Imagen que contiene objeto

Descripción generada con confianza muy alta

Centro universitario UAEM Atlacomulco

Programación avanzada

Proyecto segundo parcial

Nombre: Gustavo Blas Duran

Grupo: ICO-22

Docente: Ing. José Luis García Morales

Mayo, 2018



/\* Tienda de Musica \*/

#include <iostream>

#include <string.h>

#include <stdlib.h>

#include <iomanip>

#include <fstream>

#define USER "usuario"/\*nombre de usuario\*/

#define PASS1 "msd9bn45"/\*contraseÃ±a de usuario\*/

#define ADMIN "GUSTAVOBD"/\*nombre de administrador\*/

#define PASS2 "gblasd239"/\*contraseÃ±a 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**;

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**;

**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 dirente " << **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("listaCancioness.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("listaCancioness.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**;

//aqui 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**(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**;

//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

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** << "Â¿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**;

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 cmo 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 encuntra la cancion en la lista

validarClave = **false**;

**cout** << **endl** << "erro..." << **endl**;

**cout** << " esta canion 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**;

//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 insufieciente " << **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("listaCancionessCompradas.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("listaCancionessCompradas.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("listaCancionessCompradas.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**;

//aqui 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;

}

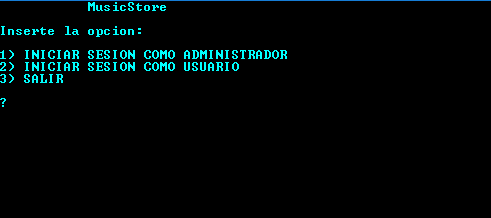
}

}

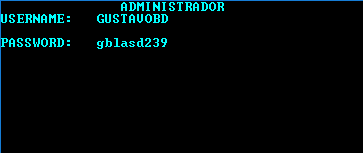
}

# Capturas de pantalla

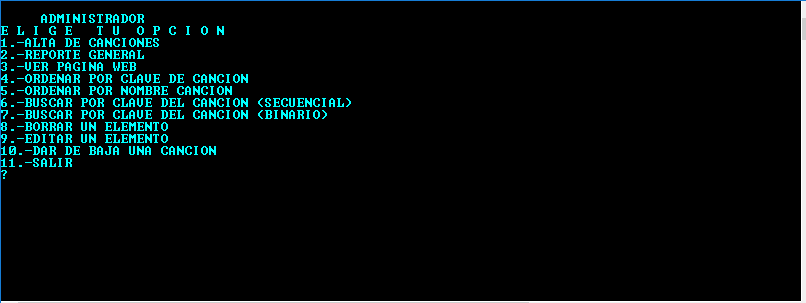
Pantalla principal se muestra el menú de como quiere acceder el usuario



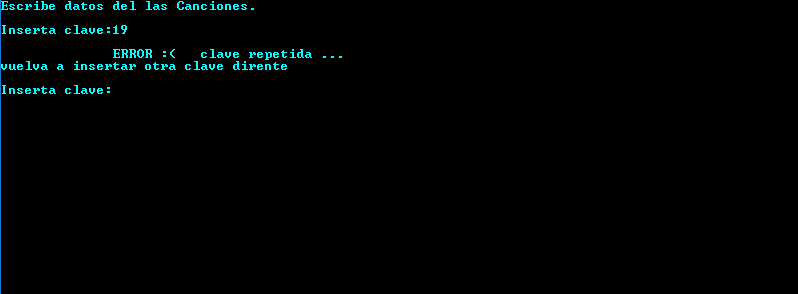
Se piden usuario y contraseña de administrador



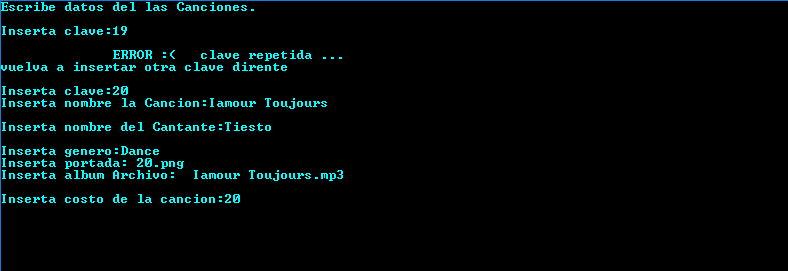
Se muestra menú principal del administrador



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:



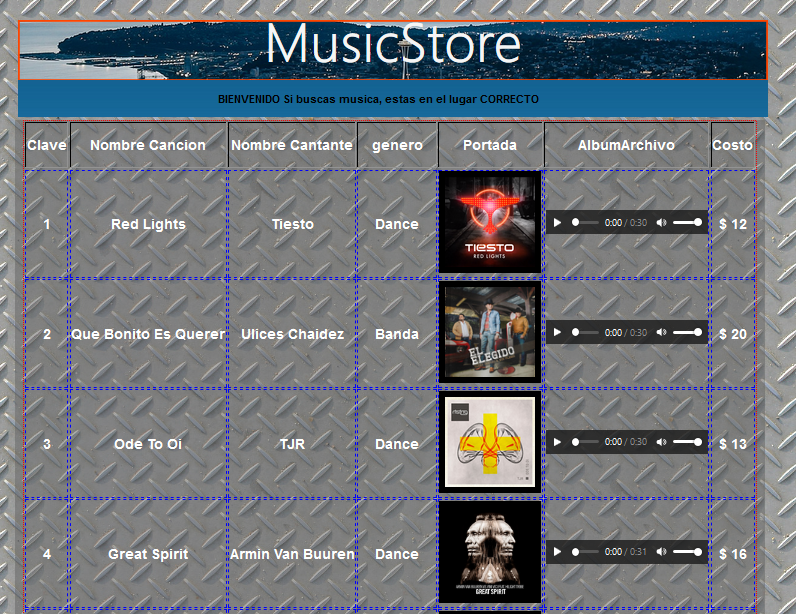
Se insertan los datos de las canciones



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



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



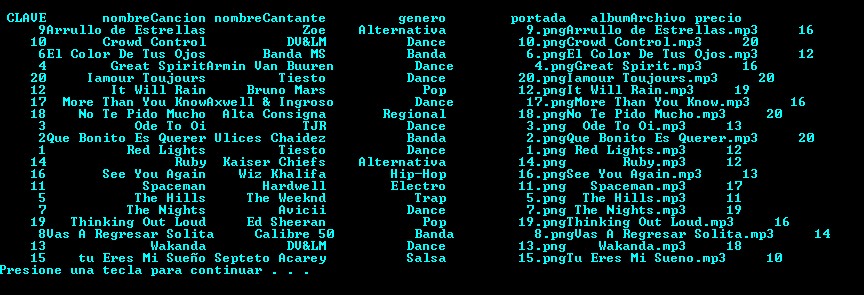
Se muestra la última canción que se dio de alta



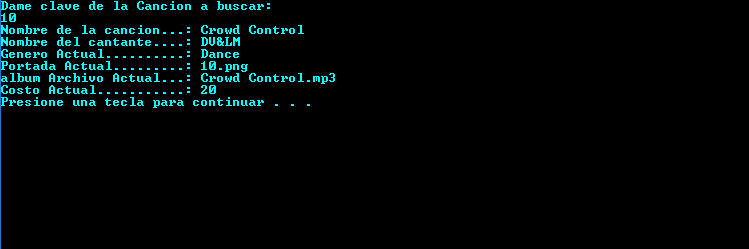
Opción 4 del menu de administrador: se muestran las canciones ordenadas por clave



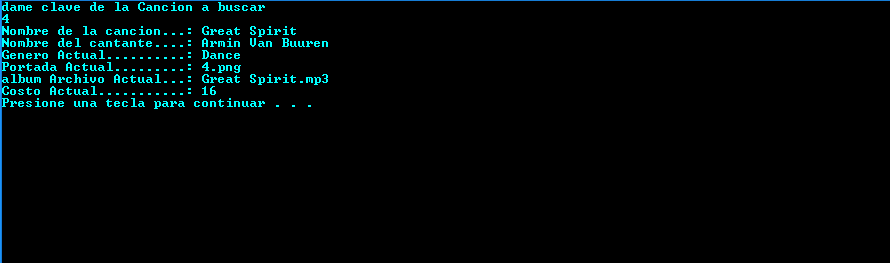
Opción 5 Aministrador: se muestran las caniones ordenadas por el nombre de la canción



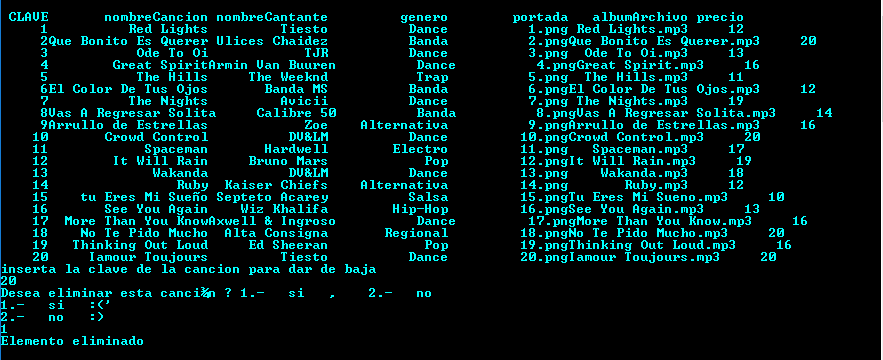
Opción 6 administrador: búsqueda secuencial por clave



Opción 7 administrador: búsqueda binaria por clave



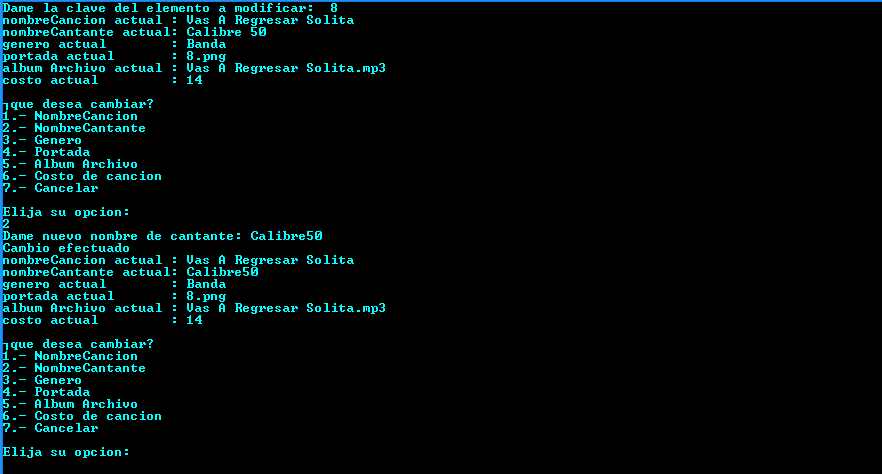
Opción 8 administrador: borrar un elemento



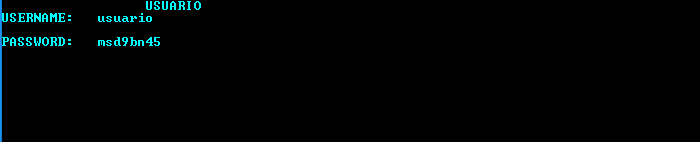
Se muestra la lista sin el elemento que se dio de baja



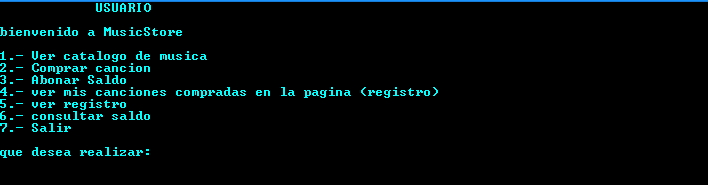
Opción 9: editar un elemento

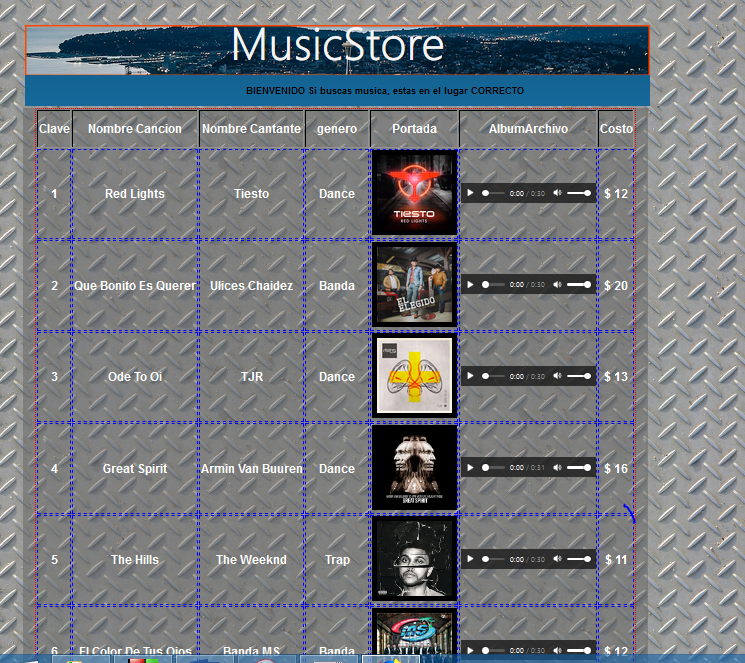


Se inicia sesión como usuario. Se insertan las claves correspondientes:

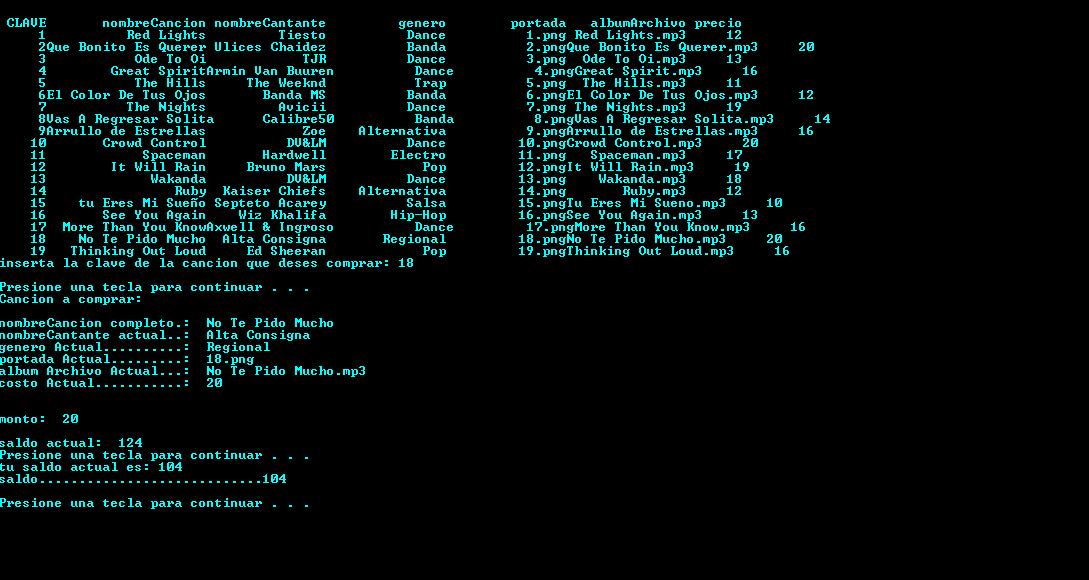


Menú principal del usuario



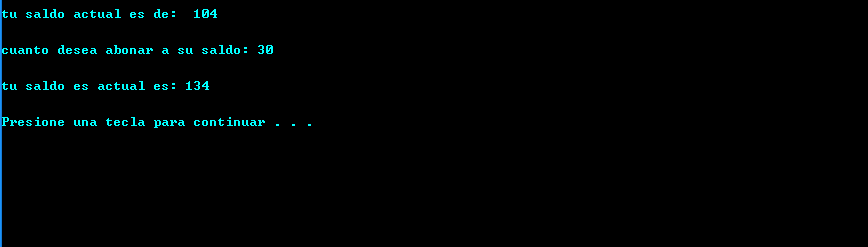
 Opción 1: se muestra el catalogo de las canciones en la pagina web.

Opción 2 usuario: comprar una canción mediante la clave

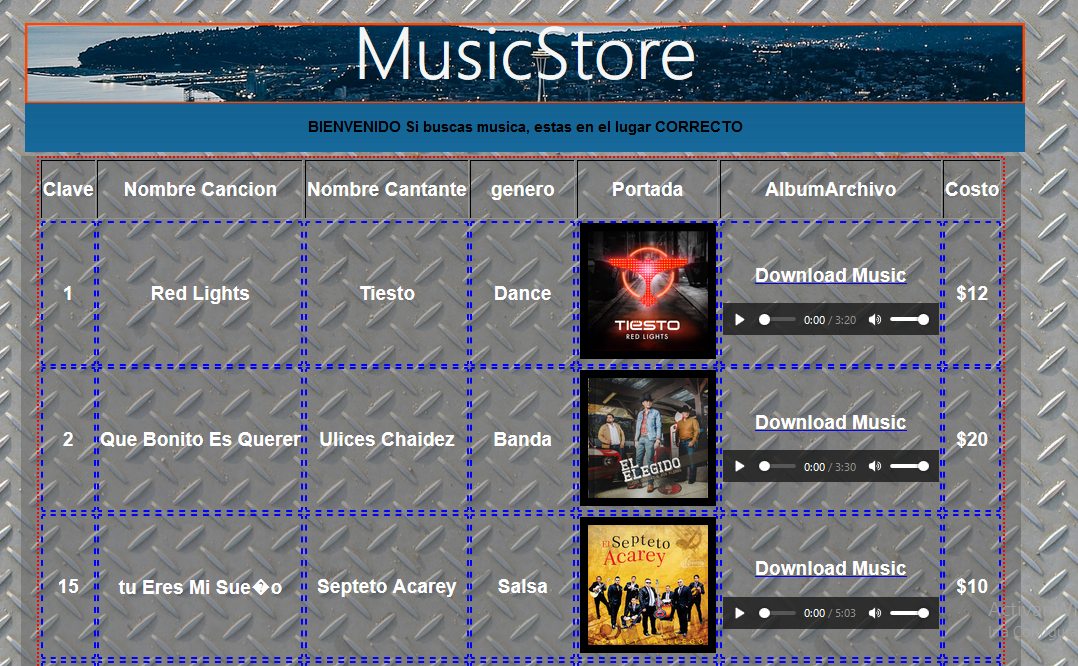


Se muestran los datos de la canción a comprar, se resta a el saldo actual el costo de la canción

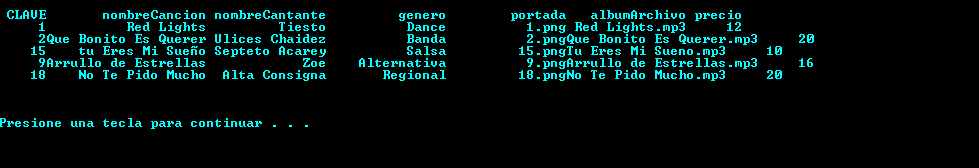
Opción 3 menú usuario; se abona saldo



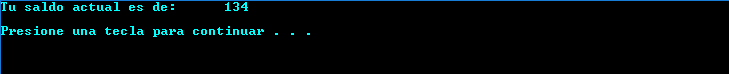
Opción 4 se abre la página web con las canciones que se han comprado. *Se muestran las canciones completas así como un link para descargarlas.*

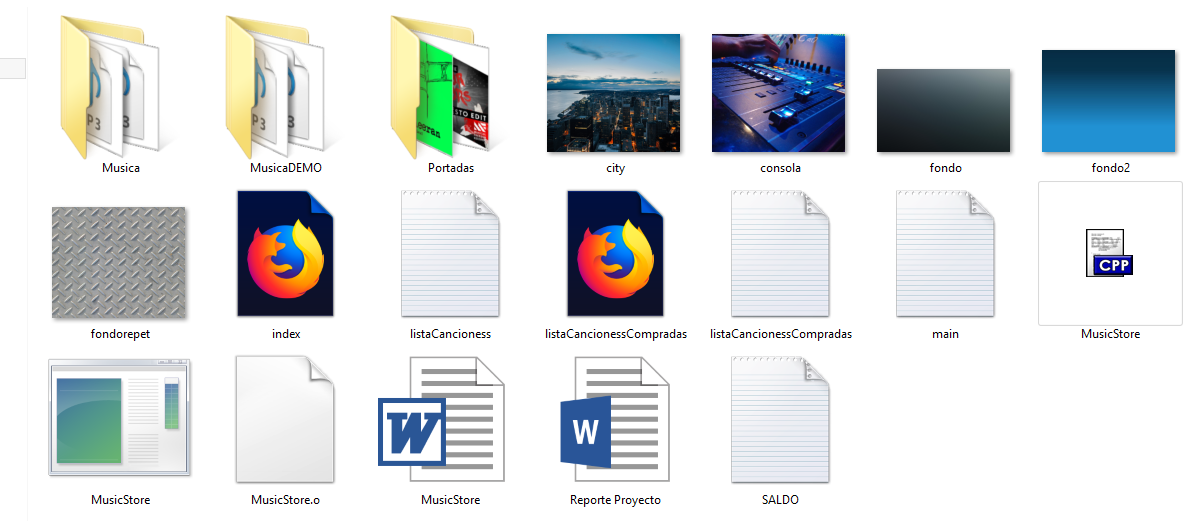


Opción 5 usuario: se muestra el registro de las canciones compradas



Opción 6 se muestran el saldo actual:





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"/\*contraseÃ±a de usuario\*/

ADMIN "GUSTAVOBD"/\*nombre de administrador\*/

PASS2 "gblasd239"/\*contraseÃ±a de administrador\*/