

Tarea 2 - Análisis de Algoritmos y Estructura de Datos
Jose Javier Toro Bustos
Universidad de Santiago de Chile 1-2021

I. INTRODUCCION

El contexto de esta tarea se sitúa en la actual pandemia del COVID-19, en la época de vacunación, el problema planteado es llevar registro de gente con dos dosis de una vacuna con su fechas de primera y segunda dosis, su tipo de vacuna, su edad, entre otros datos, también se debe tener registro de gente que solo tenga su primera dosis, debiendo calcular cuando tendrá que administrarse la segunda dosis de su vacuna correspondiente, con esta nueva información se debe crear un nuevo registro, el cual almacena la cantidad de dosis necesarias de cada tipo de vacuna para diferentes meses.

En resumidas cuentas, el problema se resume en tres problemas más pequeños, Primero crear un registro de quienes tengan ambas dosis, en segundo lugar, crear un registro de quienes tengan solo una dosis, calculando la fecha de su segunda dosis y en último lugar, crear un registro en el cual deben estar calculadas las provisiones de cada vacuna para distintos meses. Todo lo anteriormente descrito se debe resolver utilizando el lenguaje de programación C, implementando listas enlazadas y los registros se deben entregar por separado en archivos de extensión ".in".

II. SOLUCION PROPUESTA

Debido a las condiciones propuestas en la introducción la solución de esta Tarea se vio enfocada de una forma muy directa desde el inicio, la cual era simplemente, leer los archivos de entrada entregados, ordenar las listas resultantes de estas lecturas, una vez leídos ordenarlos alfabéticamente según sus apellidos, para luego ordenarlos por días y finalmente por meses, una vez ordenados los datos, calcular las fechas de las segundas dosis de quien corresponda, una vez todo ordenado crear los archivos de salida y escribir la información correspondiente dentro de estos, para hacer todo esto se utilizaron estructuras, las cuales son para representar primero a un individuo, una para representar un tipo de vacuna y luego dos estructuras para representar las listas, una para las listas y otra para las vacunas.

Estas estructuras recibían los siguientes nombres:

- nodoG, la cual representada a un individuo.
- nodoV, la cual representaba una vacuna.
- listaG, la cual representaba una lista de individuos.
- listaV, la cual representaba una lista de vacunas.

Con estas cuatro estructuras se realizan todos los cálculos y procesos para resolver el problema, a continuación, solo se mostraran las estructuras de vacunas y de individuos.

Primero la estructura que representa a los individuos:

```
typedef struct nodoG{
    char* rut;
    char* nombre;
    char* apellido;
    int edad;
    int dia;
    int mes;
    int year;
    int idVacuna;
    int dia2dosis;
    int mes2dosis;
    int year2dosis;
    struct nodoG* siguiente;
}nodoG;
```

Y ahora la estructura que representa una vacuna:

```
typedef struct nodoV{
    int IDV;
    char* codigoV;
    char* nombreV;
    int semanasV;
    int dosisNecesarias;
    struct nodoV* siguiente;
}nodoV;
```

Las estructuras de listas son simplemente un puntero a el primer nodo de estas.

Las funciones en su mayoría son de tipo void ya que esto facilitaba en gran medida la solución del problema. Ya que se utilizó en la mayoría de las funciones el paso por referencia visto en cursos anteriores.

Las funciones más importantes son las utilizadas para leer y escribir archivos y en parte las funciones utilizadas para ordenar las listas una vez creadas o durante su creación.

A continuación, se mostrarán mediante pseudocódigo algunas de estas funciones ya que debido al formato con que venía la información en los archivos de entrada, las funciones para leer los archivos son bastante similares.

La siguiente función general las listas de individuos con primera dosis y con segunda dosis según el string entregado como parámetros.

```
leerGente(listaG* lista, string doc){
    num cantidadG
    num i
    num edad
    num vacunaID
    num día,mes,year
    string slash="/"
    FILE* archivoG <- fopen(doc,"r")...abro el
archivo
    ...leo la primera linea del documento y lo
guardo en cantidadG
    For i=0 to cantidadG do
        ...dentro de este for leo línea a
línea del archivo
```

...siguiente el formato indicado
 en los archivos de entrada
 ...utilizando el string slash separo
 la fecha que venia con el formato
 ... "DD/MM/YY" lo separo
 mediante slash y lo guardo como numero
 ...con esto leído, inserto un nodo
 con los parametros leídos dentro de la lista
 fclose(archivoG)...cierro el archivo.

}
 La siguiente función genera la lista de vacunas
 leyendo el archivo indicado
 leerVacunas(listaV* listaV){
 num cantidad
 num i
 num IDV
 num semanasV
 FILE* archivoV <- fopen("vacunas.in","r")
 ...leo la primera linea del documento y lo
 guardo en cantidadV
 For i=0 to cantidadV do
 ... dentro de este for leo línea a
 línea creando un nodo por cada una de estas
 ...luego cada nodo es insertado
 dentro de la lista entregada como parametro
 fclose(archivoV)... cierro el archivo
 ahora se mostrará una de las funciones que
 generan los archivos de salida.

Esta función genera la salida
 vacunacionCompleta.out
 salidaVacunacionCompleta(listaG* vc, listaV*
 vacunas, listaG* dosis2)
 FILE* salida1 <-
 fopen("vacunacionCompleta.out", "w")
 ...escribe en el archivo el largo de la lista
 dosis2
 nodoG* agregar
 while(agregar <> NULL)
 string código
 código <- vacunaCodigo(agregar-
 >idVacuna, vacunas)
 ...escribo en cada linea el nodo
 actual entregando el string
 ... "rut nombre apellido edad
 fecha codigoVacuna"
 agregar=agregar->siguiente
 fclose(salida1)...cierro el archivo
 }

Esta función genera la salida provision.out
 calcularProvisiones(listaG* v1d, listaV* vacunas){
 nodoG* gActual
 nodoV* vacunaActual
 num i
 num vacunasMes
 string mes
 FILE* salida2 <-
 fopen("provisiones.out", "w")
 For i=1 to 13 do
 gActual <- v1d->inicio
 vacunasMes <- 0
 vacunaActual <- vacunas->inicio

while(vacunaActual <> NULL) do
 vacunaActual->
 >dosisNecesarias <- 0
 vacunaActual <-
 vacunaActual->siguiente
 while(gActual <> NULL) do
 if(gActual->mes2dosis =
 i) then
 vacunaActual <-
 buscarVacuna(vacunas, gActual->idVacuna)
 vacunaActual->
 >dosisNecesarias <-
 >dosisNecesarias+1
 vacunasMes++
 gActual <- gActual->
 >siguiente
 if(vacunasMes <> 0) then
 ...selecciono el mes en
 una matriz con el nombre de cada mes,
 dependiendo del valor de i
 ...escribo el mes y la
 cantidad de vacunas de este mes
 vacunasActual <-
 vacunas->inicio

while(vacunaActual <> NULL) do
 ..ahora
 imprimo la vacuna y su cantidad necesaria
 vacunaActual->
 >dosisNecesarias <- 0
 vacunaActual <- vacunaActual->siguiente
 fclose(salida2)...cierro el archivo
 }

Esta función genera la salida listado.out
 salidaListado(listaG* v1d, listaV* vacunas){
 num i
 nodoG* gActual
 num vacunadosMes
 FILE* salida3 <- fopen("listado.out", "w")
 For i=1 to 13 do
 gActual <- v1d->inicio
 vacunadoMes <- 0
 while(gActual <> NULL) do
 if(gActual->
 >mes2dosis=i) then
 vacunadosMes++
 gActual <- gActual->
 >siguiente
 if(vacunadosMes <> 0) then
 ...selecciono el mes en
 una matriz con el nombre de cada mes,
 dependiendo del valor de i
 ...escribo el mes y el
 nodo de cada individuo
 Fclose(salida3)...cierro el archivo
 }

Estas son las funciones principales dentro de la solución creada. El resto de las funciones se pueden encontrar dentro de los archivos .h titulados listaG.h y listaV.h los cuales contienen funciones relacionadas con la lista que lleva su mismo nombre.

III. Resultados y análisis

En cuanto a los resultados obtenidos estos pueden verse reflejados en los tres archivos de salida correspondientes, estos resultados se adjuntan en las siguientes fotos.

```
listado.out: Bloc de notas
Archivo Edición Formato Ver Ayuda
Mayo 2
12573685-8 Esther Alves 46 26/4/21 25/5/21 BTN162B2
13152318-1 Valentin Valenzuela 45 29/4/21 28/5/21 BTN162B2
Junio 8
19237827-3 Laura Gomez 25 30/3/21 11/6/21 AZD1222
19200759-3 Mateo Gonzalo 24 5/4/21 16/6/21 AZD1222
16950436-9 Eric Tejero 32 18/5/21 16/6/21 Coronavac
17965738-4 Alicia Martín 29 24/5/21 22/6/21 BTN162B2
14688538-1 Fernando Garriga 27 28/5/21 26/6/21 Coronavac
19564843-3 Marcos Sanchez 24 28/5/21 26/6/21 BTN162B2
19352231-9 Simon Candia 25 31/5/21 29/6/21 BTN162B2
19162681-8 Maria Paz 25 31/5/21 29/6/21 BTN162B2
Julio 4
19922942-7 Dario Chavez 22 7/6/21 6/7/21 Coronavac
20469156-8 Angel Acosta 21 8/6/21 7/7/21 Coronavac
20827812-6 Luna Vallejo 19 11/6/21 10/7/21 Coronavac
12955888-1 Antonia Collado 44 3/5/21 14/7/21 AZD1222
```

Figura 1. Archivo salida listado.out

```
provision.out: Bloc de notas
Archivo Edición Formato Ver A
Mayo 2
BTN162B2 2
Coronavac 0
AZD1222 0

Junio 8
BTN162B2 4
Coronavac 2
AZD1222 2

Julio 4
BTN162B2 0
Coronavac 3
AZD1222 1
```

Figura 2. Archivo salida provision.out

```
vacunacionCompleta.out: Bloc de notas
Archivo Edición Formato Ver Ayuda
7
11957103-0 Mario Acevedo 49 10/5/21 Coronavac
8822700-k Samuel Cruz 59 9/4/21 BTN162B2
8822700-k Samuel Cruz 59 7/5/21 BTN162B2
11299516-1 Vidal Gordillo 54 31/3/21 Coronavac
11299516-1 Vidal Gordillo 54 28/4/21 Coronavac
11867277-1 Lara Mayoral 49 12/4/21 Coronavac
11867277-1 Lara Mayoral 49 10/5/21 Coronavac
11703970-6 Adriana Ordoñez 50 7/4/21 BTN162B2
11703970-6 Adriana Ordoñez 50 5/5/21 BTN162B2
19937663-2 Cristobal Perez 23 5/4/21 Coronavac
19937663-2 Cristobal Perez 23 3/5/21 Coronavac
19676900-5 Josefina Sepulveda 23 5/4/21 BTN162B2
19676900-5 Josefina Sepulveda 23 3/5/21 BTN162B2
```

Figura 3. Archivo salida vacunacionCompleta.out

Estas salidas se originan con los archivos de salida entregados en la plataforma Uvirtual, por lo cual dependerán de los archivos de entrada, cabe recalcar que los archivos de entrada deben tener los mismos formatos y nombres que los descritos en la plataforma, esto implica que como los archivos de entrada son siempre el mismo el tiempo de ejecución sin importar la cantidad de veces que se intente será el mismo, el cual utilizando la librería time.h, en específico su funcionalidad clock_t entrega un tiempo de ejecución 0.000000 segs.

IV. Conclusiones

En esta experiencia el logro del objetivo el cual era generar los tres archivos de salida, fue cumplido prácticamente a cabalidad, se generan exitosamente los archivos de salida, ordenados por fecha de su segunda dosis y luego alfabéticamente, las ventajas de la solución implementada es la sencillez de su código, la facilidad de su implementación y por ultimo, la forma de recorrer, modificar y/o añadir nodos a las listas es bastante sencilla de comprender con la practica suficiente ya que al momento de que el mismo programador define la estructura básica con la que trabaja hace que el código sea más legible para el mismo.

ANEXO MANUAL DE USUARIO

Para la correcta ejecución de este código debe tener en una carpeta cuatro archivos, el archivo compilable .c y los tres archivos de entrada descritos al final de este anexo.

El uso de este código es simplemente compilar y ejecutar ese compilado, La forma de compilar varia levemente dependiendo del sistema operativo.

En Linux debe abrir la terminal, mediante el uso del comando cd, debe ir al directorio donde tiene el archivo .c y luego escribir:

gcc C3_JoseToro.c -o lab.exe

esto creara un ejecutable dentro de la carpeta donde tenga almacenada el .c luego de esto debe escribir el comando: **./lab.exe**

Por otra parte, en Windows el proceso es prácticamente el mismo, primero debe abrir la consola de comandos (CMD), mediante el uso del comando cd debe llegar a la localización del archivo .c y ejecutar el mismo comando que en Linux

gcc C3_JoseToro.c -o lab.exe

una vez ejecutado este comando, debe realizar el llamado muy similar que en Linux solo que sin utilizar el ./ delante del llamado **lab.exe**

si todo está realizado correctamente se generarán tres archivos de salida llamados de la siguiente forma:

-“listado.out”

-“provision.out”

-“vacunacionCompleta.out”

Los cuales entregan los siguientes datos:

“listado.out”

debe contener un listado ordenado de las personas vacunadas (separados por mes señalando cantidad de personas vacunadas en dicho mes) indicando rut, nombre, apellido, edad, fecha en la cual se colocó la primera dosis, fecha en la cual se debe colocar la segunda dosis y el nombre de la vacuna correspondiente.

“provision.out”

debe contener un listado ordenado por tipo de vacuna indicando la cantidad de segundas dosis requeridas para cada mes (separados por mes, señalando cantidad de segundas dosis requeridas en total).

“VacunacionCompleta.out”

debe contener un listado ordenado por nombre y apellido (priorizando apellido), el cual indicará el rut, nombre, apellido, la fecha de la primera o segunda dosis de las personas vacunadas con el proceso completo además del nombre de la vacuna, separadas por un salto de línea.

Cabe destacar que los archivos de entrada necesarios para la ejecución son los siguientes:

-vacunados1D.in

-vacunados2D.in

-vacunas.in

Los formatos dentro de estos archivos .in están descritos en la plataforma Uvirtual y resumidamente son los presentes a continuación:

“vacunas.in”

contiene la cantidad de tipos de vacunas en Chile y posteriormente el id de la vacuna, su nombre, fabricante y período en semanas entre primera y segunda dosis.

“vacunados1D.in”

contiene el número de personas que ya recibieron la primera dosis de su vacuna y posteriormente, la información de cada persona en específico, su rut, nombre, apellido, edad, fecha de primera dosis, el id del tipo de vacuna que se le colocó.

“vacunados2D.in”

contiene el número de personas que ya recibieron la segunda dosis de su vacuna y posteriormente, la información de cada persona en específico, su rut, nombre, apellido, edad, fecha de segunda dosis, el id del tipo de vacuna que se le colocó.