

Documentación y evidencias del Proyecto SIATA IoT – Examen 3

Por: Maryangela Balcarcel Alarcón

En este documento se explicará como se dio solución al examen 3 de IoT.

Como primero, este es el entorno del proyecto:

```
ubuntu@ip-172-31-81-15:~$ ls
etl_app  examen3_IOT_SIATA  visualizacion_app
ubuntu@ip-172-31-81-15:~$
```

Las carpetas etl_app y visualización_app contienen lo que se va a necesitar para ejecutar el proyecto.

Este proyecto se compone de dos contenedores:

- El contenedor de ETL que realiza el proceso de extracción de la url siata.gov.co/EntregaData1/Datos_SIATA_Aire_AQ_pm25_Last.json donde se encuentran los datos de la calidad del aire desde SIATA, transformación para limpiar los datos y carga a una base de datos en mongo atlas. Debido a esto no existe contenedor de mongo.
- El contenedor de visualización a través de streamlit. Este contenedor se conecta a la base de datos de mongo atlas (donde ya se encuentran los datos cargados y limpios, y realiza un proceso de visualización).

La carpeta de ETL contiene:

```
ubuntu@ip-172-31-81-15:~$ ls etl_app/
Dockerfile  etl.py  requirements.txt
ubuntu@ip-172-31-81-15:~$
```

Un dockerfile, etl.py requirements.txt.

Para crear este contenedor se ejecuta:

```
docker build -t etl_app .
```

Y para correrlo:

```
docker run --name etl_container etl_app
```

NOTA: Para actualizar los datos en mongo se debe volver a ejecutar el etl.py, esto para realizar el proceso de carga nuevamente.

La carpeta de VISUALIZACIÓN contiene:

```
ubuntu@ip-172-31-81-15:~$ ls visualizacion_app/  
Dockerfile  requirements.txt  visualizacion.py  
ubuntu@ip-172-31-81-15:~$
```

Un dockerfile, visualizacion.py requirements.txt.

Un dockerfile, etl.py requirements.txt.

Para crear este contenedor se ejecuta:

docker build -t visualizacion_app .

Y para correrlo:

docker run -d --name visual_container -p 8501:8501 visualizacion_app

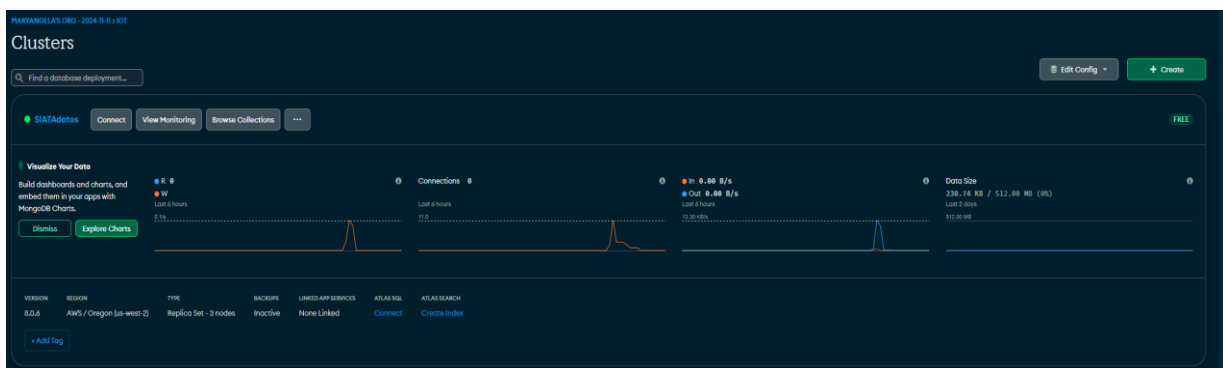
```
ubuntu@ip-172-31-81-15:~$ sudo docker run -d --name visual_container -p 8501:8501 visualizacion_app  
bd2764fe19a54e737091aa9436f24c40502eb54cff33206f3fc000f135c24301  
ubuntu@ip-172-31-81-15:~$
```

La visualización del streamlit se realiza por la siguiente URL: http://<ip_publica>:8501

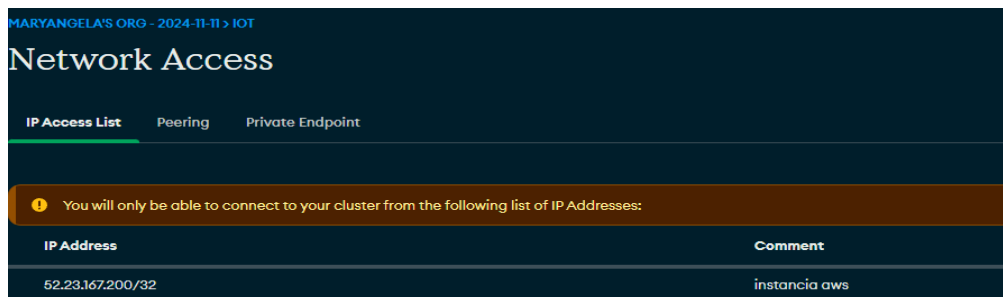
Pero... ¿Dónde están los datos?

Los datos se encuentran en un cluster de mongo atlas. Más específicamente en una colección llamada datos_siata

Clúster:



Una desventaja que evidencié de este método es que al momento de ver la visualización se debe añadir la IP pública de la instancia cada vez que esta cambie, en la configuración “Network Access” del clúster. Ya que sino, no tendría permisos para ingresar a los datos.



Colección de datos:

datos_siata.datos_siata

STORAGE SIZE: 72KB LOGICAL DATA SIZE: 196.28KB TOTAL DOCUMENTS: 367 INDEXES TOTAL SIZE: 62KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' }

QUERY RESULTS: 1-20 OF MANY

```
{
  "_id": "ObjectId('67f856412938d252adc72333')",
  "city": "Medellin",
  "attribution": Object,
  "sourceName": "SIATA",
  "country": "Colombia",
  "value": 31,
  "location": "CEN-TRAF - Estación Tráfico Centro",
  "date": Object,
  "averagingPeriod": Object,
  "coordinates": Object,
  "parameter": "pm25",
  "unit": "ug/m3"
}
```

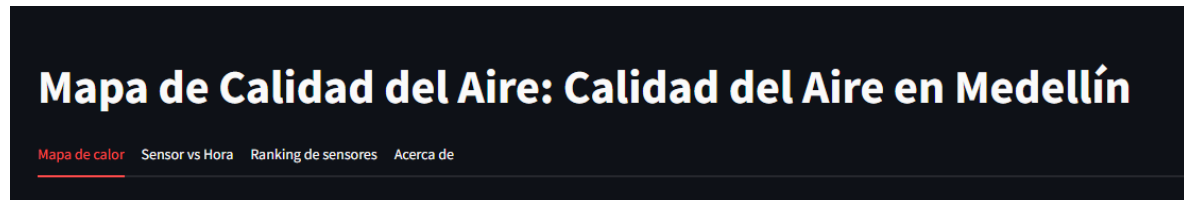
```
{
  "_id": "ObjectId('67f856412938d252adc72334')",
  "city": "Medellin",
  "attribution": Object,
  "sourceName": "SIATA",
  "country": "Colombia",
  "value": 38,
  "location": "CEN-TRAF - Estación Tráfico Centro",
  "date": Object,
  "averagingPeriod": Object,
  "coordinates": Object,
  "parameter": "pm25",
  "unit": "ug/m3"
}
```

```
{
  "_id": "ObjectId('67f856412938d252adc72335')",
  "city": "Medellin",
  "attribution": Object,
  "sourceName": "SIATA"
}
```

Dicho esto, aprendí que este proceso que es manual (cambio de IP en el network acces) se pudo haber simplificado creando un contenedor de mongo en el que se alojaran los datos simplemente, y así no tener inconvenientes con credenciales, conexión desde código y desde IP. Sin embargo, es una solución viable para proyectos con datos muy escalables.

Visualizaciones:

Al entrar en http://<ip_publica>:8501 se encontraá una visualización en streamlit. Contiene 4 tabs.

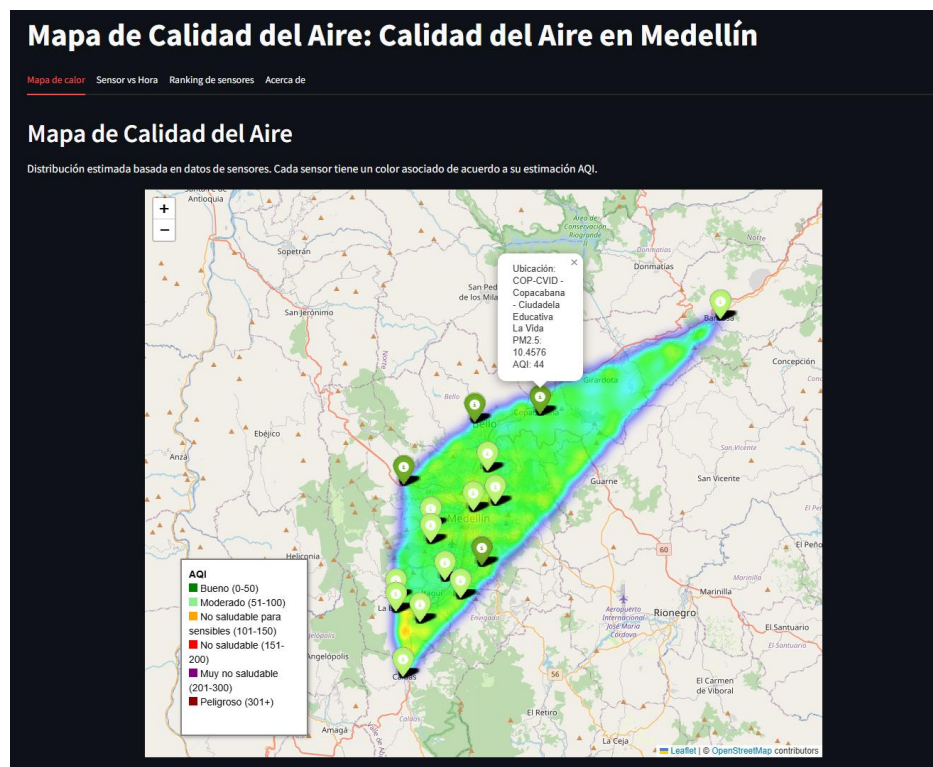


Mapa de calor:

En este tab se encuentran 2 mapas de calor:

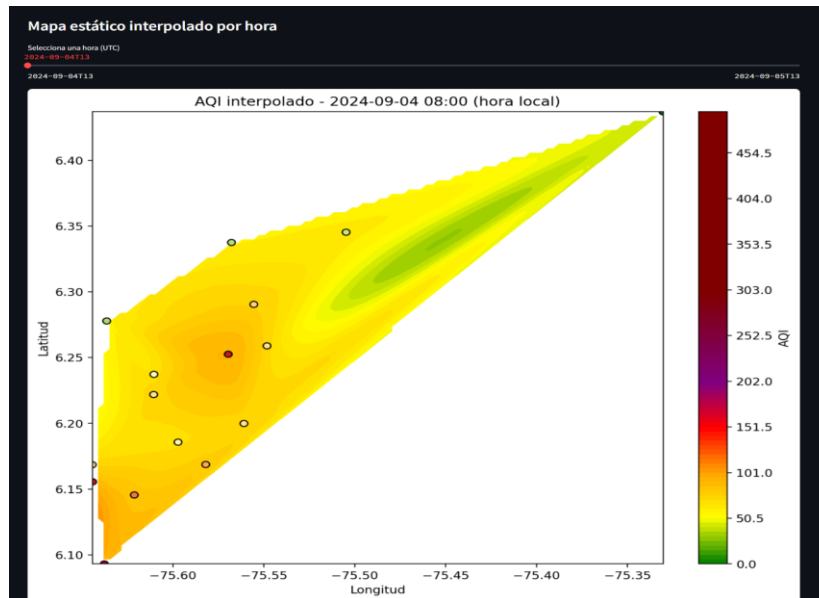
1. Mapa interactivo

Este mapa es interactivo y muestra los sensores con el correspondiente color de aqi. Y su información.



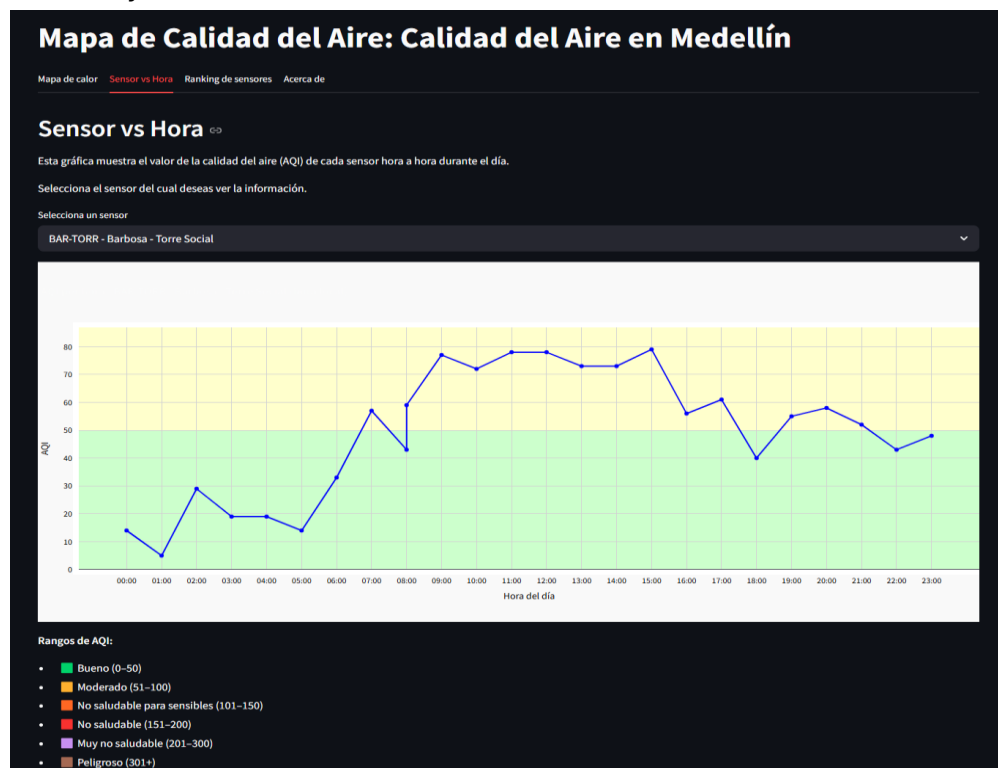
2. Mapa estático interpolado

Mas abajo, se encuentra este mapa que muestra con una interpolación cubica la información por hora seleccionada de los sensores en Medellín.



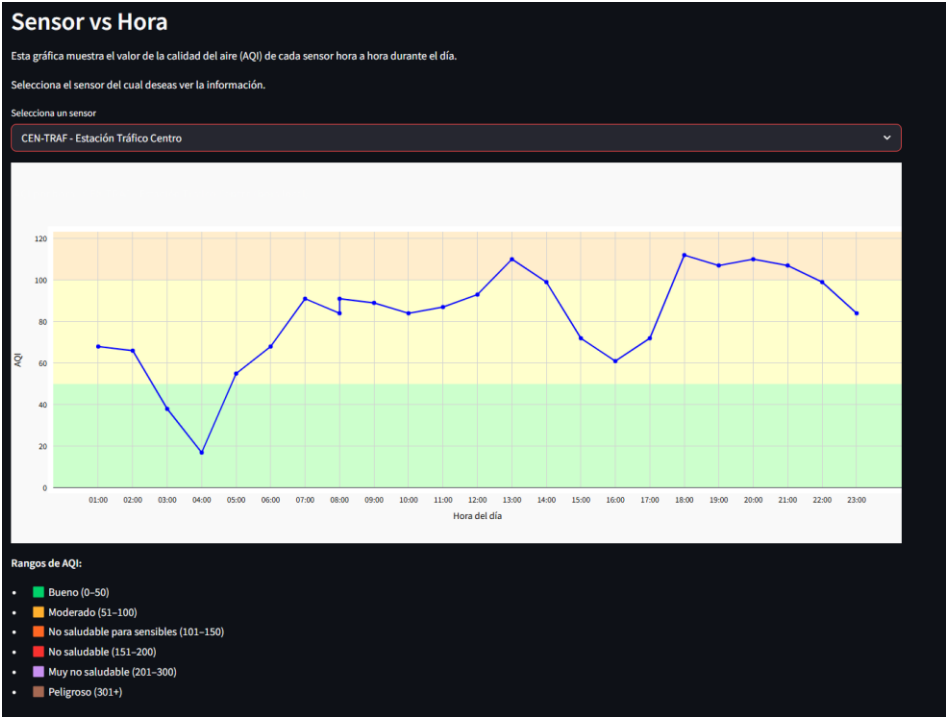
Sensor vs Hora

En este tab se muestra una grafica de aqi vs hora del día, en el que se puede seleccionar la estación y ver cómo es la variación durante el día.



Se le añadió colores de referencia al momento de cambiar la gráfica, para delimitar los rangos.

Ejemplo estación tráfico centro.



Ranking de sensores

En este tab se muestra un ranking del promedio en cuanto a aqi. En orden de peor a mejor estimación.

Mapa de Calidad del Aire: Calidad del Aire en Medellín

[Mapa de calor](#) [Sensor vs Hora](#) [Ranking de sensores](#) [Acerca de](#)

Ranking de Sensores por AQI promedio diario

Este ranking muestra los sensores ordenados según el promedio de AQI durante el día. Valores más altos indican peor calidad del aire.

	Sensor	AQI promedio	Clasificación
0	CEN-TRAF - Estación Tráfico Centro	81.83	Moderado
1	MED-ALTA - Medellín, Altavista - I.E. Pedro Octavio Amado	80.48	Moderado
2	EST-HOSP - La Estrella - Hospital	78.72	Moderado
3	ITA-CONC - Itagüí - I.E. Concejo Municipal de Itagüí	67.8	Moderado
4	ENV-HOSP - Envigado - E.S.E. Santa Gertrudis	67.68	Moderado
5	ITA-CJUS - Itagüí - Casa de Justicia Itagüí	67.44	Moderado
6	CAL-JOAR - Caldas - E U Joaquín Aristizabal	65.42	Moderado
7	SAB-RAME - Sabaneta - I.E. Rafael J. Mejía	65.2	Moderado
8	MED-TESO - Medellín, El Poblado - I.E INEM sede Santa Catalina	64.08	Moderado
9	MED-ARAN - Medellín, Aranjuez - I.E. Ciro Mendiá	61.96	Moderado
10	MED-BEME - Medellín, Belén - I.E. Pedro Justo Berrio	59.52	Moderado
11	BEL-FEVE - Bello - I.E. Fernando Vélez	58.48	Moderado
12	MED-VILL - Medellín, Villahermosa - Planta de producción de agua potable EPM	58.08	Moderado
13	MED-SCRI - Medellín, San Cristóbal - Parque Biblioteca Fernando Botero	57.05	Moderado
14	COP-CVID - Copacabana - Ciudadela Educativa La Vida	53.64	Moderado
15	BAR-TORR - Barbosa - Torre Social	49.4	Bueno

Acerca de:

En este tab se muestra la información del proyecto y cómo funciona la visualización.

Mapa de Calidad del Aire: Calidad del Aire en Medellín

Mapa de calorSensor vs HoraRanking de sensoresAcerca de

Acerca de

Este proyecto presenta una visualización interactiva y descriptiva de la **calidad del aire en Medellín**, basada en datos recopilados de sensores urbanos y procesados en tiempo real.

¿Qué vas a encontrar?

- Un **mapa interactivo** con interpolación espacial de los valores de AQI (Índice de Calidad del Aire), generado a partir de datos de sensores distribuidos en la ciudad.
- Un **mapa estático interpolado por hora**, que permite explorar la evolución espacial del AQI a lo largo del día, filtrando por hora específica.
- Una **gráfica AQI vs Hora** por sensor, que muestra cómo varía la calidad del aire a lo largo del día en una ubicación específica.
- EL **Ranking** de los sensores con peor y mejor calidad.

¿De dónde provienen los datos?

Los datos son extraídos desde una base de datos **MongoDB Atlas** que contiene registros de sensores del sistema **SIATA Medellín**, incluyendo:

- **Coordenadas geográficas** de cada sensor
- **Tiempos** en formato UTC
- **Concentraciones** de contaminantes (PM2.5)
- **Conversión a AQI** según los estándares de la EPA (Environmental Protection Agency, EE.UU.)

¿Qué tecnologías se están utilizando?

- **Python** como lenguaje principal
- **Streamlit** para la construcción del dashboard interactivo
- **MongoDB Atlas** para almacenamiento de datos en la nube
- **Folium** para mapas interactivos
- **Matplotlib** y **Plotly** para visualizaciones estáticas y dinámicas
- **Scipy** y **Numpy** para interpolación espacial
- **AQI personalizado** usando rangos y colores definidos por normativas internacionales

Objetivo

Este sistema permite visualizar de forma clara, interactiva y técnica la situación de calidad del aire en Medellín, facilitando el análisis ambiental para ciudadanos, investigadores o autoridades.

Y listo. Las graficas y mapas permiten conocer qué lugares y/o estaciones tienen mejor y peor calidad del aire, igualmente se muestran los rangos de aqi, por lo que se podrá tomar decisiones con esta información de colores y de valor aqi.

En el código de visualización se realizó la función de ecuación aqi y su clasificación.

Gracias!!