

**Proyecto 2 Estructuras de Datos**  
**Documento de diseño**  
**Santiago Talero / Daniel S Londoño**  
**(201821994 / 201821363)**

Listado de requerimientos funcionales:

- **1A:** Obtener las N letras más frecuentes por las que comienza el nombre de una zona (No diferenciar las mayúsculas de las minúsculas).
- **Datos de entrada:** N
- **Datos de salida:** Para cada letra se debe imprimir la letra y el nombre de las zonas que comienzan por esa letra. (De mayor a menor)
- **Complejidad estimada:**  $O(1)$
  
- **2A:** Buscar los nodos que delimitan las zonas por Localización Geográfica.
- **Datos de entrada:** Latitud, longitud.
- **Datos de salida:** Mostrar todos los nodos en la frontera de las zonas que tengan la misma latitud y longitud truncando a las primeras 3 cifras decimales.
- **Complejidad estimada:**  $O(N\log(N))$
  
- **3A:** Buscar los tiempos promedio de viaje que están en un rango y que son del primer trimestre del 2018. Por cada viaje se debe mostrar su zona de origen, zona de destino, mes y tiempo promedio mensual del viaje.
- **Datos de entrada:** Rango (limite bajo, limite alto)
- **Datos de salida:** Viajes cuyo tiempo promedio mensual esté en ese rango. Por cada viaje se muestra su zona de origen, zona de destino, mes y tiempo promedio mensual.
- **Complejidad estimada:**  $O(N)$
  
- **1B:** Buscar los N zonas que están más al norte.
- **Datos de entrada:** N

- **Datos de salida:** Por cada zona se imprime nombre, latitud y longitud de su punto más al norte. Los resultados se imprimen desde las que estén más al norte.
  - **Complejidad estimada:**  $O(1)$
- 
- **2B:** Buscar nodos de la malla vial por Localización Geográfica
  - **Datos de entrada:** Latitud, longitud.
  - **Datos de salida:** Todos los nodos que tengan esas mismas latitud y longitud (truncando a 2 cifras decimales).
  - **Complejidad estimada:**  $O(N\log(N))$
- 
- **3B:** Buscar los tiempos de espera que tienen una desviación estándar en un rango dado y que son del primer trimestre del 2018.
  - **Datos de entrada:** Rango (limite bajo, limite alto)
  - **Datos de salida:** Viajes cuya desviación estándar mensual este en ese rango.
  - **Complejidad estimada:**  $O(N)$
- 
- **1C:** Retornar todos los tiempos de viaje promedio que salen de una zona dada y a una hora dada.
  - **Datos de entrada:** Zona (ID), hora.
  - **Datos de salida:** Zona de origen, zona de destino, hora y tiempo promedio de cada viaje.
  - **Complejidad estimada:**  $O(N)$
  - **Justificación estructura utilizada(Heap):** *Usamos Heap ya que partiendo de los parámetros (Zona ID y hora), son variables que se pueden acceder globalmente y así optimizar la memoria usada.*
- 
- **2C:** Retornar todos los tiempos de viaje que llegan de una zona dada y en un rango de horas.
  - **Datos de entrada:** Zona (ID), Rango de horas.
  - **Datos de salida:** Zona de origen, zona de destino, hora y tiempo promedio de cada viaje.
  - **Complejidad estimada:**  $O(N)$

- **Justificación estructura utilizada(Heap):** *En este caso usamos Heap ya que partiendo de los parámetros (Zona ID y el rango de horas), son variables que se pueden acceder globalmente y en conjunto con el Garbage Collection, quien corre en la memoria del Heap para liberar memoria usada por el objeto, hacen que la **memoria sea mejor optimizada**.*
- **3C:** Obtener las N zonas priorizadas por la mayor cantidad de nodos que definen su frontera.
- **Datos de entrada:** N
- **Datos de salida:** Nombre de la zona y el número de nodos que definen su frontera (Por cada zona).
- **Complejidad estimada:**  $O(1)$
- **Justificación estructura utilizada(Hash Table):** *Usamos Tablas de Hash ya que su complejidad evidencia que para grandes volúmenes de datos (como es este caso), extraer objetos no consume mucho tiempo.*
- **4C:** Crear una gráfica (usando formato ASCII) que muestre por cada zona de origen que porcentaje de datos faltan.
- **Datos de entrada:** N/A
- **Datos de salida:** Gráfica ASCII que por cada zona de origen muestre que porcentaje de datos faltan.
- **Complejidad estimada:**  $O(1)$