**Informe de Arquitectura Big Data**

**1. Adiciones y Mejoras Sugeridas**

**Orquestación de Flujo de Datos**

* **Apache NiFi**: Ingesta de datos en tiempo real y enrutamiento de datos.
* **Apache Airflow**: Orquestación de tareas y flujos de trabajo complejos.

**Procesamiento en Tiempo Real y por Lotes**

* **Apache Flink**: Procesamiento en tiempo real, complementa a Kafka y proporciona baja latencia.
* **Apache Beam**: Proporciona un modelo unificado para el procesamiento tanto en lote como en streaming.

**Almacenamiento Distribuido y Base de Datos NoSQL**

* **Apache Cassandra**: Almacenamiento distribuido, escalabilidad horizontal y alta disponibilidad.
* **Amazon S3**: Almacenamiento en la nube, fácil de integrar y escalable.
* **Apache HBase**: Consultas rápidas y almacenamiento de datos no estructurados.

**Gestión y Gobernanza de Datos**

* **Apache Atlas**: Gestión de metadatos y linaje de datos.
* **Apache Ranger**: Gestión de la seguridad y políticas de acceso.

**Machine Learning y Análisis**

* **Apache Mahout**: Algoritmos de aprendizaje automático escalables.
* **TensorFlow**: Modelos avanzados de aprendizaje profundo.
* **MLlib**: Librería de aprendizaje automático de Spark.

**Visualización y BI**

* **Grafana**: Monitorización y visualización en tiempo real.
* **Metabase**: BI y creación de dashboards interactivos.

**Infraestructura y Orquestación**

* **Kubernetes**: Orquestación de contenedores, escalabilidad y gestión de servicios.

**Monitoreo y Logging**

* **ELK Stack**: Ingesta, búsqueda y visualización de logs.
* **Prometheus**: Monitorización y alertas.

**2. Mejoras Sugeridas en la Arquitectura**

**Optimización del Flujo de Datos**

* **Data Lineage**: Implementar un sistema como Apache Atlas para rastrear el linaje de datos.
* **Data Quality Checks**: Integrar validaciones de calidad de datos en diferentes etapas.

**Seguridad y Gobernanza**

* **Apache Ranger**: Asegurar que los datos estén protegidos con políticas de acceso estrictas.
* **Data Encryption**: Implementar cifrado de datos en reposo y en tránsito.

**Escalabilidad y Alta Disponibilidad**

* **Multi-Region Clusters**: Configurar clústeres distribuidos en múltiples regiones.
* **Auto-Scaling**: Utilizar Kubernetes para auto-escalar servicios.

**Automatización y DevOps**

* **CI/CD Pipelines**: Configurar pipelines de CI/CD para despliegue y actualización de servicios.
* **Infrastructure as Code (IaC)**: Utilizar Terraform o Ansible para gestionar y desplegar la infraestructura.

**3. Resumen de Herramientas Sugeridas**

**Orquestación de Flujo**

* Apache Airflow
* Apache NiFi

**Procesamiento**

* Apache Flink
* Apache Beam
* PySpark

**Almacenamiento**

* Apache HDFS
* Apache Cassandra
* Amazon S3

**Gobernanza**

* Apache Atlas
* Apache Ranger

**Machine Learning**

* Apache Mahout
* TensorFlow
* MLlib

**Visualización**

* Grafana
* Metabase
* Apache Superset

**Orquestación de Contenedores**

* Kubernetes

**Monitoreo y Logging**

* ELK Stack
* Prometheus