

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 ver y poder modificar el texto que se genera al elegir un archivo de entrada.

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

**Combobox:** Permite crear una caja de opciones desplegables y ejecutar un evento al seleccionar una opción.

**askopenfilename:** Permite abrir una ventana en la que se puede elegir un archivo para guardar su ubicación. Puede especificarse una extensión o bien puede abrirse todo tipo de archivos con cualquier extensión.

## 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 y: para su coordenada en eje y (positivo hacia abajo). Se utiliza para definir la posición de cualquier componente.

**bind:** Permite asignar un evento a un componente. Recibe dos parámetros. El primer parámetro es una cadena de texto que especifica el componente al que se asignará el evento dentro de dos pares de signos mayor y menor que, por ejemplo '<<ComboboxSelected>>'. El segundo parámetro es la función que se ejecutará durante el evento, a la función que se ejecutará debe de recibir el parámetro *event*, no importa si es utilizado, por ejemplo *funcion(self,event)....*; la llamada a la función debe ser realizada únicamente escribiendo el nombre de la función, por ejemplo *self.funcion*.

**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 los 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.

- **Abierto**

Es un atributo tipo booleano. Se utiliza para diferenciar cadenas de caracteres de palabras reservadas o eventos dentro del archivo de entrada. Su valor cambia conforme se encuentran comillas o comillas simples dentro de la cadena que se está analizando.

### **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 y los eventos posibles para los botones. El estado s2 determina si el caracter es el signo virgulilla. El estado s3 determina si el caracter es el signo menor que. El estado s4 determina si el caracter es el signo mayor que. El estado s5 determina si el caracter es comilla y cambia el valor del atributo abierto al verdadero. El estado s6 determina si el caracter es comilla simple y cambia el valor del atributo abierto a verdadero. El estado s7 determina los valores que tomarán los atributos de los componentes definidos en el archivo de entrada y las cadenas de texto que se mostrarán en el formulario. El estado s8 determina si el caracter es un corchete izquierdo. El estado s9 determina si el caracter es un corchete derecho. El estado s10 determina si el caracter son los dos puntos. El estado s11 determina si el caracter es la coma.

## Expresiones Regulares

| Descripción          | Patrón                                | Expresión regular         | Ejemplos          |
|----------------------|---------------------------------------|---------------------------|-------------------|
| Reservada formulario | Palabra formulario                    | formulario                | formulario        |
| Reservada tipo       | Palabra tipo                          | tipo                      | tipo              |
| Reservada valor      | Palabra valor                         | valor                     | valor             |
| Reservada fondo      | Palabra fondo                         | fondo                     | fondo             |
| Reservada valores    | Palabra valores                       | valores                   | valores           |
| Reservada evento     | Palabra evento                        | evento                    | evento            |
| Valores asignados    | Secuencia de caracteres alfanuméricos | [A-Za-z][0-9a-za-z]\s\.-* | etiqueta, Cadena1 |
| Signo virgulilla     | Un caracter '~'                       | ~'                        | ~                 |
| Signo menor que      | Un caracter '<'                       | <'                        | <                 |
| Signo mayor que      | Un caracter '>'                       | >'                        | >                 |
| Corchete izquierdo   | Un caracter '['                       | ['                        | [                 |
| Corchete derecho     | Un caracter ']'                       | ']'                       | ]                 |
| Dos puntos           | Un caracter ':'                       | :'                        | ;                 |
| Coma                 | Un caracter ','                       | ','                       | ,                 |
| Comillas             | Un caracter '"'                       | ""'                       | "                 |
| Comilla simple       | Un caracter ''                        | '''                       | '                 |

## AFD

