accedido: 13-ene-2025].

- Ranga, "What is ETL? Extract Transform and Load explained in detail," YouTube, 9 junio 2022. [Video]. Disponible: https://www.youtube.com/watch?v=lAebJGKPRdc. [Accedido: 16-dic-2025].

- Python Software Foundation, "Welcome to Python.org," [en línea]. Disponible: https://www.python.org/. [Accedido: 16-dic-2025].