



Universidad Católica del Norte
Facultad de Ingeniería y Ciencias Geológicas
Departamento de Ingeniería de Sistemas y Computación

ESTRUCTURA DE DATOS

Profesor: Rubén Sánchez.
Ayudantes: Vicente Pérez,
Ignacia Rivas, Diego Duarte.

Taller N°3: C++ (I-2021)

Periodo: 07 de junio al 03 de julio del 2021

I. Objetivo

Implementar distintos tipos de árboles binarios en C++ y entender sus diferencias.

II. Enunciado

Después de un año bastante largo con la pandemia del coronavirus por fin llegaron las vacunas para defenderse ante el C-19. Por lo cual, la gente puede ir a diferentes puestos de vacunación para inyectarse la vacuna.

Lamentablemente, el trabajo para ver a quién vacunar primero es muy desorganizado, y todos luchan por conseguir las vacunas lo más rápido posible. Ante esta situación, el ministerio de salud decide actuar y ponerse en contacto con estudiantes del curso de estructura de datos para hacer algo al respecto. Estos deciden en crear un programa de consola en C++ que permita administrar la espera y entrega de vacunas.

Detalle

El ministerio de salud determina que las vacunas debiesen administrarse primero a las personas con **más edad**. Además, quieren ver un listado en cualquier momento de las personas que ya fueron vacunadas y las que todavía no.

Así, se decide implementar dos árboles: un **Min Heap**, y un **AVL**, donde la clave a utilizar en ambos para las búsquedas, ingresos, eliminaciones, etc. es la **edad** de la persona. Su Min Heap solamente almacenará aquellos que aún no se vacunan, mientras que su AVL los contendrá a todos: vacunados y sin vacunar.

Al iniciar el programa se lee el archivo "PuestoVacunacion_inicial.txt" que contiene el listado de las personas que actualmente están en el puesto de vacunación. Los campos están separados por coma, y consideran:

Nombre, RUT, edad, vacunado (S: sí | N: no), Enfermedad Crónica (S: sí | N: no)

Diego Duarte, 24, 11.111.111-1, S, N Ignacia Rivas, 20, 22.222.222-2, N, S Vicente Pérez, 21,33.333.333-3 S, N Rubén Sánchez, 34, 44.444.444-1, N, N

Una vez cargados los datos del archivo, pueden ir llegando nuevas vacunas y/o personas que se quieran vacunar.

- Al llegar nuevas vacunas, se deben eliminar del Min Heap las personas a quienes se les deban administrar.
- Al llegar nuevas personas a vacunarse, se deben ingresar en el AVL y, en caso de no venir previamente vacunados, también en el Min Heap.

Al terminar el programa se debe crear el archivo de salida "PuestoVacunacion_final.txt" con los mismos campos que el archivo de carga inicial, pero esta vez con la información del estado final de los refugiados antiguos y nuevos. Este archivo de salida debe quedar ordenado de menor a mayor por edad.

Nota: No habrá edades repetidas (ni en el archivo de lectura ni en las nuevas personas que se vayan a vacunar).

Su sistema debe funcionar por consola, a través del siguiente menú:

1. Nuevas Personas Para Vacunar
2. Nuevas Vacunas
3. Estadísticas
4. Salir

Las acciones para tomar en cada opción se detallan a continuación:

1. **Nuevas Personas para Vacunar:** el sistema debe preguntar cuántas personas vienen llegando, y luego solicitar nombre, RUT, edad y estado de vacunación de cada una de ellas.
2. **Nuevas vacunas:** el usuario indica cuántas vacunas nuevas han llegado, y el sistema informa por pantalla a quiénes se les debe administrar.
3. **Estadísticas:** a través de un submenú el sistema informa:
 - 3.1. Cuántas personas están vacunadas y cuántas no lo están.
 - 3.2. Quién es la persona más antigua en el puesto de vacunación que no está vacunada.
 - 3.3. Cuál es la edad promedio de las personas en el puesto de vacunación.
 - 3.4. Si un RUT específico (entregado por pantalla) está o no vacunado.
 - 3.5. Listado completo de todas las personas en el refugio con toda su información.
 - 3.6. Cuántas vacunas sobran.
4. **Salir:** genera el archivo de salida, elimina ambos árboles y se cierra el programa.

Estructura de Datos

Árbol Min Heap para los refugiados sin vacunar, y árbol AVL para contener a todos los refugiados.

Informe

Debe incluir un análisis de las ventajas de ocupar Min Heap y AVL, dados los requerimientos planteados en el enunciado.

III. Entrega

Fecha de entrega: 03 de julio hasta media noche (23:59).

Se deben subir los archivos a la Plataforma Campus Virtual UCN, en la sección de **Trabajos > Taller 3**. Sólo en caso de que la plataforma presente problemas, se deben enviar los documentos al email diego.duarte@alumnos.ucn.cl que será el encargado de revisar su último taller.

Incluir:

- 1.- Programa en C++. Proyecto en Visual Studio 2017 o 2019.
- 2.- Informe explicando la solución del problema y cómo usar el programa. Incluir diagrama de clases. Al final se debe detallar el total de horas que cada estudiante dedicó a la realización del taller con el desglose correspondiente (tiempo para diseñar, codificar, escribir el informe, etc.).

POR SEGURIDAD: se recomienda que los estudiantes dispongan de una copia del taller.

EL TALLER SE PUEDE REALIZAR DE FORMA INDIVIDUAL O EN PAREJAS

PAUTA PARA LA EVALUACION DEL TALLER 3

El trabajo se divide en 3 partes:

- Diseño 30%
- Ejecución 40%
- Informe 30%

I.- Diseño:

- Diseño de las clases, con las correspondientes partes públicas y privadas: 30%
- Programa Principal (main) es simple, todo el trabajo lo hacen las clases: 30%
- Comentarios y Documentación en todo el programa: 20%
- Usar nombres nemotécnicos para las variables, las constantes simbólicas e indentación del código en el programa: 20%.

II.- Ejecución: Todos los ítems tienen la misma ponderación.

- Cada acción que puede realizar el usuario se asocia a un ítem.
- El manejo de ingreso de datos erróneos y los mensajes que el sistema entrega.
- La interfaz que se tiene para el ingreso de los datos.

III.- Informe: Descripción del trabajo realizado, estructuras de datos usadas y diagrama de clases correspondiente. Además, debe incluir el análisis solicitado sobre la utilidad de ambos árboles.

Situaciones especiales:

- Por cada hora de retraso en la entrega, se va descontando 1 punto (10 décimas) a la nota.
- No compila: dependiendo del tipo de trabajo, la nota máxima puede ser 4.0 o 2.0
- Usar otro programa, otro lenguaje u otra materia: 2.0

¡ÉXITO!