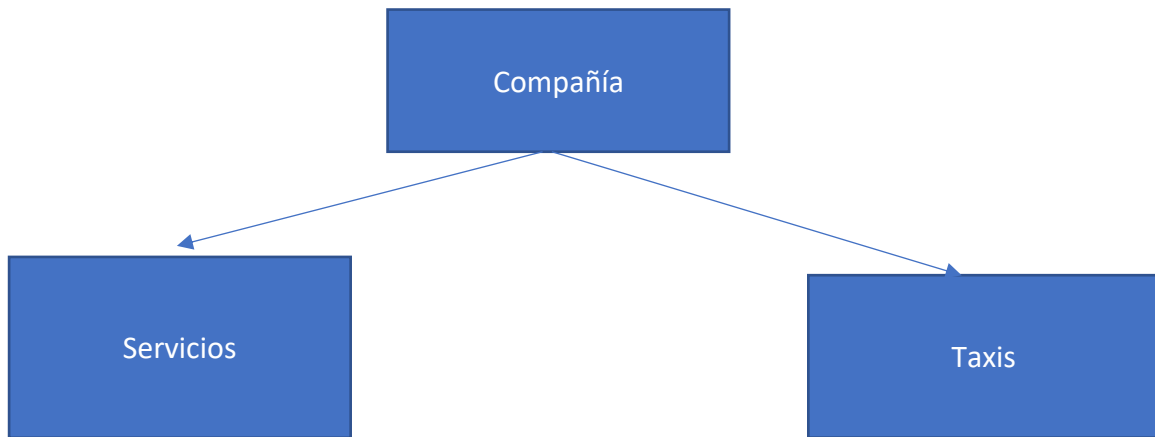


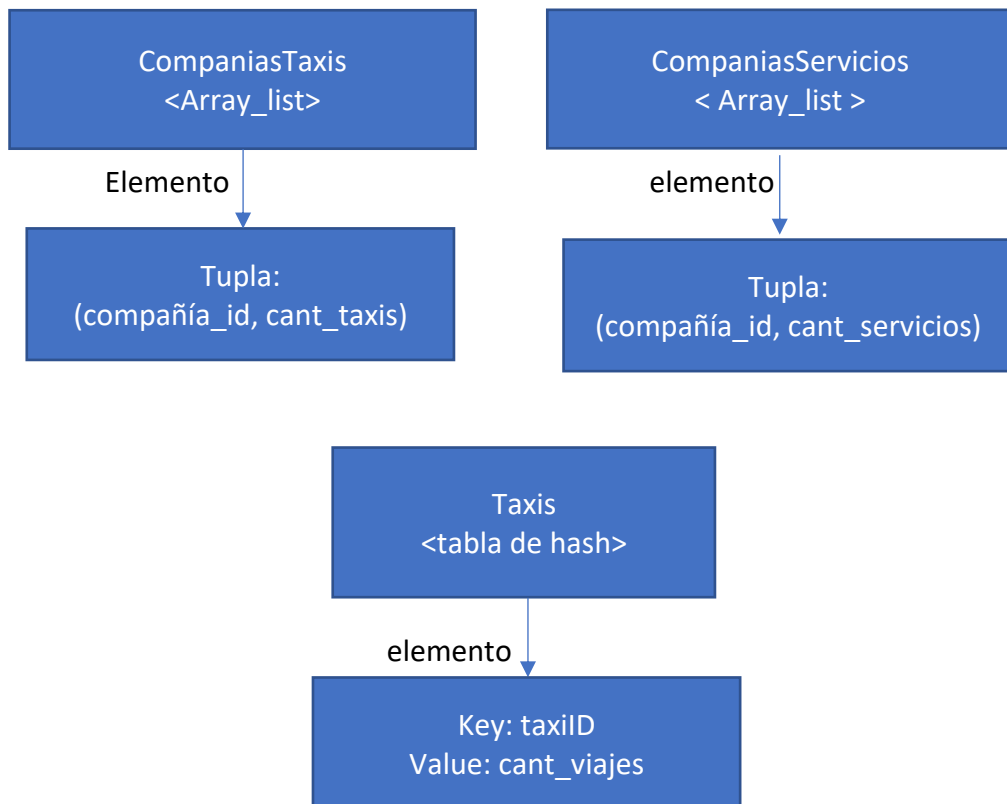
REQUERIMIENTO 1:

Sustantivos: taxis, compañías, servicios

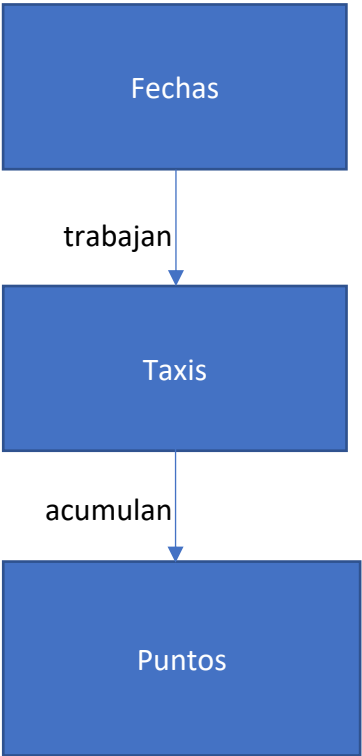


Entidad / concepto	TAD	Propósito	Estructura de datos
Compañías ordenadas por cant de taxis	Lista	Ordenar las compañías según la cantidad de taxis afiliados a cada una de ellas.	Array list
Compañías ordenadas por cant de servicios	Lista	Ordenar las compañías según la cantidad de servicios que prestaron cada una de ellas.	Array list
Taxis	Tabla de Hash	Contar los taxis que hay en la carga de datos.	linear probing

Entidad / concepto	TAD	Estructura de datos	Función	Complejidad temporal / espacial
Compañías ordenadas por cant de taxis	Lista	Array list	getTopCompaniasTaxis (cola, M) getCantCompanias(mapa)	$O(1) / O(N)$ $O(1) / O(N)$
Compañías ordenadas por cant de servicios	Lista	Array list	getTopCompaniasServicios (cola, M)	$O(1) / O(N)$
Taxis	Tabla de Hash	linear probing	getCantTaxis(mapa)	$O(1) / O(N)$

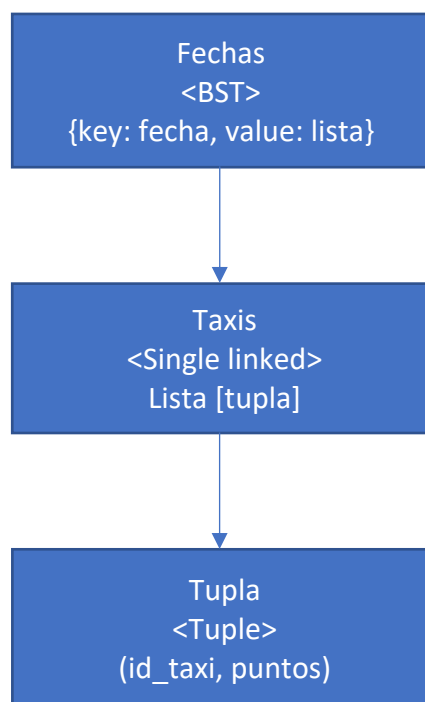


REQUERIMIENTO 2:



Entidad / concepto	TAD	Propósito	Estructura de datos
Un árbol de fechas	Árbol	Encontrar un rango de fechas fácilmente al tener un árbol por fechas	Bst
Una lista ordenada que es el valor de una fecha del árbol.	Lista	Encontrar los N taxis con mas puntos en una fecha	Single linked

Entidad / concepto	TAD	Estructura de datos	Función	Complejidad temporal / espacial
Un árbol de fechas	Árbol	Bst		
Una lista ordenada que es el valor de una fecha del árbol.	Lista	Single linked	getTopTaxisFecha(árbol, fecha, cant) getTopTaxisRange(árbol, fecha_inicio, fecha_final, cant)	O(log N) O(log N) O(log N) O(log N)

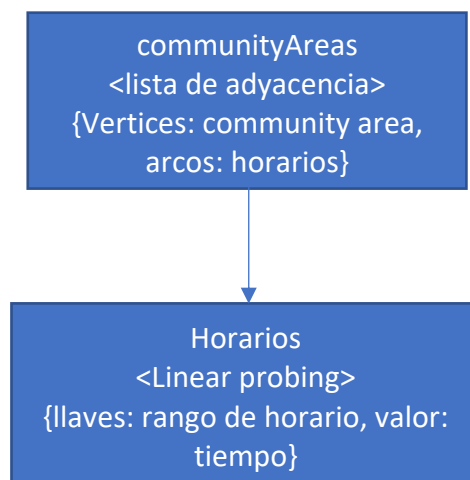


REQUERIMIENTO 3:



Entidad / concepto	TAD	Propósito	Estructura de datos
Un grafo con vértices de community areas	Grafos	Conectar las community areas siendo el peso de los arcos el tiempo entre ellas en un determinado horario	Lista de adyacencia
Una tabla de hash con el tiempo entre dos community areas en un determinado horario	Tabla de hash (el peso de los arcos del grafo)	Tener los diferentes tiempos en el días	Linear probing

Entidad / concepto	TAD	Estructura de datos	Función	Complejidad temporal / espacial
Un grafo con vértices de community areas	Grafos	Lista de adyacencia	ruta(grafo, vertice_inicial, vertice_final)	$O(v+e)$ $O(v+e)$
Una tabla de hash con el tiempo entre dos community areas en un determinado horario	Tabla de hash	Linear probing	getHorarioInRange(grafo, rango)	$O(1)$ $O(N)$



Requerimiento 1:

TAD: Lista y Tabla de Hash

Razón: Lista: Para poder ordenar las compañías según el número de taxis y servicios de cada una. Tabla: Para poder realizar el conteo de la cantidad total de taxis en menor tiempo.

Complejidad: $O(1)$

Requerimiento 2:

TAD: Árbol

Razón: Para poder tener las fechas ordenadas y que se nos facilite la búsqueda de las fechas en el rango dado.

Complejidad: $O(\log n)$

Requerimiento 3:

TAD: Grafo y Tabla de Hash

Razón: Grafo: Para poder conectar las Community Areas entre ellas. Tabla: Para poder tener los diferentes tiempos entre las Community Areas según los diferentes rangos de horario.

Complejidad: $O(V+E)$