**INFORME DE RESULTADOS**

**Unidad 3 - Tarea 7**

**Peso: 20%**

**Equipo/Grupo : Numero 2**

**Estudiantes :**

* ***JUAN FELIPE BARRERA GARCIA***
* ***SOFIA RAMIREZ HIGUITA***
* ***JUAN DIEGO GALLARDO VILORIA***

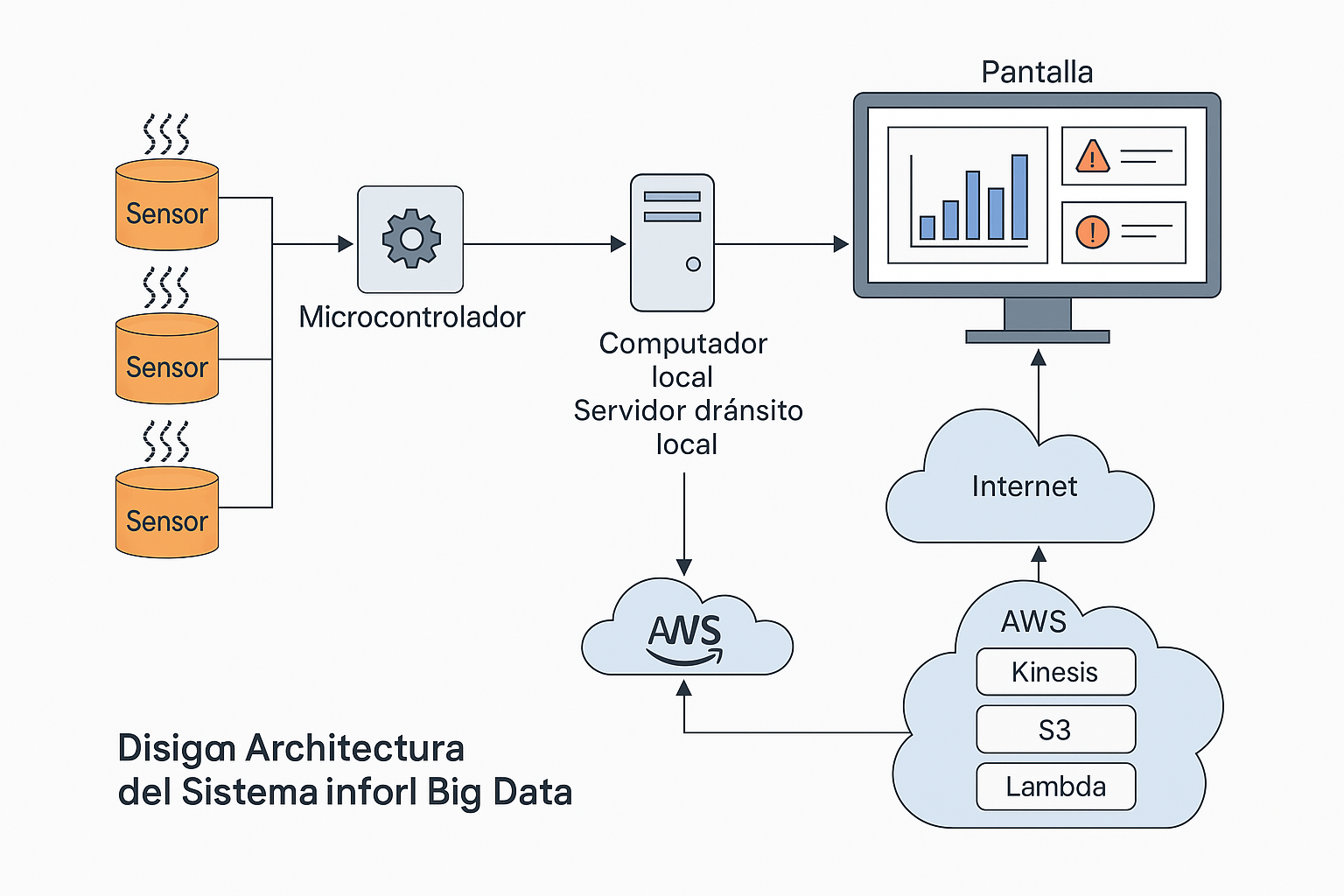
**Objetivo General**

**Diseñar una Arquitectura de Sistema de Información para el procesamiento de Big Data** que esté estructurado en una serie de componentes, tanto hardware como software, que permitan la implementación de un Proceso ETL (Extracción, Transformación, Carga), Analítica, Visualización de información del monitoreo, y la generación de alarmas durante el proceso de producción y emisión de gases tóxicos de las fábricas de la empresa “Sustancias Locas”; en particular, el Benceno que representa un peligro para la Salud humana y el medio ambiente.

Adicionalmente, se plantea la posibilidad de inclusión del total de fábricas de la empresa. La cantidad de datos que se incorporan con la integración de todas la fábricas, requiere de una modificación a la Arquitectura original para incluir mayor capacidad y potencia de cómputo. En este caso, la evaluación de Hadoop como sistema de almacenamiento distribuido y escalable.

En ese mismo orden de ideas, se requiere una prueba de esfuerzo y rendimiento del sistema para llevar a cabo almacenamiento y visualización en tiempo real a través de una simulación del proceso completo: ETL, Analítica y Visualización..

**1.- Diseño Arquitectura del Sistema de Información** (para procesamiento Big Data)

**

**2.- Cálculos de procesamiento y almacenamiento en la tabla lecturas**

*Colocar en esta sección los resultados de los cálculos. Nota: utilizar el tamaño de una tupla de la tabla de lecturas:*

**2.1.-Cálculos del procesamiento de lecturas**

|  |  |  |
| --- | --- | --- |
| **Tipo de Lectura** | **Cálculos** | **Total lecturas** |
| Lecturas diarias de un sensor | 60lecturas/hora×24horas/día | 1,440 lecturas/día |
| Lecturas de un mes de una línea de producción. Nota: un mes = 30 días. | 1,440×4sensores×30día | 172,800 lecturas/mes |
| Lecturas de un año de todas las líneas de producción de todas las fábricas. Nota: un año = 365 días. | 1,440×4sensores×4lıˊneas×3fabricas×365días | **63,072,000 lecturas/año** |

***Nota: “Cálculos” debe mostrar el procedimiento para calcular el total de lecturas***

**2.2.- Almacenamiento en tabla “lecturas”**

|  |  |  |  |
| --- | --- | --- | --- |
| **Período de almacenamiento** | **Lecturas** | **Tamaño Tupla** | **Total Bytes** |
| Un (1) minuto | **60** | 50 | |  | | --- | |  |   3,000 Bytes |
| Un (1) hora | **3,600** | 50 | 180,000 Bytes (≈ 176 KB) |
| Un (1) día | **86,400** | 50 | 4,320,000 Bytes (≈ 4.1 MB) |
| Un (1) mes | **2,592,000** | 50 | 129,600,000 Bytes (≈ 123.6 MB) |
| Un (1) año | **31,536,000** | 50 | 1,576,800,000 Bytes (≈ 1.47 GB) |

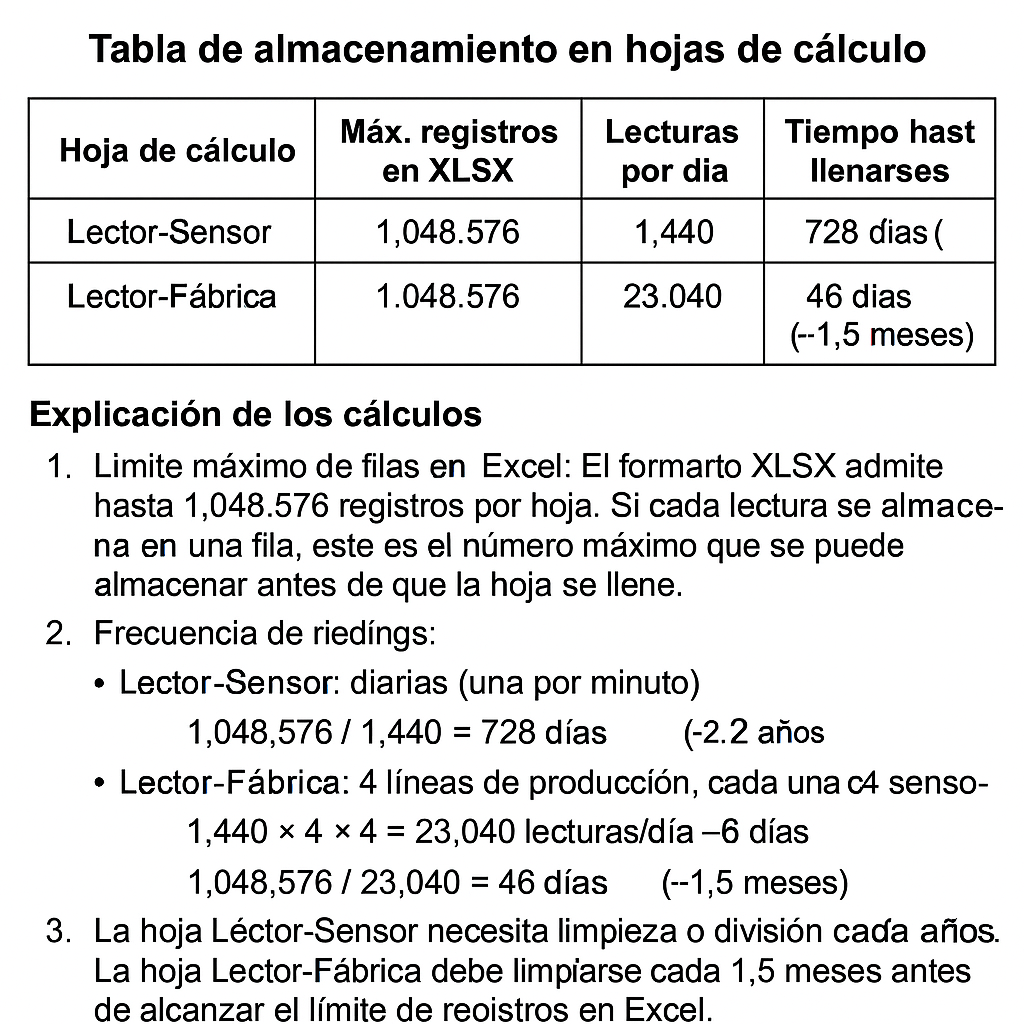
***Nota: “Tamaño Tupla” es el tamaño en BYTES del registro de la tabla “lectura” de la base de datos “monitoreo-produccion” de PostgreSQL***

**2.3.- Almacenamiento en tabla “lecturas”**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lote de “lecturas”** | **Lecturas** | **Tamaño Tupla** | **Total Bytes** |
| Lote 1 | **3.000.000** | 50 | 150,000,000 Bytes (≈ 143 MB) |
| Lote 2 | **20.000.000** | 50 | 1,000,000,000 Bytes (≈ 953 MB) |

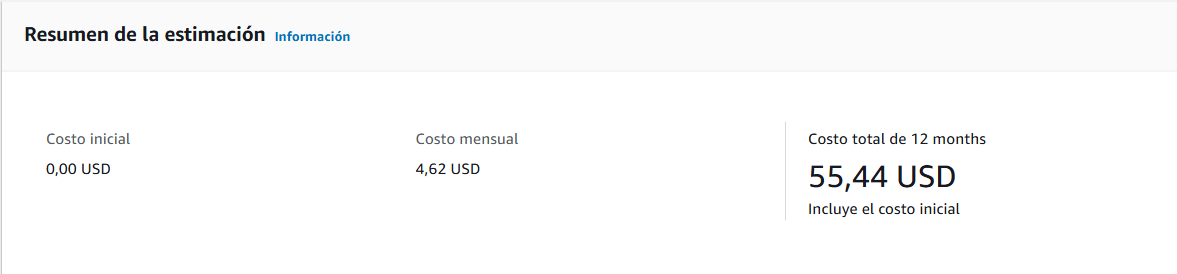
***Nota: “Tamaño Tupla” es el tamaño en BYTES del registro de la tabla “lectura” de la base de datos “monitoreo-produccion” de PostgreSQL***

***2.4.- ¿Cada cuánto tiempo se debe limpiar la hoja de cálculo “lector-sensor” y la hoja de cálculo “lector-fabrica” antes de que se llegue al límite del máximo de registros permitidos por hoja de cálculo con formato “xlxs? Explique brevemente como realizó ellos cálculos.***

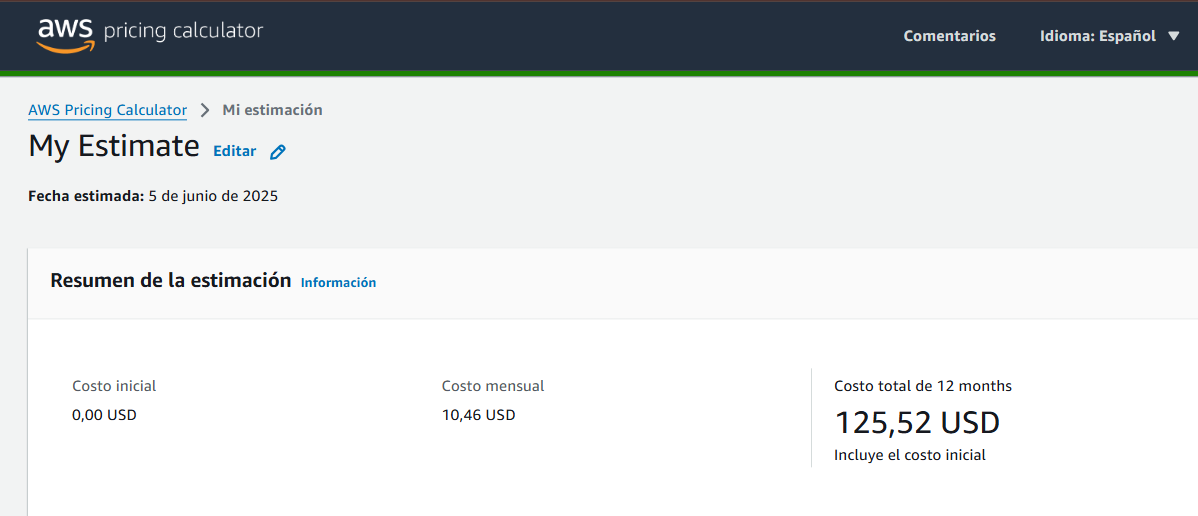
**

**3.- Cálculo del costo de almacenamiento de en AWS de 20 millones de registros.**

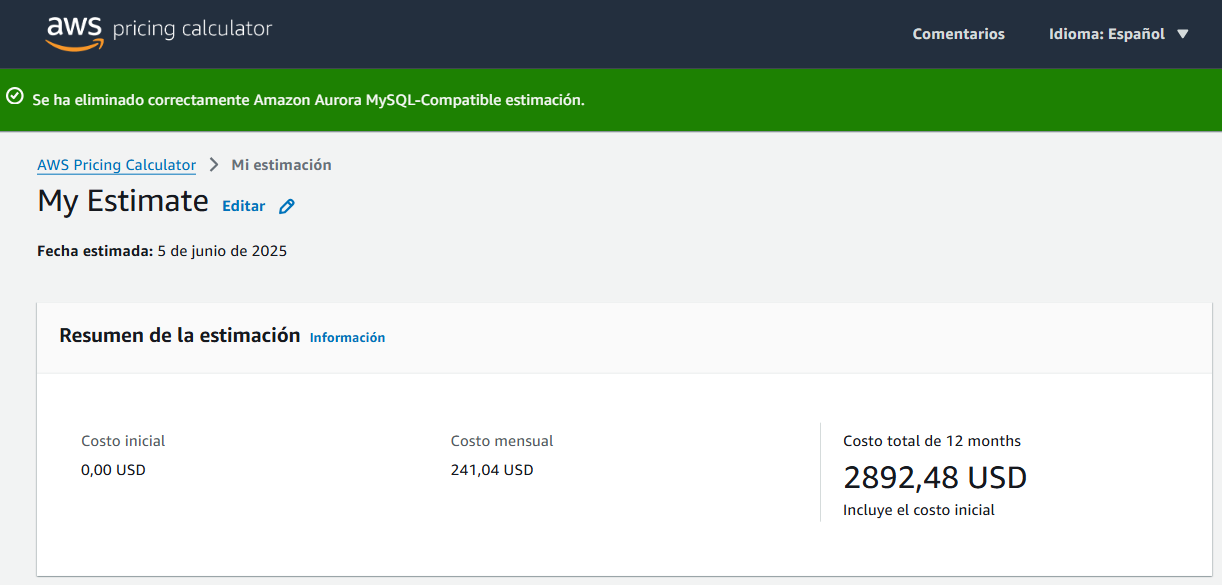
*General Purpose SSD (gp3)*

**

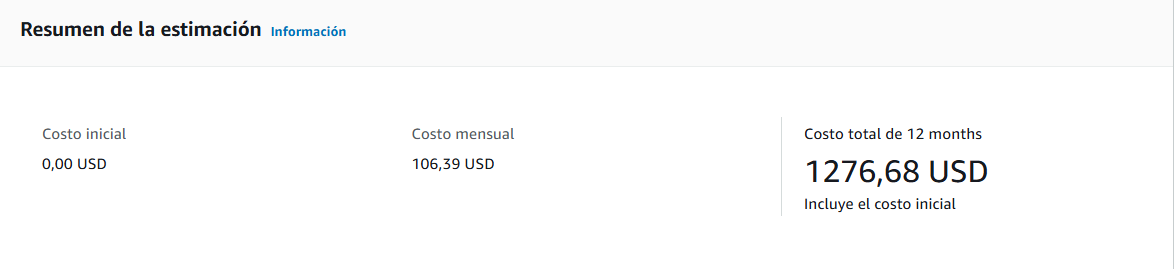
*AMAZON S3*

**

*AMAZON AURORA*

**

*Cold HDD (sc1)*

**

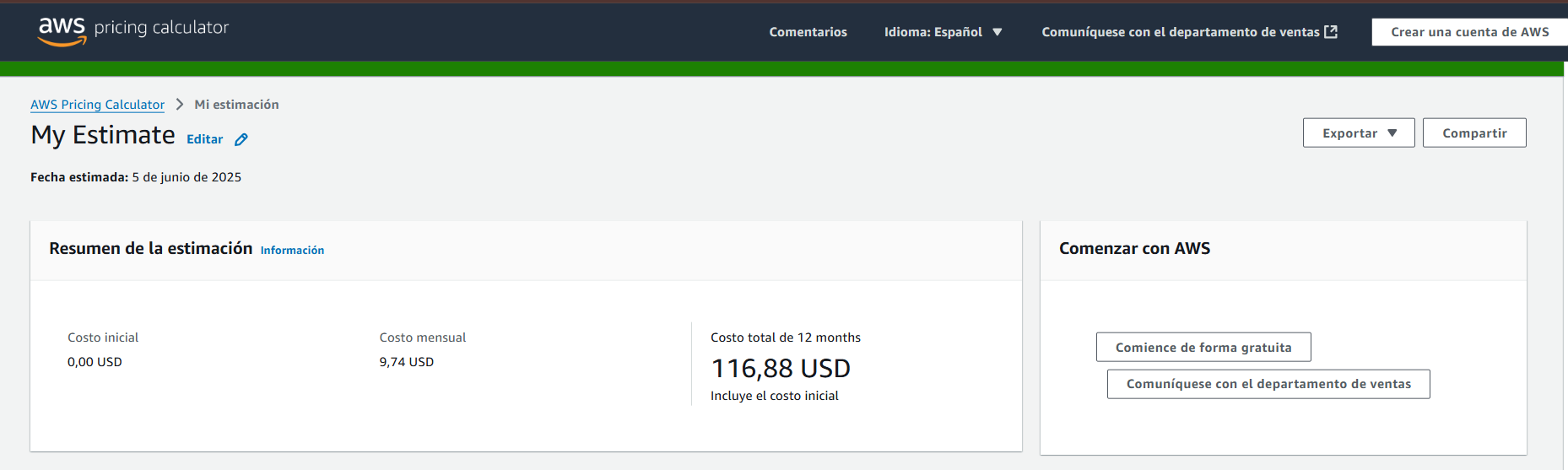
**3.1.- Cálculo de almacenamiento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo Almacenamiento** | **Total Bytes** | **Costo x byte** | **Costo Total** |
| Almacenamiento en Bloque Elástico (EBS) | 922,746,880 | $0.08 / GB | $9.74 USD |
| Amazon Simple Storage Service (Amazon S3) | 922,746,880 | $0.023 / GB | $10.46 USD |
| Amazon Aurora | 922,746,880 | $0.26/GB aprox. | $241.04 USD |

**3.2.- ¿Por qué la diferencia de costos en los diferentes tipos de almacenamiento?**

* **EBS** es un almacenamiento en bloque, rápido y de bajo costo, ideal para discos de sistemas y bases de datos simples, acceso rápido para los datos.
* **Amazon S3** es almacenamiento de objetos, más barato y escalable, pero no apto directamente para bases de datos relacionales (ideal para backups, archivos, etc.) diseñada para almacenamiento de objetos a largo plazo.
* **Amazon Aurora** es un servicio administrado de base de datos relacional que incluye almacenamiento, respaldo, escalabilidad y rendimiento avanzado. Su costo es mucho más alto porque incluye recursos de cómputo, gestión automática, redundancia, alta disponibilidad, optimización avanzada de consultas.

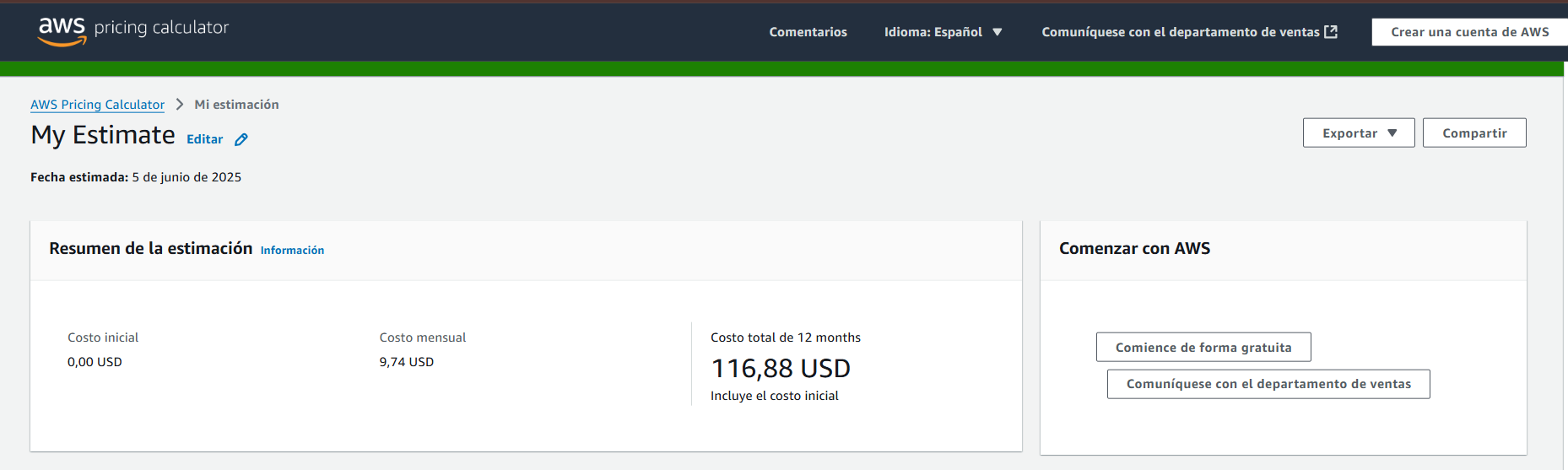
**3.3.- Cotización AWS para el tipo de almacenamiento en EBS**

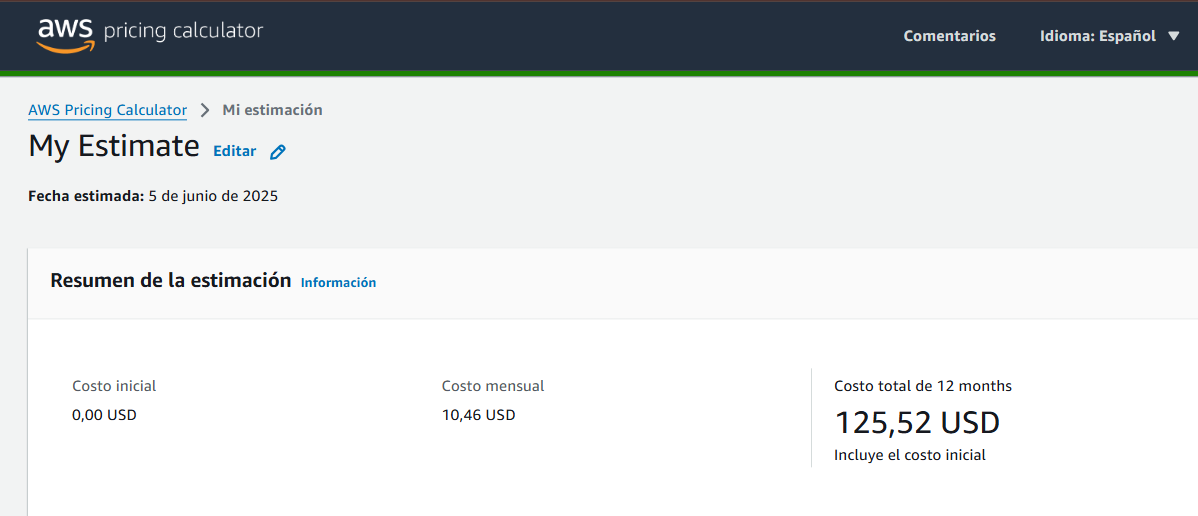
**

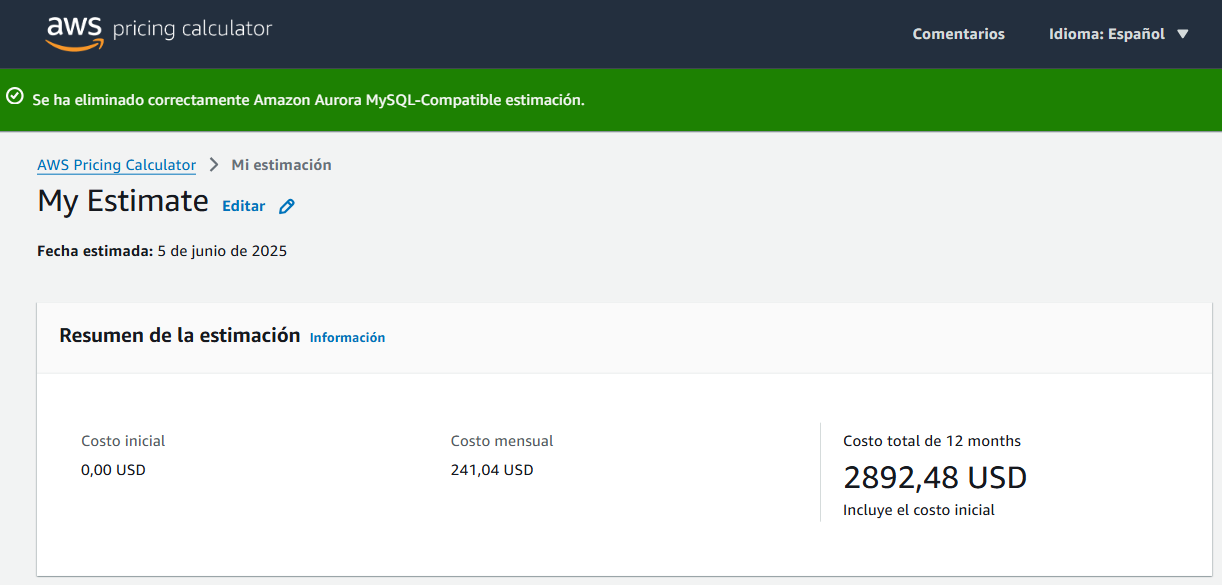
**4.- Cálculo del costo por 1 año de uso de los servicios AWS**

*Se estimó el costo anual considerando los siguientes servicios principales utilizados para almacenar y gestionar 20 millones de registros de la tabla “Lectura”:*

* ***Amazon EBS****: utilizado como almacenamiento de bloques para una instancia EC2.* ***Costo total por 12 meses: 116,88 USD****.*

**

* ***Amazon S3****: almacenamiento de objetos, útil para respaldos o archivos exportados.* ***Costo total por 12 meses: 125,52 USD****.  
  *
* ***Amazon Aurora****: utilizado como base de datos relacional para almacenar los datos con integridad y soporte SQL.* ***Costo total por 12 meses: 2.892,48 USD****.*

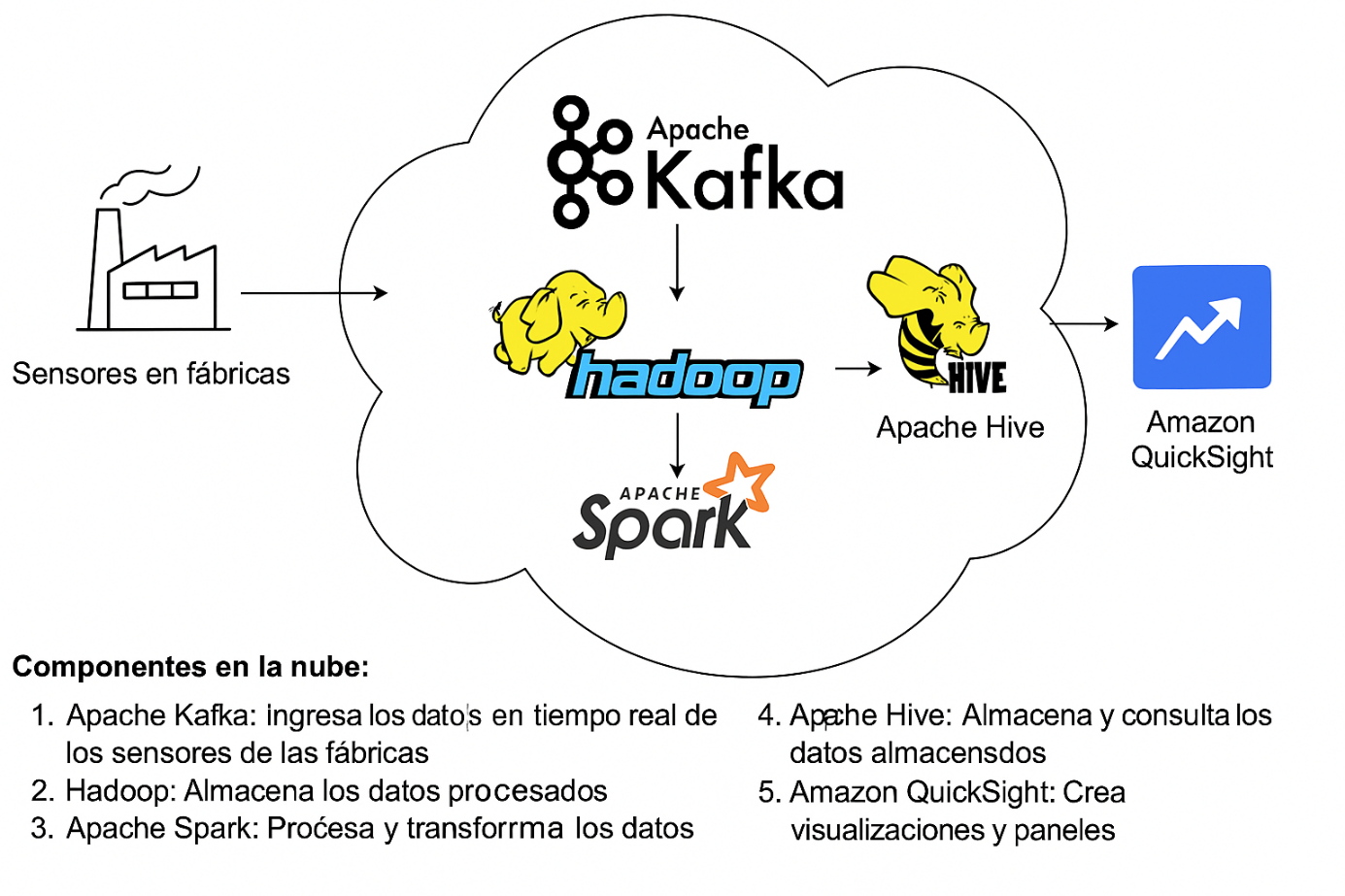
**

***Costo total de uso por 1 año en AWS: 3.134,88 USD***

**4.1.- Cálculo del costo de los servicios**

|  |  |  |
| --- | --- | --- |
| **NOMBRE SERVICIO** | **COSTO INDIVIDUAL** | **COSTO ANUAL** |
| Almacenamiento en Bloque Elástico (EBS) | **9,74 *USD****.* | ***116,88 USD****.* |
| **Amazon S3 – Almacenamiento estándar** | **10.46*USD****.* | ***125,52 USD*** |
| **Amazon Aurora – MySQL compatible** | **241,04 *USD****.* | ***2.892,48 USD****.* |
| **Amazon EBS – General Purpose SSD (gp3)** | **4,62USD** | ***55,44USD*** |
| **Amazon EBS – Cold HDD (sc1)** | **106,39USD** | ***1,276.48USD*** |

**5.- Diseño de Arquitectura de Sistema de Información Big Data con Hadoop**



1. **Ingesta de Datos con Apache Kafka**

* Recibe los datos en tiempo real desde sensores en las fábricas que monitorean emisiones de Benceno.
* Actúa como una plataforma de mensajería distribuida para garantizar que los datos lleguen correctamente al sistema en la nube.

1. **Procesamiento y Almacenamiento con Hadoop + Apache Spark**

* Apache Spark toma los datos transmitidos por Kafka y los procesa en tiempo real. Se realizan cálculos de normalización y limpieza.
* Hadoop se encarga del almacenamiento distribuido y escalable de los datos procesados, asegurando capacidad para manejar el crecimiento de información cuando se integren más fábricas.

1. **Gestión y Consulta con Apache Hive**

* Apache Hive funciona sobre Hadoop para permitir consultas SQL estructuradas sobre los datos históricos almacenados.
* Facilita la extracción de información clave como tendencias en emisiones de Benceno en diferentes períodos.

1. **Visualización en Amazon QuickSight**

* Amazon QuickSight consume los datos procesados desde Hadoop/Hive para mostrar informes y dashboards interactivos.
* Permite la generación de gráficos sobre variaciones de emisiones y posibles alertas de peligro.

**6.- Algoritmo de “poblamiento” de la tabla “lecturas” y la hoja de cálculo “lecturas-sensor”**

**EXPLICACION:**Se realiza el algoritmo PYTHON para el poblamiento de las tablas **lecturas y lecturas-sensor.**

La tabla **lectura-sensor** se crea como una vista para su manejo posterior en la realizacion del tablero de control en metabase, pero surge un problema en el computador de FELIPE, en el cual no le quiere reconocer la base de datos ni el codigo PYTHON por un error que sale en VSC.

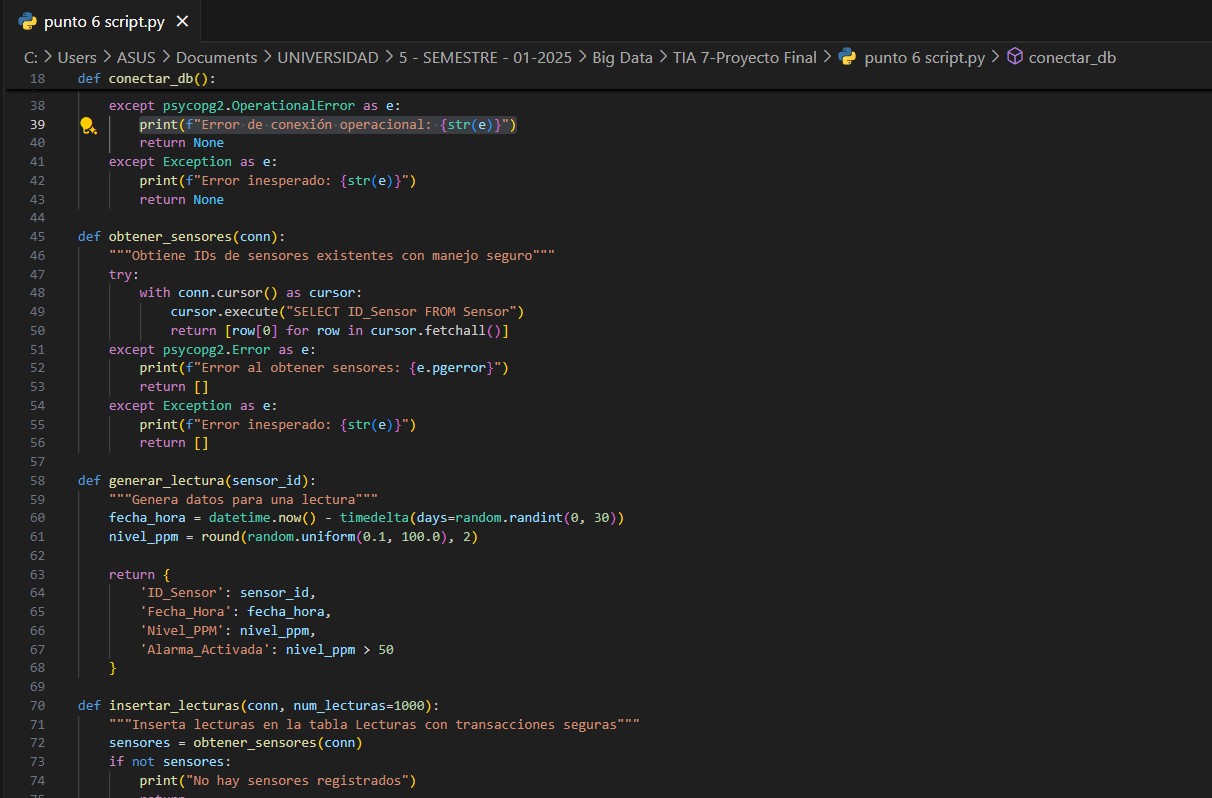
Por ende, no se ha podido tener una insercion como se deseaba en la base de datos, FELIPE el que realiza estos dos puntos opta por insertar datos por medio de INSERTS.

Se anexan pantallazos del algoritmo de poblamiento, pantallazos del error, pantallazos de la base de datos, scripts de creacion de base de datos y tablas, inserts realizados para el posterior manejo en metabase y EXPLICACION EN EL VIDEO PRESENTADO POR FELIPE.

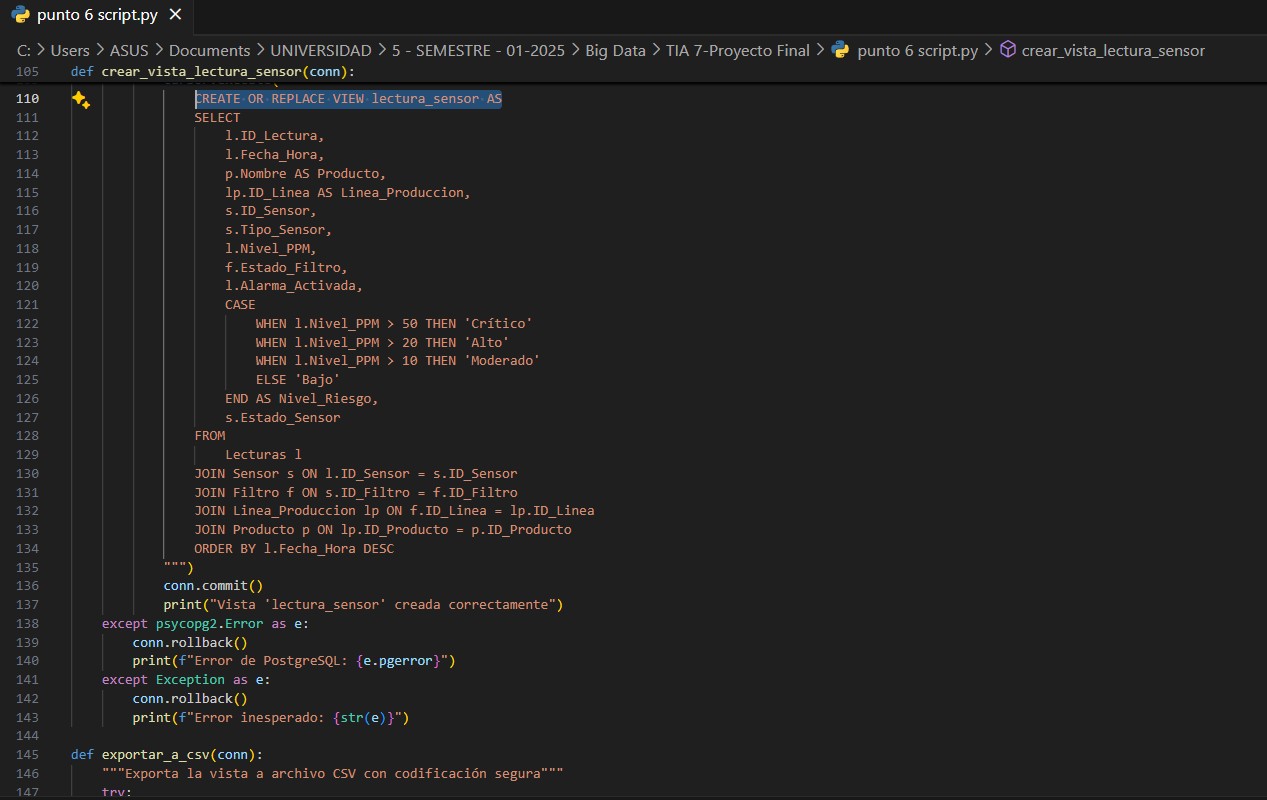
**PANTALLAZOS DEL ALGORITMO DE POBLAMIENTO**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto.**

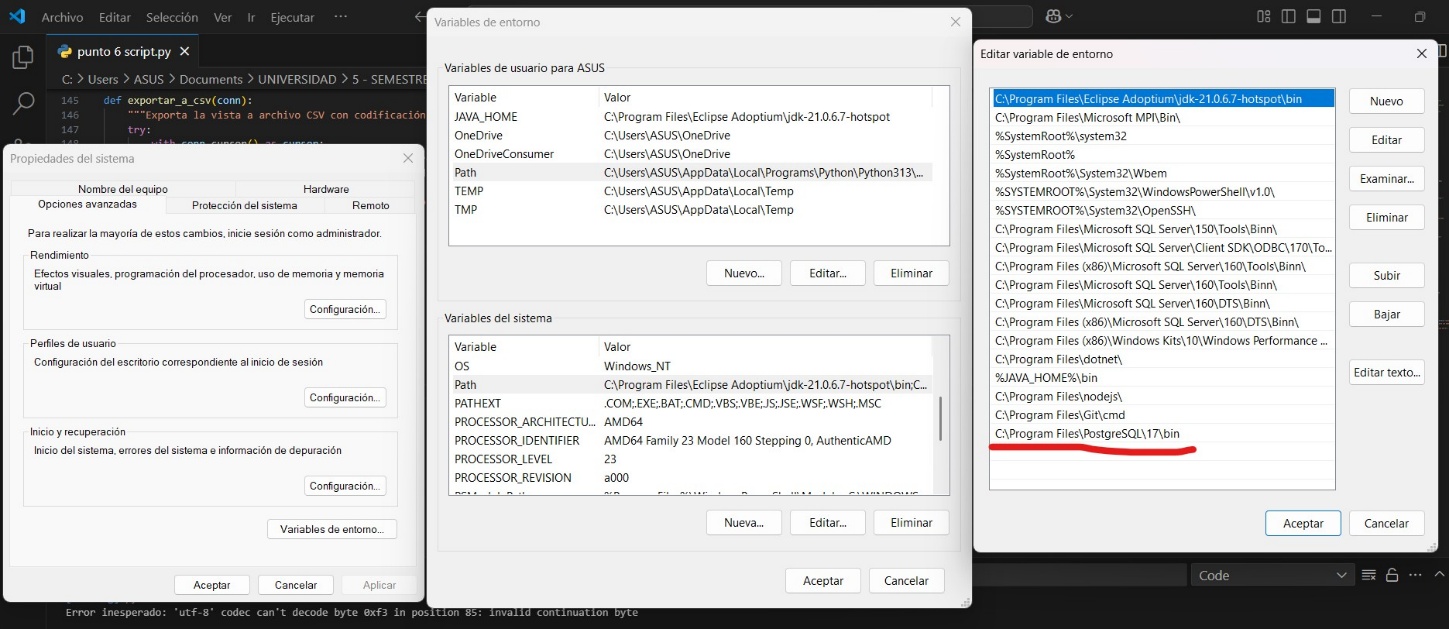
**Texto

El contenido generado por IA puede ser incorrecto.**

**PANTALLAZOS DEL ERROR GENERADO EN VSC**

**Interfaz de usuario gráfica, Aplicación

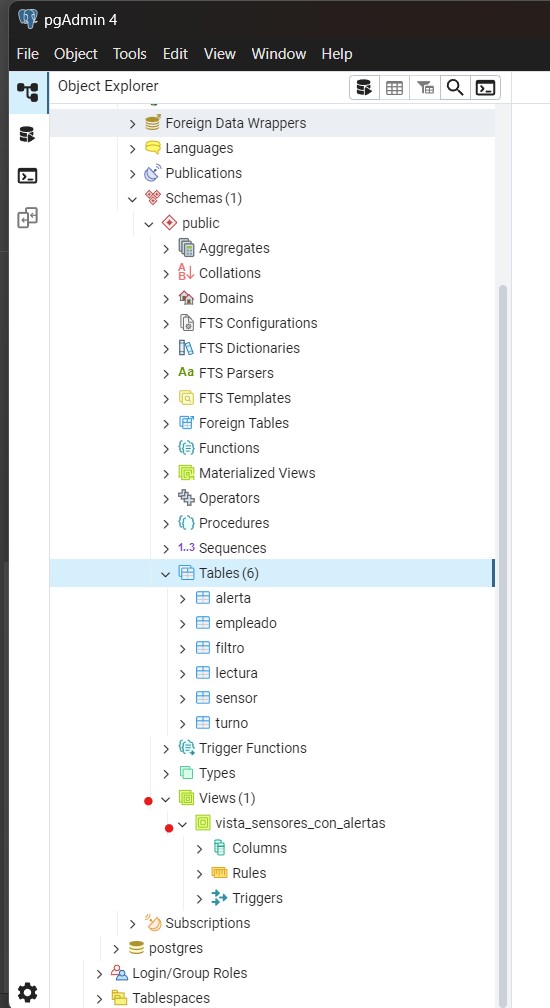
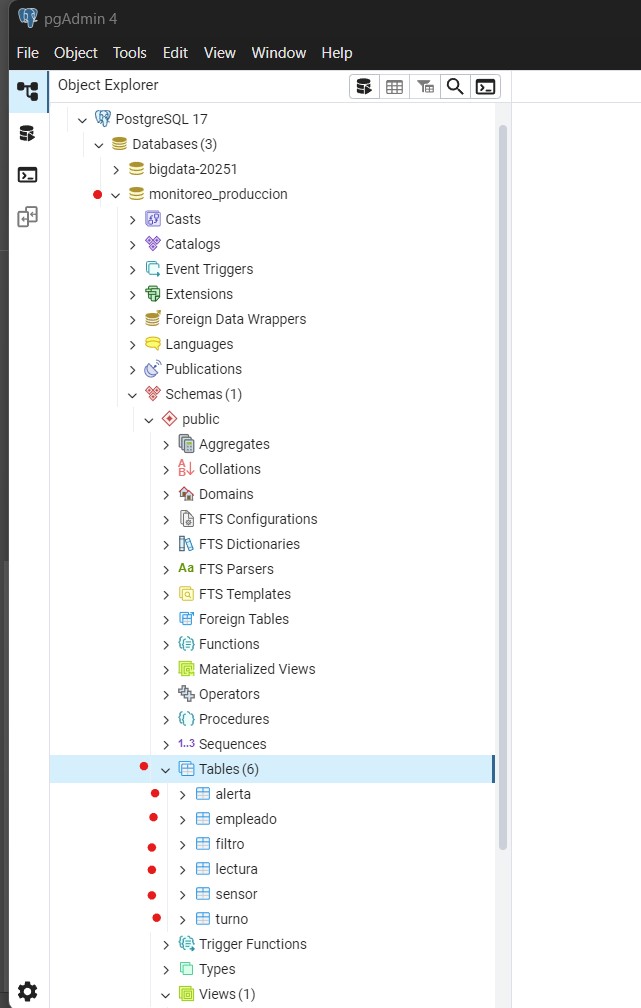
El contenido generado por IA puede ser incorrecto.**

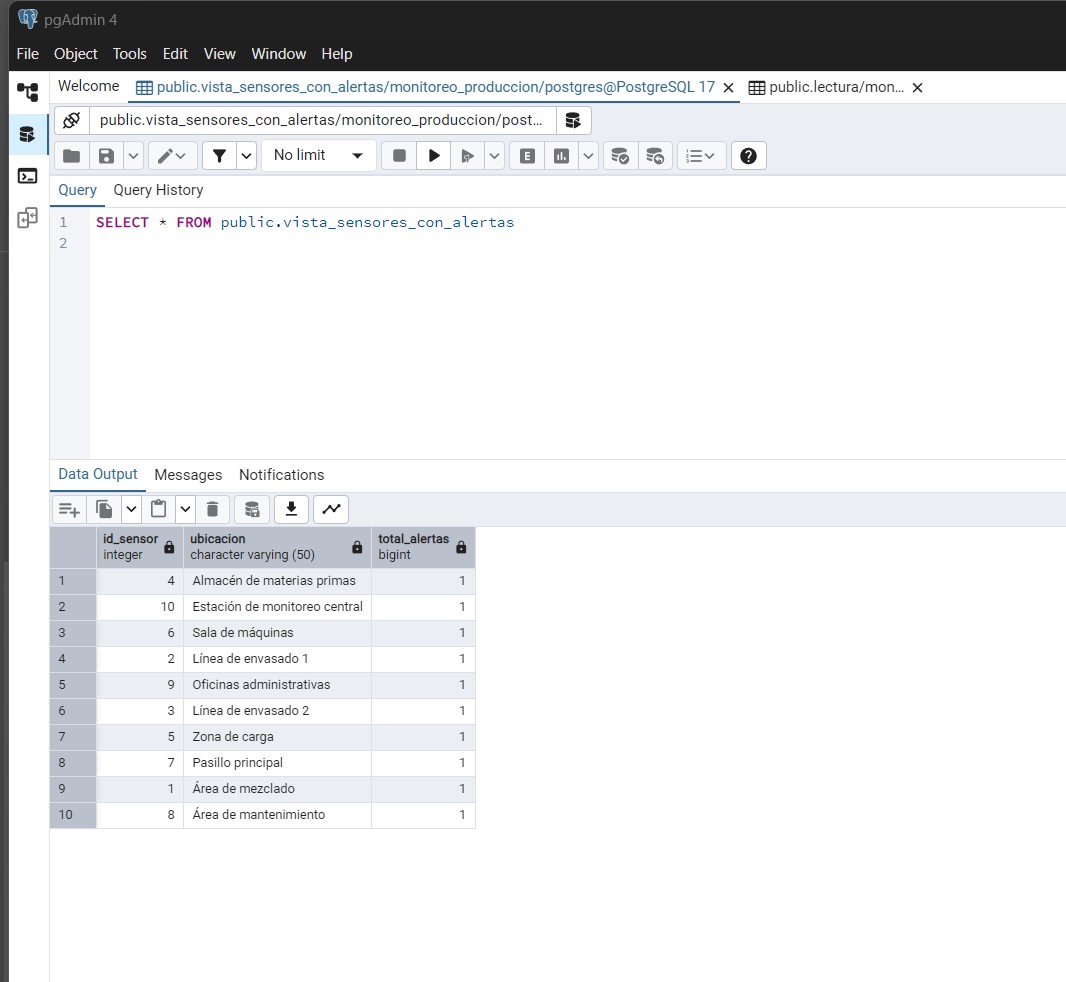
****

**Texto

El contenido generado por IA puede ser incorrecto.**

**PANTALLAZOS DE LA BASE DE DATOS**

**Tabla

El contenido generado por IA puede ser incorrecto.**

**7.- Implementación de un tablero de control y monitoreo con la herramienta “Metabase”**

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

**8.- Conclusiones.**

**SOFÍA RAMIREZ:**

Este proyecto me permitió entender con más claridad cómo se gestionan los costos de almacenamiento en la nube y lo importante que es elegir el servicio adecuado según las necesidades del sistema. Comparar servicios como Amazon EBS, S3 y Aurora me ayudó a visualizar mejor el impacto económico y técnico que tiene cada decisión en un entorno de Big Data. Además, me gustó mucho ver cómo todo el proceso ETL se integra con la visualización y el análisis de datos en tiempo real, algo que antes me parecía muy complejo.

Sinceramente, no pensé que esta materia me fuera a gustar tanto. Al principio la veía como algo muy técnico y lejano, pero con el tiempo le fui cogiendo cariño.

**JUAN FELIPE BARRERA:**

Los casos de estudio desarrollados nos permitieron aplicar los conceptos clave de Big Data en escenarios prácticos. En el primer caso, diseñamos un gobierno de datos y estructuramos un proyecto Big Data completo para el monitoreo de gases tóxicos en fábricas, integrando sensores, servicios en la nube y procesamiento en tiempo real.

En el segundo caso, construimos una arquitectura avanzada con tecnologías como Apache Kafka, Spark y Hadoop. Aunque tuve inconvenientes con el algoritmo de poblamiento en Visual Studio Code por un error de codificación UTF-8, logre poblar las tablas manualmente con sentencias INSERT. Además, trabaje con indicadores CMI y KPI en Metabase para diseñar un dashboard funcional que permite la visualización y análisis de los datos recolectados. No pude presentar mejor la información que se deseaba, he aplicado lo aprendido y aunque no pude con el error del algoritmo, quedo contento.

Estos trabajos reforzaron nuestras habilidades técnicas y nos dieron una visión clara del flujo completo de datos, desde su captura hasta su análisis, fortaleciendo nuestra preparación para retos profesionales en el campo de Big Data.

**JUAN DIEGO GALLARDO VILORIA:**

Este proyecto me permitió comprender cómo se diseña una arquitectura Big Data eficiente, considerando tanto el flujo de información como la integración de tecnologías para el procesamiento masivo de datos. Aprendí a calcular el volumen de datos que se generan en tiempo real desde sensores, lo que me dio claridad sobre la necesidad de estructuras escalables como Hadoop.

Además, al realizar los cálculos de almacenamiento y comparar costos en AWS (EBS, S3 y Aurora), entendí la importancia de seleccionar la tecnología adecuada según el tipo de carga de trabajo y presupuesto disponible. Este análisis me ayudó a valorar la planificación como parte fundamental de una solución Big Data bien implementada.

En general, estos ejercicios me dieron una visión práctica sobre cómo abordar proyectos de gran volumen de datos desde cero y cómo tomar decisiones técnicas bien fundamentadas para lograr soluciones eficientes y sostenibles.

**9.- Video de sustentación:**

*Elabore un video de sustentación con la participación de todos los integrantes (si es en equipo). Este vídeo debe informar sobre las actividades realizadas en general pero PRINCIPALMENTE debe mostrar el diseño y la explicación detallada de los componentes de las dos arquitecturas solicitadas; así como el flujo de la información desde la captura hasta el despliegue en un Tablero de Control. Debe tener buena calidad de sonido.*