

Actividad 01 – (Repaso de Programación)

Rubio Valenzuela Miguel Angel - 216567795

Seminario de Algoritmia – D02.

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 de pantalla del contenido del archivo `.txt`.
- [] Se muestra código y captura de pantalla de la opción Recuperar.

Desarrollo

Lo primero en realizarse fue la codificación de la clase Paquete

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <string>
#include <list>
#include <fstream>

using namespace std;

class Paquete
{
private:
    int id;
    string origen, destino;
    float peso;

public:
    // setter
    void SetPaquete(int i, string o, string d, float p);
    // getter
    void GetPaquete();
    void EliminarInicio();
    void Guardar();
    void Recuperar();
};
```

Luego de definir la clase Paquete definí las funciones en la parte baja las cuales definí de la siguiente manera.

```
void Paquete::SetPaquete(int i, string o, string d, float p)
{
}

}
```

Esto lo realice para cada una de las funciones anteriores.

Para guardar los datos en las listas, utilicé la recomendación que se nos dio en los requisitos de la tarea, el cual era utilizar el método *list*.

Este método realiza lo mismo que lo que hacen las listas simplemente ligadas, pero se implementan de una forma mucho mas sencilla, sin la utilización de nodos.

Esta es su implementación en el código, la lista define como valores a introducir dentro de la lista el objeto paquete y la lista se define como Paqueteria:

```
list<Paquete> Paqueteria;
```

Después de definir las listas y las funciones, cree la función main y además el menú, donde interactuaremos con el programa, para eso utilice el switch y definimos la clase de forma local llamada Paquete.

```
145 int main()
146 {
147     Paquete paquete;
148
149     int num;
150     do
151     {
152         system("cls");
153         cout << "\n\n||      PAQUETERIA      ||" << endl<<endl;
154         cout << "1.- Insertar al inicio." << endl;
155         cout << "2.- Eliminar al inicio." << endl;
156         cout << "3.- Mostrar" << endl;
157         cout << "4.- Guardar." << endl;
158         cout << "5.- Recuperar" << endl;
159         cout << "0.- Salir" << endl;
160         cout << "Seleccione una opcion: ";
161         cin >> num;
162
163         switch(num)
164         {
165             case 1:
166                 Ingresar();
167                 break;
168
169             case 2:
170                 paquete.EliminarInicio();
171                 break;
172
173             case 3:
174                 paquete.GetPaquete();
175                 break;
176
177             case 4:
178                 paquete.Guardar();
179                 break;
180
181             case 5:
182                 paquete.Recuperar();
183                 break;
184
185             default:
186                 break;
187         };
188     }while(num != 0);
189
190     system("PAUSE");
191     return 0;
192 }
193
194
```

Lo siguiente fue implementar la lógica dentro de cada una de las funciones, pero primero definí la función ingresar la cual será utilizada, para no tener que escribir los datos dentro del main, se llama a la clase Paquete para que llame a la función Set Paquete y esta es la que dará todos los valores:

```
122
123 void Ingresar()
124 {
125     Paquete package;
126     int ID;
127     string Orig, Dest;
128     float kg;
129
130     system("cls");
131     cout<<"Ingresando datos: "<<endl<<endl;
132     cout<<"Ingresa el ID: ";
133     cin>>ID;
134     cout<<"Ingresa el origen: ";
135     cin>>Orig;
136     cout<<"Ingresa el destino: ";
137     cin>>Dest;
138     cout<<"Ingresa el peso: ";
139     cin>>kg;
140
141     package.SetPaquete(ID, Orig, Dest, kg);
142     system("pause");
143 }
```

Lo siguiente que implementé, fue la función SetPaquete, que esta función mete los datos a el objeto en y siguiente los almacena en la lista, esto lo realiza de la siguiente forma:

```
28
29 void Paquete::SetPaquete(int i, string o, string d, float p)
30 {
31     Paquete package;
32
33     fflush(stdin);
34     package.id = i;
35     package.origen = o;
36     package.destino = d;
37     package.peso = p;
38     Paqueteria.push_front(package);
39 }
40
```

La manera en que introduce los datos a la lista es con el método push_front(), que introduce los datos a la lista en el inicio.

Y usamos la función fflush liberar la memoria.

Lo siguiente que implemente es la función GetPaquete() esta es la función que nos ayuda a mostrar los datos en la pantalla.

Uno de los problemas que tenía este método es que solamente podría mostrar el primero o el ultimo conjunto, así que tuve que realizar un ciclo while que donde se tenia otra lista llamada Pqt

y esta lista seria un duplicado de la lista original, ya que se tenia que vaciar la lista original para poder pasar a los datos mas cercanos al centro de la lista.

```
void Paquete::GetPaquete()
{
    Paquete package;
    list<Paquete> Pqt;

    system("cls");
    cout<<"Mostrando datos de la paqueteria"<<endl<<endl;

    Pqt = Paqueteria;
    while(!Paqueteria.empty())
    {
        package = Paqueteria.front();

        cout<<"ID: "<<package.id<<endl;
        cout<<"origen: "<<package.origen<<endl;
        cout<<"Destino: "<<package.destino<<endl;
        cout<<"Peso Kg: "<<package.peso<<endl<<endl;

        Paqueteria.pop_front();
    }

    Paqueteria = Pqt;

    system("pause");
}
```

El siguiente método que implemente fue el de la eliminación, para este tipo de listas, tenemos un método de eliminación más fácil de utilizar, donde solamente usamos el pop_front() y se eliminara el primer elemento de la lista

```
void Paquete::EliminarInicio()
{
    Paqueteria.pop_front();
    system("cls");
    cout<<"Elemento del inicio eliminado."<<endl<<endl;
    system("pause");
}
```

Lo siguiente en implementar son las dos funciones primordiales, las cuales son las de guardar y la de recuperar.

El guardado del respaldo es usar el mismo proceso que mostrar los datos, pero en lugar de mostrar los datos usamos el método de ofstream y le metemos todos los datos en un archivo txt.

El archivo donde ingresamos los datos se llama Respaldo.txt.

```

void Paquete::Guardar()
{
    system("cls");
    ofstream archivo("Respaldo.txt");
    if (archivo.is_open())
    {
        Paquete package;
        while(!Paqueteria.empty())
        {
            package = Paqueteria.front();
            archivo << package.id << setw(10) << package.origen << setw(10) << package.destino << setw(10) << package.fecha;
            Paqueteria.pop_front();
        }
        cout<<"Respaldo generado con exito"<<endl;
    }
    else
    {
        cout<<"No fue posible generar un respaldo"<<endl;
    }
    archivo.close();
    system("pause");
}

```

Y por último, se realiza la implementación del Recuperar () donde realizamos la recuperación de los datos respaldados con anterioridad.

El uso de este método es prácticamente el mismo que el de Guardar (), pero a comparación en este tiene se insertan los datos en las listas, y utilizamos el archivo.eof() debido a que esta línea de código nos ayuda a saber si el ultimo dato dentro del archivo txt existe o no.

```

void Paquete::Recuperar()
{
    system("cls");
    ifstream archivo("Respaldo.txt");
    if (archivo.is_open())
    {
        Paquete package;
        while(!archivo.eof())
        {
            archivo >> package.id >> setw(10) >> package.origen >> setw(10) >> package.destino >> setw(10) >> package.fecha;
            if(archivo.eof()){ break;}
            else
            Paqueteria.push_back(package);
        }
        cout<<"Informacion recuperada."<<endl;
    }
    else
    {
        cout<<"No se pudo recuperar"<<endl;
    }
    archivo.close();
    system("pause");
}

```

Ya por último se muestran los resultados de mi código:

Primero vemos el menú inicial:

```
||      PAQUETERIA      ||
1.- Insertar al inicio.
2.- Eliminar al inicio.
3.- Mostrar
4.- Guardar.
5.- Recuperar
0.- Salir
Seleccione una opcion: _
```

Ingresamos 1 y nos vamos a la sección de insertar datos.

En esta sección vamos a ingresar Id: 1, 2, 3, Origen: Mexico, Inglaterra, Japon, Destino, USA, China, Canada, Peso: 12.0, 130.7, 36.9.

```
Seleccione una opcion: 1

Ingresando datos:

Ingresa el ID: 3
Ingresa el origen: Japon
Ingresa el destino: Canada
Ingresa el peso: 36.9
```

Despues, mostramos los datos ingresados:

```
Mostrando datos de la paqueteria

ID: 3
origen: Japon
Destino: Canada
Peso Kg: 36.9

ID: 2
origen: Inglaterra
Destino: China
Peso Kg: 130.7

ID: 1
origen: Mexico
Destino: USA
Peso Kg: 12
```

Lo siguiente en realizar es el guardado de los datos en un archivo de texto.

```
Respaldo generado con exito
```

Aquí se muestra el archivo txt:

```
Respaldo: Bloc de notas
Archivo  Editar  Ver

3      Japon      Canada      36.9
2Inglaterra      China      130.7
1      Mexico      USA      12
```

Observamos que al mostrar los datos no se nos despliega nada de información.

```
0.- Salir
Seleccione una opcion: 3
a
Mostrando datos de la paqueteria

||      PAQUETERIA      ||
```


Finalmente tenemos que recuperar los datos.

```
Seleccione una opcion: 5

Informacion recuperada.

||      PAQUETERIA      ||
1.- Insertar al inicio.
2.- Eliminar al inicio.
3.- Mostrar
4.- Guardar.
5.- Recuperar
0.- Salir
Seleccione una opcion: 3

Mostrando datos de la paqueteria

ID: 3
origen: Japon
Destino: Canada
Peso Kg: 36.9

ID: 2
origen: Inglaterra
Destino: China
Peso Kg: 130.7

ID: 1
origen: Mexico
Destino: USA
Peso Kg: 12
```

Y así es como se demuestra el funcionamiento del programa, y comprobamos que se cumplen los requerimientos pedidos en la actividad.

Conclusiones

El mayor problema que tuve fue con respecto a volver a retomar los temas vistos en semestres pasados, no recordaba como es que funcionaban las clases y como implementarlas, además de que no supe como es que se utilizaban los nodos para hacer una lista simplemente ligada, pero gracias a la recomendación que nos marco el profe con el método list pude realizar la actividad de una forma más sencilla.

Aprendí el uso de las listas con el método list, y es un método mucho mas sencillo que usar nodos, además de que volví a reafirmar los temas vistos en semestres pasados, y pues saber el funcionamiento del uso de archivos se me hace muy bueno, porque de esta manera podemos tener una copia de los datos y usarlos de la manera en que nosotros consideremos mejor.

Referencias

<https://dis.unal.edu.co/~fgonza/courses/2003/poo/c++.htm> – Clases en C++

[https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C%2B%2B/Librer%C3%ADa_Est%C3%A1ndar_de_Plantillas/Listas#:~:text=Las%20listas%20\(Lists\)%20de%20C%2B%2B,menor%20velocidad%20de%20acceso%20aleatorio.](https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C%2B%2B/Librer%C3%ADa_Est%C3%A1ndar_de_Plantillas/Listas#:~:text=Las%20listas%20(Lists)%20de%20C%2B%2B,menor%20velocidad%20de%20acceso%20aleatorio.) – Wikilibros –

<https://cplusplus.com/doc/tutorial/classes/> - cplusplus

[Crear y Escribir un archivo es C++ \(respaldar\) - YouTube](#) – Crear y Escribir un archivo es C++ (respaldar) – Michel Davalos Boites.

[Leer un archivo en C++ \(recuperar\) - YouTube](#) – Leer un archivo en C++ (recuperar).

Código.

```
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <string>
#include <list>
#include <fstream>

using namespace std;

class Paquete
{
    private:
        int id;
        string origen, destino;
        float peso;

    public:
        // setter
        void SetPaquete(int i, string o, string d, float p);
        // getter
        void GetPaquete();
        void EliminarInicio();
        void Guardar();
        void Recuperar();
};

list<Paquete> Paqueteria;

void Paquete::SetPaquete(int i, string o, string d, float p)
```

```

{
    Paquete package;

    fflush(stdin);
    package.id = i;
    package.origen = o;
    package.destino = d;
    package.peso = p;
    Paqueteria.push_front(package);
}

void Paquete::GetPaquete()
{
    Paquete package;
    list<Paquete> Pqt;

    cout<<"\n\nMostrando datos de la paqueteria"<<endl<<endl;

    Pqt = Paqueteria;
    while(!Paqueteria.empty())
    {
        package = Paqueteria.front();

        cout<<"ID: "<<package.id<<endl;
        cout<<"origen: "<<package.origen<<endl;
        cout<<"Destino: "<<package.destino<<endl;
        cout<<"Peso Kg: "<<package.peso<<endl<<endl;

        Paqueteria.pop_front();
    }

    Paqueteria = Pqt;
}

void Paquete::EliminarInicio()
{
    Paqueteria.pop_front();
    cout<<"\n\nElemento del inicio eliminado."<<endl<<endl;
}

void Paquete::Guardar()
{
    ofstream archivo("Respaldo.txt");
    if (archivo.is_open())

```

```

{
    Paquete package;
    while(!Paqueteria.empty())
    {
        package = Paqueteria.front();
        archivo << package.id << setw(10) << package.origen << setw(10)
<< package.destino << setw(10) << package.peso << endl;
        Paqueteria.pop_front();
    }
    cout<<"\n\nRespaldo generado con exito"<<endl;
}
else
{
    cout<<"\n\nNo fue posible generar un respaldo"<<endl;
}
archivo.close();
}

void Paquete::Recuperar()
{
    ifstream archivo("Respaldo.txt");
    if (archivo.is_open())
    {
        Paquete package;

        while(!archivo.eof())
        {
            archivo >> package.id >> setw(10) >> package.origen >>
setw(10) >> package.destino >> setw(10) >> package.peso;
            if(archivo.eof()){ break;}
            else
                Paqueteria.push_back(package);
        }
        cout<<"\n\nInformacion recuperada."<<endl;
    }
    else
    {
        cout<<"\n\nNo se pudo recuperar"<<endl;
    }
    archivo.close();
}

void Ingresar()
{
    Paquete package;

```

```

int ID;
string Orig, Dest;
float kg;

cout<<"\n\nIngresando datos: "<<endl<<endl;
cout<<"Ingresa el ID: ";
cin>>ID;
cout<<"Ingresa el origen: ";
cin>>Orig;
cout<<"Ingresa el destino: ";
cin>>Dest;
cout<<"Ingresa el peso: ";
cin>>kg;

package.SetPaquete(ID, Orig, Dest, kg);
}

int main()
{
    Paquete paquete;

    int num;
    do
    {
        cout << "\n\n||      PAQUETERIA      ||" << endl<<endl;
        cout << "1.- Insertar al inicio." << endl;
        cout << "2.- Eliminar al inicio." << endl;
        cout << "3.- Mostrar" << endl;
        cout << "4.- Guardar." << endl;
        cout << "5.- Recuperar" << endl;
        cout << "0.- Salir" << endl;
        cout << "Seleccione una opcion: ";
        cin >> num;
        switch(num)
        {
            case 1:
                Ingresar();
                break;

            case 2:
                paquete.EliminarInicio();
                break;

            case 3:
                paquete.GetPaquete();

```

```
        break;

        case 4:
            paquete.Guardar();
            break;

        case 5:
            paquete.Recuperar();
            break;

        default:
            break;
    };

    }while(num != 0);
    return 0;
}
```