

# Trabajo Final de Estructuras de Datos – FAI - UNCOMA Cursado 2023

## Sistema de Mudanzas Compartidas

Se desea desarrollar un sistema para una empresa que realiza mudanzas compartidas entre localidades de Argentina.

En una mudanza compartida, cada grupo de muebles, cajas y demás objetos a transportar, viaja separada y sectorizada pero en un mismo camión, pagando cada cliente solamente por los metros cúbicos que ocupa, lo cual representa una fracción del precio de una mudanza habitual.



El principal objetivo del sistema será permitir consultar sobre qué camino tomar para cubrir viajes entre una ciudad de origen A y una ciudad destino B de forma eficiente. Para analizar la eficiencia del camino también se tendrán en cuenta la capacidad en m<sup>3</sup> que tiene un camión de transporte y los pedidos que haya entre las ciudades.

La **información de las ciudades** se almacenará en un TDA Diccionario/Tabla de Búsqueda implementado con un **Árbol AVL**. La clave será el código postal (conformada por 4 dígitos), y también se almacenará su nombre (ej: Cutral Cód) y el de la provincia (ej: Neuquén). El TDA implementado deberá almacenar claves de tipo Comparable y dato de tipo Object.

Además, para cada ciudad se desean almacenar los **solicitudes de viajes** que deben salir de dicha ciudad, indicando para cada solicitud la ciudad destino, la fecha de solicitud, el cliente, la cantidad de metros cúbicos, la cantidad de bultos del envío, domicilios de retiro y entrega, y si el envío está pago (Si/No). Para implementar este requisito **deberá elegir el propio estudiante una estructura** que sea adecuada para facilitar la búsqueda de todas las solicitudes entre dos ciudades

El **mapa de rutas** estará conformado por un conjunto de ciudades unidas entre sí por rutas que permiten la circulación en ambos sentidos. Para cada ruta se debe almacenar la información de la ruta es su distancia en kilómetros. Para implementarlo se utilizará un grafo etiquetado. El **TDA grafo** deberá estar implementado para vértices de tipo Object y etiquetas de tipo numérico (double).

Por otro lado se utilizará un Mapeo (implementado con la estrategia de **Hash abierto** o bien usando la clase **HashMap** de Java) para almacenar **información de los clientes**. De cada cliente se almacenará su clave (tipo y número de documento) y también nombre/s, apellido/s, teléfono e email.

Se pide el desarrollo de la clase **MudanzasCompartidas** con un menú de opciones para realizar las siguientes tareas:

1. *Carga inicial del sistema*: ingresar al sistema un lote fijo de ciudades, rutas, clientes y pedidos.
2. *ABM<sup>1</sup> de ciudades*
3. *ABM de la red de rutas*

---

<sup>1</sup> ABM significa ALTAS-BAJAS-MODIFICACIONES. Para modificaciones considere sólo los datos mutables: no permita modificar claves. Por ejemplo, no debería cambiarse el código postal de una ciudad ni el tipo o número de DNI de un cliente.

4. *ABM de clientes*
5. *ABM de pedidos*
6. *Consulta sobre clientes:*
  - Dada una clave de cliente, mostrar toda la información del mismo
7. *Consultas sobre ciudades:*
  - Dado un código postal de una ciudad, mostrar toda su información
  - Dado un prefijo, devolver todas las ciudades cuyo código postal comienza con dicho prefijo. Por ejemplo si el prefijo es "83" debería considerar listar todas las ciudades cuyo código postal esté en el rango 8300 hasta 8399
8. *Consultas sobre viajes:* Dada una ciudad A y una ciudad B:
  - Obtener el camino que llegue de A a B que pase por menos ciudades
  - Obtener el camino que llegue de A a B de menor distancia en kilómetros
  - (\*) Obtener todos los caminos posibles para llegar de A a B que pasen por una ciudad C dada sin pasar dos veces por la misma ciudad
  - (\*) Verificar si es posible llegar de A a B recorriendo como máximo una cantidad X de kilómetros
9. *Verificar viaje:*
  - Dada una ciudad A y una ciudad B mostrar todos los pedidos y calcular cuánto espacio total hace falta en el camión.
  - (\*) Dada una ciudad A y una ciudad B y una cantidad en metros cúbicos (espacio en un camión), verificar si sobra espacio en el camión y hacer un listado de posibles solicitudes a ciudades intermedias que se podrían aprovechar a cubrir, considerando el camino más corto en kilómetros.
  - Dada una lista de ciudades y una cantidad de metros cúbicos que corresponden a la capacidad del camión, verificar si es un "camino perfecto". Un camino perfecto es un camino que existe en el grafo y que hay por lo menos una solicitud que se pueda transportar entre las ciudades por las cuales pasará el camión. Por ejemplo, si la lista de ciudades es [A, B, C, D], siendo este un camino posible, el camino será perfecto si además de existir el camino indicado, hay por lo menos un pedido que vaya de la ciudad A a cualquiera de las ciudades B, C o D; por lo menos un pedido desde la ciudad B hacia C o D y por lo menos un pedido desde C a D. Tener en cuenta que la capacidad del camión debe ser suficiente para cubrir los traslados a lo largo de todo el camino.
10. *Mostrar sistema:* es una operación que permite ver todas las estructuras utilizadas con su contenido (grafo, AVL, Mapeo, etc.) para verificar que se encuentran bien cargadas. No deben ser listados, sino debe poder verse la estructura tal cual está cargada. Por ejemplo, en el caso del árbol AVL debe ser visible la altura de cada nodo y quienes son los nodos hijo izquierdo y derecho de cada nodo.

#### Requisitos:

- El programa debe permitir la ejecución por separado de cada una de las operaciones especificadas.
- El programa **debe ser eficiente**: Debe recorrer las estructuras sólo lo necesario y haciendo buen uso de la memoria.
- Recordar **modularizar y comentar apropiadamente** apuntando a un código reusable y fácilmente mantenible.
- Las estructuras deben estar implementadas de forma genérica para elementos de tipo Object o Comparable de Java, según la necesidad del TDA.
- La **carga inicial** del sistema debe hacerse a partir de un archivo de texto con formato preestablecido y desde las opciones del menú se deben poder cargar datos adicionales. Tanto la carga inicial como manual deberá realizar el control de la consistencia de los datos (formato, existencia previa, etc)
- **Utilizar un archivo de log** (archivo de texto) para guardar la siguiente información: (1) estado del sistema (contenido de todas las estructuras) al terminar la carga inicial, (2) a continuación anotar qué operaciones de ABM se realizan a lo largo de la ejecución

(Ej: “Se creó la ciudad X”, “Se borró el tramo de ruta entre las ciudades X e Y”, etc), y (3) estado del sistema al momento de terminar de ejecutarse.

### Condiciones y fechas de entrega:

- El trabajo debe realizarse de manera **individual** y debe presentarse **personalmente** a los docentes antes de subirlo a PEDCO en el curso “EDAT (preparando final)”.
- Al momento de la defensa, se deberá presentar un dibujo (en papel o digital) del mapa cargado (grafo) y de los diccionarios implementados con AVL
- Los **estudiantes que promocionan** la materia tendrán tiempo para entregarlo hasta el **lunes 7 de agosto de 2023** y no necesitan realizar los módulos marcados con (\*)
- Los estudiantes que no promocionan podrán entregarlo en cualquier momento, pero **como mínimo deberán hacerlo 2 semanas antes** de presentarse a rendir el final regular.

### PARA INVESTIGAR COMO USAR EN JAVA

- Clase StringTokenizer (para fraccionar un String con un caracter separador)
- Para abrir archivo de texto para lectura: FileReader, BufferedReader o similares
- Para escribir en archivo de texto: FileWriter o similares

### FORMATO DE ARCHIVO DE TEXTO (EJEMPLOS)

**Ciudades:** Clave única código postal (4 dígitos), nombre de la ciudad y de la provincia

- C;5000;Córdoba;Córdoba
- C;8300;Neuquén;Neuquén
- C;8324;Cipolletti;Río Negro

**Solicitudes de viaje:** ciudad origen, ciudad destino, fecha de solicitud, identificador del cliente, cantidad de metros cúbicos, cantidad de bultos, domicilio de retiro y domicilio de entrega, y si el envío está pago (Si/No)

- S;5000;8300;15/06/2023;DNI;35678965;13;5;Sarmiento 3400;Roca 2100;T
- S;8324;5000;14/06/2023;PAS;21923847;8;2;Belgrano 134;San Martín 752;F

**Mapa de Rutas:** código postal de las ciudades y distancia en kilómetros entre ellas

- R;5000;8324;1108.5
- R;8324;8300;11.0

**Información clientes.** De cada cliente se almacenará su clave (tipo y número de documento) y también nombre/s, apellido/s, teléfono e email.

- P;DNI;35678965;FERNANDEZ;JUAN CARLOS;299-4495117
- P;PAS;21923847;MARTINEZ;GRACIELA;351-4348260

PARA LA ENTREGA LA CARGA INICIAL DEBE CONTAR **AL MENOS** CON 30 ELEMENTOS DE TIPO CIUDAD, 20 SOLICITUDES, 20 CLIENTES Y 40 RUTAS EN EL MAPA.

ASEGURAR QUE EN EL SET DE CARGA INICIAL LOS ELEMENTOS SE LISTEN EN FORMA DESORDENADA PARA QUE SE PRODUZCAN **TODAS LAS ROTACIONES POSIBLES** AL INSERTAR EN AVL.