Actividad 14

Tranformaciones:

**Map**: Aplica una función a cada elemento del RDD y devuelve un nuevo RDD con los resultados. Esto es útil para transformaciones simples y permite el procesamiento paralelo de datos.

**Filter**: Crea un nuevo RDD con solo los elementos que cumplen con una condición específica. La función de filtro debe devolver un valor booleano .

**FlatMap**: Similar a MAP, pero permite a la función devolver múltiples elementos para cada entrada, y aplana el resultado en una sola lista. Ideal para transformar y expandir elementos.

**Union**: Combina dos RDDs y devuelve un nuevo RDD que contiene todos los elementos de ambos, incluidos duplicados.

**Intersection**: Devuelve un RDD con los elementos que están presentes en ambos RDDs, eliminando duplicados. Útil para encontrar elementos comunes.

**Distinct**: Elimina duplicados y devuelve un nuevo RDD con los elementos únicos. Esto puede ser costoso computacionalmente, ya que requiere una redistribución de los datos.

**GroupByKey**: Agrupa los valores de cada clave en un RDD de pares clave-valor. Es costoso en comparación con reduceByKey porque requiere una agrupación y transferencia de datos más extensa.

**ReduceByKey**: Similar a groupByKey, pero aplica una función de agregación en los valores de cada clave. Más eficiente, ya que realiza la agregación localmente en cada partición antes de transferir los datos.

**SortByKey**: Ordena los elementos en un RDD basado en sus claves. Utilizado principalmente en RDDs de pares clave-valor.

**Join**: Realiza una unión entre dos RDDs de pares clave-valor, creando un nuevo RDD con valores de ambas RDDs en claves coincidentes. Es útil para combinaciones de conjuntos de datos.

**Cogroup**: Combina dos RDDs por clave, pero en lugar de emparejar elementos, agrupa los valores en listas, lo cual facilita la comparación y transformación de datos agrupados.

**Coalesce**: Reduce el número de particiones del RDD, ideal para optimizar el rendimiento en tareas posteriores al filtrado o cuando se necesita reducir el costo computacional.

Acciones

**Reduce**: Aplica una función de reducción a todos los elementos del RDD y devuelve un único resultado. Es adecuado para agregaciones globales.

**Collect**: Trae todos los elementos del RDD a una lista en el nodo principal. Útil para depuración, pero debe usarse con cuidado, ya que requiere que todos los datos quepan en la memoria local.

**Count**: Devuelve el número de elementos en el RDD.

**First**: Recupera el primer elemento del RDD.

**Take**: Devuelve una lista con los primeros N elementos del RDD.

**SaveAsTextFile**: Guarda el contenido del RDD en un archivo de texto en el sistema de archivos distribuido.

**Max y Min**: Obtiene el valor máximo o mínimo en el RDD basado en un orden.

**CountByKey**: Cuenta los elementos por cada clave en un RDD de pares clave-valor.

**Foreach**: Aplica una función a cada elemento del RDD sin devolver un nuevo RDD. Ideal para acciones secundarias, como registrar datos.

<https://spark.apache.org/docs/latest/>