

Middleware en Computación de Alto Rendimiento

Computación de alto rendimiento

Índice de Contenidos

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

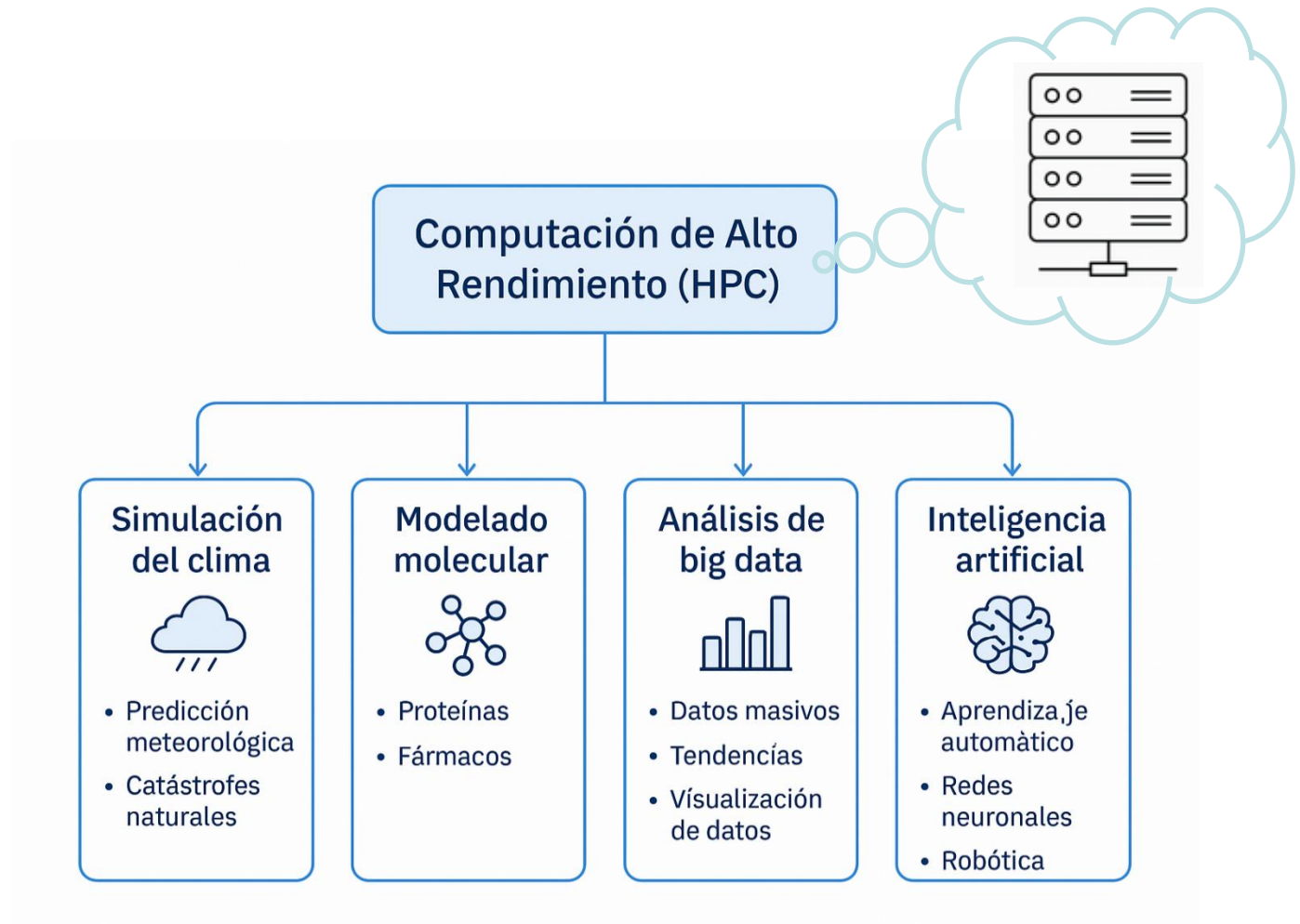
Presentación de
prácticas

1. Introducción Conceptual
2. Arquitectura y funciones del Middleware
3. Tipos de Middleware (Clasificación funcional)
4. Ventajas e inconvenientes por tipo
5. Hadoop como middleware
6. Apache Spark como evolución de Hadoop
7. Bibliografía recomendada

1. Introducción Conceptual

1. Introducción Conceptual

1.1 Qué es CAR y por qué es necesaria



Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

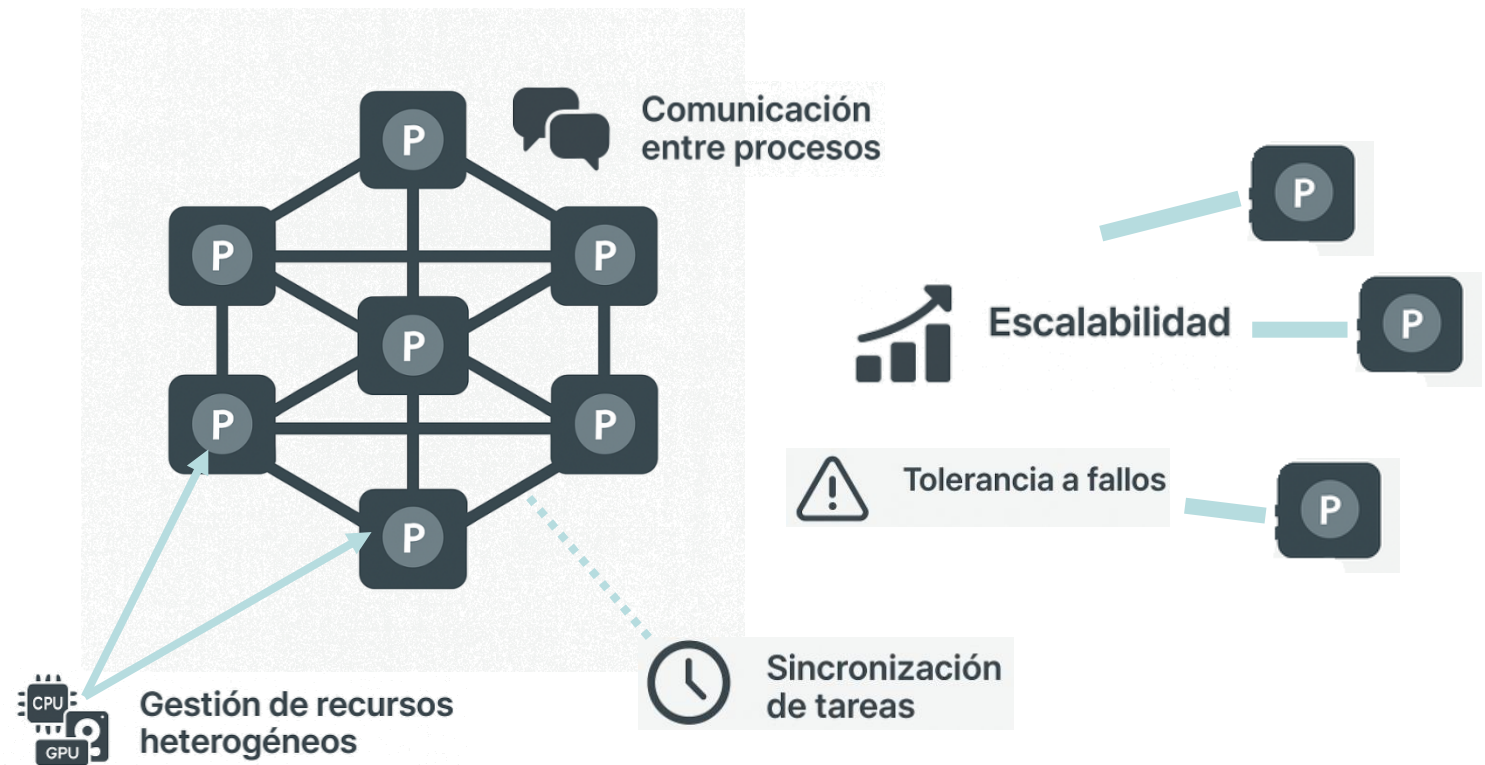
Spark

Bibliografía

Presentación de prácticas

1. Introducción Conceptual

1.2 Dificultades en la gestión de sistemas distribuidos



Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

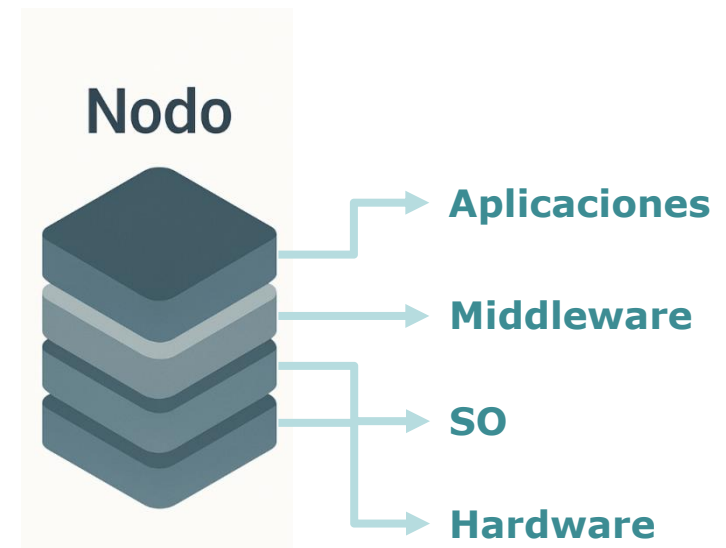
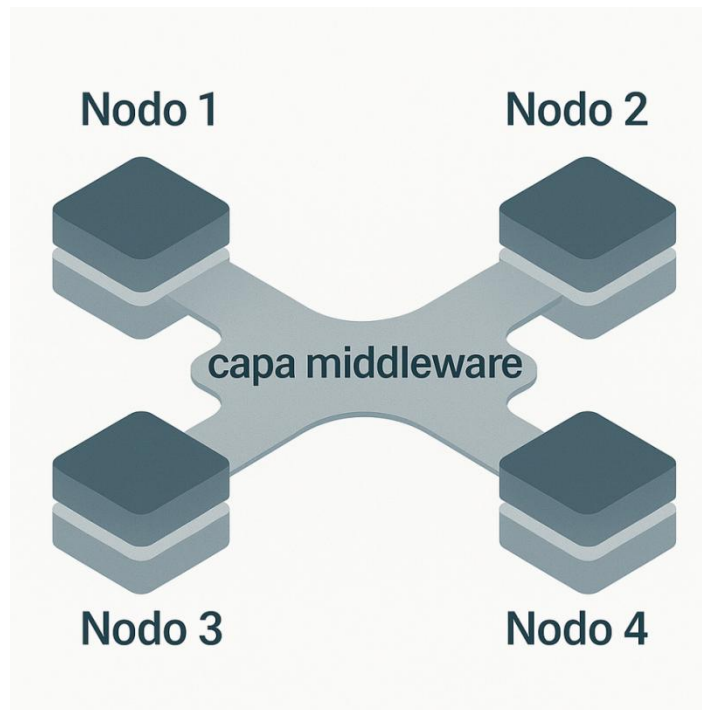
Spark

Bibliografía

Presentación de
prácticas

1. Introducción Conceptual

1.3 ¿Qué papel juega el middleware en este contexto?



Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

1. Introducción Conceptual

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

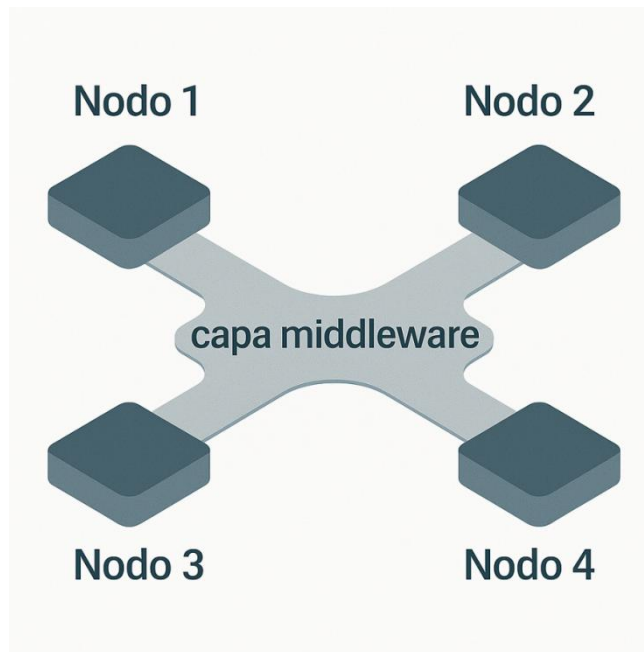
Hadoop

Spark

Bibliografía

Presentación de
prácticas

1.4 Servicios que ofrece el middleware



- Abstracción de **red** y **ubicación**.
- **Comunicación** entre procesos y paso de mensajes.
- Control de **concurrency** y **sincronización**.
- **Distribución** de tareas y **balanceo** de carga.
- **Gestión** de **errores** y **recuperación** ante fallos.
- Interoperabilidad entre plataformas y lenguajes.

1. Introducción Conceptual

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

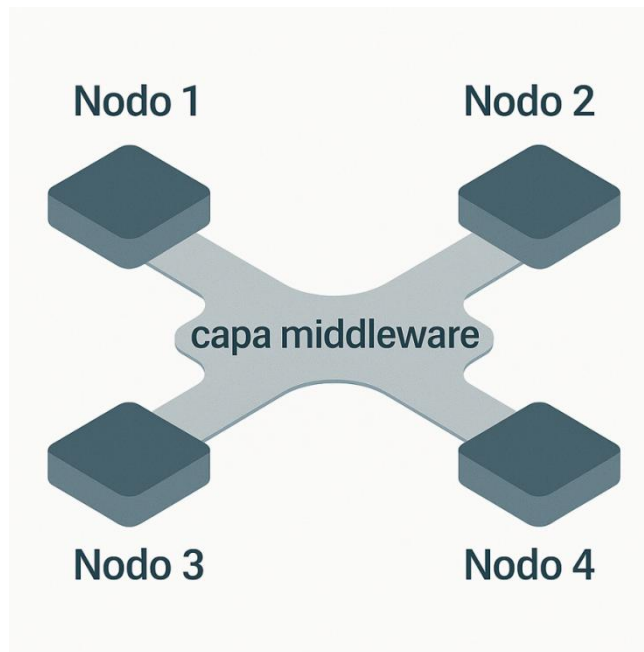
Hadoop

Spark

Bibliografía

Presentación de
prácticas

1.4 Servicios que ofrece el middleware



- Abstracción de **red** y **ubicación**.
- **Comunicación** entre procesos y paso de mensajes.
- Control de **concurrency** y **sincronización**.
- **Distribución** de tareas y **balanceo** de carga.
- **Gestión** de **errores** y **recuperación** ante fallos.
- Interoperabilidad entre plataformas y lenguajes.

2. Arquitectura y funciones del Middleware

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

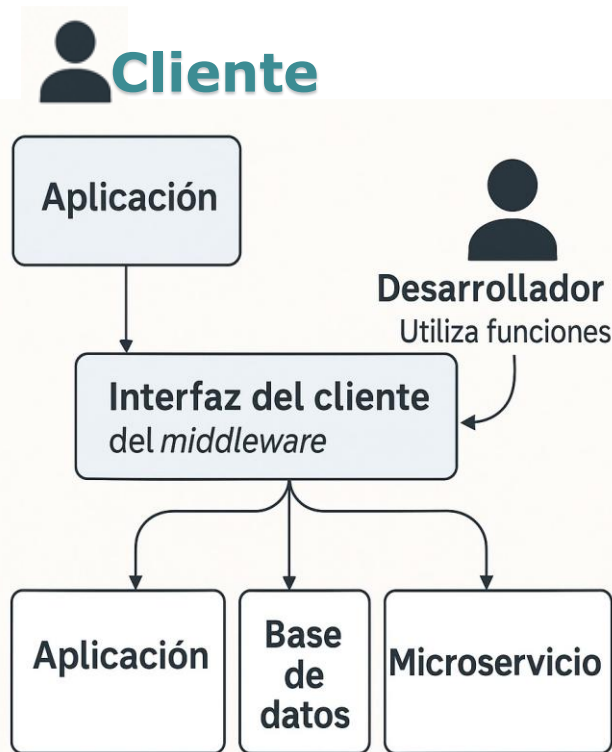
Spark

Bibliografía

Presentación de prácticas

2.1 Arquitectura básica cliente-middleware-servidor

Modelo en tres capas típico de sistemas distribuidos y entornos CAR



Cliente:

- **Solicita servicios o datos.**
- No necesita conocer la ubicación del servidor

Middleware:

- **Intermedia:** enruta, gestiona, abstrae y protege la comunicación.
- Coordina ejecución distribuida.

Servidor:

- **Proporciona los servicios.**
- Puede ser un nodo único o un clúster cooperativo.

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

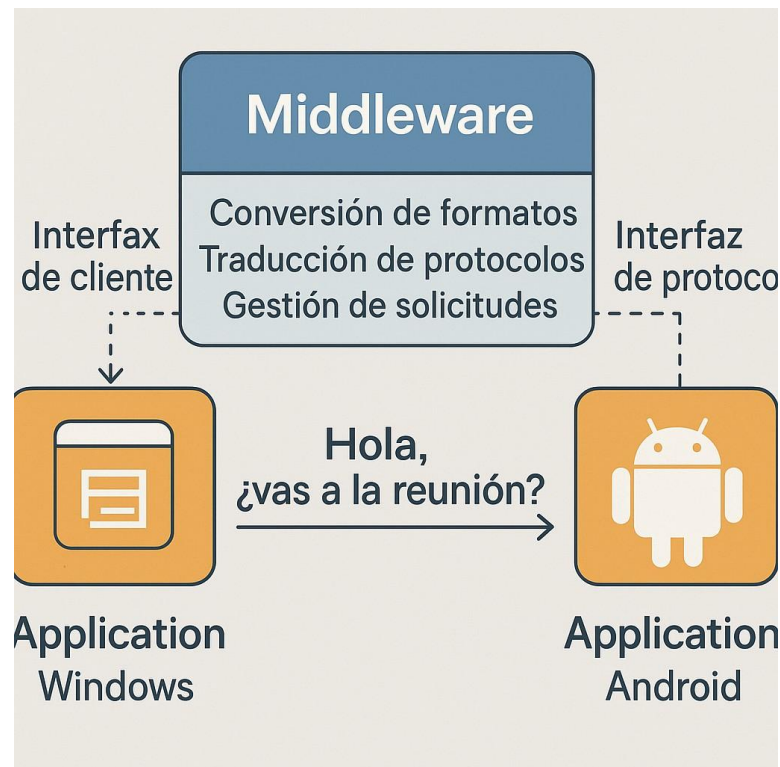
Hadoop

Spark

Bibliografía

Presentación de
prácticas

Ejemplo 1:



•**Cliente: Solicita servicios o datos.** No necesita conocer la ubicación del servidor

•**Meddleware: Intermedia:** enruta, gestiona, abstrae y protege la comunicación. Coordina ejecución distribuida.

•**Servidor: Proporciona los servicios.** Puede ser un nodo único o un clúster cooperativo.

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

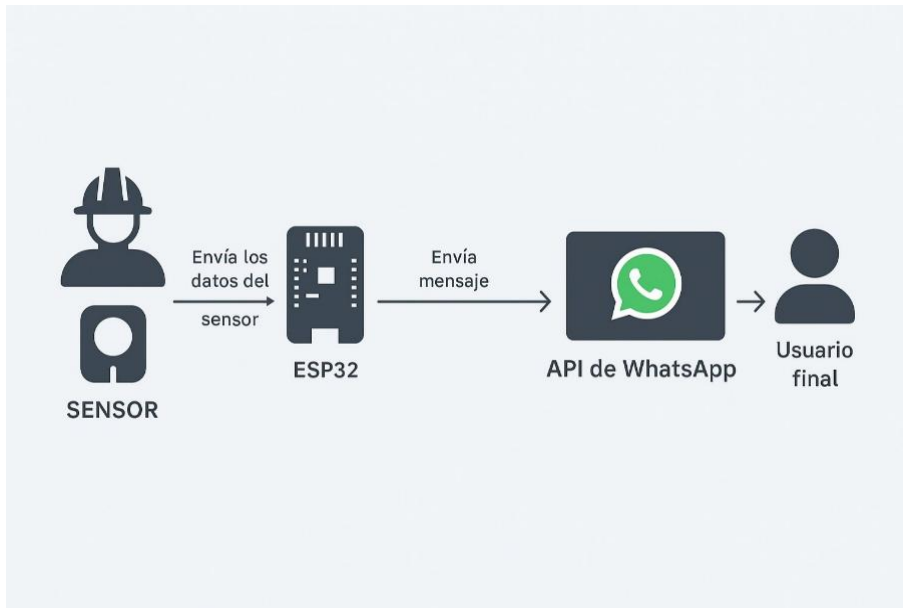
Hadoop

Spark

Bibliografía

Presentación de
prácticas

Ejemplo 2:



- Conecta sensor, dispositivo y usuario.
- Traduce y envía los datos automáticamente.
- Permite que sistemas distintos se comuniquen.
- Abstrae la complejidad

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

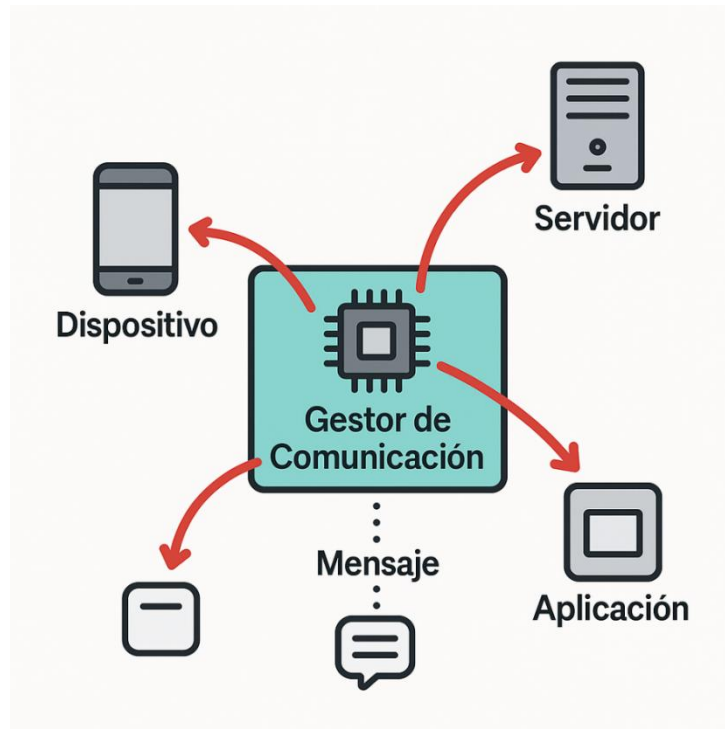
Spark

Bibliografía

Presentación de
prácticas

2.2 Componentes internos del middleware

- Módulo Gestor de comunicación:
 - Crea canales de comunicación entre Nodos



Diferentes canales

Internet (HTTP, TCP/IP...)
Redes privadas
Sistemas de colas de
mensajes...

Diferentes Protocolos

HTTPS
MQTT
FTP...

Adapta mensajes al canal

Fragmenta
Reenvia
Ordena
Prioriza

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

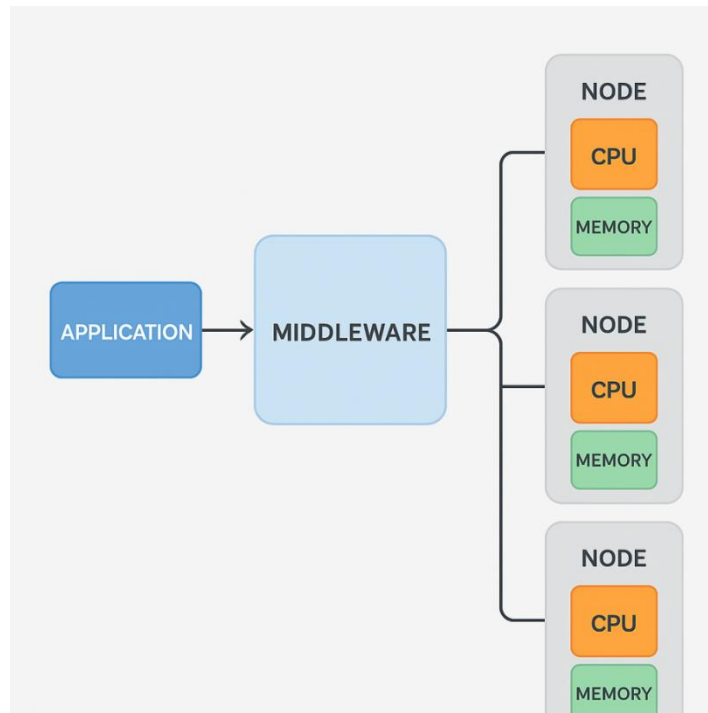
Spark

Bibliografía

Presentación de prácticas

2.2 Componentes internos del middleware

- Módulo Gestor de Recursos
 - Supervisa y asigna recursos computacionales según la carga de trabajo y las políticas del sistema.



Ejemplo de Políticas del sistema:

- Máximo 30% recursos **CPU/App**
- Si **memoria** > 90% liberar procesos en segundo plano
- Asignación de Apps críticas (prioridades)
- Apps de generación de video -> **GPU**

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

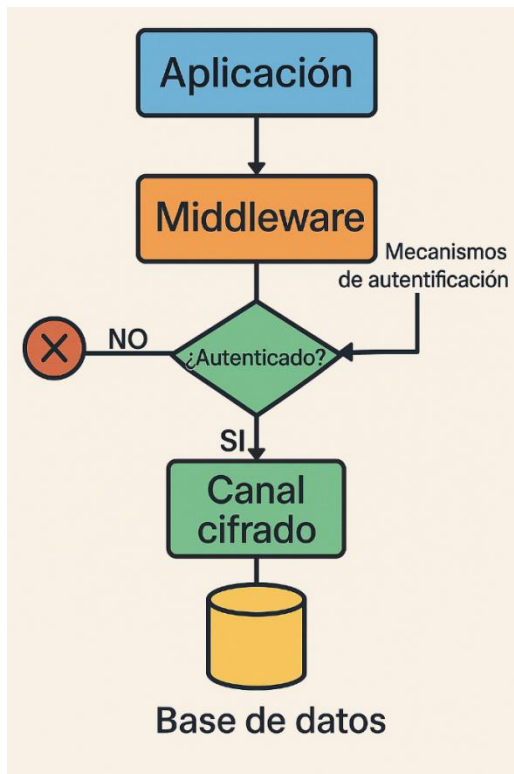
Spark

Bibliografía

Presentación de prácticas

2.2 Componentes internos del middleware

- **Módulo Gestor de Seguridad**
 - Implementa mecanismos de autenticación, autorización y cifrado para asegurar el acceso a los datos y prevenir usos no autorizados.



¿Qué previene el Gestor de Seguridad?

- **Accesos no autorizados:** (usuarios que no deberían ver cierta información).
- **Suplantación de identidad:** (que alguien se haga pasar por otro).
- **Intercepción de datos en tránsito:** (por ejemplo, con ataques tipo "man-in-the-middle").
- **Modificaciones no controladas de datos sensibles.**

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

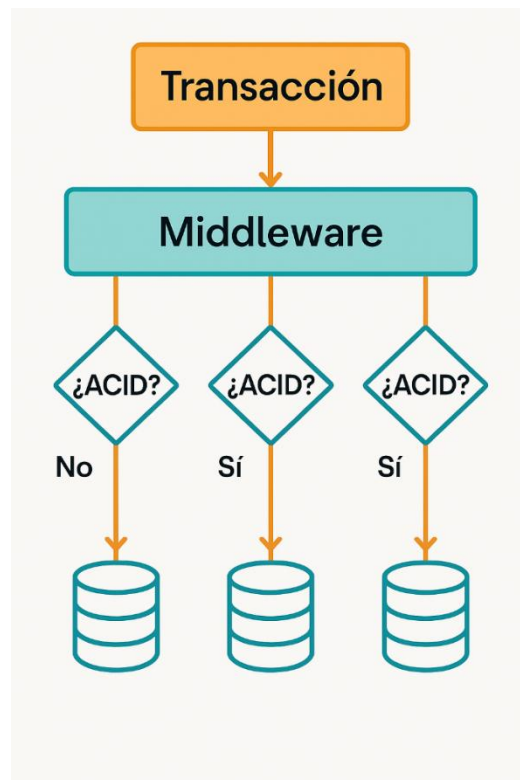
Spark

Bibliografía

Presentación de
prácticas

2.2 Componentes internos del middleware

- Módulo Gestor de Transacciones
 - Coordina operaciones distribuidas que deben cumplir propiedades **ACID** particularmente útil en sistemas de almacenamiento



- **(A)tomidad:** O se hace **todo o nada**. Si una parte falla, se revierte todo.
- **(C)onsistencia:** El sistema siempre pasa **de un estado válido a otro válido**. No se rompe ninguna regla.
- **A(i)slamiento:** Las transacciones no interfieren entre sí, aunque se ejecuten al mismo tiempo.
- **(D)urabilidad:** Una vez confirmada, la transacción **se guarda para siempre**, incluso si hay un apagón.

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

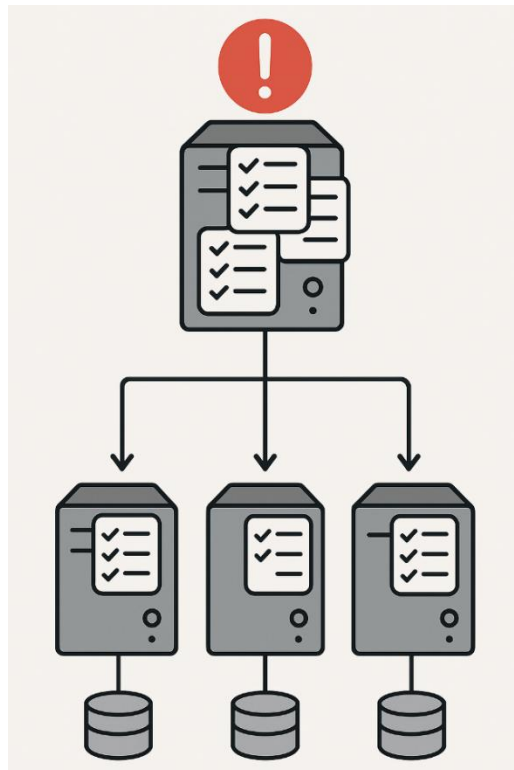
Spark

Bibliografía

Presentación de prácticas

2.2 Componentes internos del middleware

- Módulo Gestor de Fallos:
 - Detecta errores y pone en marcha estrategias de recuperación, como la replicación de procesos o la reubicación de tareas



- **Beneficios del Gestor de Fallos**
 - **Evita que el sistema se detenga** si ocurre un error.
 - Mejora la experiencia del usuario, ya que **no se notan los fallos** internos.
 - **Si un nodo o servidor falla**, las tareas que estaban en ese nodo se **mueven a otro nodo disponible**.

2. Arquitectura y funciones del Middleware distribuido

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

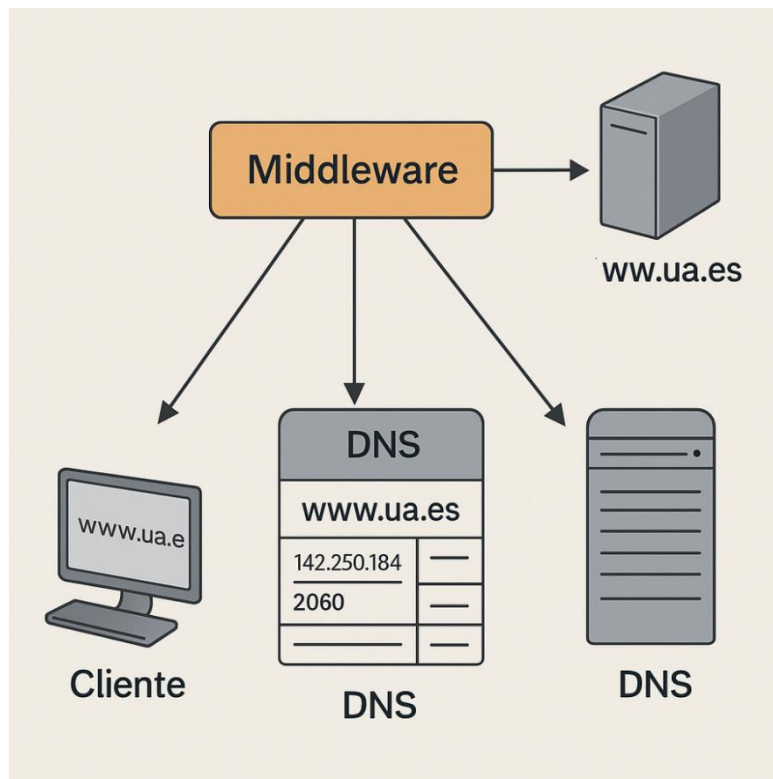
Spark

Bibliografía

Presentación de
prácticas

2.2 Componentes internos del middleware

- Módulo Gestor de Servicios de Nombres y Directorios :
 - Localiza y etiqueta componentes ocultando sus identificadores reales



• Ejemplos:

- Direcciones IP/puertos
- Credenciales de Bases de datos
- Nombres de Objetos (Corba)
- Colas de mensajes (Kafka...)
- Estructura de directorios
- ...

3. Tipos de Middleware (Clasificación funcional)

3. Tipos de Middleware (Clasificación funcional)

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

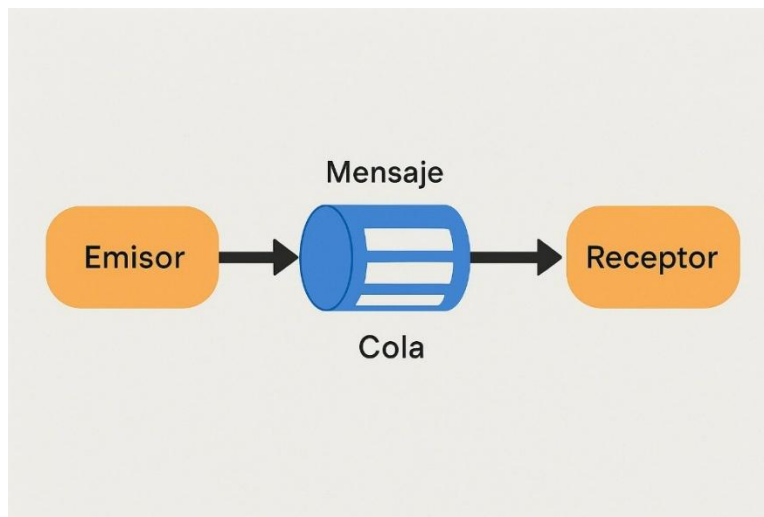
Spark

Bibliografía

Presentación de
prácticas

3.1 Middleware orientado a mensajes (MOM)

Localiza y etiqueta componentes ocultando sus identificadores reales

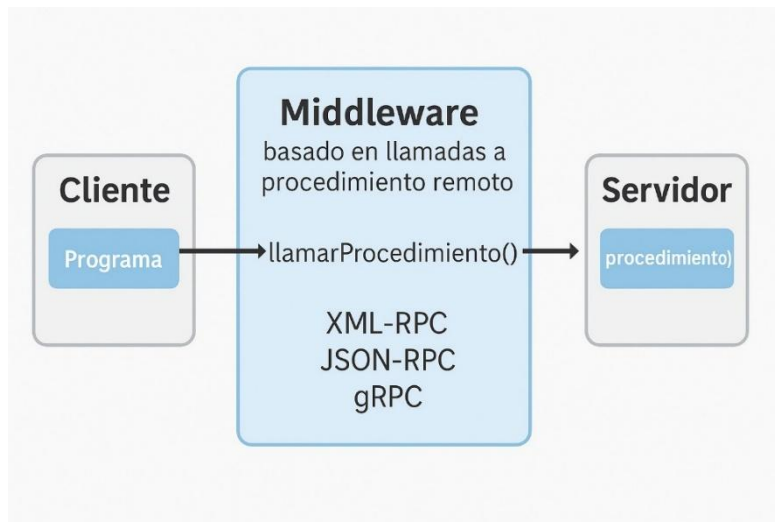


- **Comunicación asincrónica** (sin necesidad de que el receptor esté activo)
- **Uso de colas de mensajes:** se almacenan temporalmente en una cola
- **Desacoplamiento entre emisor y receptor:** El productor del mensaje y el consumidor no tienen que conocerse
- **Tolerancia a fallos:** los mensajes quedan en la cola
- **Escalabilidad**

3. Tipos de Middleware (Clasificación funcional)

3.2 Middleware basado en RPC

Basado en llamadas a procedimiento remoto



- **Llamadas remotas como si fueran locales:** Oculta que el sistema es remoto
- **Transparencia de red:** Oculta los detalles de la comunicación (IP, protocolos, etc.).
- **Síncrono por defecto:** El cliente espera la respuesta.
- **Ligero y eficiente:** Ideal para arquitecturas de microservicios

Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

Spark

Bibliografía

Presentación de prácticas

3. Tipos de Middleware (Clasificación funcional)

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

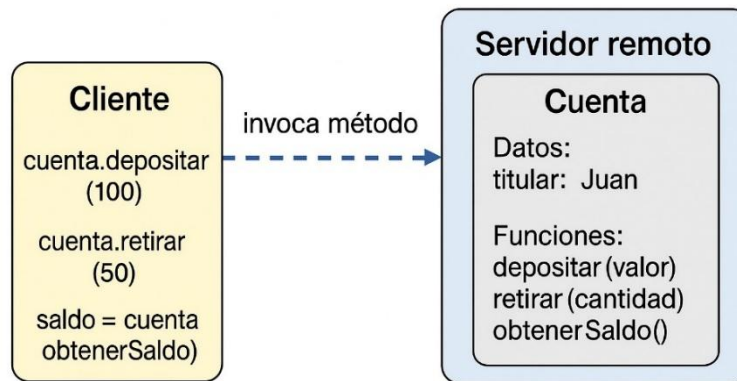
Bibliografía

Presentación de
prácticas

3.3 Middleware basado en objetos distribuidos

Invoca a métodos de Objetos distribuidos

Middleware basado en objetos distribuidos

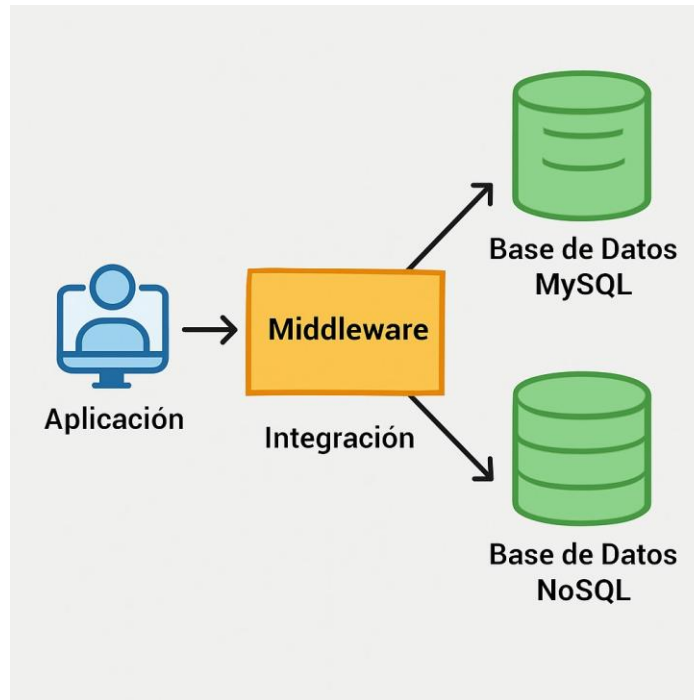


- **Transparencia de ubicación**
(como si fueran locales)
- **Orientación a objetos:**
(encapsulamiento, herencia, polimorfismo)
- **Modularidad:** Sistemas distribuidos complejos
- **Ejemplo:** Corba

3. Tipos de Middleware (Clasificación funcional)

3.4 Middleware orientado a datos

Gestiona grandes volúmenes de datos



- Optimizado para **análisis de datos masivos**
- **Procesamiento por lotes** o en tiempo real
- **Distribución y paralelismo:** Divide los datos y las tareas entre múltiples nodos
- **Ejemplos destacados:**
 - Apache Hadoop
 - Apache Spark

Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

Spark

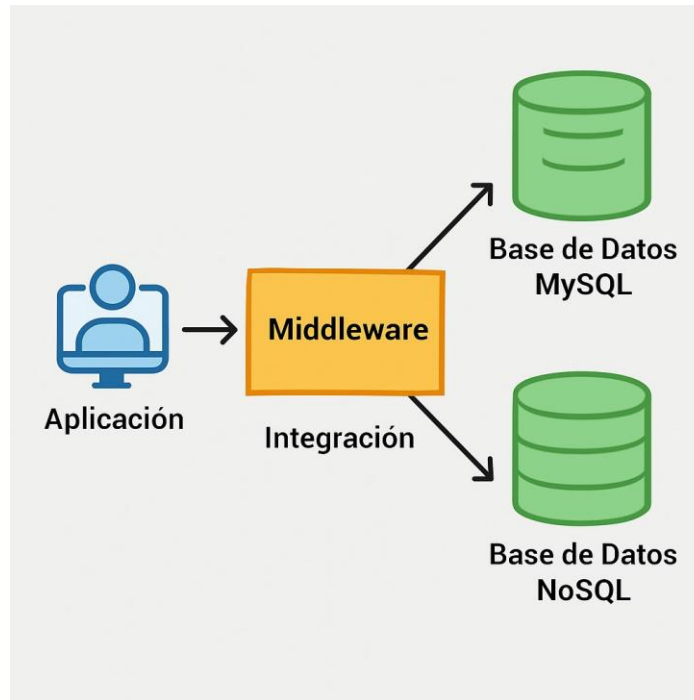
Bibliografía

Presentación de prácticas

3. Tipos de Middleware (Clasificación funcional)

3.4 Middleware orientado a datos

Gestiona grandes volúmenes de datos



- Optimizado para **análisis de datos masivos**
- **Procesamiento por lotes** o en tiempo real
- **Distribución y paralelismo:** Divide los datos y las tareas entre múltiples nodos
- **Ejemplos destacados:**
 - Apache Hadoop
 - Apache Spark

Índice

Introducción

Arquitectura

Tipos de middleware

Ventajas e inconvenientes

Hadoop

Spark

Bibliografía

Presentación de prácticas

4. Ventajas e inconvenientes por tipo

4. Ventajas e inconvenientes por tipo

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

4.2 Casos de uso recomendados según el tipo

Ejemplos de middleware representativos de cada tipo funcional:

- **Kafka:**
 - Sistemas de streaming de datos
 - Procesamiento en tiempo real (como logs de sensores en CAR).
- **gRPC (RPC moderno):**
 - Arquitecturas de microservicios que requieren comunicación estructurada y ligera.
- **CORBA (objetos distribuidos) :**
 - Sistemas científicos con fuerte encapsulamiento de procesos y datos.
- **Hadoop/Spark (orientado a datos):**
 - Grandes volúmenes de datos
 - Entrenamiento de modelos de IA
 - Búsquedas masivas

5. Hadoop como Middleware

5. Hadoop como Middleware

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

5.1 ¿Qué es Hadoop y por qué es middleware?

Hadoop



Escalabilidad horizontal
Añadir más nodos mejora
el rendimiento



Alta tolerancia a fallos
Mediante replicación



Modelo sencillo
Para procesar grandes
volúmenes de datos

- **Framework** de código abierto
- **Almacenamiento y procesamiento distribuido** de grandes volúmenes de **datos**.

5. Hadoop como Middleware

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

5.2 Arquitectura general (HDFS + MapReduce)

HDFS = el sistema de almacenamiento

Divide los datos en bloques

Distribuye los bloques en nodos

Replica los datos como respaldo

MapReduce = el sistema de procesamiento

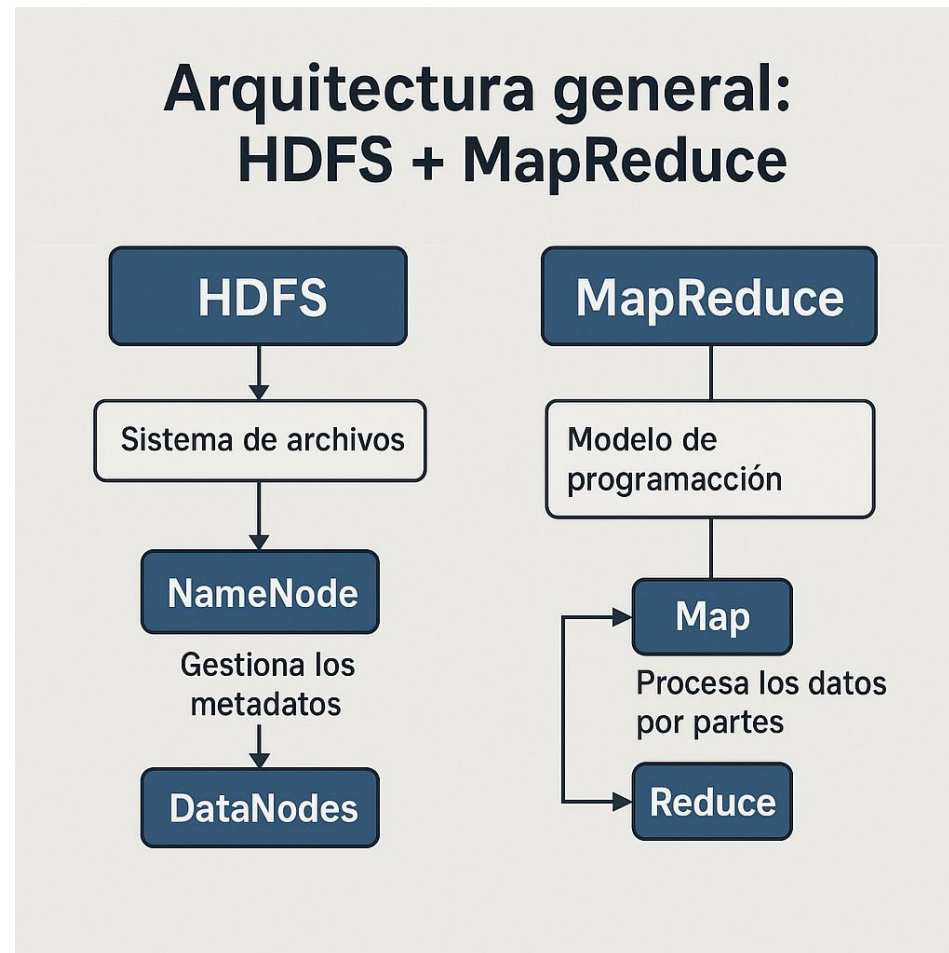
Lee los datos desde HDFS

Procesa datos en paralelo

Genera resultados

5. Hadoop como Middleware

5.2 Arquitectura general (HDFS + MapReduce)



Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

5. Hadoop como Middleware

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

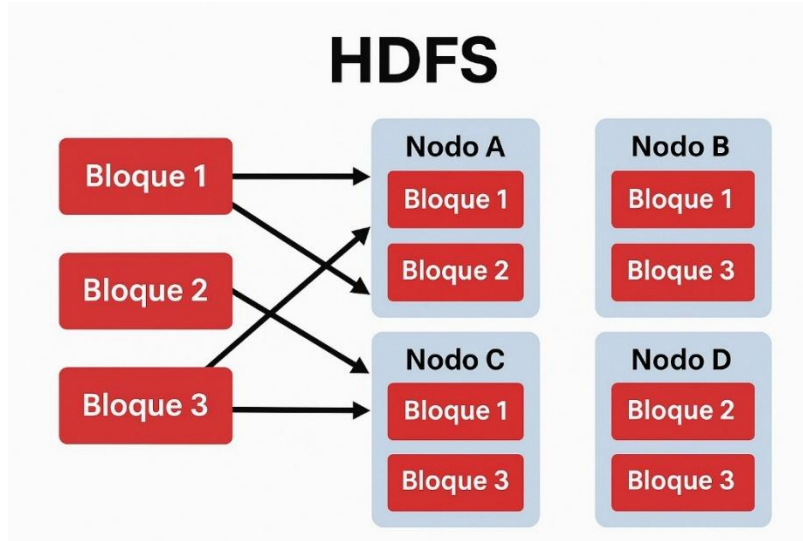
Hadoop

Spark

Bibliografía

Presentación de
prácticas

HDFS (Hadoop Distributed File System)



- **Divide** los datos en bloques
- Distribuye los bloques **en nodos**
- **Replica** los datos como respaldo

5. Hadoop como Middleware

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

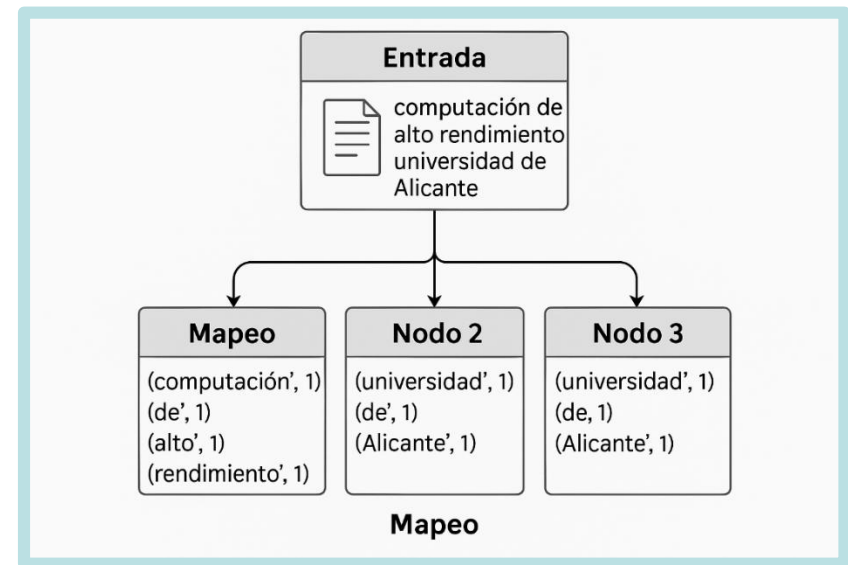
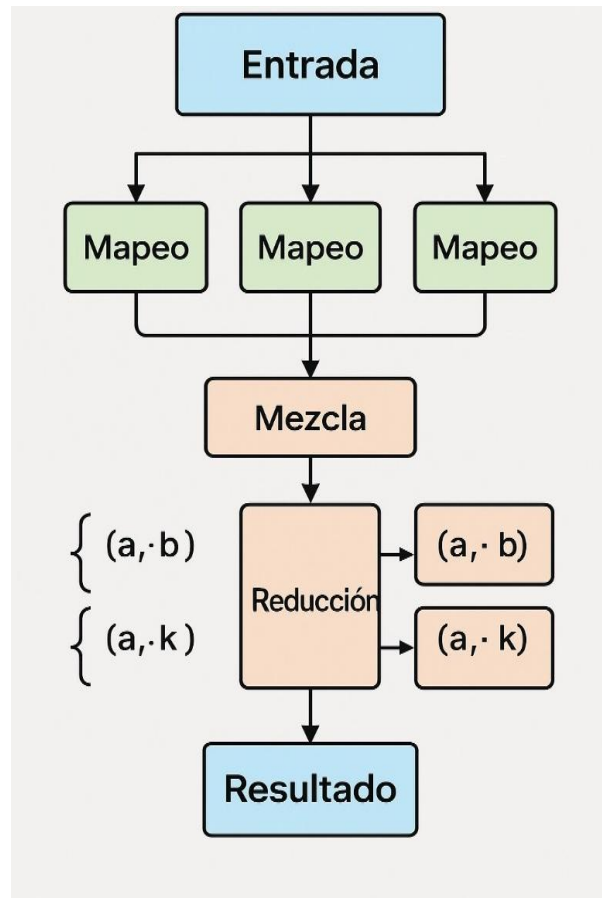
Hadoop

Spark

Bibliografía

Presentación de
prácticas

HDFS (Hadoop Distributed File System)



5. Hadoop como Middleware

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

Aplicaciones reales:

- **Análisis de registros médicos**
 - ¿Qué hace Map? Busca términos como **enfermedades, edades, regiones en millones de registros**.
 - ¿Para qué sirve? Estimar la **frecuencia** de enfermedades por zona o grupo de edad.
- **Análisis de comportamiento de usuarios**
 - ¿Qué hace Map? Recoge **clics, búsquedas, compras asociadas a un usuario**.
 - ¿Para qué sirve? Saber qué productos o **contenidos** son más populares o recomendados.
- **Motores de búsqueda y análisis web**
 - ¿Qué hace Map? **Extrae palabras** de millones de páginas web.
 - ¿Para qué sirve? Construir **índices de búsqueda**, detectar tendencias, analizar visitas.

6. Apache Spark como evolución de Hadoop

6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

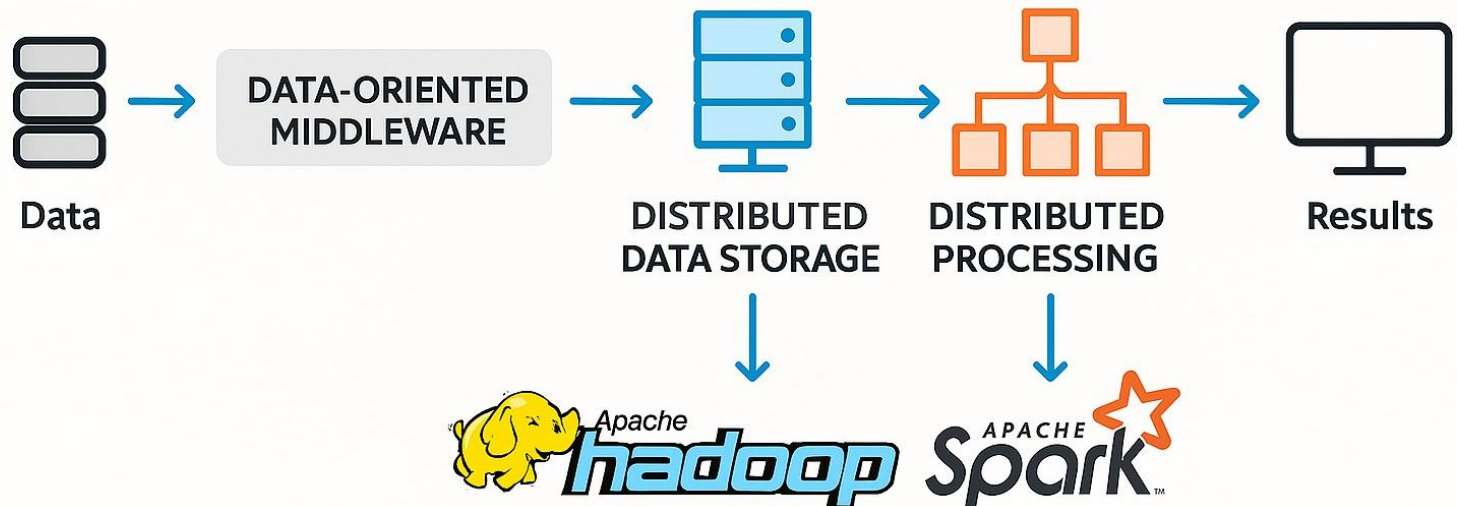
Spark

Bibliografía

Presentación de
prácticas

6.1 ¿Qué es Spark y qué mejora respecto a Hadoop?

- **Procesa** datos a gran velocidad
- **Planifica** los trabajos previamente
- No guarda datos en disco



6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

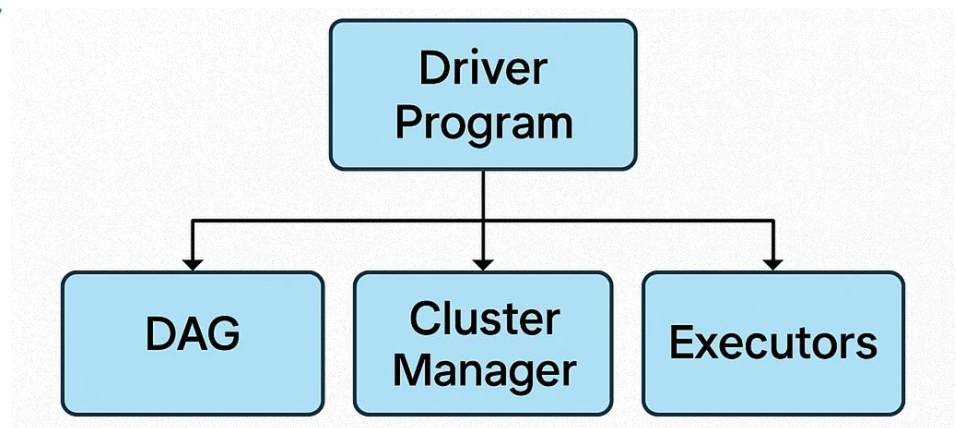
Spark

Bibliografía

Presentación de
prácticas

6.2 Procesamiento en memoria y arquitectura (RDDs, DAG, ejecutores)

- **Arquitectura Maestro-Trabajador**
- **Componentes principales:**
 - Driver Program
 - DAG
 - Cluster Manager
 - Executors



6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

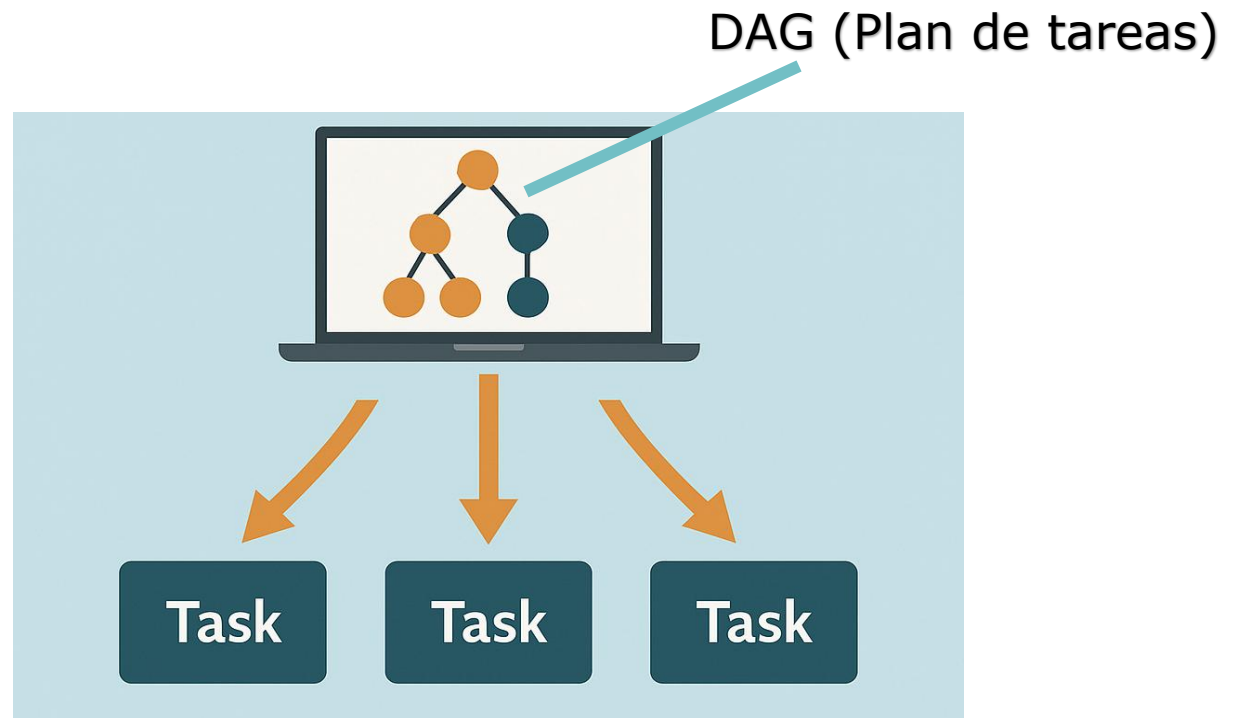
Spark

Bibliografía

Presentación de
prácticas

Driver Program:

- Organiza y supervisa el trabajo (es el cerebro del sistema)
- Crea un DAG



6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

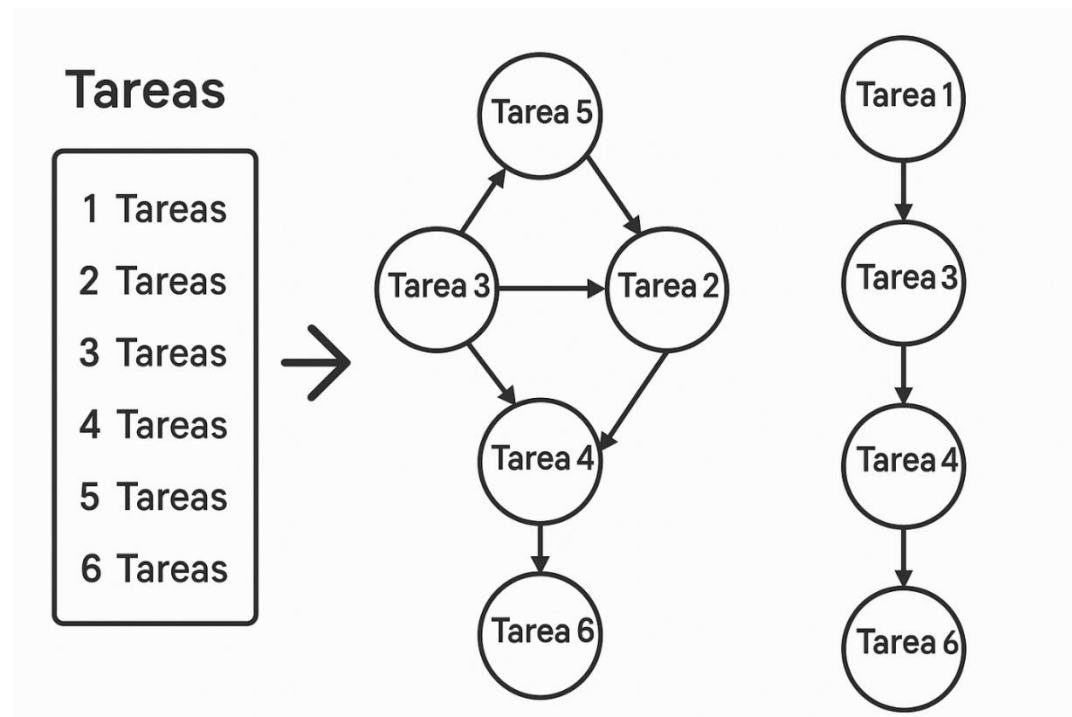
Spark

Bibliografía

Presentación de
prácticas

DAG (Grafo Dirigido Acíclico):

Primero traza el camino óptimo con todos los pasos,
y luego empieza a recorrerlo en el orden más eficiente posible.



6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

Ejemplo: un problema a resolver

- Queremos saber **cuántas veces aparece cada palabra** en una colección de textos almacenados en HDFS.
- **Datos de entrada:**
 - “computación de alto rendimiento universidad de Alicante”

6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

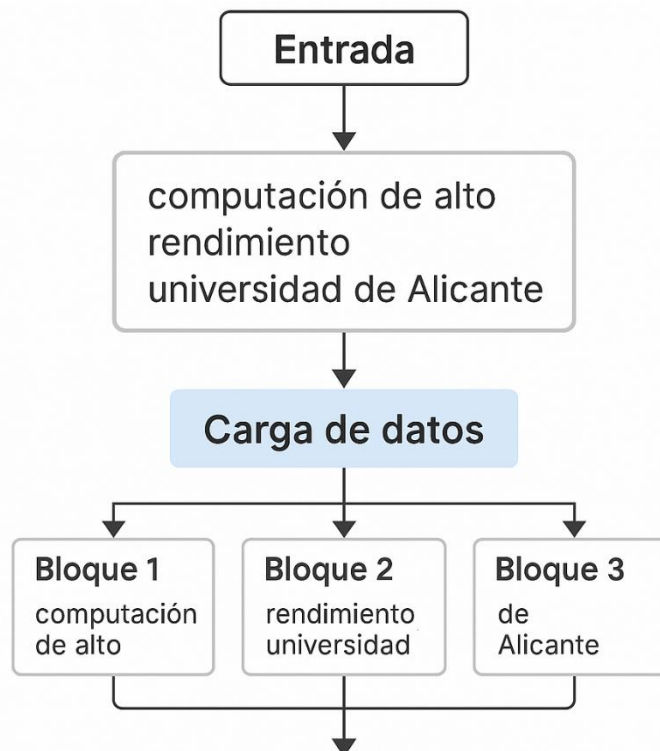
Spark

Bibliografía

Presentación de
prácticas

Ejemplo: un problema a resolver

- **Paso 1: Carga de datos:**
 - Spark carga el archivo desde **HDFS** (sistema de archivos distribuido).
 - Divide el texto en **bloques**.



6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

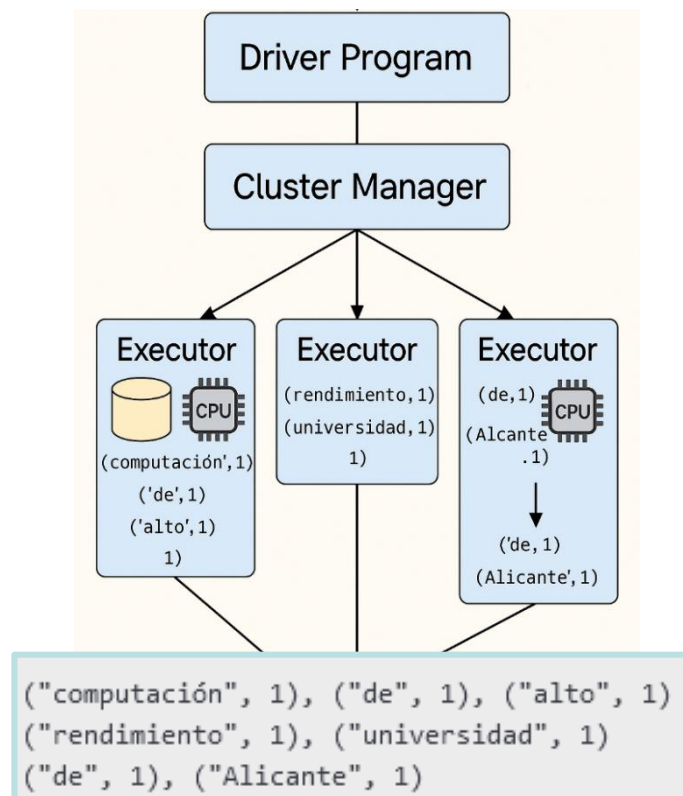
Bibliografía

Presentación de
prácticas

Ejemplo: un problema a resolver

Paso 2: Funciones Map

- Cada bloque es procesado por un **Executor**.
- Se aplica una función **Map** para convertir texto en pares:



6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

Ejemplo: un problema a resolver

Paso 3 – Shuffle

- Spark **Reagrupa** los pares por palabra
- Junta **todas las ocurrencias** de una misma palabra

```
"de" → [1, 1] → "de", 2  
"computación" → [1] → "computación", 1  
"alto" → [1] → "alto", 1  
"rendimiento" → [1] → "rendimiento", 1  
"universidad" → [1] → "universidad", 1  
"Alicante" → [1] → "Alicante", 1
```

6. Apache Spark como evolución de Hadoop

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

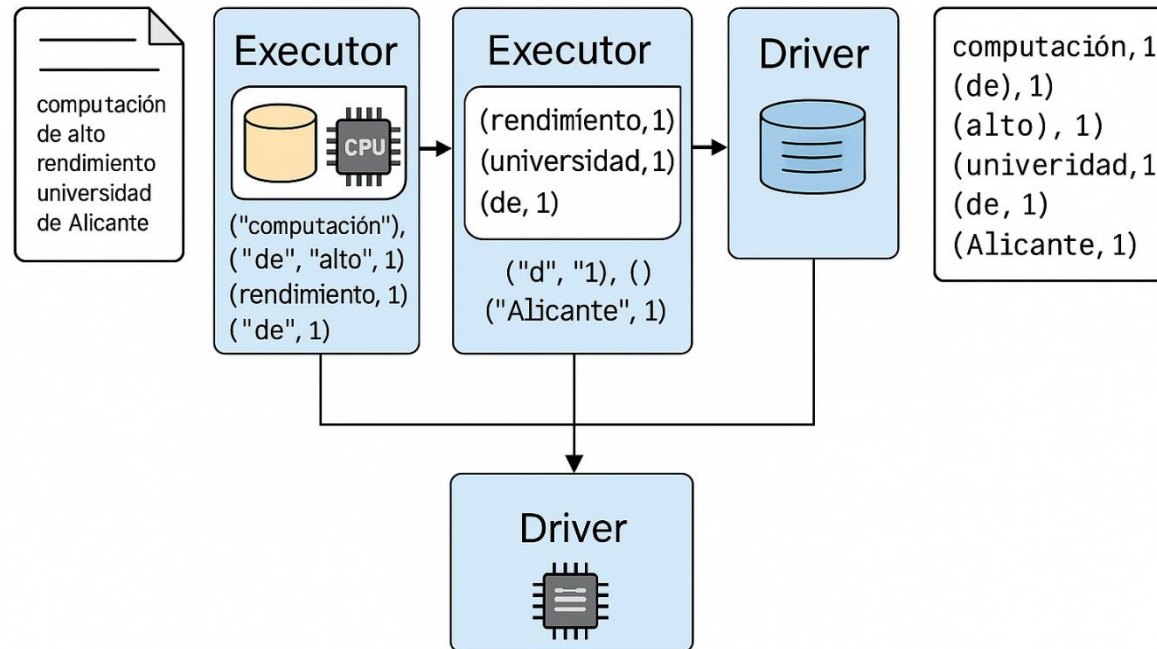
Presentación de
prácticas

Ejemplo: un problema a resolver

Paso 4 - Reduce y resultado final:

- Cada palabra se reduce a su conteo final
- El resultado se guarda en **HDFS** para el usuario

Entrada → Map → Shuffle → Reduce → Resultado



7. Bibliografía recomendada

7. Bibliografía recomendada

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

Bibliografía básica recomendada

- **Middleware y sistemas distribuidos**
- **Coulouris, G., Dollimore, J., Kindberg, T., Blair, G.**
Distributed Systems: Concepts and Design. 5ª ed. Addison-Wesley, 2011.
(*middleware, arquitecturas distribuidas y ejemplos reales*).
- **Tanenbaum, A. S. & van Steen, M.**
Distributed Systems: Principles and Paradigms. 2ª ed. Pearson, 2007.
(*fundamentos y ejemplos de middleware (CORBA, RPC, MOM...)*).
- **Roman, E. G., García-Molina, H., Ullman, J. D.**
Sistemas distribuidos y procesamiento paralelo. Pearson Educación, 2001.
(*Modelos de comunicación, sincronización y fallos.*)

7. Bibliografía recomendada

Índice

Introducción

Arquitectura

Tipos de
middleware

Ventajas e
inconvenientes

Hadoop

Spark

Bibliografía

Presentación de
prácticas

Artículos y documentación oficial (complementaria)

- Apache Software Foundation – Documentación oficial de Apache Hadoop
- Apache Spark Project – Documentación oficial de Spark
- Red Hat – What is Middleware?
→ <https://www.redhat.com/en/topics/middleware>
- Amazon AWS – ¿Qué es el middleware?
→ <https://aws.amazon.com/es/what-is/middleware/>