

Actividad 01 – Repaso de Programación

Alcaraz Valdivia Marcos Fernando

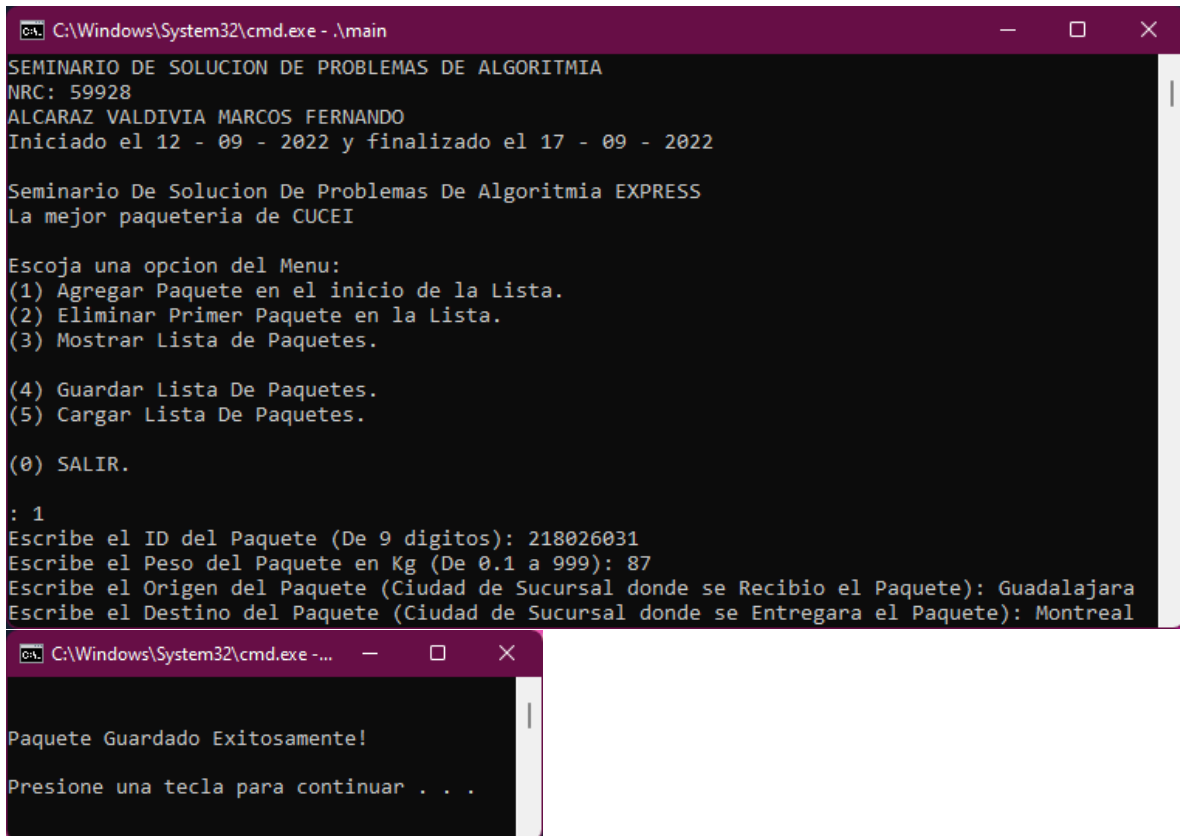
Seminario de Solución de Problemas de Algoritmia

Lineamientos de evaluación

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

Desarrollo

Captura de Pantalla y Código para Agregar Paquete a la lista Paquetería:



```
C:\Windows\System32\cmd.exe - .\main
SEMINARIO DE SOLUCION DE PROBLEMAS DE ALGORITMIA
NRC: 59928
ALCARAZ VALDIVIA MARCOS FERNANDO
Iniciado el 12 - 09 - 2022 y finalizado el 17 - 09 - 2022

Seminario De Solucion De Problemas De Algoritmia EXPRESS
La mejor paqueteria de CUCEI

Escoja una opcion del Menu:
(1) Agregar Paquete en el inicio de la Lista.
(2) Eliminar Primer Paquete en la Lista.
(3) Mostrar Lista de Paquetes.

(4) Guardar Lista De Paquetes.
(5) Cargar Lista De Paquetes.

(0) SALIR.

: 1
Escribe el ID del Paquete (De 9 digitos): 218026031
Escribe el Peso del Paquete en Kg (De 0.1 a 999): 87
Escribe el Origen del Paquete (Ciudad de Sucursal donde se Recibio el Paquete): Guadalajara
Escribe el Destino del Paquete (Ciudad de Sucursal donde se Entregara el Paquete): Montreal

C:\Windows\System32\cmd.exe - ...
Paquete Guardado Exitosamente!
Presione una tecla para continuar . . .
```

Demanda de datos

```
Paquete *nuevoPaquete = nullptr;
nuevoPaquete = new Paquete();

cout << "Escribe el ID del Paquete (De 9 digitos): ";
cin >> dato;
nuevoPaquete->setId(dato);

cout << "Escribe el Peso del Paquete en Kg (De 0.1 a 999): ";
cin >> dato;
nuevoPaquete->setPeso(dato);

cout << "Escribe el Origen del Paquete (Ciudad de Sucursal donde
se Recibio el Paquete): ";
```

```

        cin >> dato;
        nuevoPaquete->setOrigen(dato);

        cout << "Escribe el Destino del Paquete (Ciudad de Sucursal
donde se Entregara el Paquete): ";
        cin >> dato;
        nuevoPaquete->setDestino(dato);

        lista.insertarAlInicio(*nuevoPaquete);

        system("cls");
        cout << "\n\nPaquete Guardado Exitosamente!\n\n";
        break;

```

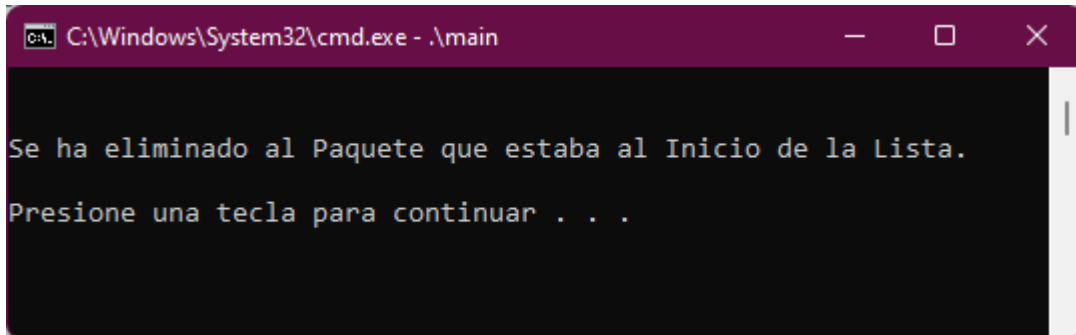
Insertión al inicio de la Lista

```

void Paqueteria::insertarAlInicio(Paquete &x)
{
    x.setSiguiente(anchor);
    anchor = &x;
}

```

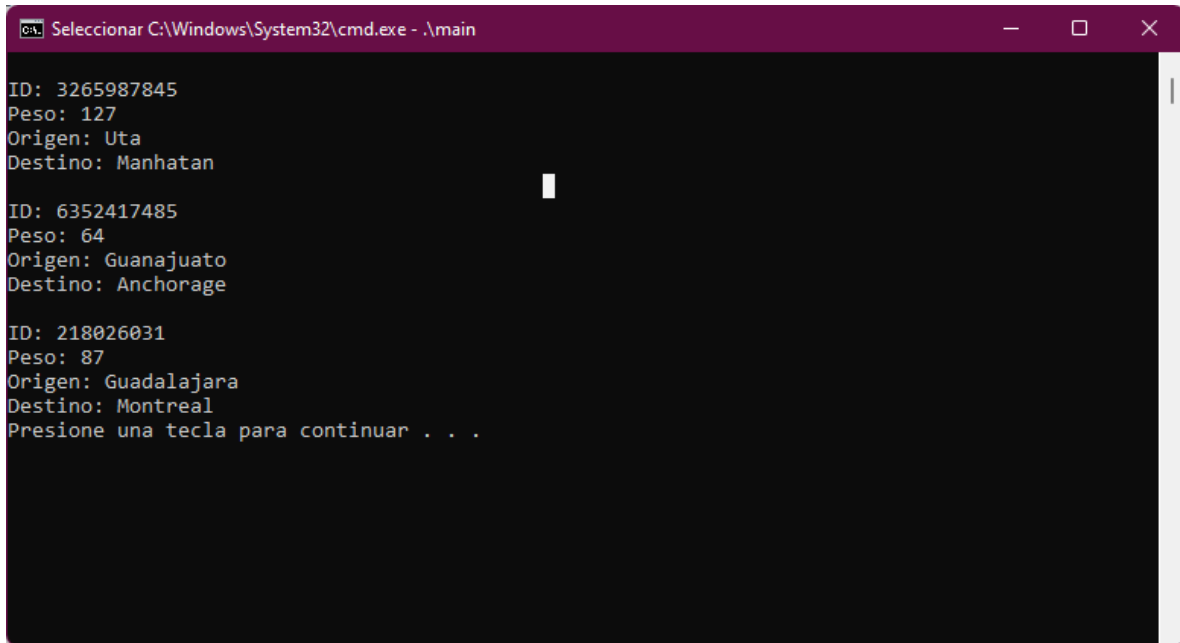
Captura de pantalla y código para eliminar paquete de paquetería:



Código Eliminar Paquete

```
void Paqueteria::eliminarLista()
{
    Paquete *aux = anchor;
    while (aux != nullptr)
    {
        aux = aux->getSiguiente();
        delete anchor;
        anchor = aux;
    }
    anchor = nullptr;
}
```

Captura de Pantalla para mostrar información de Paquetes de la lista de Paquetería:



```
Seleccionar C:\Windows\System32\cmd.exe - .\main

ID: 3265987845
Peso: 127
Origen: Uta
Destino: Manhattan

ID: 6352417485
Peso: 64
Origen: Guanajuato
Destino: Anchorage

ID: 218026031
Peso: 87
Origen: Guadalajara
Destino: Montreal
Presione una tecla para continuar . . .
```

Código que recorre la lista:

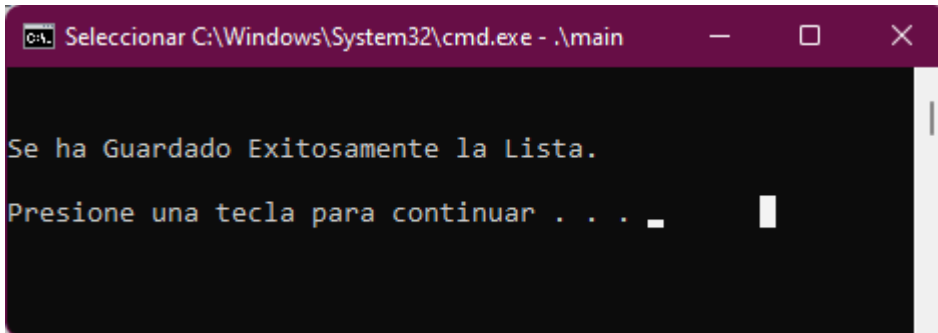
```
void Paqueteria::mostrarLista()
{
    Paquete *aux = anchor;

    if (anchor != nullptr)
    {
        while (aux != nullptr)
        {
            aux->print();
            aux = aux->getSiguiente();
        }
    } else {
        std::cout << "\n\nNo hay nada que imprimir. - Lista Vacía.\n\n";
    }
}
```

Código que imprime cada iteración del recorrido:

```
void Paquete::print() const
{
    std::cout << "\nID: " << getId() << "\nPeso: " << getPeso() << "\nOrigen: "
    << getOrigen() << "\nDestino: " << getDestino() << "\n";}
```

Captura de Pantalla y Código para Guardar Lista Paquetería en un archivo de Texto.



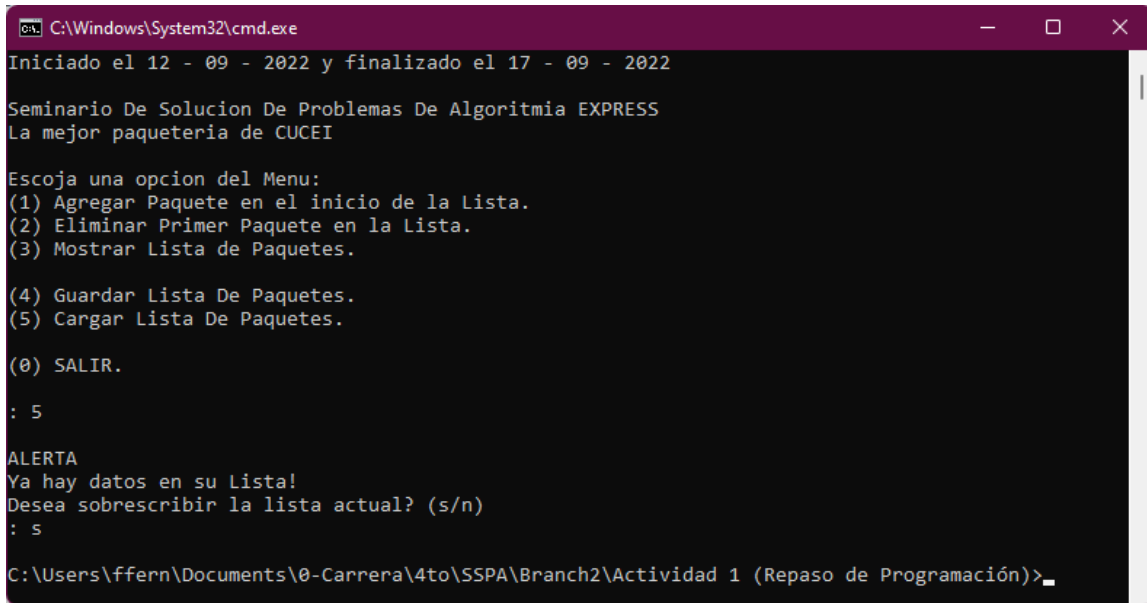
Código para guardar Lista:

```
void Paqueteria::guardarLista()
{
    Paquete *aux = anchor;

    std::ofstream file("SavedList.txt");

    while (aux != nullptr)
    {
        file << aux->getId() << "@" << aux->getPeso() << "@" << aux->
getOrigen() << "@" << aux->getDestino() << "@";
        aux = aux->getSiguiente();
    }
    file << "~";
    file.close();
}
```

Captura de Pantalla y Código para cargar lista desde un archivo de texto: **(ERROR)**



```
C:\Windows\System32\cmd.exe
Iniciado el 12 - 09 - 2022 y finalizado el 17 - 09 - 2022

Seminario De Solucion De Problemas De Algoritmia EXPRESS
La mejor paqueteria de CUCEI

Escoja una opcion del Menu:
(1) Agregar Paquete en el inicio de la Lista.
(2) Eliminar Primer Paquete en la Lista.
(3) Mostrar Lista de Paquetes.

(4) Guardar Lista De Paquetes.
(5) Cargar Lista De Paquetes.

(0) SALIR.

: 5

ALERTA
Ya hay datos en su Lista!
Desea sobrescribir la lista actual? (s/n)
: s

C:\Users\ffern\Documents\0-Carrera\4to\SSPA\Branch2\Actividad 1 (Repaso de Programación)>
```

Código para cargar lista de Paquetes:

```
void Paqueteria::cargarLista()
{
    char x = '\0';

    if (anchor != nullptr)
    {
        std::cout << "\nALERTA\nYa hay datos en su Lista!\nDesea sobrescribir la lista actual? (s/n)\n: ";
        std::cin >> x;

        if (x == 115)
        {
            eliminarLista();
        }
        else
        {
            x = 187; // cancelar operación
        }
    }

    if (x != 187)
    {
        // Creando Buffers
        char texto[4096];
        char dato[64];

        // Puntero para la variable "texto"
        int tp = 0;
```

```

// Limpiando Buffers
for (int i = 0; i < 4096; i++)
{
    texto[i] = '\0';
}
for (int i = 0; i < 20; i++)
{
    dato[i] = '\0';
}

// Se abre el archivo por defecto
std::ifstream file("SavedList.txt");

// Obtiene la cantidad de caracteres que puede almacenar el Buffer
file.getline(texto, 4096); //Supongo que aquí
origina el problema

```

```

do
{
    // Creando Nuevo Paquete
    Paquete aux;

    for (int j = 0; j < 4; j++)
    {
        for (int i = 0; texto[tp] != 64; i++, tp++)
        {
            dato[i] = texto[tp];
        }
        tp++;

        switch (j)
        {
            case 0:
                aux.setId(dato);
                break;
            case 1:
                aux.setPeso(dato);
                break;
            case 2:
                aux.setOrigen(dato);
                break;
            case 3:
                aux.setDestino(dato);
                break;
        }

        for (int i = 0; i < 20; i++)
        {
            dato[i] = '\0';
        }
        insertarAlInicio(aux);

        aux.setId(NULL);
        aux.setPeso(NULL);
        aux.setOrigen(NULL);
        aux.setDestino(NULL);
    }
} while (texto[tp] != 126);
}

```


Conclusiones

Este repaso fue realmente un reto para desarrollarlo causa de que es el tema que estoy abordando este semestre en **Estructuras de Datos 1**. He cursado ya todas las materias de estructuras de datos excepto la ya antes mencionada, y es justo la materia que me proporciona estas herramientas, no tenía idea de cómo implementar correctamente (Y no adivinando sintaxis, estándares y lógica), incluir cabeceras, qué herramientas usar y al menos cómo compilar un proyecto con esta estructura. Todo es Nuevo para mi y resolver mil dudas en una instancia es complicado causa de que, para conocer objetos, debo comprender antes estructuras de proyectos orientados a objetos, cómo estos interactúan con otros objetos, qué se puede hacer y que no, qué solución es óptima, etc. En fin, Este no fue un repaso en mi caso, ha sido un tema reforzado (a medias por no ser capaz de completar la lectura del archivo) en POO.

Referencias

<https://es.stackoverflow.com/questions/33784/c%C3%B3mo-concatenar-variables-en-c>
<https://www.programiz.com/cpp-programming/library-function/cstring/strcat>
<http://profesores.elo.utfsm.cl/~agv/elo326/list.pdf>
<https://learn.microsoft.com/en-us/cpp/standard-library/list-class?view=msvc-170>
<https://cplusplus.com/reference/vector/vector/>
<https://stackoverflow.com/questions/12919081/gcc-g-no-such-file-or-directory>
<https://dis.unal.edu.co/~fgonza/courses/2003/poo/c++.htm>
<https://www.programarya.com/Cursos/C++/Estructuras-de-Datos/Punteros>
<https://steemit.com/programacion/@vyvyrruz/como-crear-archivos-de-texto-en-c-or-el-informatico>
<https://www.delftstack.com/es/howto/cpp/cpp-create-file/>
<https://es.stackoverflow.com/questions/197665/error-el-objeto-tiene-calificadores-de-tipo-que-no-son-compatibles-con-la-funci>
<https://youtu.be/g3qtAhHqYcA>
<https://youtu.be/QqU8emKoilA>

Código

PAQUETE.h

```
#ifndef __PAQUETE_H__
#define __PAQUETE_H__

#include<iostream>
#include<string>

class Paquete
{
private:
    std::string id;
    std::string peso;
    std::string origen;
    std::string destino;

    Paquete* siguiente;
public:
    Paquete();
    //Paquete(std::string id, std::string peso, std::string getOrigen,
    std::string getDestino) : id(id), peso(peso), origen(origen),
    destino(destino) {};

    std::string getId() const;
    std::string getPeso() const;
    std::string getOrigen() const;
    std::string getDestino() const;
    Paquete* getSiguiente() const;

    void print() const;

    void setId(const std::string&);
    void setPeso(const std::string&);
    void setOrigen(const std::string&);
    void setDestino(const std::string&);
    void setSiguiente(Paquete*);

    Paquete& operator = (const Paquete&);
};

#endif // __Paquete_H__
```

PAQUETE.CPP

```
#include "paquete.h"

Paquete::Paquete()
{
    siguiente = nullptr;
}

std::string Paquete::getId() const
{
    return id;
}

std::string Paquete::getPeso() const
{
    return peso;
}

std::string Paquete::getOrigen() const
{
    return origen;
}

std::string Paquete::getDestino() const
{
    return destino;
}

Paquete* Paquete::getSiguiente() const
{
    return siguiente;
}

void Paquete::print() const
{
    std::cout << "\nID: " << getId() << "\nPeso: " << getPeso() <<
    "\nOrigen: " << getOrigen() << "\nDestino: " << getDestino() << "\n";
}

void Paquete::setId(const std::string& x)
{
    id = x;
}

void Paquete::setPeso(const std::string& x)
{
    peso = x;
}

void Paquete::setOrigen(const std::string& x)
```

```
{
    origen = x;
}

void Paquete::setDestino(const std::string& x)
{
    destino = x;
}

void Paquete::setSiguiente(Paquete* x)
{
    siguiente = x;
}
```

PAQUETERIA.H

```
#ifndef __PAQUETERIA_H__
#define __PAQUETERIA_H__

#include<iostream>
#include"paquete.h"

class Paqueteria
{
private:
    Paquete* anchor;
public:
    Paqueteria();

    void insertarAlInicio(Paquete&);
    void eliminarLista();
    void eliminarAlInicio();
    void mostrarLista();
    void guardarLista();
    void cargarLista();
};
#endif // __PAQUETERIA_H__
```

PAQUETERIA.CPP

```
#include "paqueteria.h"
#include <fstream>

Paqueteria::Paqueteria()
{
    anchor = nullptr;
}

void Paqueteria::insertarAlInicio(Paquete &x)
{
    x.setSiguiente(anchor);
    anchor = &x;
}

void Paqueteria::eliminarLista()
{
    Paquete *aux = anchor;
    while (aux != nullptr)
    {
        aux = aux->getSiguiente();
        delete anchor;
        anchor = aux;
    }
    anchor = nullptr;
}

void Paqueteria::eliminarAlInicio()
{
    Paquete *aux;

    aux = anchor;
    anchor = anchor->getSiguiente();
    delete aux;
}

void Paqueteria::mostrarLista()
{
    Paquete *aux = anchor;

    if (anchor != nullptr)
    {
        while (aux != nullptr)
        {
            aux->print();
            aux = aux->getSiguiente();
        }
    } else {
        std::cout << "\n\nNo hay nada que imprimir. - Lista Vacía.\n\n";
    }
}
```

```

void Paqueteria::guardarLista()
{
    Paquete *aux = anchor;

    std::ofstream file("SavedList.txt");

    while (aux != nullptr)
    {
        file << aux->getId() << "@" << aux->getPeso() << "@" << aux-
>getOrigen() << "@" << aux->getDestino() << "@";
        aux = aux->getSiguiente();
    }
    file << "~";
    file.close();
}

void Paqueteria::cargarLista()
{
    char x = '\0';

    if (anchor != nullptr)
    {
        std::cout << "\nALERTA\nYa hay datos en su Lista!\nDesea
sobrescribir la lista actual? (s/n)\n: ";
        std::cin >> x;

        if (x == 115)
        {
            eliminarLista();
        }
        else
        {
            x = 187; // cancelar operación
        }
    }

    if (x != 187)
    {
        // Creando Buffers
        char texto[4096];
        char dato[64];

        // Puntero para la variable "texto"
        int tp = 0;

        // Limpiando Buffers
        for (int i = 0; i < 4096; i++)
        {
            texto[i] = '\0';
        }
        for (int i = 0; i < 20; i++)
    }

```

```

{
    dato[i] = '\0';
}

// Se abre el archivo por defecto
std::ifstream file("SavedList.txt");

// Obtiene la cantidad de caracteres que puede almacenar el Buffer
file.getline(texto, 4096);

do
{
    // Creando Nuevo Paquete
    Paquete aux;

    for (int j = 0; j < 4; j++)
    {
        for (int i = 0; texto[tp] != 64; i++, tp++)
        {
            dato[i] = texto[tp];
        }
        tp++;

        switch (j)
        {
            case 0:
                aux.setId(dato);
                break;
            case 1:
                aux.setPeso(dato);
                break;
            case 2:
                aux.setOrigen(dato);
                break;
            case 3:
                aux.setDestino(dato);
                break;
        }

        for (int i = 0; i < 20; i++)
        {
            dato[i] = '\0';
        }
        insertarAlInicio(aux);

        aux.setId(NULL);
        aux.setPeso(NULL);
        aux.setOrigen(NULL);
        aux.setDestino(NULL);
    }
} while (texto[tp] != 126);
}

```


MAIN.CPP

```
/*
SEMINARIO DE SOLUCIÓN DE PROBLEMAS DE ALGORITMIA
NRC: 59928
ALCARAZ VALDIVIA MARCOS FERNANDO
Iniciado el 12 - 09 - 2022 y finalizado el 17 - 09 - 2022
*/

#include <iostream>

#include "paqueteria.h"

using namespace std;

int main(int argc, char *argv[])
{
    int menu0 = 0;
    string dato;
    Paqueteria lista;

    do
    {
        system("cls");
        cout << "SEMINARIO DE SOLUCION DE PROBLEMAS DE ALGORITMIA\nNRC:
59928\nALCARAZ VALDIVIA MARCOS FERNANDO\nIniciado el 12 - 09 - 2022 y
finalizado el 17 - 09 - 2022\n\n";
        cout << "Seminario De Solucion De Problemas De Algoritmia
EXPRESS\nLa mejor paqueteria de CUCEI\n\nEscoja una opcion del Menu:\n(1)
Agregar Paquete en el inicio de la Lista.\n(2) Eliminar Primer Paquete en la
Lista.\n(3) Mostrar Lista de Paquetes.\n\n(4) Guardar Lista De
Paquetes.\n(5) Cargar Lista De Paquetes.\n\n(0) SALIR.\n\n: ";
        cin >> menu0;
        switch (menu0)
        {
            case 1:
            {
                Paquete *nuevoPaquete = nullptr;
                nuevoPaquete = new Paquete();

                cout << "Escribe el ID del Paquete (De 9 digitos): ";
                cin >> dato;
                nuevoPaquete->setId(dato);

                cout << "Escribe el Peso del Paquete en Kg (De 0.1 a 999): ";
                cin >> dato;
                nuevoPaquete->setPeso(dato);

                cout << "Escribe el Origen del Paquete (Ciudad de Sucursal donde
se Recibio el Paquete): ";
                cin >> dato;
                nuevoPaquete->setOrigen(dato);
```

```

        cout << "Escribe el Destino del Paquete (Ciudad de Sucursal
donde se Entregara el Paquete): ";
        cin >> dato;
        nuevoPaquete->setDestino(dato);

        lista.insertarAlInicio(*nuevoPaquete);

        system("cls");
        cout << "\n\nPaquete Guardado Exitosamente!\n\n";
        break;
    }
    case 2:
    {
        lista.eliminarAlInicio();
        system("cls");
        cout << "\n\nSe ha eliminado al Paquete que estaba al Inicio de
la Lista.\n\n";
        break;
    }
    case 3:
    {
        system("cls");
        lista.mostrarLista();
        break;
    }
    case 4:
    {
        lista.guardarLista();
        system("cls");
        cout << "\n\nSe ha Guardado Exitosamente la Lista.\n\n";
        break;
    }
    case 5:
    {
        lista.cargarLista();
        cout << "\n\nSe ha ejecutado el proceso.\n\n";
        break;
    }
    default:
    {
        system("cls");
        cout << "\n\nPOR FAVOR ESCOJA UNA OPCION DEL MENU\n\n";
        break;
    }
    }
    system("pause");
    system("cls");
} while (menu0 != 0);

cout << "\n\n\n\nREALIZADO POR ALCARAZ VALDIVIA MARCOS
FERNANDO\n\n\n\n";}

```

EJEMPLO RESULTADO ARCHIVO

3265987845@127@Uta@Manhatan@6352417485@64@Guanajuato@Anchorage@218026031@87@Guadalajara@Montreal@~