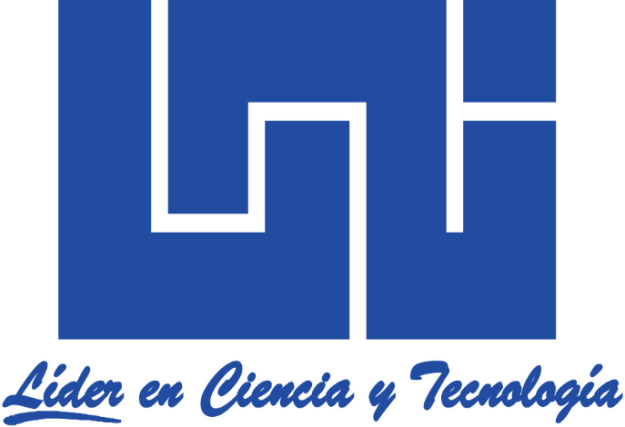
****

* Jesús Enmanuel Hernández González (2017-0505U)
* Ronaldo A. Gaitán Norori (2017-1010U)
* Kevin Cristopher Velarde (2017-1427U)
* Cinthia Michelle Palma Rivera(2017-1045U)

Fecha:29 de Julio del 2019

Fecha: 29 de Julio del 2019

Guía de Usuario

UNIVERSIDAD NACIONAL DE INGENIERIAS

FACULTAD DE ELECTROTECNIA Y COMPUTACION

Teoría de la computación

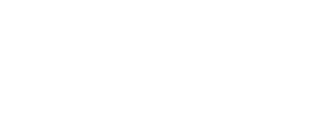
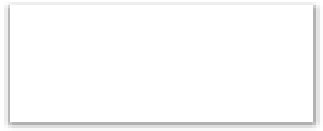
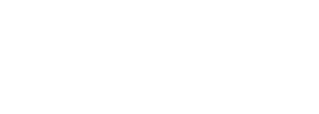
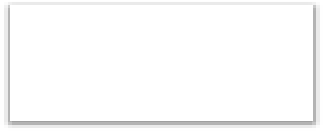
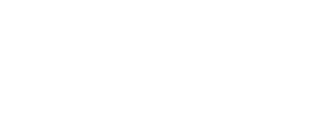
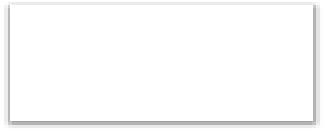
Profesora:

***Lizette Carolina Duarte Mora***

Integrantes:

# Introducción

Una de las aplicaciones más importantes de los AF s es la construcción de analizadores léxicos. Dentro del contexto general de un compilador, un analizador léxico (AL) tiene como principal función generar una lista ordenada de tokens a partir de los caracteres de entrada. Esos tokens, o componentes léxicos, serán usados por el analizador sintáctico para construir el correspondiente árbol sintáctico. Por otro lado, guarda información sobre algunos tokens, necesaria en el proceso de análisis y síntesis, en forma de atributos de esos componentes léxicos. Así pues, se debería considerar al analizador léxico como un módulo subordinado al analizador sintáctico (AS), tal y como se indica en el esquema de interacción entre ambos, que aparece en la figura a continuación.



**Componente Léxico**

**Programa Fuente**

Analizador

Léxico

Analizador

Sintáctico

**Solicita el Siguiente**

**Componente Léxico**

**Árbol de Análisis**

**Sintáctico**

Tabla de

Símbolos

1

# Analizador Léxico

Un analizador léxico es la primera fase de un compilador, consistente en un programa que recibe como entrada el código fuente de otro programa (secuencia de caracteres) y produce una salida compuesta de tokens (componentes léxicos) o símbolos. Estos tokens sirven para una posterior etapa del proceso de traducción, siendo la entrada para el analizador sintáctico (en inglés parser).

# Funciones

# 1. Manejo del fichero de entrada del programa fuente.

# 2. Eliminar comentarios, espacios en blaco, tabuladores y saltos de linea.

# 3. Inclusion de ficheros.

# 4. Contabilizar el numero de lineas y columnas para emitir mensajes de error.

# 5. Reconocimiento y ejecucion de las directivas de compilacion.

# Pasos del analizador léxico

Un analizador sintáctico o parser (viene del inglés: parse-analizar una cadena o texto en componentes sintácticos lógicos) es un programa que normalmente es parte de un compilador. El compilador se asegura de que el código se traduce correctamente a un lenguaje ejecutable. La tarea del analizador es, en este caso, la descomposición y transformación de las entradas en un formato utilizable para su posterior procesamiento. Se analiza una cadena de instrucciones en un lenguaje de programación y luego se descompone en sus componentes individuales.

# 

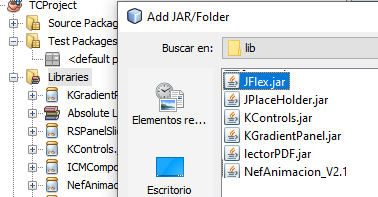
# Como crear un Analizador Léxico

El analizador léxico lee caracteres de entrada hasta que detecta que con el último carácter leído se puede formar un token, o un error en su caso, y comunica el evento correspondiente al analizador sintáctico. Si no hubo error, el AS procesa el token, y el AL no vuelve a entrar en juego hasta que el analizador sintáctico vuelva a necesitar otro token del flujo de entrada.

En este caso nos encontramos con el generador de analisis léxico para java, JFlex. Lo primero que hay que hacer es descargar el analizador.



una vez descargado en el computador, procedemos a importar la librería **jflex** en un nuevo proyecto de Netbeans.



Luego procederemos a crear dos archivos uno con extensión .flex, llamado Lexer.flex y otro con extensión .java al cual le pondremos como nombre Tokens.java

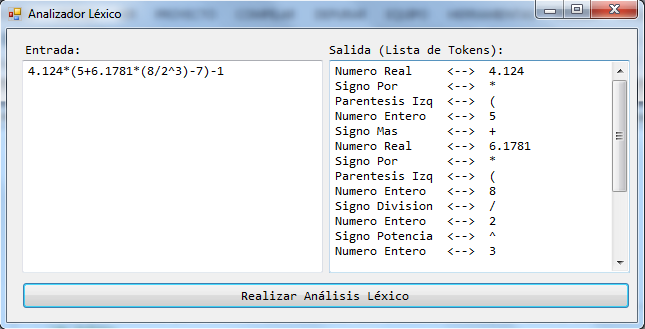


Procederemos a crear nuestras propias reglas léxicas dentro del archivo “Lexer.flex” e ir añadiendo todos sus tokens en “Tokens.java” dentro de un enum.

A continuación crearemos un Frame con dos JTextarea y un botón.

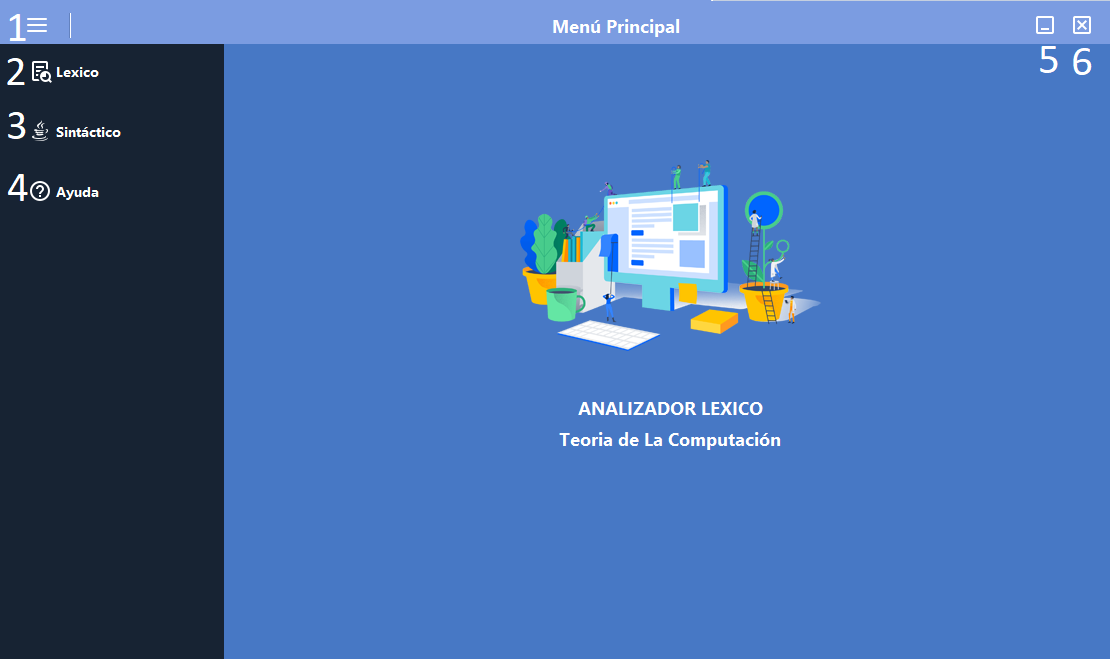


Luego presionamos doble click en el botón creado y dentro del evento click crearemos la conexión entre Java y el analizador lexico y en un select case iremos validando cada tokens creado con la leyenda que le presentaremos al usuario.



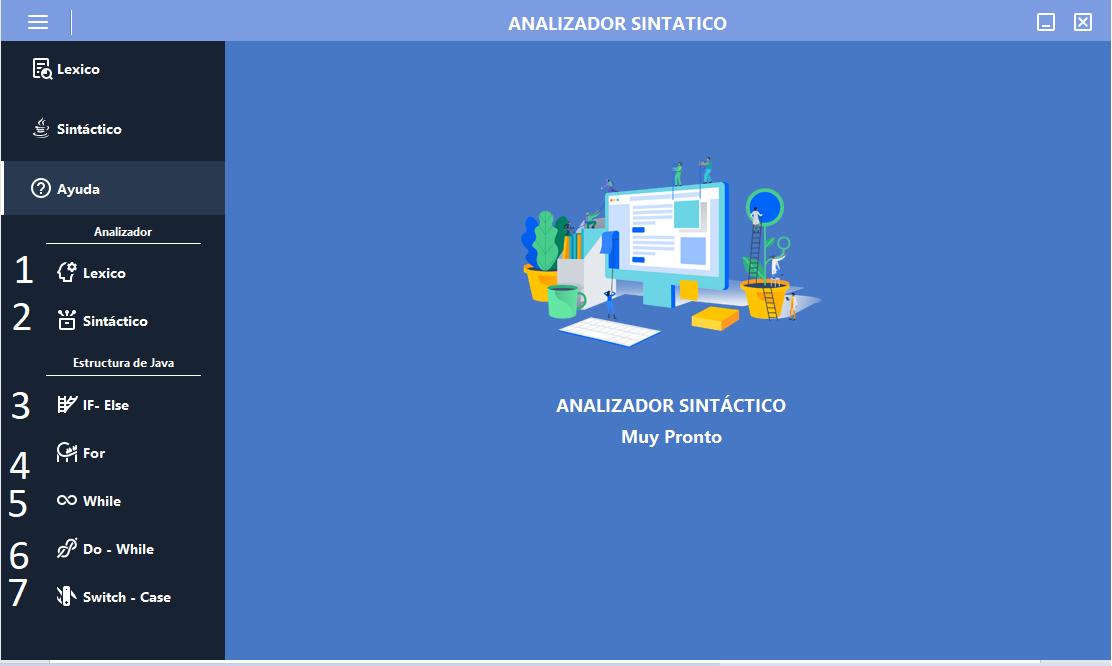
Manual del Usuario 1.2

Pantallas de Inicio

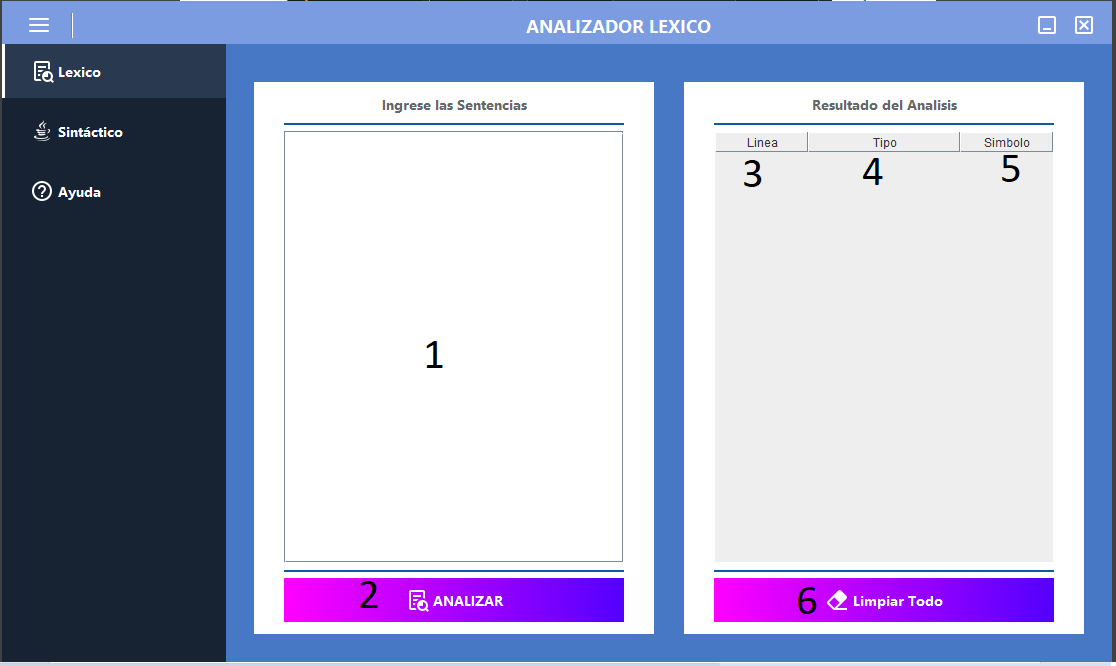


Encontramos la primera pantallas con las siguientes opciones:

1. Retornar al menú principal desde otras pantallas
2. Entrar al analizador léxico
3. Entrar al analizador sintáctico (aun en creación)
4. Ayuda sobre los analizadores y estructura de Java
5. Minimizar
6. Cerrar



1. Referencia al analizador lexico.
2. Referencia al analizador sintactico.
3. En la opción de ayuda nos mostrara el uso de las diferencias sentencias como: If-Else, uso del For, Do While y del Switch-Case.



1. Dentro de ese recuadro se ingresa la sentencia a analizar
2. Analiza la sentencia dentro del recuadro
3. Numero de línea analizada
4. Tipo de carácter analizado
5. Símbolo usado para llevar a cabo la acción
6. Limpiar la tabla de sentencias y datos analizados

# Referencias

Documentos proporcionados por la docente