Big Data en el Programa para Ciencia de los Datos, FUNDATEC

Tarea 2: Alejandro Quesada, Joshua Solís

El repositorio actual comprende los archivos utilizados durante el trabajo en la tarea.

Manual de Ejecución

Debe ejecutarse en este orden:

1. `bash build.sh` para crear la imagen con Docker. De no poder ejecutar el comando en bash, también existe un `build.bat` para usuarios en Windows, o bien se puede poner el comando directamente en la terminal: `docker build --tag bd-tarea2-new .`
2. `bash run-container.sh`. Esto levanta un contenedor usando la imagen del paso anterior, con el nombre `bd-tarea2-new`, y el contenedor se llamará `bd-tarea2-container`. De nuevo si no le es disponible bash en esta etapa, puede ejecutar este comando: `docker run -it --rm --name bd-tarea2-container bd-tarea2-new /bin/bash`
3. Procure asegurarse de estar desde el shell de Bash dentro de la imagen. Debería mostrarle `bash-5.0#` en la línea de comandos.
4. `bash run-program.sh` se encarga de ejecutar el `spark-submit ...` y las pruebas unitarias en un solo script

Ejecución separada de las pruebas unitarias (pytest)

Para las pruebas unitarias, aunque el script `run-program.sh` contiene la instrucción necesaria, se pueden ejecutar por separado utilizando el comando: `pytest`.

Datos utilizados para el programa principal

Partimos de la creación del mínimo de 5 archivos de muestra con 10 compras en cada uno, los cuales son `sample1.json`, `sample2.json` ... así hasta `sample5.json`. Estos vienen en el formato especificado dentro del ejemplo de la tarea.

Dentro del programa, el manejo del vector de argumentos y la llamada a las funciones auxiliares se encuentran en el archivo `programaestudiante.py`. Las funciones auxiliares para los cálculos necesarios y la generación de archivos csv, se encuentran respectivamente en los archivos `funciones.py` y `generar_csv.py`. Además, el programa principal también es el encargado de imprimir los DataFrames en pantalla antes de llegar a las pruebas, así si se desea revisar de manera manual contra los resultados obtenidos, es más sencillo.

Otras notas importantes

- Por la facilidad de no tener que estar reconstruyendo la imagen cada vez que se le hacían cambios al código para volver a probarlo (y a la vez ahorrarse el uso de un editor pequeño dentro de la imagen, como NANO), mucho del trabajo se probó en un entorno virtual de python (venv) que importaba todo lo necesario similar al contenedor de docker. Esto nos presentó algunas dificultades porque omitimos que debiamos "devolver" o desactualizar la versión de python del entorno virtual para que coincida con la que se utiliza en la imagen que construye el `Dockerfile`. Esto causa que las pruebas arrojen unos warnings, pero aún así deben funcionar, fue probado varias veces.
- La generación de archivos csv se puede verificar con el comando `ls *.csv` cuando termine de ejecutar el programa principal, ya que antes de esta ejecución, no habrá ningún archivo `.csv` en el directorio de trabajo, pero después de este proceso, deben haber 3: `total_productos.csv`, `total_vendidos.csv` y `metricas.csv`.