REQUERIMIENTO 3

El requerimiento 3 tiene una complejidad de O(n+k) debido a que este realiza un recorrido a lo largo del número total de taxis en el catálogo y, posteriormente, otro recorrido sobre los taxis que cumplen el filtro. Cabe aclarar que ambos ciclos de recorrido no son anidados, por lo que no se desarrolla una complejidad exponencial.

REQUERIMIENTO 4

El requerimiento 4 tiene una complejidad de O(n+k*z+c) debido a que en primer lugar, realiza una búsqueda lineal O(n). Posterior a eso, realiza una búsqueda para cada taxi que ha superado el filtro (k) en relación a su origen y destino, por lo que se tiene que aplicar un ciclo que aplica la ecuación de Haversine (z), lo cual es un ciclo anidado. Esto da como producto una complejidad de k*2z (ya que la ecuación se aplica 2 veces debido a que se tienen que identificar dos ubicaciones), lo que se traduce en un O(k*z). Por último, se realiza la evaluación de las combinaciones en términos de promedio. Luego se identifica la combinación con los costos extremos, pero sin anidar ningún ciclo entre procesos (2n). EN conclusión, sería una complejidad de O(n+k*z+c).

REQUERIMIENTO 5

Realmente me quedo muy pesado :(

REQUERIMIENTO 6

El requerimiento 6 tiene una complejidad de O(n*b+m*b) ya que el algoritmo ejecuta dos fases principales: primero filtra los n taxis por fecha y barrio de origen, requiriendo hasta n llamadas a identificar barrio (cada una O(b)). Segundo procesa los taxis filtrados (m), identificando barrios de destino con m llamadas adicionales a la misma función. Las operaciones restantes (conteo de métodos de pago, cálculo de promedios) son O(p + d), por lo que no representan mucho en la definición final.