



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Lenguajes Formales y de Programación.

Proyecto 2

Objetivos:

- Realizar las fases de análisis léxico y sintáctico
- Manejo de librerías para la creación de archivos pdf.
- Utilizar el lenguaje Visual Basic .Net

Descripción

El proyecto consiste en la realización de una aplicación con la cual se puedan generar archivos con extensión PDF.

La aplicación es un editor específico para la creación de archivos PDF, el lenguaje se detallará más adelante. Por medio de este lenguaje se podrán generar archivos PDF con todas las especificaciones que éste nos permite, como el tipo de letra, el tamaño de letra; además de esto, nos permitirá agregar imágenes, listas, etc.

La creación de archivos PDF, se realiza utilizando una librería llamada iText, la cual debe descargarse del sitio oficial de iText, e importarla al proyecto.

El propósito de la realización de este proyecto, es comprender completamente las fases de análisis léxico y sintáctico, identificando debidamente la gramática que posee el lenguaje.

Funciones

La aplicación debe ser capaz de efectuar las siguientes acciones:

- **Cargar Archivo:**
Deberá poder cargar el archivo de entrada a la aplicación, desde una ruta a especificar, y mostrarla en un área de texto.
- **Guardar Archivo:**
Permite guardar el archivo de texto, en una ruta específica, el código que se tenga actualmente en el área de texto.
- **Opción de Análisis:**
Al realizar esta operación, se realizará el análisis léxico y sintáctico sobre el contenido dentro del área de texto.
- **Mostrar Tokens:**
Al realizar esta operación, mostrará el archivo PDF donde se encuentra la tabla con los tokens encontrados.
- **Mostrar Errores:**
Al realizar esta operación mostrará dos archivos PDF, uno que contendrá los errores léxicos detectados, y otro con los errores sintácticos encontrados sobre el código en el área de texto.
- **Mostrar Resultado:**
Al realizar esta operación, mostrará el archivo PDF resultante de la interpretación del código en el área de texto.

Lenguaje:

La forma de crear el archivo PDF es la siguiente:

➤ **Comentarios:**

Para especificar comentarios en cualquier parte del programa será:

```
/* Comentario */
```

➤ **Crear Nuevo Documento:**

Para la creación de documentos, se realizará de la siguiente manera:

Parámetros:

- **Nombre:** Con este parámetro se especifica el nombre del archivo PDF a generar.
- **Path:** Con este parámetro se especifica el directorio dentro del cual se va a crear el archivo.

Ejemplo:

```
CrearNuevoDocumento ("Nombre", "Path") {  
    /*Dentro de esta función se encuentran las características del documento */  
};
```

➤ **Características del Documento:**

Se utilizará esta función para especificar las características del documento, las cuales pueden ser las siguientes:

- **Tipo de Letra**
Especifica el tipo de letra del documento.
- **Tamaño de Páginas**
Especifica el tamaño de las páginas del documento.
- **Tamaño de Letra**
Especifica el tamaño de la letra del documento.

Ejemplo:

```
Caracteristicas(){  
    TipoLetra("COURIER");  
    TamanoLetra(12);  
    TamanoPagina("Carta");  
};
```

➤ **Crear Nueva Página:**

Se utilizará esta función para la creación de páginas dentro del documento:

Ejemplo:

```
CrearNuevaPagina(){  
    /* Todo el contenido de la página.*/  
};
```

A continuación se especifican las funciones que pueden ir dentro de una nueva página.

➤ **Insertar Título:**

Para ingresar un título se hará de la siguiente forma, el título siempre tiene que ir en negrita y centrado:

Parámetros:

- Todo el texto que queramos ingresar.

Ejemplo:

```
InsertarTitulo( "Editor pdf.");
```

➤ **Insertar Párrafo:**

Para ingresar texto en un párrafo se hará de la siguiente manera, luego de ingresar un párrafo tiene que haber un cambio de línea:

Parámetros:

- Texto: Todo el texto que queramos ingresar.
- Formato de Letra: En este atributo podemos colocar Normal, Negrita o Cursiva.

Ejemplo:

InsertarParrafo("Este es el primer párrafo del documento.", Cursiva);

➤ **Insertar Lista:**

Para la creación de las listas se utilizará el siguiente patrón:

Parámetros:

- Cada uno de los parámetros son los componentes de la lista. No hay límite de parámetros para esta función.

Ejemplo:

InsertarLista("Uno", "Dos", "Tres");

➤ **Insertar Imagen:**

Para la creación de las listas utilizaremos el siguiente patrón:

Parámetros:

- Path: Se ingresa la ruta en la que se encuentra la imagen que se quiere insertar, incluyendo el nombre del archivo.

Ejemplo:

InsertarImagen("Path");

Procedimiento:

Las dos formas válidas para ingresar información, son las siguientes:

- Proporcionando un archivo de entrada con la sintaxis definida anteriormente, que deberá ser cargado a la aplicación mediante el analizador léxico y sintáctico desarrollado.
- Ingresando el código de forma manual en la aplicación.

Archivos a generar:

Se deben de generar cuatro archivos:

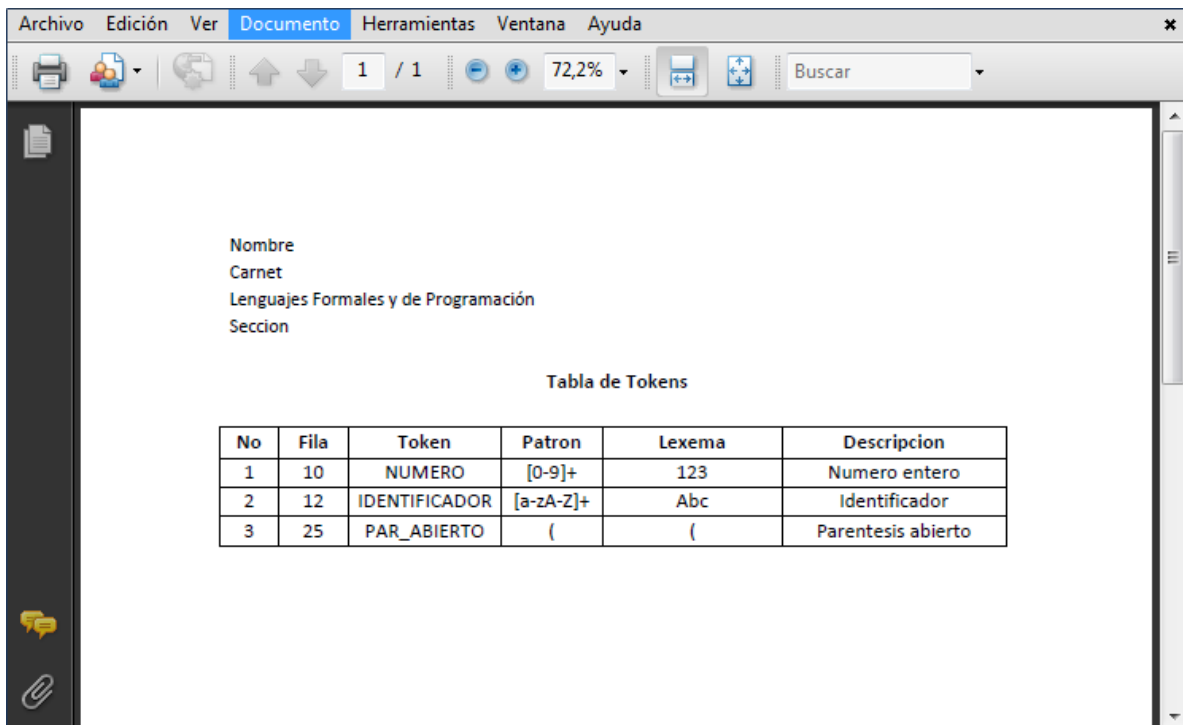
- Un archivo PDF con los tokens encontrados en el archivo de entrada (correlativo, línea, token, patrón, lexema y descripción).
- Un archivo PDF con los errores léxicos encontrados en el archivo de entrada (correlativo, símbolo, fila, columna, descripción).
- Un archivo PDF con los errores sintácticos encontrados en el archivo de entrada (correlativo, línea, posición, lexema, token leído, token esperado).
- El archivo PDF resultante.

Ejemplo:

```
CrearNuevoDocumento("prueba.pdf", "C://"){  
  
    Caracteristicas(){  
  
        TipoLetra("COURIER");  
        TamanoLetra(12);  
        TamanoPagina("Carta");  
  
    };  
  
    CrearNuevaPagina(){  
  
        InsertarTitulo("Editor pdf");  
        InsertarParrafo( "Este es el primer párrafo del documento.", Normal);  
        InsertarLista("Iniciar", "Finalizar");  
        InsertarParrafo( "Este es el segundo párrafo del documento.", Normal);  
        InsertarImagen("C://Imágenes/imagen.jpg");  
  
    };  
};
```

Archivo de Tokens:

Debe contener las columnas Número, Fila, Token, Patrón, Lexema y Descripción.



The screenshot shows a software window with a menu bar (Archivo, Edición, Ver, Documento, Herramientas, Ventana, Ayuda) and a toolbar. The main area displays a document with the following text:

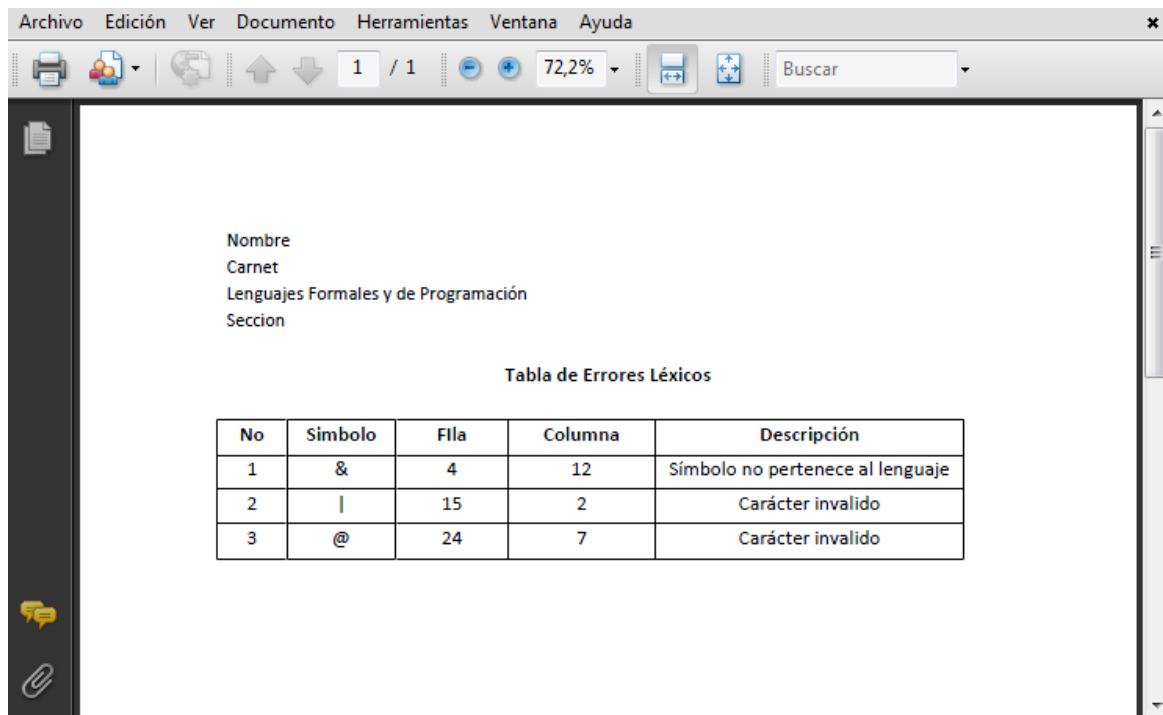
Nombre
Carnet
Lenguajes Formales y de Programación
Sección

Below the text is a table titled "Tabla de Tokens".

No	Fila	Token	Patrón	Lexema	Descripción
1	10	NUMERO	[0-9]+	123	Numero entero
2	12	IDENTIFICADOR	[a-zA-Z]+	Abc	Identificador
3	25	PAR_ABIERTO	((Parentesis abierto

Archivo de Errores Léxicos:

Debe contener las columnas Número, Símbolo, Fila, Columna y Descripción del error.



The screenshot shows a software window with a menu bar (Archivo, Edición, Ver, Documento, Herramientas, Ventana, Ayuda) and a toolbar. The main content area displays the following information:

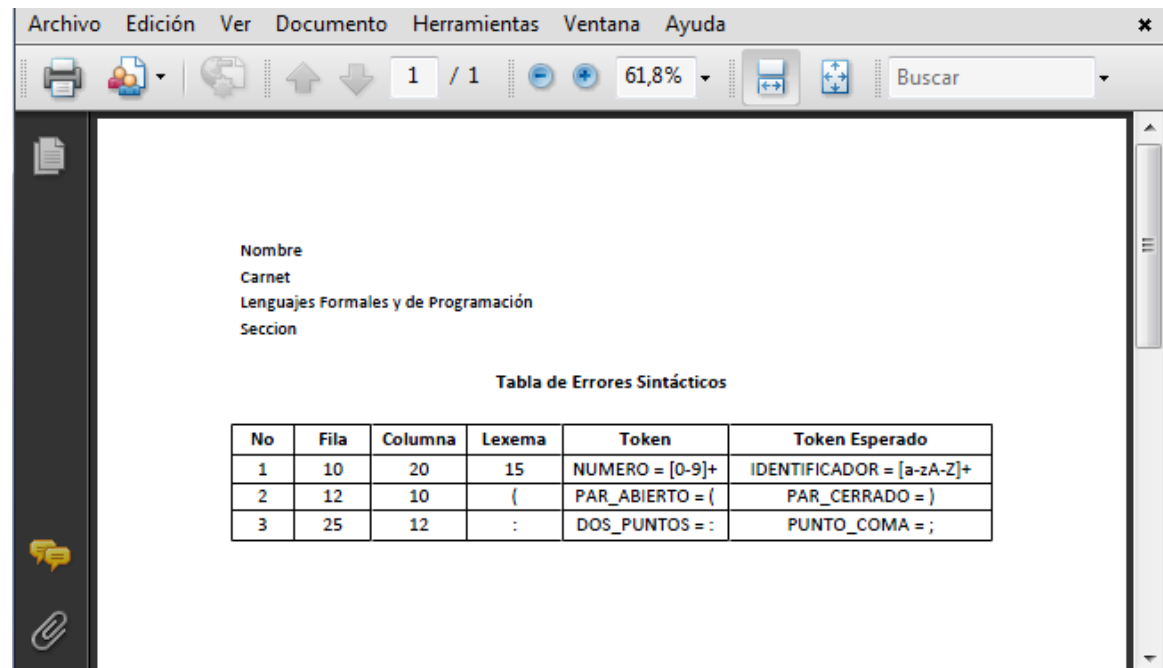
Nombre
Carnet
Lenguajes Formales y de Programación
Sección

Tabla de Errores Léxicos

No	Símbolo	Fila	Columna	Descripción
1	&	4	12	Símbolo no pertenece al lenguaje
2		15	2	Carácter inválido
3	@	24	7	Carácter inválido

Archivo de Errores Sintácticos:

Debe contener las columnas Número, Fila, Columna, Lexema, Token Leído y Token Esperado.



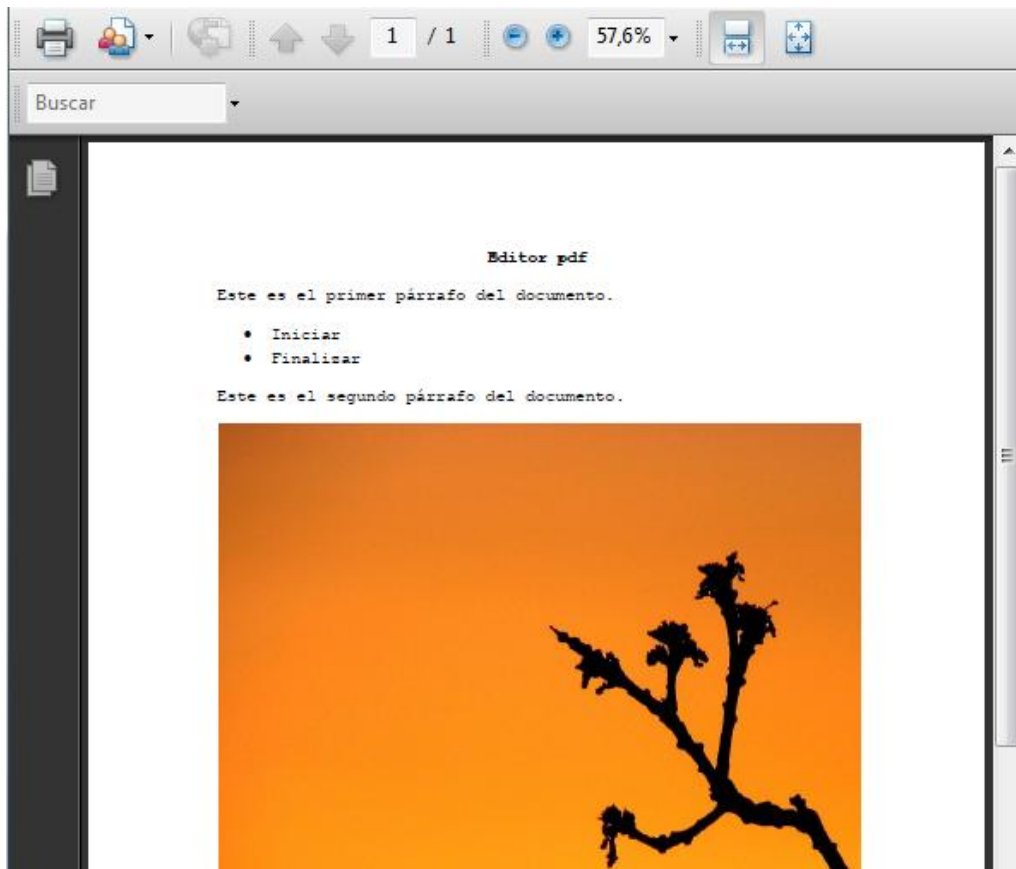
The screenshot shows a software window with a menu bar (Archivo, Edición, Ver, Documento, Herramientas, Ventana, Ayuda) and a toolbar. The main content area displays the following information:

Nombre
Carnet
Lenguajes Formales y de Programación
Sección

Tabla de Errores Sintácticos

No	Fila	Columna	Lexema	Token	Token Esperado
1	10	20	15	NUMERO = [0-9]+	IDENTIFICADOR = [a-zA-Z]+
2	12	10	(PAR_ABIERTO = (PAR_CERRADO =)
3	25	12	:	DOS_PUNTOS = :	PUNTO_COMA = ;

Archivo Resultante:



Documentación a Entregar:

Lo siguiente documentación es requisito para calificación y se deberá entregar en un disco (CD o DVD):

- Manual de Usuario y Manual Técnico; dentro del manual técnico debe incluir:
 - Patrón de los Componentes Léxicos (Tokens), por medio de la Expresión Regular que corresponda a cada Token.
 - Desarrollo del método del árbol para todos los tokens utilizados.
 - Gramática libre del contexto (tipo #2) que corresponda al proyecto (analizador sintáctico).
- Ejecutable
- Código fuente de la práctica.

****Lo único que se debe entregar impreso es la hoja de calificación.**

Notas importantes:

Se debe desarrollar de forma individual

- Debe desarrollar la aplicación utilizando Visual Basic .Net 2008
- Copia parcial o total del proyecto, tendrá una nota de 0 puntos para las personas involucradas y se notificará a la Escuela, para que se apliquen las sanciones correspondientes.
- No se dará prórroga para la entrega del proyecto.
- La forma de entrega, se les dará a conocer en los próximos días por medio de la UV.

Fecha de entrega: miércoles 2 de noviembre 2011.