
TRABAJO PRÁCTICO K1031

ALGORITMOS Y ESTRUCTURA DE DATOS



FECHA DE ENTREGA: 11/11/2021

PROFESOR: ING. PABLO MENDEZ

ALUMNOS: MICAELA ANDREA PAREDES, AGUSTIN EMANUEL TERESIN, GISSBEL CASTILLO

Legajos: 176.128-6, 175.436-1, 1466124

Correos: miparedes@frba.utn.edu.ar, ateresin@frba.utn.edu.ar,
mori@frba.utn.edu.ar

LIBRERÍAS QUE FUERON UTILIZADAS

Para los algoritmos de buscar, levantar, cargar, actualizar, desactivar y menú, fueron usadas las librerías de `<iostream>` `<stdio.h>`, `<stdlib.h>`, `<string.h>`, `<conio.h>`, `<cstring>`.

Para el algoritmo de ordenar y crear html se hizo uso de la librería `<fstream>`.

ESTRUCTURAS

CARGAR

Inicialmente empieza el programa cargando en el archivo unas pólizas ya existentes como también sus correspondientes incidentes. A la hora de cargar una nueva póliza, se le pide al usuario que ingrese todos los datos y los inserta al final de la lista de pólizas usando *insertaralfinalpolizas* (algoritmo que a la vez llama a la estructura *insertarsiguientepoliza*). Para cargar un incidente, sin embargo, es más complejo. Se pide primero el número de póliza al que se le quiere agregar el incidente y si este le pide todos los datos al usuario. Si la fecha ingresada es del año 2020, se llama a la función *sumar2020* para que actualice la cantidad de incidentes de ese año a la vez que actualiza la cantidad de incidentes de la póliza referida. Luego el algoritmo busca si el dni del otro conductor pertenece a alguno de los clientes de la lista, si es así, llama al procedimiento *otroconductor* para que actualice también la cantidad de incidentes de esa póliza. Finalmente, usando *insertaralfinalincidente*, ingresa el incidente que se agregó en la sublista del nodo de pólizas.

MOSTRAR LA LISTA DE PÓLIZAS

El siguiente procedimiento recibe por parámetro la lista del nodo de pólizas. Mediante un while lee la lista y muestra el contenido de cada nodo a la vez que llama al procedimiento *listarincidentes* para mostrar la sublista de incidentes relacionada a cada póliza. El procedimiento *listarincidentes* también se utiliza cuando se realiza la búsqueda de una póliza por el número de póliza, mostrando toda la sublista de incidentes del nodo que encontró.

OTRO CONDUCTOR

El siguiente algoritmo se implementa al momento de procesar un lote de incidentes. Recibe por referencia la lista de pólizas a modificar y por valor los datos del dni del asegurado, dni del otro conductor, número de incidente, fecha, calle y altura. Lo que hace este procedimiento es copiar en el struct incidente todos los datos pasados por parámetro con la diferencia que el dni del asegurado es el dato llamado dni otro y el dni del otro conductor es el dato llamado dni del asegurado. Una vez ya pasados los datos, crea una lista auxiliar para buscar por dni la póliza a la que se le debe agregar el incidente y cuando la encuentra llama a las funciones *actualizar* y *sumar2020* si es que la fecha coincide. Finalmente inserta al final de la sublista el nuevo incidente.

LEVANTAR

El procedimiento de levantar archivos se implementa luego de que se hayan cargado los datos existentes del sistema en los archivos correspondientes. Este algoritmo abre el archivo de pólizas "Asegurados.BAK" y mientras lo lee, carga en una lista previamente pedida por parámetros, todos los datos que contiene el archivo. Para cargar los incidentes en la sublista del nodo de pólizas, primero crea un nodo auxiliar que busca cada número de póliza que aparece en el archivo de

incidentes y si existe, lo carga en la sublista del nodo correspondiente mediante el procedimiento de *insertarsiguienteincidente*.

DESACTIVAR, ACTUALIZAR Y SUMAR2020

El algoritmo *desactivar* lo que hace es recibir por parámetros la lista que se modificará y la información a buscar. Mientras no se encuentre la información buscada se lee la lista comparando, con el uso de "strcmp", si existe algún nodo que coincida y si es correcto, modifica el valor de "activa" para que aparezca desactivado. De la misma forma, las funciones *actualizar* y *sumar2020* reciben por referencia la lista de donde tienen que buscar, el dato buscado y actualizan las cantidades de incidentes. La diferencia entre estos es que *actualizar* recibe por parámetro un dni y recorre la lista hasta el final porque en el caso de que un cliente tenga dos pólizas, ambas pólizas deben actualizarse.

ALGORITMOS DE BÚSQUEDA

Las funciones de buscar por número de póliza y buscar por dni son las mismas, la única diferencia entre ellas es el dato a buscar. Para buscar un número de póliza se le pide al usuario que ingrese el número que desea buscar, crea un nodo auxiliar x para guardar el nodo que devuelva la función *buscarnro* y luego se muestra el nodo encontrado mediante el procedimiento *mostrarpoliza*. Sin embargo, para mostrar las pólizas de un cliente, se llama al procedimiento *mostrarbuscado*, que recibe la lista de pólizas y el dato a buscar y mediante un while lee la lista hasta el final, mostrando sólo los nodos que tengan el dni que se busca.

ORDENAR

Para ordenar la lista de pólizas se utiliza el procedimiento de *burbuja* el cual recibe por referencia la lista a ordenar y llamando a la función *contarnodo*, obtiene el tamaño total de la lista. Lo que hace este procedimiento es recorrer la lista por medio de las variables "int i" "int j". El primer bucle indica la cantidad de pasadas que habrá que hacer para ordenar la lista y va hasta el final-2 para que se pueda acceder a lista->sgte sin excederse del límite. El segundo bucle es la cantidad de elementos que se comparan en cada pasada y va hasta la mitad de la lista. Primero se compara si hay intercambio, es decir si la cantidad de incidentes de aux->sgte es menor al del aux actual, si hay intercambio, luego se evalúa si la lista auxiliar es igual a la lista dada por parámetro, si es así, el nodo se ordena en el lugar que le sigue a aux; si no es así, el nodo se ordena desplazando el aux. En el caso de no haber intercambio, el algoritmo sigue con el siguiente nodo. Una vez ordenada la lista, se implementa el procedimiento *listarpolizarordenadas* que recibe la lista ya ordenada y muestra solo las pólizas que están activas.

MOSTRAR TODAS LAS PÓLIZAS QUE TENGAN MÁS DE 5 INCIDENTES DURANTE TODO EL 2020

Para el procedimiento *crearhtml* se utiliza la librería <fstream>. Lo primero que se hace es abrir un archivo ofstream (archivo de salida) y se escribe en él las sentencias para crear un archivo html. Al mismo tiempo se crea un archivo csv y escribe en éste los encabezados de las celdas. Luego lee la lista de pólizas hasta el final buscando todas las pólizas que tienen una cantidad de incidentes en el año 2020 mayor que cinco y las pólizas que encuentra las escribe en el archivo html dentro de una tabla así como también en el archivo csv.

FINALIZAR JORNADA

El procedimiento *guardarPolizasArchivo*, recibe por parámetro la lista de las pólizas. Se abre el archivo "Asegurados.BAK" en modo escritura y con la ayuda de un nodo auxiliar recorre la lista y mientras ésta no esté vacía, escribe en el archivo la información de la lista de las pólizas activas unicamente. La función empleada en el procedimiento anterior es *obtenersiguientepoliza*, que recibe como parámetro un nodo y devuelve el siguiente si es que éste no está vacío, de lo contrario retornará null. Este procedimiento, a la vez, también abre el archivo de incidentes y escribe en él todos los incidentes de las pólizas activas. Finalizados estos procedimientos, se utiliza *borrarlistapoliza* para que no queden memory leaks. Así como también se borra la lista al comienzo del programa para asegurarse que no hayan quedado datos en las listas.

DIVISIÓN DE TAREAS

ALGORITMOS DEL PROGRAMA

Teresin:

"insertaralfinalpoliza", "insertaralfinalincidente", "buscarnro", "buscarnro", "buscarincidente", "insertarsiguienteincidente", "insertarsiguientepoliza", "levantar".

Castillo:

"guardarPolizasArchivo", "guardarIncidenteArchivo", "sobreescribirArchivo", "obtenersiguientepoliza", "obtenerultimapoliza", "obtenersiguienteincidente", "obtenerultimoincidente", "borrarlistapoliza", "borrarlistaincidente", "crearhtml".

Paredes:

"levantarpoliza", "otroconductor", "actualizar", "desactivar", "sumar2020", "ordenar", "burbuja", "crearhtml", "mostrarlistapolizas", "listarincidentes", "mostrarpoliza", "mostrarbuscado", "levantar".

