



Departamento de Programación
Facultad de Informática
Universidad Nacional del Comahue



Estructuras de Datos Trabajo Práctico Final

Mundial Rusia 2018



Se desea desarrollar un sistema que brinde información del Mundial de Fútbol 2018 a disputarse en Rusia. El principal objetivo del sistema es brindar a los simpatizantes información sobre el evento y sugerir caminos para viajar de una sede a otra a ver a su equipo favorito.

Dado que los partidos se llevarán a cabo en distintas ciudades, se desea que el sistema almacene un mapa de las ciudades de Rusia y rutas terrestres entre ellas. Este mapa de ciudades deberá incluir las sedes de los partidos y otras ciudades en las que los medios de transporte puedan hacer escala. Para cada ciudad se almacenará un nombre único, su superficie aproximada, cantidad de habitantes y si es sede o no del mundial. Para las rutas se almacenará la distancia en kilómetros entre cada par de ciudades.

Además, se guardará información de las selecciones de cada país, incluyendo el nombre del director técnico, los equipos contra los cuales jugó, los puntos ganados y la cantidad de goles a favor y en contra hasta el momento.

Por otro lado, se almacenarán los resultados de los partidos, indicando los dos equipos participantes (equipo1 y equipo2, considerando el orden alfabético de sus nombres, equipo1<equipo2), ronda por la que se enfrentan (grupo, octavos, cuartos, semifinales, final), ciudad del evento, y la cantidad de goles de cada equipo. Se considera que dos equipos no pueden volver a enfrentarse.

Para la implementación se deberá utilizar un grafo etiquetado para el mapa de las ciudades. Los equipos de fútbol se guardarán en una Tabla de Búsqueda implementada con un árbol AVL, ordenada alfabéticamente por nombre del país. Los equipos contra los cuales compite, serán almacenados en una lista dinámica que guarde la información de cada país. Además se utilizará un Mapeo implementado con hash abierto o HashMap de Java, para almacenar los partidos, con dominio <eq1, eq2> (donde nombre eq1 menor alfabéticamente que eq2), y como rango los datos de los goles de eq1, goles de eq2, y la ronda y ciudad en la cual jugaron.

Desarrollar una clase Mundial2018 con un menú de opciones para realizar las siguientes tareas:

1. ABM (Altas-Bajas-Modificaciones) de ciudades
2. ABM (Altas-Bajas-Modificaciones) de equipos
3. Altas de partidos
4. Consulta sobre equipos:
 - Dado un país, mostrar toda la información disponible, incluyendo puntos ganados, goles a favor y en contra, diferencia de goles y resultado de todos los partidos jugados.
 - Dadas dos cadenas (min y max) devolver todos los equipos cuyo nombre está en el rango [min, max]. Los valores de min y max pueden no estar en el árbol.
 - (*) Listar todos los equipos con diferencia de gol negativa.
5. Consultas sobre ciudades:
 - Dado un nombre de ciudad, mostrar toda su información
6. Consultas sobre viajes: Dada una ciudad A y una ciudad B:
 - Obtener el camino que llegue de A a B de menor distancia en km
 - Obtener el camino que llegue de A a B pasando por la mínima cantidad de ciudades
 - Obtener todos los caminos posibles para llegar de A a B
 - (*) El camino más corto para llegar de A a B que pase por otra ciudad C

7. Obtener y mostrar la tabla de posiciones de los equipos de un momento dado, almacenando los datos de los equipos ordenados de mayor a menor puntaje (puede utilizar alguna estructura de datos auxiliar que considere apropiada, asegurando la eficiencia)
8. Mostrar sistema: es una operación de debugging que permite ver todas las estructuras utilizadas con su contenido (grafo, AVL y Mapeo) para verificar, en cualquier momento de la ejecución del sistema, que se encuentren cargadas correctamente.

Requisitos importantes:

- 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.
- Las estructuras deben estar implementadas de forma genérica para elementos de tipo Object o Comparable de Java, según el propósito de la estructura.
- La **carga inicial** del sistema debe hacerse a partir de un archivo de texto con el formato indicado. Ejemplo, la información se puede ingresar de a una por línea indicando el tipo (E:Equipo, P:Partido, C:Ciudad, R:Ruta), o en archivos diferentes
E: ARGENTINA; SAMPAOLI; D; 0; 0; 0
P: ARGENTINA; ISLANDIA; GRUPO; 0; 0
C: MOSCU; 2511; 12500123; TRUE
C: TOLYATTI; 314,8; 719514; FALSE
R: MOSCU; TOLYATTI; 989
- Guardar **un archivo de LOG** (archivo de texto) para guardar la siguiente información:
 - Estado del sistema al momento de terminar la carga inicial (todas las estructuras)
 - Listado de operaciones de ABM que se realizan a lo largo de la ejecución (Ej: "Se crea la ciudad X", "Se eliminó el equipo Y", "Se cargaron los datos del partido eq1-eq2", etc)
 - Estado del sistema al momento de terminar de ejecutarse (todas las estructuras incluyendo la tabla de posiciones)

Condiciones y fechas de entrega:

- El programa debe realizarse de manera individual y debe presentarse personalmente a alguno de los docentes que indicarán si está aprobado. Luego deberá subirlo a la plataforma PEDCO para su archivo.
- Al momento de la defensa, deberá presentar un dibujo en papel del mapa de ciudades (grafo) y la tabla de equipos (AVL) subido en la carga inicial desde archivo de texto.
- Los estudiantes que promocionan la materia tendrán tiempo para entregarlo hasta el viernes 3 de agosto pero NO DEBEN realizar los módulos marcados con (*)
- Los estudiantes que no promocionan podrán entregarlo en cualquier momento, como mínimo 2 semanas antes de presentarse a rendir el final regular.