

**Docente:** Dario Ernesto Del Carmen Correal Torres

**Estudiantes:**

- María Alejandra Vargas Torres 201123148 - [ma.vargas73@uniandes.edu.co](mailto:ma.vargas73@uniandes.edu.co)
- Juan Sebastián Alegría Zúñiga 202011282 - [j.alegria@uniandes.edu.co](mailto:j.alegria@uniandes.edu.co)
- Andrés Felipe Romero Brand 202013448 - [af.romerob1@uniandes.edu.co](mailto:af.romerob1@uniandes.edu.co)

**Parte A**

- Requerimientos:
  1. Retornar un catálogo con el número de taxis en los servicios reportados y el número total de compañías que tienen al menos un taxi inscrito.
  2. Como parámetro el usuario da un número específico de compañías que tienen al menos un taxi afiliado y retornar por cada compañía su nombre y número de taxis.
  3. Recibir como parámetro el número de compañías de las cuales se quiera saber el número de servicios prestados y retornar de mayor a menor el número de servicios con el nombre de la compañía.

**Parte B**

- Requerimientos:
  1. Se recibe como parámetro el número de taxis de los que se desea saber el número de puntos en una fecha determinada, se retorna los N taxis con más puntos.
  2. Se recibe como parámetro un rango de fechas y el número de taxis de los que se quieren saber sus puntos, retorna los M taxis con más puntos.

En este caso los 2 requerimientos se realizaron con las mismas funciones, lo que cambia son los parámetros.

Entidad/Concepto	TAD	Función	Estructura de datos
Fechas	Map Ordenado	Tener como llave la fecha en la que inicia el viaje y como	RBT: Garantiza que el árbol esté balanceado y que se

		valores la lista con la información del taxi: servicios prestados, dinero recibido, puntos y millas recorridas.	tengan tiempos logarítmicos en todas las operaciones.  <b>Complejidad:</b> <b><math>O(\log(N))</math></b>
Obtener información del árbol y ordenarla de mayor a menor	Map	Obtener la información del árbol según el id de cada taxi, después obtener los valores guardados en el árbol según el rango de fechas determinado y finalmente, comparar los puntos de cada uno organizandolos de mayor a menor.	Tabla de Hash: Permite obtener la información con una menor complejidad $O(1)$ .  <b>Complejidad:</b> <b><math>O(\log(N)) + O(1) + O(N)</math></b>

## Parte C

### - Requerimientos:

Finalmente, con miras a mejorar el servicio para los usuarios finales se propone la implementación de un servicio de consulta del mejor horario (hora:minutos) para desplazarse entre dos “Community Area”. Para tal fin se solicita resolver el siguiente problema: dados el área de origen, el área de destino y el rango de tiempo en el día en el que se quiere iniciar el viaje; generar un reporte que incluya: el mejor horario (hora:minutos) de inicio de viaje (en el área de origen) en el rango de tiempo dado para tener la menor duración posible, la ruta (secuencia de “community areas”) del viaje y el tiempo estimado (segundos) del viaje. Tenga en cuenta que, si en los datos hay múltiples viajes iniciando en la misma franja horaria de una “community area” origen a una “community area” destino, la duración de un viaje entre el origen y el destino será la duración promedio de estos viajes.

1. Consultar el mejor horario para desplazarse entre dos “Community Area”
  - Se utilizó el mismo concepto de “Fechas” explicado en el requerimiento anterior.