

Manual Music Player



Jesus Antonio Arciniega Reynoso
TIS61D

Interfaz de usuario



Manual técnico.

Buscar archivo..

Se importan archivos a la lista enlazada que mostrará las canciones de la lista de reproducción.

Para poder agregar varios elementos al mismo tiempo, se implementó un arreglo de tipo string para almacenar las rutas de los archivos.

Una vez que se selecciona un archivo, este se agrega a la lista de reproducción mediante el método `insertarAlInicio` de la lista enlazada `listaEnlazadaCanciones`.

El método `mostrarListado`, como su nombre lo dice, es el encargado de que aparezca la lista enlazada en pantalla.

```
//Boton de busqueda de archivos
string[] nombre, ruta;
private void pictureBox7_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog1 = new OpenFileDialog();
    openFileDialog1.Multiselect = true;

    if (openFileDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        nombre = openFileDialog1.SafeFileNames;
        ruta = openFileDialog1.FileNames;
        for (int i = 0; i < ruta.Length; i++)
        {
            cancion = new Cancion(ruta[i]);
            listaEnlazadaCanciones.insertarAlInicio(cancion);
            mostrarListado();
        }
    }
}
```

Eliminar elemento..

Para eliminar elementos de la lista de reproducción, se utiliza el método `eliminarElementoIndice`, con el cual, al seleccionar una pista de la lista de reproducción se invoca a dicho método mediante el `pictureBox11` y utilizando un try and catch se controlan las excepciones.

```
//Botón para eliminar
private void pictureBox11_Click(object sender, EventArgs e)
{
    try
    {
        Cancion c = (Cancion)listaEnlazadaCanciones.eliminarElementoIndice(1stPlaylist.SelectedIndex);
        MessageBox.Show(c.Descripcion + " eliminado");
        axWindowsMediaPlayer1.Ctlcontrols.stop();
        mostrarListado();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

```
public object eliminarElementoIndice(int posicion)
{
    object datoRemovido;
    Nodo actual;

    if (esVacia())
    {
        return null;
    }
    else
    {
        if (posicion == 0)
        {
            datoRemovido = primero.Datos;
            primero = primero.Sig;
            return datoRemovido;
        }
        else
        {
            actual = primero;
            int contador = 0;

            while ((actual != null) && (contador < posicion - 1))
            {
                actual = actual.Sig;
                contador++;
            }

            if ((actual != null) && (contador == posicion - 1))
            {
                datoRemovido = actual.Sig.Datos;
                actual.Sig = actual.Sig.Sig;
                return datoRemovido;
            }
        }
    }

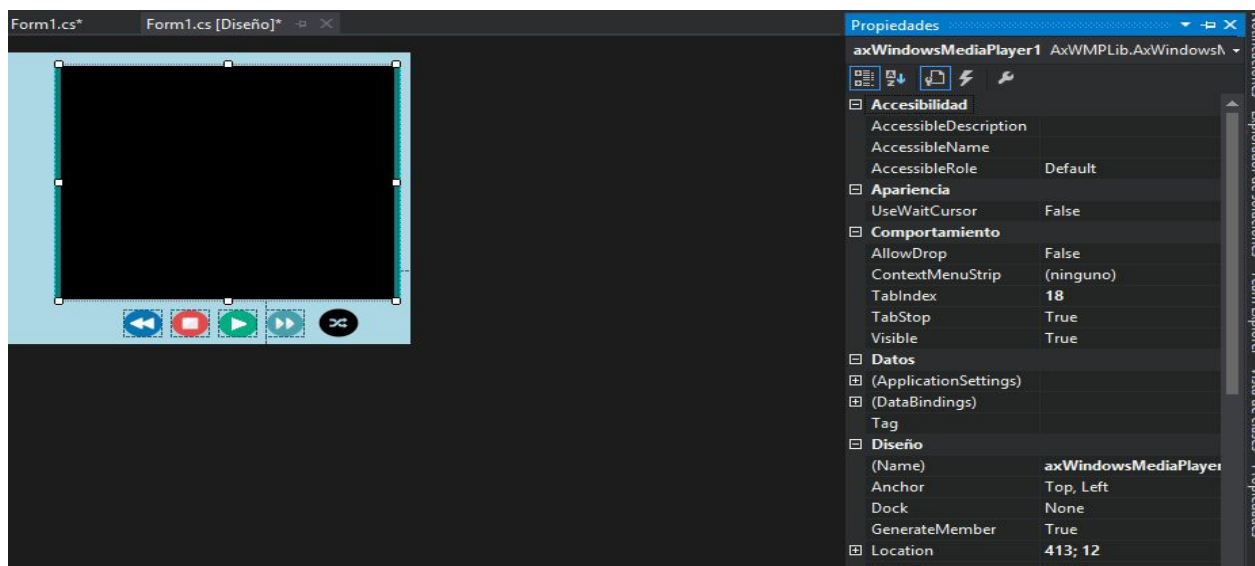
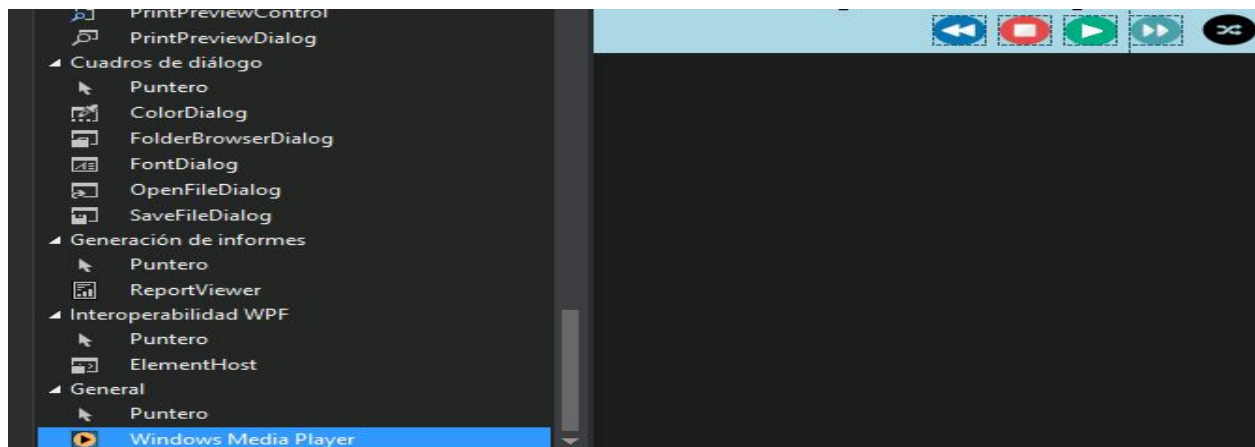
    return null;
}
```

Información de la pista.

La información de la lista se consiguió mediante el uso de la librería Windows Media Player.

Esta librería es la encargada de la animación e información de la pista

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using WMPLib;
```



Controles.

Reproducir

El botón de reproducción consta del siguiente código.

Se crea el objeto canción, el cual contiene en su elemento descripción la ruta de la canción seleccionada, para así, mediante el uso de `Ctlcontrols.play` de la librería WMP poder reproducirlo.

```
//Boton de reproduccion
private void pictureBox2_Click(object sender, EventArgs e)
{
    Cancion c = (Cancion)listaEnlazadaCanciones.getElementoIndice(1stPlaylist.SelectedIndex);
    axWindowsMediaPlayer1.URL = c.Descripcion;
    axWindowsMediaPlayer1.Ctlcontrols.play();
}
```

Detener

Simplemente es la línea que contiene el método `Ctlcontrols.stop` de WMP.

```
//Boton de detener
private void pictureBox3_Click(object sender, EventArgs e)
{
    axWindowsMediaPlayer1.Ctlcontrols.stop();
}
```

Anterior y Siguiente.

Este botón tiene una logica mas elaborada, ya que en estos, se obtiene la ruta de la canción anterior a la que se reproduce y/o la siguiente.

Ademas cuenta con la validación con la que se sabe si la reproducción es continua o aleatoria mediante la variable random que se explicará mas adelante.

```
//Boton anterior
private void pictureBox5_Click(object sender, EventArgs e)
{
    if (lstPlaylist.Items.Count == 0)
    {
        MessageBox.Show("No hay Archivos");
    }
    else
    {
        string actual, primero;
        primero = Convert.ToString(0);
        actual = Convert.ToString(lstPlaylist.SelectedIndex.ToString());
        if (random == 1)
        {
            int ultimom = lstPlaylist.Items.Count - 1;
            Random rdn = new Random();
            int shf = rdn.Next(0, ultimom);
            lstPlaylist.SelectedIndex = shf;
            Cancion c = (Cancion)listaEnlazadaCanciones.getElementoIndice(lstPlaylist.SelectedIndex);
            axWindowsMediaPlayer1.URL = c.Descripcion;
            axWindowsMediaPlayer1.Ctlcontrols.play();
        }
        else
        {
            if (actual.Equals(primerom))
            {
                lstPlaylist.SelectedIndex = lstPlaylist.Items.Count - 1;
                Cancion c = (Cancion)listaEnlazadaCanciones.getElementoIndice(lstPlaylist.SelectedIndex);
                axWindowsMediaPlayer1.URL = c.Descripcion;
                axWindowsMediaPlayer1.Ctlcontrols.play();
            }
            else
            {
                lstPlaylist.SelectedIndex = lstPlaylist.SelectedIndex - 1;
                Cancion n = (Cancion)listaEnlazadaCanciones.getElementoIndice(lstPlaylist.SelectedIndex);
                axWindowsMediaPlayer1.URL = n.Descripcion;
                axWindowsMediaPlayer1.Ctlcontrols.play();
            }
        }
    }
}
```



```

//Boton de siguiente
private void pictureBox4_Click(object sender, EventArgs e)
{
    if (lstPlaylist.Items.Count == 0)
    {
        MessageBox.Show("Lista de reproducción vacía");
    }
    else
    {
        string ultimo, actual;
        ultimo = Convert.ToString(lstPlaylist.Items.Count - 1);
        actual = Convert.ToString(lstPlaylist.SelectedIndex.ToString());
        if (random == 1)
        {
            int ultimom = lstPlaylist.Items.Count - 1;
            Random rdn = new Random();
            int shf = rdn.Next(0, ultimom);
            lstPlaylist.SelectedIndex = shf;
            Cancion c = (Cancion)listaEnlazadaCanciones.getElementoIndice(lstPlaylist.SelectedIndex);
            axWindowsMediaPlayer1.URL = c.Descripcion;
            axWindowsMediaPlayer1.Ctlcontrols.play();

        }
        else
        {
            if (actual.Equals(ultimo))
            {
                lstPlaylist.SelectedIndex = 0;
                Cancion c = (Cancion)listaEnlazadaCanciones.getElementoIndice(lstPlaylist.SelectedIndex);
                axWindowsMediaPlayer1.URL = c.Descripcion;

                axWindowsMediaPlayer1.Ctlcontrols.play();
            }
            else
            {
                lstPlaylist.SelectedIndex = lstPlaylist.SelectedIndex + 1;
                Cancion cn = (Cancion)listaEnlazadaCanciones.getElementoIndice(lstPlaylist.SelectedIndex);
                axWindowsMediaPlayer1.URL = cn.Descripcion;

                axWindowsMediaPlayer1.Ctlcontrols.play();
            }
        }
    }
}

```


Aleatorio.

Este botón regresa la validación que se utiliza en los botones anterior y siguiente para decidir si la reproducción es continua o aleatoria.

```
//Random desactivado
public int random = 0;
private void pictureBox8_Click(object sender, EventArgs e)
{
    if (random == 0)
    {
        random = 1;
        pictureBox10.Visible = true;
    }
    else
    {
        random = 0;
        pictureBox8.Visible = true;
    }
}

//Random activado
private void pictureBox10_Click(object sender, EventArgs e)
{
    if (random == 1)
    {
        random = 0;
        pictureBox10.Visible = false;
    }
    else
    {
        random = 1;
        pictureBox8.Visible = false;
    }
}
```

Código fuente en: <https://github.com/chubuntuarc/Music-Player>