

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE SAN FELIPE DEL PROGRESO

Profesor:

Ing. José Luis García Morales

Materia:

Programación orientada a objetos

Alumno:

Leonardo Yael Morales Caballero

Grupo:

201

Semestre:

2018-A

Corte a evaluar:

Proyecto Tercer Parcial

Tienda virtual de música

Apartado A. Consideraciones técnicas

1. Definir la clase Canción con los siguientes atributos:

Clave NombreCancion NombreCantante Genero
 Portada Album Archivo Precio

2. Definir una clase llamada ColeccionCanciones

| |
|---|
| ColeccionCanciones |
| -listaCanciones:Cancion Array |
| <<constructor>>Album() +meterCancion(cancion:Cancion) +obtenerListaCanciones():Array +eliminarCancion(clave:int) +editar Cancion(clave:int) +buscarCancion(clave:int):Cancion -ordenarCanciones() -busquedaBinariaCancion(clave:int):int |

Al insertar nuevas canciones no se deben repetir las claves.

Las canciones se deben obtener de un archivo de texto

3. Definir una clase que se llame Saldo

| |
|---|
| Saldo |
| -saldoTotal:double |
| +agregarSaldo(credito:double) +restarSaldo(gasto:double) +obtenerSaldo():double |

El saldo se debe obtener de un archivo de texto

4. Las canciones compradas se anexarán a un nuevo arreglo (se guardan las claves de las canciones) junto con el total.

5 Poner un mínimo de 20 canciones con precios entre 10 y 20 pesos, tener lista una canción.

6 Hacer el diagrama completo UML utilizando una herramienta de software.

Apartado B. Consideraciones de uso

1 . Se tendrán 2 opciones para entrar al programa: como Administrador o como usuario. Pedirá una clave para entrar a cada uno. **Como Administrador:**

Podrá hacer los siguientes movimientos

2 . A Canciones: altas, buscar, reporte, generar página Web con caratulas de las canciones.

Como Usuario:

3 . Ver catálogo como página Web (poner las caratulas de las canciones, poner música On Line)

4 . Comprar canción, mediante su clave (se le descuenta a su saldo)

5 . Abonar Saldo (modifica su saldo)

6. Opción para ver la página web con todas las canciones (sus datos y carátula). Se debe tener la opción de reproducir la canción.
7. Opción para ver la página web con solamente las canciones que se han comprado. (10%)
- 8 Hacer un formulario con un menú para dar altas de canciones, bajas de canciones, cambios en el género de la canción, mostrar elementos del álbum y búsquedas de canción. (25%)

Se hará una presentación del proyecto. Se entrega documentación.

Archivo.java

```
package proyectotercerparcial;
/**
 *
 * @author YAEL
 */
import java.util.*;
import java.io.*;
public class Archivo {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<Cancion> listaCanciones = new ArrayList<>();
    public Archivo(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
    }
    private void escribir() {
        //ESCRIBIR CÓDIGO
        //todos los elementos que sean de tipo cadena(String) se le debe poner
        //"quitar espacio"
        for (Cancion aux : listaCanciones) {
            salida.format("%d %s %s %s %s %s %s %.2f\n",
                aux.getClave(),
                quitarEspacios(aux.getNombreCancion()),
                quitarEspacios(aux.getNombreCantante()),
                quitarEspacios(aux.getGenero()),
                quitarEspacios(aux.getPortada()),
                quitarEspacios(aux.getAlbum()),
                quitarEspacios(aux.getArchivo()),
                aux.getPrecio());
        }
    }
    private void leer() {
        while (input.hasNext()) {
            //ESCRIBIR CÓDIGO
            //recuperar datos crear instancia y se agrega a la lista
            int clave = input.nextInt();
            String NombreCancion = ponerEspacios(input.next());
            String NombreCantante = ponerEspacios(input.next());
            String Genero = ponerEspacios(input.next());
            String Portada = ponerEspacios(input.next());
            String Album = ponerEspacios(input.next());
            String Archivo = ponerEspacios(input.next());
            double goles = input.nextDouble();
            Cancion registro = new Cancion(clave,
                NombreCancion, NombreCantante, Genero,
                Portada, Album, Archivo, goles);
        }
    }
}
```

```
        listaCanciones.add(registro);
    } // end while
}

public void EscribirArchivo(ArrayList<Cancion> lista) {
    listaCanciones = lista;
    try {
        salida = new Formatter(nombreArchivo);
    } // end try
    catch (SecurityException securityException) {
        System.err.println("No tienes permisos de escritura.");
        System.exit(1);
    } // end catch
    catch (FileNotFoundException filesNotFoundException) {
        System.err.println("Error al crear el archivo.");
        System.exit(1);
    } // end catch
    try {
        escribir();
    } // end try
    catch (FormatterClosedException formatterClosedException) {
        System.err.println("Error al escribir el archivo.");
        return;
    } // end catch
    catch (NoSuchElementException elementException) {
        System.err.println("Entrada no valida. Intente nuevamente.");
        input.nextLine();
    } // end catch
    if (salida != null) {
        salida.close();
    }
} //EscribirArchivo
// permite leer el archivo
public ArrayList<Cancion> leerArchivo() {
    try {
        input = new Scanner(new File(nombreArchivo));
    } // end try
    catch (FileNotFoundException fileNotFoundException) {
        System.err.println("Error al abrir el archivo.");
        System.exit(1);
    }
    try // read records from file using Scanner object
    {
        leer();
    } // end try
    catch (NoSuchElementException elementException) {
        System.err.println("Formato de archivo incorrecto.");
        input.close();
        System.exit(1);
    } // end catch
    catch (IllegalStateException stateException) {
        System.err.println("Error al leer de archivo.");
        System.exit(1);
    } // end catch

    if (input != null) {
        input.close(); // cierra el archivo
    }
    return listaCanciones;
}

private String quitarEspacios(String cadena) {
    return cadena.replace(" ", "#");
}
```

```
    }
    private String ponerEspacios(String cadena) {
        return cadena.replace("#", " ");
    }
} //fin de clase
```

ArchivoCompras.java

```
package proyectotercerparcial;
/**
 *
 * @author YAEL
 */
import java.util.*;
import java.io.*;
public class ArchivoCompras {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<Cancion> listaCancionesCompradas = new ArrayList<>();
    public ArchivoCompras(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
    }
    private void escribir() {
        //ESCRIBIR CÓDIGO
        //todos los elementos que sean de tipo cadena(String) se le debe poner
        //"quitar espacio"
        for (Cancion aux : listaCancionesCompradas) {
            salida.format("%d %s %s %s %s %s %s %.2f\n",
                aux.getClave(),
                quitarEspacios(aux.getNombreCancion()),
                quitarEspacios(aux.getNombreCantante()),
                quitarEspacios(aux.getGenero()),
                quitarEspacios(aux.getPortada()),
                quitarEspacios(aux.getAlbum()),
                quitarEspacios(aux.getArchivo()),
                aux.getPrecio());
        }
    }
    private void leer() {
        while (input.hasNext()) {
            //ESCRIBIR CÓDIGO
            //recuperar datos crear instancia y se agrega a la lista
            int clave = input.nextInt();
            String NombreCancion = ponerEspacios(input.next());
            String NombreCantante = ponerEspacios(input.next());
            String Genero = ponerEspacios(input.next());
            String Portada = ponerEspacios(input.next());
            String Album = ponerEspacios(input.next());
            String Archivo = ponerEspacios(input.next());
            double goles = input.nextDouble();
            Cancion registro = new Cancion(clave,
                NombreCancion, NombreCantante, Genero,
                Portada, Album, Archivo, goles);
            listaCancionesCompradas.add(registro);
        } // end while
    }
}
```

```
public void EscribirArchivo(ArrayList<Cancion> lista) {
    listaCancionesCompradas = lista;
    try {
        salida = new Formatter(nombreArchivo);
    } // end try
    catch (SecurityException securityException) {
        System.err.println("No tienes permisos de escritura.");
        System.exit(1);
    } // end catch
    catch (FileNotFoundException filesNotFoundException) {
        System.err.println("Error al crear el archivo.");
        System.exit(1);
    } // end catch
    try {
        escribir();
    } // end try
    catch (FormatterClosedException formatterClosedException) {
        System.err.println("Error al escribir el archivo.");
        return;
    } // end catch
    catch (NoSuchElementException elementException) {
        System.err.println("Entrada no valida. Intente nuevamente.");
        input.nextLine();
    } // end catch
    if (salida != null) {
        salida.close();
    }
} //EscribirArchivo
// permite leer el archivo
public ArrayList<Cancion> leerArchivo() {
    try {
        input = new Scanner(new File(nombreArchivo));
    } // end try
    catch (FileNotFoundException fileNotFoundException) {
        System.err.println("Error al abrir el archivo.");
        System.exit(1);
    }
    try // read records from file using Scanner object
    {
        leer();
    } // end try
    catch (NoSuchElementException elementException) {
        System.err.println("Formato de archivo incorrecto.");
        input.close();
        System.exit(1);
    } // end catch
    catch (IllegalStateException stateException) {
        System.err.println("Error al leer de archivo.");
        System.exit(1);
    } // end catch
    if (input != null) {
        input.close(); // cierra el archivo
    }
    return listaCancionesCompradas;
}
private String quitarEspacios(String cadena) {
    return cadena.replace(" ", "#");
}
private String ponerEspacios(String cadena) {
    return cadena.replace("#", " ");
}
```

```
//fin de class Archivo
```

ArchivoPerfilUsuario.java

```
package proyectotercerparcial;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Formatter;
import java.util.FormatterClosedException;
import java.util.NoSuchElementException;
import java.util.Scanner;
/**
 *
 * @author YAEL
 */
public class ArchivoPerfilUsuario {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<RegistroPerfilUsuario> listaRegistroCuentas;
    public ArchivoPerfilUsuario(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
        listaRegistroCuentas = new ArrayList<>();
    }
    private void escribir() {
        for (RegistroPerfilUsuario aux : listaRegistroCuentas) {
            salida.format("%d %s %s %.2f\n",
                aux.getCuenta(),
                quitarEspacios(aux.getNombre()),
                quitarEspacios(aux.getApellido()),
                aux.getDinero());
        }
    }
    private void leer() {
        while (input.hasNext()) {
            int cuenta = input.nextInt();
            String nombre = ponerEspacios(input.next());
            String apellido = ponerEspacios(input.next());
            double dinero = input.nextDouble();
            RegistroPerfilUsuario registro = new RegistroPerfilUsuario(cuenta, nombre,
                apellido, dinero);
            listaRegistroCuentas.add(registro);
        } // end while
    }
    public void EscribirArchivo(ArrayList<RegistroPerfilUsuario> lista) {
        listaRegistroCuentas = lista;
        try {
            salida = new Formatter(nombreArchivo);
        } // end try
        catch (SecurityException securityException) {
            System.err.println("No tienes permisos de escritura.");
            System.exit(1);
        } // end catch
        catch (FileNotFoundException filesNotFoundException) {
            System.err.println("Error al crear el archivo.");
            System.exit(1);
        }
    }
}
```

```

    } // end catch
    try {
        escribir();
    } // end try
    catch (FormatterClosedException formatterClosedException) {
        System.err.println("Error al escribir el archivo.");
        return;
    } // end catch
    catch (NoSuchElementException elementException) {
        System.err.println("Entrada no valida. Intente nuevamente.");
        input.nextLine();
    } // end catch
    if (salida != null) {
        salida.close();
    }
} //EscribirArchivo
// permite leer el archivo
public ArrayList<RegistroPerfilUsuario> leerArchivo() {
    try {
        input = new Scanner(new File(nombreArchivo));
    } // end try
    catch (FileNotFoundException fileNotFoundException) {
        System.err.println("Error al abrir el archivo.");
        System.exit(1);
    }
    try // read records from file using Scanner object
    {
        leer();
    } // end try
    catch (NoSuchElementException elementException) {
        System.err.println("Formato de archivo incorrecto.");
        input.close();
        System.exit(1);
    } // end catch
    catch (IllegalStateException stateException) {
        System.err.println("Error al leer de archivo.");
        System.exit(1);
    } // end catch
    if (input != null) {
        input.close(); // cierra el archivo
    }
    return listaRegistroCuentas;
}
private String quitarEspacios(String cadena) {
    return cadena.replace(" ", "#");
}
private String ponerEspacios(String cadena) {
    return cadena.replace("#", " ");
}
}

} //fin de class Archivo

```

ArchivoSaldo.java

```

package proyectotercerparcial;
import java.io.File;
import java.io.FileNotFoundException;

```



```
import java.util.ArrayList;
import java.util.Formatter;
import java.util.FormatterClosedException;
import java.util.NoSuchElementException;
import java.util.Scanner;
/**
 *
 * @author YAEL
 */
public class ArchivoSaldo {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<Saldo> listaSaldo;
    public ArchivoSaldo(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
        listaSaldo = new ArrayList<>();
    }
    private void escribir() {
        for (Saldo aux : listaSaldo) {
            salida.format("%.2f\n",
                aux.obtenerSaldo());
        }
    }
    private void leer() {
        while (input.hasNext()) {
            double saldo = input.nextDouble();
            System.out.println("'" + saldo);
            Saldo s1 = new Saldo();
            s1.establecerSaldo(saldo);
            listaSaldo.add(s1);
        } // end while
    }
    public void EscribirArchivo(ArrayList<Saldo> lista) {
        listaSaldo = lista;
        try {
            salida = new Formatter(nombreArchivo);
        } // end try
        catch (SecurityException securityException) {
            System.err.println("No tienes permisos de escritura.");
            System.exit(1);
        } // end catch
        catch (FileNotFoundException filesNotFoundException) {
            System.err.println("Error al crear el archivo.");
            System.exit(1);
        } // end catch
        try {
            escribir();
        } // end try
        catch (FormatterClosedException formatterClosedException) {
            System.err.println("Error al escribir el archivo.");
            return;
        } // end catch
        catch (NoSuchElementException elementException) {
            System.err.println("Entrada no valida. Intente nuevamente.");
            input.nextLine();
        } // end catch
        if (salida != null) {
            salida.close();
        }
    }
} //EscribirArchivo
```

```
// permite leer el archivo
public ArrayList<Saldo> leerArchivo() {
    try {
        input = new Scanner(new File(nombreArchivo));
    } // end try
    catch (FileNotFoundException fileNotFoundException) {
        System.err.println("Error al abrir el archivo.");
        System.exit(1);
    }
    try // read records from file using Scanner object
    {
        leer();
    } // end try
    catch (NoSuchElementException elementException) {
        System.err.println("Formato de archivo incorrecto.");
        input.close();
        System.exit(1);
    } // end catch
    catch (IllegalStateException stateException) {
        System.err.println("Error al leer de archivo.");
        System.exit(1);
    } // end catch
    if (input != null) {
        input.close(); // cierra el archivo
    }
    return listaSaldo;
}
private String quitarEspacios(String cadena) {
    return cadena.replace(" ", "#");
}
private String ponerEspacios(String cadena) {
    return cadena.replace("#", " ");
}
}
} //fin de class Archivo
```

Cancion.java

```
package proyectotercerparcial;
import java.util.Locale;
/**
 *
 * @author YAEL
 */
public class Cancion {
    private int Clave;
    private String NombreCancion;
    private String NombreCantante;
    private String Genero;
    private String Portada;
    private String Album;
    private String Archivo;
    private double Precio;
    public Cancion() {
        this(0, "", "", "", "", "", "", 0.0);
    }
    public Cancion(int Clave, String NombreCancion, String NombreCantante,
        String Genero, String Portada, String Album, String Archivo, double Precio) {
```

```
this.Clave = Clave;
this.NombreCancion = NombreCancion;
this.NombreCantante = NombreCantante;
this.Genero = Genero;
this.Portada = Portada;
this.Album = Album;
this.Archivo = Archivo;
this.Precio = Precio;
}
public void setClave(int Clave) {
    this.Clave = Clave;
}
public int getClave() {
    return this.Clave;
}
public void setNombreCancion(String NombreCancion) {
    this.NombreCancion = NombreCancion;
}
public String getNombreCancion() {
    return this.NombreCancion;
}
public void setNombreCantante(String NombreCantante) {
    this.NombreCantante = NombreCantante;
}
public String getNombreCantante() {
    return this.NombreCantante;
}
public void setGenero(String Genero) {
    this.Genero = Genero;
}
public String getGenero() {
    return this.Genero;
}
public void setPortada(String Portada) {
    this.Portada = Portada;
}
public String getPortada() {
    return this.Portada;
}
public void setAlbum(String Album) {
    this.Album = Album;
}
public String getAlbum() {
    return this.Album;
}
public void setArchivo(String Archivo) {
    this.Archivo = Archivo;
}
public String getArchivo() {
    return this.Archivo;
}
public void setPrecio(double Precio) {
    this.Precio = Precio;
}
public double getPrecio() {
    return this.Precio;
}
@Override
public String toString() {
    return String.format("\tDatos de la cancion\n\nClave: %d \nNombreCancion: %s"
        + "\nNombreCantante: %s\nGenero: %s\nPortada: %s"
```

```
        + "\nAlbum: %s\nArchivo: %s\nPrecio: %.2f",
        this.Clave, this.NombreCancion, this.NombreCantante, this.Genero,
        this.Portada, this.Album, this.Archivo, this.Precio);
    }
}
```

Saldo.java

```
package proyectotercerparcial;
/**
 *
 * @author YAEL
 */
public class Saldo {
    private double saldoTotal;
    public void agregarSaldo(double credito) {
        saldoTotal = saldoTotal + credito;
    }
    public void restarSaldo(double gasto) {
        saldoTotal = saldoTotal - gasto;
    }
    public double obtenerSaldo() {
        return saldoTotal;
    }
    public void establecerSaldo(double saldo) {
        this.saldoTotal = saldo;
    }
}
```

RegistroPerfilUsuario.java

```
package proyectotercerparcial;
/**
 *
 * @author YAEL
 */
public class RegistroPerfilUsuario {
    private int cuenta;
    private String nombre;
    private String apellido;
    private double dinero;
    public RegistroPerfilUsuario(int cuenta, String nombre, String apellido,
        double dinero) {
        this.cuenta = cuenta;
        this.nombre = nombre;
        this.apellido = apellido;
        this.dinero = dinero;
    }
    public int getCuenta() {
        return cuenta;
    }
    public String getNombre() {
        return nombre;
    }
}
```

```
public String getApellido() {
    return apellido;
}
public void setCuenta(int cuenta) {
    this.cuenta = cuenta;
}
public void setNombre(String nombre) {
    this.nombre = nombre;
}
public void setApellido(String apellido) {
    this.apellido = apellido;
}
public void setDinero(double dinero) {
    this.dinero = dinero;
}
public double getDinero() {
    return dinero;
}
}
```

PaginaWebPruebas.java

```
package proyectotercerparcial;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Formatter;
import java.util.FormatterClosedException;
import java.util.NoSuchElementException;
import java.util.Scanner;
/**
 *
 * @author Yael
 */
public class PaginaWebPruebas {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<Cancion> listaRegistroCanciones = new ArrayList<>();
    public PaginaWebPruebas(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
    }
    private void escribirPaginaWeb() {
        salida.format("<!doctype html>\n");
        salida.format("<!doctype html>\n");
        salida.format("<html>\n");
        salida.format("<head>\n");
        salida.format("<meta charset='UTF-8'>\n");
        salida.format("<title>ROKU MUSIC</title>\n");
        salida.format("<link rel='stylesheet'\n");
        salida.format("<href='css/bootstrap.min.css'\n");
        salida.format("<media='screen'>\n");
        salida.format("<style>\n");
        salida.format("#cuerpo{\n");
        salida.format("margin: 100px;\n");
        salida.format("}\n");
        salida.format("</style>\n");
        salida.format("</head>\n");
    }
}
```

```

//salida.format("<body>\n");
salida.format("<body background='BACKGROUND/1f58189811495ed16cd3857e8eeaf9b6.jpg'>\n");
//salida.format("<table border='1'>\n");
salida.format("<font color='black'>\n");
salida.format("<div id='cuerpo'>\n");
salida.format("<h1>Canciones Disponibles</h1>\n");
salida.format("<table class='table table-striped'>\n");
salida.format("<thead>\n");
salida.format("<tr>\n");
salida.format("<th>Clave</th>\n");
salida.format("<th>Nombre de la cancion</th>\n");
salida.format("<th>Nombre del cantante</th>\n");
salida.format("<th>Genero</th>\n");
salida.format("<th>Portada</th>\n");
salida.format("<th>Album</th>\n");
salida.format("<th>Archivo</th>\n");
salida.format("<th>Precio</th>\n");
salida.format("</tr>\n");
salida.format("</thead>\n");
salida.format("<tbody>\n");
for (Cancion aux : listaRegistroCanciones) {
    salida.format("<tr>\n");
    salida.format("<td>%d</td>\n", aux.getClave());
    salida.format("<td>%s</td>\n", aux.getNombreCancion());
    salida.format("<td>%s</td>\n", aux.getNombreCantante());
    salida.format("<td>%s</td>\n", aux.getGenero());
    salida.format("<td><img src = 'img/' + aux.getPortada() + '.jpg' width = '65'height = '50' ></td>\n");
    salida.format("<td>%s</td>\n", aux.getAlbum());
    salida.format("<td><audio src='CancionesPrueba/' + aux.getArchivo() + '.mp3'\>\n" + "</audio></td>\n");
    //salida.format("<td>%s</td>\n", aux.getArchivo());
    salida.format("<td>%.2f</td>\n", aux.getPrecio());
    salida.format("</tr>\n");
}
salida.format("</tbody>\n");
salida.format("</table>\n");
salida.format("</div>\n");
salida.format("<script src='js/jquery.min.js'></script>\n");
salida.format("<script src='js/bootstrap.min.js'></script>\n");
//salida.format("</table>\n");
salida.format("</body>\n");
salida.format("</html>\n");
}

public void EscribirArchivo(ArrayList<Cancion> lista) {
    listaRegistroCanciones = lista;
    try {
        salida = new Formatter(nombreArchivo);
    } // end try
    catch (SecurityException securityException) {
        System.err.println("No tienes permisos de escritura.");
        System.exit(1);
    } // end catch
    catch (FileNotFoundException filesNotFoundException) {
        System.err.println("Error al crear el archivo.");
        System.exit(1);
    } // end catch
    try {
        escribirPaginaWeb();
    } // end try
    catch (FormatterClosedException formatterClosedException) {
        System.err.println("Error al escribir el archivo.");
        return;
    }
}

```

```

    } // end catch
    catch (NoSuchElementException elementException) {
        System.err.println("Entrada no valida. Intente nuevamente.");
        input.nextLine();
    } // end catch
    if (salida != null) {
        salida.close();
    }
} //EscribirArchivo
} //fin de class PaginaWeb

```

PaginaWebCompras.java

```

package proyectotercerparcial;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Formatter;
import java.util.FormatterClosedException;
import java.util.NoSuchElementException;
import java.util.Scanner;
/**
 *
 * @author YAEL
 */
public class PaginaWebCompras {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<Cancion> listaCancionesCompradas = new ArrayList<>();
    public PaginaWebCompras(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
    }
    private void escribirPaginaWeb() {
        salida.format("<!doctype html>\n");
        salida.format("<!doctype html>\n");
        salida.format("<html>\n");
        salida.format("<head>\n");
        salida.format("<meta charset='UTF-8'>\n");
        salida.format("<title>ROKU MUSIC</title>\n");
        salida.format("<link rel='stylesheet'\n");
        salida.format("<href='css/bootstrap.min.css'\n");
        salida.format("<media='screen'>\n");
        salida.format("<style>\n");
        salida.format("#cuerpo{\n");
        salida.format("margin: 100px;\n");
        salida.format("}\n");
        salida.format("</style>\n");
        salida.format("</head>\n");
        salida.format("<body background='BACKGROUND/1f58189811495ed16cd3857e8eeaf9b6.jpg'>\n");
        salida.format("<font color='black'>\n");
        salida.format("<div id='cuerpo'>\n");
        salida.format("<h1>Canciones Compradas</h1>\n");
        salida.format("<table class='table table-striped'>\n");
        salida.format("<thead>\n");
        salida.format("<tr>\n");
        salida.format("<th>Clave</th>\n");
        salida.format("<th>Nombre de la cancion</th>\n");
    }
}

```

```

        salida.format("<th>Nombre del cantante</th>\n");
        salida.format("<th>Genero</th>\n");
        salida.format("<th>Portada</th>\n");
        salida.format("<th>Album</th>\n");
        salida.format("<th>Archivo</th>\n");
        salida.format("<th>Precio</th>\n");
        salida.format("</tr>\n");
        salida.format("</thead>\n");
        salida.format("<tbody>\n");
        for (Cancion aux : listaCancionesCompradas) {
            salida.format("<tr>\n");
            salida.format("<td>%d</td>\n", aux.getClave());
            salida.format("<td>%s</td>\n", aux.getNombreCancion());
            salida.format("<td>%s</td>\n", aux.getNombreCantante());
            salida.format("<td>%s</td>\n", aux.getGenero());
            salida.format("<td><img src = 'img/" + aux.getPortada() + ".jpg' width = '65'height = '50' ></td>\n");
            salida.format("<td>%s</td>\n", aux.getAlbum());
            salida.format("<td><audio src=\"CancionesCompletas/" + aux.getArchivo() + ".mp3\" controls>\n" + "</audio></td>\n");
            //salida.format("<td>%s</td>\n", aux.getArchivo());
            salida.format("<td>%.2f</td>\n", aux.getPrecio());
            salida.format("</tr>\n");
        }
        salida.format("</tbody>\n");
        salida.format("</table>\n");
        salida.format("</div>\n");
        salida.format("<script src='js/jquery.min.js'></script>\n");
        salida.format("<script src='js/bootstrap.min.js'></script>\n");
        salida.format("</body>\n");
        salida.format("</html>\n");
    }
    public void EscribirArchivo(ArrayList<Cancion> lista) {
        listaCancionesCompradas = lista;
        try {
            salida = new Formatter(nombreArchivo);
        } // end try
        catch (SecurityException securityException) {
            System.err.println("No tienes permisos de escritura.");
            System.exit(1);
        } // end catch
        catch (FileNotFoundException filesNotFoundException) {
            System.err.println("Error al crear el archivo.");
            System.exit(1);
        } // end catch
        try {
            escribirPaginaWeb();
        } // end try
        catch (FormatterClosedException formatterClosedException) {
            System.err.println("Error al escribir el archivo.");
            return;
        } // end catch
        catch (NoSuchElementException elementException) {
            System.err.println("Entrada no valida. Intente nuevamente.");
            input.nextLine();
        } // end catch
        if (salida != null) {
            salida.close();
        }
    } //EscribirArchivo
} //fin de class PaginaWeb

```


PaginaWebAdmin.java

```
package proyectotercerparcial;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Formatter;
import java.util.FormatterClosedException;
import java.util.NoSuchElementException;
import java.util.Scanner;
/**
 *
 * @author Yael
 */
public class PaginaWebAdmin {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<Cancion> listaCanciones = new ArrayList<>();
    public PaginaWebAdmin(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
    }
    private void escribirPaginaWeb() {
        salida.format("<!doctype html>\n");
        salida.format("<!doctype html>\n");
        salida.format("<html>\n");
        salida.format("<head>\n");
        salida.format("<meta charset='UTF-8'>\n");
        salida.format("<title>ROKU MUSIC</title>\n");
        salida.format("<link rel='stylesheet'\n");
        salida.format("<href='css/bootstrap.min.css'\n");
        salida.format("<media='screen'>\n");
        salida.format("<style>\n");
        salida.format("#cuerpo{\n");
        salida.format("margin: 100px;\n");
        salida.format("}\n");
        salida.format("</style>\n");
        salida.format("</head>\n");
        //salida.format("<body>\n");
        salida.format("<body background='BACKGROUND/1f58189811495ed16cd3857e8eeaf9b6.jpg'>\n");
        //salida.format("<table border='1'>\n");
        salida.format("<font color='black'>\n");
        salida.format("<div id='cuerpo'>\n");
        salida.format("<h1>Canciones registradas</h1>\n");
        salida.format("<table class='table table-striped'>\n");
        salida.format("<thead>\n");
        salida.format("<tr>\n");
        salida.format("<th>Clave</th>\n");
        salida.format("<th>Nombre de la cancion</th>\n");
        salida.format("<th>Nombre del cantante</th>\n");
        salida.format("<th>Genero</th>\n");
        salida.format("<th>Portada</th>\n");
        salida.format("<th>Album</th>\n");
        salida.format("<th>Archivo</th>\n");
        salida.format("<th>Precio</th>\n");
        salida.format("</tr>\n");
        salida.format("</thead>\n");
        salida.format("<tbody>\n");
        for (Cancion aux : listaCanciones) {
```

```

        salida.format("<tr>\n");
        salida.format("<td>%d</td>\n", aux.getClave());
        salida.format("<td>%s</td>\n", aux.getNombreCancion());
        salida.format("<td>%s</td>\n", aux.getNombreCantante());
        salida.format("<td>%s</td>\n", aux.getGenero());
        salida.format("<td><img src = 'img/' + aux.getPortada() + '.jpg' width = '65' height = '50' ></td>\n");
        salida.format("<td>%s</td>\n", aux.getAlbum());
        salida.format("<td><audio src='CancionesCompletas/' + aux.getArchivo() + '.mp3'\>\n" + "</audio></td>\n");
        //salida.format("<td>%s</td>\n", aux.getArchivo());
        salida.format("<td>%.2f</td>\n", aux.getPrecio());
        salida.format("</tr>\n");
    }
    salida.format("</tbody>\n");
    salida.format("</table>\n");
    salida.format("</div>\n");
    salida.format("<script src='js/jquery.min.js'></script>\n");

    salida.format("<script src='js/bootstrap.min.js'></script>\n");
    //salida.format("</table>\n");
    salida.format("</body>\n");
    salida.format("</html>\n");
}

public void EscribirArchivo(ArrayList<Cancion> lista) {
    listaCanciones = lista;
    try {
        salida = new Formatter(nombreArchivo);
    } // end try
    catch (SecurityException securityException) {
        System.err.println("No tienes permisos de escritura.");
        System.exit(1);
    } // end catch
    catch (FileNotFoundException filesNotFoundException) {
        System.err.println("Error al crear el archivo.");
        System.exit(1);
    } // end catch
    try {
        escribirPaginaWeb();
    } // end try
    catch (FormatterClosedException formatterClosedException) {
        System.err.println("Error al escribir el archivo.");
        return;
    } // end catch
    catch (NoSuchElementException elementException) {
        System.err.println("Entrada no valida. Intente nuevamente.");
        input.nextLine();
    } // end catch
    if (salida != null) {
        salida.close();
    }
} //EscribirArchivo
} //fin de class PaginaWeb

```

PaginaWeb.java

```

package proyectotercerparcial;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Formatter;

```

```

import java.util.FormatterClosedException;
import java.util.NoSuchElementException;
import java.util.Scanner;
/**
 *
 * @author YAEL
 */
public class PaginaWeb {
    private Scanner input;
    private Formatter salida;
    private String nombreArchivo = "";
    private ArrayList<Cancion> listaCanciones = new ArrayList<>();
    public PaginaWeb(String nombreArchivo) {
        this.nombreArchivo = nombreArchivo;
    }
    private void escribirPaginaWeb() {
        salida.format("<!doctype html>\n");
        salida.format("<!doctype html>\n");
        salida.format("<html>\n");
        salida.format("<head>\n");
        salida.format("<meta charset='UTF-8'>\n");
        salida.format("<title>ROKU MUSIC</title>\n");
        salida.format("<link rel='stylesheet'\n");
        salida.format("href='css/bootstrap.min.css'\n");
        salida.format("media='screen'>\n");
        salida.format("<style>\n");
        salida.format("#cuerpo{\n");
        salida.format("margin: 100px;\n");
        salida.format("}\n");
        salida.format("</style>\n");
        salida.format("</head>\n");
        //salida.format("<body>\n");
        salida.format("<body background='BACKGROUND/1f58189811495ed16cd3857e8eeaf9b6.jpg'>\n");
        //salida.format("<table border='1'>\n");
        salida.format("<font color='black'>\n");
        salida.format("<div id='cuerpo'>\n");
        salida.format("<h1>Canciones disponibles</h1>\n");
        salida.format("<table class='table table-striped'>\n");
        salida.format("<thead>\n");
        salida.format("<tr>\n");
        salida.format("<th>Clave</th>\n");
        salida.format("<th>Nombre de la cancion</th>\n");
        salida.format("<th>Nombre del cantante</th>\n");
        salida.format("<th>Genero</th>\n");
        salida.format("<th>Portada</th>\n");
        salida.format("<th>Album</th>\n");
        salida.format("<th>Archivo</th>\n");
        salida.format("<th>Precio</th>\n");
        salida.format("</tr>\n");
        salida.format("</thead>\n");
        salida.format("<tbody>\n");
        for (Cancion aux : listaCanciones) {
            salida.format("<tr>\n");
            salida.format("<td>%d</td>\n", aux.getClave());
            salida.format("<td>%s</td>\n", aux.getNombreCancion());
            salida.format("<td>%s</td>\n", aux.getNombreCantante());
            salida.format("<td>%s</td>\n", aux.getGenero());
            salida.format("<td><img src = 'img/' + aux.getPortada() + '.jpg' width = '65'height = '50' ></td>\n");
            salida.format("<td>%s</td>\n", aux.getAlbum());
            salida.format("<td><audio src='CancionesPrueba/' + aux.getArchivo() + '.mp3' controls>\n" + "</audio></td>\n");
            //salida.format("<td>%s</td>\n", aux.getArchivo());
            salida.format("<td>%2f</td>\n", aux.getPrecio());
        }
    }
}

```

```

        salida.format("</tr>\n");
    }
    salida.format("</tbody>\n");
    salida.format("</table>\n");
    salida.format("</div>\n");
    salida.format("<script src='js/jquery.min.js'></script>\n");
    salida.format("<script src='js/bootstrap.min.js'></script>\n");
    //salida.format("</table>\n");
    salida.format("</body>\n");
    salida.format("</html>\n");
}

public void EscribirArchivo(ArrayList<Cancion> lista) {
    listaCanciones = lista;
    try {
        salida = new Formatter(nombreArchivo);
    } // end try
    catch (SecurityException securityException) {
        System.err.println("No tienes permisos de escritura.");
        System.exit(1);
    } // end catch
    catch (FileNotFoundException filesNotFoundException) {
        System.err.println("Error al crear el archivo.");
        System.exit(1);
    } // end catch
    try {
        escribirPaginaWeb();
    } // end try
    catch (FormatterClosedException formatterClosedException) {
        System.err.println("Error al escribir el archivo.");
        return;
    } // end catch
    catch (NoSuchElementException elementException) {
        System.err.println("Entrada no valida. Intente nuevamente.");
        input.nextLine();
    } // end catch
    if (salida != null) {
        salida.close();
    }
} //EscribirArchivo
} //fin de class PaginaWeb

```

FrmAlbumAdministrador.java

```

package proyectotercerparcial;
import java.awt.Desktop;
import java.util.ArrayList;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import java.awt.Image;
import java.io.File;
import java.io.IOException;
import java.util.Iterator;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
/**
 *
 * @author YAEL

```

```
*/
public class FrmAlbumAdministrador extends javax.swing.JFrame {
    int noClave = -1;
    Archivo archivo = new Archivo("Canciones.txt");
    Archivo archivoCompras = new Archivo("CancionesCompradas.txt");
    ArrayList<Cancion> listaCanciones = new ArrayList<>();
    ArrayList<Cancion> listaCancionesCompradas = new ArrayList<>();
    public FrmAlbumAdministrador() {
        initComponents();
        setLocationRelativeTo(null);
        this.setIconImage(new ImageIcon(getClass().getResource("/Iconos/101-headset.png")).getImage());
        listaCanciones = archivo.leerArchivo();
        llenarTablaCanciones();
        listaCancionesCompradas = archivoCompras.leerArchivo();
        llenarCancionesCompradas();
    }
    private void llenarTablaCanciones() {
        DefaultTableModel llenar
            = (DefaultTableModel) tablaCanciones
                .getModel();
        Object[] fila = new Object[8];
        for (int i = 0; i < tablaCanciones.getRowCount(); i++) {
            llenar.removeRow(i);
            i = i - 1;
        }
        //llenar tabla
        for (Cancion aux : listaCanciones) {
            fila[0] = aux.getClave();
            fila[1] = aux.getNombreCancion();
            fila[2] = aux.getNombreCantante();
            fila[3] = aux.getGenero();
            fila[4] = aux.getPortada();
            fila[5] = aux.getAlbum();
            fila[6] = aux.getArchivo();
            fila[7] = aux.getPrecio();
            llenar.addRow(fila);
        }
    }
    public void LimpiarCajas() {
        txtClave.setText("");
        txtNombreCancion.setText("");
        txtNombreCantante.setText("");
        txtGenero.setText("");
        txtPortada.setText("");
        txtAlbum.setText("");
        txtArchivo.setText("");
        txtPrecio.setText("");
    }
    private void llenarCancionesCompradas() {
        DefaultTableModel llenar
            = (DefaultTableModel) tablaCompradas
                .getModel();
        Object[] fila = new Object[8];
        for (int i = 0; i < tablaCompradas.getRowCount(); i++) {
            llenar.removeRow(i);
            i = i - 1;
        }
        //llenar tabla
        for (Cancion aux : listaCancionesCompradas) {
            fila[0] = aux.getClave();
            fila[1] = aux.getNombreCancion();
```

```
        fila[2] = aux.getNombreCantante();
        fila[3] = aux.getGenero();
        fila[4] = aux.getPortada();
        fila[5] = aux.getAlbum();
        fila[6] = aux.getArchivo();
        fila[7] = aux.getPrecio();
        llenar.addRow(fila);
    }
}
private boolean cuentaRepetida() {
    for (Cancion aux : listaCanciones) {
        if (noClave == aux.getClave()) {
            return true;
        }
    }
    return false;
}
private boolean buscar() {
    for (Cancion aux : listaCancionesCompradas) {
        if (noClave == aux.getClave()) {
            return true;
        }
    }
    return false;
}
public boolean verificarCajas() {
    try {
        if (txtClave.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner la clave");
        }
        if (txtNombreCancion.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el nombre de la cancion");
        }
        if (txtNombreCantante.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el nombre del cantante");
        }
        if (txtGenero.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el genero");
        }
        if (txtPortada.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el nombre de la portada");
        }
        if (txtAlbum.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el nombre del album");
        }
        if (txtArchivo.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el nombre del archivo");
        }
        if (txtPrecio.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el precio");
        }
    } catch (Exception ex) {
```



```

        Image.SCALE_DEFAULT));
        image.setIcon(icono2);
        this.repaint();
        return;
    }
}
String path = "img/no.jpg";
ImageIcon fot = new ImageIcon(path);
Icon icono2 = new ImageIcon(fot.getImage().getScaledInstance(
    image.getWidth(), image.getHeight(),
    Image.SCALE_DEFAULT));
image.setIcon(icono2);
this.repaint();
txtDatos.setText("No se encontro");
} catch (NumberFormatException ex) {
    System.out.println("Error:" + ex);
    JOptionPane.showMessageDialog(this, "Debe ingresar valores enteros",
        "Error", 0);
}
}

```

Boton borrar

```

private void btnBorrarActionPerformed(java.awt.event.ActionEvent evt) {
    if (noClave == -1) {
        JOptionPane.showMessageDialog(this, "Debe tener un registro activo");
        return;
    }
    int boton = JOptionPane.showConfirmDialog(this, "Desea borrar este elemento?",
        "Advertencia", JOptionPane.YES_NO_OPTION);
    if (boton == JOptionPane.YES_OPTION) {
        Iterator<Cancion> iteratorX = listaCanciones.iterator();
        while (iteratorX.hasNext()) {
            if (iteratorX.next().getClave() == noClave) {
                iteratorX.remove();
                JOptionPane.showMessageDialog(this, "Eliminado");
                archivo.EscribirArchivo(listaCanciones);
                noClave = -1;
                llenarTablaCanciones();
            }
        }
    }
}

```

Boton cambiar

```

private void btnCambiarActionPerformed(java.awt.event.ActionEvent evt) {
    if (noClave == -1) {
        JOptionPane.showMessageDialog(this, "Debe tener un registro activo");
        return;
    }
    for (Cancion aux : listaCanciones) {
        if (noClave == aux.getClave()) {
            FrmEditor frm = new FrmEditor(aux);
            frm.setVisible(true);
            this.dispose();
        }
    }
}

```

Boton buscar

```

private void btnBuscarClaveActionPerformed(java.awt.event.ActionEvent evt) {
    if (txtClaveBuscar.getText().compareTo("") == 0) {
        llenarTablaCanciones();
        return;
    }
}

```



```

        int claveB;
        try {
            claveB = Integer.parseInt(txtClaveBuscar.getText());
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Debe escribir numero");
            return;
        }
        DefaultTableModel llenar = (DefaultTableModel) tablaCanciones.getModel();
        Object[] fila = new Object[8];
        //limpiar tabla
        for (int i = 0; i < tablaCanciones.getRowCount(); i++) {
            llenar.removeRow(i);
            i = i - 1;
        }
        //PONER EL ELEMENTO EN LA TABLA, SI LO ENCUENTRA
        for (Cancion aux : listaCanciones) {
            if (aux.getClave() == claveB) {
                fila[0] = aux.getClave();
                fila[1] = aux.getNombreCancion();
                fila[2] = aux.getNombreCantante();
                fila[3] = aux.getGenero();
                fila[4] = aux.getPortada();
                fila[5] = aux.getAlbum();
                fila[6] = aux.getArchivo();
                fila[7] = aux.getPrecio();
                llenar.addRow(fila);
            }
        }
    }

    private void tablaCancionesKeyReleased(java.awt.event.KeyEvent evt) {
        if (tablaCanciones.getSelectedRows().length > 0) {
            int fila = tablaCanciones.getSelectedRow();
            noClave = Integer.parseInt(String.valueOf(
                tablaCanciones.getValueAt(fila, 0)));
        }
    }

    private void tablaCancionesMouseClicked(java.awt.event.MouseEvent evt) {
        int fila = tablaCanciones.getSelectedRow();
        noClave = Integer.parseInt(String.valueOf(
            tablaCanciones.getValueAt(fila, 0)));
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        FrmLoginAdministrador P1 = new FrmLoginAdministrador();
        P1.setVisible(true);
        dispose();
    }
}

Boton ver pagina
private void btnVerPaginalActionPerformed(java.awt.event.ActionEvent evt) {
    PaginaWebCompras pagweb = new PaginaWebCompras("Canciones.html");
    pagweb.EscribirArchivo(listaCancionesCompradas);
    //abrir el archivo en el programador
    try {
        File pagina = new File("Canciones.html");
        Desktop.getDesktop().open(pagina);
    } catch (IOException ex) {
        System.out.println(ex.getMessage());
    }
}

```

FrmAlbumUsuario.java

```
package proyectotercerparcial;
import java.awt.Desktop;
import java.awt.HeadlessException;
import java.util.ArrayList;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import java.awt.Image;
import java.io.File;
import java.io.IOException;
import java.util.Iterator;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
/**
 *
 * @author YAEL
 */
public class FrmAlbumUsuario extends javax.swing.JFrame {
    int noClave = -1;
    double saldoTotal;
    double gasto;
    double cont = 0;
    Archivo archivo = new Archivo("Canciones.txt");
    Archivo archivoCompras = new Archivo("CancionesCompradas.txt");
    ArchivoPerfilUsuario archivoPerfilUsuario = new ArchivoPerfilUsuario("Clientes.txt");
    ArchivoSaldo archivoSaldo = new ArchivoSaldo("Saldo.txt");
    ArrayList<Cancion> listaCanciones = new ArrayList<>();
    ArrayList<Cancion> listaCancionesCompradas = new ArrayList<>();
    ArrayList<RegistroPerfilUsuario> listaUsuario = new ArrayList<>();
    ArrayList<Saldo> listaSaldo = new ArrayList<>();
    public FrmAlbumUsuario() {
        initComponents();
        setLocationRelativeTo(null);
        this.setIconImage(new ImageIcon(getClass().getResource("/Iconos/101-headset.png")).getImage());
        listaCanciones = archivo.leerArchivo();
        llenarTablaCanciones();
        listaCancionesCompradas = archivoCompras.leerArchivo();
        llenarCancionesCompradas();
        //System.out.println("" + saldoTotal);
        listaSaldo = archivoSaldo.leerArchivo();
        for (Saldo aux : listaSaldo) {
            saldoTotal = aux.obtenerSaldo();
            //System.out.println("" + saldoTotal);
            txtSaldoTotal.setText("" + saldoTotal);
        }
    }
    private void llenarTablaCanciones() {
        DefaultTableModel llenar
            = (DefaultTableModel) tablaCancionesDisponibles
                .getModel();
        Object[] fila = new Object[8];
        for (int i = 0; i < tablaCancionesDisponibles.getRowCount(); i++) {
            llenar.removeRow(i);
            i = i - 1;
        }
        //llenar tabla
        for (Cancion aux : listaCanciones) {
```

```
        fila[0] = aux.getClave();
        fila[1] = aux.getNombreCancion();
        fila[2] = aux.getNombreCantante();
        fila[3] = aux.getGenero();
        fila[4] = aux.getPortada();
        fila[5] = aux.getAlbum();
        fila[6] = aux.getArchivo();
        fila[7] = aux.getPrecio();
        llenar.addRow(fila);
    }
}

public void LimpiarCajas() {
    txtNombre.setText("");
    txtApellido.setText("");
    txtSaldo.setText("");
}

private void llenarCancionesCompradas() {
    DefaultTableModel llenar
        = (DefaultTableModel) tablaCancionesCompradas
            .getModel();
    Object[] fila = new Object[8];
    for (int i = 0; i < tablaCancionesCompradas.getRowCount(); i++) {
        llenar.removeRow(i);
        i = i - 1;
    }
    //llenar tabla
    for (Cancion aux : listaCancionesCompradas) {
        fila[0] = aux.getClave();
        fila[1] = aux.getNombreCancion();
        fila[2] = aux.getNombreCantante();
        fila[3] = aux.getGenero();
        fila[4] = aux.getPortada();
        fila[5] = aux.getAlbum();
        fila[6] = aux.getArchivo();
        fila[7] = aux.getPrecio();
        llenar.addRow(fila);
    }
}

private boolean buscar() {
    for (Cancion aux : listaCancionesCompradas) {
        if (noClave == aux.getClave()) {
            return true;
        }
    }
    return false;
}

public boolean verificarCajas() {
    try {
        if (txtNombre.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner la clave");
        }
        if (txtApellido.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el nombre de la cancion");
        }
        if (txtSaldo.getText().compareTo("") == 0) {
            throw new Exception(
                "Debe poner el precio");
        }
    } catch (Exception ex) {
```

```

        JOptionPane.showMessageDialog(this,
            ex.getMessage());
        return false;
    }
    return true;
}

```

Boton Agregar Perfil

```

private void btnAgregarPerfilActionPerformed(java.awt.event.ActionEvent evt) {
    int Cuenta = Integer.parseInt(txtCuenta.getText());
    String Nombre = txtNombre.getText();
    String Apellido = txtApellido.getText();
    double dinero = Double.parseDouble(txtSaldo.getText());
    RegistroPerfilUsuario registro = new RegistroPerfilUsuario(Cuenta, Nombre, Apellido, dinero);
    listaUsuario.add(registro);
    archivoPerfilUsuario.EscribirArchivo(listaUsuario);
    JOptionPane.showMessageDialog(this, "Registro dado de alta");
    txtSaldoTotal.setText(txtSaldo.getText());
    archivoSaldo.EscribirArchivo(listaSaldo);
    // BORRA LA CAJA
    txtCuenta.setText("");
    txtNombre.setText("");
    txtApellido.setText("");
    txtSaldo.setText("");
    //archivoPerfilUsuario.EscribirArchivo(listaUsuario);
    // textArea.setText(txtNombre.getText());
    // textArea.setText(txtApellido.getText());
    //textArea.setText(txtSaldo.getText());
    //JOptionPane.showMessageDialog(this, "Registro dado de alta");
}

```

Boton buscar clave

```

private void btnBuscarClaveActionPerformed(java.awt.event.ActionEvent evt) {
    if (txtClaveBuscar.getText().compareTo("") == 0) {
        llenarTablaCanciones();
        return;
    }
    int claveB;
    try {
        claveB = Integer.parseInt(txtClaveBuscar.getText());
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Debe escribir numero");
        return;
    }
    DefaultTableModel llenar = (DefaultTableModel) tablaCancionesCompradas.getModel();
    Object[] fila = new Object[8];
    //limpiar tabla
    for (int i = 0; i < tablaCancionesCompradas.getRowCount(); i++) {
        llenar.removeRow(i);
        i = i - 1;
    }
    //PONER EL ELEMENTO EN LA TABLA, SI LO ENCUENTRA
    for (Cancion aux : listaCanciones) {
        if (aux.getClave() == claveB) {
            fila[0] = aux.getClave();
            fila[1] = aux.getNombreCancion();
            fila[2] = aux.getNombreCantante();
            fila[3] = aux.getGenero();
            fila[4] = aux.getPortada();
            fila[5] = aux.getAlbum();
            fila[6] = aux.getArchivo();
            fila[7] = aux.getPrecio();
        }
    }
}

```

```

        llenar.addRow(fila);
    }
}

}

Boton comprar
private void btnComprarActionPerformed(java.awt.event.ActionEvent evt) {
    if (noClave == -1) {
        JOptionPane.showMessageDialog(this, "Debe tener un registro activo");
        return;
    }
    if (buscar()) {
        JOptionPane.showMessageDialog(this, "Ya comprado");
        return;
    }
    for (Cancion aux : listaCanciones) {
        if (noClave == aux.getClave()) {
            listaCancionesCompradas.add(aux);
            llenarCancionesCompradas();
            listaCanciones.add(aux);
            archivoCompras.EscribirArchivo(listaCancionesCompradas);
            JOptionPane.showMessageDialog(this, "Comprada");
            return;
        }
    }
    double gasto = Double.parseDouble(txtSaldoTotal.getText());
    for (Saldo aux : listaSaldo) {
        aux.restarSaldo(gasto);
        txtSaldoTotal.setText("" + aux.obtenerSaldo());
        archivoSaldo.EscribirArchivo(listaSaldo);
    }
    //JOptionPane.showMessageDialog(this, "Comprado");
    /*int filaseleccionada = tablaCancionesDisponibles.getSelectedRow();
    try {
        String Precio;
        if (filaseleccionada == -1) {
            JOptionPane.showMessageDialog(null, "Debes seleccionar una canción", "Advertencia", 1);
        } else {
            m = (DefaultTableModel) tablaCancionesDisponibles.getModel();
            //Clave = tablaCancionesDisponibles.getValueAt(filaseleccionada, 0).toString();
            //Cancion = tablaCancionesDisponibles.getValueAt(filaseleccionada, 1).toString();
            //Cantante = tablaCancionesDisponibles.getValueAt(filaseleccionada, 2).toString();
            //Genero = tablaCancionesDisponibles.getValueAt(filaseleccionada, 3).toString();
            //Portada = tablaCancionesDisponibles.getValueAt(filaseleccionada, 4).toString();
            //Album = tablaCancionesDisponibles.getValueAt(filaseleccionada, 5).toString();
            //Archivo = tablaCancionesDisponibles.getValueAt(filaseleccionada, 6).toString();
            Precio = tablaCancionesDisponibles.getValueAt(filaseleccionada, 7).toString();
            m = (DefaultTableModel) tablaCancionesCompradas.getModel();
            String filaelemento[] = {Precio};
            m.addRow(filaelemento);
        }
    } catch (HeadlessException | NumberFormatException e) {
        JOptionPane.showMessageDialog(null, "Verificar seleccion de canción", "Error", 1);
        /*if (tablaCanciones.getSelectedRow() == filaseleccionada) {
            JOptionPane.showMessageDialog(this, "Ya esta en tu lista de compras");
        }*/
    */double sumatoriaCargo = 0.0;
    int totalRowc = tablaCancionesCompradas.getRowCount();
    totalRowc -= 1;
    for (int i = 0; i <= (totalRowc); i++) {
        double suma =
Double.parseDouble(String.valueOf(tablaCancionesCompradas.getValueAt(i, 7)));
        sumatoriaCargo += suma;
    }
}

```

```

        txtDinero.setText("" + sumatoriaCargo);
    }*/
}

```

Boton borrar cancion

```

private void btnBorrarActionPerformed(java.awt.event.ActionEvent evt) {
    if (noClave == -1) {
        JOptionPane.showMessageDialog(this, "Debe tener un registro activo");
        return;
    }
    int boton = JOptionPane.showConfirmDialog(this, "Desea borrar este elemento?",
        "Advertencia", JOptionPane.YES_NO_OPTION);
    if (boton == JOptionPane.YES_OPTION) {
        Iterator<Cancion> iteratorX = listaCancionesCompradas.iterator();
        while (iteratorX.hasNext()) {
            if (iteratorX.next().getClave() == noClave) {
                iteratorX.remove();
                JOptionPane.showMessageDialog(this, "Eliminado");
                archivoCompras.EscribirArchivo(listaCancionesCompradas);
                noClave = -1;
                llenarCancionesCompradas();
            }
        }
    }
}

```

Ver pagina web

```

private void btnVerPaginaActionPerformed(java.awt.event.ActionEvent evt) {
    PaginaWebCompras pagweb = new PaginaWebCompras("CancionesCompradas.html");
    pagweb.EscribirArchivo(listaCancionesCompradas);
    //abrir el archivo en el programador
    try {
        File pagina = new File("CancionesCompradas.html");
        Desktop.getDesktop().open(pagina);
    } catch (IOException ex) {
        System.out.println(ex.getMessage());
    }
}

private void tablaCancionesCompradasKeyReleased(java.awt.event.KeyEvent evt) {
    if (tablaCancionesCompradas.getSelectedRows().length > 0) {
        int fila = tablaCancionesCompradas.getSelectedRow();
        noClave = Integer.parseInt(String.valueOf(
            tablaCancionesCompradas.getValueAt(fila, 0)));
    }
}

private void tablaCancionesCompradasMouseClicked(java.awt.event.MouseEvent evt) {
    int fila = tablaCancionesCompradas.getSelectedRow();
    noClave = Integer.parseInt(String.valueOf(
        tablaCancionesCompradas.getValueAt(fila, 0)));
}

private void tablaCancionesDisponiblesMouseClicked(java.awt.event.MouseEvent evt) {
    int fila = tablaCancionesDisponibles.getSelectedRow();
    noClave = Integer.parseInt(String.valueOf(
        tablaCancionesDisponibles.getValueAt(fila, 0)));
}

```

FrmEditar.java

```

package proyectotercerparcial;

```

```

import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
/**
 *
 * @author YAEL
 */
public class FrmEditor extends javax.swing.JFrame {
    int Clave;
    Archivo archivo = new Archivo("Canciones.txt");
    ArrayList<Cancion> listaCanciones = new ArrayList<>();
    public FrmEditor(Cancion aux) {
        initComponents();
        setLocationRelativeTo(null);
        this.setIconImage(new ImageIcon(getClass().getResource("/Iconos/101-headset.png")).getImage());
        txtNombreCancion.setText(aux.getNombreCancion());
        txtNombreCantante.setText(aux.getNombreCantante());
        txtGenero.setText(aux.getGenero());
        txtPortada.setText(aux.getPortada());
        txtAlbum.setText(aux.getAlbum());
        txtArchivo.setText(aux.getArchivo());
        txtPrecio.setText("" + aux.getPrecio());
        Clave = aux.getClave();
        listaCanciones = archivo.leerArchivo();
    }
    private void cerrar() {
        FrmAlbumAdministrador frmC = new FrmAlbumAdministrador();
        frmC.setVisible(true);
        this.dispose();
    }
}

Boton cancelar
private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    cerrar();
}

Boton guardar
private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    String NombreCancion = txtNombreCancion.getText();
    String NombreCantante = txtNombreCantante.getText();
    String Genero = txtGenero.getText();
    String Portada = txtPortada.getText();
    String Album = txtAlbum.getText();
    String Archivo = txtArchivo.getText();
    double Precio;
    try {
        Precio = Double.parseDouble(txtPrecio.getText());
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Debe escribir un numero");
        return;
    }
    for (Cancion aux : listaCanciones) {
        if (aux.getClave() == Clave) {
            aux.setNombreCancion(NombreCancion);
            aux.setNombreCantante(NombreCantante);
            aux.setGenero(Genero);
            aux.setPortada(Portada);
            aux.setAlbum(Album);
            aux.setArchivo(Archivo);
            aux.setPrecio(Precio);
            archivo.EscribirArchivo(listaCanciones);
            JOptionPane.showMessageDialog(this, "Registro guardado");
            cerrar();
        }
    }
}

```

```

    }
}

Cerrar ventana
private void formWindowClosed(java.awt.event.WindowEvent evt) {
    cerrar();
}

```

FrmLoginAdministrador.java

```

package proyectotercerparcial;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
/**
 *
 * @author YAEL
 */
public class FrmLoginAdministrador extends javax.swing.JFrame {
    public FrmLoginAdministrador() {
        initComponents();
        setLocationRelativeTo(null);
        this.setIconImage(new ImageIcon(getClass().getResource("/Iconos/101-handshake_15x15.png")).getImage());
    }

Boton ingresar
private void btnIngresarActionPerformed(java.awt.event.ActionEvent evt) {
    String usu, pass;
    usu = this.txtUsuario.getText();
    pass = this.txtPassword.getText();
    if ((usu.equals("Leonardo")) && (pass.equals("2519"))) {
        javax.swing.JOptionPane.showMessageDialog(null, "Bienvenido " + usu, "Mensaje", 1);
        {
            FrmAlbumAdministrador P1 = new FrmAlbumAdministrador();
            P1.setVisible(true);
            dispose();
        }
    } else {
        javax.swing.JOptionPane.showMessageDialog(null, "Error: acceso no permitido", "mensaje", 0);
    }
}

Boton salir
private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {
    FrmPanelTipoUsuario P1 = new FrmPanelTipoUsuario();
    P1.setVisible(true);
    dispose();
}

private void txtUsuarioActionPerformed(java.awt.event.ActionEvent evt) {
    String Usuario, Contraseña;
    Usuario = this.txtUsuario.getText();
    Contraseña = this.txtPassword.getText();
    if ((Usuario.equals("Leonardo")) && (Contraseña.equals("2519"))) {
        JOptionPane.showMessageDialog(null, "Bienvenido " + Usuario, "Mensaje", 1);
        FrmAlbumAdministrador ventana = new FrmAlbumAdministrador();
        ventana.setVisible(true);
        ventana.setLocationRelativeTo(null);
    } else {
        JOptionPane.showMessageDialog(null, "Error:acceso no permitido", "Mensaje", 0);
    }
}
}

```


FrmLoginUsuario.java

```
package proyectotercerparcial;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
/**
 *
 * @author YAEL
 */
public class FrmLoginUsuario extends javax.swing.JFrame {
    public FrmLoginUsuario() {
        initComponents();
        setLocationRelativeTo(null);
        this.setIconImage(new ImageIcon(getClass().getResource("/Iconos/101-handshake_15x15.png")).getImage());
    }
    Boton salir
    private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {
        FrmPanelTipoUsuario P1 = new FrmPanelTipoUsuario();
        P1.setVisible(true);
        dispose();
    }
    Boton ingresar
    private void btnIngresarActionPerformed(java.awt.event.ActionEvent evt) {
        String usu, pass;
        usu = this.txtUsuario.getText();
        pass = this.txtPassword.getText();
        if ((usu.equals("OSAMA")) && (pass.equals("2519"))) {
            javax.swing.JOptionPane.showMessageDialog(null, "Bienvenido " + usu, "Mensaje", 1);
            {
                FrmAlbumUsuario P1 = new FrmAlbumUsuario();
                P1.setVisible(true);
                dispose();
            }
        } else {
            javax.swing.JOptionPane.showMessageDialog(null, "Error: acceso no permitido", "mensaje", 0);
        }
    }
    private void txtUsuarioActionPerformed(java.awt.event.ActionEvent evt) {
        String Usuario, Contraseña;
        Usuario = this.txtUsuario.getText();
        Contraseña = this.txtPassword.getText();
        if ((Usuario.equals("Leonardo")) && (Contraseña.equals("2519"))) {
            JOptionPane.showMessageDialog(null, "Bienvenido " + Usuario, "Mensaje", 1);
            FrmAlbumAdministrador ventana = new FrmAlbumAdministrador();
            ventana.setVisible(true);
            ventana.setLocationRelativeTo(null);
        } else {
            JOptionPane.showMessageDialog(null, "Error:acceso no permitido", "Mensaje", 0);
        }
    }
}
```

FrmPanelTipoUsuario.java

```
package proyectotercerparcial;
import javax.swing.ImageIcon;
/**
 *
 * @author YAEL
 */
public class FrmPanelTipoUsuario extends javax.swing.JFrame {
    public FrmPanelTipoUsuario() {
        initComponents();
        setLocationRelativeTo(null);
        this.setIconImage(new ImageIcon(getClass().getResource("/Imagenes/INICIAR2_10x10.png")).getImage());
    }
    Boton Ingresar administrador
    private void btnAdministradorActionPerformed(java.awt.event.ActionEvent evt) {
        FrmLoginAdministrador P1 = new FrmLoginAdministrador();
        P1.setVisible(true);
        dispose();
    }
    Boton Ingresar Cliente
    private void btnClienteActionPerformed(java.awt.event.ActionEvent evt) {
        FrmLoginUsuario P1 = new FrmLoginUsuario();
        P1.setVisible(true);
        dispose();
    }
    Boton cancelar
    private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
        dispose();
    }
}
```

FrmEditarUsuario.java

```
package proyectotercerparcial;
import java.util.ArrayList;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
/**
 *
 * @author YAEL
 */
public class FrmEditarUsuario extends javax.swing.JFrame {
    int Clave;
    Archivo archivoCompras = new Archivo("CancionesCompradas.txt");
    ArrayList<Cancion> listaCancionesCompradas = new ArrayList<>();
    public FrmEditarUsuario(Cancion aux) {
        initComponents();
        setLocationRelativeTo(null);
        this.setIconImage(new ImageIcon(getClass().getResource("/Iconos/101-headset.png")).getImage());
        txtNombreCancion.setText(aux.getNombreCancion());
        txtNombreCantante.setText(aux.getNombreCantante());
        txtGenero.setText(aux.getGenero());
        txtPortada.setText(aux.getPortada());
        txtAlbum.setText(aux.getAlbum());
        txtArchivo.setText(aux.getArchivo());
    }
}
```

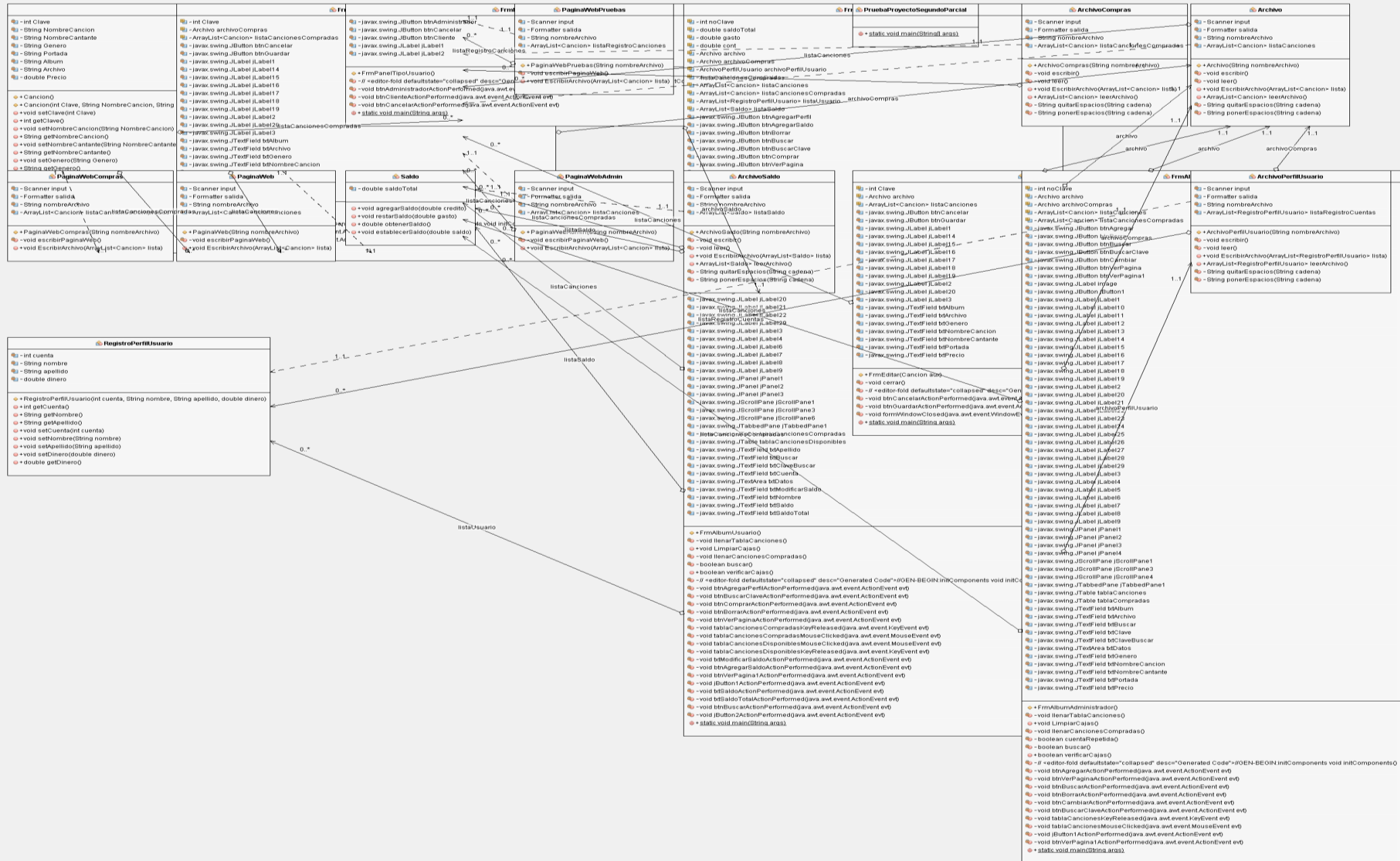
```
        txtPrecio.setText("" + aux.getPrecio());
        Clave = aux.getClave();
        listaCancionesCompradas = archivoCompras.leerArchivo();
    }
    private void cerrar() {
        FrmAlbumUsuario frmC = new FrmAlbumUsuario();
        frmC.setVisible(true);
        this.dispose();
    }
}
```

Boton cancelar






```
private void btnCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    cerrar();
}
}
```







Boton guardar

```
private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    String NombreCancion = txtNombreCancion.getText();
    String NombreCantante = txtNombreCantante.getText();
    String Genero = txtGenero.getText();
    String Portada = txtPortada.getText();
    String Album = txtAlbum.getText();
    String Archivo = txtArchivo.getText();
    double Precio;
    try {
        Precio = Double.parseDouble(txtPrecio.getText());
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Debe escribir un numero");
        return;
    }
    for (Cancion aux : listaCancionesCompradas) {
        if (aux.getClave() == Clave) {
            aux.setNombreCancion(NombreCancion);
            aux.setNombreCantante(NombreCantante);
            aux.setGenero(Genero);
            aux.setPortada(Portada);
            aux.setAlbum(Album);
            aux.setArchivo(Archivo);
            aux.setPrecio(Precio);
            archivoCompras.EscribirArchivo(listaCancionesCompradas);
            JOptionPane.showMessageDialog(this, "Registro guardado");
            cerrar();
        }
    }
}
}
```










FrmPanelTipoUsuario

 - javax.swing.JButton btnAdministrador
 - javax.swing.JButton btnCancelar
 - javax.swing.JButton btnCliente
 - javax.swing.JLabel jLabel1
 - javax.swing.JLabel jLabel2

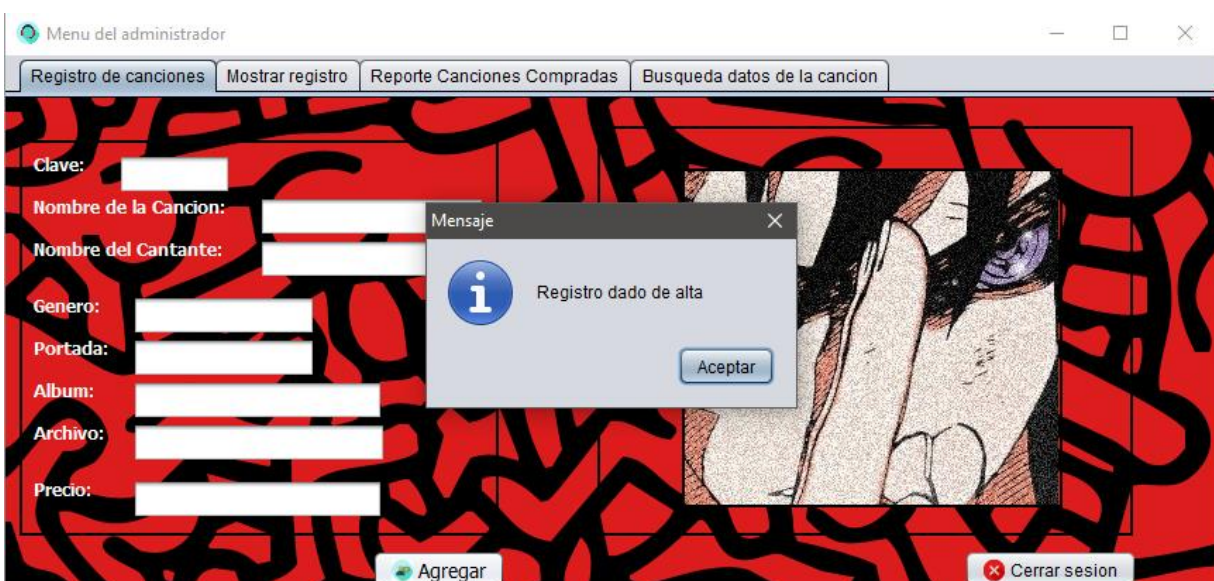
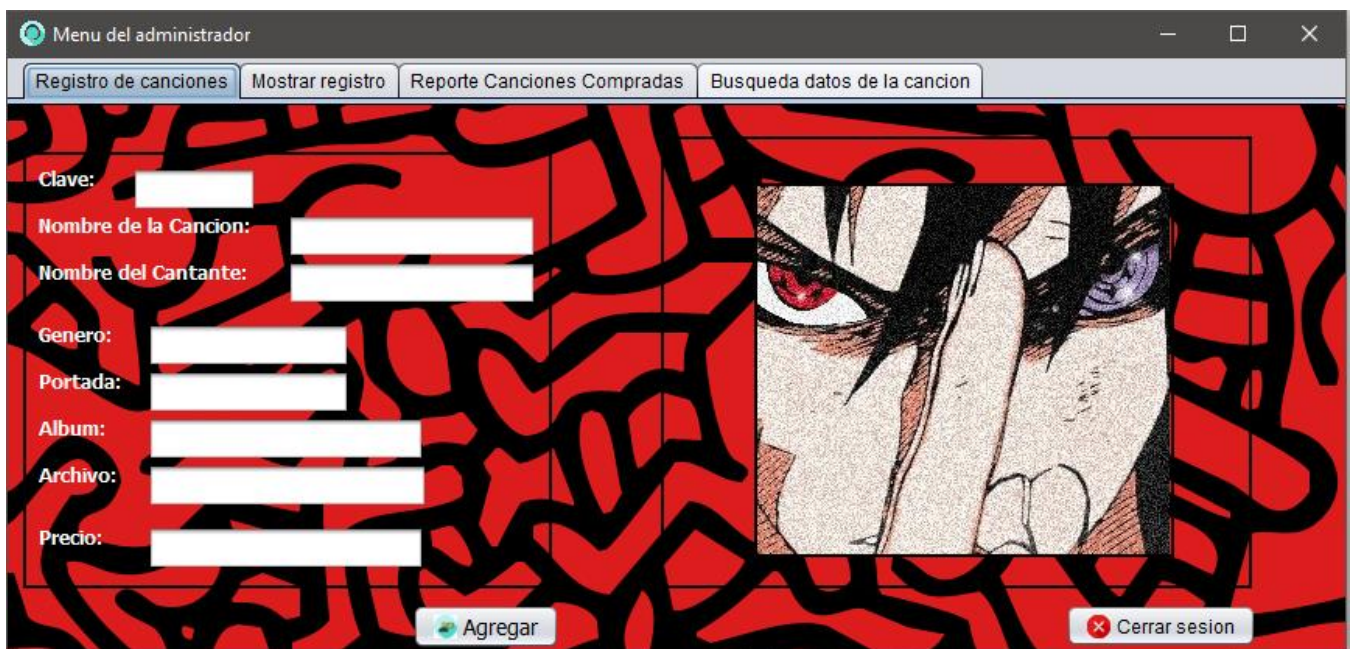
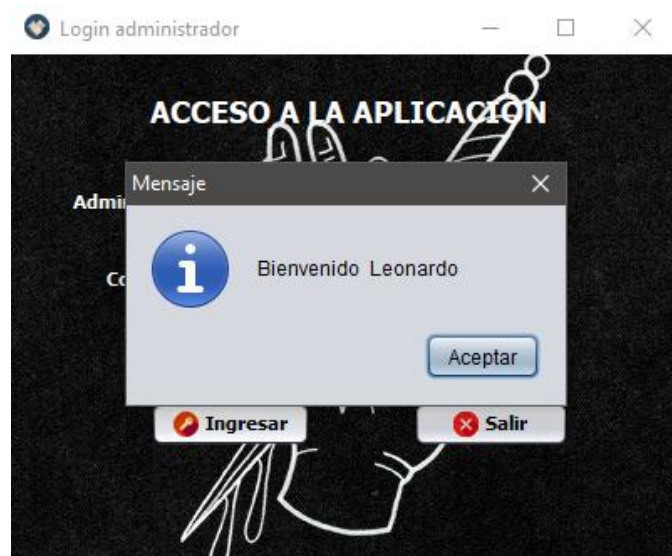
 + FrmPanelTipoUsuario()
 -// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()
 -void btnAdministradorActionPerformed(java.awt.event.ActionEvent evt)
 -void btnClienteActionPerformed(java.awt.event.ActionEvent evt)
 -void btnCancelarActionPerformed(java.awt.event.ActionEvent evt)
 + static void main(String args)

FrmEditor

 ~int Clave
 ~Archivo archivo
 ~ArrayList<Cancion> listaCanciones
 - javax.swing.JButton btnCancelar
 - javax.swing.JButton btnGuardar
 - javax.swing.JLabel jLabel1
 - javax.swing.JLabel jLabel14
 - javax.swing.JLabel jLabel15
 - javax.swing.JLabel jLabel16
 - javax.swing.JLabel jLabel17
 - javax.swing.JLabel jLabel18
 - javax.swing.JLabel jLabel19
 - javax.swing.JLabel jLabel2
 - javax.swing.JLabel jLabel20
 - javax.swing.JLabel jLabel3
 - javax.swing.JTextField txtAlbum
 - javax.swing.JTextField txtArchivo
 - javax.swing.JTextField txtGenero
 - javax.swing.JTextField txtNombreCancion
 - javax.swing.JTextField txtNombreCantante
 - javax.swing.JTextField txtPortada
 - javax.swing.JTextField txtPrecio

 + FrmEditor(Cancion aux)
 -void cerrar()
 -// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()
 -void btnCancelarActionPerformed(java.awt.event.ActionEvent evt)
 -void btnGuardarActionPerformed(java.awt.event.ActionEvent evt)
 -void formWindowClosed(java.awt.event.WindowEvent evt)
 + static void main(String args)

Salida



Menu del administrador

Registro de canciones | **Mostrar registro** | Reporte Canciones Compradas | Busqueda datos de la cancion

Canciones dadas de alta

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|---------------|---------------|---------|----------------|----------------|---------------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | August 26th | Lonely | 12.5 |
| 3 | Slide | Calvin Harris | Funk | FunkWavBoun... | FunkWavBou... | Slide | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 6 | Advance | Juicy J | Hip-Hop | Shut | Shutdafukup | Advance | 12.5 |
| 7 | Cold | Post Malone | Hip-Hop | Stoney | Stoney | Cold | 12.5 |
| 8 | TheHumanCe... | Ski Mask | Hip-Hop | VERYRAREL... | VERYRAREL... | TheHumanCe... | 12.5 |
| 9 | LTE | SUICIDEBOYS | Hip-Hop | YUNGDEATH... | YUNGDEATH... | LTE | 12.5 |
| 10 | Befour | ZAYN | Pop | ZAYN | Mind Of Mine | Befour | 12.5 |

Borrar Editar Ver pagina web

Dame clave a buscar: Buscar clave

Menu del administrador

Registro de canciones | **Mostrar registro** | Reporte Canciones Compradas | Busqueda datos de la cancion

Canciones dadas de alta

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-------------|---------------|---------|---------|-------|-------------|--------|
| 12 | Yno | Rae Sremm | | | | hLife | 15.5 |
| 13 | UpLikeTrump | Rae Sremm | | | | hLife | 13.5 |
| 14 | SuperSaiyan | Lil Uzi Vert | | | | Rage | 12.55 |
| 15 | Stunna | Pouya | | | | Stunna | 12.5 |
| 16 | OceanEyes | Billie Eilish | | | | smile at... | 15.5 |
| 17 | IcedOut | Lil Pump | | | | np | 15.5 |
| 18 | Copycat | Billie Eilish | | | | smile at... | 12.5 |
| 19 | ComeGetHer | Rae Sremm | | | | hLife | 13.5 |
| 20 | ChokeHold | Juicy J | | | | Shutdafukup | 15.5 |
| 21 | AS | AS | Hip-Hop | Shut | AS | AS | 12.0 |

Borrar Editar Ver pagina web

Dame clave a buscar: Buscar clave

Advertencia

?

Desea borrar este elemento?

Menu del administrador

Registro de canciones | **Mostrar registro** | Reporte Canciones Compradas | Busqueda datos de la cancion

Canciones dadas de alta

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-------------|---------------|---------|---------|-------|-------------|--------|
| 12 | Yno | Rae Sremm | | | | hLife | 15.5 |
| 13 | UpLikeTrump | Rae Sremm | | | | hLife | 13.5 |
| 14 | SuperSaiyan | Lil Uzi Vert | | | | Rage | 12.55 |
| 15 | Stunna | Pouya | | | | Stunna | 12.5 |
| 16 | OceanEyes | Billie Eilish | | | | smile at... | 15.5 |
| 17 | IcedOut | Lil Pump | | | | np | 15.5 |
| 18 | Copycat | Billie Eilish | | | | smile at... | 12.5 |
| 19 | ComeGetHer | Rae Sremm | | | | hLife | 13.5 |
| 20 | ChokeHold | Juicy J | | | | Shutdafukup | 15.5 |
| 21 | AS | AS | Hip-Hop | Shut | AS | AS | 12.0 |

Borrar Editar Ver pagina web

Dame clave a buscar: Buscar clave

Mensaje

i

Eliminado

Editar cancion

Realizar Cambios

Nombre de la Cancion: ChokeHold

Nombre del Cantante: Juicy J



Genero: Hip-Hop

Portada: Shut

Album: Shutdafukup

Archivo: ChokeHold

Precio: 15.5

 **Guardar**  **Cancelar**

Editar cancion


Realizar Cambios

Nombre de la Cancion: ChokeHold

Nombre del C

Archivo: ChokeHold

Precio: 15.5

 **Guardar**  **Cancelar**

Mensaje










Registro guardado

Aceptar

Mostrar aplicaciones

SLEEP WHEN

Canciones registradas

| Clave | Nombre de la canción | Nombre del cantante | Genero | Portada | Album | Archivo | Precio |
|-------|----------------------|---------------------|---------|---|-----------------------|---------------|--------|
| 1 | Ayala | XXXTENTACION | Hip-Hop |  | 17 | ▶ 0:00 / 1:40 | 12.50 |
| 2 | Loney | Post Malone | Hip-Hop |  | August 26th | ▶ 0:00 / 4:14 | 12.50 |
| 3 | Slide | Calvin Harris | Funk |  | FunkWavBounces Vol. 1 | ▶ 0:00 / 3:50 | 12.50 |
| 4 | Veins | Lil Peep | Hip-Hop |  | LiPeepPartOne | ▶ 0:00 / 3:30 | 12.50 |
| 5 | PattyCake | XXXTENTACION | Hip-Hop |  | Deleted Files | ▶ 0:00 / 2:12 | 12.50 |
| 6 | Adva | Post Malone | Hip-Hop |  | Shutdafukup | ▶ 0:00 / 4:36 | 12.50 |
| 7 | Cold | Post Malone | Hip-Hop |  | Stoney | ▶ 0:00 / 4:28 | 12.50 |

Activar Windows
Ve a Configuración para activar Windows.

Menu del administrador

Registro de canciones | **Mostrar registro** | Reporte Canciones Compradas | Busqueda datos de la cancion

Canciones dadas de alta

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|---------|--------------|---------|---------|-------|---------|--------|
| 1 | Ayala | XXXTENTACION | Hip-Hop | 17 | 17 | Ayala | 12.5 |

[Borrar](#) [Editar](#) [Ver pagina web](#)

Dame clave a buscar: [Buscar clave](#)

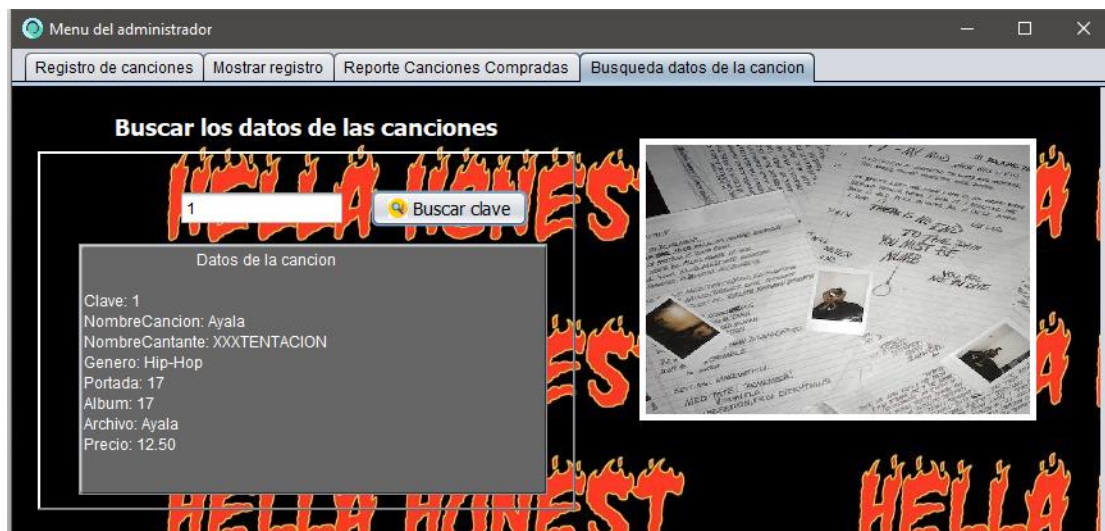
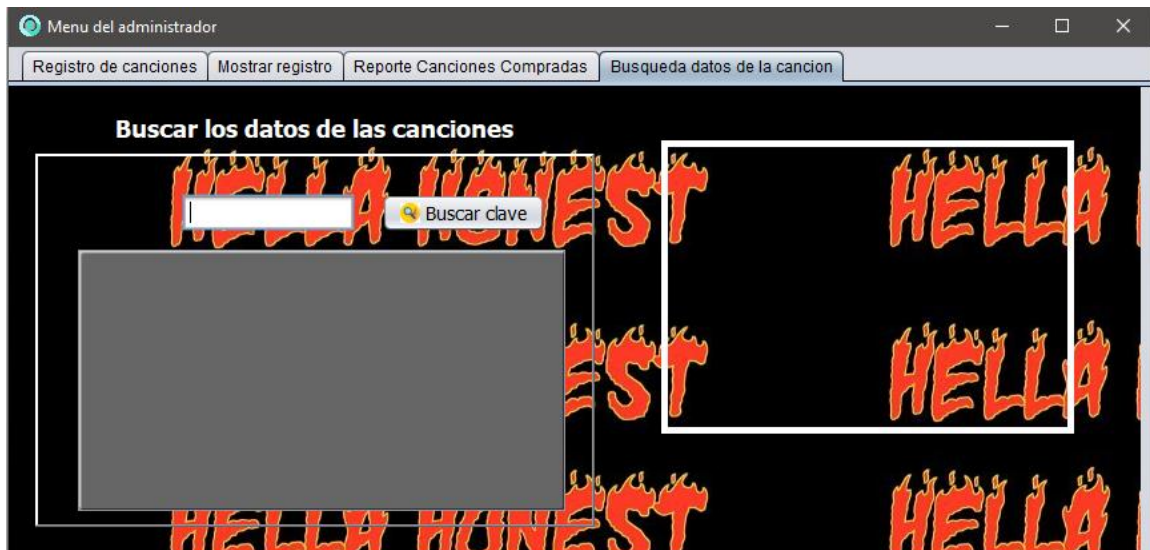
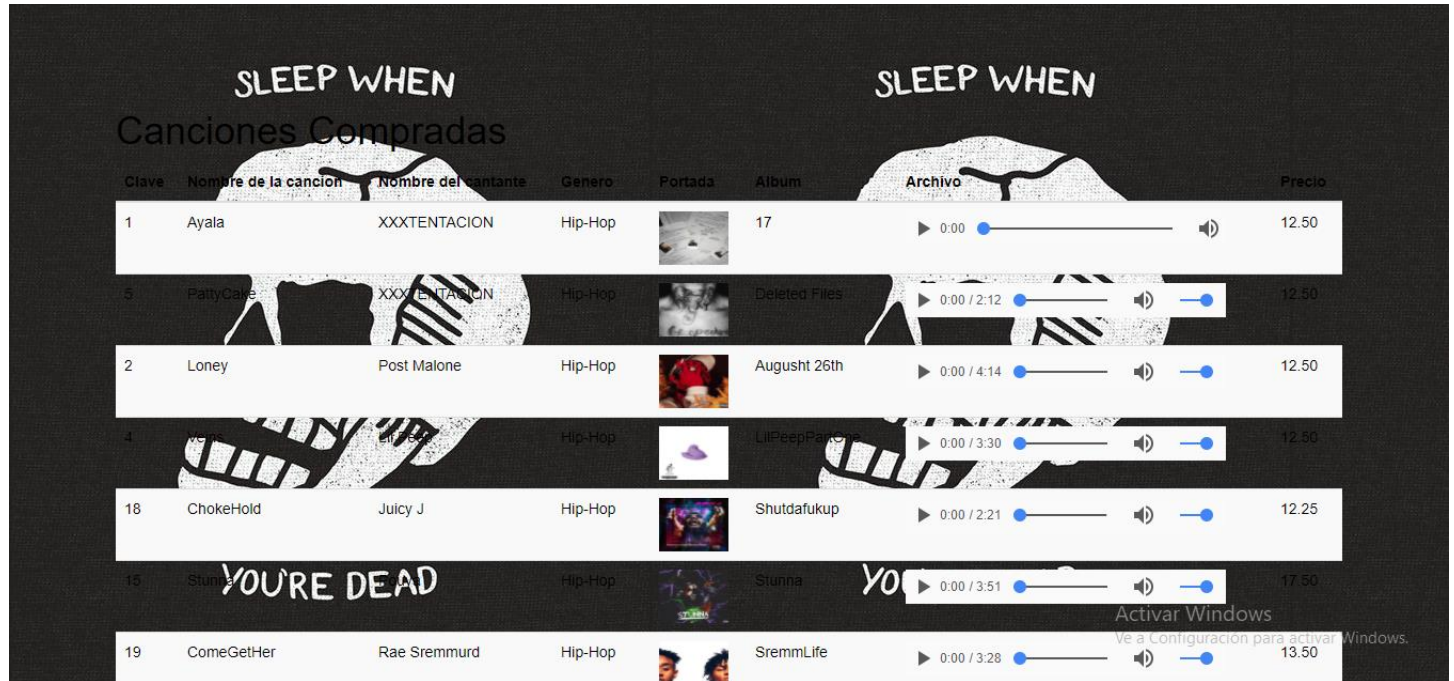
Menu del administrador

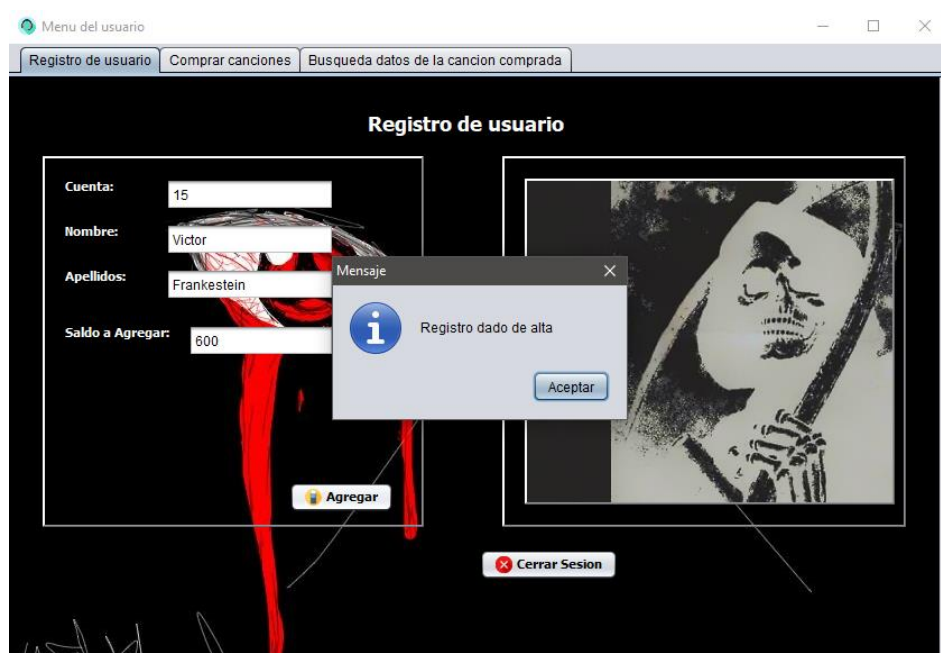
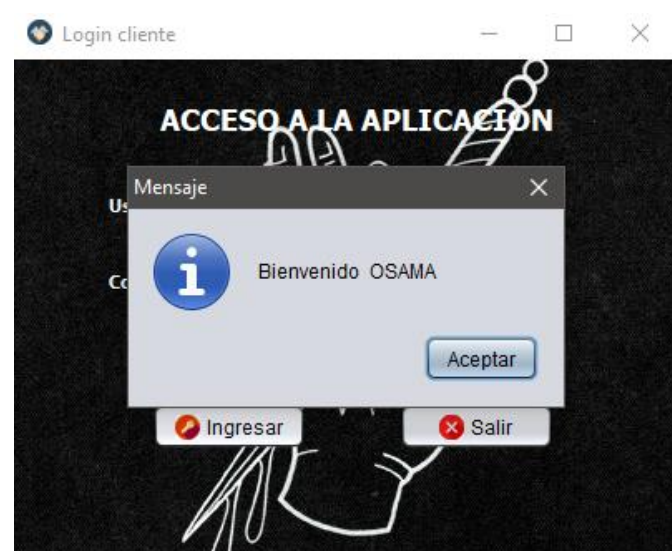
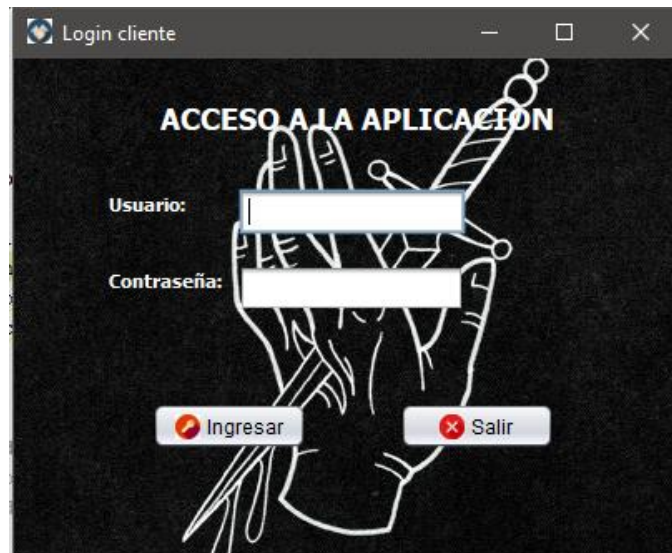
Registro de canciones | **Mostrar registro** | Reporte Canciones Compradas | Busqueda datos de la cancion

Canciones compradas por usuarios

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|------------|---------------|-------------|---------------|--------------------|------------|--------|
| 1 | Ayala | XXXTENTACION | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 5 | PattyCake | XXXTENTACION | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | August 26th | Loney | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LiPeepPartOne | LiPeepPartOne | Veins | 12.5 |
| 18 | ChokeHold | Juicy J | Hip-Hop | Shut | Shutdafukup | ChokeHold | 12.25 |
| 15 | Stunna | Pouya | Hip-Hop | Stunna | Stunna | Stunna | 17.5 |
| 19 | ComeGetHer | Rae Sremmurd | Hip-Hop | Download | SremmLife | ComeGetHer | 13.5 |
| 16 | OceanEyes | Billie Eilish | Alternative | BillieEilish | Don't Smile at ... | OceanEyes | 15.5 |
| 20 | ChokeHold | Juicy J | Hip-Hop | Shut | Shutdafukup | ChokeHold | 15.5 |
| 17 | IcedOut | Lil Pump | Hip-Hop | Cover | Lil Pump | IcedOut | 15.5 |

[Ver pagina web de compras realizadas por usuarios](#)





Registro de usuario

Comprar canciones

Busqueda datos de la cancion comprada

Saldo total: 800.0

Canciones disponibles para comprar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | August 26th | Lonely | 12.5 |
| 3 | Slide | Calvin Harris | Funk | FunkWavBoun... | FunkWavBou... | Slide | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Deleted Files | Deleted Files | PattyCake | 12.5 |
| 6 | Advance | Juicy J | Hip-Hop | Shutdafukup | Shutdafukup | Advance | 12.5 |

Modificar Saldo:



Agregado correctamente

Aceptar

Ver canciones

Comprar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | August 26th | Lonely | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 18 | ChokeHold | Juicy J | Hip-Hop | Shut | Shutdafukup | ChokeHold | 12.25 |
| 15 | Stunna | Pouya | Hip-Hop | Stunna | Stunna | Stunna | 17.5 |

Borrar compra

Ver canciones compradas

Dame clave a buscar:

Buscar clave

SLEEP WHEN
Catalogos de canciones

| Clave | Nombre de la cancion | Nombre del cantante | Genero | Portada | Album | Archivo | Precio |
|-------|----------------------|---------------------|---------|---------|-----------------------|---------|--------|
| 1 | Ayala | XXXTENTACION | Hip-Hop | | 17 | ▶ 0:00 | 12.50 |
| 2 | Loney | Post Malone | Hip-Hop | | August 26th | ▶ 0:00 | 12.50 |
| 3 | Slide | Calvin Harris | Funk | | FunkWavBounces Vol. 1 | ▶ 0:00 | 12.50 |
| 4 | Veins | Lil Peep | Hip-Hop | | LilPeepPartOne | ▶ 0:00 | 12.50 |
| 5 | PattyCake | XXXTENTACION | Hip-Hop | | Deleted Files | ▶ 0:00 | 12.50 |
| 6 | Advance | Juicy J | Hip-Hop | | Shutdafukup | ▶ 0:00 | 12.50 |
| 7 | Cold | Post Malone | Hip-Hop | | Stoney | ▶ 0:00 | 12.50 |

Activar Windows

Ve a Configuración para activar Windows.

Menu del usuario

Registro de usuario Comprar canciones Busqueda datos de la cancion comprada

Saldo total: 800.0

Canciones disponibles para comprar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 3 | Slide | Calvin Harris | Funk | FunkWavBoun... | FunkWavBou... | Slide | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 6 | Advance | Juicy J | Hip-Hop | Shut | Shutdafukup | Advance | 12.5 |

Modificar Saldo:

Mensaje

Comprada

Aceptar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 18 | ChokeHold | Juicy J | Hip-Hop | Shut | Shutdafukup | ChokeHold | 12.25 |
| 15 | Stunna | Pouya | Hip-Hop | Stunna | Stunna | Stunna | 17.5 |

Borrar compra Ver canciones compradas

Dame clave a buscar:

Buscar clave

Menu del usuario

Registro de usuario Comprar canciones Busqueda datos de la cancion comprada

Saldo total: 800.0

Canciones disponibles para comprar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 3 | Slide | Calvin Harris | Funk | FunkWavBoun... | FunkWavBou... | Slide | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 6 | Advance | Juicy J | Hip-Hop | Shut | Shutdafukup | Advance | 12.5 |

Modificar Saldo:

Mensaje

Ya comprado

Aceptar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 18 | ChokeHold | Juicy J | Hip-Hop | Shut | Shutdafukup | ChokeHold | 12.25 |
| 15 | Stunna | Pouya | Hip-Hop | Stunna | Stunna | Stunna | 17.5 |

Borrar compra Ver canciones compradas

Dame clave a buscar:

Buscar clave

Saldo total: 800.0

Canciones disponibles para comprar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 3 | Slide | Calvin Harris | Funk | FunkWavBoun... | FunkWavBou... | Slide | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 6 | Advance | Juicy J | Hip-Hop | Shut | Shutdafukup | Advance | 12.5 |

Modificar Saldo:



Desea borrar este elemento?

Sí

No

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 18 | ChokeHold | Juicy J | Hip-Hop | Shut | Shutdafukup | ChokeHold | 12.25 |
| 15 | Stunna | Pouya | Hip-Hop | Stunna | Stunna | Stunna | 17.5 |

Borrar compra

Ver canciones compradas

Dame clave a buscar:

Buscar clave

https://twitter.com

SLEEP WHEN
Canciones Compradas

| Clave | Nombre de la cancion | Nombre del cantante | Genero | Portada | Album | Archivo | Precio |
|-------|----------------------|---------------------|---------|---------|----------------|---------|--------|
| 1 | Ayala | XXXTENTACION | Hip-Hop | | 17 | | 12.50 |
| 5 | PattyCake | XXXTENTACION | Hip-Hop | | Deleted Files | | 12.50 |
| 2 | Loney | Post Malone | Hip-Hop | | Augusht 26th | | 12.50 |
| 4 | Veins | Lil Peep | Hip-Hop | | LilPeepPartOne | | 12.50 |
| 18 | ChokeHold | Juicy J | Hip-Hop | | Shutdafukup | | 12.25 |
| 15 | Stunna | Pouya | Hip-Hop | | Stunna | | 17.50 |
| 19 | ComeGetHer | Rae Sremmurd | Hip-Hop | | SremmLife | | 13.50 |

Activar Windows

Ve a Configuración para activar Windows.

Menu del usuario

Registro de usuarioComprar cancionesBusqueda datos de la cancion comprada

Saldo total: 800.0

Canciones disponibles para comprar

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 3 | Slide | Calvin Harris | Funk | FunkWavBoun... | FunkWavBou... | Slide | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 6 | Advance | Juicy J | Hip-Hop | Shut | Shutdafukup | Advance | 12.5 |

Modificar Saldo:

Agregar Saldo

Ver catalogo de canciones

Comprar

Canciones compradas FIRE

| Clave | Cancion | Cantante | Genero | Portada | Album | Archivo | Precio |
|-------|-----------|---------------|---------|----------------|----------------|-----------|--------|
| 1 | Ayala | XXXTENTACI... | Hip-Hop | 17 | 17 | Ayala | 12.5 |
| 5 | PattyCake | XXXTENTACI... | Hip-Hop | Pistol | Deleted Files | PattyCake | 12.5 |
| 2 | Loney | Post Malone | Hip-Hop | August26 | Augusht 26th | Lonely | 12.5 |
| 4 | Veins | Lil Peep | Hip-Hop | LilPeepPartOne | LilPeepPartOne | Veins | 12.5 |
| 18 | ChokeHold | Juicy J | Hip-Hop | Shut | Shutdafukup | ChokeHold | 12.25 |
| 15 | Stunna | Pouya | Hip-Hop | Stunna | Stunna | Stunna | 17.5 |

Borrar compra

Editar genero

Ver canciones compradas

Dame clave a buscar:

Buscar clave

Editar genero

Realizar Cambio

Nombre de la Cancion:

Loney

Nombre del Cantante:

Post Malone

Genero:

Hip-Hop

Portada:

August26

Album:

Augusht 26th

Archivo:

Lonely

Precio:

12.5

Guardar

Cancelar



Asi queda la capeta del proyecto

- Carpeta de canciones de prueba y completas.
- Documentos de texto con los datos guardados.
- Paginas web.
- Bootstrap.
- Imágenes.

