Actividad práctica — Ecosistema de Big Data

1. Arquitectura Hadoop y componentes principales

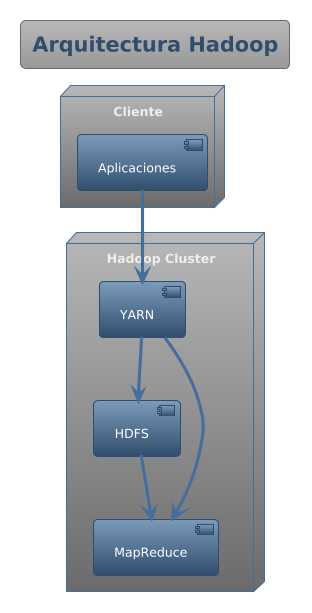
Componentes:

HDFS (Hadoop Distributed File System): Sistema de archivos distribuido que divide y replica los datos en múltiples nodos para garantizar disponibilidad y tolerancia a fallos.

MapReduce: Motor de procesamiento que divide las tareas en pequeños bloques para ejecutarlas en paralelo en diferentes nodos.

YARN (Yet Another Resource Negotiator): Gestor de recursos que asigna y controla el uso de CPU, memoria y tareas dentro del clúster Hadoop.

Diagrama:

@startuml

!theme spacelab

title Arquitectura Hadoop

node "Cliente" {

[Aplicaciones]

}

node "Hadoop Cluster" {

[HDFS] --> [MapReduce]

[YARN] --> [MapReduce]

[YARN] --> [HDFS]

}

Aplicaciones --> [YARN]

@enduml

2. Apache Spark: procesamiento en memoria

Explicación:  
Apache Spark es un motor de procesamiento distribuido en memoria que permite analizar grandes volúmenes de datos de forma mucho más rápida que Hadoop MapReduce. Su ventaja principal es que mantiene los datos en RAM, evitando lecturas/escrituras constantes en disco, lo que mejora el rendimiento en análisis en tiempo real, machine learning y streaming.

Esquema de flujo en memoria (texto):

Ingesta de datos →

Carga en memoria (RDD/DataFrames) →

Procesamiento paralelo →

Resultados inmediatos en dashboards, ML o bases de datos.

3. Bases de datos NoSQL (MongoDB, Cassandra, Redis)

| Base de datos | Modelo de datos | Casos de uso | Ventajas | Limitaciones | Ejemplo práctico |
| --- | --- | --- | --- | --- | --- |
| MongoDB | Documentos (JSON) | Aplicaciones web dinámicas | Flexible, escalable, fácil integración | Menor rendimiento en transacciones complejas | Gestión de catálogos de productos en e-commerce |
| Cassandra | Columnares distribuidas | Big Data en tiempo real, IoT | Alta disponibilidad, escalabilidad horizontal | Configuración compleja | Registro de datos de sensores en tiempo real |
| Redis | Clave-valor en memoria | Caching, sesiones, ranking | Velocidad extrema, soporta estructuras de datos | No apto para grandes volúmenes persistentes | Manejo de sesiones de usuario en plataformas online |

4. Herramientas de ingesta de datos (Kafka, Flume)

Definición:  
La ingesta de datos es el proceso de recopilar, transportar y preparar datos desde múltiples fuentes hacia un sistema de almacenamiento o análisis en Big Data.

Funciones:

Apache Kafka: Plataforma de mensajería distribuida para manejar flujos de datos en tiempo real con alta confiabilidad.

Apache Flume: Herramienta especializada en la captura, agregación y transporte de grandes volúmenes de datos de logs hacia HDFS.

Diagrama de flujo de ingesta (texto):  
Sensores IoT → Kafka/Flume → Hadoop HDFS o Spark → Análisis y dashboards.

5. Almacenamiento en la nube (AWS, Azure, GCP)

Servicios:

AWS S3 (Simple Storage Service): Servicio de almacenamiento de objetos escalable y económico, base de la mayoría de arquitecturas Big Data en Amazon.

Azure Data Lake Storage: Diseñado para grandes volúmenes de datos analíticos, integra seguridad empresarial y escalabilidad.

Google Cloud Storage (GCS): Almacenamiento unificado de objetos, altamente disponible y con integración nativa a BigQuery para análisis.

Mapa comparativo:

AWS S3: Líder en ecosistema, gran comunidad, costo competitivo.

Azure Data Lake: Integración nativa con herramientas Microsoft (Power BI, Azure ML).

GCP Storage: Optimizado para análisis avanzado e IA, integración fluida con BigQuery y TensorFlow.