Tarea #2 - Big Data

Objetivo

Introducir a los estudiantes al procesamiento de tipos de datos complejos, agregaciones y métricas con Apache Spark.

Resultados esperados

Para esta asignación se espera que los estudiantes concluyan dos entregables relacionados:

- Un programa principal que recibirá como entrada un patrón de archivos tipo JSON que contienen las transacciones diarias de diferentes cajas en supermercados y generará múltiples archivos de salida, descritos posteriormente.
- Una serie de pruebas unitarias que permitan corroborar la correctitud de las diferentes funciones internas al programa.

Entrega: Archivo comprimido con código y PDF en TEC Digital a más tardar el 5 de diciembre de 2019 a las 11:59 AM. Pueden usar un *notebook* como parte de la documentación.

Datos de entrada

Cada archivo de entrada será un objeto JSON que describe las compras realizadas en una caja de un supermercado. La expectativa es que cada corrida del programa cargará un número posiblemente grande de estos archivos.

El formato de cada archivo es:

- Un atributo **numero_caja** numérico que sirve como identificador
- Un nodo **compras** que contiene la lista de cada una de las compras hechas por clientes.
- Cada **compra** es una colección de productos
- Cada producto tiene tres atributos: **nombre**, **cantidad**, **precio_unitario**

El siguiente es un ejemplo del contenido que tendría un archivo:


```
{
	"numero_caja":	"45",
	"compras": [
		[
			{
				"nombre": "manzana",
				"cantidad": "3" ,
				"precio_unitario": "22"
			},
			{
				"nombre": "brocoli",
				"cantidad": "2" ,
				"precio_unitario": "33"
			}
		],
		[
			{
				"nombre": "aguacate",
				"cantidad": "1" ,
				"precio_unitario": "5000"
			}
		]
	]
}
```


Para la ejecución del programa principal, los estudiantes deberán proveer 5 archivos de prueba con al menos 10 compras diferentes cada uno; cada compra con un número variable de productos.

Los archivos deben contener JSON válido.

Programa principal (25 puntos)

Se espera que los estudiantes entreguen un manual en PDF con las instrucciones para ejecutar el programa principal. Idealmente esto debería realizarse con una simple llamada a "spark-submit programaestudiante.py archivos*.json"

Cualquier detalle necesario para la ejecución debe agregarse en este documento. La imposibilidad de ejecución del programa impedirá la obtención de los puntos.

El producto de la ejecución del programa será una serie de archivos de texto:

- total_productos.csv: contiene 2 columnas que representan el nombre de cada producto y la cantidad total de ese producto vendida en todas las cajas.
- total_cajas.csv: contiene 2 columnas que representan el identificador de cada caja y el total vendido por esa caja
- metricas.csv: contiene 2 columnas que representan el tipo de métrica y su valor. En particular deberá generarse las siguientes métricas
    - caja_con_mas_ventas: identificador de la caja con más ventas (en dinero)
    - caja_con_menos_ventas: identificador de la caja con menos ventas (en dinero)
    - percentil_25_por_caja: si se ordenan todas las cajas de menor cantidad de ventas a mayor, cuál valor monetaria representa el percentil 25
    - percentil_50_por_caja
    - percentil_75_por_caja
    - producto_mas_vendido_por_unidad: nombre del producto que tuvo más unidades vendidas
    - producto_de_mayor_ingreso: nombre del producto que generó más cantidad de dinero (e.g. cantidad * precio)

Pruebas esperadas

Para realizar las pruebas unitarias se espera que los estudiantes piensen en las diferentes partes necesarias para conseguir el objetivo final. Éstas deberán arrancar de datos que se encuentren en memoria. En este caso, pueden arrancar de dataframes en los que cada fila es el string del JSON para una caja.

Los estudiantes deberán diseñar sus propias pruebas unitarias, utilizando la discusión en clase como base para guiar su diseño. Para efectos de evaluación se espera que haya suficientes pruebas para probar las diferentes áreas funcionales. Se espera que cada área funcional tenga su propia función de entrada:

- **Total de productos: 20 puntos**

- **Total de cajas: 20 puntos**

- **Métricas: 5 puntos cada una**

Las pruebas deben cubrir casos excepcionales. Tanto el profesor y asistente se reservan el derecho de agregar pruebas unitarias adicionales en cada apartado para asegurar el correcto funcionamiento.

Se recuerda a los estudiantes que la nota será completamente derivada de las pruebas unitarias. Deberá ser posible ejecutar las pruebas simplemente al correr el comando pytest en la carpeta que se entrega con el código.