



Muy importante

- Todos los métodos deben ser **eficientes**, es decir, recorrer de forma eficiente las estructuras utilizadas.
- Escriba código claro: indente y modularice apropiadamente.
- Utilice nombres de variables, métodos y clases respetando *buenas pautas de programación*
- La comprensión de los enunciados forma parte de la evaluación

Se desea implementar un **sistema de información para el viajero automovilista**.

Para administrar el sistema es necesario contar con información de todas las ciudades en una Tabla de Búsqueda. De cada ciudad se almacena la siguiente información: nombre, provincia, cantidad de habitantes y si hay alojamiento disponible o no.

Como requerimiento de implementación, se pide representar la Tabla de Búsqueda mediante una estructura de árbol AVL, donde el orden de las ciudades está dado por su nombre (identificador unívoco).

Además, es necesario guardar los datos de todas las rutas que unen cada par de ciudades, indicando su distancia en kilómetros. Ejemplo, entre las ciudades de Neuquén y Cipolletti hay un tramo de ruta que las une de 8 km de distancia.

Realizar una clase ServicioViajero que contenga un menú que permita las siguientes tareas:

1. Alta y baja de una ciudad (recuerde que cada ciudad está almacenada en la tabla de búsqueda y también en el grafo).
2. Alta y baja de un tramo de ruta (arco entre dos ciudades).
3. Dado su nombre, mostrar todos los atributos de dicha ciudad.
4. Dados 2 nombres de ciudades, mostrar todas las ciudades cuyo nombre se encuentre en dicho rango de la manera más eficiente.
5. Dadas 2 ciudades A y B, devolver la menor cantidad de kilómetros a realizar para llegar desde A hasta B.
6. Dadas 2 ciudades A y B y una cantidad de kilómetros K, verificar si existe algún camino para ir desde A hasta B que no supere K kilómetros en total.
7. (*) Dadas 2 ciudades A y B, devolver el camino que pasa por menos ciudades que va desde A hasta B. El camino que debe devolver será una lista de ciudades.
8. (*) Dadas 2 ciudades A y B, devolver un camino desde A hasta B que pase sólo por ciudades que tienen alojamiento disponible (excepto A y B).
9. Mostrar todas las ciudades ordenadas alfabéticamente.
10. Ejecutar un “script de carga” de al menos 15 ciudades y con al menos 25 tramos de ruta. Al momento de presentar el trabajo a los docentes, traer en papel un diagrama de los datos cargados.
11. Para fines prácticos de la corrección y verificación de las estructuras, es necesario contar con opciones del menú que permitan mostrar la información almacenada en cada estructura utilizada, de manera que pueda verse quién está conectado con quién (si es un grafo ver sus adyacentes o si es un árbol ver quienes son hijos de quién).

(*) Incisos 7 y 8 sólo para alumnos que no promocionan la materia.

Forma de entrega:

- El trabajo deberá ser realizado de forma individual
- Armar un proyecto dividido adecuadamente en varios paquetes
- Se deberá crear un archivo ZIP o RAR con todos los archivos correspondientes al código fuente (.java), y llamarlo con el Apellido_Nombre correspondiente.
- El archivo comprimido deberá subirse a la Plataforma de Educación a Distancia de la Universidad (<http://pedco.uncoma.edu.ar>) en el curso “**Estructuras de datos (para alumnos preparando el final)**” en la tarea Trabajo Práctico Final. El link habilitado para subir la tarea no permitirá reentregas, por lo que se recomienda verificar el archivo antes de ingresarlo a la plataforma.

Nota:

- El trabajo práctico es de carácter obligatorio y condición necesaria para rendir el examen final, por lo que, deberá ser presentado 15 días antes de la mesa de examen en la cual se inscriban para rendir.