PROJECT DELIVERY

MARIA FERNANDA TELLO VERGARA
ANA CRISTINA QUINTERO CARPINTERO
JOHAN EDELBERTO HURTADO ENRIQUEZ

TEACHER: JAVIER ALEJANDRO VERGARA

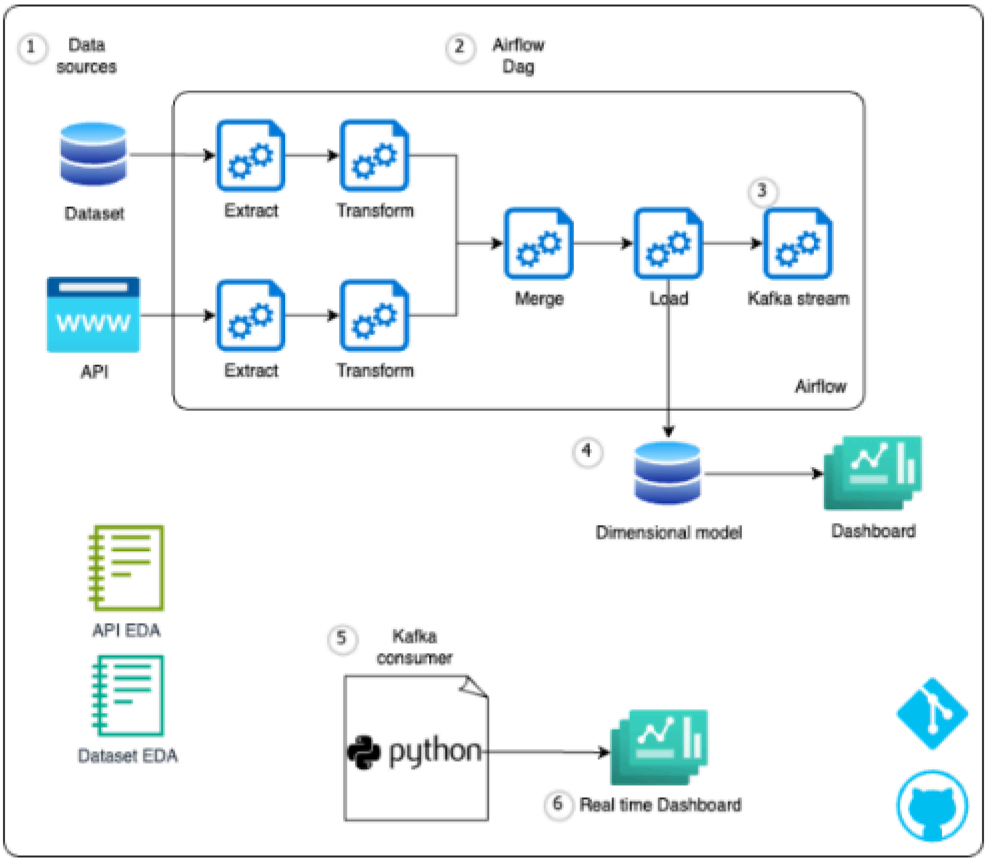
ELT

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE

SANTIAGO DE CALI
13 NOVEMBER
2024

INTRODUCTION

The final delivery of the project course has six steps (as in the figure) described as follows:
1. Data sources: two different data sources selected in previous deliveries, the first one from a dataset, the second one from an API.
2. Airflow DAG: A DAG with different tasks to extract,. transform and load, the load task has to store the dimensional model into a database.
3. Streaming: A streaming Kafka task to stream a metric from the Fact table
4. Dashboard: Get the data from the dimensional model created in the Airflow ETL DAG and create a dashboard in your preferred tool (Power BI , Looker Studio).
5. Kafka Consumer: In a python APP, instance the Kafka consumer to get the metric from the step 3.
6. Real time dashboard: Connect the visualization tool you selected to the python app with the Kafka consumer and create a real time dashboard to visualize the streaming of the data.



TECHNOLOGIES USED

- Python: Primary language for developing ETL scripts and implementing the Kafka consumer.
- Jupyter Notebook: For exploratory data analysis (EDA) performed in previous stages of the project.
- Database: [Specify the chosen database], used to store the dimensional model.
- Apache Airflow: Orchestration tool for the ETL workflow.
- Apache Kafka: Real-time data streaming and messaging platform.
- CSV and API: Data sources used in the project.
- Visualization Tool (Power BI or Looker Studio): For creating dashboards.
- Git/GitHub: Version control and repository to store code, EDA files, and project evidence.

DATASET CONTEXT

This project focuses on the analysis of a health dataset that includes detailed information on 1,879 patients. Each patient is uniquely identified with an ID between 6000 and 7878. The dataset covers demographic details, lifestyle factors, medical history, clinical measurements, medication use, symptoms, quality of life scores, environmental exposures, and health behaviors.

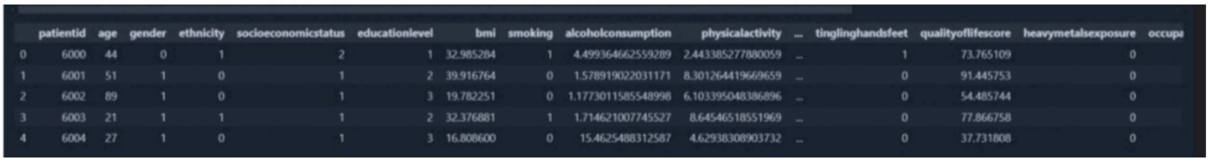
The analysis is performed with the aim of extracting valuable information about health patterns,treatment effectiveness, and factors affecting patients' quality of life. Confidentiality and privacy of information is ensured by associating each patient with a treating physician, whose name is kept confidential.

GOALS

The aim of this project is to apply advanced data analysis techniques to improve understanding of the factors that influence patients' health, optimize medical treatments, and promote a better quality of life. In addition, the project seeks to ensure proper and secure handling of patients' confidential information through the use of appropriate encryption and data storage technologies.

DESCRIPTION

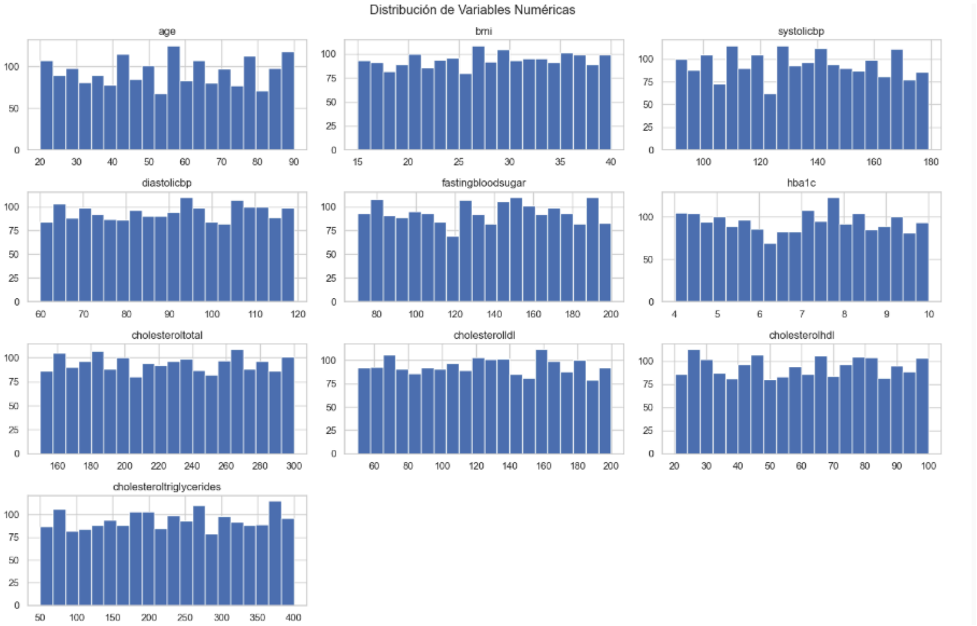
This project involves the comprehensive analysis of a health dataset comprising detailed information on 1,879 patients, each uniquely identified with an ID in the range of 6000 to 7878. The dataset includes a wide variety of information, such as demographic details (age, sex, ethnicity), lifestyle factors (dietary habits, physical activity), medical history (previous illnesses, family history), clinical measurements (blood pressure, glucose levels), medication use (current and previous prescriptions), reported symptoms, quality of life scores (measures of physical and mental well-being), environmental exposures (pollution levels, exposure to toxicants), and health behaviors (smoking, alcohol consumption).



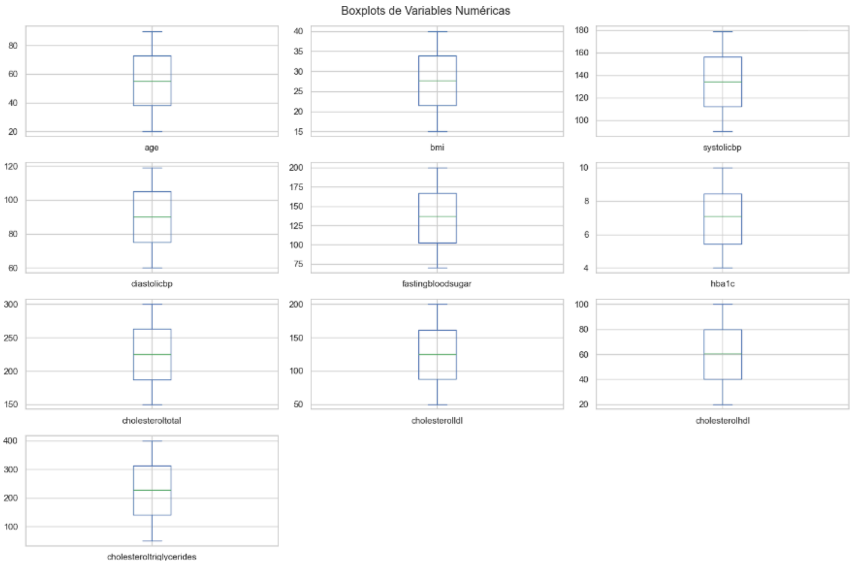
	patientid	age	gender	ethnicity	socioeconomicstatus	educationlevel	bmi	smoking	alcoholconsumption	physicalactivity	...	tinglinghandsfeet	qualityoflifescore	heavymetalsexposure	occupa
0	6000	44	0	1	2	1	32.985284	1	4.499364662559289	2.443385277880059	...	1	73.765109	0	
1	6001	51	1	0	1	2	39.916764	0	1.578919022031171	8.301264419669659	...	0	91.445753	0	
2	6002	89	1	0	1	3	19.782251	0	1.177301158548998	6.103395048386896	...	0	54.485744	0	
3	6003	21	1	1	1	2	32.376881	1	1.714621007745527	8.64546518551969	...	0	77.866758	0	
4	6004	27	1	0	1	3	16.808600	0	15.4625488312587	4.62938308903732	...	0	37.731808	0	

EDA DATASET

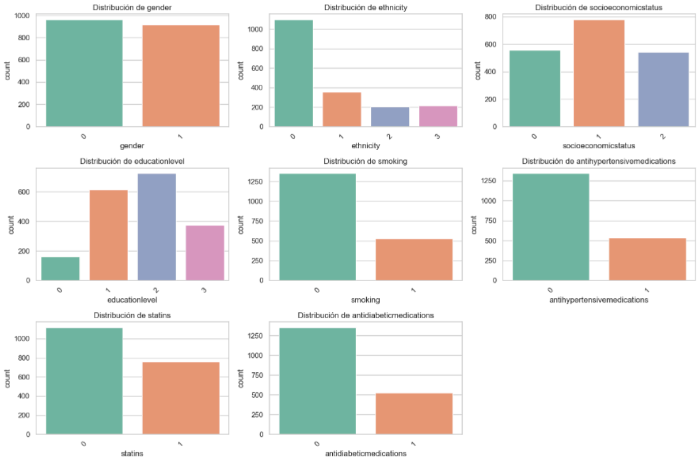
Here we can see some graphs that help us understand the exploratory analysis of the data.



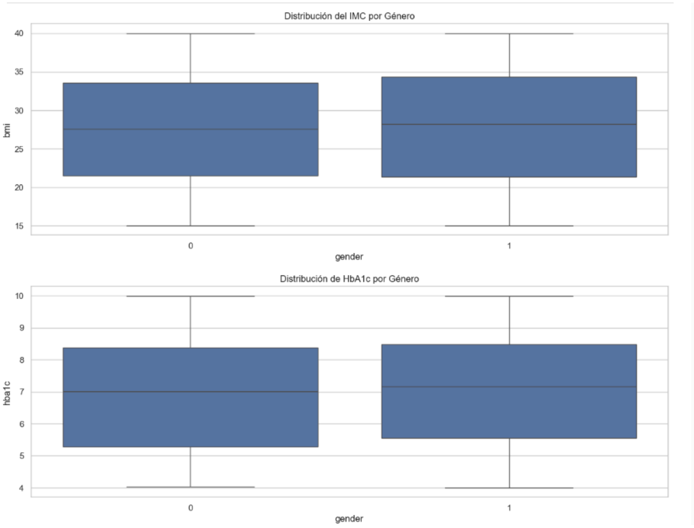
The results of the graph show the distribution of several health-related numerical variables, such as age, blood pressure, blood sugar levels, and different types of cholesterol. Each histogram reveals how the data is distributed for each variable, allowing you to identify patterns, trends, and potential anomalies. For example, we can look at the frequency of different age ranges in the sample, systolic and diastolic blood pressure levels, and HDL, LDL, and total cholesterol levels.



The results of the graph show the distribution of several numerical variables related to health, such as age, body mass index (BMI), systolic and diastolic blood pressure, testosterone levels and different types of cholesterol. Each boxplot reveals the dispersion and central trend of the data, highlighting the median, interquartile range, and possible outliers.

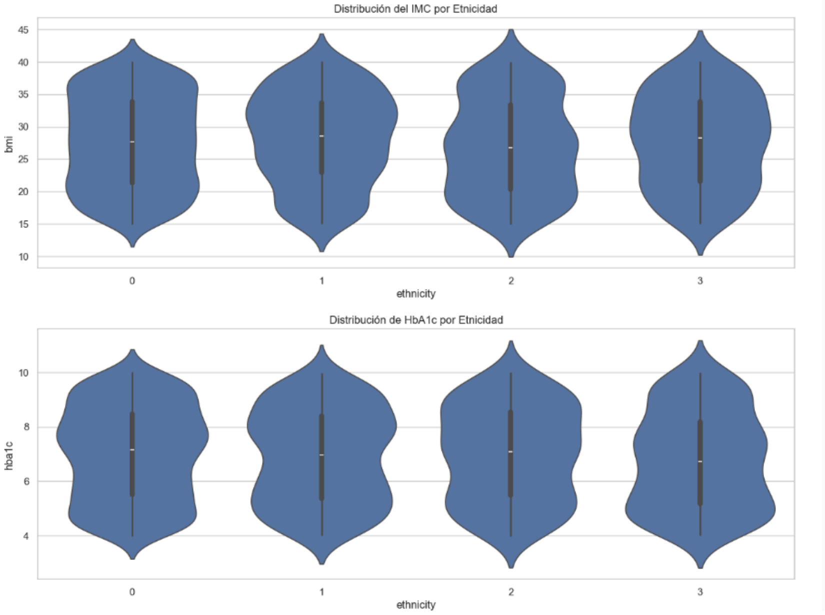


The results in the graph show the distribution of various demographic and health variables. Each bar graph represents a different variable, such as gender, ethnicity, socioeconomic status, education level, smoking status, use of antihypertensive drugs, use of statins, and use of antidiabetic drugs. The graph allow you to visualize the frequency or count of each category within these variables.



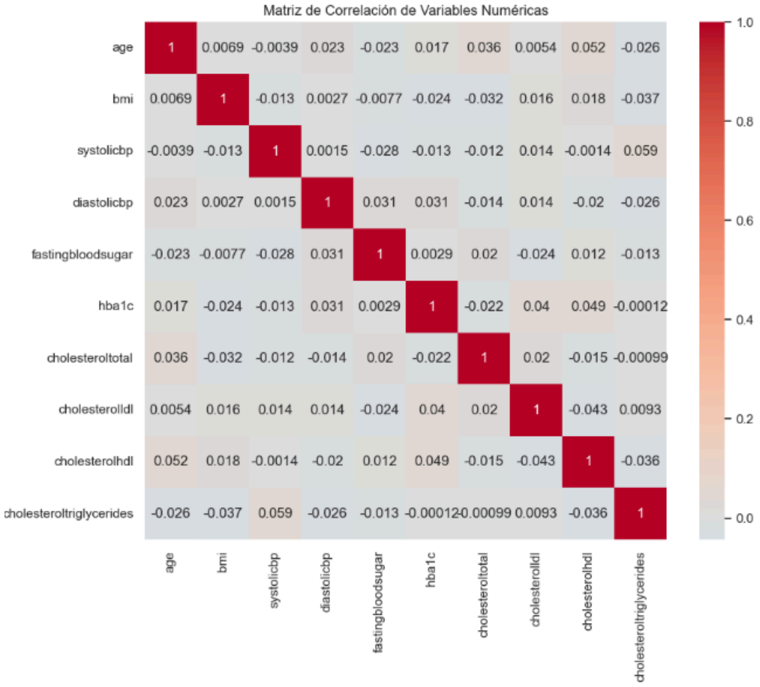
The graph shows the distribution of Body Mass Index (BMI) by gender. There are two boxplots, one for each gender, which represent the dispersion of BMI values. The box in each boxplot indicates the interquartile range (IQR), which contains the middle 50% of the data, while the line inside the box represents the median BMI. The "whiskers" extend to the furthest values that are not considered outliers, and the points outside the whiskers are outliers.

The graph shows the distribution of HbA1c levels by gender. HbA1c is a measure that reflects the average blood glucose levels over the past three months. There are two boxplots, one for each gender, which represent the dispersion of HbA1c values. The box in each boxplot indicates the interquartile range (IQR), which contains the central 50% of the data, while the line inside the box represents the median HbA1c. The "whiskers" extend to the furthest values that are not considered outliers, and the points outside the whiskers are outliers.



The graph shows the distribution of Body Mass Index (BMI) by ethnicity using a violin diagram. Each violin represents a category of ethnicity and shows the density of BMI values within each group. The width of each violin at different BMI levels indicates the frequency of the data at those levels; wider sections mean more data in that range. In addition, the median (white point) and the interquartile range (thick black bar) are marked within each violin.

The graph shows the distribution of HbA1c levels by ethnicity using a violin diagram. Each violin represents a category of ethnicity and shows the density of HbA1c values within each group. The width of each violin at different HbA1c levels indicates the frequency of the data at those levels; wider sections mean more data in that range. In addition, the median (white point) and the interquartile range (thick black bar) are marked within each violin.



Matriz de Correlación de Variables Numéricas

	age	bmi	systolicbp	diastolicbp	fastingbloodsugar	hba1c	cholesteroltotal	cholesterolldl	cholesterolhdl	cholesteroltriglycerides
age	1	0.0069	-0.0039	0.023	-0.023	0.017	0.036	0.0054	0.052	-0.026
bmi	0.0069	1	-0.013	0.0027	-0.0077	-0.024	-0.032	0.016	0.018	-0.037
systolicbp	-0.0039	-0.013	1	0.0015	-0.028	-0.013	-0.012	0.014	-0.0014	0.059
diastolicbp	0.023	0.0027	0.0015	1	0.031	0.031	-0.014	0.014	-0.02	-0.026
fastingbloodsugar	-0.023	-0.0077	-0.028	0.031	1	0.0029	0.02	-0.024	0.012	-0.013
hba1c	0.017	-0.024	-0.013	0.031	0.0029	1	-0.022	0.04	0.049	-0.00012
cholesteroltotal	0.036	-0.032	-0.012	-0.014	0.02	-0.022	1	0.02	-0.015	-0.00099
cholesterolldl	0.0054	0.016	0.014	0.014	-0.024	0.04	0.02	1	-0.043	0.0093
cholesterolhdl	0.052	0.018	-0.0014	-0.02	0.012	0.049	-0.015	-0.043	1	-0.036
cholesteroltriglycerides	-0.026	-0.037	0.059	-0.026	-0.013	-0.00012	-0.00099	0.0093	-0.036	1

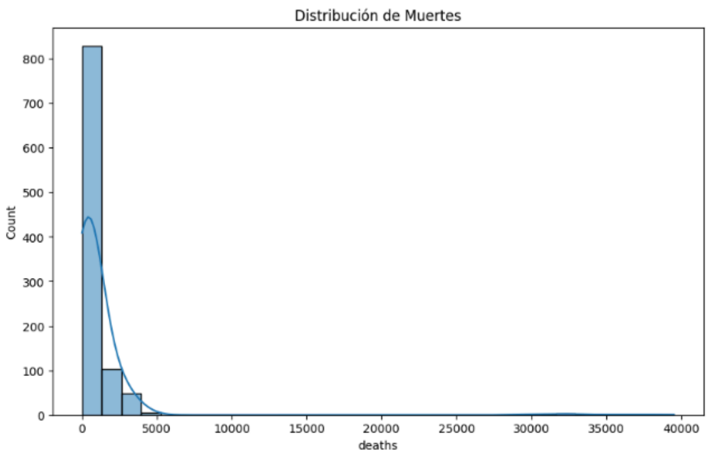
Here we can see the correlation matrix, which is a table that shows the evaluation coefficients between different variables. Each cell in the table shows the valuation between two variables. The values range from -1 to 1:

- 1 indicates a perfect positive improvement (when one variable increases, the other does too).
- -1 indicates perfect negative compensation (when one variable increases, the other decreases).
- 0 indicates that there is no offset between the variables.

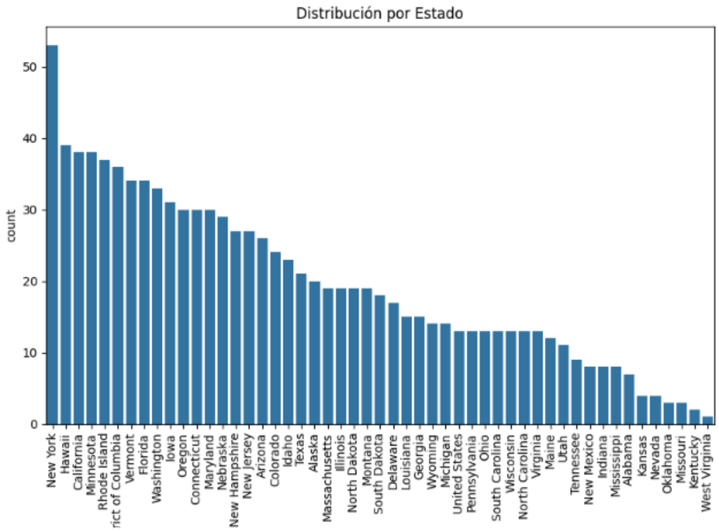
Various health-related metrics are included in this matrix, such as body mass index (BMI), age, systolic blood pressure (systolicbp), diastolic blood pressure (diastolicbp), fasting blood sugar (fasting sugar), hemoglobin A1c levels (hba1c), total cholesterol levels (totalcholesterol) and HDL and LDL cholesterol ratios (hdlcholesterol and ldlcholesterol).

EDA API

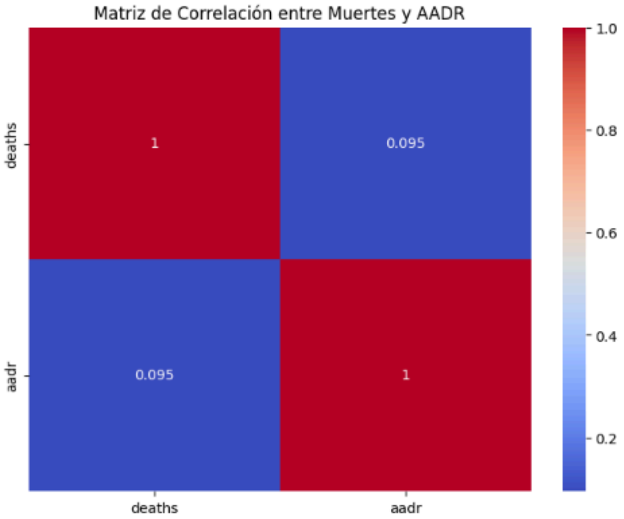
We started by loading the dataset, using pandas, we made observations to see each column, validating the null data, therefore no null values were detected, in addition to this, descriptive statistics were performed to better understand the characteristics of the variables. In addition, to identify outliers, we applied the Interquartile Range (IQR) method in the deaths and aadr columns.



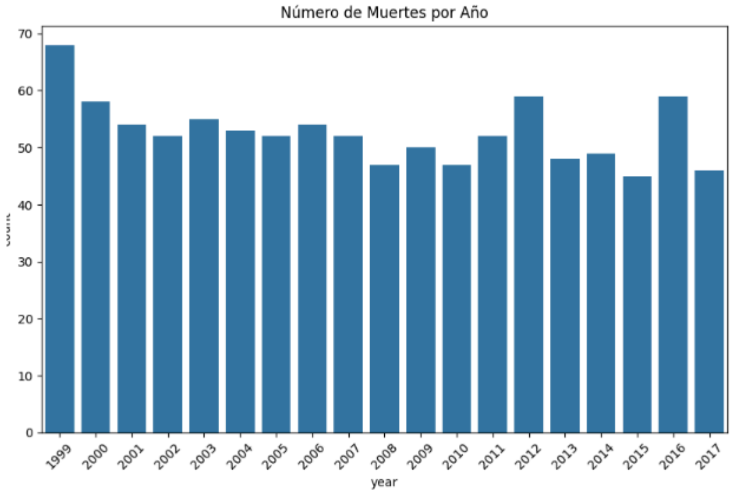
In the image above we can see that the distribution suggests that most records in the dataset represent causes of death with a low number of deaths, while only a small number of causes of death have a very high number of deaths.



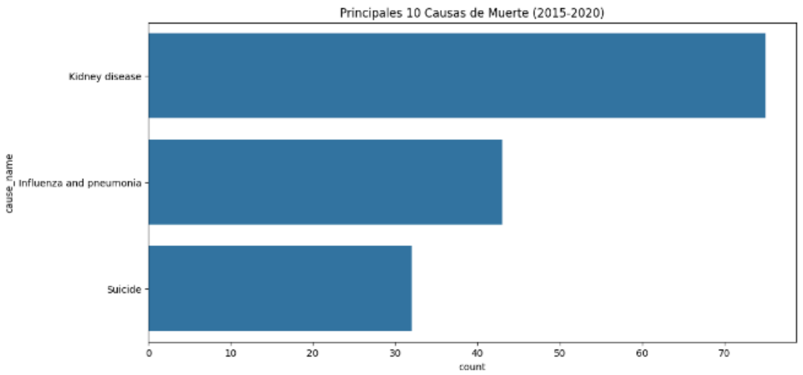
In the image above we can see that the higher number of records in certain states such as New York and California could reflect a larger population or a greater diversity of causes of death in those places.



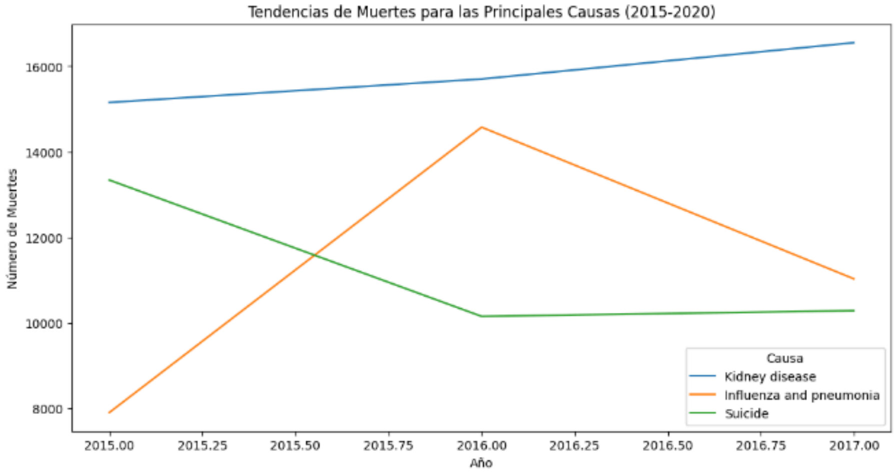
The image above shows a Correlation Matrix between the variables deaths and aadr (adjusted mortality rate) in the dataset, using a heat map to represent the correlation coefficients.



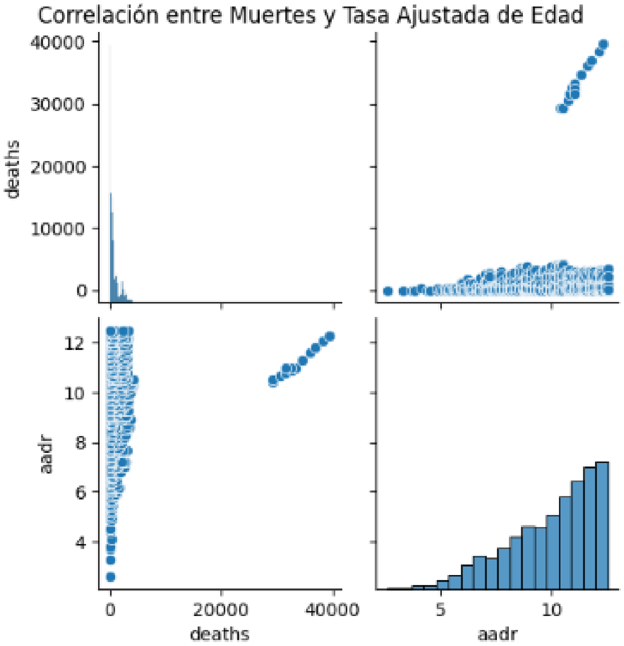
The graph above shows stability in the number of deaths per year over the period observed, with slight variations in certain years.



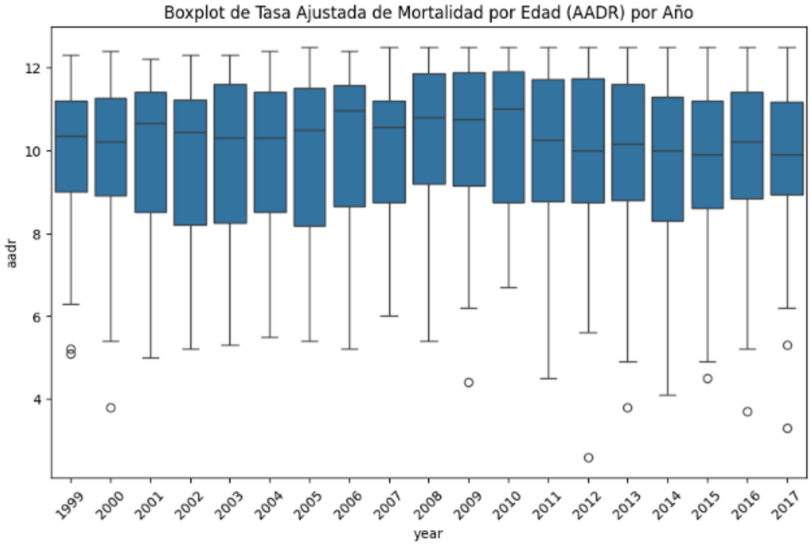
The presence of suicide among the three main causes of death underlines the importance of addressing mental health and suicide prevention as part of public health strategies.



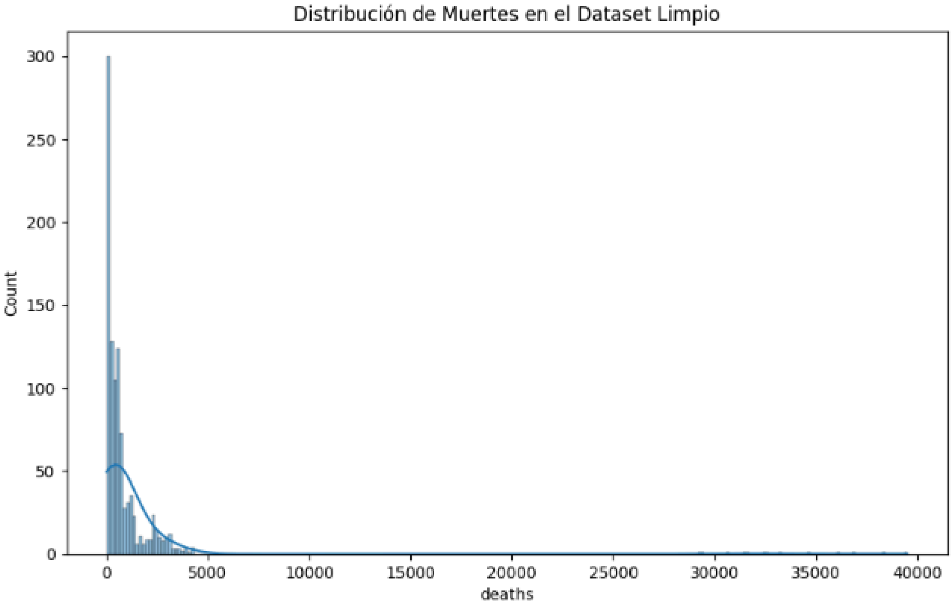
The graph above shows the Death Trends for the Main Causes (2015-2020), which represents the three main causes of death (kidney disease, influenza and pneumonia, and suicide) over time.



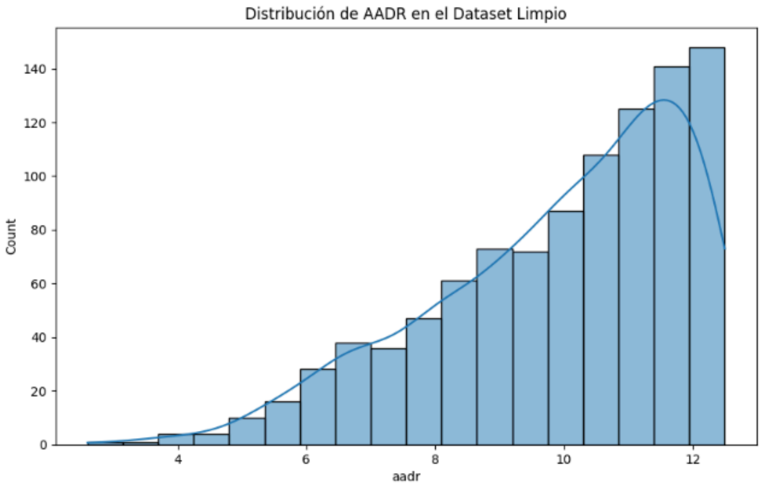
The image above shows a correlation graph (pairplot) between the variables deaths and aadr (adjusted mortality rate) in the dataset.



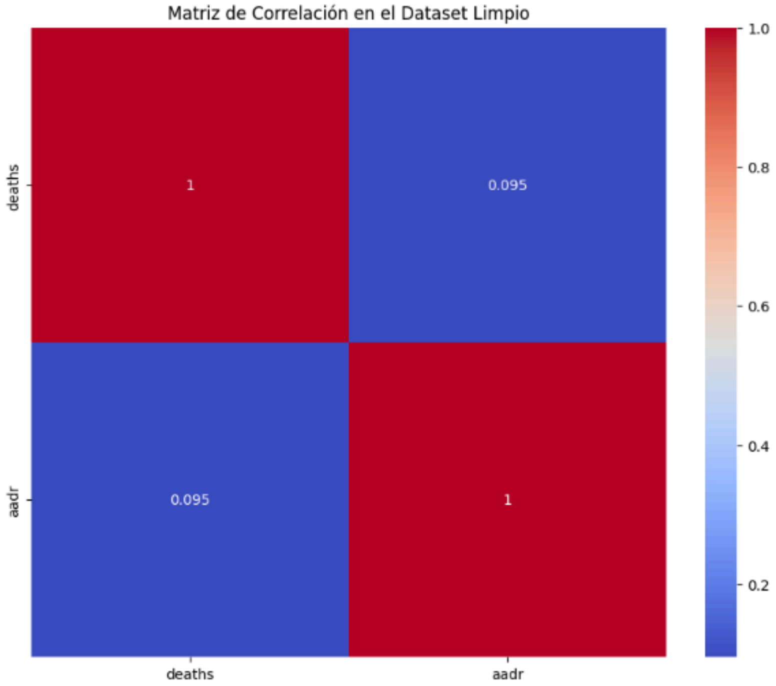
This boxplot shows that the age-adjusted mortality rate (AADR) has remained fairly stable over time, with no significant changes in the median or dispersion.



The distribution of deaths in the clean dataset confirms a large concentration of cases with few death values and a minority with extremely high values.

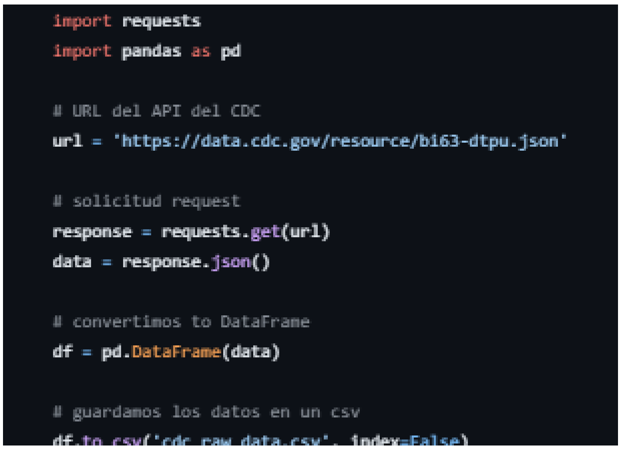


The distribution of aadr in the clean dataset is relatively symmetric and concentrated in the range 8 to 12, suggesting stability and moderation in age-adjusted mortality rates.



The image above shows the Correlation Matrix in the Clean Dataset between the variables deaths and aadr (age-adjusted mortality rate). This correlation matrix visualizes the correlation coefficients between both variables in a heat map.

API

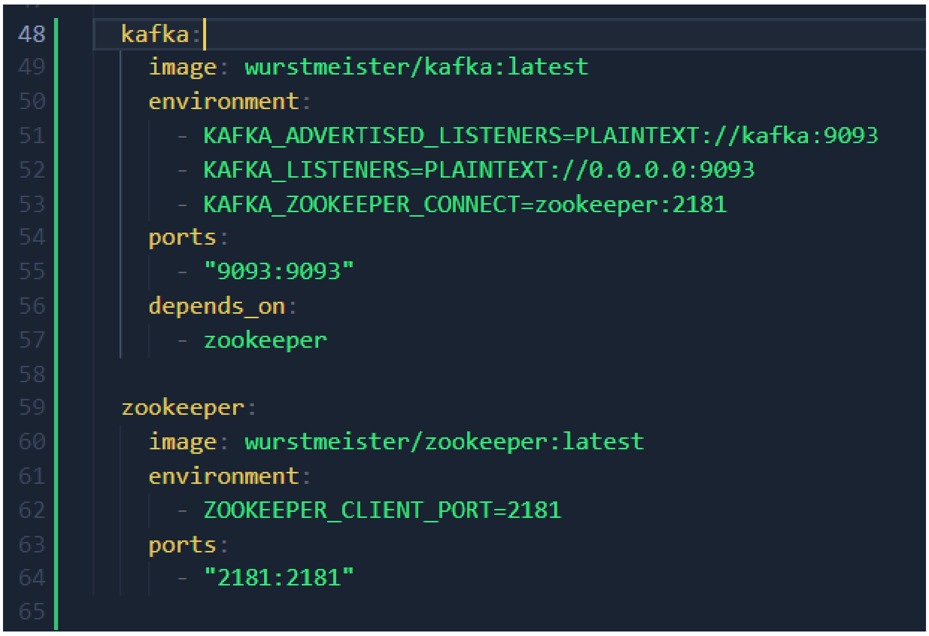


```python
import requests
import pandas as pd

# URL del API del CDC
url = 'https://data.cdc.gov/resource/bi63-dtpu.json'

# solicitud request
response = requests.get(url)
data = response.json()

# convertimos to DataFrame
df = pd.DataFrame(data)

# guardamos los datos en un csv
df.to_csv('cdc_raw_data.csv', index=False)
```

In the image above, it can be seen that data is extracted in real time from the CDC API, then converted to a structured format and saved locally for analysis.

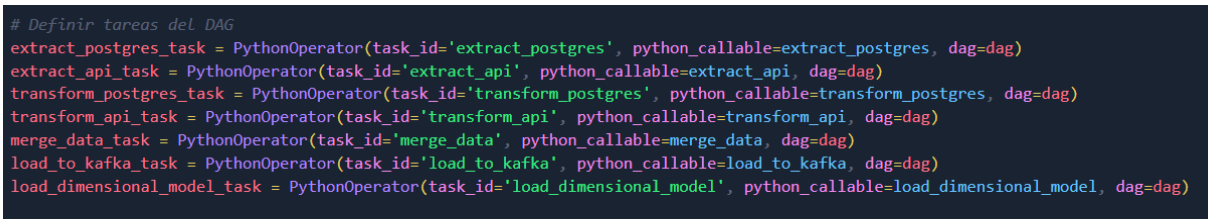
Setting up Kafka and Zookeeper in Docker Compose

The project uses Docker Compose to quickly and easily deploy Apache Kafka and Zookeeper services, making it easy to configure and deploy a real-time data streaming environment.



```yaml
  kafka:
    image: wurstmeister/kafka:latest
    environment:
      - KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://kafka:9093
      - KAFKA_LISTENERS=PLAINTEXT://0.0.0.0:9093
      - KAFKA_ZOOKEEPER_CONNECT=zookeeper:2181
    ports:
      - "9093:9093"
    depends_on:
      - zookeeper

  zookeeper:
    image: wurstmeister/zookeeper:latest
    environment:
      - ZOOKEEPER_CLIENT_PORT=2181
    ports:
      - "2181:2181"
```

In this project, a DAG in Apache Airflow is used to orchestrate data extraction, transformation, and loading (ETL) tasks. Each task in the DAG is defined by a PythonOperator, which allows specific Python functions to be executed in a predefined order.

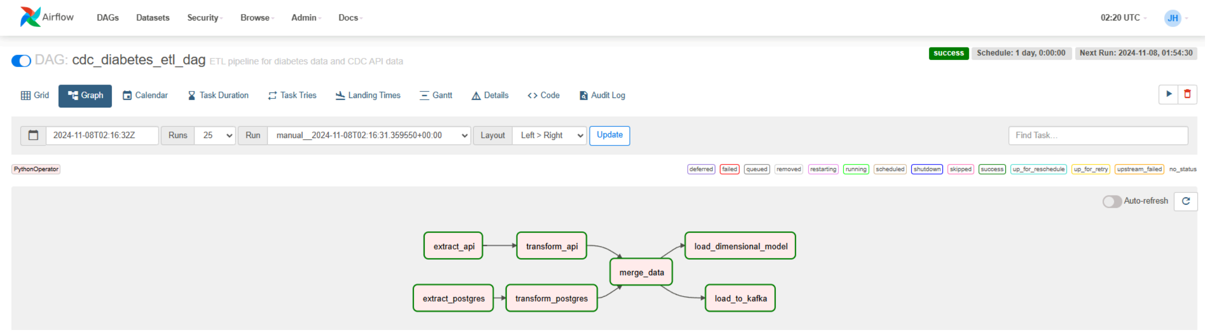


```
# Definir tareas del DAG
extract_postgres_task = PythonOperator(task_id='extract_postgres', python_callable=extract_postgres, dag=dag)
extract_api_task = PythonOperator(task_id='extract_api', python_callable=extract_api, dag=dag)
transform_postgres_task = PythonOperator(task_id='transform_postgres', python_callable=transform_postgres, dag=dag)
transform_api_task = PythonOperator(task_id='transform_api', python_callable=transform_api, dag=dag)
merge_data_task = PythonOperator(task_id='merge_data', python_callable=merge_data, dag=dag)
load_to_kafka_task = PythonOperator(task_id='load_to_kafka', python_callable=load_to_kafka, dag=dag)
load_dimensional_model_task = PythonOperator(task_id='load_dimensional_model', python_callable=load_dimensional_model, dag=dag)
```

In this interface, you can observe the structure and order of the tasks, as well as their interdependence, within the ETL pipeline.

DAG Visualization Description

In the graphical visualization of the DAG, each task is represented as a node, and arrows indicate the sequential flow between tasks. This design allows you to follow the ETL process from the initial data extraction to its final loading. Each of the steps and their connections are described below:



DAG: cdc_diabetes_etl_dag ETL pipeline for diabetes data and CDC API data

success Schedule: 1 day, 0:00:00 Next Run: 2024-11-08, 01:54:30

2024-11-08T02:16:32Z Runs 25 Run manual__2024-11-08T02:16:31.359550+00:00 Layout Left > Right

In the following image we detail the DAG visualization

1.**Extract_api** y **extract_postgres**

- These two tasks are executed in parallel at the start of the flow. extract_api is responsible for fetching data from an API, while extract_postgres extracts data from a PostgreSQL database.

2.Transform_api y **transform_postgres**

- These tasks are the respective transformation steps for data extracted from the API and from PostgreSQL. transform_api applies transformations to the API data, and transform_postgres does the same for PostgreSQL data.

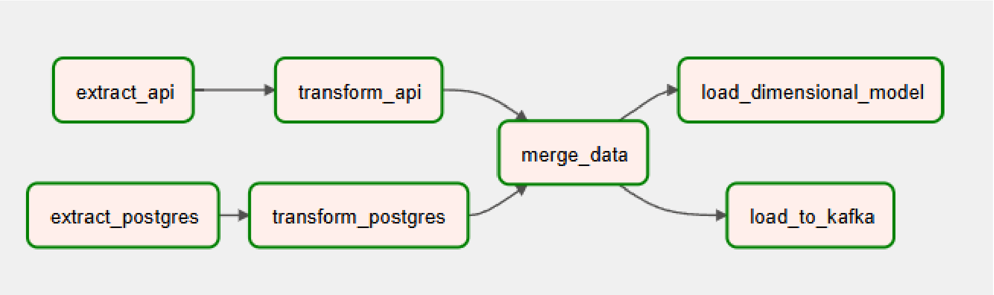
3.Merge_data

- This task took the transformed data from both sources and combined them into a single unified dataset.

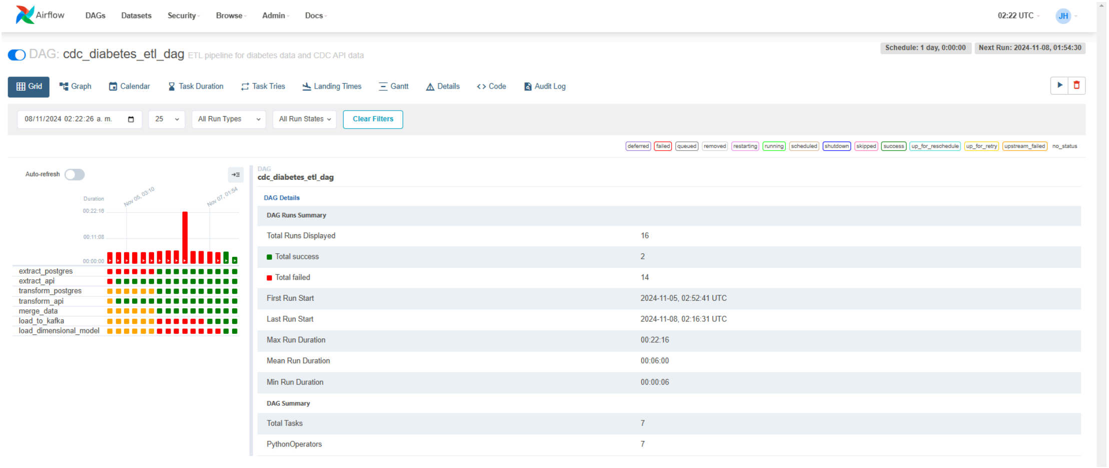
4.Load_dimensional_model y **load_to_kafka**

  - load_dimensional_model: Loads the combined dataset into a dimensional model within a database, facilitating storage.
  - load_to_kafka: Sends the data to Kafka for real-time streaming, enabling immediate consumption of the data by real-time applications.

- Both tasks depend on merge_data, thus completing the workflow.



On the left side, a grid view of the execution history of each task in the DAG is presented, where each cell represents a specific execution and is color-coded by outcome, where we highlight the execution of all tasks.



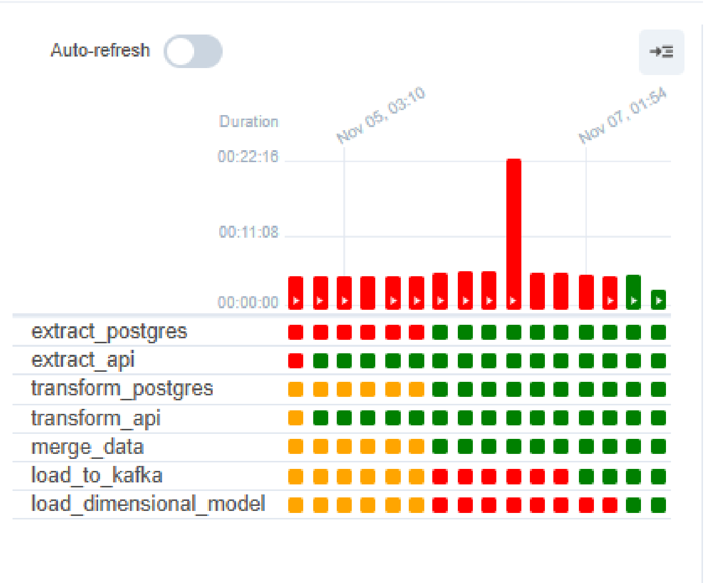
DAG: cdc_diabetes_etl_dag ETL pipeline for diabetes data and CDC API data

Schedule: 1 day, 0:00:00 Next Run: 2024-11-08, 01:54:30

DAG
cdc_diabetes_etl_dag

DAG Details

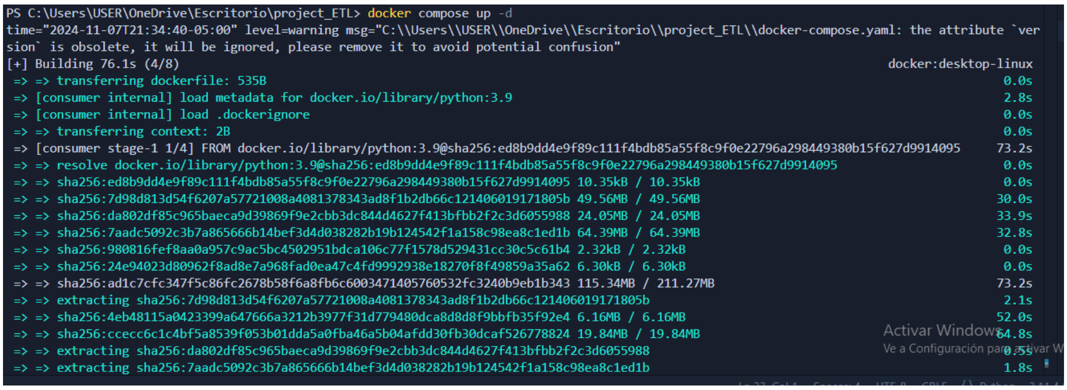
DAG Runs Summary	
Total Runs Displayed	16
🟩 Total success	2
🟥 Total failed	14
First Run Start	2024-11-05, 02:52:41 UTC
Last Run Start	2024-11-08, 02:16:31 UTC
Max Run Duration	00:22:16
Mean Run Duration	00:06:00
Min Run Duration	00:00:06
DAG Summary	
Total Tasks	7
PythonOperators	7



extract_postgres	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
extract_api	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
transform_postgres	🟧	🟧	🟧	🟧	🟧	🟧	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
transform_api	🟧	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
merge_data	🟧	🟧	🟧	🟧	🟧	🟧	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
load_to_kafka	🟧	🟧	🟧	🟧	🟧	🟧	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	🟩
load_dimensional_model	🟧	🟧	🟧	🟧	🟧	🟧	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟩	🟩

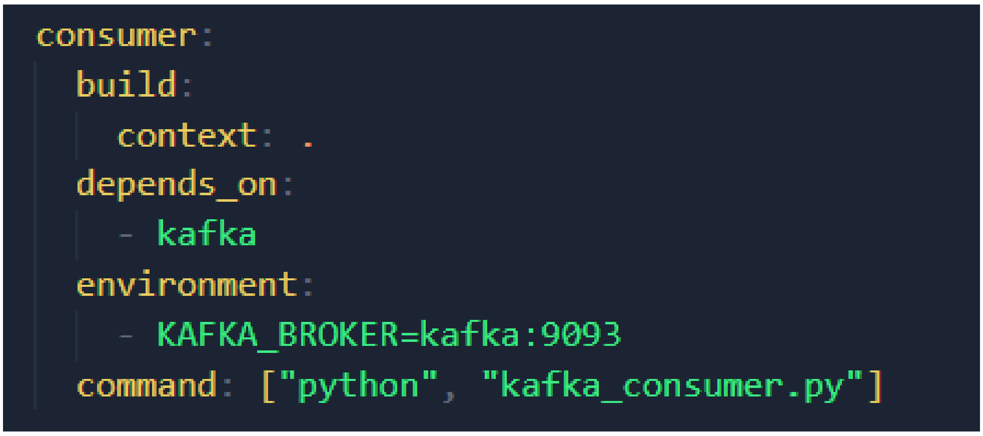
In the image above we can see that the grid view provides a quick and effective overview of the overall DAG performance and helps operators to keep the ETL pipeline in good shape.

This component of the project is a Kafka consumer developed in Python. Its purpose is to connect to the merged_data_topic topic in the Kafka broker and process in real time the transmitted messages, which contain merged data from the ETL pipeline.



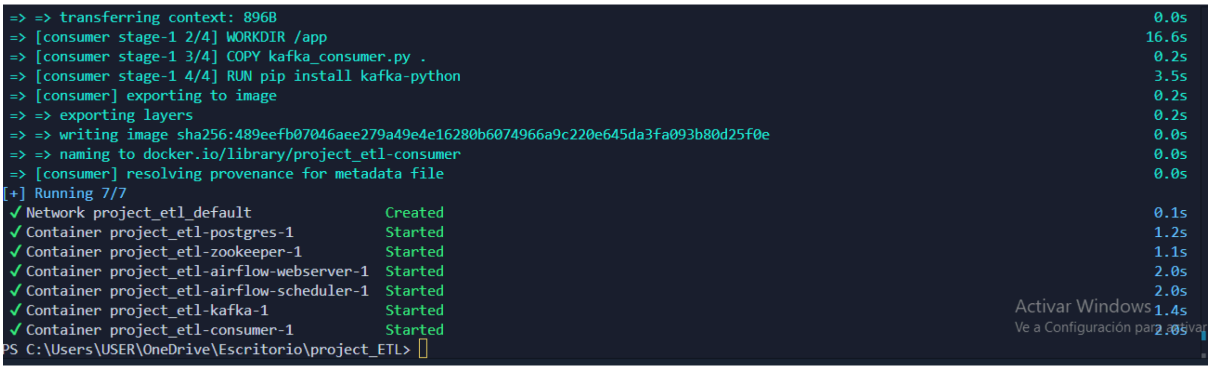
```
PS C:\Users\USER\OneDrive\Escritorio\project_ETL> docker compose up -d
time="2024-11-07T21:34:40-05:00" level=warning msg="C:\\Users\\USER\\OneDrive\\Escritorio\\project_ETL\\docker-compose.yaml: the attribute `ver
sion` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Building 76.1s (4/8)                                                                                              docker:desktop-linux
 => => transferring dockerfile: 535B                                                                                                   0.0s
 => [consumer internal] load metadata for docker.io/library/python:3.9                                                                 2.8s
 => [consumer internal] load .dockerignore                                                                                             0.0s
 => => transferring context: 2B                                                                                                        0.0s
 => [consumer stage-1 1/4] FROM docker.io/library/python:3.9@sha256:ed8b9dd4e9f89c111f4bdb85a55f8c9f0e22796a298449380b15f627d9914095     73.2s
 => => resolve docker.io/library/python:3.9@sha256:ed8b9dd4e9f89c111f4bdb85a55f8c9f0e22796a298449380b15f627d9914095                     0.0s
 => => sha256:ed8b9dd4e9f89c111f4bdb85a55f8c9f0e22796a298449380b15f627d9914095 10.35kB / 10.35kB                                        0.0s
 => => sha256:7d98d813d54f6207a57721008a4081378343ad8f1b2db66c121406019171805b 49.56MB / 49.56MB                                       30.0s
 => => sha256:da802df85c965baeca9d39869f9e2cbb3dc844d4627f413bfbb2f2c3d6055988 24.05MB / 24.05MB                                       33.9s
 => => sha256:7aadc5092c3b7a865666b14bef3d4d038282b19b124542f1a158c98ea8c1ed1b 64.39MB / 64.39MB                                       32.8s
 => => sha256:980816fef8aa0a957c9ac5bc4502951bdca106c77f1578d529431cc30c5c61b4 2.32kB / 2.32kB                                          0.0s
 => => sha256:24e94023d80962f8ad8e7a968fad0ea47c4fd9992938e18270f8f49859a35a62 6.30kB / 6.30kB                                          0.0s
 => => sha256:ad1c7cfc347f5c86fc2678b58f6a8fb6c6003471405760532fc3240b9eb1b343 115.34MB / 211.27MB                                     73.2s
 => => extracting sha256:7d98d813d54f6207a57721008a4081378343ad8f1b2db66c121406019171805b                                               2.1s
 => => sha256:4eb48115a0423399a647666a3212b3977f31d779480dca8d8d8f9bbfb35f92e4 6.16MB / 6.16MB                                         52.0s
 => => sha256:ccecc6c1c4bf5a8539f053b01dda5a0fba46a5b04afdd30fb30dcaf526778824 19.84MB / 19.84MB                                       64.8s
 => => extracting sha256:da802df85c965baeca9d39869f9e2cbb3dc844d4627f413bfbb2f2c3d6055988                                               0.5s
 => => extracting sha256:7aadc5092c3b7a865666b14bef3d4d038282b19b124542f1a158c98ea8c1ed1b                                               1.8s
```

This section describes the configuration of the consumer service in the docker-compose.yml file, which deploys the Kafka consumer that connects to the merged_data_topic topic.



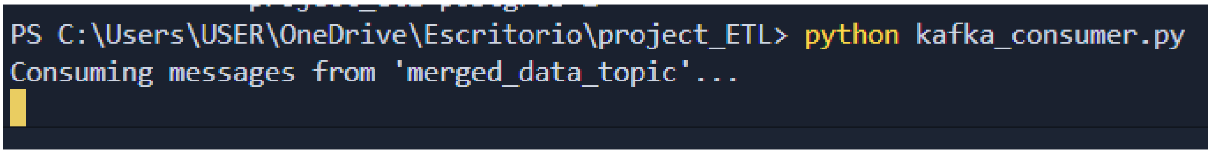
```yaml
consumer:
  build:
    context: .
  depends_on:
    - kafka
  environment:
    - KAFKA_BROKER=kafka:9093
  command: ["python", "kafka_consumer.py"]
```

All services show a status of Started, indicating that the environment is fully operational and each component has been successfully started.



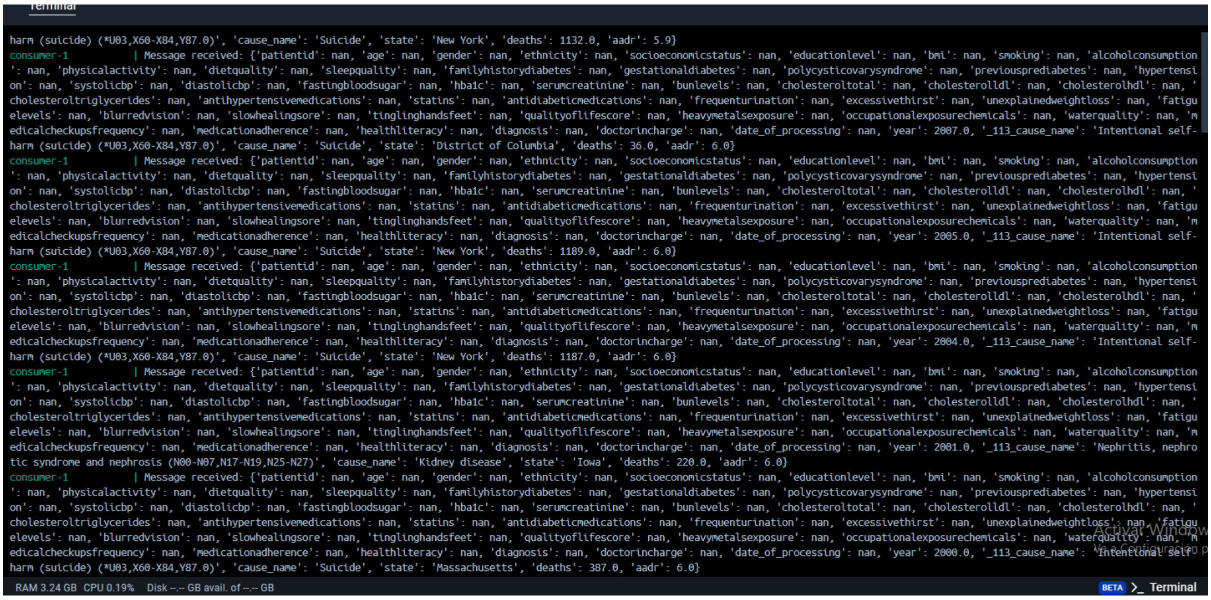
```
=> => transferring context: 896B                                                                                0.0s
=> [consumer stage-1 2/4] WORKDIR /app                                                                         16.6s
=> [consumer stage-1 3/4] COPY kafka_consumer.py .                                                              0.2s
=> [consumer stage-1 4/4] RUN pip install kafka-python                                                          3.5s
=> [consumer] exporting to image                                                                                0.2s
=> => exporting layers                                                                                          0.2s
=> => writing image sha256:489eefb07046aee279a49e4e16280b6074966a9c220e645da3fa093b80d25f0e                     0.0s
=> => naming to docker.io/library/project_etl-consumer                                                          0.0s
=> [consumer] resolving provenance for metadata file                                                            0.0s
[+] Running 7/7
 ✓ Network project_etl_default                  Created                                                         0.1s
 ✓ Container project_etl-postgres-1             Started                                                         1.2s
 ✓ Container project_etl-zookeeper-1            Started                                                         1.1s
 ✓ Container project_etl-airflow-webserver-1    Started                                                         2.0s
 ✓ Container project_etl-airflow-scheduler-1    Started                                                         2.0s
 ✓ Container project_etl-kafka-1                Started                                                         1.4s
 ✓ Container project_etl-consumer-1             Started                                                         2.0s
PS C:\Users\USER\OneDrive\Escritorio\project_ETL> 
```

In this step, the kafka_consumer.py script is executed, which is a Kafka consumer configured to connect to the merged_data_topic topic. This consumer is designed to receive and process in real time the messages transmitted to this topic as part of the ETL pipeline.



```
PS C:\Users\USER\OneDrive\Escritorio\project_ETL> python kafka_consumer.py
Consuming messages from 'merged_data_topic'...
```

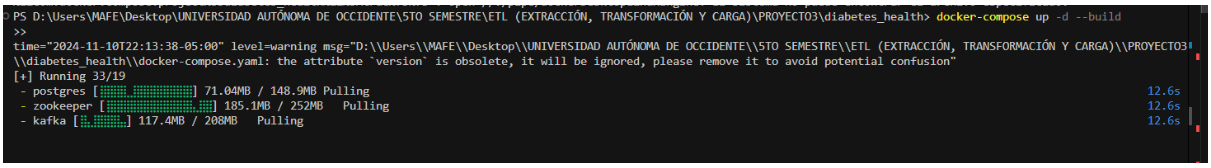
It is evident that it is functioning satisfactorily.



```
harm (suicide) (*U03,X60-X84,Y87.0)', 'cause_name': 'Suicide', 'state': 'New York', 'deaths': 1132.0, 'aadr': 5.9}
consumer-1           | Message received: {'patientid': nan, 'age': nan, 'gender': nan, 'ethnicity': nan, 'socioeconomicstatus': nan, 'educationlevel': nan, 'bmi': nan, 'smoking': nan, 'alcoholconsumption
': nan, 'physicalactivity': nan, 'dietquality': nan, 'sleepquality': nan, 'familyhistorydiabetes': nan, 'gestationaldiabetes': nan, 'polycysticovarysyndrome': nan, 'previousprediabetes': nan, 'hypertensi
on': nan, 'systolicbp': nan, 'diastolicbp': nan, 'fastingbloodsugar': nan, 'hba1c': nan, 'serumcreatinine': nan, 'bunlevels': nan, 'cholesteroltotal': nan, 'cholesterolldl': nan, 'cholesterolhdl': nan, '
cholesteroltriglycerides': nan, 'antihypertensivemedications': nan, 'statins': nan, 'antidiabeticmedications': nan, 'frequenturination': nan, 'excessivethirst': nan, 'unexplainedweightloss': nan, 'fatigu
elevels': nan, 'blurredvision': nan, 'slowhealingsore': nan, 'tinglinghandsfeet': nan, 'qualityoflifescore': nan, 'heavymetalsexposure': nan, 'occupationalexposurechemicals': nan, 'waterquality': nan, 'm
edicalcheckupsfrequency': nan, 'medicationadherence': nan, 'healthliteracy': nan, 'diagnosis': nan, 'doctorincharge': nan, 'date_of_processing': nan, 'year': 2007.0, '_113_cause_name': 'Intentional self-
harm (suicide) (*U03,X60-X84,Y87.0)', 'cause_name': 'Suicide', 'state': 'District of Columbia', 'deaths': 36.0, 'aadr': 6.0}
consumer-1           | Message received: {'patientid': nan, 'age': nan, 'gender': nan, 'ethnicity': nan, 'socioeconomicstatus': nan, 'educationlevel': nan, 'bmi': nan, 'smoking': nan, 'alcoholconsumption
': nan, 'physicalactivity': nan, 'dietquality': nan, 'sleepquality': nan, 'familyhistorydiabetes': nan, 'gestationaldiabetes': nan, 'polycysticovarysyndrome': nan, 'previousprediabetes': nan, 'hypertensi
on': nan, 'systolicbp': nan, 'diastolicbp': nan, 'fastingbloodsugar': nan, 'hba1c': nan, 'serumcreatinine': nan, 'bunlevels': nan, 'cholesteroltotal': nan, 'cholesterolldl': nan, 'cholesterolhdl': nan, '
cholesteroltriglycerides': nan, 'antihypertensivemedications': nan, 'statins': nan, 'antidiabeticmedications': nan, 'frequenturination': nan, 'excessivethirst': nan, 'unexplainedweightloss': nan, 'fatigu
elevels': nan, 'blurredvision': nan, 'slowhealingsore': nan, 'tinglinghandsfeet': nan, 'qualityoflifescore': nan, 'heavymetalsexposure': nan, 'occupationalexposurechemicals': nan, 'waterquality': nan, 'm
edicalcheckupsfrequency': nan, 'medicationadherence': nan, 'healthliteracy': nan, 'diagnosis': nan, 'doctorincharge': nan, 'date_of_processing': nan, 'year': 2005.0, '_113_cause_name': 'Intentional self-
harm (suicide) (*U03,X60-X84,Y87.0)', 'cause_name': 'Suicide', 'state': 'New York', 'deaths': 1189.0, 'aadr': 6.0}
consumer-1           | Message received: {'patientid': nan, 'age': nan, 'gender': nan, 'ethnicity': nan, 'socioeconomicstatus': nan, 'educationlevel': nan, 'bmi': nan, 'smoking': nan, 'alcoholconsumption
': nan, 'physicalactivity': nan, 'dietquality': nan, 'sleepquality': nan, 'familyhistorydiabetes': nan, 'gestationaldiabetes': nan, 'polycysticovarysyndrome': nan, 'previousprediabetes': nan, 'hypertensi
on': nan, 'systolicbp': nan, 'diastolicbp': nan, 'fastingbloodsugar': nan, 'hba1c': nan, 'serumcreatinine': nan, 'bunlevels': nan, 'cholesteroltotal': nan, 'cholesterolldl': nan, 'cholesterolhdl': nan, '
cholesteroltriglycerides': nan, 'antihypertensivemedications': nan, 'statins': nan, 'antidiabeticmedications': nan, 'frequenturination': nan, 'excessivethirst': nan, 'unexplainedweightloss': nan, 'fatigu
elevels': nan, 'blurredvision': nan, 'slowhealingsore': nan, 'tinglinghandsfeet': nan, 'qualityoflifescore': nan, 'heavymetalsexposure': nan, 'occupationalexposurechemicals': nan, 'waterquality': nan, 'm
edicalcheckupsfrequency': nan, 'medicationadherence': nan, 'healthliteracy': nan, 'diagnosis': nan, 'doctorincharge': nan, 'date_of_processing': nan, 'year': 2004.0, '_113_cause_name': 'Intentional self-
harm (suicide) (*U03,X60-X84,Y87.0)', 'cause_name': 'Suicide', 'state': 'New York', 'deaths': 1187.0, 'aadr': 6.0}
consumer-1           | Message received: {'patientid': nan, 'age': nan, 'gender': nan, 'ethnicity': nan, 'socioeconomicstatus': nan, 'educationlevel': nan, 'bmi': nan, 'smoking': nan, 'alcoholconsumption
': nan, 'physicalactivity': nan, 'dietquality': nan, 'sleepquality': nan, 'familyhistorydiabetes': nan, 'gestationaldiabetes': nan, 'polycysticovarysyndrome': nan, 'previousprediabetes': nan, 'hypertensi
on': nan, 'systolicbp': nan, 'diastolicbp': nan, 'fastingbloodsugar': nan, 'hba1c': nan, 'serumcreatinine': nan, 'bunlevels': nan, 'cholesteroltotal': nan, 'cholesterolldl': nan, 'cholesterolhdl': nan, '
cholesteroltriglycerides': nan, 'antihypertensivemedications': nan, 'statins': nan, 'antidiabeticmedications': nan, 'frequenturination': nan, 'excessivethirst': nan, 'unexplainedweightloss': nan, 'fatigu
elevels': nan, 'blurredvision': nan, 'slowhealingsore': nan, 'tinglinghandsfeet': nan, 'qualityoflifescore': nan, 'heavymetalsexposure': nan, 'occupationalexposurechemicals': nan, 'waterquality': nan, 'm
edicalcheckupsfrequency': nan, 'medicationadherence': nan, 'healthliteracy': nan, 'diagnosis': nan, 'doctorincharge': nan, 'date_of_processing': nan, 'year': 2001.0, '_113_cause_name': 'Nephritis, nephro
tic syndrome and nephrosis (N00-N07,N17-N19,N25-N27)', 'cause_name': 'Kidney disease', 'state': 'Iowa', 'deaths': 220.0, 'aadr': 6.0}
consumer-1           | Message received: {'patientid': nan, 'age': nan, 'gender': nan, 'ethnicity': nan, 'socioeconomicstatus': nan, 'educationlevel': nan, 'bmi': nan, 'smoking': nan, 'alcoholconsumption
': nan, 'physicalactivity': nan, 'dietquality': nan, 'sleepquality': nan, 'familyhistorydiabetes': nan, 'gestationaldiabetes': nan, 'polycysticovarysyndrome': nan, 'previousprediabetes': nan, 'hypertensi
on': nan, 'systolicbp': nan, 'diastolicbp': nan, 'fastingbloodsugar': nan, 'hba1c': nan, 'serumcreatinine': nan, 'bunlevels': nan, 'cholesteroltotal': nan, 'cholesterolldl': nan, 'cholesterolhdl': nan, '
cholesteroltriglycerides': nan, 'antihypertensivemedications': nan, 'statins': nan, 'antidiabeticmedications': nan, 'frequenturination': nan, 'excessivethirst': nan, 'unexplainedweightloss': nan, 'fatigu
elevels': nan, 'blurredvision': nan, 'slowhealingsore': nan, 'tinglinghandsfeet': nan, 'qualityoflifescore': nan, 'heavymetalsexposure': nan, 'occupationalexposurechemicals': nan, 'waterquality': nan, 'm
edicalcheckupsfrequency': nan, 'medicationadherence': nan, 'healthliteracy': nan, 'diagnosis': nan, 'doctorincharge': nan, 'date_of_processing': nan, 'year': 2000.0, '_113_cause_name': 'Intentional self-
harm (suicide) (*U03,X60-X84,Y87.0)', 'cause_name': 'Suicide', 'state': 'Massachusetts', 'deaths': 387.0, 'aadr': 6.0}
```

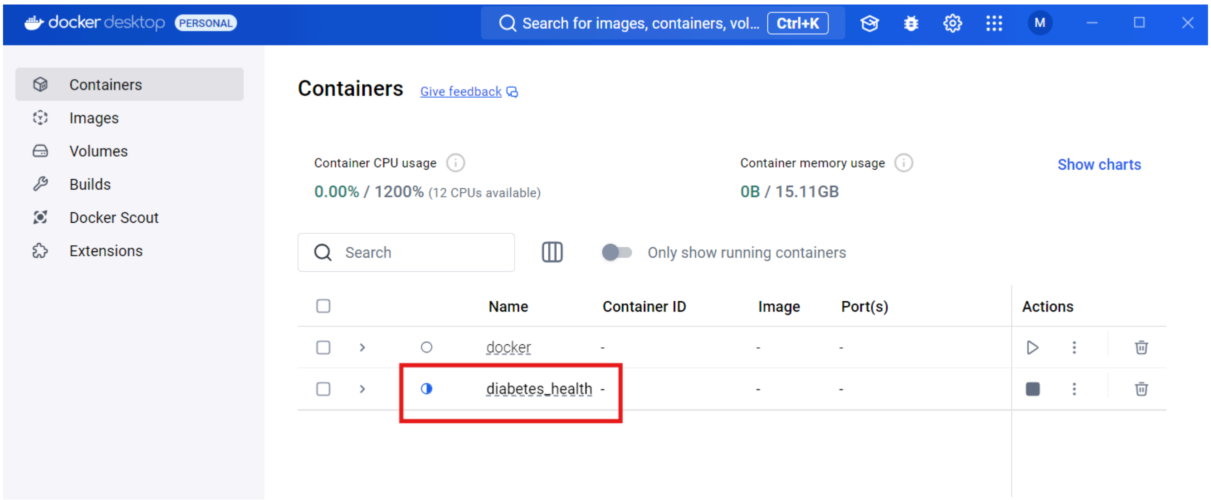
Service-Specific Dockerfile: We now use Dockerfile.airflow for Airflow services (airflow-webserver and airflow-scheduler) and Dockerfile.consumer for the consumer service.
Using dockerfile in build: For each service that needs a specific Dockerfile, we specify the Dockerfile with the dockerfile key.
Separation of Dependencies: Each Dockerfile installs only the dependencies needed for the specific service, avoiding duplication and potential conflicts.

We run the docker



```
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\PROYECTO3\diabetes_health> docker-compose up -d --build
>>
time="2024-11-10T22:13:38-05:00" level=warning msg="D:\\Users\\MAFE\\Desktop\\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\\5TO SEMESTRE\\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\\PROYECTO3
\\diabetes_health\\docker-compose.yaml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 33/19
 - postgres [⣿⣿⣿⣿⣀⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿] 71.04MB / 148.9MB Pulling                                                    12.6s
 - zookeeper [⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣿⣀⣿⣿] 185.1MB / 252MB   Pulling                                                12.6s
 - kafka [⣿⣀⣿⣿⣿⣿⣿⣀] 117.4MB / 208MB   Pulling                                                             12.6s
```

This container, identified as diabetes.health, is part of the Docker environment used for the ETL project aimed at managing and analyzing health data, specifically related to diabetes.



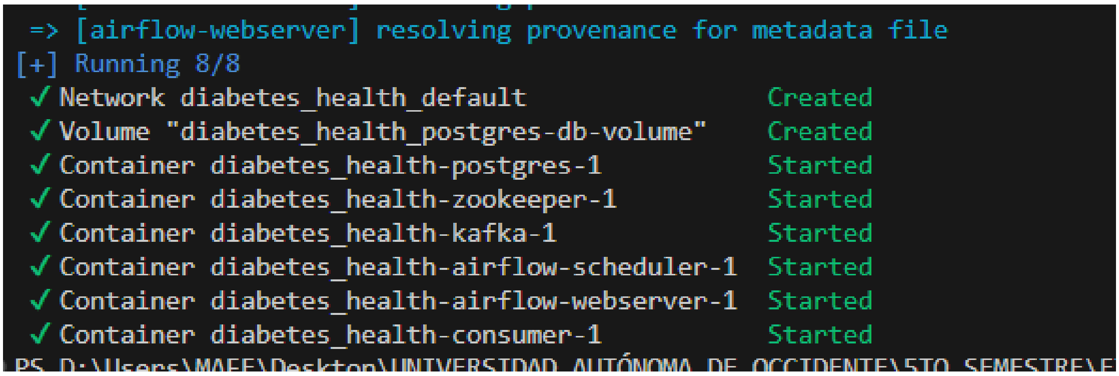
Containers

Container CPU usage 0.00% / 1200% (12 CPUs available)

Container memory usage 0B / 15.11GB

	Name	Container ID	Image	Port(s)	Actions
○	docker	-	-	-	
◑	diabetes_health	-	-	-	

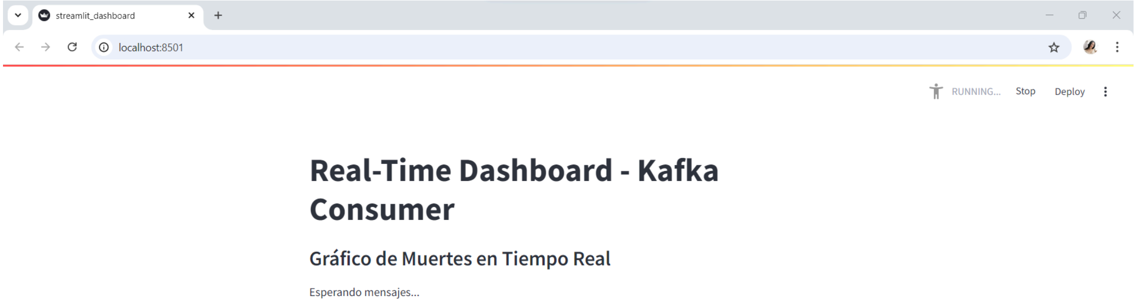
This Docker Compose environment has been configured to run several essential services for the ETL pipeline focused on health data, specifically diabetes-related data. The output shows the successful creation and startup of all containers and components in the diabetes_health network.



```
 => [airflow-webserver] resolving provenance for metadata file
[+] Running 8/8
 ✓ Network diabetes_health_default                  Created
 ✓ Volume "diabetes_health_postgres-db-volume"      Created
 ✓ Container diabetes_health-postgres-1             Started
 ✓ Container diabetes_health-zookeeper-1            Started
 ✓ Container diabetes_health-kafka-1                Started
 ✓ Container diabetes_health-airflow-scheduler-1   Started
 ✓ Container diabetes_health-airflow-webserver-1   Started
 ✓ Container diabetes_health-consumer-1             Started
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\E
```

REAL TIME DASHBOARD

The real-time dashboard has been developed to visualize health-related data, using Streamlit for the user interface and Kafka for real-time data streaming.



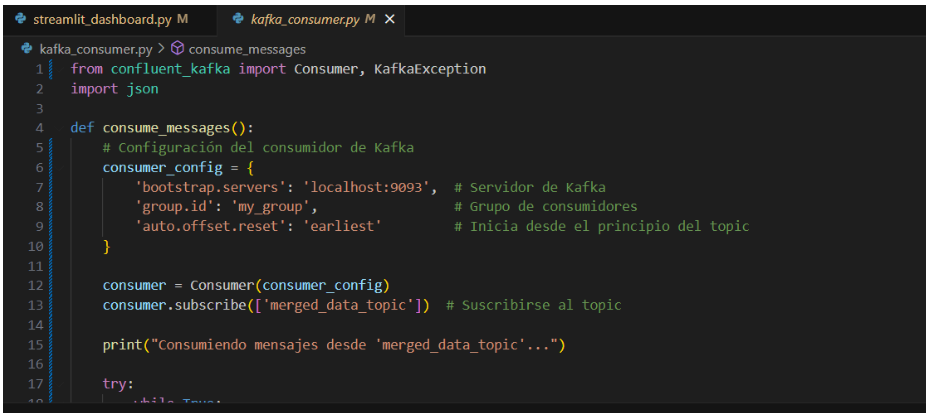
Real-Time Dashboard - Kafka Consumer

Gráfico de Muertes en Tiempo Real

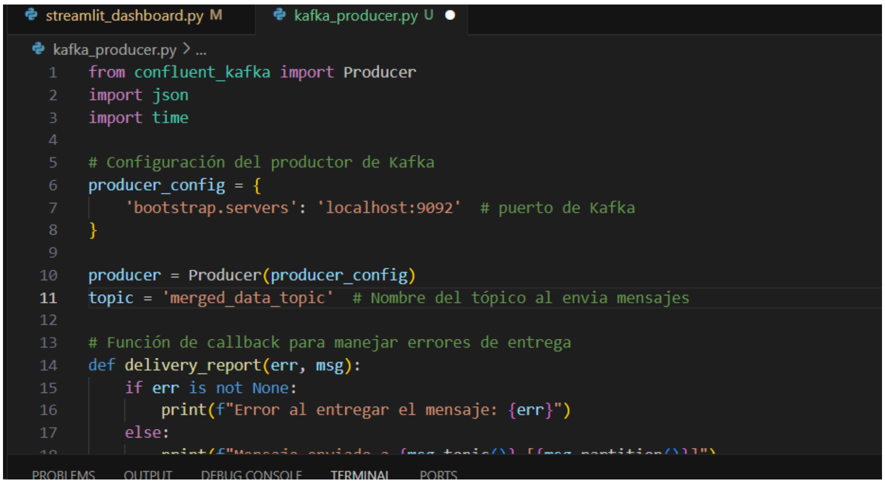
Esperando mensajes...

We modified the _kafka_consumer.py, the stremlit_dashboard.py, and the kafka_producer.py.

In the following image we can see that a Kafka consumer is defined using the confluent_kafka library, configured to subscribe to the merged_data_topic topic and receive messages in real time. The implementation allows for continuous streaming of data to a dashboard or for other real-time analysis purposes.

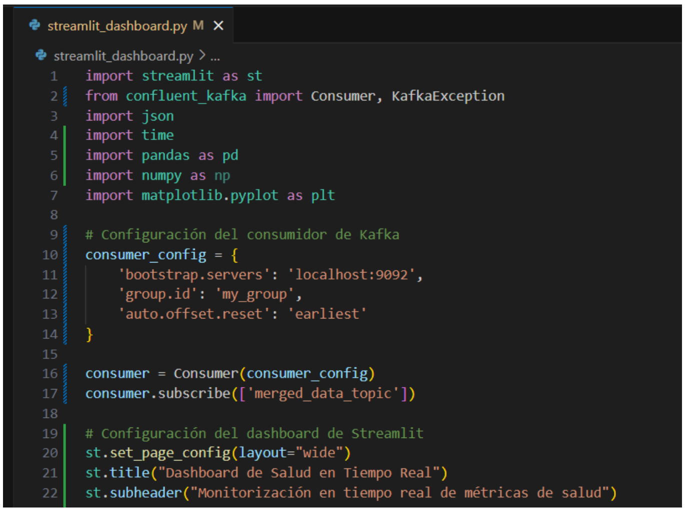


```python
from confluent_kafka import Consumer, KafkaException
import json

def consume_messages():
    # Configuración del consumidor de Kafka
    consumer_config = {
        'bootstrap.servers': 'localhost:9093',  # Servidor de Kafka
        'group.id': 'my_group',                 # Grupo de consumidores
        'auto.offset.reset': 'earliest'         # Inicia desde el principio del topic
    }

    consumer = Consumer(consumer_config)
    consumer.subscribe(['merged_data_topic'])  # Suscribirse al topic

    print("Consumiendo mensajes desde 'merged_data_topic'...")

    try:
        while True:
```



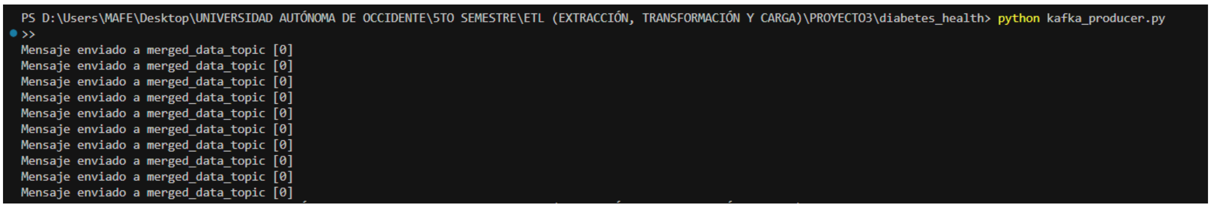
```python
from confluent_kafka import Producer
import json
import time

# Configuración del productor de Kafka
producer_config = {
    'bootstrap.servers': 'localhost:9092'  # puerto de Kafka
}

producer = Producer(producer_config)
topic = 'merged_data_topic'  # Nombre del tópico al envia mensajes

# Función de callback para manejar errores de entrega
def delivery_report(err, msg):
    if err is not None:
        print(f"Error al entregar el mensaje: {err}")
    else:
        print(f"Mensaje enviado a {msg.topic()} [{msg.partition()}]")
```

In the image above we can see that a Kafka producer is defined using the confluent_kafka library. Its purpose is to send messages to the merged_data_topic topic, which can then be consumed in real time by a Kafka consumer and visualized in a Streamlit dashboard.



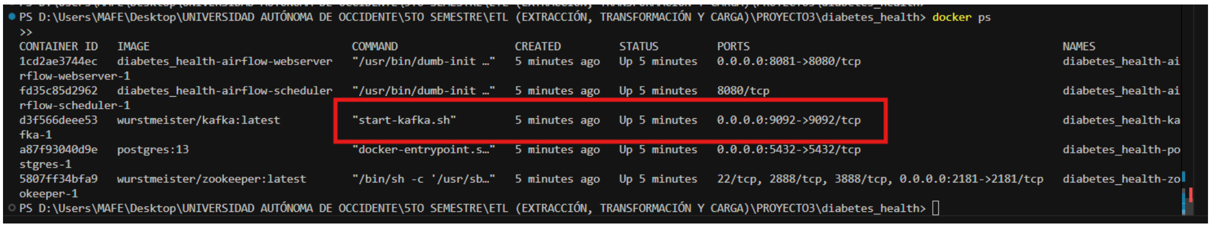
```python
import streamlit as st
from confluent_kafka import Consumer, KafkaException
import json
import time
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Configuración del consumidor de Kafka
consumer_config = {
    'bootstrap.servers': 'localhost:9092',
    'group.id': 'my_group',
    'auto.offset.reset': 'earliest'
}

consumer = Consumer(consumer_config)
consumer.subscribe(['merged_data_topic'])

# Configuración del dashboard de Streamlit
st.set_page_config(layout="wide")
st.title("Dashboard de Salud en Tiempo Real")
st.subheader("Monitorización en tiempo real de métricas de salud")
```

Here we can see that data is already being sent to the topic (merged_data_topic).



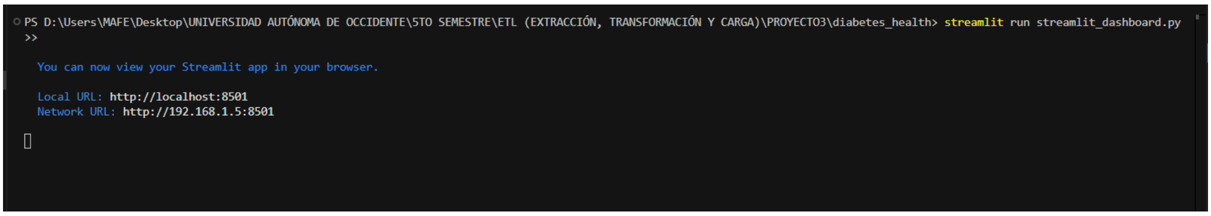
```
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\PROYECTO3\diabetes_health> python kafka_producer.py
>>
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
Mensaje enviado a merged_data_topic [0]
```

We check the port on which kafka is running in Docker (9092).



```
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\PROYECTO3\diabetes_health> docker ps
>>
CONTAINER ID   IMAGE                                 COMMAND                  CREATED         STATUS         PORTS                                                  NAMES
1cd2ae3744ec   diabetes_health-airflow-webserver     "/usr/bin/dumb-init …"   5 minutes ago   Up 5 minutes   0.0.0.0:8081->8080/tcp                                 diabetes_health-ai
rflow-webserver-1
fd35c85d2962   diabetes_health-airflow-scheduler     "/usr/bin/dumb-init …"   5 minutes ago   Up 5 minutes   8080/tcp                                               diabetes_health-ai
rflow-scheduler-1
d3f566deee53   wurstmeister/kafka:latest             "start-kafka.sh"         5 minutes ago   Up 5 minutes   0.0.0.0:9092->9092/tcp                                 diabetes_health-ka
fka-1
a87f93040d9e   postgres:13                           "docker-entrypoint.s…"   5 minutes ago   Up 5 minutes   0.0.0.0:5432->5432/tcp                                 diabetes_health-po
stgres-1
5807ff34bfa9   wurstmeister/zookeeper:latest         "/bin/sh -c '/usr/sb…"   5 minutes ago   Up 5 minutes   22/tcp, 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp     diabetes_health-zo
okeeper-1
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\PROYECTO3\diabetes_health>
```

We then run the streamlit to display the dashboard in real time, with the data sent previously.

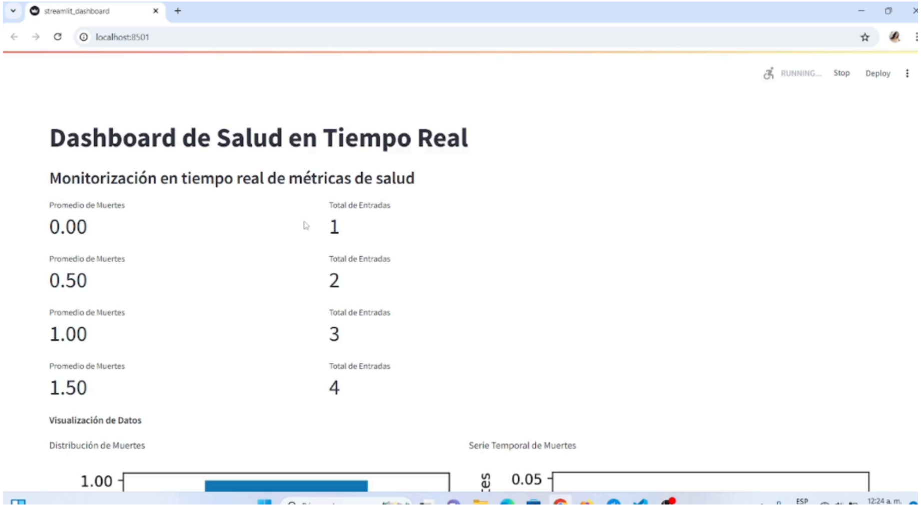


```
PS D:\Users\MAFE\Desktop\UNIVERSIDAD AUTÓNOMA DE OCCIDENTE\5TO SEMESTRE\ETL (EXTRACCIÓN, TRANSFORMACIÓN Y CARGA)\PROYECTO3\diabetes_health> streamlit run streamlit_dashboard.py
>>

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.1.5:8501
```

- Productor -> Kafka -> Consumidor (en Streamlit) -> Dashboard en tiempo real.

Some screenshots of the dashboard in real time.

In the following image we can see that this dashboard is designed for real-time monitoring of health metrics, specifically indicators such as the "Average Deaths" and the "Total Entries" of data. The dashboard allows users to view this data as it is received, updating live thanks to the integration with Kafka and Streamlit.



Dashboard de Salud en Tiempo Real

Monitorización en tiempo real de métricas de salud

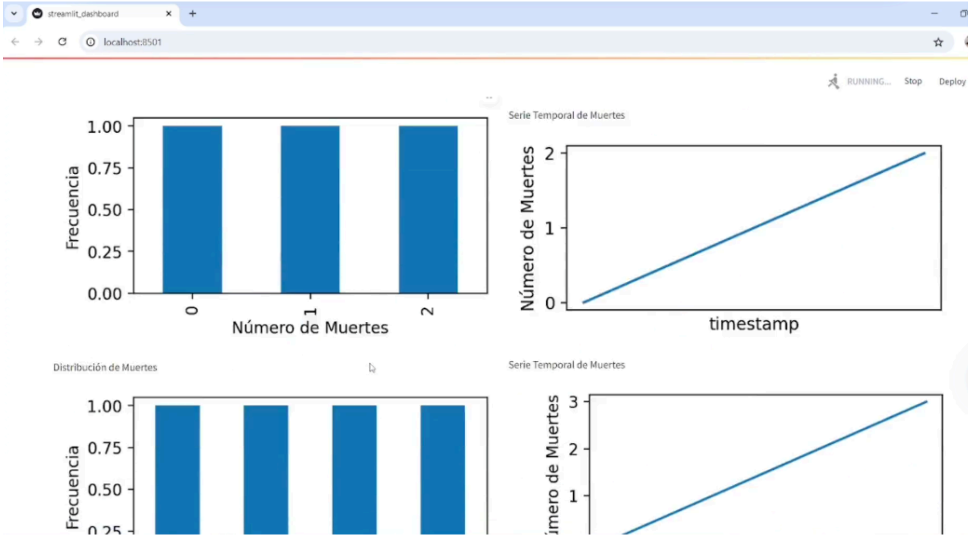
Promedio de Muertes	Total de Entradas
0.00	1
0.50	2
1.00	3
1.50	4

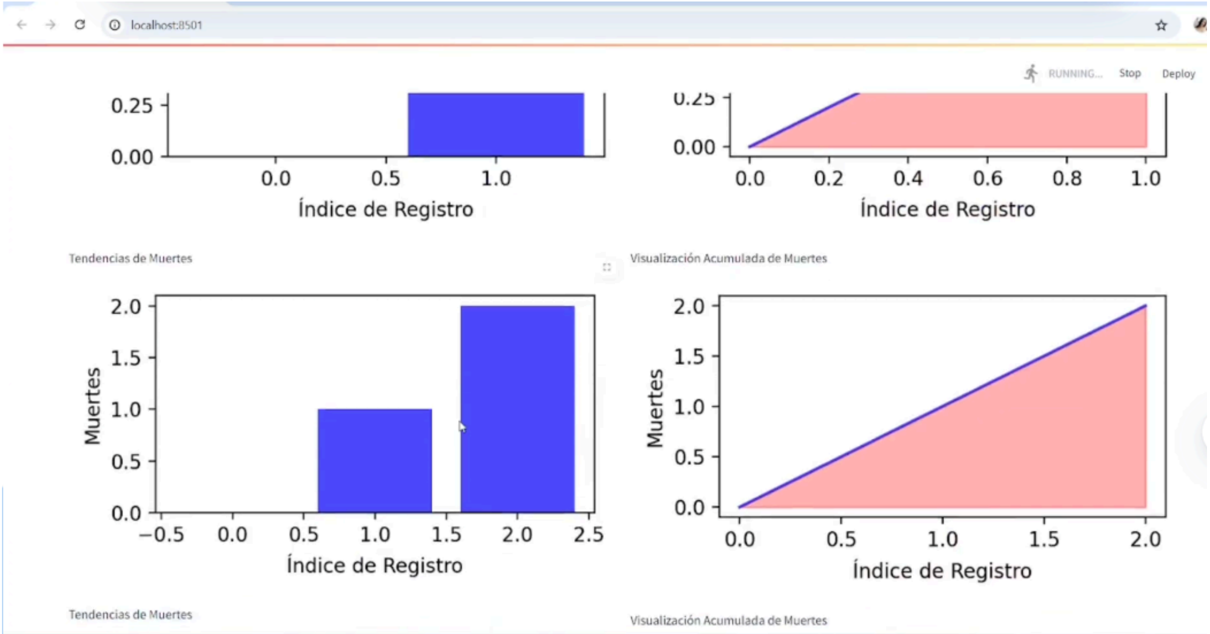
Visualización de Datos

Distribución de Muertes

Serie Temporal de Muertes

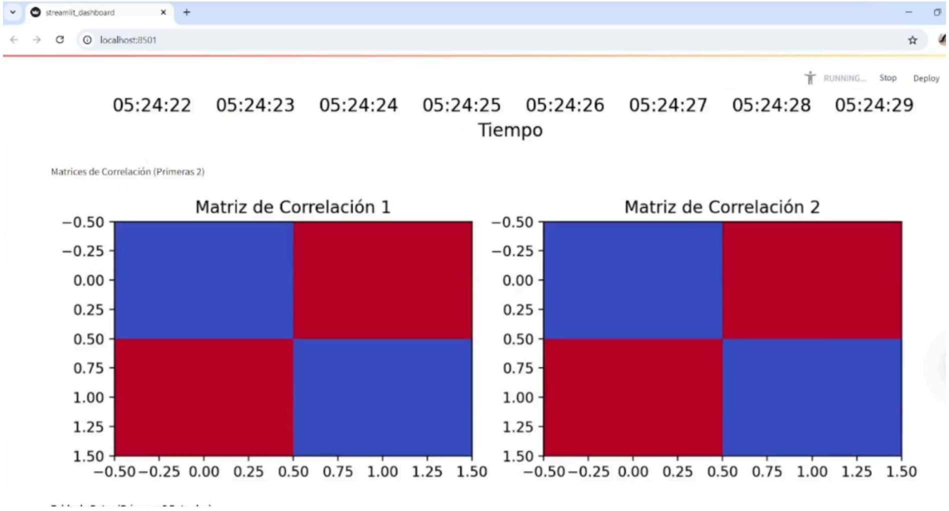
In the image below we can see that the first graph tells us the frequency of different values in the number of deaths recorded in real time. Each bar represents the number of times a specific value of deaths has appeared in recent data.
The second graph tells us the "Number of Deaths" as a function of time (timestamp). Each point on the line represents an update in the number of deaths as data is received in real time.





In the graph above we can see that the bar chart shows the frequency of deaths grouped by the "Record Index". Each bar represents the number of deaths for a specific value of the index at a given time. In addition, in the second graph we can see a combination of line and area graph showing the accumulation of deaths with the index. The line represents the trend and the shaded area shows the accumulation.

In the following graph we can see that the first correlation matrix visually represents the relationship between a set of health variables. The colors indicate the degree and direction of the correlation between pairs of variables.
Similar to Correlation Matrix 1, the second correlation matrix visualizes the relationships between another set of variables or the same set at a different time.



In the following image we can see that this table shows the first 5 entries of the real-time data received by the system. It is a quick view of the raw data that is being analyzed and visualized in the dashboard.

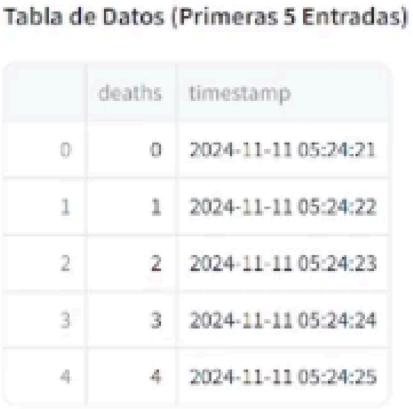
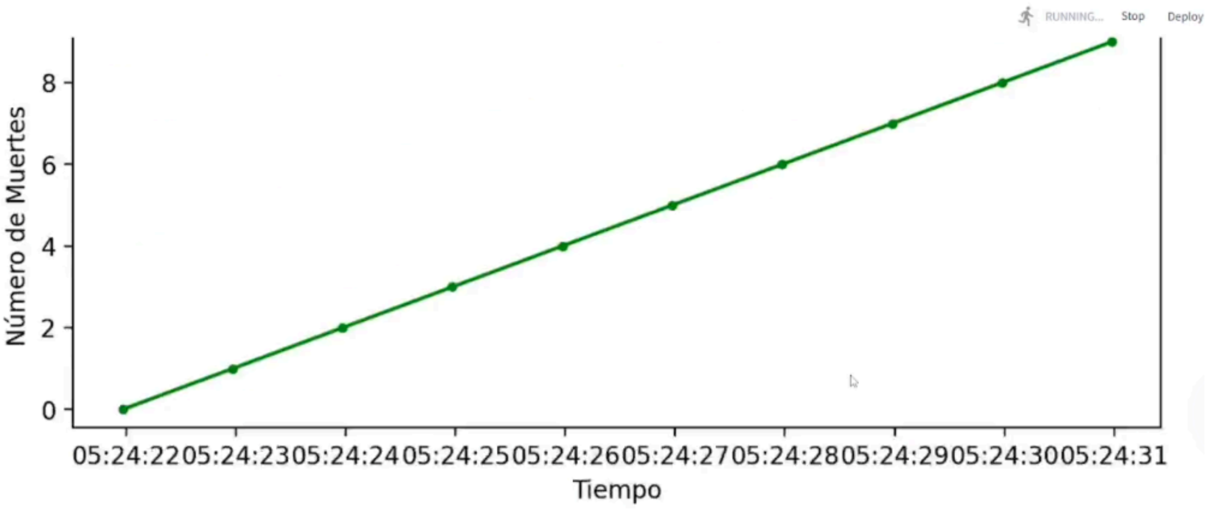


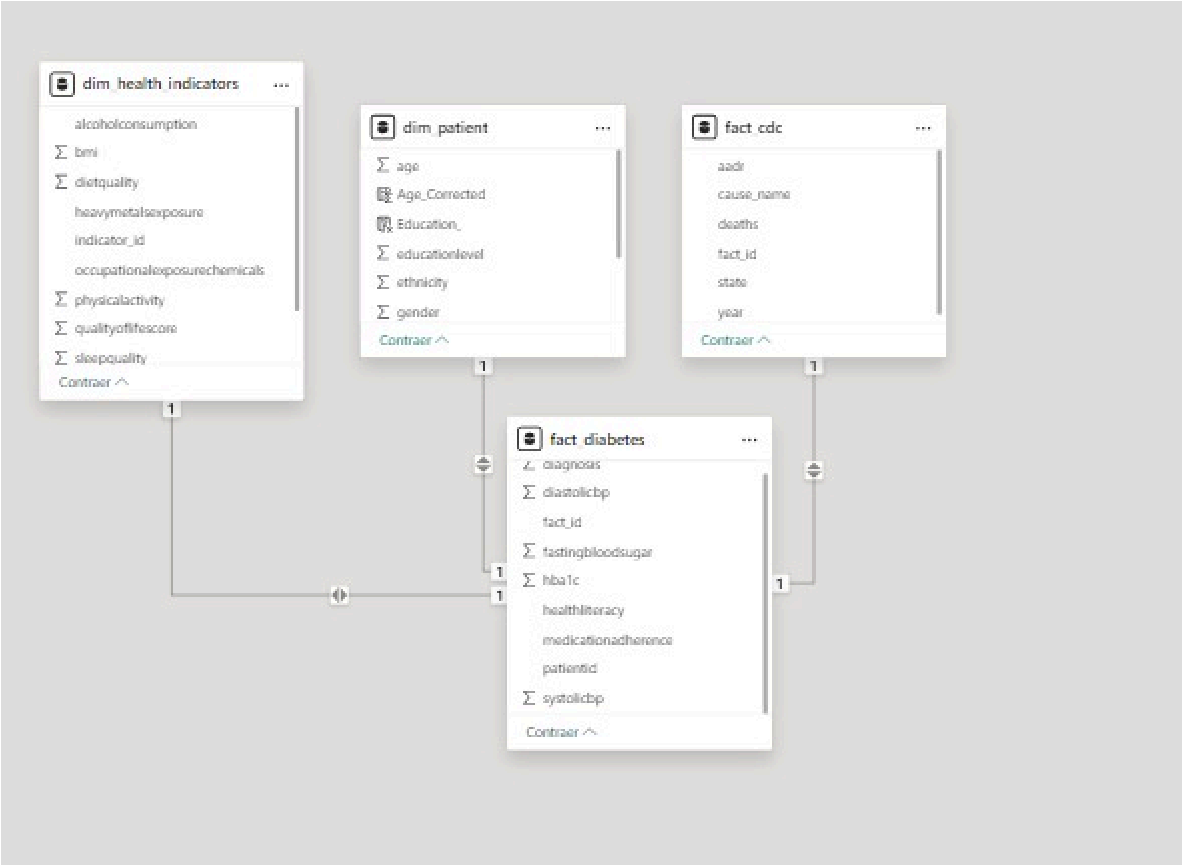
Tabla de Datos (Primeras 5 Entradas)

	deaths	timestamp
0	0	2024-11-11 05:24:21
1	1	2024-11-11 05:24:22
2	2	2024-11-11 05:24:23
3	3	2024-11-11 05:24:24
4	4	2024-11-11 05:24:25

The following line graph shows the evolution of the "Number of Deaths" over time, updating in real time as new data is received in the system. This graph is essential to monitor how health metrics vary continuously and detect possible growth patterns or anomalies.



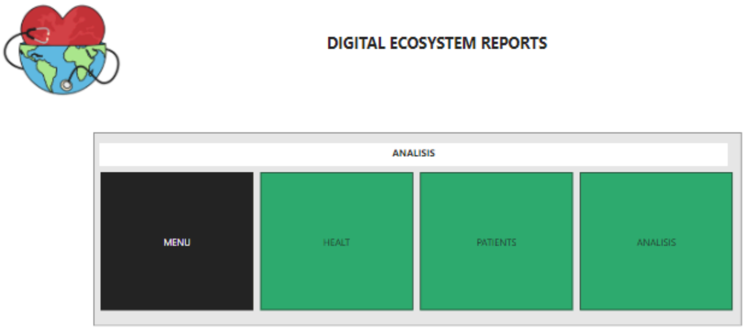
DIMENSIONAL MODEL



This dimensional model is structured to organize health data, specifically focused on diabetes metrics and causes of death. The model is composed of fact tables and dimensions that allow for detailed analysis and efficient queries in a data analytics system.

POWER BI BOARDS

First we have the menu which allows us to navigate through the other dashboards in the report.

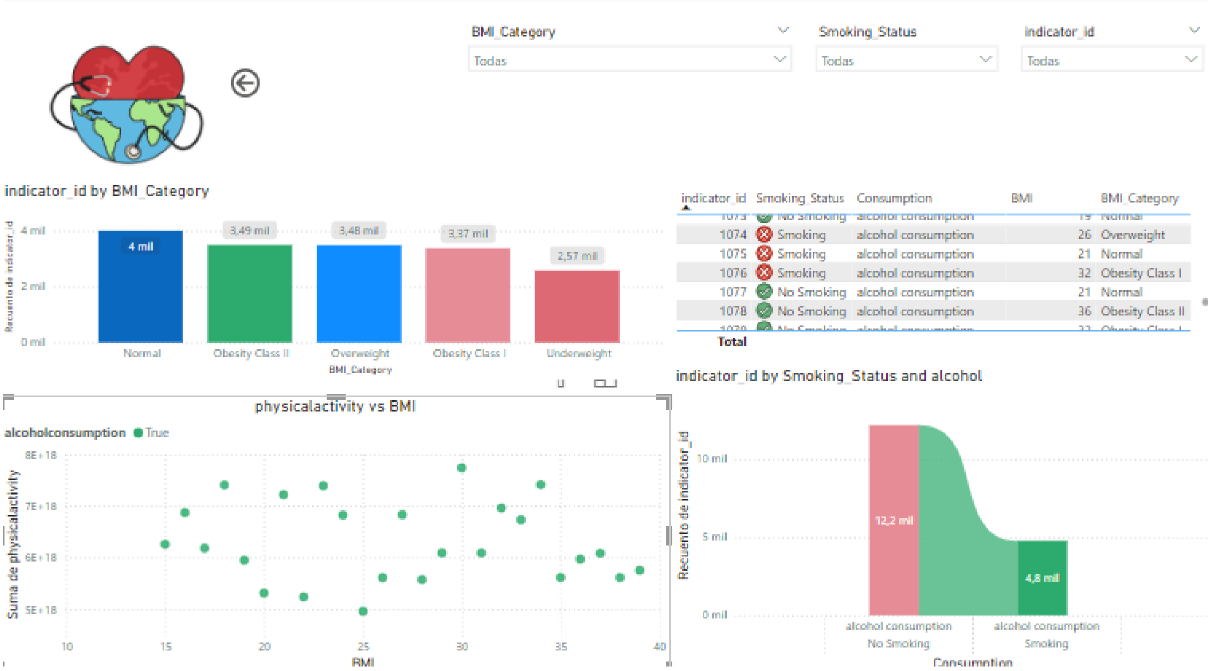


HEALTH BOARD

The dashboard makes it easier to understand how different factors such as BMI, smoking and alcohol consumption affect overall health.

The dashboard's objectives are:

- To facilitate understanding how patients are distributed across different BMI categories. This analysis is crucial for observing the prevalence of obesity, overweight and underweight in the population.
- To help analyze whether specific levels of physical activity are associated with particular BMI values, which is useful for assessing the impact of exercise on weight and overall health.
- To allow you to see if there is a relationship between alcohol consumption and smoking. This visualization is useful for assessing how these habits behave together and whether smokers have a greater or lesser tendency to consume alcohol compared to non-smokers.
- To provide a tabular view that allows you to review specific and segmented data, useful for further analysis or to filter data based on certain criteria.



BMI_Category: Todas

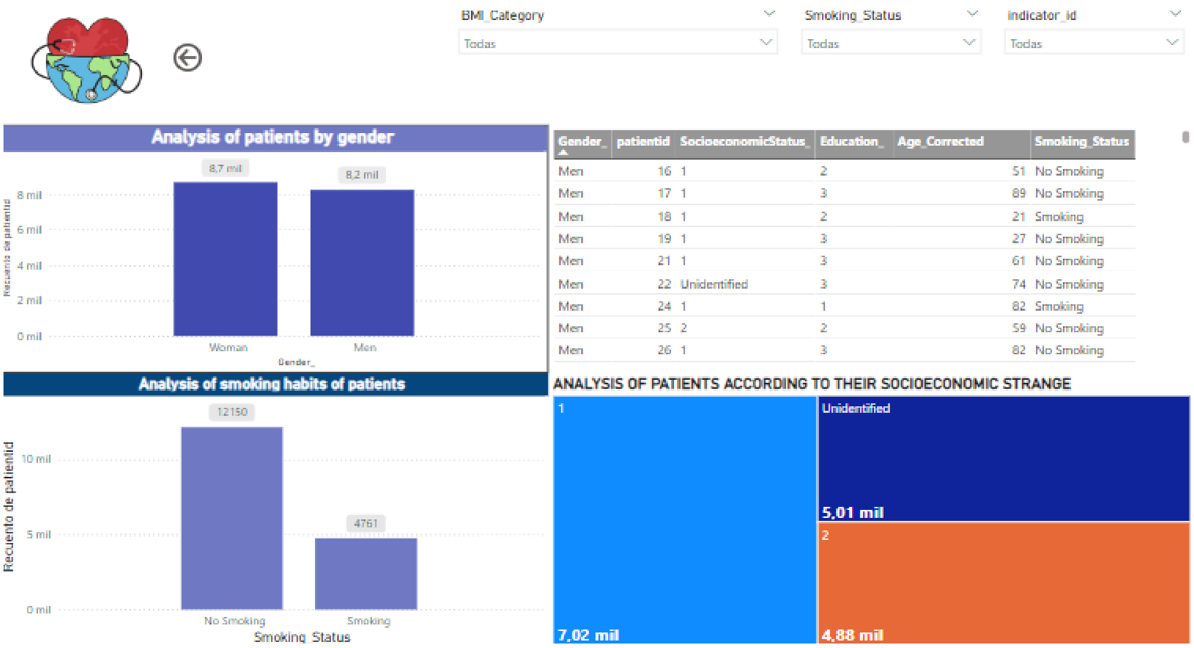
Smoking_Status: Todas

indicator_id: Todas

indicator_id	Smoking_Status	Consumption	BMI	BMI_Category
1073	✓ No Smoking	alcohol consumption	19	Normal
1074	☒ Smoking	alcohol consumption	26	Overweight
1075	☒ Smoking	alcohol consumption	21	Normal
1076	☒ Smoking	alcohol consumption	32	Obesity Class I
1077	✓ No Smoking	alcohol consumption	21	Normal
1078	✓ No Smoking	alcohol consumption	36	Obesity Class II
1079	✓ No Smoking	alcohol consumption	33	Obesity Class I
Total				

PATIENTS BOARD

This dashboard provides a detailed view of patient data, segmented by gender, smoking habits, and socioeconomic status. It allows users to visualize demographic and behavioral patterns that may be relevant for population health analysis and for designing public health interventions.



BMI_Category: Todas

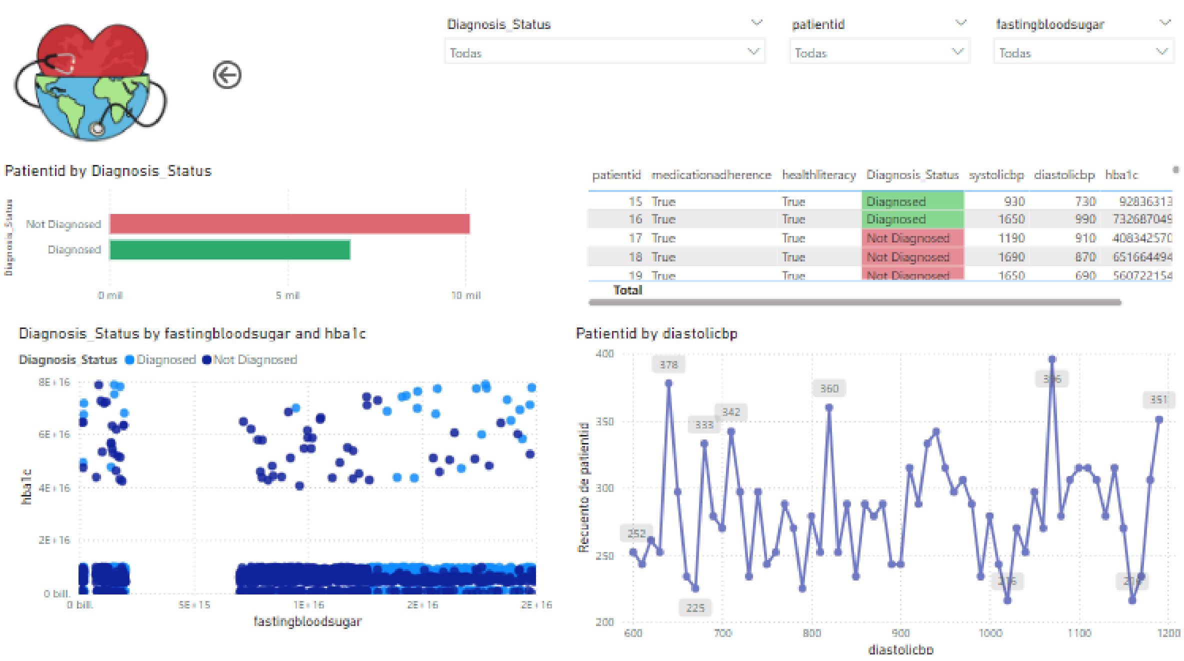
Smoking_Status: Todas

indicator_id: Todas

Gender_	patientid	SocioeconomicStatus_	Education_	Age_Corrected	Smoking_Status
Men	16	1	2	51	No Smoking
Men	17	1	3	89	No Smoking
Men	18	1	2	21	Smoking
Men	19	1	3	27	No Smoking
Men	21	1	3	61	No Smoking
Men	22	Unidentified	3	74	No Smoking
Men	24	1	1	82	Smoking
Men	25	2	2	59	No Smoking
Men	26	1	3	82	No Smoking

ANALYSIS BOARD

This dashboard allows you to view key information about patients' diabetes diagnostic status, fasting blood sugar levels, hemoglobin A1c, and diastolic blood pressure. This data is essential for analyzing patients' diabetes health status and for observing associated risk factors.



patientid	medicationadherence	healthliteracy	Diagnosis_Status	systolicbp	diastolicbp	hba1c
15	True	True	Diagnosed	930	730	92836313
16	True	True	Diagnosed	1650	990	732687049
17	True	True	Not Diagnosed	1190	910	408342570
18	True	True	Not Diagnosed	1690	870	651664494
19	True	True	Not Diagnosed	1650	690	560722154
Total						