Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

Sistemas de Operación

Septiembre-Diciembre 2021

Proyecto n° 2

Integrantes:

Daniela Torres # 12-11337

Ernesto Guadarrama #14-10445

Luis Silvera #12-11278

Danimar Suárez #15-11394

Caracas , enero 2022

Introducción

La llegada de un nuevo virus es sin duda un suceso difícil de predecir y en efecto controlar. Se esparce tan fácilmente que descubrir su origen o el punto de partida que dio paso a la propagación resulta una tarea compleja. Sin embargo una comienza a propagarse no queda de otra que realizar una investigación pertinente para determinar qué individuos han tenido contacto con el agente transmisor y ponerlos en estado de cuarentena. Este proceso resultaría más sencillo si se llevase un control computarizado de todas aquellas personas con las que la persona infectada ha tenido contacto pues permitiría tomar medidas de forma rápida.

En la Universidad Simón Bolívar un nuevo brote ha llegado, se debe evitar a toda costa su propagación, En este sentido, el objetivo principal del proyecto es desarrollar una aplicación de rastreo de contactos (contact tracing) que pueda analizar la severidad de casos potenciales, haciendo uso de un directorio suministrado por DACE que contiene todos los comprobantes de inscripción y listas de clase. ¿Se logrará evitar la propagación?.

Decisiones del diseño

El primer paso para desarrollar el proyecto consistió en comprender los requerimientos del problema para poder dar una solución correcta. Nuestras soluciones parten a raíz de la siguiente premisa, “dado el carnet del “estudiante cero” (infectado) se requiere hacer una búsqueda en el directorio suministrado para obtener su comprobante, de esta forma tener acceso a las materias que cursa dicho estudiante y así obtener de cada asignatura los carnets de sus compañeros y profesores que automáticamente pasan a ser posibles infectados”.

Posteriormente se tomaron las siguientes decisiones para dar paso a nuestra solución:

- Hacer uso de estructuras como listas y listas enlazadas, para almacenar los carnets de las personas infectadas y asignaturas. También es necesario construir las funciones que se utilizan para recorrer, imprimir y agregar elementos a dichas listas.

- Construir una función que lea los archivos de estudiantes y de cursos. Luego almacena en la lista los estudiantes encontrados. considerando antes que no se hayan guardado en una instancia o curso anterior.

- Hacer una estructura que contiene cada una de las variables que guardan toda la información que introduce un determinado usuario como las instancias, direcciones y carnets.

- Hacer uso de directorios que permitan almacenar los archivos para ser manipulados posteriormente y crear una función que guarde las funciones que recorren los directorios.

- Una función donde se lean todos los archivos de carnet y materias.

Detalles de la implementación.

A continuación se presentara de forma detallada cada una las funciones,listas y estructuras que se diseñaron para lograr cumplir con cada uno de los requerimientos del proyecto “rastreo de contactos”:

- Función “ListaData”:

Dicha función está compuesta por los archivos “ListaData.c” y “ListaData.h”, donde se realizan todas las operaciones de las listas del proyecto. En principio tiene tres estructuras: la primera es “LEst” que corresponde a la lista de los estudiantes que pueden estar enfermos, en esta se extraen los carnets para buscar los comprobantes que le corresponden a cada uno y a partir de ello obtener las materias que se encuentran cursando, con el fin de agregarla a la lista de materias que es la segunda estructura llamada “LMat”, encargada de buscar la materia y los demás estudiantes que están contagiados. En este sentido, ambas estructuras se usan para el control interno del programa, una para llevar el control de los estudiantes que pueden estar enfermos y la segunda para llevar el control de las materias que contienen estudiantes enfermos. La tercera estructura es “LFinal” que es una lista de materias, donde cada materia tiene un puntero a cada estudiante; básicamente lo que hace dicha estructura es crear materias y una lista de estudiantes anexadas a esas materias.

Otro elemento que tiene esta función es un identificador “identificar_LEst” para control y por si se requiere borrar algo de una lista. Tiene una función llamada “agregarLFinal” que se usa para agregar una materia a la lista final de las materias

complejas, también están las funciones “agregarLEst” y “agregarLMat” que agregan estudiantes y materias.

Seguidamente se tiene las funciones que imprimen: la primera es “imprimirLMat”, la segunda es “imprimirLMatPorPasos” y la tercer que es la impresión final se llama “imprimirLFinal” que tiene dos while, uno que recorre las materias y dentro de este hay otro que recorre los estudiantes de cada materia.

Por último tiene dos funciones llamadas “compararCarnet” y “compararCodigo” que se usan para que no exista una repetición de estudiantes o materias en ninguna de las listas, la primera es para que no se agreguen estudiantes enfermos nuevos a las lista de estudiantes enfermos y la segunda para no agregar materias que ya han sido agregadas a la lista final.

- Función “Variables”:

Está compuesta por el archivos “Variables.h”, donde se crea una estructura llamada “Entrada”. En la función del “main.c” se coloca una instancia de esta estructura como una variable global, cuando se procede a darle argumentos al programa, una función toma lo que introduce el estudiante y lo guarda en esa variable global.

Dicha estructura está compuesta por las siguientes variables: las instancias que guarda el número de instancias que da el usuario, el directorio que guarda la dirección de la carpeta a revisar dada por el usuario y es donde se puede encontrar la carpeta DACE y el pacienteCero que guarda el número de carnet que dió el usuario.

- Función “Recorredor”:

Función compuesta por los archivos “Recorredor.c” y “Recorredor.h”, esta es llamada en el “main.c” , donde se le pasa el directorio metido por el usuario y con ella regresa un “char* contenido” el path de la carpeta DACE, si la encuentra en ese directorio.

Dicha función lo que hace es una variable “DIR*” que es de “dirent.h” para abrir el directorio que refiere el “char*”. Ahora bien, si el directorio no se puede abrir se regresa un “char* ”” ”, en cambio, si lo puede abrir se declara una estructura “dirent*” para recorrer ese directorio que se abre con “readdir”, y usa un bucle para recorrer ese directorio hasta que logre leer todo. Al encontrar un archivo que sea del tipo “directorio/carpeta”, lo que se hace es revisar si su nombre es “DACE”, si no lo es continúa el proceso, pero si coincide con el nombre lo que hace es retornar el path de la carpeta DACE antes de que termine el bucle. Si el bucle termina, quiere decir que no encontró la carpeta, por lo que manda un mensaje de error y cierra el programa.

- Función “LectorArchivos”:

Formado por “LectorArchivos.c” y “LectorArchivos.h”. Dicha función incluye “ListaData” porque usa las listas y todas las variables de las listas. Tiene la función “pasarCadena2” que se usa para agregar los estudiantes a la lista compleja final que es la que tiene la salida, es decir, que dicha función lo que hace es pasarle a lista compleja todos los estudiantes que fueron leídos en un curso y al mismo tiempo lo agrega a la lista de estudiantes enfermos. Otra de las funciones es la llamada “LeerCarnet” que básicamente lo que hace es abrir el comprobante de un estudiante y guarda las materias que se encuentra cursando en la lista de materias que será posteriormente leída; “LeerCurso” es usado para leer una materia, extrae cuál es el profesor, la sección, el código y los estudiantes que están cursando ese curso. La función “ArmarDireccionCurso” se encarga de armar la dirección que se usa para abrir la materia; “recorrerLMatAbrirCursos” es la función que recorre la lista de materias y abre todos los cursos para extraer los datos para colocarlos en la lista de materias. Por último está la de “ArmarDireccion” que define la dirección para abrir el comprobante y ”recorrerLEstAbrirComprobantes” que recorre la lista de estudiantes para abrir sus comprobantes.

- Función “main.c” que corresponde a nuestro programa principal donde se llaman y se integran todas las funciones, listas, variables y estructuras que hacen posible la corrida del presente proyecto

Análisis de corridas.

Las corridas fueron realizadas en función de la lista de ***"posibles"*** super contagiadores suministrada, cuyos carnets son los siguientes: 1410261, 1410848, 1511326, 1710248, 1800262. A continuación se exponen los resultados obtenidos.

Corrida nº1- Carnet: 1410261

Al efectuar la primera corrida correspondiente al "estudiante cero" contagiado se puede observar efectivamente (Figura 1) los cursos en los que dicho estudiante está inscrito, por ende las personas que han tenido contacto con él y que automáticamente se convierten en posibles contagiados.

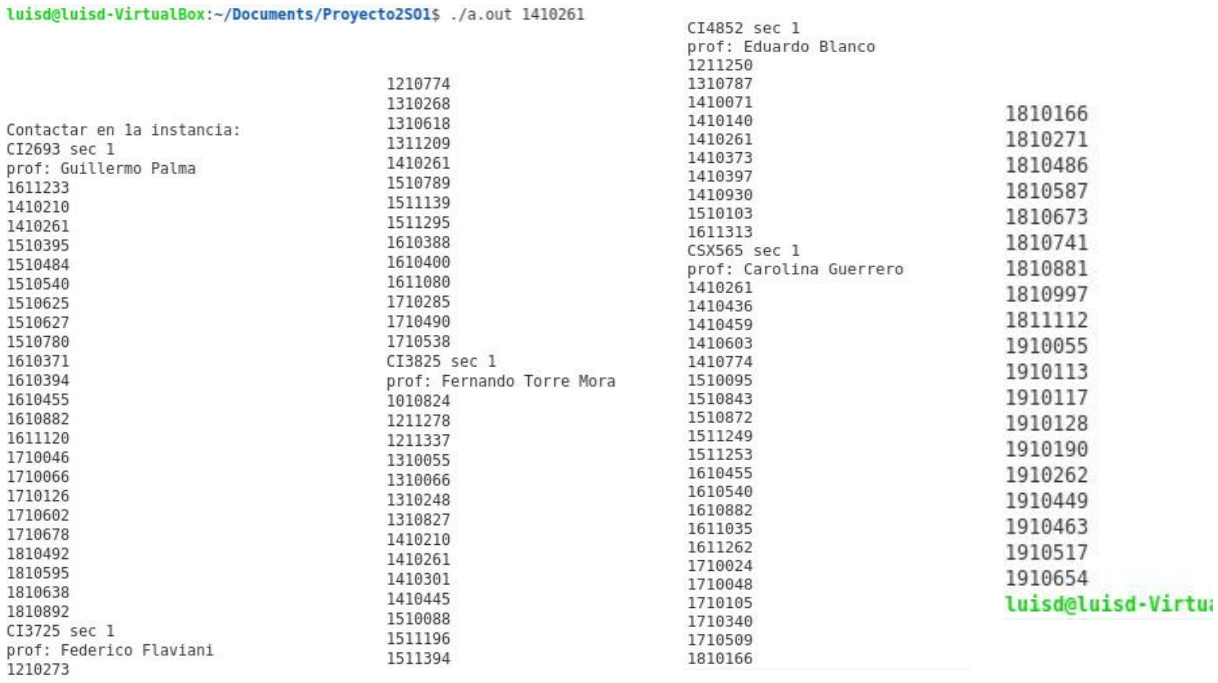

Figura 1. Estudiante supercontagiador 1, carnet 1410261

Corrida nº2- Carnet: 1410848

Al efectuar la segunda corrida correspondiente al estudiante cuyo carnet es 1410848 no se obtuvo resultados pues el carnet no existe en el archivo (Figura 2) , de igual forma se realizó una revisión manual de dicho archivo y efectivamente no fue localizado.

```
luisd@luisd-VirtualBox:~/Documents/Proyecto2SO1$ ./a.out 1410848
No existe el comprobante de este estudiante.
luisd@luisd-VirtualBox:~/Documents/Proyecto2SO1$
```

Figura 2. Estudiante supercontagiador 2, carnet 1410848

Para las corridas *nº3, nº4, nº5* correspondientes a los estudiantes con carnets 1511326, 1710248, 1800262 no se obtuvieron resultados pues tampoco se encontraban en el archivo.

Sin embargo se decidió hacer unas corridas adicionales con los carnets de los integrantes de nuestro equipo como por ejemplo el carnet 1511394, aqui se observa las asignaturas inscritas, personas que han interactuado con d

```
1410445
luisd@luisd-VirtualBox:~/Documents/Proyecto2SO1$ ./a.out 1511394


Contactar en la instancia:
CI3825 sec 1
prof: Fernando Torre Mora
1010824
1211278
1211337
1310055
1310066
1310248
1310827
1410210
1410261
1410301
1410445
1510088
1511196
1511394
EC1311 sec 1
prof: Ivan Scheurmman
1610715
1610923
1710651
1810394
0810436
1110108
1210067
1310066
1310936
1311124
1511196
1511394
EC1311 sec 1
prof: Ivan Scheurmman
1610715
1610923
1710651
1810394
0810436
1110108
1210067
1310066
1310936
1311124
1410125
1410301
1410722
1410936
1511394
1610382
1610629
1710196
1710498
EC5756 sec 1
prof: Emma DBatista
0740794
1211044
1211337
1310269
1410079
1410767
1510023
1511078
1511394
```

Figura 3. Integrante del equipo seleccionado al azar, carnet 1511394

Conclusiones

Los directorios son contenedores virtuales que almacenan archivos informáticos y otros subdirectorios que permiten el acceso a los datos de la manera más óptima.

A partir de cada una de las funciones, estructuras, variables y listas se logró hacer un programa donde se puede ver de manera ordenada y automatizada los estudiantes que se encuentran cursando una determinada materia, con la finalidad de llevar un control en la comunidad universitaria para que sea más sencillo detectar los contagios que se puedan presentar y tomar las acciones pertinentes. A partir de este proyecto, se puede saber de una manera directa y sencilla las personas que pueden estar contagiadas con saber el paciente cero y todos los que tuvieron contacto con este en las materias que pudieron compartir.

En conclusión, estos proyectos valen la pena ser aplicados en todas las comunidades universitarias en estos tiempos de pandemia, porque podría ayudar a ser posible el regreso de las clases presenciales al tener mucho más orden y control.