UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Trabajo práctico Nº 2

Aseguradora BIT

Algoritmos y Estructuras de datos

Docentes:

- Pablo Damian Mendez

Integrantes:

Nombre y apellido	Legajo	email
Franco Tomas Callero	1782514	fcallero@frba.utn.edu.ar
Luciano Martinez	174408-2	lumartinez@frba.utn.edu.ar
Dario Victor Ozuna	175996-6	dozuna@frba.utn.edu.ar
Benjamín Soto	159879-0	elvissotopascacio@frba.utn.edu.ar

Fecha de entrega:

- 28 de Octubre de 2021

Explicación

Para el problema propuesto, decidimos utilizar el concepto de listas y nodos para las operaciones que se realizan en el día como desactivar una póliza o listarlas en pantalla, etc. La lista principal contendrá las pólizas activas y tendrá una sublista para el lote de incidentes según cada póliza. Para guardar las pólizas activas y los incidentes se usarán los archivos binarios y finalmente para procesar los registros en CSV y HTML se usarán los archivos de texto, que vimos en clase. Luego para que el programa realice las funciones especificadas se nos ocurrió realizar un menú en la que el usuario indica la operación a realizar por teclado.

División de tareas

En cuanto a la división de tareas, cuando ya aclaramos los que tenemos que hacer (las funciones y listas que necesitamos) , cada uno eligió qué función, quedando de la siguiente manera:

- Dario Victor Ozuna:
  void agregarElemento(Nodo *&lista, registroAsegurado elemento);
  void levantar_cuentas(Nodo *&lista);
  int contarLista (Nodo *lista);
  Nodo* buscarAnterior(Nodo *lista, Nodo *aux); -
  void mostrar_polizaCSV(Nodo *&lista);
  void ordenar_polizas(Nodo *&lista);
  int contarAccidentes (Nodo *lista)
  void listar_polizas(Nodo *lista);
- Luciano Martinez:
  Nodo *busqueda_sec (int pol_busc ,char dni_busc[], Nodo *lista);
  void cargar_indicentes();
  void procesar_indicentes(Nodo *lista);

void AgregarElemento2(NodoIncidente *&lista2, registroIncidente elemento);

- Franco Tomas Callero:

  void limpiarListas(Nodo *&lista);

  void limpirarListaIns (NodoIncidente *&lista);

  void desactivar_poliza(int npol, Nodo *&asegurados);

  void cargar_pol(Nodo *&lista);

  void finalizar_jornada(Nodo *&lista , NodoIncidente *&incidente);

  void buscar_poliza(Nodo *&lista);

- Benjamín Soto:

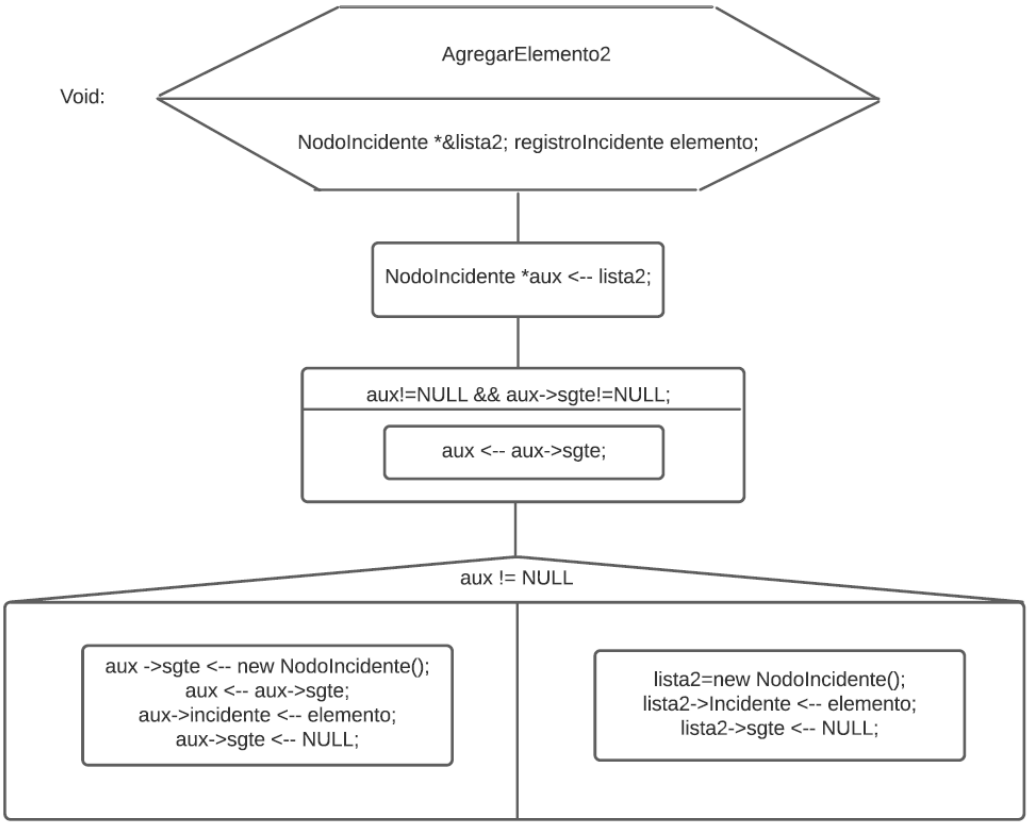
  void mostrar_polizaHTML(Nodo *lista_RegistroAsegurado);

  void mostrarMenu();

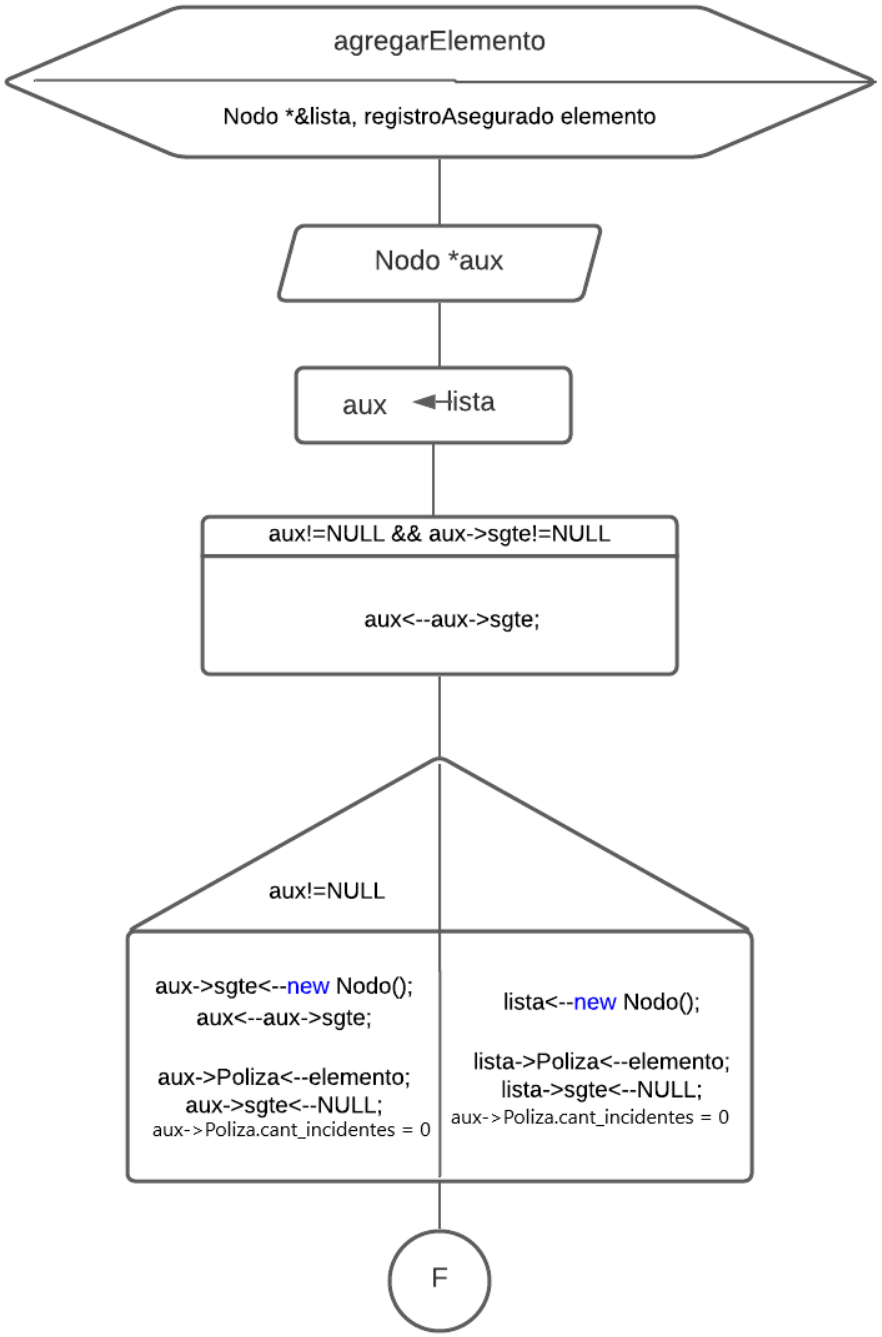
  int main();

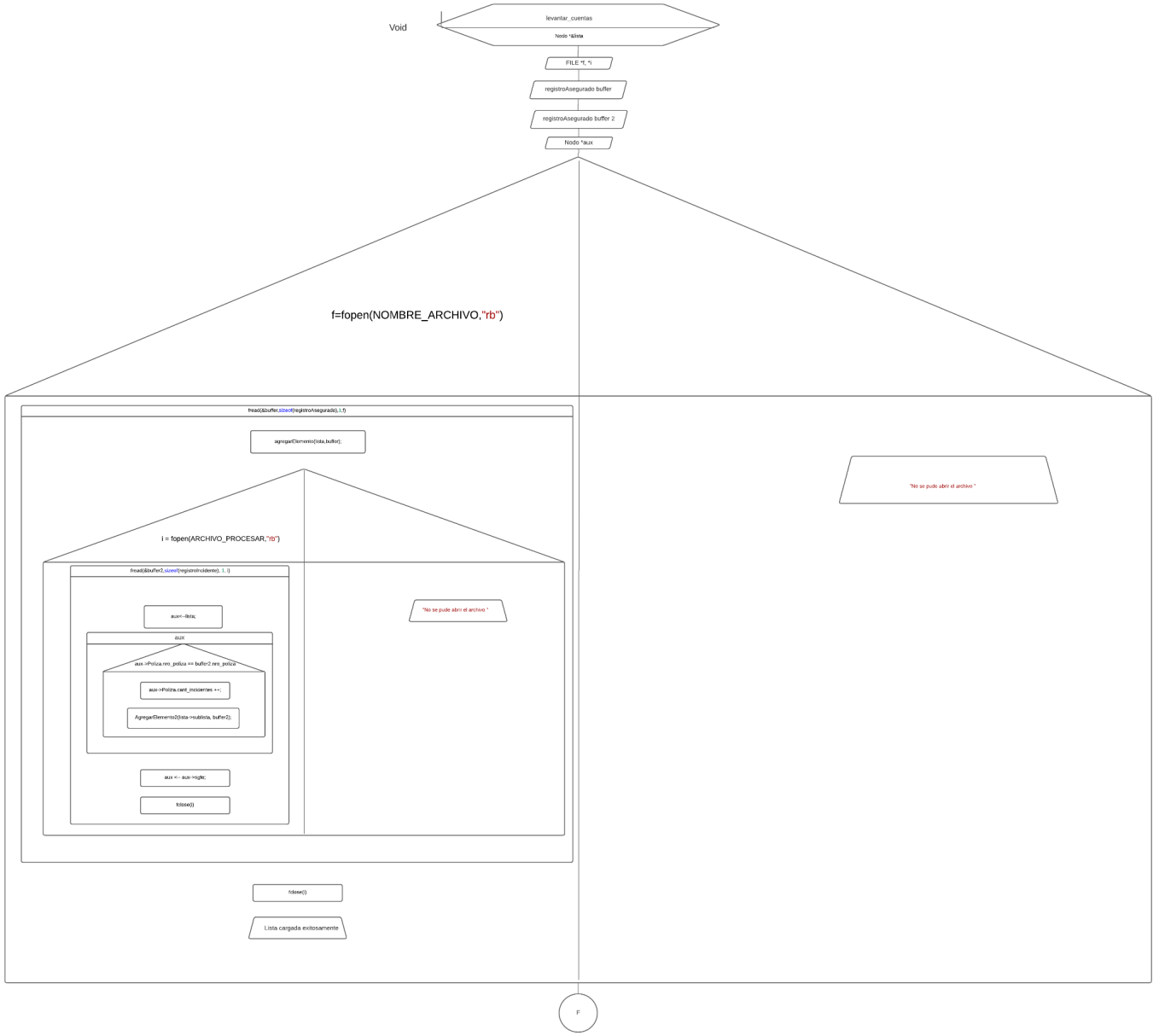
Diagramas de lindsay

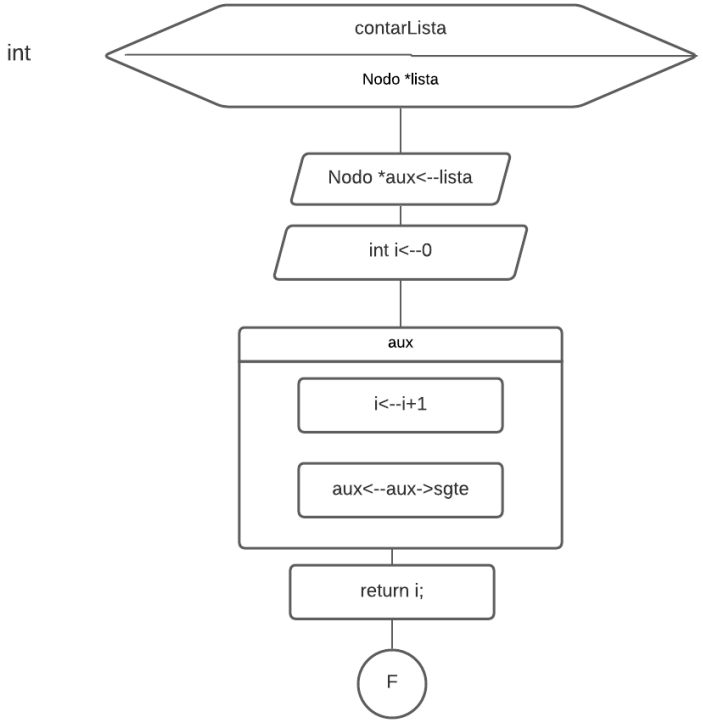
Ahora lo que hace cada subprograma se los puede entender de la siguiente manera:

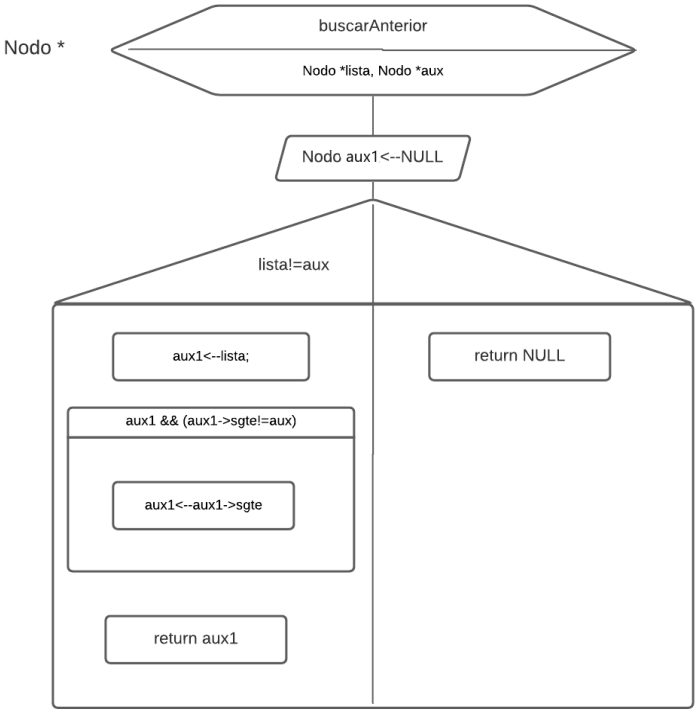

Void:
AgregarElemento2
NodoIncidente *&lista2; registroIncidente elemento;
NodoIncidente *aux <-- lista2;
aux!=NULL && aux->sgte!=NULL;
aux <-- aux->sgte;
aux != NULL
aux ->sgte <-- new NodoIncidente();
aux <-- aux->sgte;
aux->incidente <-- elemento;
aux->sgte <-- NULL;
lista2=new NodoIncidente();
lista2->Incidente <-- elemento;
lista2->sgte <-- NULL;

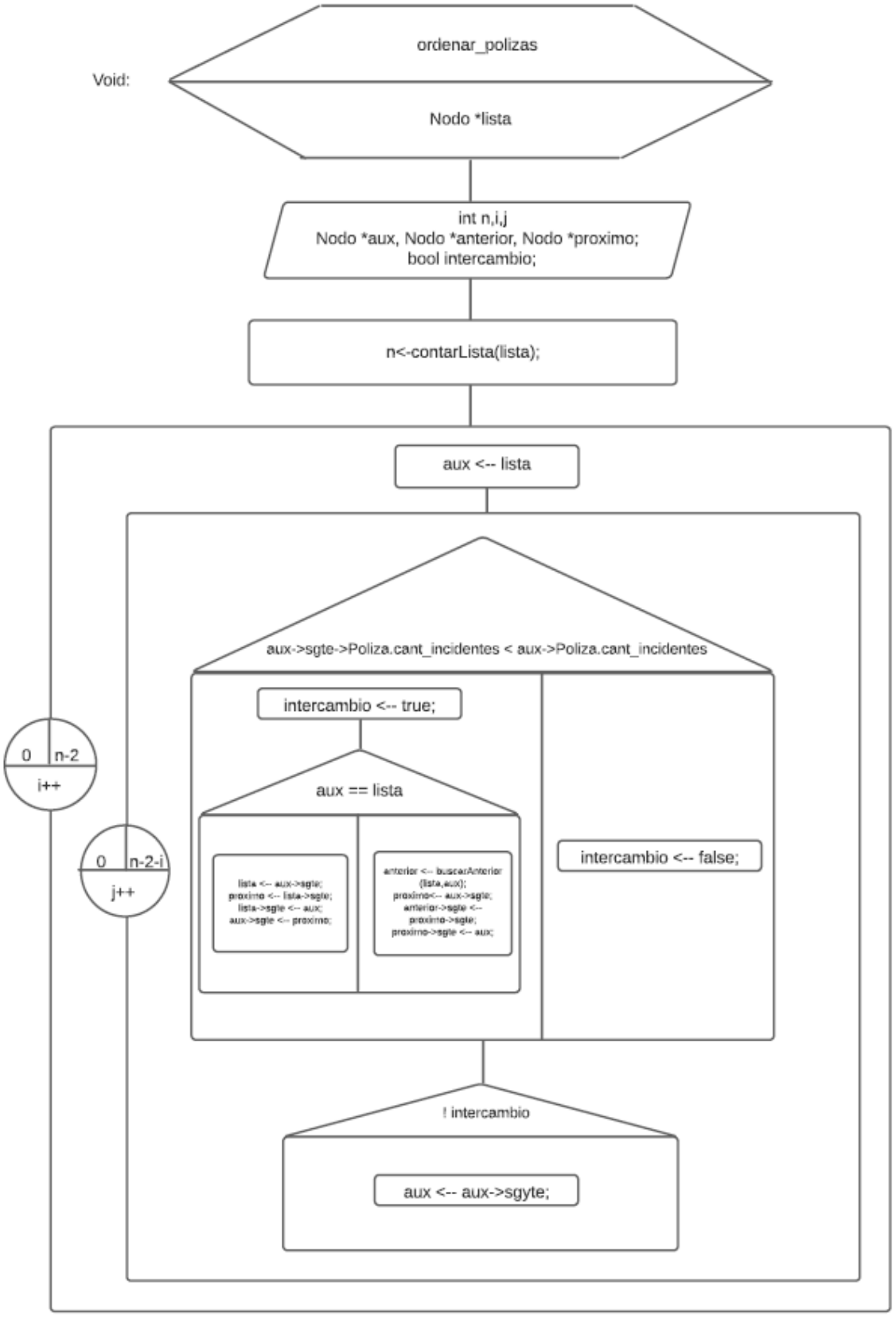
void:


agregarElemento
Nodo *&lista, registroAsegurado elemento
Nodo *aux
aux
lista
aux!=NULL && aux->sgte!=NULL
aux<--aux->sgte;
aux!=NULL
aux->sgte<--new Nodo();
aux<--aux->sgte;
aux->Poliza<--elemento;
aux->sgte<--NULL;
aux->Poliza.cant_incidentes = 0
lista<--new Nodo();
lista->Poliza<--elemento;
lista->sgte<--NULL;
aux->Poliza.cant_incidentes = 0
F

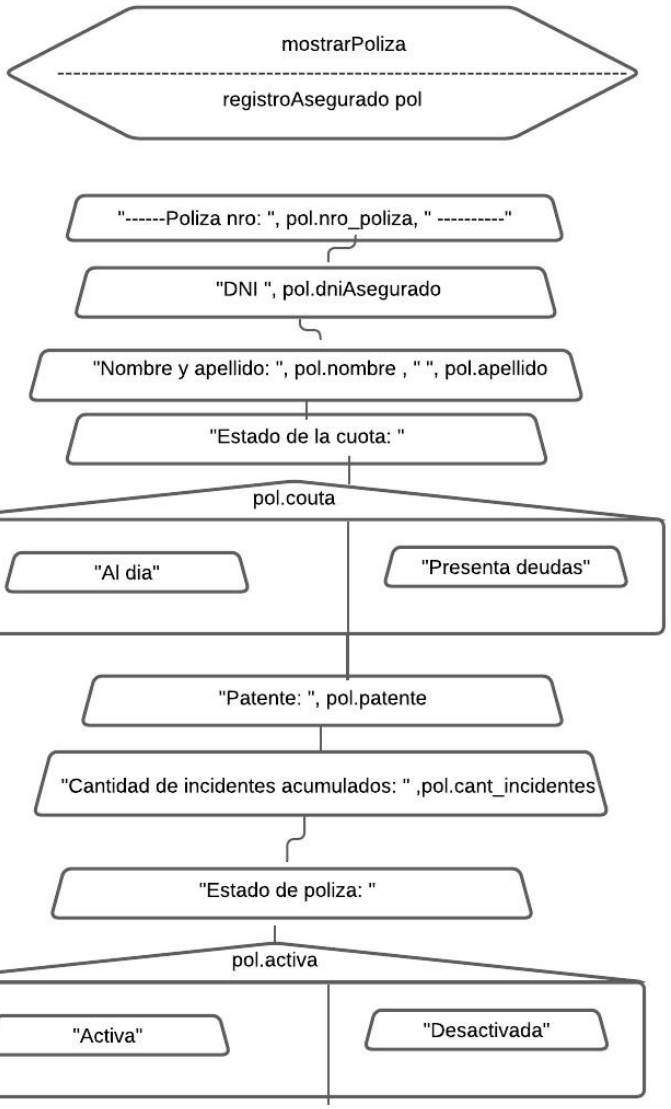

Void
levantar_cuentas
Nodo *&lista
FILE *f, *l
registroAsegurado buffer
registroAsegurado buffer 2
Nodo *aux
f=fopen(NOMBRE_ARCHIVO,"rb")
fread(&buffer,sizeof(registroAsegurado),1,f)
agregarElemento(lista,buffer);
"No se pude abrir el archivo "
l = fopen(ARCHIVO_PROCESAR,"rb")
fread(&buffer2,sizeof(registroIncidente), 1, l)
"No se pude abrir el archivo "
aux<--lista;
aux
aux->Poliza.nro_poliza == buffer2.nro_poliza
aux->Poliza.cant_incidentes ++;
AgregarElemento2(sta->sublista, buffer2);
aux <- aux->sgte;
fclose()
fclose()
Lista cargada exitosamente
F

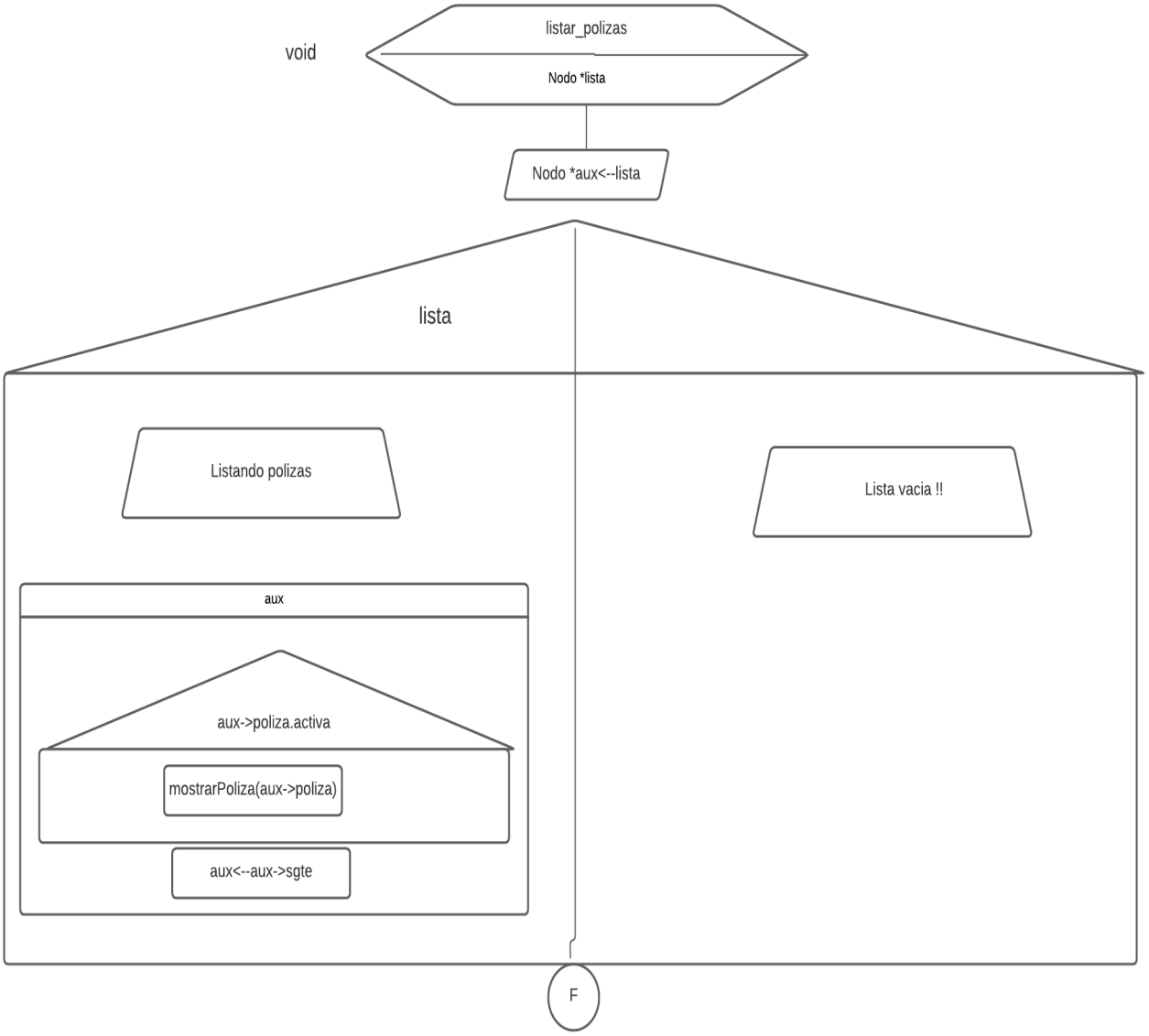

int
contarLista
Nodo *lista
Nodo *aux<--lista
int i<--0
aux
i<--i+1
aux<--aux->sgte
return i;
F

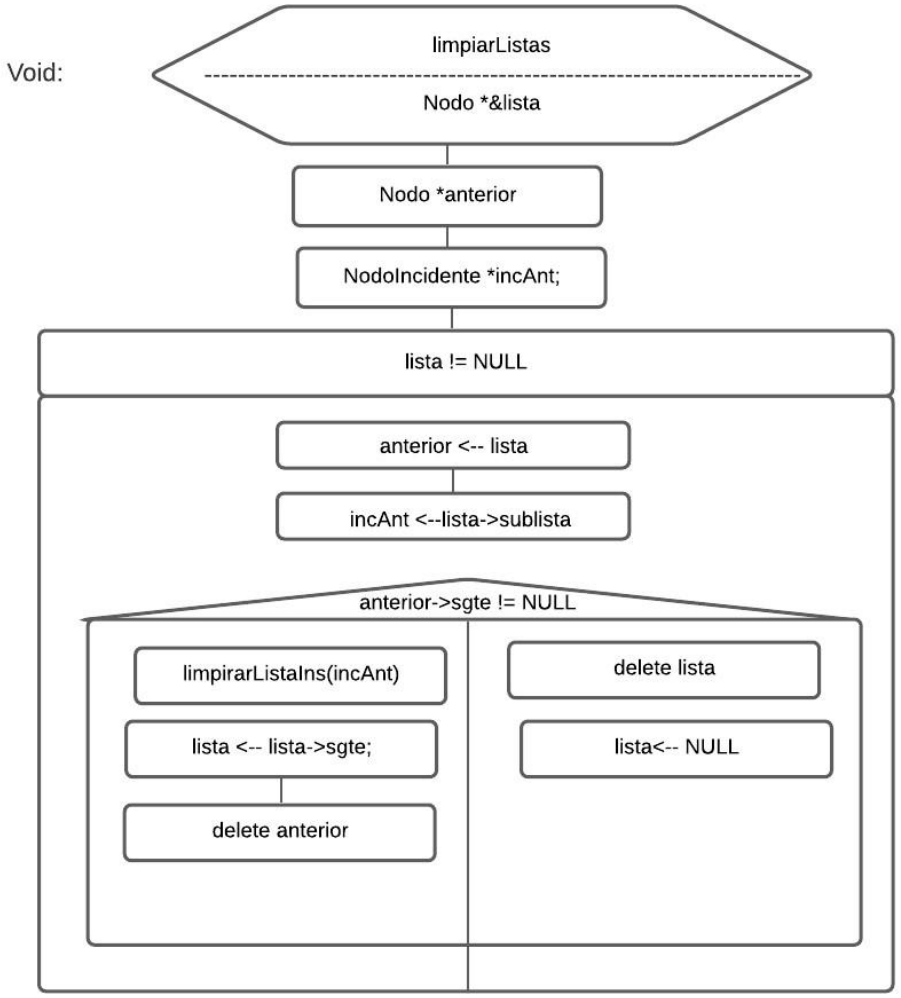

Nodo *
buscarAnterior
Nodo *lista, Nodo *aux
Nodo aux1<--NULL
lista!=aux
aux1<--lista;
return NULL
aux1 && (aux1->sgte!=aux)
aux1<--aux1->sgte
return aux1

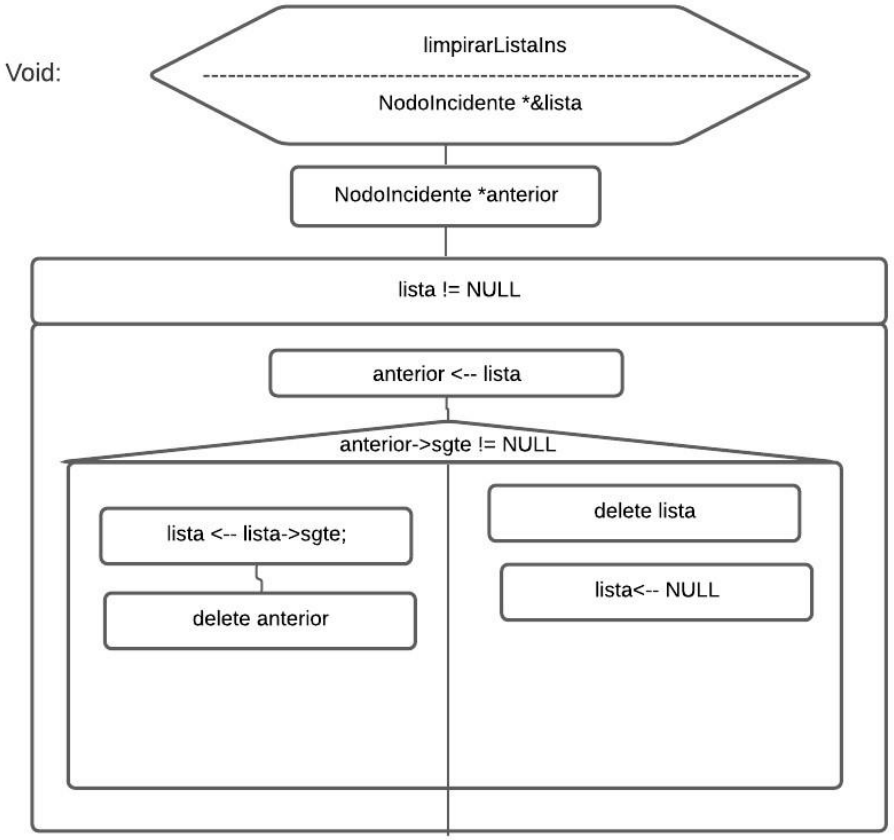

Void:
ordenar_polizas
Nodo *lista
int n,i,j
Nodo *aux, Nodo *anterior, Nodo *proximo;
bool intercambio;
n<-contarLista(lista);
aux <-- lista
0
n-2
i++
0
n-2-i
j++
aux->sgte->Poliza.cant_incidentes < aux->Poliza.cant_incidentes
intercambio <-- true;
aux == lista
lista <-- aux->sgte;
proximo <-- lista->sgte;
lista->sgte <-- aux;
aux->sgte <-- proximo;
anterior <-- buscarAnterior (lista,aux);
proximo<-- aux->sgte;
anterior->sgte <-- proximo->sgte;
proximo->sgte <-- aux;
intercambio <-- false;
! intercambio
aux <-- aux->sgyte;

Void:

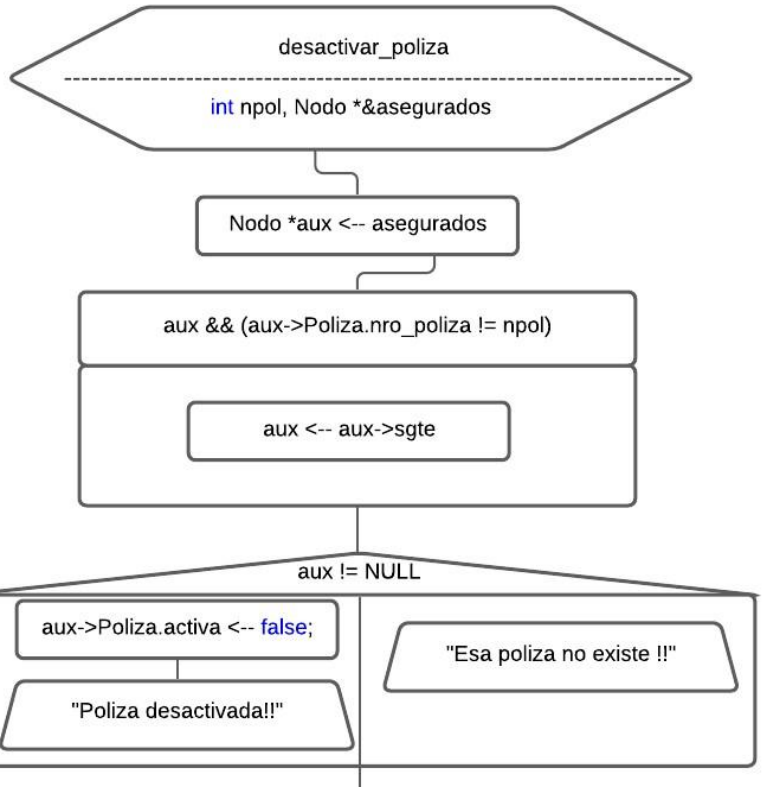

mostrarPoliza
registroAsegurado pol
"------Poliza nro: ", pol.nro_poliza, " ----------"
"DNI ", pol.dniAsegurado
"Nombre y apellido: ", pol.nombre , " ", pol.apellido
"Estado de la cuota: "
pol.couta
"Al dia"
"Presenta deudas"
"Patente: ", pol.patente
"Cantidad de incidentes acumulados: " ,pol.cant_incidentes
"Estado de poliza: "
pol.activa
"Activa"
"Desactivada"

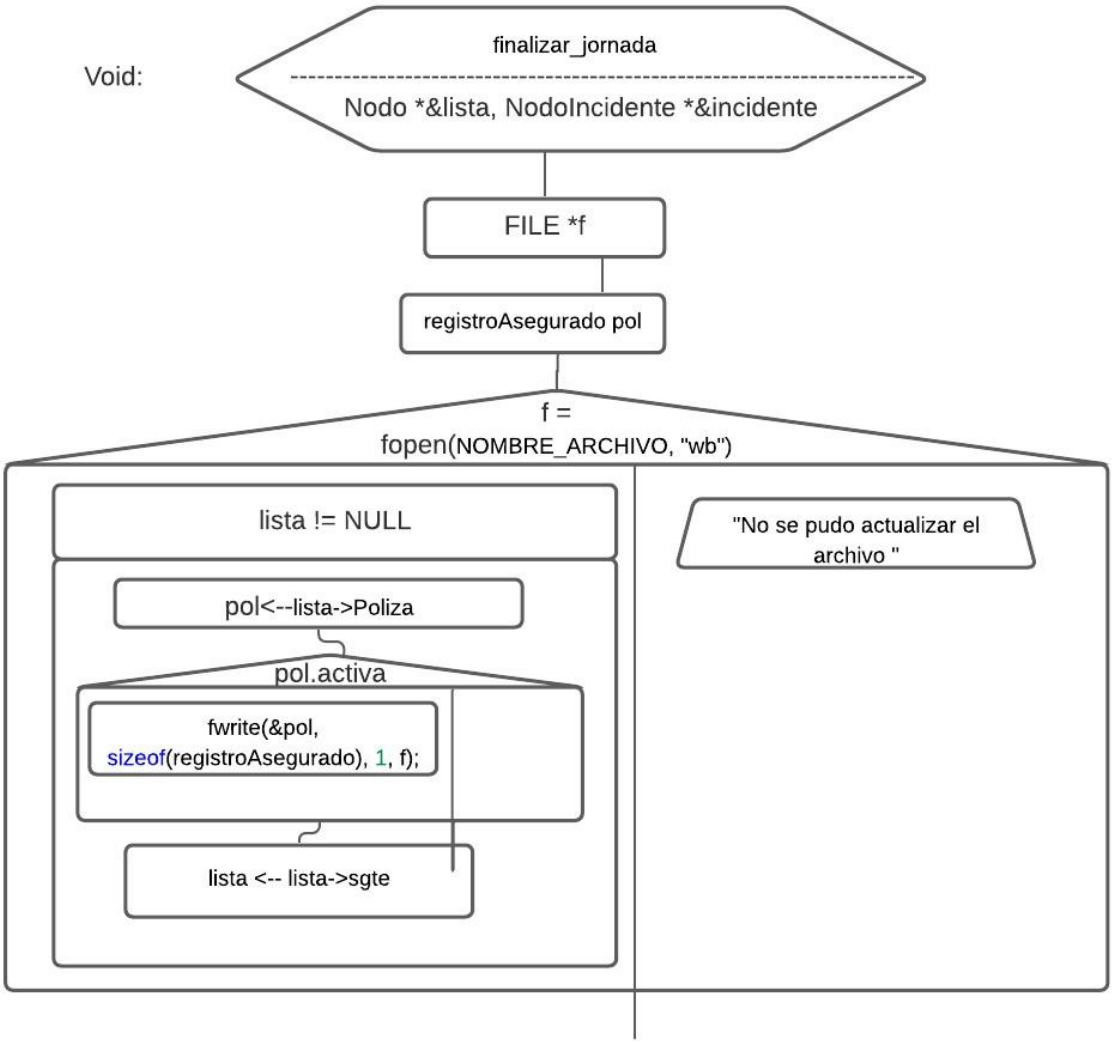

void
listar_polizas
Nodo *lista
Nodo *aux<--lista
lista
Listando polizas
Lista vacia !!
aux
aux->poliza.activa
mostrarPoliza(aux->poliza)
aux<--aux->sgte
F

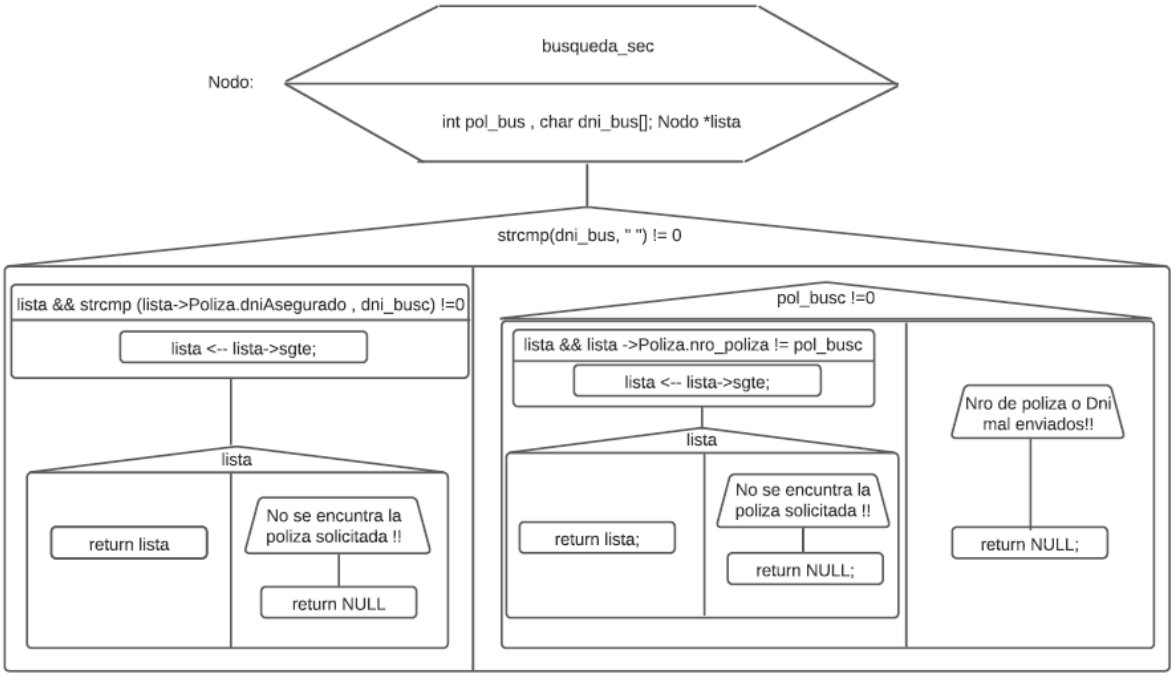

Void:
limpiarListas
Nodo *&lista
Nodo *anterior
NodoIncidente *incAnt;
lista != NULL
anterior <-- lista
incAnt <--lista->sublista
anterior->sgte != NULL
limpirarListaIns(incAnt)
lista <-- lista->sgte;
delete anterior
delete lista
lista<-- NULL

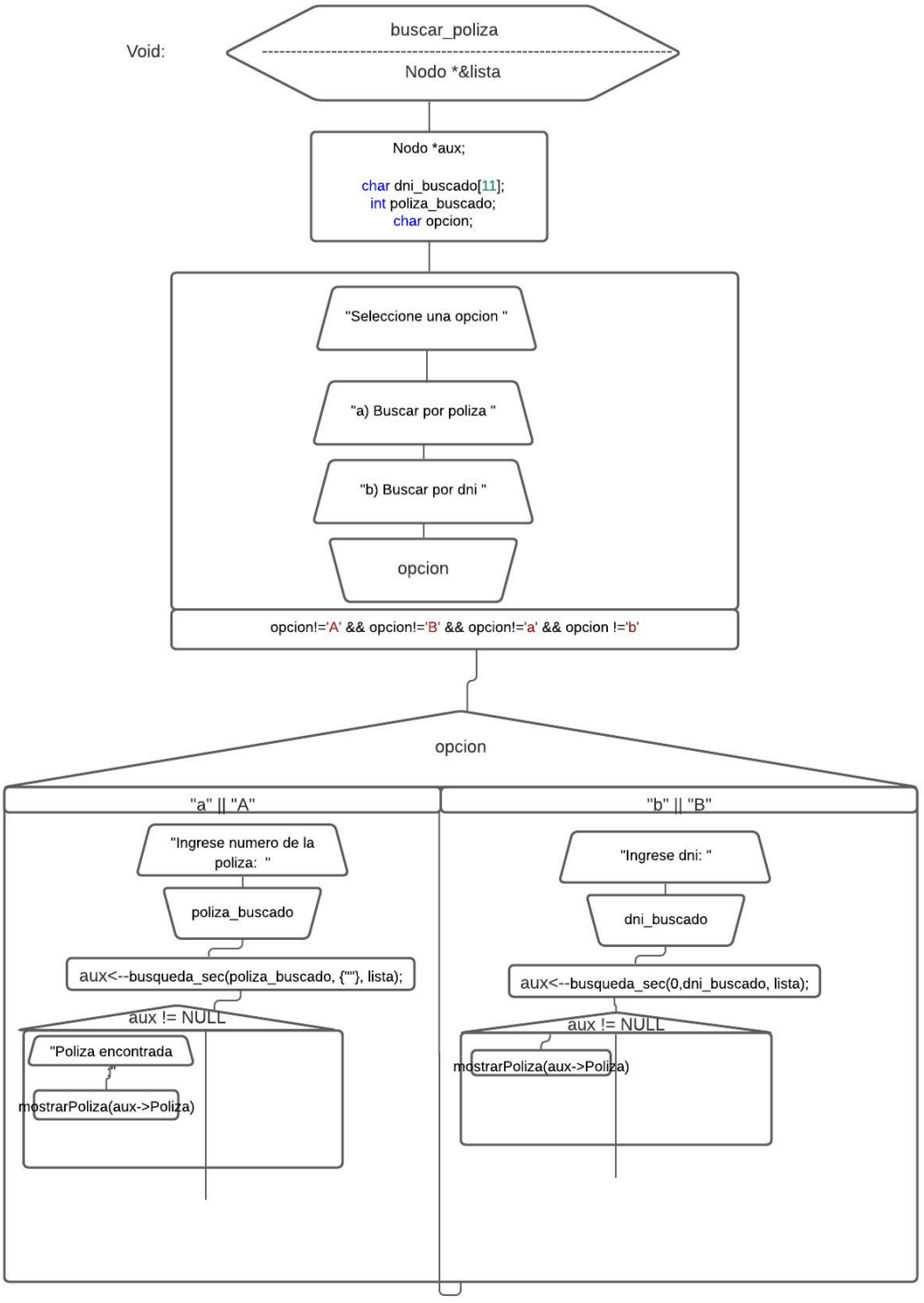

Void:
limpirarListaIns
NodoIncidente *&lista
NodoIncidente *anterior
lista != NULL
anterior <-- lista
anterior->sgte != NULL
lista <-- lista->sgte;
delete anterior
delete lista
lista<-- NULL

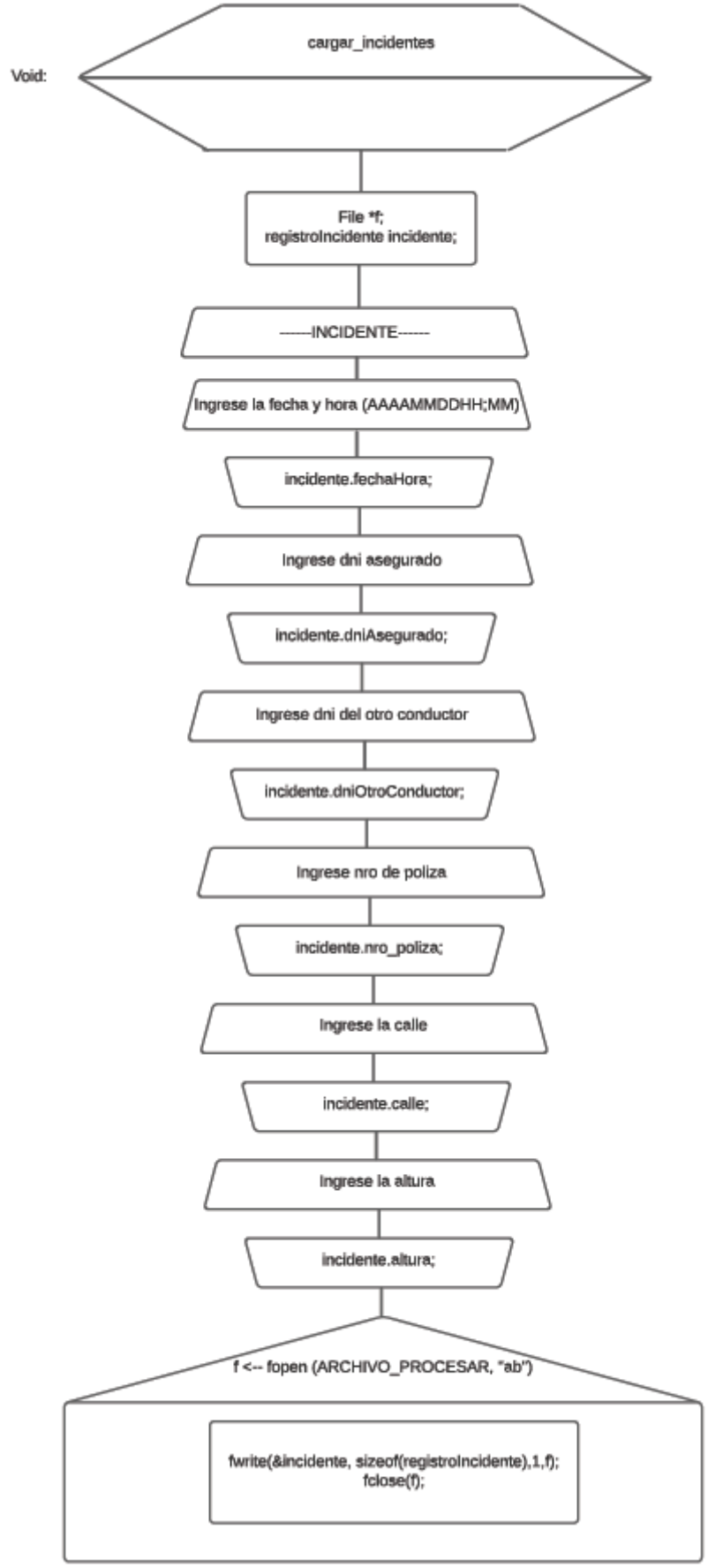
Void:

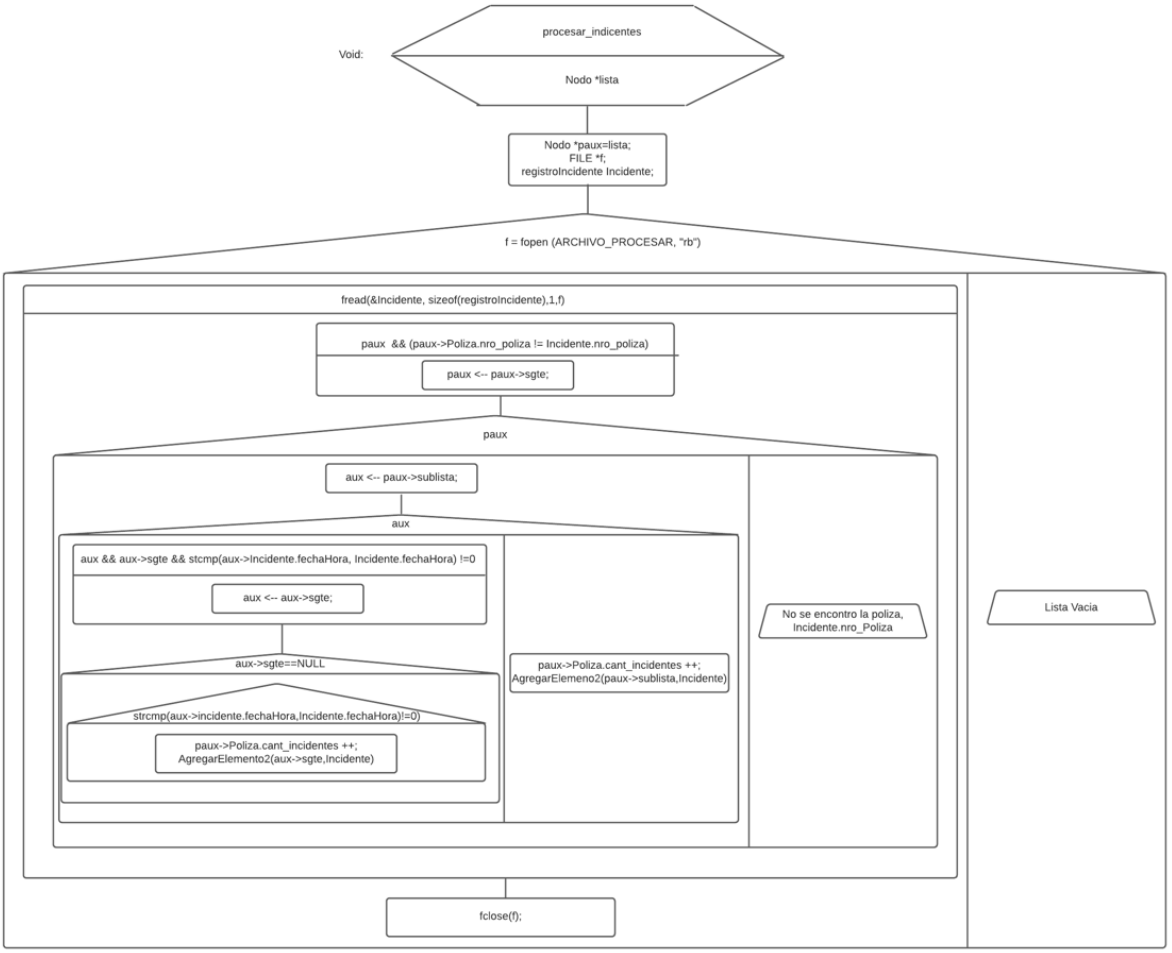

desactivar_poliza
int npol, Nodo *&asegurados
Nodo *aux <-- asegurados
aux && (aux->Poliza.nro_poliza != npol)
aux <-- aux->sgte
aux != NULL
aux->Poliza.activa <-- false;
"Poliza desactivada!!"
"Esa poliza no existe !!"

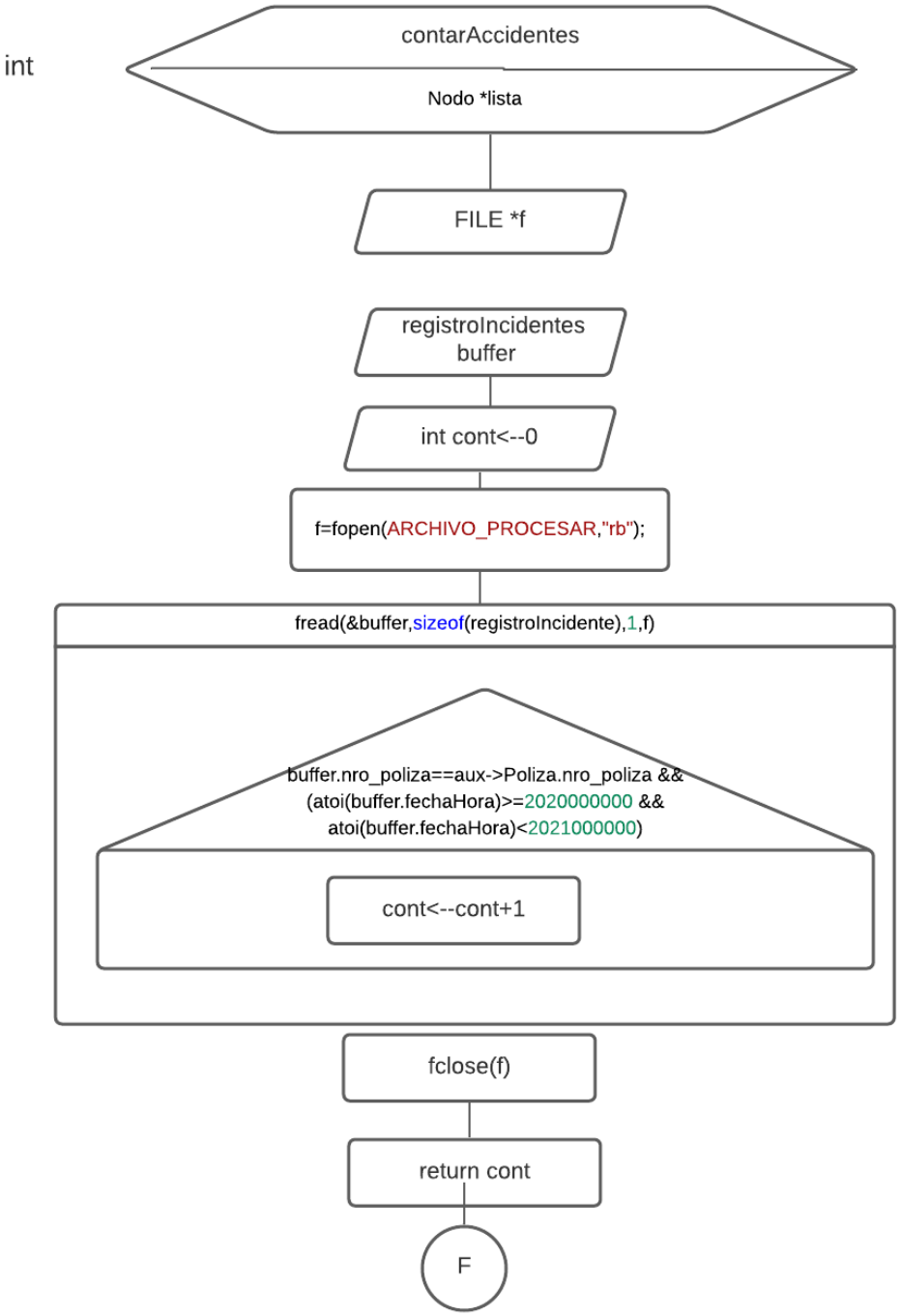

Void:
finalizar_jornada
Nodo *&lista, NodoIncidente *&incidente
FILE *f
registroAsegurado pol
f =
fopen(NOMBRE_ARCHIVO, "wb")
lista != NULL
"No se pudo actualizar el archivo "
pol<--lista->Poliza
pol.activa
fwrite(&pol,
sizeof(registroAsegurado), 1, f);
lista <-- lista->sgte

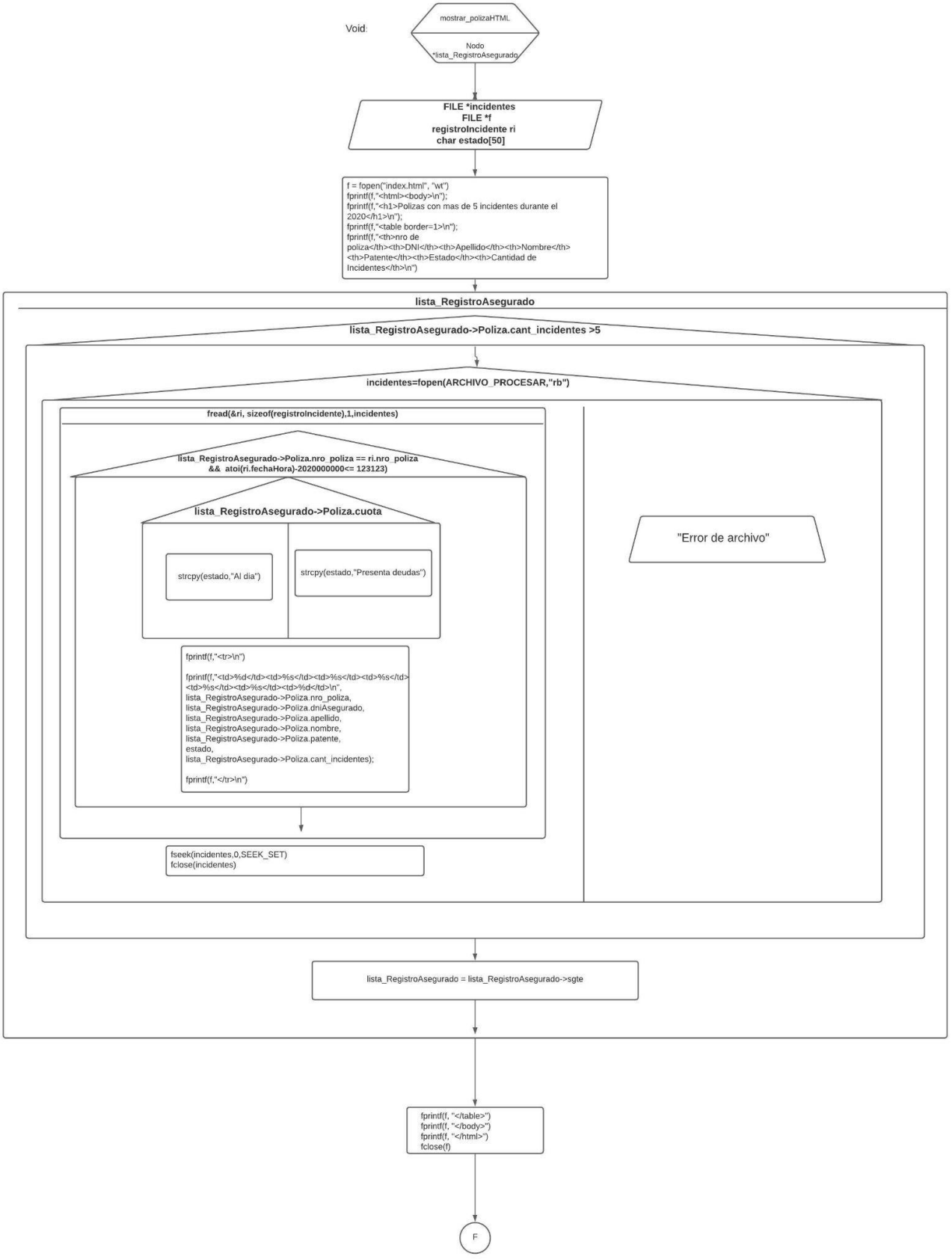

Nodo:
busqueda_sec
int pol_bus , char dni_bus[]; Nodo *lista
strcmp(dni_bus, " ") != 0
lista && strcmp (lista->Poliza.dniAsegurado , dni_busc) !=0
lista <-- lista->sgte;
lista
return lista
No se encuntra la poliza solicitada !!
return NULL
pol_busc !=0
lista && lista ->Poliza.nro_poliza != pol_busc
lista <-- lista->sgte;
lista
return lista;
No se encuntra la poliza solicitada !!
return NULL;
Nro de poliza o Dni mal enviados!!
return NULL;

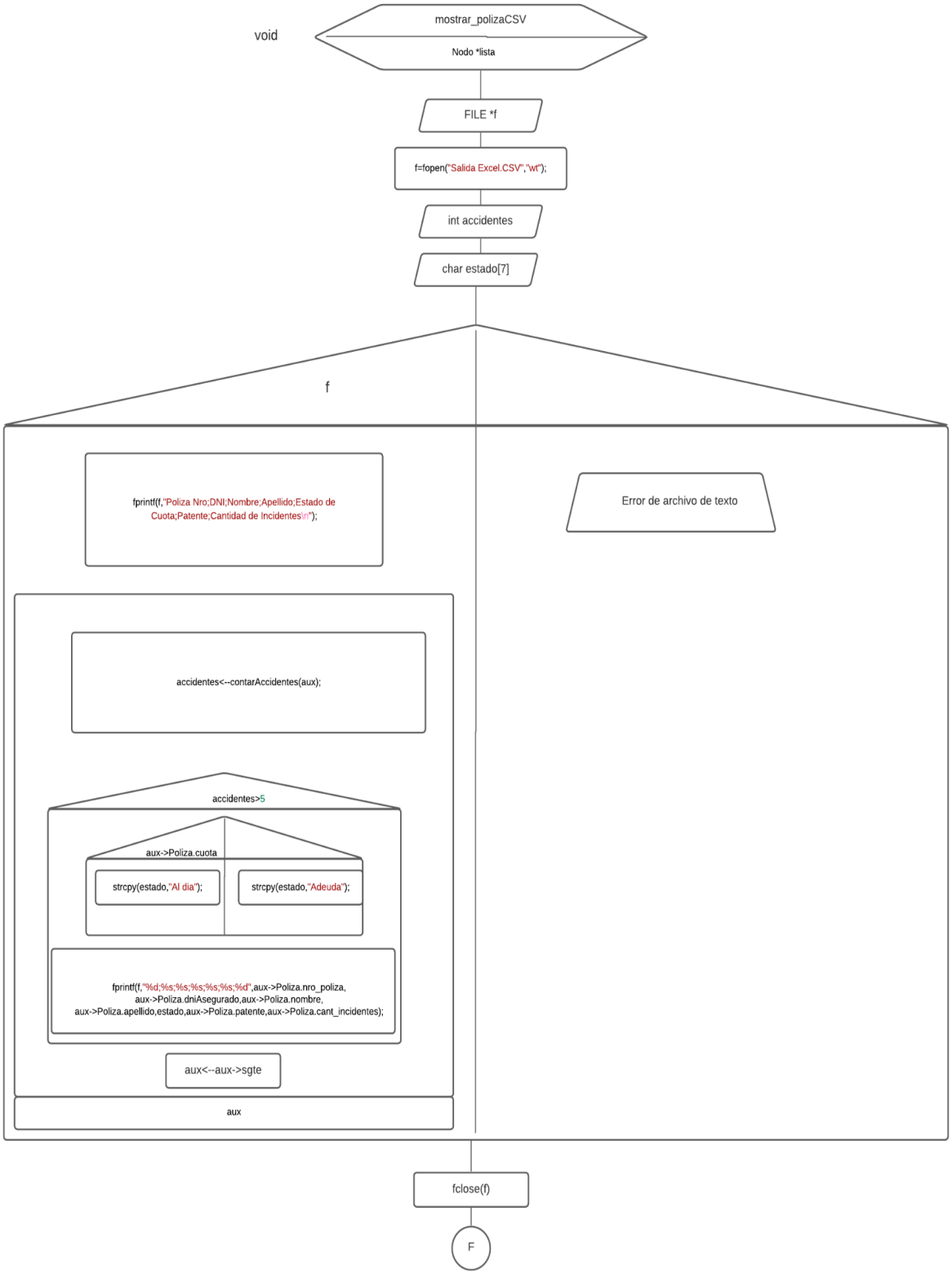

Void:
buscar_poliza
Nodo *&lista
Nodo *aux;
char dni_buscado[11];
int poliza_buscado;
char opcion;
"Seleccione una opcion "
"a) Buscar por poliza "
"b) Buscar por dni "
opcion
opcion!='A' && opcion!='B' && opcion!='a' && opcion !='b'
opcion
"a" || "A"
"Ingrese numero de la poliza: "
poliza_buscado
aux<--busqueda_sec(poliza_buscado, {""}, lista);
aux != NULL
"Poliza encontrada
mostrarPoliza(aux->Poliza)
"b" || "B"
"Ingrese dni: "
dni_buscado
aux<--busqueda_sec(0,dni_buscado, lista);
aux != NULL
mostrarPoliza(aux->Poliza)

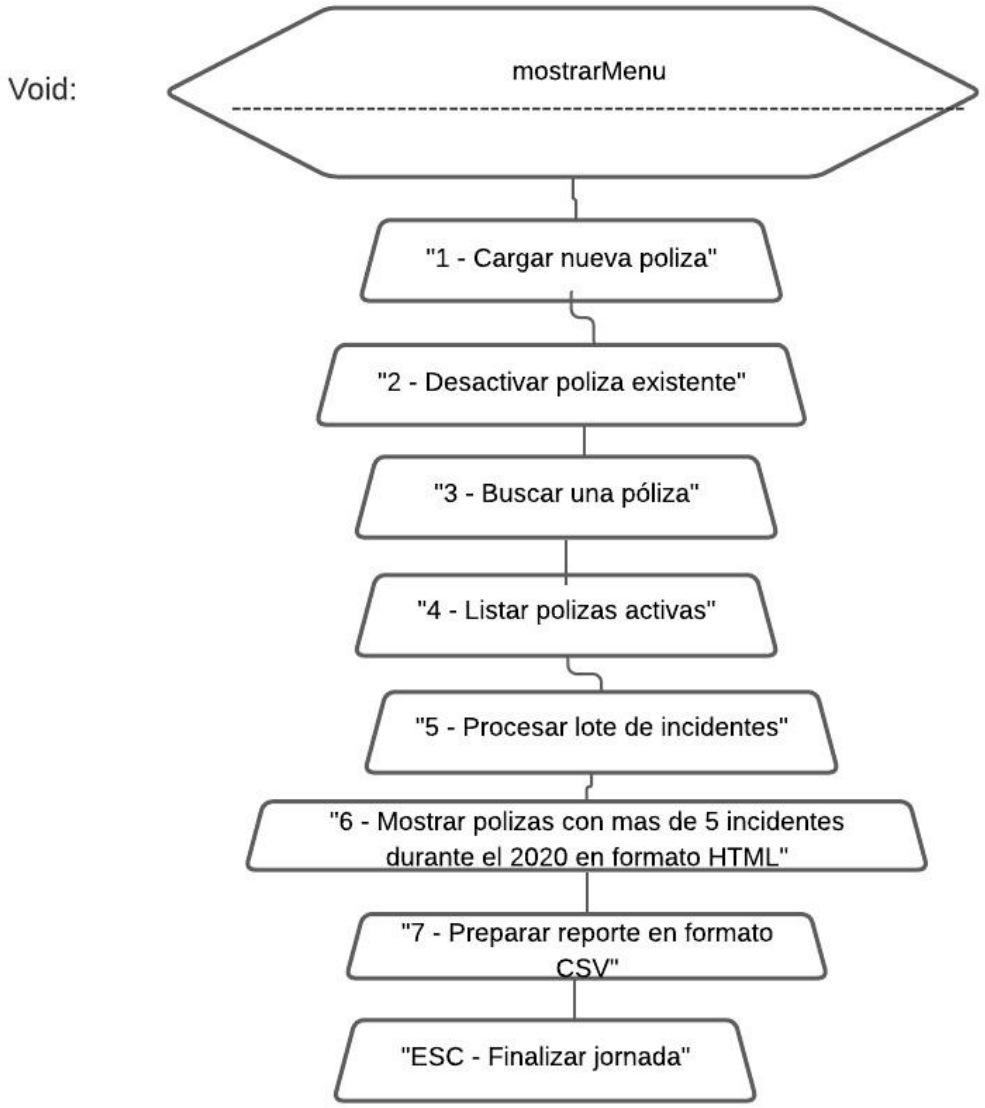

Void:
cargar_incidentes
File *f;
registroIncidente incidente;
------INCIDENTE------
Ingrese la fecha y hora (AAAAMMDDHH;MM)
incidente.fechaHora;
Ingrese dni asegurado
incidente.dniAsegurado;
Ingrese dni del otro conductor
incidente.dniOtroConductor;
Ingrese nro de poliza
incidente.nro_poliza;
Ingrese la calle
incidente.calle;
Ingrese la altura
incidente.altura;
f <-- fopen (ARCHIVO_PROCESAR, "ab")
fwrite(&incidente, sizeof(registroIncidente),1,f);
fclose(f);


Void:
procesar_indicentes
Nodo *lista
Nodo *paux=lista;
FILE *f;
registroIncidente Incidente;
f = fopen (ARCHIVO_PROCESAR, "rb")
fread(&Incidente, sizeof(registroIncidente),1,f)
paux && (paux->Poliza.nro_poliza != Incidente.nro_poliza)
paux <-- paux->sgte;
paux
aux <-- paux->sublista;
aux
aux && aux->sgte && stcmp(aux->Incidente.fechaHora, Incidente.fechaHora) !=0
aux <-- aux->sgte;
aux->sgte==NULL
strcmp(aux->incidente.fechaHora,Incidente.fechaHora)!=0)
paux->Poliza.cant_incidentes ++;
AgregarElemento2(paux->sgte,Incidente)
paux->Poliza.cant_incidentes ++;
AgregarElemeno2(paux->sublista,Incidente)
No se encontro la poliza,
Incidente.nro_Poliza
Lista Vacia
fclose(f);


int
contarAccidentes
Nodo *lista
FILE *f
registroIncidentes
buffer
int cont<--0
f=fopen(ARCHIVO_PROCESAR,"rb");
fread(&buffer,sizeof(registroIncidente),1,f)
buffer.nro_poliza==aux->Poliza.nro_poliza &&
(atoi(buffer.fechaHora)>=2020000000 &&
atoi(buffer.fechaHora)<2021000000)
cont<--cont+1
fclose(f)
return cont
F


Void:
mostrar_polizaHTML
Nodo *lista_RegistroAsegurado
FILE *incidentes
FILE *f
registroIncidente ri
char estado[50]
f = fopen("index.html", "wt")
fprintf(f,"<html><body>\n");
fprintf(f,"<h1>Polizas con mas de 5 incidentes durante el 2020</h1>\n");
fprintf(f,"<table border=1>\n");
fprintf(f,"<th>nro de poliza</th><th>DNI</th><th>Apellido</th><th>Nombre</th><th>Patente</th><th>Estado</th><th>Cantidad de Incidentes</th>\n")
lista_RegistroAsegurado
lista_RegistroAsegurado->Poliza.cant_incidentes >5
incidentes=fopen(ARCHIVO_PROCESAR,"rb")
fread(&ri, sizeof(registroIncidente),1,incidentes)
lista_RegistroAsegurado->Poliza.nro_poliza == ri.nro_poliza
&& atoi(ri.fechaHora)-2020000000<= 123123)
lista_RegistroAsegurado->Poliza.cuota
strcpy(estado,"Al dia")
strcpy(estado,"Presenta deudas")
"Error de archivo"
fprintf(f,"<tr>\n")
fprintf(f,"<td>%d</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%s</td><td>%d</td>\n",
lista_RegistroAsegurado->Poliza.nro_poliza,
lista_RegistroAsegurado->Poliza.dniAsegurado,
lista_RegistroAsegurado->Poliza.apellido,
lista_RegistroAsegurado->Poliza.nombre,
lista_RegistroAsegurado->Poliza.patente,
estado,
lista_RegistroAsegurado->Poliza.cant_incidentes);
fprintf(f,"</tr>\n")
fseek(incidentes,0,SEEK_SET)
fclose(incidentes)
lista_RegistroAsegurado = lista_RegistroAsegurado->sgte
fprintf(f, "</table>")
fprintf(f, "</body>")
fprintf(f, "</html>")
fclose(f)
F


void
mostrar_polizaCSV
Nodo *lista
FILE *f
f=fopen("Salida Excel.CSV","wt");
int accidentes
char estado[7]
f
fprintf(f,"Poliza Nro;DNI;Nombre;Apellido;Estado de Cuota;Patente;Cantidad de Incidentes\n");
Error de archivo de texto
accidentes<--contarAccidentes(aux);
accidentes>5
aux->Poliza.cuota
strcpy(estado,"Al dia");
strcpy(estado,"Adeuda");
fprintf(f,"%d;%s;%s;%s;%s;%s;%d",aux->Poliza.nro_poliza,
aux->Poliza.dniAsegurado,aux->Poliza.nombre,
aux->Poliza.apellido,estado,aux->Poliza.patente,aux->Poliza.cant_incidentes);
aux<--aux->sgte
aux
fclose(f)
F



Código fuente

Repositorio en github: https://github.com/Fcallero/Tp2-AyED.git
versión: 1.0.0

casos de validación: