

Actividad 01 - Repaso de Programación

Gabriel Eduardo Sevilla Chavez

Seminario 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

- lista del menú, instancias de las clases "*Paquete*" y "*Paqueteria*" y un contador global

.

- subrutinas y funciones para la clase *main*

```
1 using namespace std;
2 enum {AGREGAR=1,ELIMINAR,
  MOSTRAR, GUARDAR, RECUPERAR, SALIR};
3 Paquete packet;
4 Paqueteria carrito;
5 int contador=1;
```

```
1 void MenuPrincipal(int& opc);
2 int opcionesMenu(int& opc);
3 void agregarPaquete();
4 void eliminarPaquete();
5 void pausar();
```

• Clase *Paquete*

- Atributos

```
1 class Paquete{
2     int id;
3     float peso;
4     string origen;
5     string destino;
```

- Métodos getters y setters

```
1 public:
2     void setId(int idP){
3         id = idP;
4     }
5     int getId(){
6         return id;
7     }
8
9     void setPeso(float pesoP){
10        peso = pesoP;
11    }
12    float getPeso(){
13        return peso;
14    }
15
16    void setOrigen(string origenP){
17        origen = origenP;
18    }
19    string getOrigen(){
20        return origen;
21    }
22
23    void setDestino(string destinoP){
24        destino = destinoP;
25    }
26    string getDestino(){
27        return destino;
28    }
29 };
```

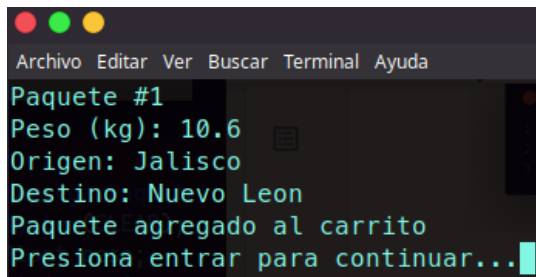
- agregar un objeto paquete dentro de la lista de la clase Paqueteria.

```
1 class Paqueteria{
2     vector<Paquete> paquetes;
```

```
1 void insertarALInicio(Paquete paquete){
2     paquetes.push_back(paquete);
3 }
```

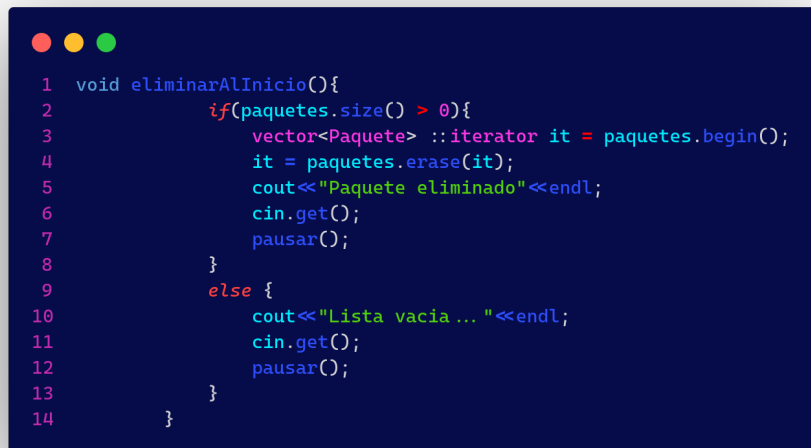
```
1 void agregarPaquete(){
2     system(CLEAR);
3     float peso;
4     string origen;
5     string destino;
6
7     packet.setId(contador);
8     cout<<"Paquete #"<<packet.getId()<<endl;
9     cout<<"Peso (kg): ";
10    cin>>peso;
11    packet.setPeso(peso);
12    cin.get();
13    cout<<"Origen: ";
14    getline(cin,origen);
15    packet.setOrigen(origen);
16    cout<<"Destino: ";
17    getline(cin,destino);
18    packet.setDestino(destino);
19    carrito.insertarALInicio(packet);
20    cout<<"Paquete agregado al carrito"<<endl;
21    contador++;
22    carrito.pausar();
23 }
24
```

○ Captura

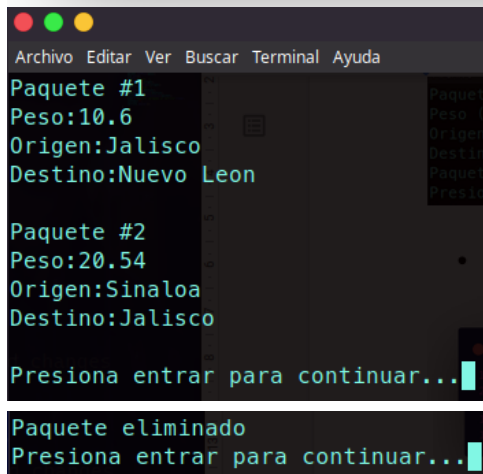


```
Archivo Editar Ver Buscar Terminal Ayuda
Paquete #1
Peso (kg): 10.6
Origen: Jalisco
Destino: Nuevo Leon
Paquete agregado al carrito
Presiona entrar para continuar...
```

- eliminar un objeto de la lista de paquete en la clase Paqueteria.



```
1 void eliminarAlInicio(){
2     if(paquetes.size() > 0){
3         vector<Paquete> ::iterator it = paquetes.begin();
4         it = paquetes.erase(it);
5         cout<<"Paquete eliminado"<<endl;
6         cin.get();
7         pausar();
8     }
9     else {
10        cout<<"Lista vacia ... "<<endl;
11        cin.get();
12        pausar();
13    }
14 }
```

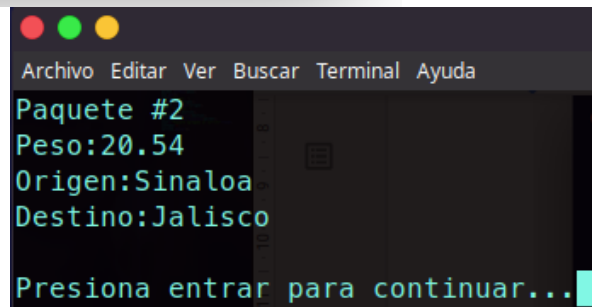


```
Archivo Editar Ver Buscar Terminal Ayuda
Paquete #1
Peso:10.6
Origen:Jalisco
Destino:Nuevo Leon

Paquete #2
Peso:20.54
Origen:Sinaloa
Destino:Jalisco

Presiona entrar para continuar...

Paquete eliminado
Presiona entrar para continuar...
```



```
Archivo Editar Ver Buscar Terminal Ayuda
Paquete #2
Peso:20.54
Origen:Sinaloa
Destino:Jalisco

Presiona entrar para continuar...
```

- Guardar la lista de paquete en un archivo .txt.

```

1 void guardarPaquetes(){
2     ofstream archivo("Paquetes.txt");
3     if (archivo.is_open()){
4         archivo << left;
5         for(size_t i=0;i< paquetes.size();i++){
6             archivo<<"Paquete #"<<paquetes[i].getId()<<endl;
7             archivo<<"Peso:"<<paquetes[i].getPeso()<<endl;
8             archivo<<"Origen:"<<paquetes[i].getOrigen()<<endl;
9             archivo<<"Destino:"<<paquetes[i].getDestino()<<endl<<endl;
10        }
11    }
12    archivo.close();
13    cout<<"Lista de paquetes guardados"<<endl;
14    cin.get();
15    pausar();
16 }

```

Archivo Editar Ver Buscar Terminal Ayuda

Menu Principal

1. Agregar paquete
2. Eliminar paquete
3. Mostrar paquete
4. Guardar paquete
5. Recuperar paquete
6. Salir

4

Lista de paquetes guardados

Presiona entrar para continuar...

.gitignore

main U

main.cpp

Paquete.h

Paqueteria.h

Paquetes.txt M

Recuperar.txt M

```

1 Paquete #2
2 Peso:20.54
3 Origen:Sinaloa
4 Destino:Jalisco
5
6 Paquete #3
7 Peso:30.5
8 Origen:Japon
9 Destino:China
10

```

Opción Recuperar.

```
1 int recuperarArchivo(){
2     ifstream archivo("Recuperar.txt");
3     int contador;
4     if(archivo.is_open()){
5         Paquete p;
6         string temp; //temporal
7         int id;
8
9         float peso;
10
11         while (true){
12             getline(archivo,temp); //ID
13             if(archivo.eof()){
14                 cout<<"Paquetes recuperados"<<endl;
15                 cin.get();
16                 pausar();
17                 break;
18             }
19             id = stoi(temp);
20             p.setId(id);
21             contador = id;
22             getline(archivo,temp); //peso
23             peso = stof(temp);
24             p.setPeso(peso);
25
26             getline(archivo,temp); //Origen
27             p.setOrigen(temp);
28
29             getline(archivo,temp); //Destino
30             p.setDestino(temp);
31
32             insertarALInicio(p);
33         }
34     }
35     archivo.close();
36     return contador+1;
37 }
38 }
```

```
Archivo Editar Ver Buscar Terminal Ayuda
Menu Principal
1.Agregar paquete
2.Eliminar paquete
3.Mostrar paquete
4.Guardar paquete
5.Recuperar paquete
6.Salir
5
Paquetes recuperados
Presiona entrar para continuar...
```

```
Archivo Editar Ver Buscar Terminal Ayuda
Paquete #2 100%
Peso:20.54
Origen:Sinaloa
Destino:Jalisco

Paquete #3
Peso:30.5
Origen:Japon
Destino:China

Presiona entrar para continuar...
```

Conclusiones

El programa pudo concluirse satisfactoriamente. Hubo varios intentos fallidos, ya que, había por ejemplo métodos de <vetor> que no recordaba y unos detalles con sintaxis y presentación del programa pero se pudo terminar todo correctamente.

Referencias

- MICHEL DAVALOS BOITES. (2020, 14 octubre). *Crear y Escribir un archivo es C++ (respaldar)*. YouTube. Recuperado 17 de septiembre de 2022, de <https://www.youtube.com/watch?v=g3gtAhHgYcA>
- MICHEL DAVALOS BOITES. (2020b, octubre 14). *Leer un archivo en C++ (recuperar)*. YouTube. Recuperado 17 de septiembre de 2022, de <https://www.youtube.com/watch?v=QqU8emKoiIA>
- *Programación en C++/Biblioteca Estándar de Plantillas/Vectores - Wikilibros*. (s. f.). Recuperado 17 de septiembre de 2022, de https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_C%2B%2B/Biblioteca_Est%C3%A1ndar_de_Plantillas/Vectores

Código

main.cpp

```
#include <iostream>
#include "Paquete.h"
#include "Paqueteria.h"
#include <fstream>
#ifdef _WIN32
#define CLEAR "cls"
#else
#define CLEAR "clear"
#endif // _WIN32
```

```
using namespace std;
enum {AGREGAR=1,ELIMINAR,MOSTRAR,GUARDAR,RECUPERAR,SALIR};
Paquete packet;
Paqueteria carrito;
int contador=1;
void MenuPrincipal(int& opc);
int opcionesMenu(int& opc);
void agregarPaquete();
void eliminarPaquete();
void pausar();

int main(){
    int opc;
    MenuPrincipal(opc);
    return 0;
}

void MenuPrincipal(int& opc){
    do{
        opcionesMenu(opc);
        switch(opc){
            case AGREGAR:
                agregarPaquete();
                break;
            case ELIMINAR:
                carrito.eliminarAlInicio();
                break;
            case MOSTRAR:
```

```
        carrito.listarPaquetes();
        break;
    case GUARDAR:
        carrito.guardarParaRecuperar();
        carrito.guardarPaquetes();
        break;
    case RECUPERAR:
        contador = carrito.recuperarArchivo();
        break;
    case SALIR:
        break;
    default:
        cout<<"Opcion no valida...";
    }
}
while(opc!=SALIR);
}
```

```
int opcionesMenu(int& opc){
    system(CLEAR);
    cout<<"Menu Principal"<< endl;
    cout<<"1.Agregar paquete"<<endl;
    cout<<"2.Eliminar paquete"<<endl;
    cout<<"3.Mostrar paquete"<<endl;
    cout<<"4.Guardar paquete"<<endl;
    cout<<"5.Recuperar paquete"<<endl;
    cout<<"6.Salir"<<endl;
```

```
        cin>>opc;
        return opc;
    }

void agregarPaquete(){
    system(CLEAR);
    float peso;
    string origen;
    string destino;

    packet.setId(contador);
    cout<<"Paquete #"<<packet.getId()<<endl;
    cout<<"Peso (kg): ";
    cin>>peso;
    packet.setPeso(peso);
    cin.get();
    cout<<"Origen: ";
    getline(cin,origen);
    packet.setOrigen(origen);
    cout<<"Destino: ";
    getline(cin,destino);
    packet.setDestino(destino);
    carrito.insertarALInicio(packet);
    cout<<"Paquete agregado al carrito"<<endl;
    contador++;
    carrito.pausar();
}
```

Paquete.h

```
#ifndef PAQUETE_H_INCLUDED
#define PAQUETE_H_INCLUDED
using namespace std;
class Paquete{
    int id;
    float peso;
    string origen;
    string destino;
public:
    void setId(int idP){
        id = idP;
    }
    int getId(){
        return id;
    }
    void setPeso(float pesoP){
        peso = pesoP;
    }
    float getPeso(){
        return peso;
    }
    void setOrigen(string origenP){
        origen = origenP;
    }
    string getOrigen(){
        return origen;
    }
}
```

```
    }  
    void setDestino(string destinoP){  
        destino = destinoP;  
    }  
    string getDestino(){  
        return destino;  
    }  
};  
#endif // PAQUETE_H_INCLUDED
```

Paqueteria.h

```
#ifndef PAQUETERIA_H_INCLUDED  
#define PAQUETERIA_H_INCLUDED  
#include <iostream>  
#include <vector>  
#include <fstream>  
#ifdef _WIN32  
#define CLEAR "cls"  
#else  
#define CLEAR "clear"  
#endif // _WIN32  
using namespace std;  
class Paqueteria{  
    vector<Paquete> paquetes;  
public:  
    void pausar(){  
        cout << "Presiona entrar para continuar...";  
    }  
};
```

```
cin.get();
}
void insertarAlInicio(Paquete paquete){
    paquetes.push_back(paquete);
}
void eliminarAlInicio(){
    if(paquetes.size() > 0){
        vector<Paquete> ::iterator it = paquetes.begin();
        it = paquetes.erase(it);
        cout<<"Paquete eliminado"<<endl;
        cin.get();
        pausar();
    }
    else {
        cout<<"Lista vacia..."<<endl;
        cin.get();
        pausar();
    }
}
void listarPaquetes(){
    system(CLEAR);
    if(paquetes.size()==0){
        cout<<"lista vacia..."<<endl;
        pausar();
    }
    for(size_t i=0; i< paquetes.size();i++){
        cout<<"Paquete #"<<paquetes[i].getId()<<endl;
```

```
        cout<<"Peso:"<<paquetes[i].getPeso()<<endl;
        cout<<"Origen:"<<paquetes[i].getOrigen()<<endl;

cout<<"Destino:"<<paquetes[i].getDestino()<<endl<<endl;

    }
    cin.get();
    pausar();
}

void guardarPaquetes(){
    ofstream archivo("Paquetes.txt");
    if (archivo.is_open()){
        archivo << left;
        for(size_t i=0;i< paquetes.size();i++){
            archivo<<"Paquete #"<<paquetes[i].getId()<<endl;
            archivo<<"Peso:"<<paquetes[i].getPeso()<<endl;
            archivo<<"Origen:"<<paquetes[i].getOrigen()<<endl;

archivo<<"Destino:"<<paquetes[i].getDestino()<<endl<<endl;

        }
    }
    archivo.close();
    cout<<"Lista de paquetes guardados"<<endl;
    cin.get();
    pausar();
}

void guardarParaRecuperar(){
    ofstream archivo("Recuperar.txt");
```

```
    if (archivo.is_open()){
        archivo << left;
        for(size_t i=0;i< paquetes.size();i++){
            archivo<<paquetes[i].getId()<<endl;
            archivo<<paquetes[i].getPeso()<<endl;
            archivo<<paquetes[i].getOrigen()<<endl;
            archivo<<paquetes[i].getDestino()<<endl;
        }
    }
    archivo.close();
}

int recuperarArchivo(){
    ifstream archivo("Recuperar.txt");
    int contador;
    if(archivo.is_open()){
        Paquete p;
        string temp; //temporal
        int id;

        float peso;

        while (true){
            getline(archivo,temp); //ID
            if(archivo.eof()){
                cout<<"Paquetes recuperados"<<endl;
                cin.get();
                pausar();
            }
        }
    }
}
```

```
        break;
    }
    id = stoi(temp);
    p.setId(id);
    contador = id;
    getline(archivo, temp); //peso
    peso = stof(temp);
    p.setPeso(peso);

    getline(archivo, temp); //Origen
    p.setOrigen(temp);

    getline(archivo, temp); //Destino
    p.setDestino(temp);

    insertarALInicio(p);
}
}
archivo.close();
return contador+1;
}
};
#endif // PAQUETERIA_H_INCLUDED
```