

EXPLICACIÓN DE LA ELECCIÓN DE LAS ESTRUCTURAS DE DATOS

Francisco Pertíñez Perea.

TDA PUNTO: en este caso he optado por utilizar dos atributos de tipo double, latitud y longitud, la razón de esto es que en este problema la representación de un punto se daba en el txt de almacen_rutas.txt como la coordenada del punto como latitud y longitud, razón de esto es que en la implementación se haya puesto las condiciones de:

$$-90 \leq \text{latitud} \leq 90 \text{ y } -180 \leq \text{longitud} \leq 180.$$

TDA RUTA: en este caso he optado por representar el TDA con dos atributos: un string, que guardara el código de la ruta, y una lista donde se almacenaran los puntos de dicha ruta. En cuanto a la elección de la lista en un primer momento mis dos opciones era usar una lista o un vector (se debe guardar el orden de los puntos por los que pasa la ruta y estas estructuras lo representan muy bien), finalmente opte por la lista debido a que la veo más flexible que el vector al ser más eficiente en inserciones y borrados en medio de la estructura, cosa que es necesaria pues si se quiere ampliar una ruta se debería de poder colocar en cualquier lugar el nuevo destino y no siempre al principio o al final, además de que también se puede tener que borrar cualquier destino de una ruta y no el primero o el último (aunque esto también lo permite el vector, es mucho más ineficiente para estas operaciones pues no se creo con ese cometido).

TDA ALMACEN RUTAS: en este caso he optado por representar el TDA con un solo atributo: un map con clave string que representa el código de la ruta, y como valor la ruta con dicho código, la razón de esto es que en este TDA lo que prima es buscar y borrar a partir del código de la ruta, además de añadir nuevas rutas lo más eficiente posible, y en estos aspectos el map se desempeña muy bien;