

Universidad Rafael Landívar
Facultad de ingeniería
Ingeniería en sistemas
Programación avanzada
Catedrático: Daniel Mejía

PROYECTO PRÁCTICO NO. 1

Megan Naomi Morales Betancourt- 1221120

Guatemala, 24 de octubre del 2021

INTRODUCCIÓN

La documentación técnica del proyecto No.1 de la clase de programación avanzada titulado “Party Mix”, tiene como objetivo dar a conocer al usuario el uso correcto del programa. Así como un análisis de las entradas, salidas, procesos, restricciones, librerías y códigos utilizados.

El programa de party mix, tiene como funcionalidad leer los datos de una playlist por medio de un archivo csv. Con estos datos se genera una simulación con una playlist y una fila de reproducción. El usuario tiene la posibilidad de ordenar de forma ascendente y descendientemente, por nombre de canción y por nombre del artista. Puede reproducir la playlist y al acabar las canciones, trasladar las canciones en la cola de reproducción. También tiene la opción de iniciar una fila de reproducción, agregar y eliminar canciones de forma manual. Puede reproducir la fila de reproducción de forma aleatoria y secuencial y avanzar en estas mismas. Para finalizar, el usuario cuenta con la opción de exportar la playlist actual.

ANÁLISIS

a) Entradas

Archivo csv separado por comas

b) Salidas

Exportar la playlist actual en un archivo csv

c) Procesos

Métodos más importantes:

```
private: void llenarListBox() { //Método para llenar el listBox de la playlist y
actualiza también
    int contador = 0;
    while (playlist->get(contador) != nullptr) { //Mientras playlist sea
diferente de vacío
        string Cancion;
        string Artista;
        Cancion = playlist->get(contador)->getName();
        Artista = playlist->get(contador)->getArtist();
        String^ cancion = gcnew String(Cancion.c_str()); //Conversión de
string a String^
        String^ artista = gcnew String(Artista.c_str());
        listPlaylist->Items->Add(contador + " - " + cancion + " - " +
artista); //Imprimir los datos en el listBox
        contador++; //Se suma para ir recorriendo la playlist
    }
}

void sincronizarPlaylistyCola() { //Sincronizar mi playList con la cola
    if (!colaReproduccion->isEmpty()) {
        for (int i = 0; i < colaReproduccion->getSize(); i++)
        {
            playlist->InsertAtStart(colaReproduccion-
>get(i)); //Para insertar modo cola
        }
        if (estados == 0) { //si el estado es sin ordenamiento
/ no hacer nada porque no hay que ordenar nada.
            listPlaylist->Items->Clear();
            llenarListBox();
        }
        else if (estados == 1) { // si el estado es 1,
sincronizar de forma ascendente.
            ordenarAscendenteNombre();
        }
        else if (estados == 2) {
            ordenarAscendenteArtista();
        }
        else if (estados == 3) {
            ordenarDescendenteNombre();
        }
    }
}
```

```

        else {
            ordenarDescendenteArtista();
        }
    }
    else {
        MessageBox::Show("La fila de reproducción está vacía",
"Error", MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
}

```

```

int aleatorio() { //Función random
    int numeroAleatorio = 0;
    numeroAleatorio = rand() % colaReproduccion->getSize();
    return numeroAleatorio;
}

```

```

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
//Botón de importar
    lbEstadoPlayList->Text = "Ninguno";
    ofdImportar->Filter = "Archivos separados por coma (csv) | *.csv";
    ofdImportar->FileName = "";

    if (ofdImportar->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
        restablecerPlaylist();
        //Se utiliza el objeto File para leer el archivo solo cuando el
        FileName es correcto
        array<String^>^ archivoLineas = File::ReadAllLines(ofdImportar-
>FileName);
        if (archivoLineas->Length > 0) {
            //Llenar list playlist
            for (int i = 0; i < archivoLineas->Length; i++) {
                array<String^>^ columnaArchivo = archivoLineas[i]-
>Split(',');

                int j = 0;
                while (j < columnaArchivo->Length) {
                    array<String^>^ nomCancionArtista =
columnaArchivo[j]->Split('-');

                    if (nomCancionArtista->Length >= 2) {
                        string nameCancion;
                        string nameArtista;
                        MarshalString(nomCancionArtista[0],
nameCancion);

                        MarshalString(nomCancionArtista[1],
nameArtista);

                        if (nameArtista == "") {
                            Canciones* cancion = new
Canciones(nameCancion, "Desconocido");

                            playlist->add(cancion);
                        }
                    }
                }
            }
        }
    }
}

```

```

        else {
            Canciones* cancion = new
Canciones(nameCancion, nameArtista);
            playlist->add(cancion);
        }
    }
    else if (nomCancionArtista->Length == 1) {
        string nameCancion;
        if (nomCancionArtista[0] == "") {}
        else {
            MarshalString(nomCancionArtista[0], nameCancion);
            Canciones* cancion = new
Canciones(nameCancion, "Desconocido");
            playlist->add(cancion);
        }
    }
    j++;
}
llenarListBox();
}
}
else {
    MessageBox::Show("No se seleccionó ningún archivo", "Archivo no
seleccionado", MessageBoxButtons::OK, MessageBoxIcon::Exclamation);
}
}
}

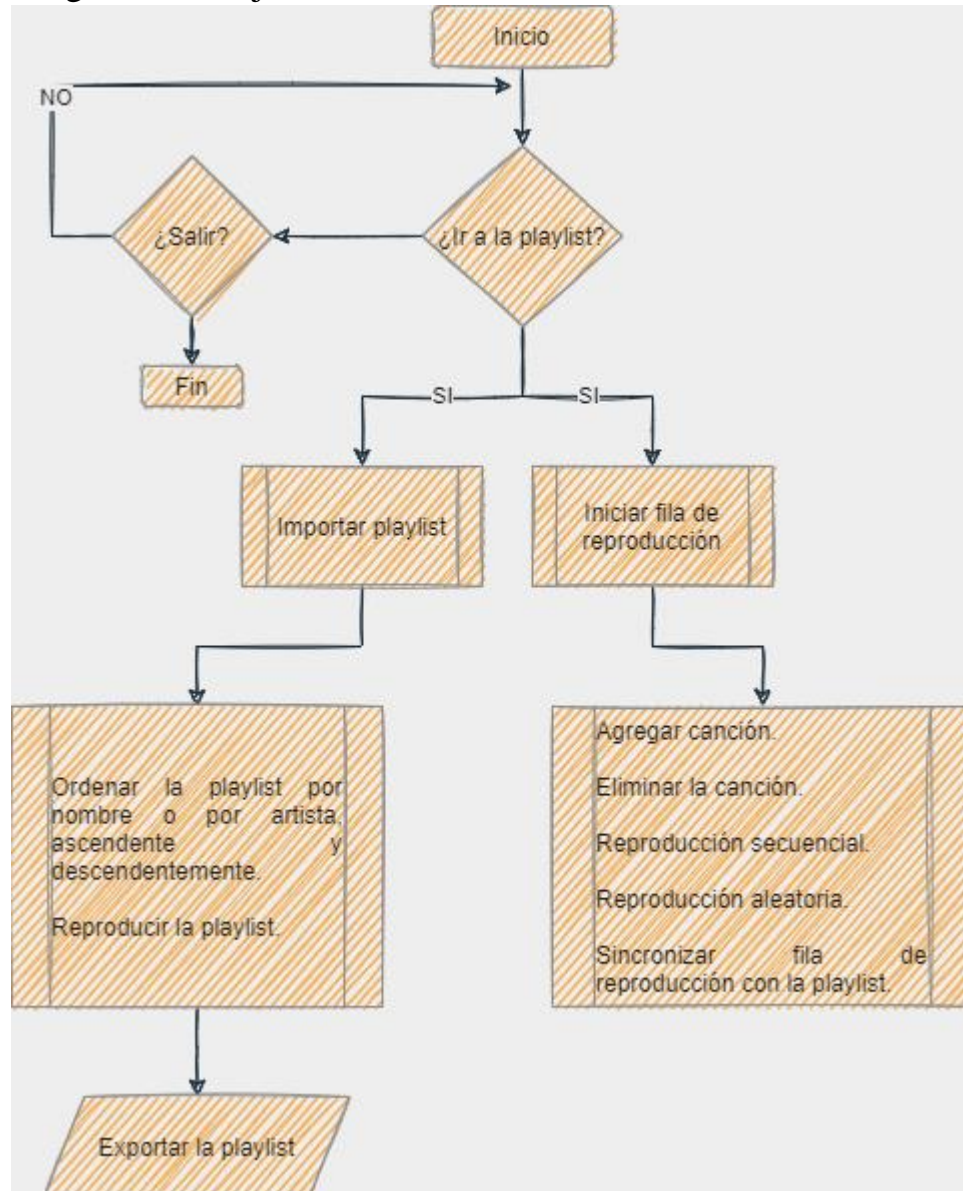
```

d) Restricciones

1. Agregar nombre y canción en la forma manual. Utilizar la primera letra mayúscula.
2. No eliminar la última canción y luego agregar una nueva.

DISEÑO

a) Diagrama de flujo



CONCLUSIONES

1. Para la realización de este proyecto, se necesita investigar acerca de las estructuras de datos lineales. Listas, pilas, colas.
2. El manejo de los errores es importante para dar el usuario un uso cómodo del proyecto.

RECOMENDACIONES

Dedicarle tiempo a la realización y documentación del proyecto, investigar lo más posible para tener varias ideas de la realización, y así tener un plan b por si el plan A no funciona.

Investigar sobre el manejo de punteros y estructuras lineales para un mejor entendimiento.

REFERENCIAS

Las librerías usadas para este proyecto fueron:

1. `#include <string>`
2. `using namespace System::IO;` Es la librería utilizada para poder exportar e importar archivos dentro del programa.

También fue necesario el siguiente código para hacer la conversión de `String^` a `string` :

```
void MarshalString(String^ s, string& os) {  
    using namespace Runtime::InteropServices;  
    const char* chars = (const  
        char*)(Marshal::StringToHGlobalAnsi(s)).ToPointer();  
    os = chars;  
    Marshal::FreeHGlobal(IntPtr((void*)chars));  
}
```

El siguiente código es un ejemplo del código que fue utilizado para hacer la conversión de `string` a `String^`:

```
string Cancion;  
string Artista;  
Cancion = colaReproduccion->get(countReproduccion)->getName();  
Artista = colaReproduccion->get(countReproduccion)->getArtist();  
String^ cancion = gcnew String(Cancion.c_str());  
String^ artista = gcnew String(Artista.c_str());
```

También fue utilizado el siguiente código para base del proyecto:

https://github.com/rdmejia/cpp_sorting_algorithms

ANEXOS

Manual de usuario:

1. Al iniciar el programa, aparecerá una pequeña ventana, esta ventana es el menú.

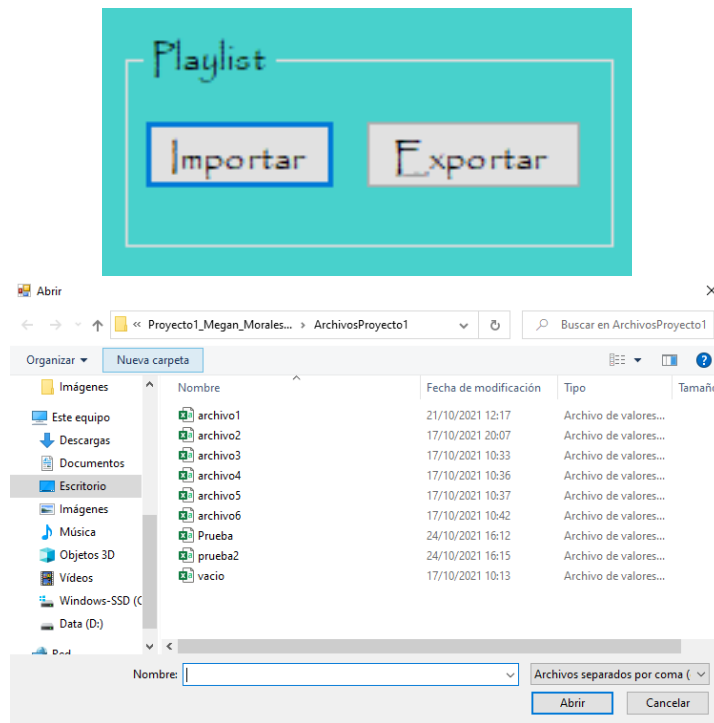


2. Al darle iniciar, se mostrará una ventana con la interfaz de Party mix.



De lo contrario, al darle click en salir, la pestaña cerrará automáticamente.

3. Al presionar el botón de importar, se desplegará una ventana para seleccionar el archivo deseado.



4. Al seleccionar el archivo, su playlist estará cargada en el ListBox.
5. De lado izquierdo aparecerán todas las funciones disponibles para la playlist.



6. Al presionar el botón de reproducir, en la parte superior aparecerá la canción en reproducción y la playlist modificada.

Playlist

Importar Exportar

Canción en reproducción

Amate - Micro TDH

— Mi playlist —

— Canción — Artista —

0 - La Canción - J Balvin
1 - Ojos Color Sol - Calle 13
2 - Playa - Nicky Jam

Funciones de la playlist

Ordenar la playlist por:

Ascendentemente por:

Nombre de la canción

Nombre del artista

Descendentemente por:

Nombre de la canción

Nombre del artista

Estado de ordenamiento:

Ascendentemente/nombre

-- Reproducir --

7. Al no tener más canciones en la playlist, sino se ha iniciado la fila de reproducción, no realizará ninguna acción. De lo contrario, las canciones agregadas a la fila de reproducción se enviarán a la playlist.

Playlist

Importar Exportar

Canción en reproducción

Playa - Nicky Jam

— Mi playlist —

— Canción — Artista —

0 - Forget me too - MGK
1 - My ex's best friend - MGK
2 - Bloody Valentine - MGK

Funciones de la playlist

Ordenar la playlist por:

Ascendentemente por:

Nombre de la canción

Nombre del artista

Descendentemente por:

Nombre de la canción

Nombre del artista

Estado de ordenamiento:

Ascendentemente/nombre

-- Reproducir --

— Fila de reproducción —

— Canción — Artista —

0 - Bloody Valentine - MGK
1 - My ex's best friend - MGK
2 - Forget me too - MGK

Funciones de la fila de reproducción

Agregar canción:

Nombre: Forget me too

Artista: MGK

Agregar canción

Eliminarla canción número:

Estado de la reproducción:

Estado

Reproducción secuencial

Reproducción aleatoria

< >

8. Al presionar el botón de la fila de reproducción, se desplegará un menú con todas las opciones para la fila de reproducción.

— Fila de reproducción —

— Canción — Artista —

Inicio fila de reproducción

Sincronizar fila de reproducción con la playlist

Funciones de la fila de reproducción

Agregar canción:

Nombre:

Artista:

Agregar canción

Eliminar la canción número:

Estado de la reproducción:

Estado

Reproducción secuencial

Reproducción aleatoria

< >

9. Cuenta con la opción de sincronizar la lista de reproducción con la playlist.

— Mi playlist —

— Canción — Artista —

Funciones de la playlist:

Ordenar la playlist por:

Ascendentemente por:

Nombre de la canción

Nombre del artista

Descendentemente por:

Nombre de la canción

Nombre del artista

Estado de ordenamiento:

Ninguno

Reproducir

0 - Playa - Nicky Jam

1 - Amate - Micro TDH

2 - La Canción - J Balvin

3 - Ojos Color Sol - Calle 13

— Fila de reproducción —

— Canción — Artista —

0 - Bloody Valentine - MGK

1 - My ex's best friend - MGK

2 - Forget me too - MGK

Sincronizar fila de reproducción con la playlist

— Canción — Artista —

Funciones de la playlist:

Ordenar la playlist por:

Ascendentemente por:

Nombre de la canción

Nombre del artista

Descendentemente por:

Nombre de la canción

Nombre del artista

Estado de ordenamiento:

Ascendentemente/nombre

Reproducir

0 - Amate - Micro TDH

1 - Bloody Valentine - MGK

2 - Forget me too - MGK

3 - La Canción - J Balvin

4 - My ex's best friend - MGK

5 - Ojos Color Sol - Calle 13

6 - Playa - Nicky Jam

Sincronizar fila de reproducción con la playlist