Manual Tarea 1

Estudiante: María Yorleni Alfaro Alfaro

En el presente documento se encuentran las instrucciones respectivas para ejecutar el programa principal.

Pasos para la ejecución:

1. Primero se debe descomprimir el archivo tarea1.zip entregado en la tarea.

   Los siguientes son los archivos que se encuentran en dicha carpeta y son necesarios para la correcta ejecución:

   - **tarea1.py**: es el programa principal, desde el cual se realiza el llamado a las diferentes funciones.
   - **tarea1_funciones.py**: en este archivo se encuentra el código de cada una de las funciones desarrolladas (que son llamadas desde el programa principal "tarea1.py")
   - **test_tarea1.py**: contiene una serie de pruebas unitarias que permitan corroborar la correctitud de las diferentes funciones internas al programa.
   - **Archivos .csv**:
     - ciclista.csv
     - ruta.csv
     - actividad.csv
   - **Archivos de configuración**:
     - __init__.py
     - conftest.py
     - Dockerfile

2. Abrir Power Shell o línea de comandos
3. Navegar al directorio donde se descomprimieron los archivos del código fuente
4. Realizar el build de la imagen con el comando: **docker build --tag tarea1 .**
5. Ejecutar la imagen del Docker con el comando: **docker run -i -t tarea1 /bin/bash**
6. Para la ejecución del programa principal, ejecutar el siguiente comando: "**spark-submit tarea1.py ciclista.csv ruta.csv actividad.csv 1**".

   Nota: El último argumento de ese comando corresponde al valor de **N**, dado que en la tarea lo que se solicita para el resultado final es el top N de ciclistas por provincia. Para la ejecución anterior, 1 corresponde al top 1 de ciclistas por provincia. Si por ejemplo se desea consultar el top 2 debe ejecutar el siguiente comando: "spark-submit tarea1.py ciclista.csv ruta.csv actividad.csv 2". Por lo tanto en ese último argumento debe indicar el número deseado.

7. Para la ejecución de las pruebas, ejecutar el siguiente comando: "**pytest test_tarea1.py -v**".

Con la ejecución del comando anterior se muestra el resultado de cada una de las pruebas.

Pruebas realizadas:

- **Sección de Unión de los datos:**
  - **test_join_normal_dataframes**: prueba del join con datos en los 3 datasets (actividades de ciclistas y rutas existentes); se realiza el join normalmente.
  - **test_join_ciclista_no_tiene_actividad**: prueba con datos cuando existe un ciclista que aún no ha hecho ninguna actividad; el registro del ciclista no se toma en cuenta en el join.
  - **test_join_ciclista_ejecuta_misma_ruta_varias_veces_al_dia**: prueba cuando el mismo ciclista ejecuta la misma ruta múltiples veces en un día; cada uno de los registros se toma en cuenta por separado en el join.
  - **test_join_ruta_no_tiene_actividad**: prueba cuando existe una ruta en la cual aún no se ha registrado actividad; el registro la ruta no se toma en cuenta en el join.
  - **test_join_actividades_con_ciclistas_no_registrados**: prueba cuando hay actividades de ciclistas que no existen; el registro de la actividad no se toma en cuenta en el join.
  - **test_join_actividades_con_rutas_no_registradas**: prueba cuando hay actividades de rutas que no existen; el registro de la actividad no se toma en cuenta en el join.

- **Sección de Agregaciones parciales:**
  - **test_kilometros_por_ciclista_normal**: prueba de obtener los kilómetros recorridos por ciclista, por ruta, por provincia y por día, cuando en los datos de entrada (del join) viene un solo registro por ciclista para una ruta, con kilómetros válidos; se obtiene el registro de cada ciclista con la ruta, la provincia, el día y los kilómetros recorridos.
  - **test_KmCiclista_ciclista_ejecuta_misma_ruta_varias_veces_al_dia**: prueba de obtener los kilómetros recorridos por ciclista, por ruta, por provincia y por día, cuando un ciclista ejecuta misma ruta varias veces al día; se obtiene el registro de cada ciclista con la ruta, la provincia, el día y la suma de kilómetros recorridos (para esa ruta y ese día).
  - **test_KmCiclista_kilometros_en_cero**: prueba de obtener los kilómetros recorridos por ciclista, por ruta, por provincia y por día, cuando los kilómetros vienen en 0; se excluyen los registros que vienen con kilómetros en 0.
  - **test_KmCiclista_kilometros_en_null**: prueba de obtener los kilómetros recorridos por ciclista, por ruta, por provincia y por día, cuando los kilómetros vienen en null; se excluyen los registros que vienen con kilómetros en null.
  - **test_KmCiclista_kilometros_negativos**: prueba de obtener los kilómetros recorridos por ciclista, por ruta, por provincia y por día, cuando los kilómetros vienen con un valor negativo; se excluyen los registros que vienen con kilómetros negativos.

- **Sección de Resultados finales:**
  - **test_top1_ciclistas_por_provincia_total_km_un_registro_por_ciclista**: prueba de obtener el top 1 de ciclistas por provincia en total de kilómetros, cuando en los

datos de entrada viene un solo registro por ciclista para una ruta; se obtiene el ciclista con más kilómetros recorridos para cada provincia.

- **test_top1_ciclistas_por_provincia_total_km_varios_registros_por_ciclista**: prueba de obtener el top 1 de ciclistas por provincia en total de kilómetros, cuando en los datos de entrada vienen varios registros por ciclista de diferentes rutas; obtiene el ciclista con mayor total de kilómetros recorridos (suma de kilómetros) para cada provincia.
- **test_top1_ciclistas_por_provincia_total_km_ciclistas_Empatados**: prueba de obtener el top 1 de ciclistas por provincia en total de kilómetros, cuando hay ciclistas empatados, es decir, que tienen la misma cantidad de kilómetros recorridos; en este caso elige al de menor número de cédula (se obtiene el ciclista con más kilómetros recorridos y menor cédula, para cada provincia).
- **test_top2_ciclistas_por_provincia_total_km**: prueba de obtener el top 2 de ciclistas por provincia en total de kilómetros; se obtienen los dos ciclistas con más kilómetros recorridos (ordenado en total de kilómetros descendente) para cada provincia.
- **test_top5_ciclistas_por_provincia_total_km**: prueba de obtener el top 5 de ciclistas por provincia en total de kilómetros; se obtienen los 5 ciclistas con más kilómetros recorridos (ordenado en total de kilómetros descendente) para cada provincia.
- **test_top1_ciclistas_por_provincia_promedio_KmDia_un_registro_por_ciclista**: prueba de obtener el top 1 de ciclistas por provincia en promedio de kilómetros por día, cuando en los datos de entrada viene un solo registro por ciclista para una ruta; se obtiene el ciclista con mayor promedio de kilómetros por día, para cada provincia.
- **test_top1_ciclistas_por_provincia_promedio_KmDia_varios_registros_por_ciclista**: prueba de obtener el top 1 de ciclistas por provincia en promedio de kilómetros por día, cuando en los datos de entrada vienen varios registros por ciclista de diferentes rutas; se obtiene el ciclista con mayor promedio de kilómetros por día (suma de kilómetros dividido entre la cantidad de días en que el ciclista tuvo actividad) para cada provincia.
- **test_top1_ciclistas_por_provincia_promedio_KmDia_ciclistas_Empatados**: prueba de obtener el top 1 de ciclistas por provincia en promedio de kilómetros por día, cuando hay ciclistas empatados, es decir, que ambos tienen el mismo promedio de kilómetros recorridos por día; en este caso elige al de menor número de cédula (se obtiene el ciclista con mayor promedio de kilómetros por día y menor cédula, para cada provincia).
- **test_top2_ciclistas_por_provincia_promedio_KmDia**: prueba de obtener el top 2 de ciclistas por provincia en promedio de kilómetros por día; se obtienen los dos ciclistas con mayor promedio de kilómetros por día (ordenado en promedio de kilómetros por día descendente) para cada provincia.
- **test_top5_ciclistas_por_provincia_promedio_KmDia**: prueba de obtener el top 5 de ciclistas por provincia en promedio de kilómetros por día; se obtienen los 5

ciclistas con mayor promedio de kilómetros por día (ordenado en promedio de kilómetros por día descendente) para cada provincia.

- **test_unir_dataframes_Top_1_ciclistas_por_provincia**: prueba la unión del dataset que contiene el top 1 de ciclistas por provincia en total de kilómetros con el dataset que contiene el top 1 de ciclistas por provincia en promedio de kilómetros por día; se obtiene la unión de ambos datasets.
- **test_unir_dataframes_Top_2_ciclistas_por_provincia**: prueba la unión del dataset que contiene el top 2 de ciclistas por provincia en total de kilómetros con el dataset que contiene el top 2 de ciclistas por provincia en promedio de kilómetros por día; se obtiene la unión de ambos datasets.
- **test_unir_dataframes_Top_5_ciclistas_por_provincia**: prueba la unión del dataset que contiene el top 5 de ciclistas por provincia en total de kilómetros con el dataset que contiene el top 5 de ciclistas por provincia en promedio de kilómetros por día; se obtiene la unión de ambos datasets.

Restricciones del programa:

- Se asume que los kilómetros son valores numéricos (no vienen en texto, con espacios, etc).
- Se asume que la fecha siempre viene en formato yyyy-mm-dd.
- Se asume que las columnas siempre vienen en el mismo orden mencionado en el enunciado de la tarea.
- Se asume que los archivos .csv no llevan fila de encabezado.
- Decisiones tomadas:
  - Si un ciclista no tiene actividades registradas, no se toma en cuenta en el join.
  - Si una actividad está ligada a un ciclista que no está registrado, no se toma en cuenta en el join.
  - Si una actividad está ligada a una ruta que no está registrada, no se toma en cuenta en el join.
  - A la hora de obtener los kilómetros recorridos por ciclista, por ruta, por provincia y por día, no se toman en cuenta los registros que vengan con kilómetros negativos, en null o en 0.
  - A la hora de obtener el top N de ciclistas por provincia, tanto en total de kilómetros como en promedio de kilómetros por día, si hay un empate se toma de primero el de menor cédula.
  - El promedio de kilómetros por día de cada ciclista se obtiene del total de kilómetros recorrido por el ciclista dividido entre la cantidad de días que tuvo actividad ese ciclista.