**INFORME DE RESULTADOS**

**Unidad 3 - Tarea 7**

**Peso: 20%**

**Equipo/Grupo : 3**

**Estudiantes :**

**- *Luis Felipe Gil Gómez***

* ***Gerson Gustavo Fernández Badillo***
* ***Davidson Harley Rave Buitrango***
* ***David Pabón García***

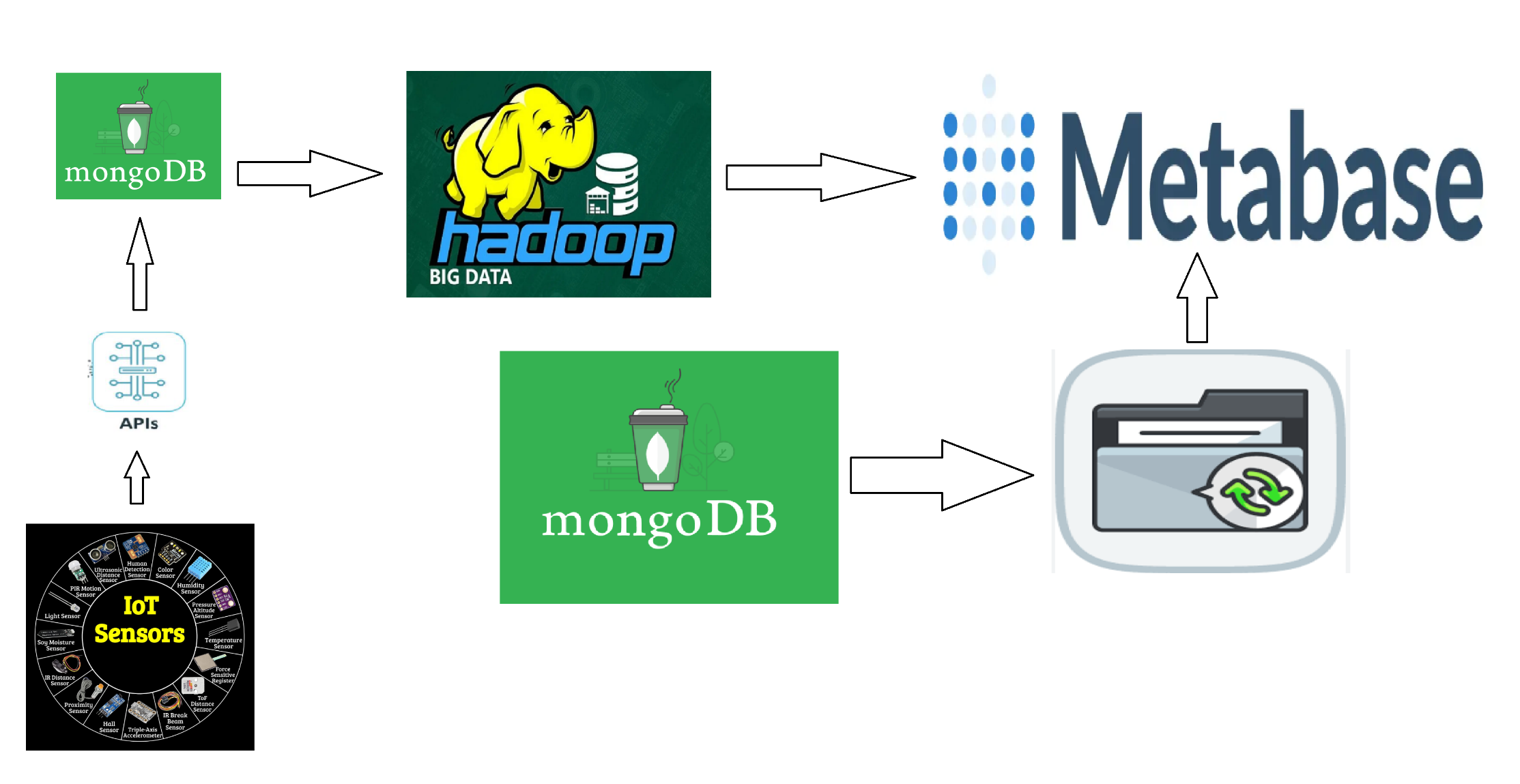
**Objetivo General**

**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 | 6 lecturas por minuto x 60 min x 24H =8.640 lecturas/dia | 8.640 |
| Lecturas de un mes de una línea de producción. Nota: un mes = 30 días. | 8640 lecturas/día x 30 dias x 20 sensores = 5.184.000 | 5.184.000 |
| Lecturas de un año de todas las líneas de producción de todas las fábricas. Nota: un año = 365 días. | 8.640 lecturas/día x 365 dias x 3 fabricas x 4 lineas x 20 sensores = 756.864.000 | 755.136.000 |

***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 | **6 lecturas** | 65 bytes | 390 |
| Un (1) hora | **6 x 60 = 360** | 65 bytes | 23.400 |
| Un (1) día | **8.640** | 65 bytes | 561.600 |
| Un (1) mes | **259.200 (8.640 x 30)** | 65 bytes | 16.848.000 |
| Un (1) año | **3.153.600 (8.640 x 365)** | 65 bytes | 204.984.000 |

***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** | 65 | 195.000.000 |
| Lote 2 | **20.000.000** | 65 | 1.300.000.000 |

***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.***

***1-***Una hoja de calculo XLSX tiene un limite de 1.048.576 filas

***2-***Cada sensor genera 8.640 lecturas por día.

***3-***Tiempo hasta alcanzar el limite

= 120.3 dias → Limpiar cada 120 dias (4 meses) para prevenir sobrecarga.

Para la hoja “lecturas-fabrica” que centraliza datos de 80 sensores:

***1-***80 x 8.640 = 691.200 lecturas/día

= 1.5 dias → Limpiar diariamente.

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

**3.1.- Cálculo de almacenamiento**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo Almacenamiento** | **Total Bytes** | **Costo x byte** | **Costo Total** |
| Almacenamiento en Bloque Elástico (EBS) | 1.21 GB | $0.00000000492 | $6.39 |
| Amazon Simple Storage Service (Amazon S3) | 1.21 GB | $0.0000000000385 | $0.05 |
| Amazon Aurora | 1.21 GB | $0.0000003503 | $455.05 |

**Nota:** Se considera que 20.000.000 registros x 65 bytes = 1.300.000.000 bytes ≈ 1.21 GB

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

En EBS esta diseñado para almacenamiento de bloques es más rápido y escalable, donde los datos son muy variables y se necesita de una respuesta más rápida a la hora de intervenir, en S3 es un almacenamiento barato para datos estáticos o archivados, estos datos no tendrán modificaciones, sino que guardaran información que requiera solo consulta, El Aurora es un servicio de gestión de base de datos con redundancia y rendimiento optimizado

La diferencia de costos entre los servicios de almacenamiento de AWS se debe principalmente al tipo de servicio, nivel de rendimiento, durabilidad, disponibilidad y características especializadas que ofrece cada uno:

**Amazon EBS (Elastic Block Store):**

Este servicio de almacenamiento es usado como discos duros virtuales. Es rápido, de baja latencia y adecuado para sistemas operativos, bases de datos y aplicaciones que requieren acceso frecuente y de alta velocidad a los datos. Su costo es favorable, mostrando su rendimiento y disponibilidad.

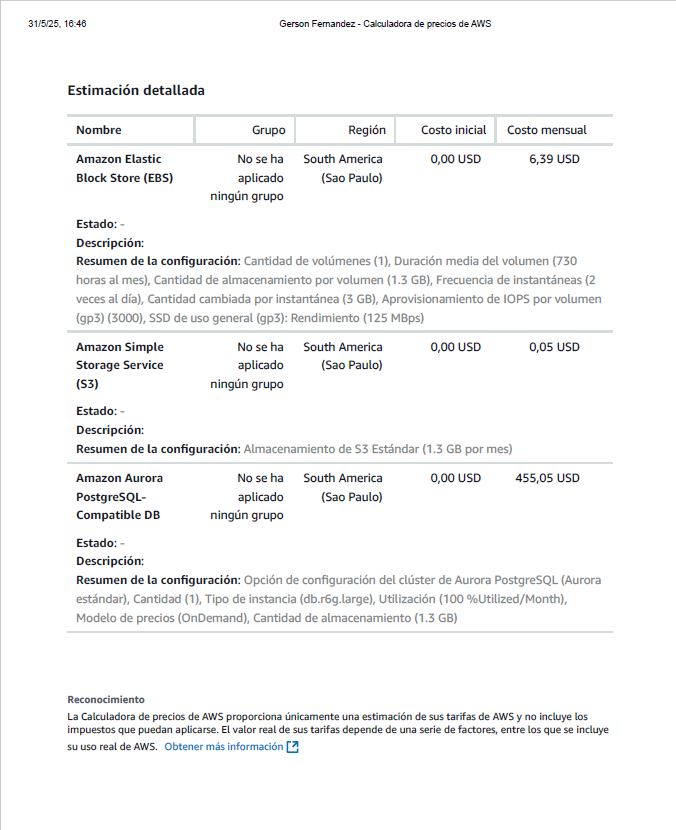
**Amazon S3 (Simple Storage Service):**

Es un servicio de almacenamiento orientada a objetos escalable y económico, ideal para almacenar grandes volúmenes de datos que no necesitan modificaciones frecuentes. Es el mas economico porque está optimizado para almacenamiento masivo con acceso menos frecuente.

**Amazon Aurora:**

Aurora es un sistema de gestión de bases de datos relacional administrado, que combina la velocidad y disponibilidad de bases de datos comerciales con la simplicidad y economía de código abierto. Su costo es considerablemente más alto porque incluye no solo almacenamiento, sino procesamiento, replicación automática, respaldos continuos, y tolerancia a fallos. Es ideal para aplicaciones críticas que requieren alta disponibilidad y rendimiento.

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



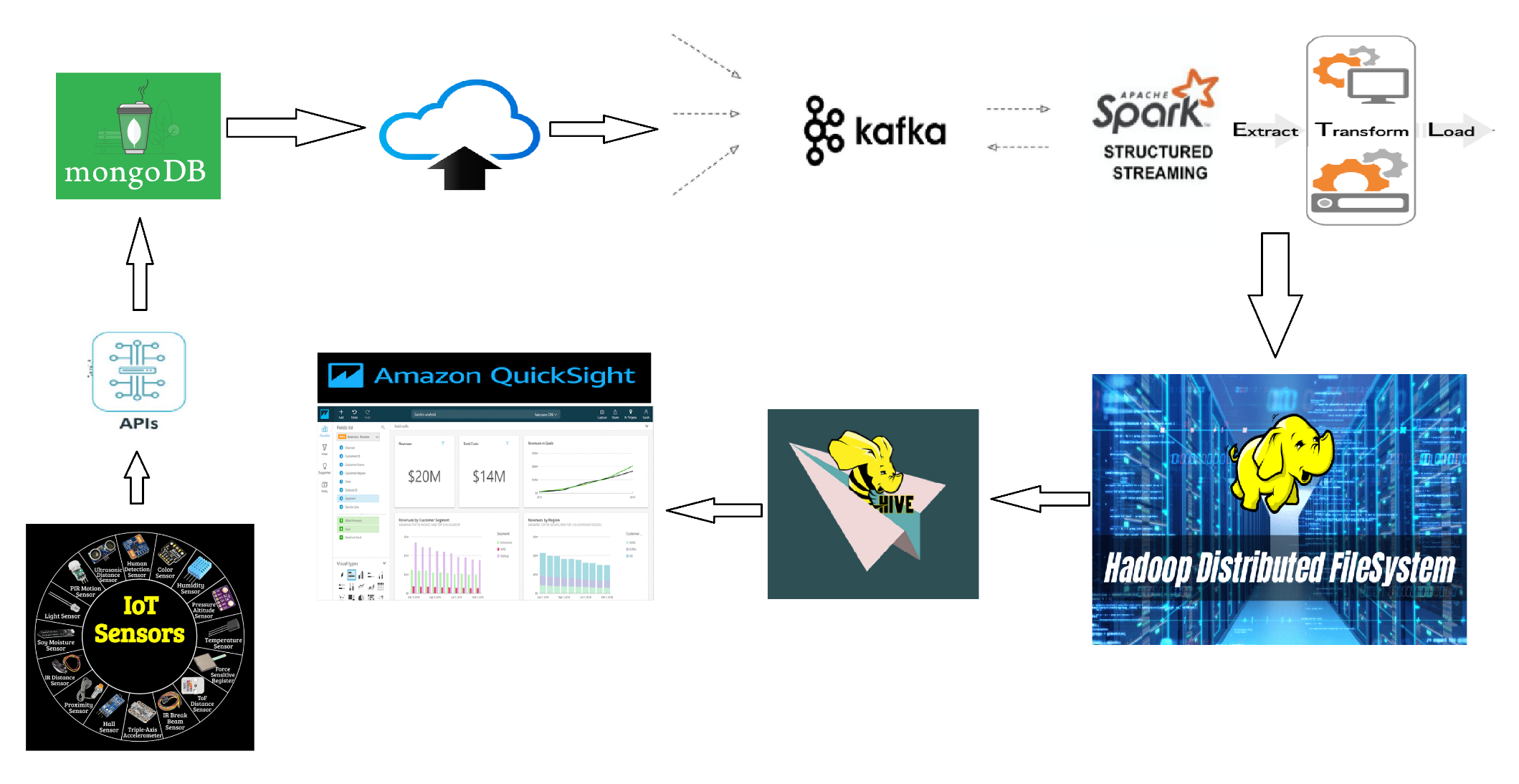
**4.- Cálculo del costo por 1 año de uso de los servicios AWS**

*Colocar en esta sección los cálculos del presupuesto (cotización) obtenida con la calculadora de AWS*

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

|  |  |  |
| --- | --- | --- |
| **Nombre Servicio** | **Costo individual** | **Costo anual** |
| Amazon EC2 | $29.44 | $353.28 |
| Amazon S3 | $0.05 | $0.60 |
| Amazon EBS | $6.39 | $76.68 |
| Amazon Aurora | $455.05 | $5,460.60 |
| AWS Lambda (1 millón de invocaciones/mes) | $0.20 | $2.40 |
| Amazon CloudWatch (monitoreo básico) | $3.00 | $36.00 |
| Amazon VPC (tráfico y NAT Gateway) | $10.00 | $120.00 |
| AWS Glue (procesamiento ETL esporádico) | $5.00 | $60.00 |
| Amazon QuickSight (1 usuario estándar) | $9.00 | $108.00 |
| Amazon EC2 (Instancia t3.medium, 24/7) | $29.44 | $353.28 |

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

**** **Apache Kafka:** Recibe los datos desde MongoDB local en tiempo real desde las 11 fábricas.

**Apache Spark:** Procesa los datos en tiempo real para realizar el ETL (limpieza, validación, transformación y agregación).

**Hadoop HDFS:** Almacén escalable de datos y distribución de los datos procesados por Spark.

**Apache Hive:** Capa de consulta que permite usar SQL sobre los datos en HDFS. Facilitando la exploración con herramientas externas.

**Amazon QuickSight:** Plataforma de visualización que se conecta a Hive (vía AWS) para dashboards.

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

*Elabore un algoritmo que genere de manera aleatorio los datos necesarios para poblar la tabla “lecturas” de la base de datos y la hoja del cálculo del sensor “lecturas-sensor”. Se sugiere tomar como punto de partida el algoritmo ETL y el algoritmo de cálculo de tiempo y almacenamiento de la primera Tarea; y adaptar el programa Python al nuevo requerimiento.*

*Coloque un pantallazo en esta sección y entregue el código fuente con los otros productos solicitados*

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

*Elabore un tablero de control y monitoreo que contenga los gráficos que se encuentran en los requerimientos del documento de instrucciones. Coloque un pantallazo del tablero en esta sección. NOTA: recuerde que durante la grabación del video de sustentación, debe mostrar el tablero (Dashboard) en funcionamiento.*

**8.- Conclusiones.**

*Elabore las conclusiones planteando la importancia y utilidad de esta tarea y su relación con el contenido de la asignatura. ¿Cuáles eran sus expectativas al inicio de la asignatura y en este momento final? ¿Cómo cree Ud. que el conocimiento aprendido en la asignatura afectará sus oportunidades laborales y desempeño profesional una vez obtenga su título en el Pascual Bravo?*

*Adicionalmente a estas conclusiones generales del grupo de trabajo, cada estudiante debe expresar sus propias conclusiones.*

**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.*