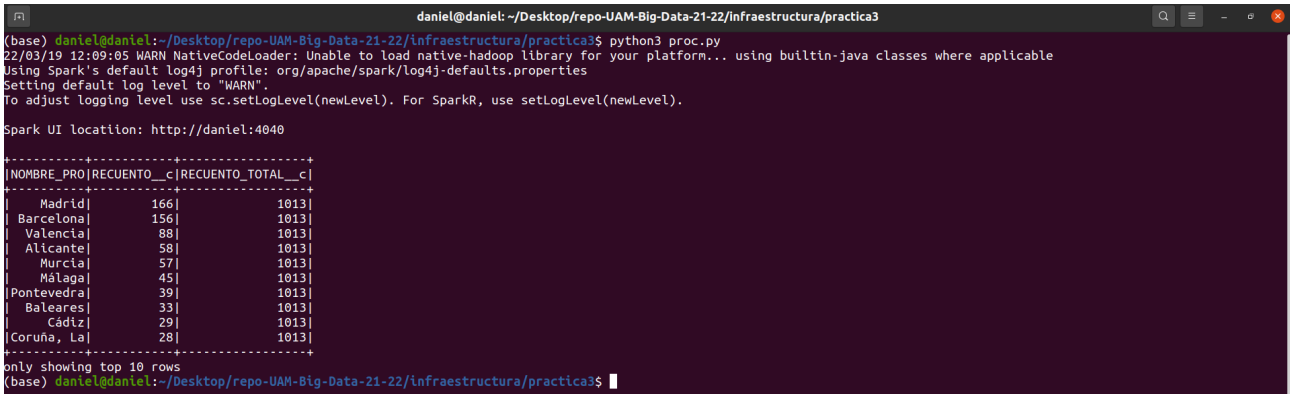
Infraestructura para Big Data

Práctica 3: Ecosistema Spark

Daniel Pérez Efremova

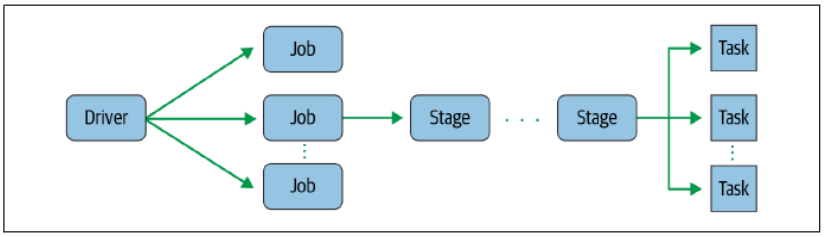
1. Resultado de la operación

En la figura se muestra el resultado de ejecutar el script pedido.



2. Descripción de elementos identificados en la UI

En esta sección procedemos a identificar mediante la UI los elementos fundamentales de una aplicación Spark. Aunque solo se piden Tasks y Stages, se incluyen también los Jobs por ser la entidad principal de la aplicación ya que juega un papel central en el proceso de paralelización. En la figura se muestra un esquema de las entidades que participan en la ejecución de una aplicación Spark.

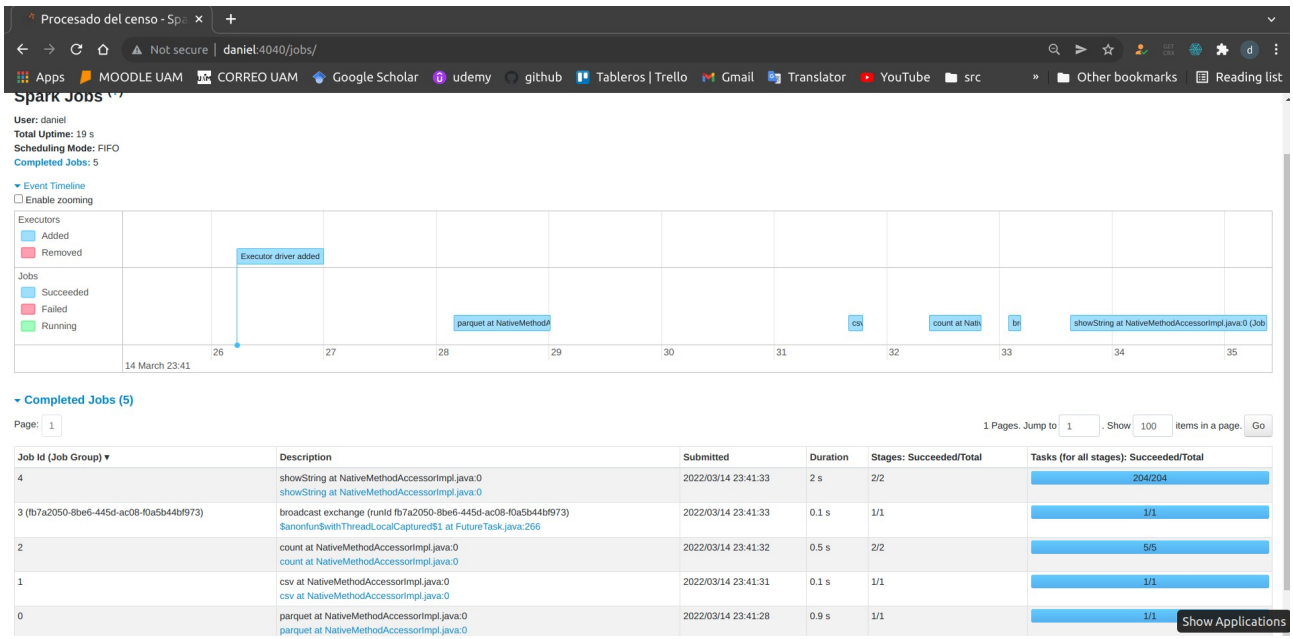


2.1 Identificación de jobs.

Un job consiste en múltiples tareas que aparecen en respuesta a una acción en la aplicación Spark (ej .collect()). El Driver de la aplicación se encarga de convertir la aplicación en uno o múltiples trabajos. En un cluster Spark la paralelización se consigue al repartir los jobs en distintas máquinas.

En la imagen se muestran los trabajos en que se divide la aplicación propuesta. En resumen, se puede leer en las descripciones que **los trabajos son**:

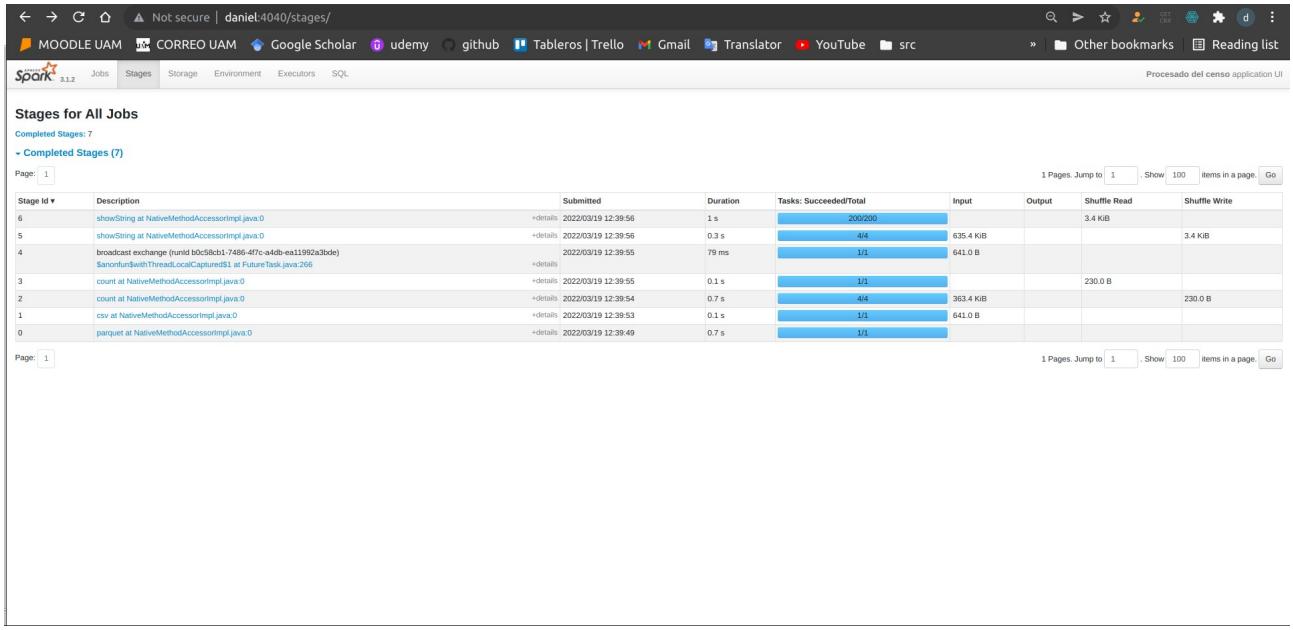
- Lectura de un fichero parquet
- Lectura de un fichero csv
- Operación de recuento
- Operación Join (Broadcast join entre dos dataframes de tamaños distintos)
- Impresión por consola de resultados



2.2 Identificación de stages.

Un stage es una división de un job en un conjunto de tareas. Las tareas de cada stage pueden realizarse independientemente del resto de stages. Sin embargo, los stages son dependientes entre sí (secuenciales).

En la figura se ven los distintos stages que se forman cuando el driver define los jobs.

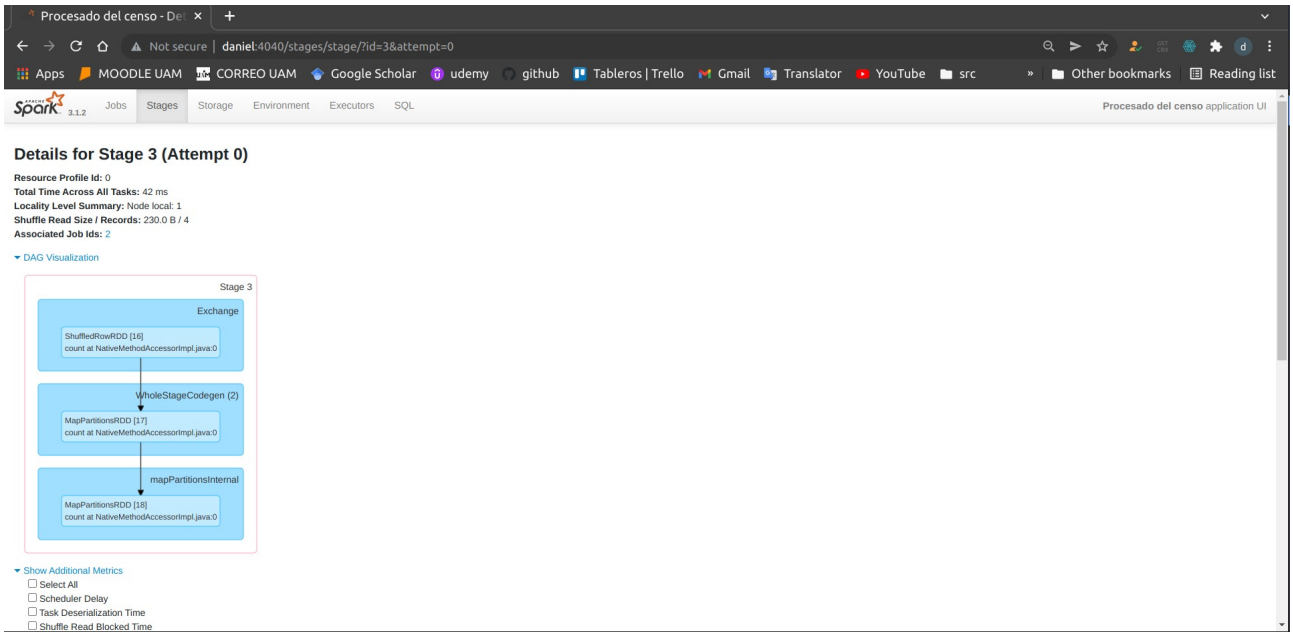


Al no ser una aplicación excesivamente compleja, los stages son muy similares a los jobs.

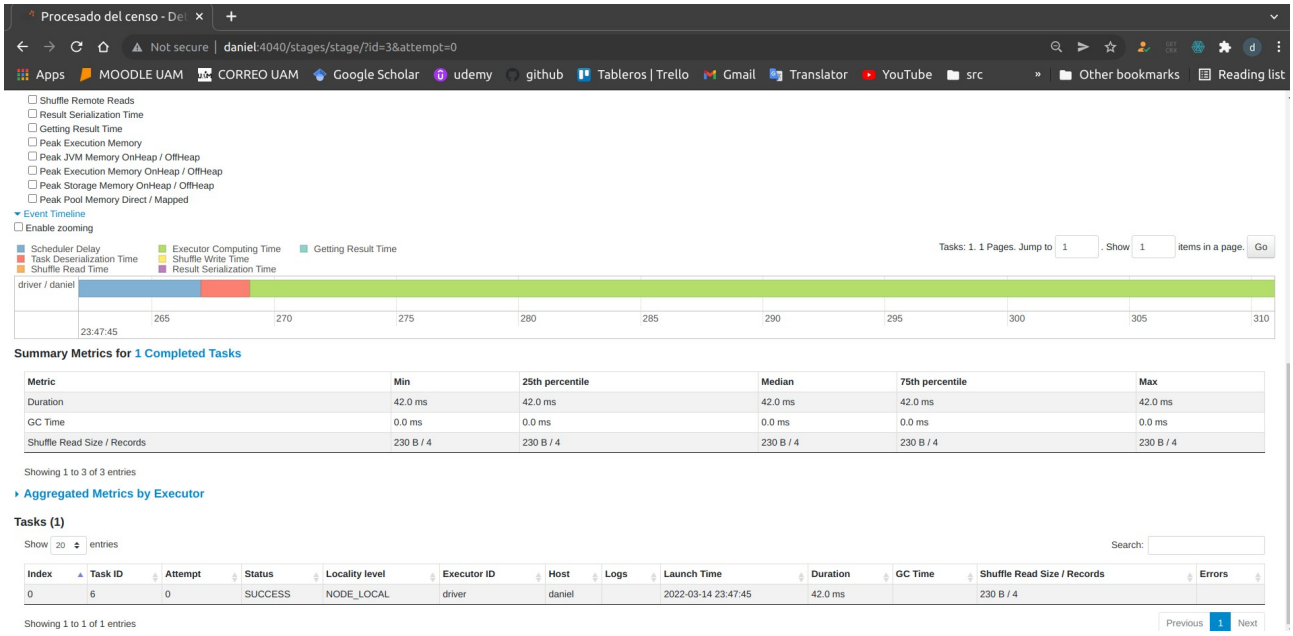
2.3 Identificación de tasks.

Una Task es la unidad mínima de ejecución en un aplicación Spark, se dirige a un único core y trabaja sobre una única partición del fichero que se esté procesando.

A modo de ejemplo, se muestran las tasks a ejecutar para el stage 3. Este stage aglutina tasks relacionadas con un count. Vemos el **Exchange**, que a grandes rasgos significa que la task consiste en enviar datos a la red, y en las siguientes tasks, mediante **MapPartition** aplica la función count a todas las particiones.

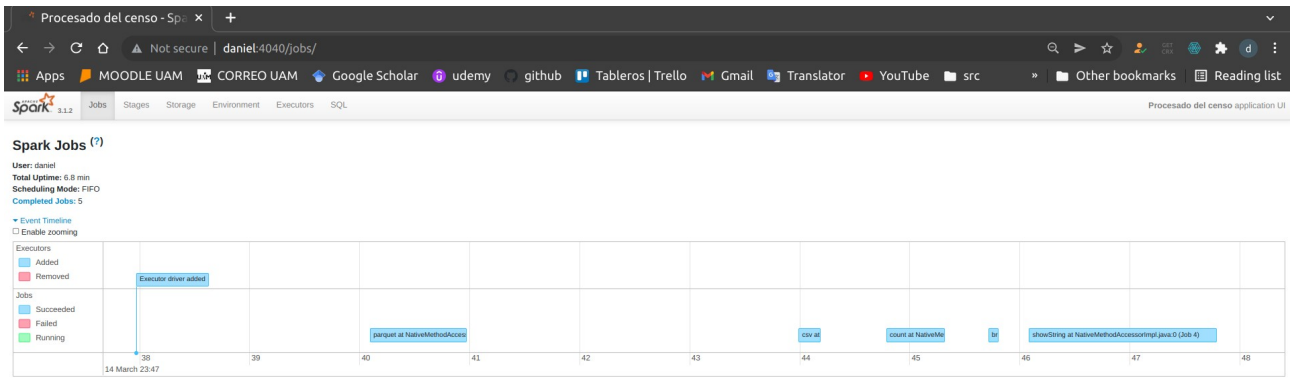


En la parte inferior del stage se puede ver el tiempo de ejecución segmentado por el tipo de recurso consumido: Scheduler, Executor, SerDe, etc.



2.4 Línea temporal

En la figura inferior se observa la línea temporal de ejecución de la aplicación. Se muestran los jobs y el tiempo total de ejecución.



2.5 Resultados de las operaciones con dataframes

En la pestaña SQL se puede ver el flujo de ejecución de las operaciones con dataframes. Se adjunta en pdf la impresión del flujo completo de la aplicación. Se pueden identificar pasos clave en las operaciones:

- **Scan parquet**: para leer un fichero parquet y poder trabajar con él.
- **Filter**: para coger las filas que pertenecen a personas nacidad en Francia o Portugal.
- **HashAggregate**: para agrupar por provincias y poder contar el número de personas por provincia de residencia
- **BroadcastHashJoin:** para asignar a cada código de provincia su nombre desde el csv complementario.