



UAEM | Universidad Autónoma
del Estado de México
Centro universitario UAEM Atlacomulco



Programación avanzada

Proyecto segundo parcial

Nombre: Gustavo Blas Duran

Grupo: ICO-22

Docente: Ing. José Luis García Morales

Mayo, 2018



Universidad Autónoma del Estado de México

```

/* Tienda de Musica */
#include <iostream>
#include <string.h>
#include <stdlib.h>
#include <iomanip>
#include <fstream>

#define USER "usuario"/*nombre de usuario*/
#define PASS1 "msd9bn45"/*contraseña de usuario*/

#define ADMIN "GUSTAVOBD"/*nombre de administrador*/
#define PASS2 "gblasd239"/*contraseña de administrador*/

using namespace std;
const int NUM = 20;//numero maximo de canciones

struct Usuario{
    string username;
    string password;
    double SALDO;
};
Usuario usuario;

struct ListaCanciones {/* canciones del catalogo */
    int clave;
    string nombreCancion;
    string nombreCantante;
    string genero;
    string portada;
    string albumArchivo;
    double precio;
};
ListaCanciones listaCanciones[NUM];

void insertar(int, string, string, string, string, string, double);
void mostrarDatos();
void escribirEnArchivo();
void recuperarDeArchivo();
void pedirDatos();
void inicializar();
int buscarLugarVacio();
void menu();
void escribirPaginaWeb();
void ordenacionBurbujaPorClaveCancion();
void ordenacionBurbujaPorNombreCancion();
int busquedaSecuencial(int);
int busquedaBinaria(int);
void mostrarRegistro(int);
void ordenacionBurbujaPorClave();
void borrar(int);
void cambiar(int);
void menuCambiar();

void inicializarSaldo();
void insertarSaldo(int);
void escribirEnArchivoMiSaldo();
void agregarSaldo();
double restarSaldo(double);
void obtenerSaldo();
void comprarCancion();
void extraerDatos(int);

string reemplazarEspacios(string);

```

```

string reemplazarSharp(string);
void menuPrincipal();
void ADMINISTRADOR ();
void USUARIO ();

struct ListaCancionesCompradas { /*canciones compradas del catalogo*/
    int claveCompradas;
    string nombreCancionCompradas;
    string nombreCantanteCompradas;
    string generoCompradas;
    string portadaCompradas;
    string albumArchivoCompradas;
    double precioCompradas;
};
ListaCancionesCompradas listaCancionesCompradas[NUM];

string reemplazarEspacios(string);
string reemplazarSharp(string);
void insertarCompradas(int, string, string, string, string, string, double);
void mostrarDatosCompradas();
void escribirEnArchivoCompradas();
void recuperarDeArchivoCompradas();
void pedirDatosCompradas();
void inicializarCompradas();
int buscarLugarVacioCompradas();
void escribirPaginaWebCompradas();
void mostrarRegistroCompradas(int);
int busquedaSecuencialCompradas(int);
int busquedaBinariaCompradas(int);
void ordenacionBurbujaPorClaveCompradas();

int main () {

    int option;
    do {
        cout << "                MusicStore                " << endl;
        menuPrincipal();
        cout << "? " ;
        cin >> option;
        switch(option) {
            case 1: //////////////////////////////////// * ACCIONES PARA EL ADMINISTRADOR * /
                //string usuario, contrasenia;

                bool validarAccesoAdministrador ;
                do {
                    string usuarioA, contraseniaA;
                    system("cls");
                    cout << "ADMINISTRADOR" << endl;
                    cout << "USERNAME: ";
                    getline(cin, usuarioA);
                    getline(cin, usuarioA);
                    cout << endl;
                    cout << "PASSWORD: ";
                    getline(cin, contraseniaA);
                    if(usuarioA == ADMIN && contraseniaA == PASS2) {
                        validarAccesoAdministrador = true;
                    } else {
                        validarAccesoAdministrador = false;
                        cout << endl << endl;
                        cout << "contrasenia y/o usuario son incorrectos " << endl;
                        cout << "intente nuevamente "<< endl;
                    }
                } while (!validarAccesoAdministrador);
            }
        }
    } while (option != 0);
}

```

```

        system("pause");
        system("cls");
    }
    }while(validarAccesoAdministrador != true);
    system("cls");
    ADMINISTRADOR();
    system("cls");
    break;
case 2://///////////////////////* ACCIONES PARA EL USUARIO */

    bool validarAcceso ;
    do{
        system("cls");
        string usuario, contrasenia;
        cout << "USUARIO" << endl;
        cout << "USERNAME:  ";
        getline(cin, usuario);
        getline(cin, usuario);
        cout << endl;
        cout << "PASSWORD:  ";
        getline(cin, contrasenia);
        if(usuario == USER && contrasenia == PASS1){
            validarAcceso = true;
        }else{
            validarAcceso = false;
            cout << endl << endl;
            cout << "contrasenia y/o usuario son incorrectos " << endl;
            cout << "intente nuevamente " << endl;
            system("pause");
            system("cls");
        }
    }while(validarAcceso != true);
    system("cls");
    USUARIO();
    system("cls");
    break;
case 3:
    system("cls");
    cout << setw(100) << "Saliendo....." << endl;
    break;
default:
    system("cls");
    break;

    }//fin de case
}while (option != 3);
return 0;
} //fin de main

void menuPrincipal(){
    cout << endl;
    cout << "Inserte la opcion:  " << endl;
    cout << endl;
    cout << "1) INICIAR SESION COMO ADMINISTRADOR  " << endl;
    cout << "2) INICIAR SESION COMO USUARIO " << endl;
    cout << "3) SALIR  " << endl;
    cout << endl;
}

```

```

void ADMINISTRADOR() {
    inicializar();
    recuperarDeArchivo();
    int opcion, resultado;
    do {
        menu();
        cout << "?";
        cin >> opcion;
        switch (opcion) {
            case 1:

                system("cls");
                pedirDatos();
                //ordenacionBurbujaPorNombreCancion();
                escribirEnArchivo();
                system("pause");
                system("cls");
                break;
            case 2:
                system("cls");
                mostrarDatos();
                system("pause");
                system("cls");
                break;
            case 3:

                system("cls");
                escribirPaginaWeb();
                system("INDEX.html");//manda llamar al archivo y lo abre en el navegador
                system("pause");
                system("cls");
                break;
            case 4:
                system("cls");
                ordenacionBurbujaPorClaveCancion();
                mostrarDatos();
                system("pause");
                system("cls");
                break;
            case 5:
                system("cls");
                ordenacionBurbujaPorNombreCancion();
                mostrarDatos();
                system("pause");
                system("cls");
                break;
            case 6://BUSQUEDA SECUENCIAL
                system("cls");
                int buscar;
                cout << "Dame clave de la Cancion a buscar: " << endl;
                cin >> buscar;
                resultado = busquedaSecuencial(buscar);
                if(resultado == -1){
                    cout << "No encontrado " << endl;
                }else {
                    mostrarRegistro(resultado);
                }
                system("pause");
                system("cls");
                break;
        }
    } while (opcion != 0);
}

```

```

    case 7: { //BUSQUEDA NINARIA
        system("cls");
        ordenacionBurbujaPorClave();
        int buscarN;
        cout << "dame clave de la Cancion a buscar" << endl;
        cin >> buscarN;
        resultado = busquedaBinaria(buscarN);
        if (resultado == -1) {
            cout << "No encontrado" << endl;
        } else {
            mostrarRegistro(resultado);
        }
        system("pause");
        system("cls");
        break; }
    case 8:
        system("cls");
        mostrarDatos();
        int claveB;
        cout << "Dame la clave del elemento a borrar: ";
        cin >> claveB;
        borrar(claveB);
        system("pause");
        system("cls");
        break;
    case 9:
        system("cls");
        int claveM;
        cout << "Dame la clave del elemento a modificar: ";
        cin >> claveM;
        cambiar(claveM);
        system("pause");
        system("cls");
        break;
    case 10:
        system("cls");
        mostrarDatos();
        int baja;
        cout << "inserta la clave de la cancion para dar de baja" << endl;
        cin >> baja;
        borrar(baja);
        break;
    case 11:
        break;
    default:
        system("cls");
        cout << "Opcion no valida" << endl;
        cout << "Intente nuevamente...." << endl;
        system("pause");
        system("cls");

        break;
} //fin de switch
} while (opcion != 11);
}

string reemplazarEspacios(string texto) {
    for (int i = 0; i < (int) texto.length(); ++i) {
        if (texto[i] == ' ') {
            texto.replace(i, 1, "#");
        }
    }
    return texto;
}

```

```

string reemplazarSharp(string texto) {
    for (int i = 0; i < (int) texto.length(); ++i) {
        if (texto[i] == '#') {
            texto.replace(i, 1, " ");
        }
    }
    return texto;
}

void menu() {
    cout << "\n      ADMINISTRADOR  " << endl;
    cout << "E L I G E   T U   O P C I O N" << endl;
    cout << "1.-ALTA DE CANCIONES" << endl;
    cout << "2.-REPORTE GENERAL" << endl;
    cout << "3.-VER PAGINA WEB" << endl;
    cout << "4.-ORDENAR POR CLAVE DE CANCION" << endl;
    cout << "5.-ORDENAR POR NOMBRE CANCION" << endl;
    cout << "6.-BUSCAR POR CLAVE DEL CANCION (SECUENCIAL)" << endl;
    cout << "7.-BUSCAR POR CLAVE DEL CANCION (BINARIO)" << endl;
    cout << "8.-BORRAR UN ELEMENTO" << endl;
    cout << "9.-EDITAR UN ELEMENTO" << endl;
    cout << "10.-DAR DE BAJA UNA CANCION" << endl;
    cout << "11.-SALIR " << endl;
}

void pedirDatos() {
    int clavex;
    string nombreCancionx;
    string nombreCantantex;
    string generox;
    string portadax;
    string albumArchivox;
    double preciox;

    cout << "Escribe datos del las Canciones. " << endl;

    int resultado;
    bool validarClave = true;
    do{
        cout << endl;
        validarClave = true;
        cout << "Inserta clave:";
        cin>>clavex;
        resultado = busquedaSecuencial(clavex);
        if(resultado == 1){
            validarClave = false;
            cout << endl;
            cout << "                ERROR :(   clave repetida ..." << endl;
            cout << "vuelva a insertar otra clave diferente " << endl;
        }
        else{
            validarClave = true;
        }
    }while (validarClave != true);
}

```

```

    cout << "Inserta nombre la Cancion:";
    getline(cin, nombreCancionx);
    getline(cin, nombreCancionx);
    cout << "\nInserta nombre del Cantante:";
    getline(cin, nombreCantantex);
    cout << "\nInserta genero:";
    getline(cin, generox);
    cout << "Inserta portada: ";
    getline(cin, portadax);
    cout << "Inserta album Archivo: ";
    getline(cin, albumArchivox);
    cout << "\nInserta costo de la cancion:";
    cin>>preciox;
    insertar(clavex, nombreCancionx, nombreCantantex, generox, portadax, albumArchivox,
    preciox);
}

```

```

void insertar(int clavex, string nombreCancionx, string nombreCantantex,
    string generox, string portadax, string albumArchivox, double preciox) {
    int lugarVacio = buscarLugarVacio();
    if (lugarVacio == -1) {
        cout << "Ya no hay lugar" << endl;
        return;
    }
    listaCanciones[lugarVacio].clave = clavex;
    listaCanciones[lugarVacio].nombreCancion = nombreCancionx;
    listaCanciones[lugarVacio].nombreCantante = nombreCantantex;
    listaCanciones[lugarVacio].genero = generox;
    listaCanciones[lugarVacio].portada = portadax;
    listaCanciones[lugarVacio].albumArchivo = albumArchivox;
    listaCanciones[lugarVacio].precio = preciox;
}

```

```

void mostrarDatos() {
    cout << endl;
    cout << setw(6) << "CLAVE"
        << setw(20) << "nombreCancion"
        << setw(15) << "nombreCantante"
        << setw(15) << "genero"
        << setw(15) << "portada"
        << setw(15) << "albumArchivo"
        << setw(7) << "precio" << endl;
    for (int i = 0; i < NUM; i++) {
        if (listaCanciones[i].clave != 0) {
            cout << setw(6) << listaCanciones[i].clave
                << setw(20) << listaCanciones[i].nombreCancion
                << setw(15) << listaCanciones[i].nombreCantante
                << setw(15) << listaCanciones[i].genero
                << setw(15) << listaCanciones[i].portada
                << setw(15) << listaCanciones[i].albumArchivo
                << setw(7) << listaCanciones[i].precio
                << endl;
        }
    }
}

```



```

void recuperarDeArchivo() {
    ifstream lecturaArchivo("listaCanciones.txt", ios::in);
    if (!lecturaArchivo) {
        cerr << "No se pudo abrir el archivo" << endl;
        exit(1);
    }
    int clavex;
    string nombreCancionx;
    string nombreCantantex;
    string generox;
    string portadax;
    string albumArchivox;
    double preciox;

    while (lecturaArchivo >> clavex >> nombreCancionx >> nombreCantantex >> generox
    >> portadax >> albumArchivox >> preciox) {
        insertar(clavex, reemplazarSharp(nombreCancionx),
reemplazarSharp(nombreCantantex), reemplazarSharp(generox), reemplazarSharp(portadax),
reemplazarSharp(albumArchivox), preciox);
    }
}

void escribirEnArchivo() {
    ofstream salidaArchivo("listaCanciones.txt", ios::out);
    if (!salidaArchivo) {
        cerr << "No se pudo abrir archivo" << endl;
        exit(1);
    }
    for (int i = 0; i < NUM; i++) {
        if (listaCanciones[i].clave != 0) {
            salidaArchivo << listaCanciones[i].clave << ' '
                << reemplazarEspacios(listaCanciones[i].nombreCancion) << ' '
                << reemplazarEspacios(listaCanciones[i].nombreCantante) << ' '
                << reemplazarEspacios(listaCanciones[i].genero) << ' '
                << reemplazarEspacios(listaCanciones[i].portada) << ' '
                << reemplazarEspacios(listaCanciones[i].albumArchivo) << ' '
                << listaCanciones[i].precio << ' ' << endl;
        }
    }
}

void inicializar() {
    for (int i = 0; i < NUM; i++) {
        listaCanciones[i].clave = 0;
        listaCanciones[i].nombreCancion = "";
        listaCanciones[i].nombreCantante = "";
        listaCanciones[i].genero = "";
        listaCanciones[i].portada = "";
        listaCanciones[i].albumArchivo = "";
        listaCanciones[i].precio = 0;
    }
}

int buscarLugarVacio() {
    for (int i = 0; i < NUM; i++) {
        if (listaCanciones[i].clave == 0) {
            return i;
        }
    }
    return -1;
}

```

```

void escribirPaginaWeb() {
    ofstream salidaArchivo("INDEX.html", ios::out);
    if (!salidaArchivo) {
        cerr << "No se pudo abrir el archivo" << endl;
        exit(1);
    }
    salidaArchivo << "<!DOCTYPE html>" << endl;
    salidaArchivo << "<html>" << endl;
    salidaArchivo << "<head>" << endl;
    salidaArchivo << "<meta charset='utf-8'>" << endl;
    salidaArchivo << "<title>MusicStore.mx</title>" << endl;
    salidaArchivo << " <meta name='Gustavo ICO-22' content='Mi tienda virtual de musica'>" << endl;
    salidaArchivo << " <link rel='stylesheet' href='main.css'>" << endl;
    salidaArchivo << " <style type='text/css'>" << endl;
    salidaArchivo << "table{border: red dotted medium; margin: auto; vertical-align: sub}" <<
endl;
    salidaArchivo << "td{border: blue dashed medium; margin: auto}" << endl;
    salidaArchivo << " article{margin:0% }" << endl;
    salidaArchivo << " section{margin: 5px;height: auto;width: 1250px}" << endl;
    salidaArchivo << "h1{" << endl;
    salidaArchivo << "border-color: orangered;" << endl;
    salidaArchivo << "border-style: solid;" << endl;
    salidaArchivo << "margin: auto;" << endl;
    salidaArchivo << "height: 95px;" << endl;
    salidaArchivo << "font-size: 90px;" << endl;
    salidaArchivo << "font-family: Segoe Ui Light;" << endl;
    salidaArchivo << "}" << endl;
    salidaArchivo << "</style>" << endl;
    salidaArchivo << "</head>" << endl;
    salidaArchivo << "<body>" << endl;
    salidaArchivo << " <header><h1>MusicStore</h1></header>" << endl;
    salidaArchivo << " <nav><marquee WIDTH=60% HEIGHT=60 ALIGN=BOTTOM
BEHAVIOR=ALTERNATE><h3>BIENVENIDO Si buscas canciones, estas en el lugar
CORRECTO</h3></marquee></nav>" << endl;
    salidaArchivo << " <section>" << endl;
    //aquí empieza la tabla
    salidaArchivo << "<table border = 1>" << endl;
    salidaArchivo << "<thead> <tr>" << endl;
    salidaArchivo << "<th> <h2>Clave</h2> </th><th> <h2>Nombre Cancion </h2></th>" << endl;
    salidaArchivo << "<th> <h2>Nombre Cantante </h2></th> <th> <h2>genero </h2></th>" << endl;
    salidaArchivo << "<th> <h2>Portada </h2></th> <th> <h2>AlbumArchivo </h2></th>" << endl;
    salidaArchivo << "<th> <h2>Costo </h2></th>" << endl;
    salidaArchivo << "</tr></thead>" << endl;
    salidaArchivo << "<t/body>" << endl;
    //cuerpo de la tabla
    for(int i=0; i < NUM; i++){
        if(listaCanciones[i].clave != 0){
            salidaArchivo << "<tr>"
                << "<td><h2>" << listaCanciones[i].clave << "</h2></td>"
                << "<td><h2>" << listaCanciones[i].nombreCancion << "</h2></td>"
                << "<td><h2>" << listaCanciones[i].nombreCantante << "</h2></td>"
                << "<td><h2>" << listaCanciones[i].genero << "</h2></td>"
                << "<td><img src='Portadas/'<< listaCanciones[i].portada << " " border='10'
width='150' height='150'></td>"
                << "<td><audio src='MusicaDEMO/'<< listaCanciones[i].albumArchivo << "
controls='controls' type='audio/mpeg' preload='preload'></audio></td>"
                << "<td> <h2>$ " << listaCanciones[i].precio << "</h2></td>"
                << "</tr>" << endl;
        }
    }
    salidaArchivo << "<t/body>" << endl;
    salidaArchivo << "</table>" << endl;
    //aquí finaliza la tabla
    salidaArchivo << " </section>" << endl;
    salidaArchivo << "<footer> <h2>By: Gustavo Blas Duran</h2></footer>" << endl;
    salidaArchivo << "</body>" << endl;
    salidaArchivo << "</html>" << endl;
} //fin de escribir pagina web

```

```
ListaCanciones aux;//auxiliar
```

```
void ordenacionBurbujaPorClaveCancion() {
    for(int i=0; i<NUM-1; i++){
        for(int j=0; j<NUM-1; j++){
            if(listaCanciones[j].clave > listaCanciones[j + 1].clave){
                //intercambio
                aux = listaCanciones[j];
                listaCanciones[j] = listaCanciones[j+1];
                listaCanciones[j+1] = aux;
            }
        }
    }
}

void ordenacionBurbujaPorNombreCancion() {
    for(int i=0; i<NUM-1; i++){
        for(int j=0; j<NUM-1; j++){
            if(listaCanciones[j].nombreCancion > listaCanciones[j + 1].nombreCancion){
                //intercambio
                aux = listaCanciones[j];
                listaCanciones[j] = listaCanciones[j+1];
                listaCanciones[j+1] = aux;
            }
        }
    }
}

int busquedaSecuencial(int b){
    for(int i=0; i<NUM; i++){
        if(listaCanciones[i].clave == b){
            return i;
        }
    }
    return -1;
}

int busquedaBinaria(int b){
    int bajo, alto, central;
    bajo = 0;
    alto = NUM - 1;
    while (bajo <= alto){
        central = (bajo + alto) / 2;
        if(b == listaCanciones[central].clave){
            return central;
        }else if(b < listaCanciones[central].clave){
            alto = central - 1;
        }else {
            bajo = central + 1;
        }
    }
    return -1;
}

void mostrarRegistro(int i){
    cout << "Nombre de la cancion...: " << listaCanciones[i].nombreCancion << endl;
    cout << "Nombre del cantante....: " << listaCanciones[i].nombreCantante << endl;
    cout << "Genero Actual.....: " << listaCanciones[i].genero << endl;
    cout << "Portada Actual.....: " << listaCanciones[i].portada << endl;
    cout << "album Archivo Actual....: " << listaCanciones[i].albumArchivo << endl;
    cout << "Costo Actual.....: " << listaCanciones[i].precio << endl;
}

//fin de registro
```

```

void ordenacionBurbujaPorClave() {
    for(int i=0; i < NUM-1; i++){
        for(int j = 0; j < NUM - 1; j++){
            if(listaCanciones[j].clave > listaCanciones[j + 1].clave){
                //intercambio
                aux = listaCanciones[j];
                listaCanciones[j] = listaCanciones[j + 1];
                listaCanciones[j + 1] = aux;
            }
        }
    }
}

void borrar (int clavex){
    for(int i = 0; i < NUM; i++){
        if(listaCanciones[i].clave == clavex){
            cout << "Desea eliminar esta canciÃ³n ? 1.- si , 2.- no" <<endl;
            cout << "1.- si :('" <<endl;
            cout << "2.- no :)" <<endl;
            int res;
            cin >>res;
            if(res == 1){
                listaCanciones[i].clave = 0;
                cout << "Elemento eliminado " << endl;
                escribirEnArchivo();
            }
            return;
        }
    }
    cout << "Elemento no encontrado" << endl;
}

void menuCambiar(){
    cout << endl;
    cout << "¿que desea cambiar?" << endl;
    cout << "1.- NombreCancion" << endl;
    cout << "2.- NombreCantante" << endl;
    cout << "3.- Genero" << endl;
    cout << "4.- Portada" << endl;
    cout << "5.- Album Archivo" << endl;
    cout << "6.- Costo de cancion" << endl;
    cout << "7.- Cancelar" << endl;
    cout << endl;
    cout << "Elija su opcion: " << endl;
}

//fin de menuCambiar
void cambiar(int clavex){
    for(int i = 0; i < NUM; i++){
        if(listaCanciones[i].clave == clavex){
            int opcion;
            do{
                cout << "nombreCancion actual : " << listaCanciones[i].nombreCancion <<
endl;
                cout << "nombreCantante actual: " << listaCanciones[i].nombreCantante <<
endl;
                cout << "genero actual : " << listaCanciones[i].genero <<
endl;
                cout << "portada actual : " << listaCanciones[i].portada <<
endl;
                cout << "album Archivo actual : " << listaCanciones[i].albumArchivo <<
endl;
                cout << "costo actual : " << listaCanciones[i].precio <<
endl;
            } while (opcion < 0 || opcion > 7);
        }
    }
}

```

```

menuCambiar();
cin>>opcion;
switch(opcion){
    case 1:
        cout << "Dame nueva nombre de la cancion: " ;
        getline(cin, listaCanciones[i].nombreCancion);
        getline(cin, listaCanciones[i].nombreCancion);
        cout << "Cambio efectuado" << endl;
        escribirEnArchivo();
        break;
    case 2:
        cout << "Dame nuevo nombre de cantante: " ;
        getline(cin, listaCanciones[i].nombreCantante);
        getline(cin, listaCanciones[i].nombreCantante);
        cout << "Cambio efectuado" << endl;
        escribirEnArchivo();
        break;
    case 3:
        cout << "Dame nuevo genero: " ;
        getline(cin, listaCanciones[i].genero);
        getline(cin, listaCanciones[i].genero);
        cout << "Cambio efectuado" << endl;
        escribirEnArchivo();
        break;

    case 4:
        cout << "Dame nueva portada: " ;
        getline(cin, listaCanciones[i].portada);
        getline(cin, listaCanciones[i].portada);
        cout << "Cambio efectuado" << endl;
        escribirEnArchivo();
        break;
    case 5:
        cout << "Dame nuevo album Archivo: " ;
        getline(cin, listaCanciones[i].albumArchivo);
        getline(cin, listaCanciones[i].albumArchivo);
        cout << "Cambio efectuado" << endl;
        escribirEnArchivo();
        break;
    case 6:
        cout << "Dame nuevo costo: " ;
        cin >> listaCanciones[i].precio;
        cout << "Cambio efectuado" << endl;
        escribirEnArchivo();
        break;
    case 7:
        break;
    default:
        cout <<"Opcion no valida.";
        break;
} //fin de switvh
}while(opcion != 7);
return;
}
}
cout << "Elemento no encontrado" << endl;
}

```

```

void USUARIO() {
    int resultadosaldo;
    inicializar();
    recuperarDeArchivo();
    inicializarSaldo();
    obtenerSaldo();
    inicializarCompradas();
    recuperarDeArchivoCompradas();
    int abs;
    do{
        cout << setw(20) << " USUARIO " << endl << endl;
        cout << "bienvenido a MusicStore " << endl << endl;

        cout << "1.- Ver catalogo de musica " << endl;
        cout << "2.- Comprar cancion " << endl;
        cout << "3.- Abonar Saldo " << endl;
        cout << "4.- ver mis canciones compradas en la pagina (registro)" << endl;
        cout << "5.- ver registro " << endl;
        cout << "6.- consultar saldo " << endl;
        cout << "7.- Salir " << endl;
        cout << endl;
        cout << "que desea realizar: ";
        cin >> abs;
        switch(abs){
            case 1:
                escribirPaginaWeb();
                system("INDEX.html");//manda llamar al archivo y lo abre en el
                navegador

                break;
            case 2:
                system("cls");
                comprarCancion();
                escribirEnArchivoCompradas();

                break;
            case 3:
                system("cls");
                cout << endl;
                inicializarSaldo();
                obtenerSaldo();
                cout << "tu saldo actual es de: " << usuario.SALDO << endl;
                agregarSaldo();
                escribirEnArchivoMiSaldo();
                cout << endl << endl;
                cout << "tu saldo es actual es: " << usuario.SALDO << endl;
                cout << endl << endl;
                system("pause");
                system("cls");
                break;
            case 4:
                system("pause");
                cout << "abriendo pagina web...." << endl;
                escribirPaginaWebCompradas();
                system("listaCancionessCompradas.html");
                system("cls");
                break;
            case 5:
                system("cls");
                mostrarDatosCompradas();
                cout << endl << endl << endl ;
                system("pause");

```

```

        system("cls");
        break;

    case 6:
        system("cls");
        cout << "Tu saldo actual es de: ";
        cout << usuario.SALDO << endl << endl;
        system("pause");
        system("cls");
        break;

    case 7:
        cout << "saliendo..." << endl;
        system("pause");
        break;

    default:
        cout << "inserta una opcion valida..." << endl;
        break;
    } //fin de switch
} while (abs != 7);
} //fin de USUARIO

void agregarSaldo(){
    int SALDOx;
    cout << endl << endl;
    cout << "cuanto desea abonar a su saldo: ";
    cin >> SALDOx;
    //cout << "tu saldo es: " << credito << endl;
    insertarSaldo(SALDOx);
}

void inicializarSaldo(){
    usuario.SALDO = 0;
}

void insertarSaldo(int SALDOx){
    int c;
    c = SALDOx + usuario.SALDO;
    usuario.SALDO = c;
}

double restarSaldo( double monto){
    double montoInicial, montoFinal;
    cout << endl << endl;
    cout << "monto: " << monto << endl;
    cout << endl;
    //system("pause");
    montoInicial = usuario.SALDO;
    cout << "saldo actual: " << montoInicial << endl;
    system("pause");
    montoFinal = montoInicial - monto;
    usuario.SALDO = montoFinal;
    //escribirEnArchivoMiSaldo();
    cout << "tu saldo actual es: " << usuario.SALDO << endl;
    return usuario.SALDO;
}

void escribirEnArchivoMiSaldo(){
    ofstream salidaArchivo("SALDO.txt", ios::out);
    if (!salidaArchivo) {
        cerr << "No se pudo abrir archivo" << endl;
        exit(1);
    }
}

```

```

    }
    salidaArchivo << usuario.SALDO << ' ' << endl; //se escribe en el archivo el saldo
}

void obtenerSaldo(){//es como resuperar de archivo
    ifstream lecturaArchivo("SALDO.txt", ios::in);
    if (!lecturaArchivo) {
        cerr << "No se pudo abrir el archivo" << endl;
        exit(1);
    }
    int creditox;

    while (lecturaArchivo >> creditox) {
        insertarSaldo(creditox);
    }
}

void comprarCancion(){
    inicializar();
    recuperarDeArchivo();
    int claveCompra;
    int resultadoBusqueda, resultadoBusquedaCompradas;
    bool validarClave = true;
    do{
        cout << endl;
        mostrarDatos();
        validarClave = true;
        cout << "inserta la clave de la cancion que deses comprar: ";
        cin >> claveCompra;
        ordenacionBurbujaPorClave();
        resultadoBusqueda = busquedaBinaria(claveCompra); //se busca la cancion en la
lista del catalogo
        if(resultadoBusqueda == -1){//no se encuentra la cancion en la lista
            validarClave = false;
            cout << endl << "erro..." << endl;
            cout << " esta cancion no se encuentra en la lista." << endl;
            cout << " inserte clave correcta nuevamente." << endl;
        }else{
            resultadoBusquedaCompradas = busquedaSecuencialCompradas(claveCompra);
            if(resultadoBusquedaCompradas == -1){//no se encuentra en canciones compradas
por lo tanto se puede comprar
                validarClave = true;
                inicializar();
                recuperarDeArchivo();
                cout << endl;
                int cl;
                cl = busquedaSecuencial(claveCompra);
                system("pause");
                cout << "Cancion a comprar: " << endl << endl;
                for(int i = 0; i < NUM; i++){
                    if(listaCanciones[i].clave == claveCompra){
                        cout << "nombreCancion completo.: " <<
listaCanciones[cl].nombreCancion << endl;
                        cout << "nombreCantante actual...: " <<
listaCanciones[cl].nombreCantante << endl;
                        cout << "genero Actual.....: " << listaCanciones[cl].genero
<< endl;
                        cout << "portada Actual.....: " << listaCanciones[cl].portada
<< endl;
                        cout << "album Archivo Actual....: " <<
listaCanciones[cl].albumArchivo << endl;
                        cout << "costo Actual.....: " << listaCanciones[cl].precio
<< endl;
                    }
                }
            }
        }
    } while (!validarClave);
}

```



```

        //restarSaldo(listaCanciones[cl].precio);
    }
}

double validarSaldo;
validarSaldo = restarSaldo(listaCanciones[cl].precio);
cout << "saldo....."<<validarSaldo << endl << endl;
if(validarSaldo < 0){
    validarClave = true;
    cout << endl;
    cout << "lo sentimos, saldo insuficiente " << endl;
    insertarSaldo(listaCanciones[cl].precio);
}else{
    insertarCompradas(claveCompra,
                      listaCanciones[cl].nombreCancion,
                      listaCanciones[cl].nombreCantante,
                      listaCanciones[cl].genero,
                      listaCanciones[cl].portada,
                      listaCanciones[cl].albumArchivo,
                      listaCanciones[cl].precio);

    escribirEnArchivoMiSaldo();
}
system("pause");
system("cls");

}else{
    cout << endl << endl;
    cout << " ya has comprado esta cancion, elige otra diferente." << endl;
    validarClave = false;
    system("pause");
    system("cls");
}
}
} //fin de primer condicional else
}while(validarClave != true);
}

void insertarCompradas(int claveCompradasx, string nombreCancionCompradasx, string
nombreCantanteCompradasx,
    string generoCompradasx, string portadaCompradasx, string albumArchivoCompradasx,
double precioCompradasx) {
    int lugarVacioCompradas = buscarLugarVacioCompradas();
    if (lugarVacioCompradas == -1) {
        cout << "Ya no hay lugar" << endl;
        return;
    }
    listaCancionesCompradas[lugarVacioCompradas].claveCompradas = claveCompradasx;
    listaCancionesCompradas[lugarVacioCompradas].nombreCancionCompradas = nombreCancionCompradasx;
    listaCancionesCompradas[lugarVacioCompradas].nombreCantanteCompradas =
nombreCantanteCompradasx;
    listaCancionesCompradas[lugarVacioCompradas].generoCompradas = generoCompradasx;
    listaCancionesCompradas[lugarVacioCompradas].portadaCompradas = portadaCompradasx;
    listaCancionesCompradas[lugarVacioCompradas].albumArchivoCompradas = albumArchivoCompradasx;
    listaCancionesCompradas[lugarVacioCompradas].precioCompradas = precioCompradasx;
}

void mostrarDatosCompradas() {
    cout << endl;
    cout << setw(6) << "CLAVE"
        << setw(20) << "nombreCancion"
        << setw(15) << "nombreCantante"
        << setw(15) << "genero"
        << setw(15) << "portada"
        << setw(15) << "albumArchivo"

```

```

        << setw(7) << "precio" << endl;
for (int i = 0; i < NUM; i++) {
    if (listaCancionesCompradas[i].claveCompradas != 0) {
        cout << setw(6) << listaCancionesCompradas[i].claveCompradas
            << setw(20) << listaCancionesCompradas[i].nombreCancionCompradas
            << setw(15) << listaCancionesCompradas[i].nombreCantanteCompradas
            << setw(15) << listaCancionesCompradas[i].generoCompradas
            << setw(15) << listaCancionesCompradas[i].portadaCompradas
            << setw(15) << listaCancionesCompradas[i].albumArchivoCompradas
            << setw(7) << listaCancionesCompradas[i].precioCompradas
            << endl;
    }
}

}

void recuperarDeArchivoCompradas() {
    ifstream lecturaArchivo("listaCancionesCompradas.txt", ios::in);
    if (!lecturaArchivo) {
        cerr << "No se pudo abrir el archivo" << endl;
        exit(1);
    }
    int claveCompradasx;
    string nombreCancionCompradasx;
    string nombreCantanteCompradasx;
    string generoCompradasx;
    string portadaCompradasx;
    string albumArchivoCompradasx;
    double precioCompradasx;

    while (lecturaArchivo >> claveCompradasx >> nombreCancionCompradasx >>
nombreCantanteCompradasx >> generoCompradasx >> portadaCompradasx >>
albumArchivoCompradasx >> precioCompradasx) {
        insertarCompradas(claveCompradasx, reemplazarSharp(nombreCancionCompradasx),
reemplazarSharp(nombreCantanteCompradasx), reemplazarSharp(generoCompradasx),
reemplazarSharp(portadaCompradasx),
reemplazarSharp(albumArchivoCompradasx), precioCompradasx);
    }
}

void escribirEnArchivoCompradas() {
    ofstream salidaArchivo("listaCancionesCompradas.txt", ios::out);
    if (!salidaArchivo) {
        cerr << "No se pudo abrir archivo" << endl;
        exit(1);
    }
    for (int i = 0; i < NUM; i++) {
        if (listaCancionesCompradas[i].claveCompradas != 0) {
            salidaArchivo << listaCancionesCompradas[i].claveCompradas << ' '
            << reemplazarEspacios(listaCancionesCompradas[i].nombreCancionCompradas) << ' '
            << reemplazarEspacios(listaCancionesCompradas[i].nombreCantanteCompradas) << ' '
            << reemplazarEspacios(listaCancionesCompradas[i].generoCompradas) << ' '
            << reemplazarEspacios(listaCancionesCompradas[i].portadaCompradas) << ' '
            << reemplazarEspacios(listaCancionesCompradas[i].albumArchivoCompradas) << ' '
            << listaCancionesCompradas[i].precioCompradas << ' ' << endl;
        }
    }
}
}

```

```

void inicializarCompradas() {
    for (int i = 0; i < NUM; i++) {
        listaCancionesCompradas[i].claveCompradas = 0;
        listaCancionesCompradas[i].nombreCancionCompradas = "";
        listaCancionesCompradas[i].nombreCantanteCompradas = "";
        listaCancionesCompradas[i].generoCompradas = "";
        listaCancionesCompradas[i].portadaCompradas = "";
        listaCancionesCompradas[i].albumArchivoCompradas = "";
        listaCancionesCompradas[i].precioCompradas = 0;
    }
}

int buscarLugarVacioCompradas() {
    for (int i = 0; i < NUM; i++) {
        if (listaCancionesCompradas[i].claveCompradas == 0) {
            return i;
        }
    }
    return -1;
}

void escribirPaginaWebCompradas() {
    ofstream salidaArchivo("listaCancionesCompradas.html", ios::out);
    if (!salidaArchivo) {
        cerr << "No se pudo abrir el archivo" << endl;
        exit(1);
    }

    salidaArchivo << "<!DOCTYPE html>" << endl;
    salidaArchivo << "<html>" << endl;

    salidaArchivo << "<head>" << endl;
    salidaArchivo << "    <meta charset='utf-8'>" << endl;
    salidaArchivo << "    <title>MusicStore.mx</title>" << endl;
    salidaArchivo << "    <meta name='Gustavo ICO-22' content='Mi tienda virtual de musica'>" << endl;
    salidaArchivo << "    <link rel='stylesheet' href='main.css'>" << endl;
    salidaArchivo << "    <style type='text/css'>" << endl;
    salidaArchivo << "table{border: red dotted medium; margin: auto; vertical-align: sub}" << endl;
    salidaArchivo << "td{border: blue dashed medium; margin: auto}" << endl;
    salidaArchivo << "    article{margin:0% }" << endl;
    salidaArchivo << "    section{margin: 5px;height: auto;width: 1250px}" << endl;
    salidaArchivo << "h1{" << endl;
    salidaArchivo << "border-color: orangered;" << endl;
    salidaArchivo << "border-style: solid;" << endl;
    salidaArchivo << "margin: auto;" << endl;
    salidaArchivo << "height: 95px;" << endl;
    salidaArchivo << "font-size: 90px;" << endl;
    salidaArchivo << "font-family: Segoe Ui Light;" << endl;
    salidaArchivo << "}" << endl;
    salidaArchivo << "</style>" << endl;

    salidaArchivo << "</head>" << endl;
    salidaArchivo << "<body>" << endl;
    salidaArchivo << "    <header><h1>MusicStore</h1></header>" << endl;
    salidaArchivo << "    <nav><marquee WIDTH=60% HEIGHT=60 ALIGN=BOTTOM"
    BEHAVIOR=ALTERNATE><h3>BIENVENIDO Si buscas canciones, estas en el lugar
    CORRECTO</h3></marquee></nav>" << endl;
    salidaArchivo << "    <section>" << endl;
    //aquí empieza la tabla
    salidaArchivo << "    <table border = 1>" << endl;
    salidaArchivo << "    <thead> <tr>" << endl;
    salidaArchivo << "    <th> <h2>Clave</h2> </th><th> <h2>Nombre Cancion </h2></th>" << endl;
    salidaArchivo << "    <th> <h2>Nombre Cantante </h2></th> <th> <h2>genero </h2></th> " << endl;
    salidaArchivo << "    <th> <h2>Portada </h2></th> <th> <h2>AlbumArchivo </h2></th> " << endl;
    salidaArchivo << "    <th> <h2>Costo </h2></th> " << endl;

```

```

salidaArchivo << "</tr></head> " << endl;
salidaArchivo << "<t/body>" << endl;
//cuerpo de la tabla
for(int i=0; i < NUM; i++){
    if(listaCancionesCompradas[i].claveCompradas != 0){
        salidaArchivo << "<tr>"
            <<"<td><h2>"<< listaCancionesCompradas[i].claveCompradas << "</h2></td>"
            <<"<td><h2>"<< listaCancionesCompradas[i].nombreCancionCompradas <<
"</h2></td>"
            <<"<td><h2>"<< listaCancionesCompradas[i].nombreCantanteCompradas <<
"</h2></td>"
            <<"<td><h2>"<< listaCancionesCompradas[i].generoCompradas<< "</h2></td>"
            <<"<td><img src='Portadas/"<< listaCancionesCompradas[i].portadaCompradas<< "'
border='10' width='150' height='150'></td>"
            <<"<td><a href='Musica/"<< listaCancionesCompradas[i].albumArchivoCompradas<<
"' download='"<<
listaCancionesCompradas[i].albumArchivoCompradas<<"'><h2>Download Music</h2></a><audio
src='Musica/"<< listaCancionesCompradas[i].albumArchivoCompradas<< "' controls='controls'
type='audio/mpeg' preload='preload'> </audio></td>"
            <<"<td> <h2>$"<< listaCancionesCompradas[i].precioCompradas << "</h2></td>"
            << "</tr>" << endl;
    }
}
salidaArchivo << "<t/body>" << endl;
salidaArchivo << "</table>" << endl;
//aqui finaliza la tabla

salidaArchivo << "    </section>" << endl;
salidaArchivo << "<footer> <h2>By: Gustavo Blas Duran</h2></footer>" << endl;
salidaArchivo << "</body>" << endl;
salidaArchivo << "</html>" << endl;

}

//fin de escribir pagina web

int busquedaSecuencialCompradas(int b){
    for(int i=0; i<NUM; i++){
        if(listaCancionesCompradas[i].claveCompradas == b){
            return 1;
        }
    }
    return -1;
}

int busquedaBinariaCompradas(int b){
    inicializarCompradas();
    recuperarDeArchivoCompradas();
    int bajo, alto, central;
    bajo = 0;
    alto = NUM - 1;
    while (bajo <= alto){
        central = (bajo + alto) / 2;
        if(b == listaCancionesCompradas[central].claveCompradas){
            return central;
        }else if(b < listaCancionesCompradas[central].claveCompradas){
            alto = central - 1;
        }else {
            bajo = central + 1;
        }
    }
    return -1;
}

```

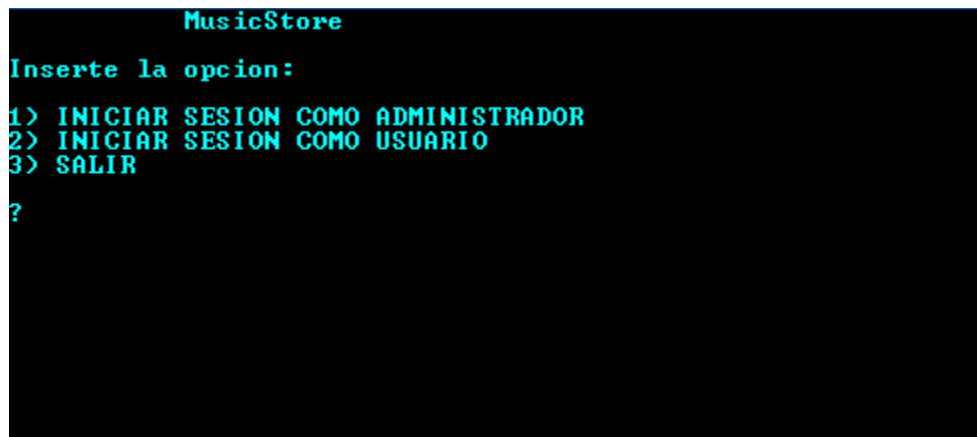
```

void mostrarRegistroCompradas(int i){
    cout << "nombreCancion completo: " <<
listaCancionesCompradas[i].nombreCancionCompradas << endl;
    cout << "nombreCantante actual:" <<
listaCancionesCompradas[i].nombreCantanteCompradas << endl;
    cout << "genero Actual: "<< listaCancionesCompradas[i].generoCompradas << endl;
    cout << "portada Actual: "<< listaCancionesCompradas[i].portadaCompradas << endl;
    cout << "album Archivo Actual: "<< listaCancionesCompradas[i].albumArchivoCompradas
<< endl;
    cout << "costo Actual: "<< listaCancionesCompradas[i].precioCompradas << endl;
} //fin de registro
ListaCancionesCompradas auxC;
void ordenacionBurbujaPorClaveCompradas(){
    for(int i=0; i < NUM-1; i++){
        for(int j = 0; j < NUM - 1; j++){
            if(listaCancionesCompradas[j].claveCompradas > listaCancionesCompradas[j +
1].claveCompradas){
                //intercambio
                auxC = listaCancionesCompradas[j];
                listaCancionesCompradas[j] = listaCancionesCompradas[j + 1];
                listaCancionesCompradas[j + 1] = auxC;
            }
        }
    }
}

```

Capturas de pantalla

Pantalla principal se muestra el menú de como quiere acceder el usuario



Se piden usuario y contraseña de administrador

```
ADMINISTRADOR
USERNAME:  GUSTAUOBD
PASSWORD:  gblasd239
```

Se muestra menú principal del administrador

```
ADMINISTRADOR
E L I G E   T U   O P C I O N
1.-ALTA DE CANCIONES
2.-REPORTE GENERAL
3.-VER PAGINA WEB
4.-ORDENAR POR CLAVE DE CANCION
5.-ORDENAR POR NOMBRE CANCION
6.-BUSCAR POR CLAVE DEL CANCION <SECUENCIAL>
7.-BUSCAR POR CLAVE DEL CANCION <BINARIO>
8.-BORRAR UN ELEMENTO
9.-EDITAR UN ELEMENTO
10.-DAR DE BAJA UNA CANCION
11.-SALIR
?
```

Se inserta una clave de la canción, la cual ya existe, por lo tanto muestra un mensaje de error y no acepta la clave, la vuelve a pedir:

```
Escribe datos del las Canciones.  
Inserta clave:19  
ERROR :(<  clave repetida ...  
vuelva a insertar otra clave diferente  
Inserta clave:
```

Se insertan los datos de las canciones


```
Escribe datos del las Canciones.  
Inserta clave:19  
ERROR :(<  clave repetida ...  
vuelva a insertar otra clave diferente  
Inserta clave:20  
Inserta nombre la Cancion:Iamour Toujours  
Inserta nombre del Cantante:Tiesto  
Inserta genero:Dance  
Inserta portada: 20.png  
Inserta album Archivo: Iamour Toujours.mp3  
Inserta costo de la cancion:20
```

Opción dos del menú: se muestran los datos de las canciones que se han dado de alta

CLAVE	nombreCancion	nombreCantante	genero	portada	albumArchivo	precio
1	Red Lights	Tiesto	Dance	1.png	Red Lights.mp3	12
2	Que Bonito Es Querer	Ulices Chaidez	Banda	2.png	Que Bonito Es Querer.mp3	20
3	Ode To Oi	TJR	Dance	3.png	Ode To Oi.mp3	13
4	Great Spirit	Armin Van Buuren	Dance	4.png	Great Spirit.mp3	16
5	The Hills	The Weeknd	Trap	5.png	The Hills.mp3	11
6	El Color De Tus Ojos	Banda MS	Banda	6.png	El Color De Tus Ojos.mp3	12
7	The Nights	Avicii	Dance	7.png	The Nights.mp3	19
8	Uas A Regresar Solita	Calibre 50	Banda	8.png	Uas A Regresar Solita.mp3	14
9	Arrullo de Estrellas	Zoe	Alternativa	9.png	Arrullo de Estrellas.mp3	16
10	Crowd Control	DU&LM	Dance	10.png	Crowd Control.mp3	20
11	Spaceman	Hardwell	Electro	11.png	Spaceman.mp3	17
12	It Will Rain	Bruno Mars	Pop	12.png	It Will Rain.mp3	19
13	Wakanda	DU&LM	Dance	13.png	Wakanda.mp3	18
14	Ruby	Kaiser Chiefs	Alternativa	14.png	Ruby.mp3	12
15	tu Eres Mi Sueño	Septeto Acarey	Salsa	15.png	Tu Eres Mi Sueño.mp3	10
16	See You Again	Wiz Khalifa	Hip-Hop	16.png	See You Again.mp3	13
17	More Than You Know	Axwell & Ingrosso	Dance	17.png	More Than You Know.mp3	16
18	No Te Pido Mucho	Alta Consigna	Regional	18.png	No Te Pido Mucho.mp3	20
19	Thinking Out Loud	Ed Sheeran	Pop	19.png	Thinking Out Loud.mp3	16
20	Iamour Toujours	Tiesto	Dance	20.png	Iamour Toujours.mp3	20





Presione una tecla para continuar . . .

Opción tres del menú del administrador: se muestra la página web con las canciones, solo se reproducen 30 segundos por canción.



MusicStore

BIENVENIDO Si buscas musica, estas en el lugar CORRECTO

Clave	Nombre Cancion	Nombre Cantante	genero	Portada	AlbumArchivo	Costo
1	Red Lights	Tiesto	Dance		▶ ● 0:00 / 0:30 ◀ —●	\$ 12
2	Que Bonito Es Querer	Ulices Chaidez	Banda		▶ ● 0:00 / 0:30 ◀ —●	\$ 20
3	Ode To Oi	TJR	Dance		▶ ● 0:00 / 0:30 ◀ —●	\$ 13
4	Great Spirit	Armin Van Buuren	Dance		▶ ● 0:00 / 0:31 ◀ —●	\$ 16

Se muestra la última canción que se dio de alta



Opción 4 del menu de administrador: se muestran las canciones ordenadas por clave

CLAVE	nombreCancion	nombreCantante	genero	portada	album	Archivo	precio
1	Red Lights	Tiesto	Dance	1.png	Red Lights.mp3		12
2	Que Bonito Es Querer	Ulices Chaidez	Banda	2.png	Que Bonito Es Querer.mp3		
3	Ode To Oi	TJR	Dance	3.png	Ode To Oi.mp3		13
4	Great Spirit	Armin Van Buuren	Dance	4.png	Great Spirit.mp3		16
5	The Hills	The Weeknd	Trap	5.png	The Hills.mp3		11
6	El Color De Tus Ojos	Banda MS	Banda	6.png	El Color De Tus Ojos.mp3		
7	The Nights	Avicii	Dance	7.png	The Nights.mp3		19
8	Uas A Regresar Solita	Calibre 50	Banda	8.png	Uas A Regresar Solita.mp3		
9	Arrullo de Estrellas	Zoe	Alternativa	9.png	Arrullo de Estrellas.mp3		
10	Crowd Control	DU&LM	Dance	10.png	Crowd Control.mp3		20
11	Spaceman	Hardwell	Electro	11.png	Spaceman.mp3		17
12	It Will Rain	Bruno Mars	Pop	12.png	It Will Rain.mp3		19
13	Wakanda	DU&LM	Dance	13.png	Wakanda.mp3		18
14	Ruby	Kaiser Chiefs	Alternativa	14.png	Ruby.mp3		12
15	tu Eres Mi Sueño	Septeto Acarey	Salsa	15.png	Tu Eres Mi Sueño.mp3		10
16	See You Again	Wiz Khalifa	Hip-Hop	16.png	See You Again.mp3		13
17	More Than You Know	Axwell & Ingroso	Dance	17.png	More Than You Know.mp3		1
18	No Te Pido Mucho	Alta Consigna	Regional	18.png	No Te Pido Mucho.mp3		20
19	Thinking Out Loud	Ed Sheeran	Pop	19.png	Thinking Out Loud.mp3		16
20	Iamour Toujours	Tiesto	Dance	20.png	Iamour Toujours.mp3		20

Presione una tecla para continuar . . .

Opción 5 Administrador: se muestran las canciones ordenadas por el nombre de la canción

CLAVE	nombreCancion	nombreCantante	genero	portada	album	Archivo	precio
9	Arrullo de Estrellas	Zoe	Alternativa	9.png	Arrullo de Estrellas.mp3		16
10	Crowd Control	DU&LM	Dance	10.png	Crowd Control.mp3		20
6	El Color De Tus Ojos	Banda MS	Banda	6.png	El Color De Tus Ojos.mp3		12
4	Great Spirit	Armin Van Buuren	Dance	4.png	Great Spirit.mp3		16
20	Iamour Toujours	Tiesto	Dance	20.png	Iamour Toujours.mp3		20
12	It Will Rain	Bruno Mars	Pop	12.png	It Will Rain.mp3		19
17	More Than You Know	Axwell & Ingroso	Dance	17.png	More Than You Know.mp3		16
18	No Te Pido Mucho	Alta Consigna	Regional	18.png	No Te Pido Mucho.mp3		20
3	Ode To Oi	TJR	Dance	3.png	Ode To Oi.mp3		13
2	Que Bonito Es Querer	Ulices Chaidez	Banda	2.png	Que Bonito Es Querer.mp3		20
1	Red Lights	Tiesto	Dance	1.png	Red Lights.mp3		12
14	Ruby	Kaiser Chiefs	Alternativa	14.png	Ruby.mp3		12
16	See You Again	Wiz Khalifa	Hip-Hop	16.png	See You Again.mp3		13
11	Spaceman	Hardwell	Electro	11.png	Spaceman.mp3		17
5	The Hills	The Weeknd	Trap	5.png	The Hills.mp3		11
7	The Nights	Avicii	Dance	7.png	The Nights.mp3		19
19	Thinking Out Loud	Ed Sheeran	Pop	19.png	Thinking Out Loud.mp3		16
8	Uas A Regresar Solita	Calibre 50	Banda	8.png	Uas A Regresar Solita.mp3		14
13	Wakanda	DU&LM	Dance	13.png	Wakanda.mp3		18
15	tu Eres Mi Sueño	Septeto Acarey	Salsa	15.png	Tu Eres Mi Sueño.mp3		10

Presione una tecla para continuar . . .

Opción 6 administrador: búsqueda secuencial por clave

```

Dame clave de la Cancion a buscar:
10
Nombre de la cancion...: Crowd Control
Nombre del cantante....: DU&LM
Genero Actual.....: Dance
Portada Actual.....: 10.png
album Archivo Actual...: Crowd Control.mp3
Costo Actual.....: 20
Presione una tecla para continuar . . .

```

Opción 7 administrador: búsqueda binaria por clave

```

dame clave de la Cancion a buscar
4
Nombre de la cancion...: Great Spirit
Nombre del cantante....: Armin Van Buuren
Genero Actual.....: Dance
Portada Actual.....: 4.png
album Archivo Actual...: Great Spirit.mp3
Costo Actual.....: 16
Presione una tecla para continuar . . .

```

Opción 8 administrador: borrar un elemento

CLAVE	nombreCancion	nombreCantante	genero	portada	albumArchivo	precio
1	Red Lights	Tiesto	Dance	1.png	Red Lights.mp3	12
2	Que Bonito Es Querer	Ulices Chaidez	Banda	2.png	Que Bonito Es Querer.mp3	20
3	Ode To Oi	TJR	Dance	3.png	Ode To Oi.mp3	13
4	Great Spirit	Armin Van Buuren	Dance	4.png	Great Spirit.mp3	16
5	The Hills	The Weeknd	Trap	5.png	The Hills.mp3	11
6	El Color De Tus Ojos	Banda MS	Banda	6.png	El Color De Tus Ojos.mp3	12
7	The Nights	Avicii	Dance	7.png	The Nights.mp3	19
8	Uas A Regresar Solita	Calibre 50	Banda	8.png	Uas A Regresar Solita.mp3	14
9	Arrullo de Estrellas	Zoe	Alternativa	9.png	Arrullo de Estrellas.mp3	16
10	Crowd Control	DU&LM	Dance	10.png	Crowd Control.mp3	20
11	Spaceman	Hardwell	Electro	11.png	Spaceman.mp3	17
12	It Will Rain	Bruno Mars	Pop	12.png	It Will Rain.mp3	19
13	Wakanda	DU&LM	Dance	13.png	Wakanda.mp3	18
14	Ruby	Kaiser Chiefs	Alternativa	14.png	Ruby.mp3	12
15	tu Eres Mi Sueño	Septeto Acarey	Salsa	15.png	Tu Eres Mi Sueño.mp3	10
16	See You Again	Wiz Khalifa	Hip-Hop	16.png	See You Again.mp3	13
17	More Than You Know	Axwell & Ingrosso	Dance	17.png	More Than You Know.mp3	16
18	No Te Pido Mucho	Alta Consigna	Regional	18.png	No Te Pido Mucho.mp3	20
19	Thinking Out Loud	Ed Sheeran	Pop	19.png	Thinking Out Loud.mp3	16
20	Iamour Toujours	Tiesto	Dance	20.png	Iamour Toujours.mp3	20

```

inserta la clave de la cancion para dar de baja
20
Desea eliminar esta cancion ? 1.- si , 2.- no
1.- si :(<
2.- no :>
1
Elemento eliminado

```

Se muestra la lista sin el elemento que se dio de baja

CLAVE	nombreCancion	nombreCantante	genero	portada	albumArchivo	precio
1	Red Lights	Tiesto	Dance	1.png	Red Lights.mp3	12
2	Que Bonito Es Querer	Ulices Chaidez	Banda	2.png	Que Bonito Es Querer.mp3	20
3	Ode To Oi	TJR	Dance	3.png	Ode To Oi.mp3	13
4	Great Spirit	Armin Van Buuren	Dance	4.png	Great Spirit.mp3	16
5	The Hills	The Weeknd	Trap	5.png	The Hills.mp3	11
6	El Color De Tus Ojos	Banda MS	Banda	6.png	El Color De Tus Ojos.mp3	12
7	The Nights	Avicii	Dance	7.png	The Nights.mp3	19
8	Uas A Regresar Solita	Calibre 50	Banda	8.png	Uas A Regresar Solita.mp3	14
9	Arrullo de Estrellas	Zoe	Alternativa	9.png	Arrullo de Estrellas.mp3	16
10	Crowd Control	DU&LM	Dance	10.png	Crowd Control.mp3	20
11	Spaceman	Hardwell	Electro	11.png	Spaceman.mp3	17
12	It Will Rain	Bruno Mars	Pop	12.png	It Will Rain.mp3	19
13	Wakanda	DU&LM	Dance	13.png	Wakanda.mp3	18
14	Ruby	Kaiser Chiefs	Alternativa	14.png	Ruby.mp3	12
15	tu Eres Mi Sueño	Septeto Acarey	Salsa	15.png	Tu Eres Mi Sueno.mp3	10
16	See You Again	Wiz Khalifa	Hip-Hop	16.png	See You Again.mp3	13
17	More Than You Know	Axwell & Ingrosso	Dance	17.png	More Than You Know.mp3	16
18	No Te Pido Mucho	Alta Consigna	Regional	18.png	No Te Pido Mucho.mp3	20
19	Thinking Out Loud	Ed Sheeran	Pop	19.png	Thinking Out Loud.mp3	16
Presione una tecla para continuar . . .						

Opción 9: editar un elemento

Dame la clave del elemento a modificar: 8						
nombreCancion actual : Uas A Regresar Solita						
nombreCantante actual: Calibre 50						
genero actual : Banda						
portada actual : 8.png						
album Archivo actual : Uas A Regresar Solita.mp3						
costo actual : 14						
¿que desea cambiar?						
1.- NombreCancion						
2.- NombreCantante						
3.- Genero						
4.- Portada						
5.- Album Archivo						
6.- Costo de cancion						
7.- Cancelar						
Elija su opcion:						
2						
Dame nuevo nombre de cantante: Calibre50						
Cambio efectuado						
nombreCancion actual : Uas A Regresar Solita						
nombreCantante actual: Calibre50						
genero actual : Banda						
portada actual : 8.png						
album Archivo actual : Uas A Regresar Solita.mp3						
costo actual : 14						
¿que desea cambiar?						
1.- NombreCancion						
2.- NombreCantante						
3.- Genero						
4.- Portada						
5.- Album Archivo						
6.- Costo de cancion						
7.- Cancelar						
Elija su opcion:						

Se inicia sesión como usuario. Se insertan las claves correspondientes:

USUARIO	
USERNAME:	usuario
PASSWORD:	msd9bn45

Menú principal del usuario

```
USUARIO
bienvenido a MusicStore
1.- Ver catalogo de musica
2.- Comprar cancion
3.- Abonar Saldo
4.- ver mis canciones compradas en la pagina <registro>
5.- ver registro
6.- consultar saldo
7.- Salir
que desea realizar:
```

Opción 1: se muestra el catalogo de las canciones en la pagina web.

MusicStore						
BIENVENIDO Si buscas musica, estas en el lugar CORRECTO						
Clave	Nombre Cancion	Nombre Cantante	genero	Portada	AlbumArchivo	Costo
1	Red Lights	Tiesto	Dance			\$ 12
2	Que Bonito Es Querer	Ulices Chaidez	Banda			\$ 20
3	Ode To Oi	TJR	Dance			\$ 13
4	Great Spirit	Armin Van Buuren	Dance			\$ 16
5	The Hills	The Weeknd	Trap			\$ 11
6	El Color De Tus Ojos	Banda MS	Banda			\$ 12

Opción 2 usuario: comprar una canción mediante la clave

```
CLAVE      nombreCancion nombreCantante      genero      portada      albumArchivo      precio
1          Red Lights      Tiesto      Dance      1.png      Red Lights.mp3      12
2Que Bonito Es Querer Ulices Chaidez      Banda      2.png      Que Bonito Es Querer.mp3      20
3          Ode To Oi      TJR      Dance      3.png      Ode To Oi.mp3      13
4          Great SpiritArmin Van Buuren      Dance      4.png      Great Spirit.mp3      16
5          The Hills      The Weeknd      Trap      5.png      The Hills.mp3      11
6El Color De Tus Ojos      Banda MS      Banda      6.png      El Color De Tus Ojos.mp3      12
7          The Nights      Avicii      Dance      7.png      The Nights.mp3      19
8Uas A Regresar Solita      Calibre50      Banda      8.png      Uas A Regresar Solita.mp3      14
9Arrullo de Estrellas      Zoe      Alternativa      9.png      Arrullo de Estrellas.mp3      16
10         Crowd Control      DU&LM      Dance      10.png      Crowd Control.mp3      20
11         Spaceman      Hardwell      Electro      11.png      Spaceman.mp3      17
12         It Will Rain      Bruno Mars      Pop      12.png      It Will Rain.mp3      19
13         Wakanda      DU&LM      Dance      13.png      Wakanda.mp3      18
14         Ruby      Kaiser Chiefs      Alternativa      14.png      Ruby.mp3      12
15         tu Eres Mi Sueño Septeto Arcarey      Salza      15.png      tu Eres Mi Sueño.mp3      10
16         See You Again      Wiz Khalifa      Hip-Hop      16.png      See You Again.mp3      13
17         More Than You KnowAxwell & Ingrosso      Dance      17.png      More Than You Know.mp3      16
18         No Te Pido Mucho      Alta Consigna      Regional      18.png      No Te Pido Mucho.mp3      20
19         Thinking Out Loud      Ed Sheeran      Pop      19.png      Thinking Out Loud.mp3      16

inserta la clave de la cancion que deses comprar: 18

Presione una tecla para continuar . . .
Cancion a comprar:

nombreCancion completo.: No Te Pido Mucho
nombreCantante actual.: Alta Consigna
genero Actual.: Regional
portada Actual.: 18.png
album Archivo Actual.: No Te Pido Mucho.mp3
costo Actual.: 20

monto: 20

saldo actual: 124
Presione una tecla para continuar . . .
tu saldo actual es: 104
saldo.....104

Presione una tecla para continuar . . .
```

Se muestran los datos de la canción a comprar, se resta a el saldo actual el costo de la canción

Opción 3 menú usuario; se abona saldo

```
tu saldo actual es de: 104

cuanto desea abonar a su saldo: 30

tu saldo es actual es: 134

Presione una tecla para continuar . . .
```


Opción 4 se abre la página web con las canciones que se han comprado. Se muestran las canciones completas así como un link para descargarlas.

BIENVENIDO Si buscas musica, estas en el lugar CORRECTO

Clave	Nombre Cancion	Nombre Cantante	genero	Portada	AlbumArchivo	Costo
1	Red Lights	Tiesto	Dance		Download Music 	\$12
2	Que Bonito Es Querer	Ulices Chaidez	Banda		Download Music 	\$20
15	tu Eres Mi Sueño	Septeto Acarey	Salsa		Download Music 	\$10

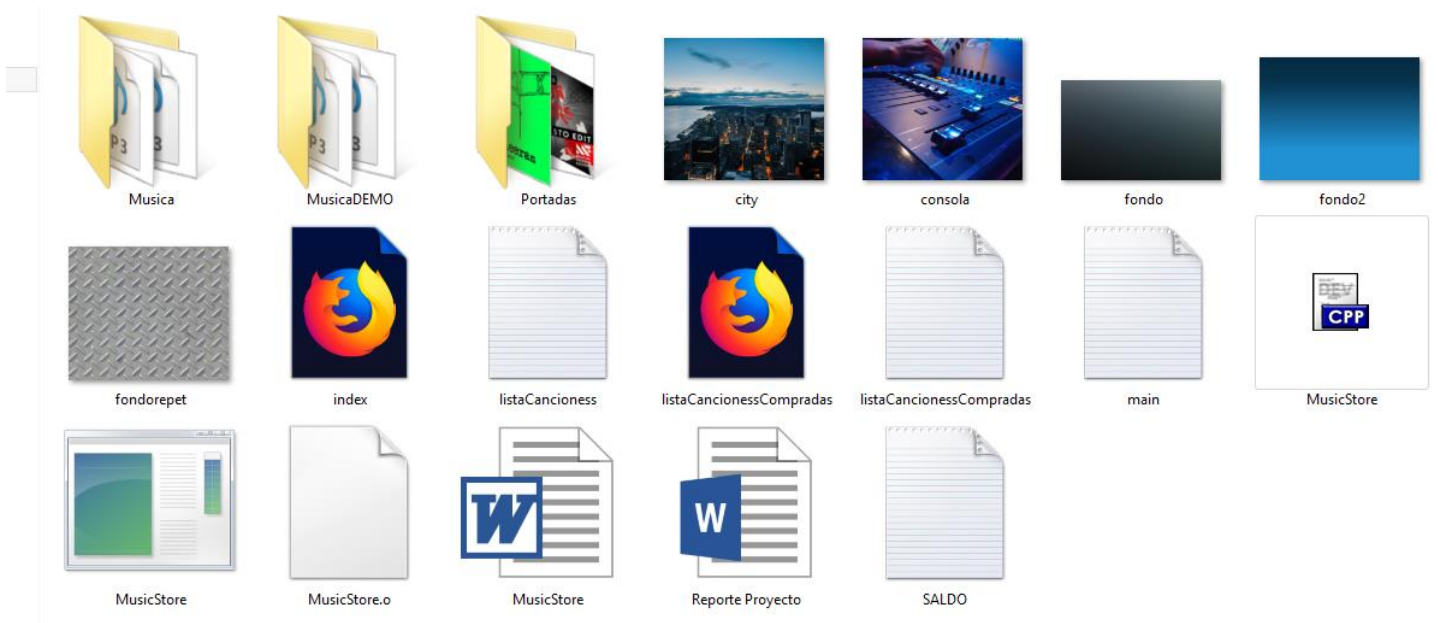
Opción 5 usuario: se muestra el registro de las canciones compradas

```
CLAVE      nombreCancion  nombreCantante      genero      portada  albumArchivo  precio
1          Red Lights  Tiesto              Dance       1.png    Red Lights.mp3  12
2Que Bonito Es Querer  Ulices Chaidez      Banda       2.png    Que Bonito Es Querer.mp3  20
15         tu Eres Mi Sueño  Septeto Acarey      Salsa       15.png   tu Eres Mi Sueño.mp3  10
9Arrullo de Estrellas  Zoe                Alternativa  9.png    Arrullo de Estrellas.mp3  16
18         No Te Pido Mucho  Alta Consigna       Regional    18.png   No Te Pido Mucho.mp3  20

Presione una tecla para continuar . . .
```

Opción 6 se muestran el saldo actual:

```
Tu saldo actual es de:      134
Presione una tecla para continuar . . .
```



Carpeta de archivos

- Cpp
- Índex (catálogo de música, pagina web)
- Archivo de texto (lista de canciones)
- Css (hoja de estilos para el HTML)
- Archivo de texto (saldo9)
- Carpetas (portadas de canciones, canciones)
- Archivo de texto (lista de canciones compradas)
- Archivo HTML (pagina con las canciones compradas)
- Imágenes que se usaron en la página web

```

USER "usuario"/*nombre de usuario*/
PASS1 "msd9bn45"/*contraseña de usuario*/

ADMIN "GUSTAVOBD"/*nombre de administrador*/
PASS2 "gblasd239"/*contraseña de administrador*/

```