

Actividad 01 (Repaso de Programación)

Jose Eduardo Silva Canizales

Seminario de solución de problemas de algoritmia

Lineamientos de evaluación

- El reporte está en formato Google Docs o PDF.
- El reporte sigue las pautas del [Formato de Actividades](#) .
- Se muestra código y captura de pantalla para agregar un objeto paquete dentro de la lista de la clase Paqueteria.
- Se muestra código y captura de pantalla para eliminar un objeto de la lista de paquete en la clase Paqueteria.
- Se muestra código y captura de pantalla para mostrar la información de toda la lista de paquete en la clase Paqueteria.
- Se muestra código y captura de pantalla para guardar la lista de paquete en un archivo .txt.
- Se muestra captura del pantalla del contenido del archivo .txt.
- Se muestra código y captura de pantalla de la opción Recuperar.

Desarrollo

Función agregar paquete

```
void Paqueteria::Insertar_Al_Inicio(Paquete paquete){
    Nodo *tmp = new Nodo();
    tmp->elemento = paquete;

    if (h == NULL) {
        h = tmp;
    } else {
        tmp->sig = h;
        h = tmp;
    }
}
```

```
Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 2_
```

```
Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 2
Id: 11
origen: zapopan
destino: guadalajara
peso: 23
Presione una tecla para continuar . . .
```

```
Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 2
Id: 12
origen: zacatecas
destino: jalisco
peso: 67
Presione una tecla para continuar . . .
```

Función Mostrar

```
void Paqueteria::Mostrar() {
    Nodo *aux;
    aux = h;
    if (aux == NULL)
    {
        cout<<"Lista Vacía"<<endl;
    }
    while (aux != NULL)
    {
        aux->elemento.imprimir();
        aux = aux->sig;
    }
}
```

```
Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 4_
```

```
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 4

Id: 12
origen: zacatecas
destino: jalisco
peso: 67

Id: 11
origen: zapopan
destino: guadalajara
peso: 23
Presione una tecla para continuar . . . _
```

Función Eliminar al Inicio

```
66 void Paqueteria::Eliminar_Al_Inicio()
67 {
68
69     Nodo *tmp=h->sig;
70     delete h;
71
72     h=tmp;
73
74 }
75
```

```

Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir

Opción: 3_

```

```

Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir

Opción: 3
Presione una tecla para continuar . . .

```

Resultado de la primera eliminación

```

Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir

Opción: 4

Id: 11
origen: zapopan
destino: guadalajara
peso: 23

Presione una tecla para continuar . . .

```

Función Guardar

```

76 void Paqueteria::Guardar()
77 {
78     ofstream archivo("file01.txt", ios::out);
79     if(archivo.is_open()){
80         Nodo *temp = h;
81         while(temp != NULL){
82             archivo<<temp->elemento.format();
83             temp=temp->sig;
84         }
85     }
86     archivo.close();
87 }
88

```

```

Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir

Opción: 5_

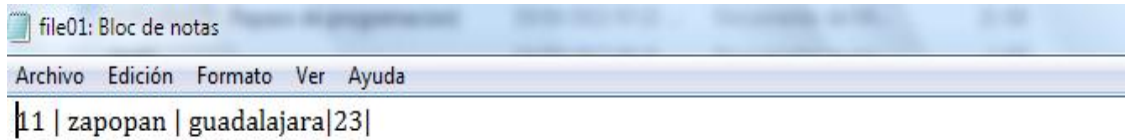
```

```

Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 5
Datos Guardados correctamente
Presione una tecla para continuar . . .

```

Archivo generado



file01: Bloc de notas

Archivo Edición Formato Ver Ayuda

11 | zapopan | guadalajara|23|

Función Recuperar

```

89 void Paqueteria::Recuperar()
90 {
91
92     ifstream archivo;
93     string texto;
94
95     archivo.open("file01.txt", ios::in);
96
97     if(archivo.fail())
98     {
99         cout<<"No se pudo abrir el archivo";
100        exit(1);
101    }
102    while(!archivo.eof())
103    {
104        getline(archivo, texto);
105        cout<<texto<<endl;
106    }
107
108    archivo.close();
109 }

```

```

Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 6

```

```

Actividad 01
1.- Inicializar
2.- Agregar paquete
3.- Eliminar al Inicio
4.- Mostrar
5.- Guardar
6.- Recuperar
7.- Salir
Opción: 6
11 | zapopan | guadalajara|23|
Datos Cargados correctamente
Presione una tecla para continuar . . .

```

Conclusiones

Problemas con los que te enfrentaste

El primer problema la que me enfrente fue que no sabía cómo insertar múltiples atributos en una lista simple enlazada; el segundo problema al que me enfrente fue que no sabía cómo hacer que se leyera un archivo desde el código compilado.

Que aprendiste

Aprendí hacer una lista simplemente ligada con múltiples atributos u objetos, por otra parte aprendí a que el código compilado mostrara o leyera el archivo previamente creado.

Que no entendiste desde el inicio y como hiciste para entenderlo

El cómo insertar múltiples atributos en una lista simple enlazada y mostrar los elementos de un archivo desde el código compilado, lo que hice fue ver varios videos para tener más claro el cómo se hace.

Referencias

Primera referencia

Url: <https://www.youtube.com/watch?v=g3gtAhHgYcA>

Título: Crear y Escribir un archivo es C++ (respaldar)

Autor: MICHEL DAVALOS BOITES

Segunda referencia

Url: <https://www.youtube.com/watch?v=QqU8emKoiIA>

Título: Leer un archivo en C++ (recuperar)

Autor: MICHEL DAVALOS BOITES

Tercera referencia

Url: <https://www.youtube.com/watch?v=ksnBUo-08Uw>

Título: 124. Programación en C++ || Archivos || Lectura de un archivo de texto

Autor: Programación ATS

Cuarta referencia

Url: <https://www.youtube.com/watch?v=ijMvMOIInXo>

Título: Lista enlazada simple C++ | Parte 1 | Agregar nodo al Inicio | Agregar al final | Agregar por indice

Autor: Código 404

Código

main.cpp

```
#include <iostream>
#include <stdlib.h>
#include <stdio.h>
#include <fstream>
#include <string.h>
#include "Paqueteria.h"

using namespace std;

int main()
{
    Paqueteria *listasimple = new Paqueteria;
    Paquete paque;
    int opc, tamano;
    string id, origen, destino, peso;
    bool vacio;
    Nodo *primero;
    Nodo *ultimo;
    Nodo *siguiente;
    Nodo *anterior;
    do
    {
        cout<<"\tActividad 01"<<endl<<endl;
        cout<<"1.- Inicializar"<<endl;
        cout<<"2.- Agregar paquete"<<endl;
        cout<<"3.- Eliminar al Inicio"<<endl;
        cout<<"4.- Mostrar"<<endl;
        cout<<"5.- Guardar"<<endl;
        cout<<"6.- Recuperar"<<endl;
        cout<<"7.- Salir"<<endl<<endl;
        cout<<"Opci\xA2n: ";
        cin>>opc;
        cout<<endl;

        switch(opc)
        {
            case 1:
                listasimple->Inicializar();
                cout<<"\tLista Inicializada con Exito!"<<endl;
                cout<<endl;
                system("pause");
                system("cls");
                break;
            case 2:
                cout<<"Id: ";
                fflush(stdin);
                getline(cin, id);

                cout<<"origen: ";
                fflush(stdin);
                getline(cin, origen);

                cout<<"destino: ";
                fflush(stdin);
                getline(cin, destino);
```



```

        cout<<"peso: ";
        fflush(stdin);
        getline(cin, peso);

        paque.setOrigen(origen);
        paque.setId(id);
        paque.setDestino(destino);
        paque.setPeso(peso);

        listasimple->Insertar_Al_Inicio(paque);
        cout<<endl;
        system("pause");
        system("cls");
        break;

    case 3:
        listasimple->Eliminar_Al_Inicio();
        system("pause");
        system("cls");
        break;

    case 4:
        listasimple->Mostrar();
        cout<<endl;
        system("pause");
        system("cls");
        break;

    case 5:
        listasimple->Guardar();
        cout<<"Datos Guardados correctamente"<<endl;
        system("pause");
        system("cls");
        break;

    case 6:
        listasimple->Recuperar();
        cout<<"Datos Cargados correctamente"<<endl;
        system("pause");
        system("cls");
        break;

    case 7:
        break;

    default:
        cout<<"Opci\xA2n invalida\n"<<endl;
        cout<<endl;
        system("pause");
        system("cls");
        break;
    }
}
while(opc != 7);

return 0;
}

```

Paquete.h

```

#ifndef PAQUETE_H
#define PAQUETE_H
#include <stdlib.h>

```

```

#include <fstream>
#include <string.h>
#include <iostream>

using namespace std;

class Paquete
{
public:
    Paquete();
    Paquete(string, string, string);
    ~Paquete();

    void imprimir() {
        cout<<"\n Id: "<<id<<" \n origen: "<<origen<<" \n destino: "<<destino<<" \n peso:
"<<peso<<endl<<endl;
    }

    string format() {
        string datos;
        datos = id+" | " +origen+" | " +destino+"|" +peso+"|";
        return datos;
    }

    string getId(void) {return id;};
    void setId(string num){this->id = num;};

    string getOrigen(void){ return origen;};
    void setOrigen (string orig){origen = orig;};

    string getDestino(void){ return destino;};
    void setDestino (string des){destino = des;};

    string getPeso(void){ return peso;};
    void setPeso (string pe){peso = pe;};

private:
    string id;
    string origen;
    string destino;
    string peso;
};

Paquete::Paquete(string num, string orig, string des){
    id = num;
    origen = orig;
    destino=des;
}

Paquete::Paquete()
{
    id = "";
    origen = "";
    destino = "";
    peso = "";
}

```

```

Paquete::~~Paquete()
{
    //dtor
}

#endif

```

Nodo.h

```

#ifndef NODO_H
#define NODO_H
#include "Paquete.h"

class Nodo
{
public:
    Nodo() {sig = NULL; };
    ~Nodo();

    Paquete elemento;
    Nodo *sig;

private:
};

Nodo::~~Nodo()
{
    //dtor
}

#endif

```

Paqueteria.h

```

#ifndef PAQUETERIA_H
#define PAQUETERIA_H
#include<iostream>
#include "Nodo.h"
#include <stdlib.h>
#include <stdio.h>
#include <fstream>
#include <string.h>

class Paqueteria
{
public:
    Paqueteria();
    ~Paqueteria();

    void Inicializar();

```

```

        void Insertar_Al_Inicio(Paquete);
        void Mostrar();
        void Eliminar_Al_Inicio();
        void Guardar();
        void Recuperar();
protected:
    Nodo *h;

};

Paqueteria::Paqueteria()
{
    h = NULL;
}

void Paqueteria::Inicializar()
{
    h = NULL;
}

void Paqueteria::Insertar_Al_Inicio(Paquete paquete){
    Nodo *tmp = new Nodo();
    tmp->elemento = paquete;

    if (h == NULL) {
        h = tmp;
    } else {
        tmp->sig = h;
        h = tmp;
    }
}

void Paqueteria::Mostrar(){
    Nodo *aux;
    aux = h;
    if(aux == NULL)
    {
        cout<<"Lista Vacia"<<endl;
    }
    while(aux != NULL)
    {
        aux->elemento.imprimir();
        aux = aux->sig;
    }
}

void Paqueteria::Eliminar_Al_Inicio()
{
    Nodo *tmp=h->sig;
    delete h;

    h=tmp;
}

void Paqueteria::Guardar()
{
    ofstream archivo("file01.txt", ios::out);
    if(archivo.is_open()){
        Nodo *temp = h;
        while(temp != NULL){

```

```

        archivo<<temp->elemento.format();
        temp=temp->sig;
    }
    }
    archivo.close();
}

void Paqueteria::Recuperar()
{
    ifstream archivo;
    string texto;

    archivo.open("file01.txt",ios::in);

    if(archivo.fail())
    {
        cout<<"No se pudo abrir el archivo";
        exit(1);
    }
    while(!archivo.eof())
    {
        getline(archivo,texto);
        cout<<texto<<endl;
    }

    archivo.close();
}

Paqueteria::~~Paqueteria()
{
}

#endif

```