



UAEM

Universidad Autónoma
del Estado de México



Universidad Autónoma del Estado de México

Centro Universitario UAEM Atlacomulco

“Funciones Update, Delete y Find con estructuras de datos”

Presenta:

Elias Edgardo Segundo Antonio

Revisa:

Ingeniero José Luis García Morales

Unidad de Aprendizaje:

Programación avanzada

Fecha de entrega:

02 de abril de 2017

**Funciones a implementar**

Además de las operaciones realizadas en los trabajos anteriores, implementar las siguientes funcionalidades al programa, realizarlo con tres casos: pasajeros, estudiantes, libros.

- Modificar el campo de un elemento con la clave
- Eliminar un elemento con la clave
- Encontrar un elemento con la clave, con búsqueda binaria o lineal a elección del usuario.

traveller.cpp

```
/*
Programa de registro de pasajeros
Elias Edgardo Segundo Antonio
*/
#include <iostream>
#include <vector>
#include <iomanip>
#include <fstream>
#include <stdlib.h>
using namespace std;

struct traveller{
    int key;
    string name;
    string origin;
    string destiny;
    double price;
};

vector<traveller> travellers(10);

//Prototype section
void makeUserPage();
void debug();
int findT(int);
void sort();
void setup();
void read();
void showData();
void getData();
void sort();
bool insertData(traveller);
int findPlace();
string join(string);
string split(string);
```



```
void write();
void menu();
bool updateTraveller(int);
bool eraseTraveller(int);
int findTraveller(int);

int main()
{
    setup();
    int option,key;
    read();
    sort();
    while(option!=7)
    {
        cout<<"\n\n\n";
        menu();
        cin>>option;
        switch(option)
        {
            case 1:
                getData();
                break;
            case 2:
                sort();
                write();
                showData();
                break;
            case 3:
                cout<<"Ingresar clave: ";
                cin>>key;
                updateTraveller(key);
                break;
            case 4:
                cout<<"Ingresar clave: ";
                cin>>key;
                if(eraseTraveller(key))
                {
                    cout<<"Pasajero eliminado!\n";
                }
                else cout<<"Error al borrar pasajero\n";
                break;
            case 5:
                cout<<"Ingrese clave: ";
                cin>>key;
                findT(key);
        }
    }
}
```



```

        break;
    case 6:
        makeUserPage();
        system("travellers.html");
        break;
    default:
        if(option!=7)
            cout<<"\nIngrese una opcion valida\n";
    }
}

int findT(int key)
{
    int option;
    cout<<"Que tipo de busqueda desea realizar?\n";
    cout<<"1.-Lineal  2.-Binaria\n";
    cout<<"Ingrese: ";
    cin>>option;
    switch(option)
    {
        case 1:
            for(int i=0;i<travellers.size();i++)
            {
                if(travellers[i].key==key)
                {
                    cout<<"Elemento encontrado en la posicion "<<i<<"\n";
                    cout<<travellers[i].name<<"\n";
                    cout<<travellers[i].origin<<"\n";
                    cout<<travellers[i].destiny<<"\n";
                    cout<<travellers[i].price<<"\n\n";
                    return 1;
                }
            }
            return -1;
            break;
        case 2:
            sort();
            int right=0,left=0,m;
            while(travellers[right].key!=-1)right++;
            while(left<=right)
            {
                m=int((left+right)/2);
                if(travellers[m].key==key)

```



```

        {
            cout<<"Elemento encontrado en la posicion "<<m<<"\n";
            cout<<travellers[m].name<<"\n";
            cout<<travellers[m].origin<<"\n";
            cout<<travellers[m].destiny<<"\n";
            cout<<travellers[m].price<<"\n\n";
            return 1;
        }
        if(travellers[m].key>key) right=m-1;
        else left=m+1;
    }
    return -1;
    break;
}
}

void sort()
{
    for(int i=0;i<travellers.size();i++)
    {
        for(int j=0;j<travellers.size();j++)
        {
            if(travellers[i].key<travellers[j].key)
            {
                if(travellers[i].key!=-1)
                    swap(travellers[i],travellers[j]);
            }
        }
    }
}

bool updateTraveller(int key)
{
    int index=-1;
    for(int i=0;i<travellers.size();i++)
    {
        if(travellers[i].key==key)
            index=i;
    }
    if(index!=-1)
    {
        int option;
        cout<<"Que campo desea cambiar?\n"
        <<"1.-Nombre 2.-Origen 3.-Destino 4.-Precio\n";
        cout<<"Ingrese la opcion: ";
    }
}

```



```

        cin>>option;
        switch(option)
        {
            case 1:
                cin.get();
                cout<<"Ingrese el nuevo nombre: ";
                getline(cin,travellers[index].name);
                break;
            case 2:
                cin.get();
                cout<<"Ingrese el nuevo origen: ";
                getline(cin,travellers[index].origin);
                break;
            case 3:
                cin.get();
                cout<<"Ingrese el nuevo destino: ";
                getline(cin,travellers[index].destiny);
                break;
            case 4:
                cout<<"Ingrese el nuevo precio: ";
                cin>>travellers[index].price;
                break;
        }
        write();
    }
    else return false;
}

void menu()
{
    cout<<"Ingrese el numero de la opcion deseada: \n"
    <<"1.-Registrar Un pasajero\n"
    <<"2.-Mostrar datos del archivo\n"
    <<"3.-Modificar un pasajero por clave\n"
    <<"4.-Eliminar un elemento por clave\n"
    <<"5.-Buscar un elemento por clave\n"
    <<"6.-Mostrar pagina web\n"
    <<"7.- Salir\n"
    <<"Ingrese: ";
}

bool eraseTraveller(int key)
{
    int index=-1;
    for(int i=0;i<travellers.size();i++)

```



```

    {
        if(travellers[i].key==key)
            index=i;
    }
    if(index!=-1)
    {
        travellers.erase(travellers.begin()+index);
        write();
        return true;
    }
    else return false;
}

void showData()
{
    read();
    cout<<setw(6)<<"Clave"<<setw(40)<<"Nombre"<<
    setw(30)<<"Origen"<<setw(30)<<"Destino"<<setw(12)<<"Pasaje\n";
    for(traveller t: travellers)
    {
        if(t.key!=-1)
        {
            cout<<setw(6);
            cout<<t.key;
            cout<<setw(40);
            cout<<split(t.name);
            cout<<setw(30);
            cout<<split(t.origin);
            cout<<setw(30);
            cout<<split(t.destiny);
            cout<<setw(12);
            cout<<t.price<<" \n";
        }
    }
}

void read()
{
    setup();
    ifstream in("passengers.txt",ios::in);
    int pos=0;
    while(in>>travellers[pos].key>>travellers[pos].name>>
        travellers[pos].origin>>travellers[pos].destiny>>
        travellers[pos].price

```



```
        )
    {
        pos++;
    }
}

void write()
{
    ofstream out("passengers.txt",ios::out);
    if(out)
    {
        for(traveller t: travellers)
        {
            if(t.key!=-1)
                out<<t.key<<" "<<join(t.name)<<" "
                <<join(t.origin)<<" "
                <<join(t.destiny)<<" "
                <<t.price<<" ";
        }
        out.close();
    }
    else cout<<"Error al abrir el archivo";
}

string split(string s)
{
    for(int i=0;i<s.size();i++)
        if(s[i]=='#')s[i]=' ';
    return s;
}

string join(string s)
{
    for(int i=0;i<s.size();i++)
        if(s[i]==' ')s[i]=='#';
    return s;
}

int findPlace()
{
    for(int i=0;i<travellers.size();i++)
    {
        if(travellers[i].key==-1)return i;
    }
}
```




```
        cout<<"debug\n";
        return -1;
    }

    bool insertData(traveller temp)
    {
        int placeEmpty=findPlace();
        if(placeEmpty!=-1)
        {
            travellers[placeEmpty]=temp;
            return true;
        }
        else return false;
    }

    void getData()
    {
        cin.get();
        traveller temp;
        cout<<"Introduzca el nombre del pasajero: ";
        getline(cin,temp.name);
        cout<<"Introduzca la clave del pasajero: ";
        cin>>temp.key;
        cin.get();
        cout<<"Ahora introduzca el origen: ";
        getline(cin,temp.origin);
        cout<<"Introduzca el destino: ";
        getline(cin,temp.destiny);
        cout<<"Finalmente introduzca el precio del viaje: ";
        cin>>temp.price;
        cin.get();
        if(insertData(temp))
        {
            write();
            cout<<"\nRegistro exitoso!\n";

        } else cout<<"\nNo hay lugares\n";
    }

    void setup()
    {
        traveller empty;
        empty.key=-1;
    }
}
```



```

empty.name="";
empty.origin="";
empty.destiny="";
empty.price=0;
for(int i=0;i<travellers.size();i++)
{
    travellers[i]=empty;
}
}

void makeUserPage()
{
    fstream out("travellers.html",ios::out);
    out<<"<!DOCTYPE html>\n";
    out<<"<html lang='en'>\n";
    out<<"<head>\n";
    out<<"<meta charset='UTF-8'>\n";
    out<<"<meta name='viewport' content='width=device-width,";
    out<<"user-scalable=no,    initial-scale=1.0,maximum-scale=1.0,    minimum-
scale=1.0'>\n";
    out<<"<link                                rel='stylesheet'
href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css'";
    out<<"integrity='sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u'\n";
    out<<"crossorigin='anonymous'>\n";

    out<<"<title>Pagina de pasajeros</title>\n";
    out<<"</head>\n";
    out<<"<body>\n";
    out<<"<table class='table table-striped'>\n";
    out<<"<tr>\n";
    out<<"<th>Clave</th>\n";
    out<<"<th>Nombre</th>\n";
    out<<"<th>Origen</th>\n";
    out<<"<th>Destino</th>\n";
    out<<"<th>Precio</th>\n";
    out<<"</tr>\n";
    for(traveller t:travellers)
    {
        if(t.key!=-1){
            out<<"<tr>\n";
            out<<"<td>"<<t.key<<"</td>\n";
            out<<"<td>"<<split(t.name)<<"</td>\n";
            out<<"<td>"<<split(t.origin)<<"</td>\n";
            out<<"<td>"<<split(t.destiny)<<"</td>\n";

```



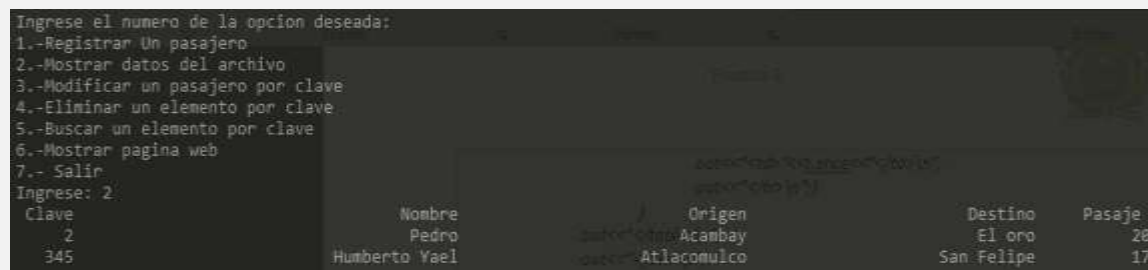
```

        out<<"<td>"<<t.price<<"</td>\n";
        out<<"</tr>\n";}

    }
    out<<"</table>\n";
    out<<"</div>\n";
    out<<"<script
src='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js'\n";
    out<<"integrity='sha384-Tc5lQib027qvyjSMfHjOMaLkf'\n";
    out<<"uWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNlcpPD7Txa '\n";
    out<<"crossorigin='anonymous'>\n";
    out<<"</script>\n";
    out<<"</body>\n";
    out<<"</html>\n";
    }

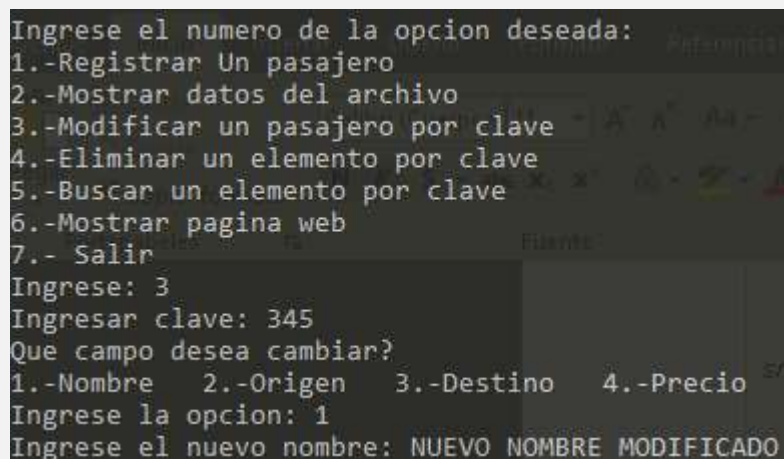
```

Capturas de pantalla



Página mostrada

Clave	Nombre	Origen	Destino	Precio
2	Pedro	Acambay	El oro	20
345	Humberto Yael	Atlacomulco	San Felipe	17



***Página después de modificar***

Clave	Nombre	Origen
2	Pedro	Acambay
345	NUEVO NOMBRE MODIFICADO	Atlacomulco

Eliminando un elemento

```
Ingrese el numero de la opcion deseada:
1.-Registrar Un pasajero
2.-Mostrar datos del archivo
3.-Modificar un pasajero por clave
4.-Eliminar un elemento por clave
5.-Buscar un elemento por clave
6.-Mostrar pagina web
7.- Salir
Ingrese: 4
Ingresar clave: 345
Pasajero eliminado!
```

Página después de eliminar

Clave	Nombre	Origen	Destino	Precio
2	Pedro	Acambay	El oro	20

Buscando un elemento



```
Ingrese el numero de la opcion deseada:
1.-Registrar Un pasajero
2.-Mostrar datos del archivo
3.-Modificar un pasajero por clave
4.-Eliminar un elemento por clave
5.-Buscar un elemento por clave
6.-Mostrar pagina web
7.- Salir
Ingrese: 5
Ingrese clave: 2
Que tipo de busqueda desea realizar?
1.-Lineal  2.-Binaria
Ingrese: 2
Elemento encontrado en la posicion 0
Pedro
Acambay
El#oro
20
```

student.cpp

```
/*
Programa de registro de estudiantes
Elias Edgardo Segundo Antonio
*/
#include <iostream>
#include <vector>
#include <iomanip>
#include <fstream>
#include <stdlib.h>
using namespace std;

struct student{
    string name;
    int key;
    double notes[2];
    double note;
};

vector<student> students(10);

//Prototype section
void makeUserPage();
void debug();
int findT(int);
void sort();
```



```
void setup();
void read();
void showData();
void getData();
void sort();
bool insertData(student);
int findPlace();
string join(string);
string split(string);
void write();
void menu();
bool updateStudent(int);
bool eraseStudent(int);
int findStudent(int);

int main()
{
    setup();
    int option,key;
    read();
    sort();
    while(option!=7)
    {
        cout<<"\n\n\n";
        menu();
        cin>>option;
        switch(option)
        {
            case 1:
                getData();
                break;
            case 2:
                sort();
                write();
                showData();
                break;
            case 3:
                cout<<"Ingresar clave: ";
                cin>>key;
                updateStudent(key);
                break;
            case 4:
                cout<<"Ingresar clave: ";
                cin>>key;
                if(eraseStudent(key))
```



```

        {
            cout<<"Pasajero eliminado!\n";
        }
        else cout<<"Error al borrar estudiante\n";
        break;
    case 5:
        cout<<"Ingrese clave: ";
        cin>>key;
        findT(key);
        break;
    case 6:
        makeUserPage();
        system("students.html");
        break;
    default:
        if(option!=7)
            cout<<"\nIngrese una opcion valida\n";
    }
}

int findT(int key)
{
    int option;
    cout<<"Que tipo de busqueda desea realizar?\n";
    cout<<"1.-Lineal  2.-Binaria\n";
    cout<<"Ingrese: ";
    cin>>option;
    switch(option)
    {
        case 1:
            for(int i=0;i<students.size();i++)
            {
                if(students[i].key==key)
                {
                    cout<<"Elemento encontrado en la posicion "<<i<<"\n";
                    cout<<students[i].name<<"\n";
                    cout<<students[i].notes[0]<<"\n";
                    cout<<students[i].notes[1]<<"\n";
                    cout<<students[i].note<<"\n";
                    return 1;
                }
            }
        }
    }
    return -1;
}

```



```

        break;
    case 2:
        sort();
        int right=0, left=0, m;
        while(students[right].key!=-1) right++;
        while(left<=right)
        {
            m=int((left+right)/2);
            if(students[m].key==key)
            {
                cout<<"Elemento encontrado en la posicion "<<m<<"\n";
                cout<<students[m].name<<"\n";
                cout<<students[m].notes[0]<<"\n";
                cout<<students[m].notes[1]<<"\n";
                cout<<students[m].note<<"\n";
                return 1;
            }
            if(students[m].key>key) right=m-1;
            else left=m+1;
        }
        return -1;
        break;
    }
}

void sort()
{
    for(int i=0; i<students.size(); i++)
    {
        for(int j=0; j<students.size(); j++)
        {
            if(students[i].key<students[j].key)
            {
                if(students[i].key!=-1)
                    swap(students[i], students[j]);
            }
        }
    }
}

bool updateStudent(int key)
{
    int index=-1;
    for(int i=0; i<students.size(); i++)
    {

```




```

        if(students[i].key==key)
            index=i;
    }
    if(index!=-1)
    {
        int option;
        cout<<"Que campo desea cambiar?\n"
        <<"1.-Nombre 2.-Nota primer parcial 3.-Nota segundo parcial\n";
        cout<<"Ingrese la opcion: ";
        cin>>option;
        switch(option)
        {
            case 1:
                cin.get();
                cout<<"Ingrese el nuevo nombre: ";
                getline(cin,students[index].name);
                break;
            case 2:
                cout<<"Ingrese la nueva nota: ";
                cin>>students[index].notes[0];
                break;
            case 3:
                cin.get();
                cout<<"Ingrese la nueva nota: ";
                cin>>students[index].notes[1];
                break;
        }
        students[index].note=(students[index].notes[0]+students[index].notes[1])/2;
        write();
    }
    else return false;
}

void menu()
{
    cout<<"Ingrese el numero de la opcion deseada: \n"
    <<"1.-Registrar un estudiante\n"
    <<"2.-Mostrar datos del archivo\n"
    <<"3.-Modificar un estudiante por clave\n"
    <<"4.-Eliminar un elemento por clave\n"
    <<"5.-Buscar un elemento por clave\n"
    <<"6.-Mostrar pagina web\n"
    <<"7.- Salir\n"
    <<"Ingrese: ";
}

```



```

bool eraseStudent(int key)
{
    int index=-1;
    for(int i=0;i<students.size();i++)
    {
        if(students[i].key==key)
            index=i;
    }
    if(index!=-1)
    {
        students.erase(students.begin()+index);
        write();
        return true;
    }
    else return false;
}

void showData()
{
    read();
    cout<<setw(6)<<"Clave"<<setw(40)<<"Nombre"<<
    setw(30)<<"Nota 1"<<setw(30)<<"Nota 2"<<setw(12)<<"Calificacion final\n";
    for(student s: students)
    {
        if(s.key!=-1)
        {
            cout<<setw(6);
            cout<<s.key;
            cout<<setw(40);
            cout<<split(s.name);
            cout<<setw(30);
            cout<<s.notes[0];
            cout<<setw(30);
            cout<<s.notes[1];
            cout<<setw(12);
            cout<<s.note<<" \n";
        }
    }
}

void read()
{
    setup();
}

```



```

    ifstream in("students.txt",ios::in);
    int pos=0;
    while(in>>students[pos].key>>students[pos].name>>
        students[pos].notes[0]>>students[pos].notes[1]>>
        students[pos].note
        )
    {
        pos++;
    }
}

void write()
{
    ofstream out("students.txt",ios::out);
    if(out)
    {
        for(student s: students)
        {
            if(s.key!=-1)
                out<<s.key<<" "<<join(s.name)<<" "
                <<s.notes[0]<<" "<<s.notes[1]<<" "<<s.note<<" ";
        }
        out.close();
    }
    else cout<<"Error al abrir el archivo";
}

string split(string s)
{
    for(int i=0;i<s.size();i++)
        if(s[i]=='#')s[i]=' ';
    return s;
}

string join(string s)
{
    for(int i=0;i<s.size();i++)
        if(s[i]==' ')s[i]='#';
    return s;
}

int findPlace()
{
    for(int i=0;i<students.size();i++)

```



```
{
    if(students[i].key==-1)return i;
}
return -1;
}

bool insertData(student temp)
{
    int placeEmpty=findPlace();
    if(placeEmpty!=-1)
    {
        students[placeEmpty]=temp;
        return true;
    }
    else return false;
}

void getData()
{
    cin.get();
    student temp;
    cout<<"Introduzca la clave del estudiante: ";
    cin>>temp.key;
    cin.get();
    cout<<"Introduzca el nombre del estudiante: ";
    getline(cin,temp.name);
    cout<<"Introduzca la calificacion del primer parcial: ";
    cin>>temp.notes[0];
    cout<<"Introduzca la calificacion del segundo parcial: ";
    cin>>temp.notes[1];
    temp.note=(temp.notes[0]+temp.notes[1])/2;
    if(insertData(temp))
    {
        write();
        cout<<"\nRegistro exitoso!\n";
    } else cout<<"\nNo hay lugares\n";
}

void setup()
{
    student empty;
    empty.key=-1;
    empty.name="";
}
```



```

empty.notes[0]=0;
empty.notes[1]=0;
empty.note=0;
for(int i=0;i<students.size();i++)
{
    students[i]=empty;
}
}

void makeUserPage()
{
    ofstream out("students.html",ios::out);
    out<<"!DOCTYPE html>\n";
    out<<"<html lang='en'>\n";
    out<<"<head>\n";
    out<<"<meta charset='UTF-8'>\n";
    out<<"<meta name='viewport' content='width=device-width,
    out<<"user-scalable=no,    initial-scale=1.0,maximum-scale=1.0,    minimum-
scale=1.0'>\n";
    out<<"<link                                rel='stylesheet'
href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css'";
    out<<"integrity='sha384-
BVYiISiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u'\n";
    out<<"crossorigin='anonymous'>\n";

    out<<"<title>Pagina de estudiantes</title>\n";
    out<<"</head>\n";
    out<<"<body>\n";
    out<<"<table class='table table-striped'>\n";
    out<<"<tr>\n";
    out<<"<th>Clave</th>\n";
    out<<"<th>Nombre</th>\n";
    out<<"<th>Primer parcial</th>\n";
    out<<"<th>Segundo Parcial</th>\n";
    out<<"<th>Promedio</th>\n";
    out<<"</tr>\n";
    for(student s:students)
    {
        if(s.key!=-1){
            out<<"<tr>\n";
            out<<"<td>"<<s.key<<"</td>\n";
            out<<"<td>"<<split(s.name)<<"</td>\n";
            out<<"<td>"<<s.notes[0]<<"</td>\n";
            out<<"<td>"<<s.notes[1]<<"</td>\n";
            out<<"<td>"<<s.note<<"</td>\n";
        }
    }
}

```



```

        out<<"</tr>\n";}
    }
    out<<"</table>\n";
    out<<"</div>\n";
    out<<"<script
src='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js'\n";
    out<<"integrity='sha384-Tc5IQib027qvyjSMfHjOMaLkf'\n";
    out<<"uWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNlcpPD7Txa '\n";
    out<<"crossorigin='anonymous'>\n";
    out<<"</script>\n";
    out<<"</body>\n";
    out<<"</html>\n";
}

```

Capturas de pantalla

Ingrese el numero de la opcion deseada:

- 1.-Registrar un estudiante
- 2.-Mostrar datos del archivo
- 3.-Modificar un estudiante por clave
- 4.-Eliminar un elemento por clave
- 5.-Buscar un elemento por clave
- 6.-Mostrar pagina web
- 7.- Salir

Ingrese: 2

Clave	Nombre	Nota 1	Nota 2	Calificacion final
333	Elias Segundo Antonio	10	9.56	9.78
444	Juan Lopez	9	0	4.5

Página mostrada

Clave	Nombre	Primer parcial	Segundo Parcial	Promedio
333	Elias Segundo Antonio	10	9.56	9.78
444	Juan Lopez	9	0	4.5

Modificando un campo



Ingrese el numero de la opcion deseada:

- 1.-Registrar un estudiante
- 2.-Mostrar datos del archivo
- 3.-Modificar un estudiante por clave
- 4.-Eliminar un elemento por clave
- 5.-Buscar un elemento por clave
- 6.-Mostrar pagina web
- 7.- Salir

Ingrese: 3

Ingresar clave: 333

Que campo desea cambiar?

- 1.-Nombre 2.-Nota primer parcial 3.-Nota segundo parcial

Ingrese la opcion: 2

Ingrese la nueva nota: 5.0

Página mostrada

Clave

333

444

Modificar la

Página después de modificar

Nombre	Primer parcial	Segundo Parcial	Promedio
Elias Segundo Antonio	5	9.56	7.28
Juan Lopez	9	0	4.5

Eliminando un elemento

Ingrese el numero de la opcion deseada:

- 1.-Registrar un estudiante
- 2.-Mostrar datos del archivo
- 3.-Modificar un estudiante por clave
- 4.-Eliminar un elemento por clave
- 5.-Buscar un elemento por clave
- 6.-Mostrar pagina web
- 7.- Salir

Ingrese: 4

Ingresar clave: 444

Página después de eliminar

Clave	Nombre	Primer parcial	Segundo Parcial	Promedio
333	Elias Segundo Antonio	5	9.56	7.28

Buscando un elemento



```
Ingrese el numero de la opcion deseada:
1.-Registrar un estudiante
2.-Mostrar datos del archivo
3.-Modificar un estudiante por clave
4.-Eliminar un elemento por clave
5.-Buscar un elemento por clave
6.-Mostrar pagina web
7.- Salir
Ingrese: 5
Ingrese clave: 333
Que tipo de busqueda desea realizar?
1.-Lineal  2.-Binaria
Ingrese: 1
Elemento encontrado en la posicion 0
Elias#Segundo#Antonio
5
9.56
7.28
```

book.cpp

```
/*
Programa de registro de libros
Elias Edgardo Segundo Antonio
*/
#include <iostream>
#include <vector>
#include <iomanip>
#include <fstream>
#include <stdlib.h>
using namespace std;

struct book{
    string title;
    int key;
    string auth;
    int stock;
};

vector<book> books(10);

//Prototype section
void makeUserPage();
void debug();
int findT(int);
void sort();
```




```
void setup();
void read();
void showData();
void getData();
void sort();
bool insertData(book);
int findPlace();
string join(string);
string split(string);
void write();
void menu();
bool updateBook(int);
bool eraseBook(int);
int findBook(int);

int main()
{
    setup();
    int option,key;
    read();
    sort();
    while(option!=7)
    {
        cout<<"\n\n\n";
        menu();
        cin>>option;
        switch(option)
        {
            case 1:
                getData();
                break;
            case 2:
                sort();
                write();
                showData();
                break;
            case 3:
                cout<<"Ingresar clave: ";
                cin>>key;
                updateBook(key);
                break;
            case 4:
                cout<<"Ingresar clave: ";
                cin>>key;
                if(eraseBook(key))
```



```

        {
            cout<<"Libro eliminado!\n";
        }
        else cout<<"Error al borrar libro\n";
        break;
    case 5:
        cout<<"Ingrese clave: ";
        cin>>key;
        findT(key);
        break;
    case 6:
        makeUserPage();
        system("books.html");
        break;
    default:
        if(option!=7)
            cout<<"\nIngrese una opcion valida\n";
    }
}

int findT(int key)
{
    int option;
    cout<<"Que tipo de busqueda desea realizar?\n";
    cout<<"1.-Lineal  2.-Binaria\n";
    cout<<"Ingrese: ";
    cin>>option;
    switch(option)
    {
        case 1:
            for(int i=0;i<books.size();i++)
            {
                if(books[i].key==key)
                {
                    cout<<"Elemento encontrado en la posicion "<<i<<"\n";
                    cout<<books[i].title<<"\n";
                    cout<<books[i].auth<<"\n";
                    cout<<books[i].stock<<"\n";
                    return 1;
                }
            }
        }
        return -1;
        break;
    }
}

```



```

        case 2:
            sort();
            int right=0, left=0, m;
            while(books[right].key!=-1) right++;
            while(left<=right)
            {
                m=int((left+right)/2);
                if(books[m].key==key)
                {
                    cout<<"Elemento encontrado en la posicion "<<m<<"\n";
                    cout<<books[m].title<<"\n";
                    cout<<books[m].auth<<"\n";
                    cout<<books[m].stock<<"\n";
                    return 1;
                }
                if(books[m].key>key) right=m-1;
                else left=m+1;
            }
            return -1;
            break;
    }
}

void sort()
{
    for(int i=0; i<books.size(); i++)
    {
        for(int j=0; j<books.size(); j++)
        {
            if(books[i].key<books[j].key)
            {
                if(books[i].key!=-1)
                    swap(books[i], books[j]);
            }
        }
    }
}

bool updateBook(int key)
{
    int index=-1;
    for(int i=0; i<books.size(); i++)
    {
        if(books[i].key==key)
            index=i;
    }
}

```



```

    }
    if(index!=-1)
    {
        int option;
        cout<<"Que campo desea cambiar?\n"
        <<"1.-Nombre 2.-Autor 3.-Inventario\n";
        cout<<"Ingrese la opcion: ";
        cin>>option;
        switch(option)
        {
            case 1:
                cin.get();
                cout<<"Ingrese el nuevo nombre: ";
                getline(cin,books[index].title);
                break;
            case 2:
                cin.get();
                cout<<"Ingrese el nuevo: ";
                getline(cin,books[index].auth);
                break;
            case 3:
                cin.get();
                cout<<"Ingrese la nueva cantidad: ";
                cin>>books[index].stock;
                break;
        }
        write();
    }
    else return false;
}

void menu()
{
    cout<<"Ingrese el numero de la opcion deseada: \n"
    <<"1.-Registrar un libro\n"
    <<"2.-Mostrar datos del archivo\n"
    <<"3.-Modificar un libro por clave\n"
    <<"4.-Eliminar un elemento por clave\n"
    <<"5.-Buscar un elemento por clave\n"
    <<"6.-Mostrar pagina web\n"
    <<"7.- Salir\n"
    <<"Ingrese: ";
}

bool eraseBook(int key)

```



```
{
    int index=-1;
    for(int i=0;i<books.size();i++)
    {
        if(books[i].key==key)
            index=i;
    }
    if(index!=-1)
    {
        books.erase(books.begin()+index);
        write();
        return true;
    }
    else return false;
}

void showData()
{
    read();
    cout<<setw(6)<<"Clave"<<setw(40)<<"Titulo"<<
    setw(30)<<"Autor"<<setw(30)<<"Inventario"<<"\n";
    for(book b: books)
    {
        if(b.key!=-1)
        {
            cout<<setw(6);
            cout<<b.key;
            cout<<setw(40);
            cout<<split(b.title);
            cout<<setw(30);
            cout<<split(b.auth);
            cout<<setw(30);
            cout<<b.stock<<"\n";
        }
    }
}

void read()
{
    setup();
    ifstream in("books.txt",ios::in);
    int pos=0;
    while(in>>books[pos].key>>books[pos].title>>
        books[pos].auth>>books[pos].stock
```



```
        )
    {
        pos++;
    }
}

void write()
{
    ofstream out("books.txt",ios::out);
    if(out)
    {
        for(book b: books)
        {
            if(b.key!=-1)
                out<<b.key<<" "<<join(b.title)<<" "
                <<join(b.auth)<<" "<<b.stock<<" ";
        }
        out.close();
    }
    else cout<<"Error al abrir el archivo";
}

string split(string s)
{
    for(int i=0;i<s.size();i++)
        if(s[i]=='#')s[i]=' ';
    return s;
}

string join(string s)
{
    for(int i=0;i<s.size();i++)
        if(s[i]==' ')s[i]='#';
    return s;
}

int findPlace()
{
    for(int i=0;i<books.size();i++)
    {
        if(books[i].key==-1)return i;
    }
    return -1;
}
```



```
bool insertData(book temp)
{
    int placeEmpty=findPlace();
    if(placeEmpty!=-1)
    {
        books[placeEmpty]=temp;
        return true;
    }
    else return false;
}

void getData()
{
    cin.get();
    book temp;
    cout<<"Introduzca la clave del libro: ";
    cin>>temp.key;
    cin.get();
    cout<<"Introduzca el nombre del libro: ";
    getline(cin,temp.title);
    cout<<"Introduzca el autor del libro: ";
    getline(cin,temp.auth);
    cout<<"Introduzca el stock del libro: ";
    cin>>temp.stock;
    if(insertData(temp))
    {
        write();
        cout<<"\nRegistro exitoso!\n";
    } else cout<<"\nNo hay lugares\n";
}

void setup()
{
    book empty;
    empty.key=-1;
    empty.title="";
    empty.auth="";
    empty.stock=0;
    for(int i=0;i<books.size();i++)
    {
        books[i]=empty;
    }
}
```



```

}

void makeUserPage()
{
    fstream out("books.html",ios::out);
    out<<"<!DOCTYPE html>\n";
    out<<"<html lang='en'>\n";
    out<<"<head>\n";
    out<<"<meta charset='UTF-8'>\n";
    out<<"<meta title='viewport' content='width=device-width, ";
    out<<"user-scalable=no,    initial-scale=1.0,maximum-scale=1.0,    minimum-
scale=1.0'>\n";
    out<<"<link                                rel='stylesheet'
href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css'";
    out<<"integrity='sha384-
BVYiSiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u'\n";
    out<<"crossorigin='anonymous'>\n";

    out<<"<title>Pagina de libros</title>\n";
    out<<"</head>\n";
    out<<"<body>\n";
    out<<"<table class='table table-striped'>\n";
    out<<"<tr>\n";
    out<<"<th>Clave</th>\n";
    out<<"<th>Nombre</th>\n";
    out<<"<th>Autor</th>\n";
    out<<"<th>Stock</th>\n";
    out<<"</tr>\n";
    for(book b:books)
    {
        if(b.key!=-1){
            out<<"<tr>\n";
            out<<"<td>"<b.key<<"</td>\n";
            out<<"<td>"<<split(b.title)<<"</td>\n";
            out<<"<td>"<<split(b.auth)<<"</td>\n";
            out<<"<td>"<b.stock<<"</td>\n";
            out<<"</tr>\n";}
        }
    out<<"</table>\n";
    out<<"</div>\n";
    out<<"<script
src='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js'\n";
    out<<"integrity='sha384-Tc5IQib027qvyjSMfHjOMaLkf'\n";
    out<<"uWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNlcpPD7Txa '\n";
    out<<"crossorigin='anonymous'>\n";

```

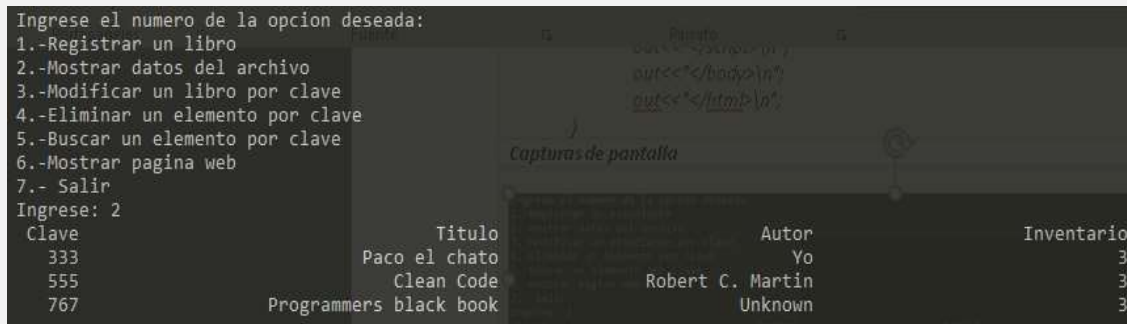



```

    out<<"</script>\n";
    out<<"</body>\n";
    out<<"</html>\n";
}

```

Capturas de pantalla



Página mostrada

Clave	Nombre	Autor	Stock
333	Paco el chato	Yo	3
555	Clean Code	Robert C. Martin	3
767	Programmers black book	Unknown	3

Modificando un campo

```

Ingrese el numero de la opcion deseada:
1.-Registrar un libro
2.-Mostrar datos del archivo
3.-Modificar un libro por clave
4.-Eliminar un elemento por clave
5.-Buscar un elemento por clave
6.-Mostrar pagina web
7.- Salir
Ingrese: 3
Ingresar clave: 333
Que campo desea cambiar?
1.-Nombre 2.-Autor 3.-Inventario
Ingrese la opcion: 2
Ingrese el nuevo: SEP

```

Página después de modificar



Nombre	Autor
Paco el chato	SEP
Clean Code	Robert C. Martin
Programmers black book	Unknown

Eliminando un elemento

```
Ingrese el numero de la opcion deseada:
1.-Registrar un libro
2.-Mostrar datos del archivo
3.-Modificar un libro por clave
4.-Eliminar un elemento por clave
5.-Buscar un elemento por clave
6.-Mostrar pagina web
7.- Salir
Ingrese: 4
Ingresar clave: 333
Libro eliminado!
```

Página después de eliminar

Clave	Nombre	Autor
555	Clean Code	Robert C. Martin
767	Programmers black book	Unknown

Buscando un elemento



```
Ingrese el numero de la opcion deseada:
1.-Registrar un libro
2.-Mostrar datos del archivo
3.-Modificar un libro por clave
4.-Eliminar un elemento por clave
5.-Buscar un elemento por clave
6.-Mostrar pagina web
7.- Salir
Ingrese: 5
Ingrese clave: 767
Que tipo de busqueda desea realizar?
1.-Lineal  2.-Binaria
Ingrese: 2
Elemento encontrado en la posicion 1
Programmers#black#book
Unknown
3
```