**Proyecto Final del Curso**

# Requerimiento 1: Reporte de información de compañías y taxis

## ¿Qué TAD utilizaron en la solución del requerimiento?

def newAnalyzer():

    analyzer={  'servicioIndex':None,

                'companias':None,

                'taxiIndex':None,

                'CompaniasConServicios':None,

                'CompaniasConTaxis':None }

    analyzer['servicioIndex']=lt.newList('SINGLE\_LINKED',compareIds)

    analyzer['companias']=om.newMap(omaptype='RBT',comparefunction=compareServicio)

    analyzer['taxiIndex']=om.newMap(omaptype='RBT',comparefunction=compareTaxi)

    analyzer['CompaniasConTaxis']=om.newMap(omaptype='RBT',comparefunction=compareTaxi)

    analyzer['CompaniasConServicios']=om.newMap(omaptype='RBT',comparefunction=compareTaxi)

    return analyzer

## ¿Por qué eligieron esa estructura de datos?

Básicamente escogimos estas estructuras las cuales son las que mas se alinean al requerimiento, toda vez que si bien los registros de entrada (23 campos para cada registro), teníamos que agruparlos en categorías, una categoría era, (a) Compañía con la cantidad de taxis afiliados y (b) Compañía con la cantidad de servicios prestados.

En este sentido, la lista que recibe los servicios prestados, fuer almacenada en una Single linked list, y fue usada cómo pivote para este requerimiento y otros requerimientos, se preservaron inicialmente los 23 campos de cada servicio.

Los demás fueron Mapas Ordenados (Tabla de símbolos ordenadas), estos mapas nos permitían almacenar de una manera eficiente las Compañías y los campos que se requerían para este requerimiento que solo fueron: Nombre de compañía (company), identificador del viaje (trip\_id) y placa de taxi (taxi\_id). Con lo anterior como no quedan duplicados las llaves, solo con el llamado del método Size, ya teníamos las cantidades respectivas de Compañías, servicios y taxis.

## ¿Cuál es la complejidad estimada del algoritmo?

En general, los métodos usados fueron sobre Mapas Ordenados (Tabla de símbolos ordenadas, om.newMap(omaptype='RBT',comparefunction=compareTaxi)), por ser Map Red Black Tree, la complejidad para las operaciones usadas fue O(log (n)), por eso los cálculos se ven casi de inmediato.

# Requerimiento 2: SISTEMA DE PUNTOS Y PREMIOS A TAXIS

## ¿Qué TAD utilizaron en la solución del requerimiento?

## ¿Por qué eligieron esa estructura de datos?

## ¿Cuál es la complejidad estimada del algoritmo?

# Requerimiento 3: CONSULTA DEL MEJOR HORARIO EN TAXI ENTRE 2 “COMMUNITY AREAS”

## ¿Qué TAD utilizaron en la solución del requerimiento?

## ¿Por qué eligieron esa estructura de datos?

## ¿Cuál es la complejidad estimada del algoritmo?