CESTE

Escuela Internacional de Negocios

Zaragoza (España)

¿Quién soy?

- Pablo Sanz Caperote
- Doble Grado en Ingeniería Informática – Matemáticas
- Profesional con más de 3 años de experiencia entorno al mundo de los datos.
- Casi 2 años trabajando con Databricks
- Varias certificaciones en Clouds (AWS, Azure, GCP)

www.linkedin.com/in/pablosanzcaperote

CESTE

Índice del curso

1. Fundamentos y diagnóstico de rendimiento
    - Funcionamiento de Spark
    - Cuellos de botella
    - Uso de herramientas

2. Técnicas de optimización
    - Buenas prácticas con código
    - Optimización de particionado y archivos
    - Uso de Delta Lake en Databricks

3. Diseño de pipelines escalables y sostenibles
    - Modularización, parámetros y manejo de errores
    - Orquestación con Databricks Workflows
    - Control de costes y clusters eficientes
    - Casos reales de optimización de pipelines

1. Fundamentos y diagnóstico de rendimiento

Índice | Fundamentos y diagnóstico de rendimiento

1. Ciclo de ejecución en Spark
2. Arquitectura interna de Spark y Databricks
3. Identificación de cuellos de botella
4. Herramientas de diagnóstico

Objetivos | Conocer las transformaciones de Spark, sus cuellos de botella y las herramientas de diagnóstico

1. Explicar cómo Spark ejecuta un pipeline de datos desde el código hasta la ejecución distribuida.
2. Identificar los cuellos de botella más comunes en Spark.
3. Usar las herramientas de diagnóstico en Databricks

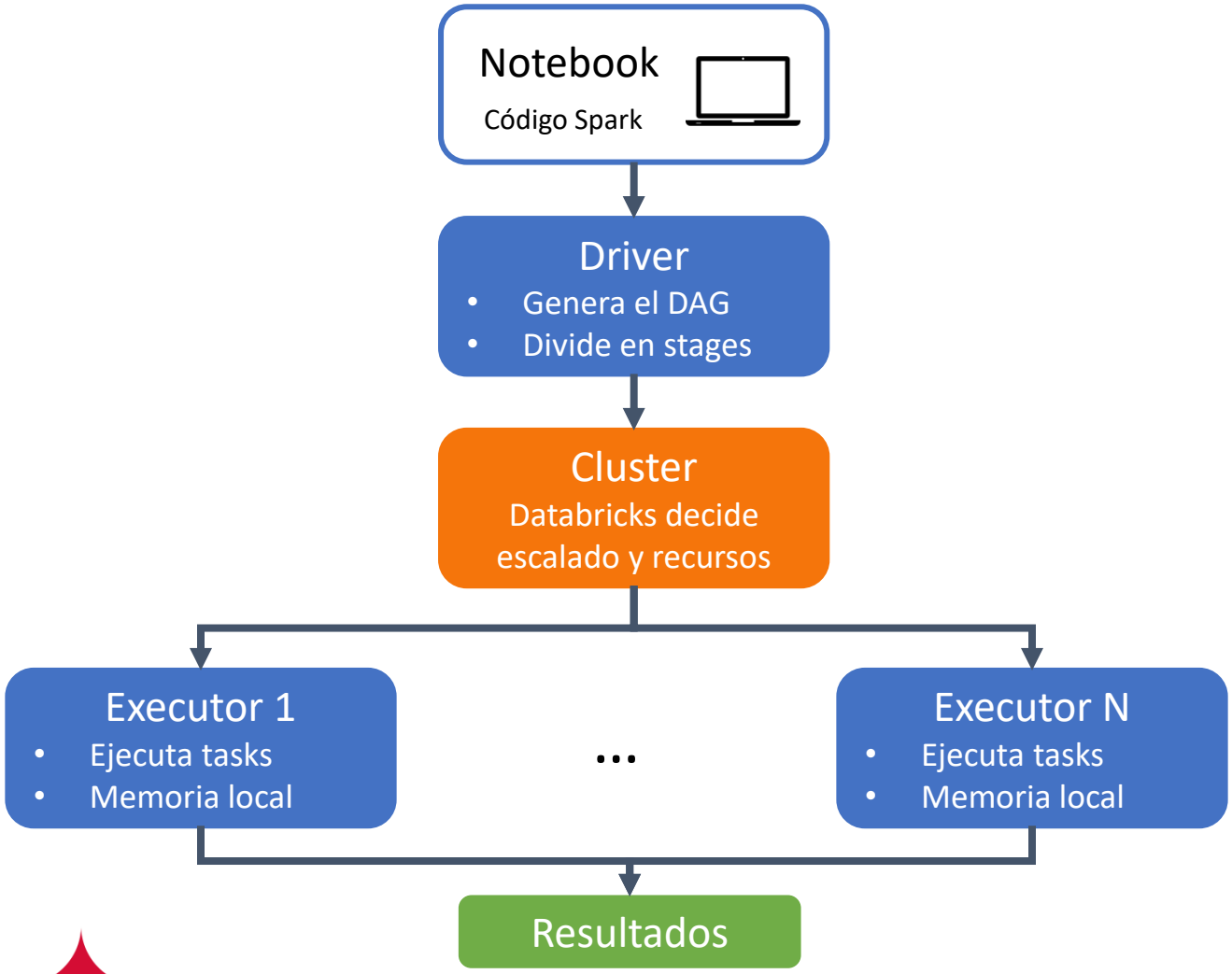
Ciclo de ejecución | Spark planifica primero y ejecuta después

Transformaciones (lazy)
Generan plan lógico
select, filter, join

Acciones
Disparan ejecución real
count, show, write

Spark no ejecuta hasta encontrar una acción

Spark y Databrick por dentro | Spark reparte el trabajo y Databricks gestiona automaticamente



Cuellos de botella | Detectarlos es el primer paso para optimizar un pipeline

Shuffles

- Redistribución de datos entre nodos
- Causas: join, groupBy,…
- Impacto: alto coste

Spark UI

Data Skew

- Particiones con datos muy desbalanceados
- Una tarea tarda mucho más que el resto

DAG Viewer

Particionado ineficiente

- Pocas particiones implica poco paralelismo
- Demasiadas particiones implica sobrecarga de tareas

explain()
queryExecution

Herramientas de diagnóstico en Databricks| No se puede optimizar lo que no se entiende. Lo primero es diagnosticar

databricks

www.ceste.es