

Universidad De San Carlos De Guatemala
Facultad De Ingeniería
Escuela de Ciencias Y Sistemas
Lenguajes Formales Y De Programación



Proyecto 1

201908355 – Danny Hugo Bryan Tejaxún Pichiyá

Interfaz

Se utilizó la librería tkinter para crear cada componente de la interfaz como botones, campos de texto, etc.

Tk: Permite crear el formulario base para ubicar los componentes internos de la ventana.

Label: Permite crear las etiquetas para mostrar textos dentro de la ventana.

Text: Permite crear campos de texto para ingresar texto. En este caso para poder escribir los comandos y enviarlos a analizar.

Button: Permite crear botones para realizar eventos o ejecutar funciones al dar click.

Métodos de los Componentes

Cada componente tiene ciertos métodos para ejecutar acciones lógicas o coherentes al componente creado.

title: Dentro del parámetro se envía el texto que se verá en la barra de la ventana.

geometry: En su parámetro recibe una cadena indicando las dimensiones en pixeles de la ventana. Por ejemplo: '200x200'

config: Recibe varios parámetros como color de fondo, color de borde, grosor de borde, altura, anchura, etc.

place: Recibe dos parámetros que son x: para su coordenada en eje x (positivo a la derecha); y x: para su coordenada en eje y (positivo hacia abajo). Se utiliza para definir la posición de cualquier componente.

delete: Borra el contenido textual de un componente. Recibe dos parámetros como cadenas. El primer parámetro es '1.0' que indica el principio de la cadena y el segundo es 'end' que indica el fin de la cadena.

insert: Inserta contenido textual en un componente. Recibe dos parámetros. El primer parámetro es un método de la librería tkinter llamado *INSERT* y el segundo parámetro es la cadena de texto que se quiere enviar al componente.

Clase 'AnalizadorLexico'

Método Constructor

Al instanciar la clase automáticamente se inicializarán algunos atributos de la clase, siendo las siguientes:

- **Lista de Tokens**

La lista se utilizará para guardar tokens encontrados durante el análisis léxico. Los tokens se guardan como objetos.

- **Lista de Errores**

La lista se utilizará para guardar los errores léxicos encontrados durante el análisis léxico. Cada error se guardan como objetos.

- **Línea**

Es un atributo tipo entero. Se utiliza para el conteo de líneas en el archivo de entrada, varía por cada salto de línea encontrado.

- **Columna**

Es un atributo tipo entero. Se utiliza para el conteo de caracteres en el archivo de entrada, varía por cada carácter analizado.

- **Buffer**

Es un atributo tipo cadena. Se utiliza para almacenar los caracteres válidos del lenguaje y se guarda como token al cumplirse ciertas condiciones y se reinicia automáticamente después de guardar un token.

- **Iterador (i)**

Es un atributo tipo entero. Se utiliza para definir una posición en la cadena (lista) de caracteres ingresados para el análisis.

Método 'agregar_Token'

Recibe cuatro parámetros. El primer parámetro es el carácter o cadena de caracteres que se guardara como token, el segundo parámetro es la línea en la que se encontró el token, el tercer parámetro es la columna en la que se encontró el token y el último parámetro es el tipo de token que se guardará.

Los tipos de tokens son: palabras reservadas, los valores (cadenas entre comillas), eventos, signos de mayor y menor que, comillas, comilla simple, coma, dos puntos, corchete izquierdo, corchete derecho.

Método 'agregar_Error'

Recibe tres parámetros. El primer parámetro es el carácter que no está definido dentro del lenguaje objeto y se guardará como error, el segundo parámetro es la línea en la que se encontró el error y el último parámetro es la columna en la que se encontró el carácter inválido o no reconocido por el lenguaje.

Método 'analizar'

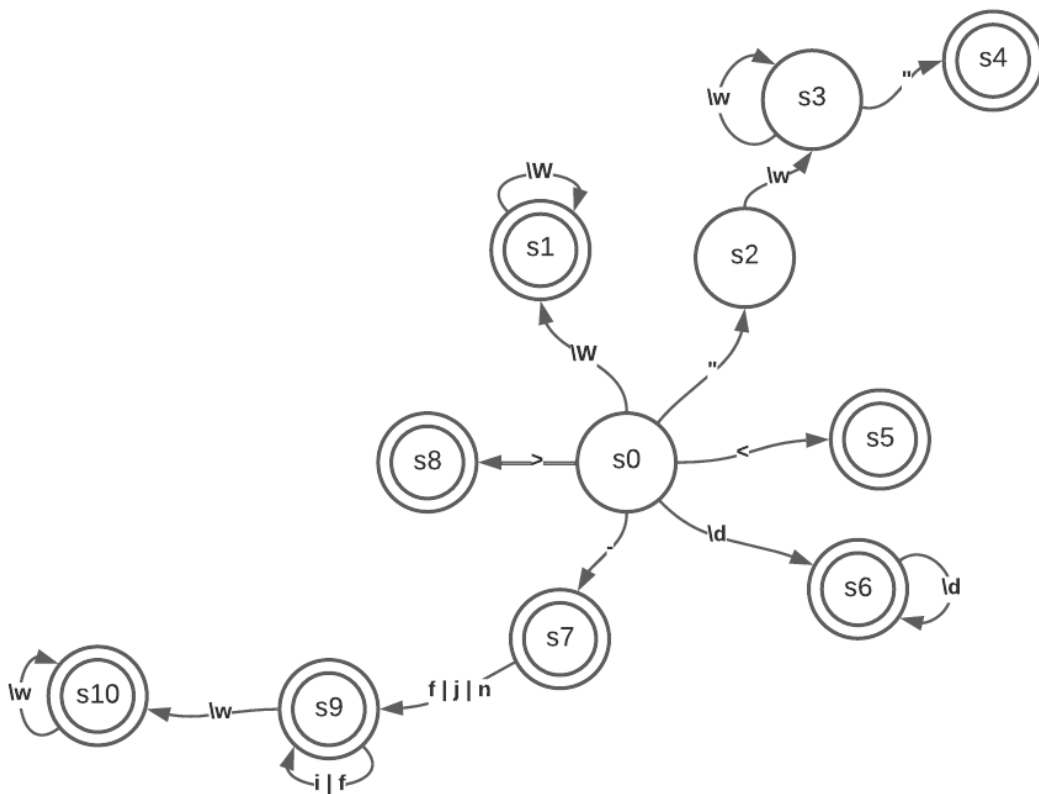
El método recibe como parámetro la cadena que contiene el archivo de entrada para ser analizado. Se concatena un caracter de finalización (\$) para poder completar el análisis. Se itera en los caracteres de la cadena para analizar y se envía caracter por caracter a un estado correspondiente. Inicialmente se envía los caracteres al estado inicial que es el que analiza si el carácter está definido por el lenguaje, y determina el estado para el siguiente caracter, mientras el siguiente caracter sea distinto al caracter de finalización.

El estado s0 es el estado inicial que determina el estado siguiente. El estado s1 determina las palabras reservadas. El estado s2 determina si el carácter es comilla de apertura. El s3 determina si el caracter es numérico o alfabético. El estado s4 determina que el caracter cadena encerrado en comilla doble y se agrega como token. El estado s5 determina si el caracter es el signo menor que. El estado s6 determina si el caracter es un número. El estado s7 determina si el caracter es guion. El estado s8 determina si el caracter es un guion. El estado s9 determina si el buffer es una bandera. El estado s10 determina si el buffer es una cadena alfanumérica.

Expresiones Regulares

Descripción	Patrón	Expresión regular	Ejemplos
Reservada RESULTADO	Palabra RESULTADO	RESULTADO	RESULTADO
Reservada TEMPORADA	Palabra TEMPORADA	TEMPORADA	TEMPORADA
Reservada JORNADA	Palabra JORNADA	JORNADA	JORNADA
Reservada GOLES	Palabra GOLES	GOLES	GOLES
Reservada TABLA	Palabra TABLA	TABLA	TABLA
Reservada PARTIDOS	Palabra PARTIDOS	PARTIDOS	PARTIDOS
Reservada TOP	Palabra TOP	TOP	TOP
Reservada ADIOS	Palabra ADIOS	ADIOS	ADIOS
Reservada VS	Palabra VS	VS	VS
Reservada LOCAL	Palabra LOCAL	LOCAL	LOCAL
Reservada VISITANTE	Palabra VISITANTE	VISITANTE	VISITANTE
Reservada TOTAL	Palabra TOTAL	TOTAL	TOTAL
Reservada SUPERIOR	Palabra SUPERIOR	SUPERIOR	SUPERIOR
Reservada INFERIOR	Palabra INFERIOR	INFERIOR	INFERIOR
Valores asignados	Secuencia de caracteres alfanuméricos	"[A-Za-z][0-9a-za-z]\ "*"	"Sevilla", "Villarreal"
Valores numéricos	Secuencia de caracteres numéricos	[0-9]*	1,32,17,28
Signo menor que	Un caracter '<'	<	<
Signo mayor que	Un caracter '>'	>	>
Guion	Un caracter '-'	-	-

AFD



Gramática Libre de Contexto

S ::= INICIO

INICIO ::= COMANDO

COMANDO ::= RESULTADO | JORNADA | GOLES | TABLA | PARTIDOS | TOP |
pr_ADIOS

RESULTADO ::= pr_RESULTADO nomEquipo pr_VS nomEquipo pr_TEMPORADA
TEMPORADA

JORNADA ::= pr_JORNADA numeroEntero pr_TEMPORADA TEMPORADA BANDERAS

GOLES ::= pr_GOLES CONDICIONEQUIPO pr_TEMPORADA TEMPORADA

TABLA ::= pr_TABLA pr_TEMPORADA TEMPORADA BANDERAS

PARTIDOS ::= pr_PARTIDOS nomEquipo pr_TEMPORADA TEMPORADA BANDERAS

TOP ::= pr_TOP CONDICIONTOP pr_TEMPORADA TEMPORADA BANDERAS

TEMPORADA ::= menorQue año guion año mayorQue

BANDERAS ::= BANDERAS BANDERA valor

BANDERAS ::= BANDERA valor | cadena_vacia

BANDERA ::= -f | -ji | -jf | -n

CONDICIONEQUIPO ::= pr_LOCAL | pr_VISITANTE | pr_TOTAL

CONDICIONTOP ::= pr_SUPERIOR | pr_INFERIOR