

# Tarea #1 - Big Data

## Objetivo

Introducir a los estudiantes al uso de operaciones de Apache Spark para cargar e integrar datos a través del uso de pytest.

## Resultados esperados

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

- Un programa principal que dada la información sobre ciclistas retorne el top 5 por provincia, tanto en total de kilómetros como en promedio de kilómetros por día.
- Una serie de pruebas unitarias que permitan corroborar la correctitud de las diferentes funciones internas al programa.
- Ambos deben ser entregados como un directorio comprimido donde también se encuentra la configuración de Docker para crear el contenedor y ejecutar las pruebas y el programa principal exitosamente. Los/as estudiantes serán responsables de documentar en su entregable cualquier instrucción necesaria para la correcta ejecución.

**Entrega:** Archivo comprimido con código y PDF en TEC Digital a más tardar el 8 de agosto de 2021 a las 11:00 PM

## Datos de entrada

Asumiremos que existen 3 entidades cada una con los siguientes atributos:

1. Ciclista
  - a. Cédula (numérico)
  - b. Nombre completo (string)
  - c. Provincia (San José, Alajuela, etc. Expresado como string)
2. Ruta
  - a. Código de ruta (identificador numérico)
  - b. Nombre ruta (string)
  - c. Kilómetros (numérico / decimal)
3. Actividad
  - a. Código de ruta
  - b. Cédula
  - c. Fecha (Formato YYYY-MM-DD)

Para la ejecución del programa principal, los estudiantes deberán proveer 3 archivos en formato CSV (separados por comas) con suficientes datos para ejemplificar la correcta funcionalidad.

Los archivos no deben llevar fila de encabezado y las columnas deben llevar el mismo orden mencionado en cada uno de los 3 apartados anteriores.

## Programa principal (20 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 ciclista.csv ruta.csv actividad.csv"

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

## Pruebas esperadas

Para realizar las pruebas unitarias se espera que los estudiantes piensen en las diferentes partes necesarias para conseguir el objetivo final. Éstas podrán arrancar de datos que se **encuentren en memoria**, asumiendo que el código deberá ser suficientemente modular para que el programa principal simplemente llame a funciones reutilizables que son probadas en diferentes pruebas unitarias.

**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:

- **Unión de los datos.** El primer paso debe ser unir los 3 conjuntos de datos diferentes. Deberá existir funciones que solamente se encarguen de esta parte. Nótese que la unión de los datos no necesariamente es trivial. Por ejemplo, es posible que se tenga el nombre de un ciclista pero todavía no haya hecho ninguna actividad. De manera similar, podría suceder que el mismo ciclista ejecute la misma ruta múltiples veces en un día. **(25 puntos)**
- **Agregaciones parciales.** Con el objetivo final de encontrar los/las ciclistas con mayor cantidad de actividades se pide que se realice código que permita crear dataframes intermedios donde se tiene el total de **kilómetros recorridos por persona, por ruta, por provincia y por día**. Se espera que haya pruebas que arranquen de dataframes intermedios ya contruidos (i.e., no empezarán desde tener que unirlos) y revisen la correcta agregación de los datos. **(35 puntos)**

- **Resultados finales.** Los estudiantes deberán crear pruebas que arranquen de las agregaciones ya construidas y pueden retornar el top N de ciclistas por provincia, tanto en total de kilómetros como en promedio de kilómetros por día. **(20 puntos)**

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

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.