



**Primer Semestre 2016**

Sección	Catedrática	Tutor académico
A-	Inga. Damaris Campos de López	Eiji de Paz
B-	Inga. Zulma Aguirre	Luis Fernando Lara Lemus

## **Enunciado de Práctica No. 1**

### **Objetivos:**

- Que el estudiante implemente un analizador léxico básico para poder obtener información de un archivo de entrada.
- Conocer y aprender el lenguaje de programación Visual Basic.Net.
- Aplicar los conocimientos adquiridos en la clase y laboratorio.

### **Descripción:**

La práctica consiste en realizar un reproductor de canciones musicales, desarrollarlo utilizando Windows Forms en Visual Basic.NET, para dicho reproductor se contará con un lenguaje definido, que tendrá que ser analizado léxicamente para poder obtener las notas musicales y posteriormente reproducirlas, mostrando la nota que se esté tocando en tiempo real.

La información de la canción, nombre y notas musicales se obtendrá a partir de un archivo de entrada con la extensión “.xml”, que se tendrá que mostrar en un editor de texto. También es posible crear una o varias canciones en diferentes pestañas dentro del editor de texto. Para iniciar el análisis del archivo de la canción se contará con el botón “Scanner”, y para comenzar la reproducción de la canción es necesario presionar el botón “play” y para detenerla se presionará el botón “stop”.

Como salidas, la aplicación contará con un apartado donde se mostrará el nombre de la canción que se está reproduciendo, así como la nota musical que actualmente se está tocando.

Adicionalmente es necesario que la aplicación muestre en una página HTML los resultados del análisis, dichos resultados son Tabla de Token's; y si existen errores, mostrarlos.

### **Definición del lenguaje:**

El lenguaje contará con las siguientes etiquetas:

```
<cancion>
    .....
</cancion>
```

Es la etiqueta inicial del archivo, sirve para que el analizador sepa que se está iniciando la escritura de una canción.

```
<nombre>
    .....
</nombre>
```

Esta etiqueta sirve para indicarle al analizador el nombre que tendrá nuestra canción. (puede ser cualquier nombre, incluso con dígitos y caracteres especiales como: ‘\_’, ‘@’, ‘#’)

```
<partitura>
    .....
</partitura>
```

Esta etiqueta le sirve al analizador para saber que vendrá un conjunto de notas dentro de ella, es importante mencionar que esta etiqueta puede venir varias veces.

```
<nota>
.....
</nota>
```

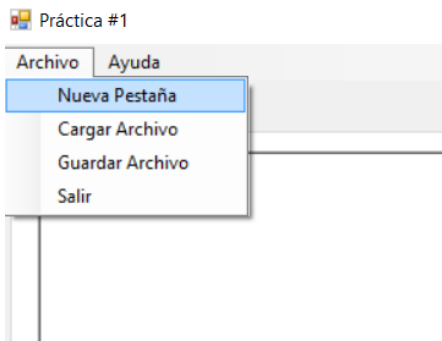
Dentro de esta etiqueta vendrá una nota musical (se especificará más adelante como vienen), es importante mencionar que pueden venir muchas notas musicales.

A continuación, se muestra un archivo de entrada completo, el cual contiene una canción, con 2 partituras y varias notas musicales:

```
<cancion>
  <nombre>Bugle call</nombre>
  <partitura>
    <nota>G2</nota>
    <nota>C3</nota>
  </partitura>
  <partitura>
    <nota>E2</nota>
    <nota>C3</nota>
  </partitura>
</cancion>
```

## Interfaz gráfica:

### 1. Menú



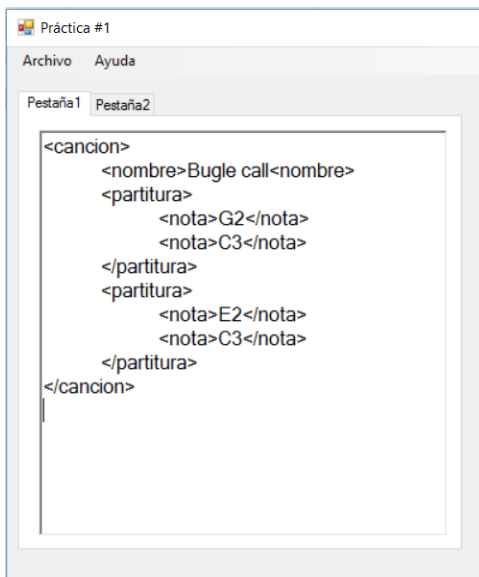
de la aplicación.

- **Nueva Pestaña:** Al presionar este botón es posible crear una nueva pestaña dentro del editor de texto.

- **Cargar Archivo:** Al presionar este botón es posible cargar un archivo desde cualquier ruta de la computadora, a nuestra pestaña actual

- **Guardar Archivo:** Al presionar este botón es posible guardar la canción de la pestaña actual en cualquier ruta dentro de la computadora.

- **Salir:** Al presionar este botón se saldrá



- **Manual Aplicación:** Al presionar este botón se abrirá un PDF que contendrá el manual de usuario de la aplicación.

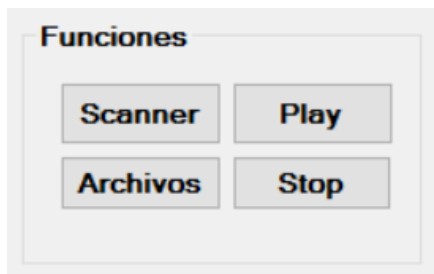
- **Acerca de...:** Al presionar este botón se mostrará una pequeña ventana con los datos del estudiante.

## 2. Editor de Texto

En esta área es posible editar el texto que será analizado para cargar las notas musicales, es importante mencionar que es posible abrir  $n$  pestañas con  $n$  archivos de canciones y al momento de presionar el botón de “analizar”, se analizara la canción que esté en la pestaña seleccionada; el nombre de la pestaña es igual al nombre del archivo (si ya existe).

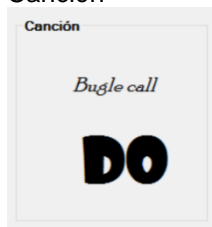
También hay que destacar que, al momento de cargar un archivo, se mostrará la información dentro del área de texto de la pestaña seleccionada actualmente.

## 3. Funciones



- **Scanner:** Botón que inicia el análisis léxico del archivo de entrada.
- **Play:** Botón que da inicio a la reproducción de la canción.
- **Stop:** Botón que finaliza la reproducción de la canción.
- **Archivos:** Muestra las páginas HTML que contendrán los tokens encontrados o bien los errores.

## 4. Canción



- En este apartado se debe mostrar el nombre de la canción que se está reproduciendo (Bugle call) y abajo se debe mostrar la nota musical que se está tocando actualmente.

Es importante mencionar que las notas musicales deben de cambiar según se vayan tocando en la canción.

- **Especificación de Notas Musicales:**

Letra (Notación Inglesa)	Número	Significado (Notación Latina)	Ejemplo en este proyecto
C	[0..9]	Do	C4
D	[0..9]	Re	D2
E	[0..9]	Mi	E4
F	[0..9]	Fa	F3
G	[0..9]	Sol	G3
A	[0..9]	La	A5
B	[0..9]	Si	B8

Nota: El número solo indica la intensidad de la nota, puede ir desde 0 (grave) hasta 9 (agudo)

### **Archivos de Salida:**

El archivo de Tokens se debe mostrar en una página HTML, la información en una tabla como se muestra a continuación:

#	Lexema	Id Token	Token
1	nota	11	Reservada NOTA
2	E2	12	Nota Musical
3	/	13	Cierre Etiqueta

El archivo de Errores, si existieran, se debe mostrar en una página HTML con la información en una tabla como se muestra a continuación:

#	Fila	Columna	Carácter	Descripción
1	05	10		Desconocido
2	08	30	¬	Desconocido
3	10	05	~	Desconocido

### **Entregables que se deben incluir en el CD:**

- Manual de Usuario
- Manual Técnico, debe incluir el DFA que se use para el analizador léxico.
- Código Fuente
- Ejecutable de la Aplicación

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

- Hoja de calificación (Original y una copia)

### **Notas importantes:**

- La práctica se debe desarrollar de forma individual.
- Esta práctica se deberá desarrollar utilizando Visual Basic .Net con Visual Studio 2013.
- Se puede usar solamente la librería "Toub.Sound.Midi". Pueden tomar como referencia la siguiente documentación (y descarga de la misma):  
<http://grouplab.cpsc.ucalgary.ca/cookbook/index.php/VisualStudio/HowToPlayMIDIInstruments>
- El proceso de obtener tokens, se debe hacer a través de la implementación del autómata finito determinista desarrollado por el propio estudiante.
- No se puede agregar o quitar algún símbolo en el archivo de entrada. El proyecto deberá funcionar con los archivos de prueba que se disponga para la calificación, sin modificación.
- La calificación de la práctica 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 de la práctica.
- **Copia parcial o total de la práctica 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: 18 de febrero de 2016**