**PREWORK**

Sesión 07

**Análisis de Big Data con Spark**

Uso de los componentes de Spark

**🎯 Objetivo**

* Comprender y delimitar el alcance de las acciones elementales en Spark
* Comprender y delimitar el alcance de las transformaciones elementales en Spark

### 🚀 **Desarrollo**

### **Parte 1: ¿Qué son las Acciones en Spark?**

Las acciones son operaciones que devuelven un valor al Usuario. Un dato interesante es que las transformaciones en Spark suelen ser perezosas, lo que quiere decir que Spark “recuerda” cada transformación que le hayamos aplicado a un RDD y las aplica de la forma más óptima en el momento en que llamamos a una acción nueva. Hasta que llamamos a esa acción Spark espera para diseñar el plan de ejecución que sea más eficiente.

Tenemos la opción de persistir los resultados intermedios de cada transformación de manera que Spark no tenga que volver a computar todo el proceso cada vez que ejecutemos una acción.

Algunas de las acciones más habituales son:

#### **Reduce()**

Realiza una agregación del conjunto de datos de entrada, lo que suele ser un resultado de una función Map

#### **Collect()**

Devuelve el contenido del RDD sobre el que lo llamamos de vuelta al programa conductor (driver). Habitualmente es un subconjunto de los datos de entrada que hemos transformado y filtrado aplicando alguna de las transformaciones disponibles.

#### **Count()**

Como se puede suponer devolverá el número total de elementos que hay en un RDD

#### **Take(n)**

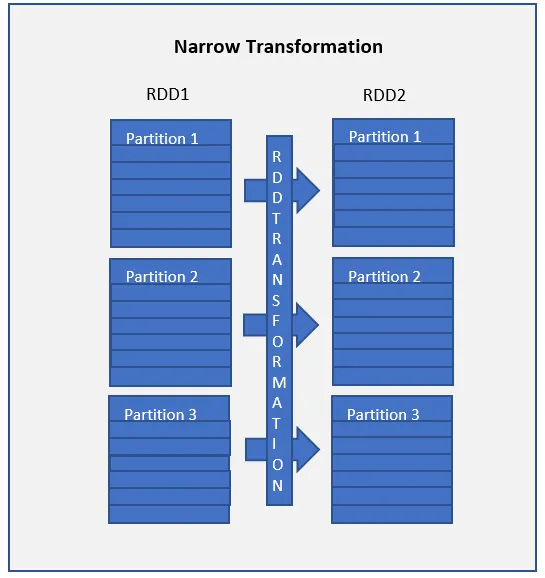
Se trata de una función muy util que permite echar una ojeada a los datos resultantes del proceso al permitir obtener los primeros n elementos del RDD.

### **Parte 2: ¿Qué son las Transformaciones en Spark?**

Las transformaciones son una operación elemental de Spark para operar sobre los ya conocidos RDD: a partir de un solo RDD, podemos generar uno o varios nuevos RDD’s derivados. Los tipos de transformaciones que hay son las Narrow (cortas) y Wide (amplias).

**Narrow (cortas)**

Las transformaciones cortas o estrechas (narrow) son el resultado de las funciones map () y filter () y estos computan datos que viven en una sola partición, lo que significa que no habrá ningún movimiento de datos entre particiones para ejecutar transformaciones estrechas.



**Wide (amplias)**

Las transformaciones amplias (Wide) son el resultado de las funciones groupByKey() y reduceByKey() y estas computan datos que están alojados en muchas particiones, lo que significa que habrá movimientos de datos entre las n particiones para ejecutar transformaciones... más amplias. Dado que esta operación “baraja” los datos, también se denominan transformaciones barajadas.

