

TP Algoritmos y estructura de datos

K1025 – Grupo 5



Francisco Bosi - 2034270 - fbosi@frba.utn.edu.ar

Juan Manuel Menes – 2038468 - jmenes@frba.utn.edu.ar

Matías Zuran - 2041881 – mzuran@frba.utn.edu.ar

Diagrama

Descripción generada automáticamente

# Estructura del programa

Todo el programa funciona según las pólizas que se levantan al principio del programa. Estas pólizas se encuentran en el archivo asegurados.bak que se encuentra dentro de la carpeta principal del programa. Cabe destacar que el programa está preparado para funcionar sin ningún contratiempo en el caso de que el archivo asegurados.bak no exista.

El levantamiento de pólizas consiste en el proceso de pasar a memoria toda la información contenida en el archivo, recorriéndolo con un puntero auxiliar de principio a fin. Cada vez que el puntero lector lee una póliza contenida en el archivo, se generará un nuevo módulo de lista Simplemente Enlazada para almacenar la nueva información antes de seguir leyendo. Este proceso se repite tantas veces sea necesario hasta que se acaben las pólizas contenidas en el archivo.

La ventaja de usar como estructura de datos a una lista Simplemente Enlazada en lugar de alguna otra opción como podría ser un vector, es que la lista permite un número flexible de pólizas, sólo utilizando la cantidad de memoria necesaria y no dejando espacios en blanco u basura. El pasaje a memoria también agiliza muchísimo el procesamiento de datos en lugar de tener que leer y sobrescribir el archivo cada vez que se quiera hacer hasta el más mínimo de los cambios.

# Funciones Adicionales

validarDatos(nodoListaSE\*&);

Esta función recibe el nodo de una lista y determina si tiene algún dato NULL u de otro modo inválido (Como un dni 0 o campos vacíos en nombre, apellido o patente).

En caso de que todos los datos sean válidos, la función devolverá un no cero. En caso contrario, el valor de return de la función será 0.

# Procesado de incidentes

La función de nuestro programa es simular el manejo de datos de una empresa aseguradora que permita registrar asegurados y los incidentes que los envuelven.

Al igual que los asegurados, los incidentes denunciados también son levantados desde un archivo externo de la misma forma ya descrita y pasados a memoria por medio de una lista Simplemente Enlazada.

Al momento de procesar los incidentes, el programa tiene 2 tareas;

Aumentar la cantidad de incidentes de los asegurados atributados al incidente

Trasladar los registros de los incidentes procesados a un nuevo archivo llamado “procesados.bak”

Todo incidente que afecte directamente a uno de los asegurados por la empresa (o sea, todo aquel que forme parte de asegurados.bak) se considerará un incidente procesado y será copiado a procesados.bak junto a todos los demás incidentes que cumplan esta misma condición. De lo contrario, si se copiasen todos los incidentes que recibe nuestro programa, los contenidos de incidentes.bak y procesados.bak serían exactamente iguales y la existencia de uno de los dos sería redundante.

# División de tareas en el equipo

Antes de explicar la división de responsabilidades en el grupo de trabajo debemos describir los cambios sufridos en la composición de nuestro grupo. Una vez divididas las tareas originalmente, el estudiante Valentino Ariotti dejó la carrera, y por extensión, nuestro grupo. Desgraciadamente, no nos comunicó de su decisión y por consecuencia seguimos contando con su apoyo hasta que le preguntamos directamente por qué no hablaba en el grupo de whatsapp.

El código fue trabajado y retocado por los integrantes del grupo a medida que surgía la necesidad, ya sea por un nuevo error o para hacer compatibles unas funciones con otras. Al momento de implementar una función, dos o más integrantes del grupo se reunían a la vez a una llamada en línea para trabajar los diferentes puntos del trabajo listados abajo.

Una captura de pantalla de un celular con texto

Descripción generada automáticamente con confianza media

Se incluye el punto 10 en la entrega dentro de la carpeta *Creador de archivos de incidentes a procesar* como una aplicación c++ independiente del programa principal. Si al momento de procesar los archivos, el programa no cuenta con el archivo de incidentes, se le notificará al usuario de su ausencia y se volverá al menú principal. El usuario simplemente deberá crear un nuevo archivo de incidentes con el programa de creación y meterlo en la carpeta del programa principal. Una vez hecho esto, la función de procesados funcionará normalmente.

Cabe destacar que no es necesario cerrar la instancia del programa principal cuando se reemplaza o se le adjunta un archivo de incidentes.