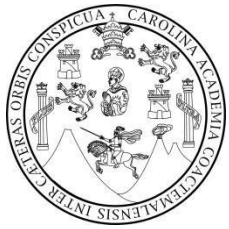
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES Y DE PROGRAMACIÓN



Segundo Semestre 2018

Sección	Catedrática	Tutor académico
A-	Inga. Damaris Campos de López	Aylin Zulema Aroche Arbizú
B-	Inga. Zulma Aguirre	César Javier Solares Orozco

Enunciado de Proyecto No. 1

Objetivos:

Esta práctica tiene como objetivo que el estudiante practiqué el análisis léxico de un lenguaje formal y que amplié sus conocimientos con el lenguaje de programación C#, así como aplicar los conocimientos adquiridos en la clase y en el laboratorio.

Descripción:

La empresa Digital InGame Solutions ha quedado satisfecho con el programa previo que ha realizado, razón por la cual ha decidido volver a solicitar sus servicios. Esta vez la empresa le solicita crear un reproductor de música para el área de entretenimiento con la que esta cuenta.

El reproductor de música cuenta con un lenguaje definido y la carga de canciones a la playlist se realiza desde un archivo con extensión .plst; además en la ventana donde se carga el contenido, se podrá editar el texto y llenar directamente la playlist.

Para la reproducción de la música se le pide utilizar la librería de Windows Media Player por su facilidad de uso. La aplicación debe proveer la opción de analizar el contenido del editor, y deberá generar como salida una ventana con los resultados, en caso de que el archivo contenga errores, mostrará en una nueva ventana que será mostrada a partir de un botón una tabla con el detalle de estos.

Otro requerimiento que se le solicita es que utilice la herramienta **ListView** para mostrar y poder seleccionar las canciones con todos sus atributos.

Definición del lenguaje:

El lenguaje tiene varios bloques:

- **Etiqueta Playlist**
  Dentro del lenguaje propio de la aplicación es posible indicar que se desea realizar una playlist de la siguiente manera:

```
 <Playlist nombre=” nombre de la playlist”>
<Cancion></Cancion> </Playlist>
```

Para cada etiqueta ***playlist*** que se ingrese en el archivo de entrada, se generará una tabla en una nueva ventana que mostrará el nombre de la playlist y el reproductor correspondiente. El segmento <nombre de la playlist> en la etiqueta ***playlist*** es una cadena que contiene el nombre de la lista de reproducción, la etiqueta <Cancion> se detallará más adelante.

- **Etiqueta Cancion:**

  Esta etiqueta permite agregar una canción a la playlist donde se encuentre, la sintaxis es la siguiente:

```
<Cancion url=”url” artista=”artista” álbum=”álbum” año=”año” duracion=”duración” >
nombre canción </Cancion>
```

  A continuación, se detallan los elementos de la etiqueta ***Cancion***:

  - ***url***: es una cadena de texto que indica la dirección donde se encuentra el archivo.
  - ***artista***: es una cadena de texto que indica el artista.
  - ***álbum***: cadena de texto que indica el álbum de la canción.
  - ***Año***: secuencia de números que indican el año de lanzamiento de la canción.
  - ***Duración***: secuencia de caracteres que indican la duración de la canción.
  - ***nombre canción:*** *en este segmento se encontrará el nombre de la canción.*

  Dentro del bloque ***playlist*** pueden venir una o más etiquetas ***canción***.

- **Etiqueta Repetición:**
  Esta etiqueta permite agregar un numero determinado de veces la canción o canciones a la playlist.

```
<Repeticion nveces=”5”>

<Cancion url=”url” artista=” artista” álbum=” álbum” año=” año” duracion=” duración” >
        nombre canción </Cancion>
<Cancion url=”url2” artista=” artista2” álbum=” álbum2” año=” año2” duracion=” duración2” >
        nombre canción2 </Cancion>
</Repeticion>
```

  A continuación, se detalla el elemento de la etiqueta **Repeticion:**

  - **Nveces**: Número de veces que se agregaran los elementos que se encuentren dentro.

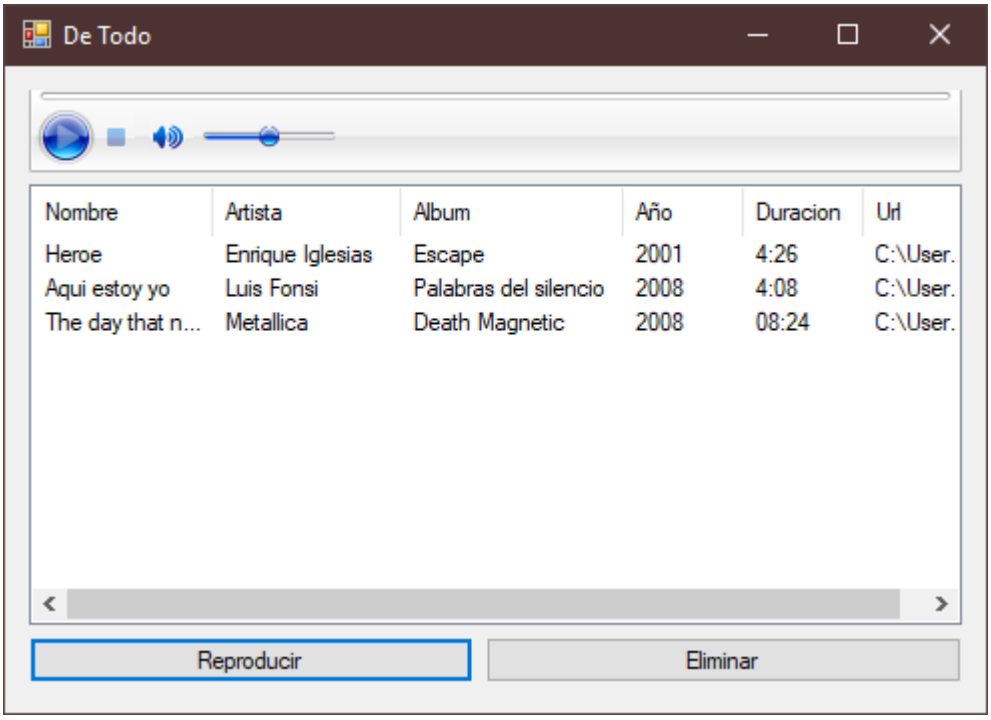
Archivo de entrada ejemplo:

```
<playlist nombre=” De Todo”>
        <Cancion url=” C:\Users\Javier Solares\Music\iTunes\iTunes Media\Downloads\The day
         that never comes.mp3” artista=” Metallica” álbum=” Death Magnetic” año=”2008”
         duracion=” 8:24”> The day that never comes
        </Cancion>
        <Repeticion nveces=”8”>
        <Cancion url=” C:\Users\Javier Solares\Music\iTunes\iTunes Media\Downloads\Heroe.mp3”
        artista=” Enrique Iglesias” álbum=” Escape” año=”2001” duracion=” 4:26”>
        Héroe
        </Cancion>
 <Cancion url=” C:\Users\Javier Solares\Music\iTunes\iTunes Media\Downloads\Aqui estoy
        yo.mp3” artista=” Luis Fonsi” álbum=” Palabras del silencio” año=”2008”
         duracion=” 4:08”> Aquí estoy yo
        </Cancion>

        </Repeticion>


</playlist>
```

Al ejecutar las instrucciones anteriores en el editor, se generaría una ventana que mostraría algo como esto:



En la imagen anterior podemos observar que en la parte superior de la ventana resultante se muestra el nombre que se le dio a la playlist como título.
Cada una de canciones agregadas con el archivo de ejemplo se pueden observar como elemento dentro de la herramienta de visual studio **ListView.** También se puede apreciar los botones reproducir y eliminar que seleccionando una canción primero, se puede reproducir o eliminar dicho elemento.

Funcionalidad del reproductor:

El reproductor deberá reproducir secuencialmente las canciones en el orden en el que se agregaron al darle el botón reproducir, a excepción de que, si se selecciona una canción y se presiona el botón reproducir, en este caso el reproductor empezará desde la canción seleccionada.

Aplicación:
La aplicación debe analizar el texto del editor, de encontrar errores léxicos, la aplicación debe informar al usuario y mostrar un reporte de errores en formato tabla en una nueva ventana, en este caso no debe mostrar la ventana del reproductor. En la tabla con el reporte de errores se deben mostrar los siguientes campos:

No.	Error	Descripción	Fila	Columna
1	%	Elemento léxico desconocido	3	4

De no encontrar error alguno, se debe generar la ventana de salida correspondiente. La aplicación debe pintar dentro del editor cada uno de los lexemas, según su tipo, los colores se definen a continuación:

Tipo	Color
Palabras reservadas	Celeste
Números	Amarillo
Cadenas	Verde
Dos puntos	Rosado
Punto y coma	Rojo
Llaves	Morado
Coma	Azul
Otros	Negro

Además de colorear los elementos léxicos y generar la ventana del reproductor, se deberá poder acceder mediante un botón a una nueva ventana que contenga una Tabla de tokens con los siguientes campos:

No.	Lexema	Tipo	Fila	Columna
1	Cancion	Reservada	2	1
2	“canción 2”	Cadena	2	12
3	:	Dos puntos	2	11

Interfaz gráfica:

Componentes mínimos de la interfaz:

1. **Editor de texto:** Debe ser un área de texto en el cual se puedan escribir nuevas instrucciones de entrada o bien cargar texto desde archivos con extensión .plst
2. **Abrir:** La interfaz debe proveer la capacidad de abrir archivos con extensión .plst.
3. **Guardar:** La aplicación debe proveer la capacidad de guardar el texto contenido en el editor. Si no ha sido guardada nunca, debe proveer una opción para ingresar el nombre del nuevo archivo y la ubicación en la que se desea guardar.
4. **Guardar como:** Está opción permite guardar el archivo de entrada con otro nombre, se debe preguntar el nombre del nuevo archivo.
5. **Analizar:** Debe realizar el análisis léxico del lenguaje que se encuentra actualmente en el editor y generar la salida correspondiente.
6. **Tabla de Tokens:** Debe generar una ventana donde se visualice la tabla de tokens.
7. **Errores:** En caso de existir algún error deberá generar una ventana donde se visualice la tabla de errores.
8. **Acerca de:** Debe desplegar una ventana con los datos del estudiante y del curso.
9. **Salir:** Debe terminar la ejecución de la aplicación.

Entregables:

- Manual de Usuario
- Manual Técnico: Dentro de este manual debe incluirse el autómata finito determinista que se utilizó para la implementación del analizador léxico.
  Para la obtención del autómata se debe de utilizar expresiones regulares y el método del árbol los cuales también deben de adjuntarse en este documento.
- Código fuente.
- Ejecutable del proyecto.

Documentación a entregar de forma física el día de la calificación:

- Hoja de calificación (Original y una copia)

Notas importantes:

- El proyecto se debe desarrollar de forma individual.
- Este proyecto se deberá desarrollar utilizando C# con Visual Studio.
  El proceso de obtener tokens, se debe hacer a través de la implementación del autómata finito determinista desarrollado por el propio estudiante a través del método del árbol y expresiones regulares.
- La calificación se realizará **ÚNICAMENTE** con los archivos que el auxiliar provea en el momento de la calificación, dichos archivos no se deben modificar.
- La calificación del proyecto será personal y durará como máximo 30 minutos, en un horario que posteriormente será establecido. Se debe tomar en cuenta que durante la calificación no podrán estar terceras personas alrededor, de lo contrario no se calificará la práctica.
- No se dará prórroga para la entrega del proyecto.
- **Copia parcial o total del proyecto tendrá una nota de 0 puntos, y se notificará a la Escuela de Sistemas para que se apliquen las sanciones correspondientes.**
- **En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará el proyecto; por lo cual, se tendrá una nota de cero puntos.**

Fecha de entrega: 19 de septiembre de 2018