



Manual Técnico

P R O Y E C T O 1

**ORGANIZACIÓN DE LENGUAJES Y
COMPILADORES 1**

**FABIO ANDRE SANCHEZ CHAVEZ
201709075**

Especificaciones Técnicas

Compilador

gcc (Fedora 21) 4.9.2

Copyright (C) 2015 Free Software Foundation, Inc.

Computadora

HP NoteBook 14

RAM: 4 GB o superior

Disco Duro: 320 GB o superior

Procesador: Intel® Core™ i5-4210U o superior

Ficheros

- primero.py

Librerías

- Tikinter
- Graphviz

Variables globales

```
archivos=""
texto=""
cadena=""
cadenas=""
contVar=0
ID=0
noComentario=0
noCadena=0
nombre=""
nombres=""
extension=""
estados=1
rees=""
patsi=""
errores=""
pintarComentario=""
pintarComentarioL=""
pintarCadena=""
rmt=""
```

MÉTODOS USADOS

```
def __init__(self, window):
```

Inicializa los componentes de la ventana creada con la libreria Tkinter, entre los cuales botones abrir guardar y guardar como

```
def nuevof(self):
```

Funcion que limpia los text para realizar varios analisis

```
def guardarArchivo(self):
```

Funcion que guarda el texto mostrado en el text de entrada en el archivo abierto

```
def guardarComof(self):
```

Funcion que guarda el texto mostrado en el text de entrada en un nuevo archivo

```
def fileMenu(self):
```

Funcion que abre un archivo y analiza el nombre y su extension y recupera el contenido del archivo en el text de entrada

```
def analizarRMT(self):
```

Funcion que analiza un texto en el text de entrada correspondiente al archivo rmt, donde verifica el correcto orden de los lexemas de una funcion aritmética, enviando el resultado a la funcion reporteRMT

```
def reporteRMT(self):
```

Funcion que realiza el reporte analizado por la funcion analizarRMT en un html de salida

```
def analizarHTML(self):
```

Funcion que analiza un texto en el text de entrada correspondiente al archivo HTML, donde verifica el correcto orden de los lexemas de un archivo html, enviando los errores a un archivo HTML

```
def analizarCSS(self):
```

Funcion que analiza un texto en el text de entrada correspondiente al archivo CSS, donde verifica el correcto orden de los lexemas de un archivo CSS, enviando los errores a un archivo HTML

```
def reescribir(self, letra):
```

Reescribe el texto analizado de cada archivo en un nuevo archivo sin errores

```
def crear(self):
```

crea el archivo de reportes y los nuevos archivos analizados

```
def erroresHTML(self):
```

Crea el archive de reportes de errores

```
def analizarScript(self):
```

Funcion que analiza un texto en el text de entrada correspondiente al archivo JS, donde verifica el correcto orden de los lexemas de un archivo JS, enviando los errores a un archivo HTML

```
def find(self, buscar, color):
```

recupera el texto de entrada y pinta las palabras reservadas, variables y demás tokens del archivo

```
def expreionJS(self, expresion):
```

Recupera las expresiones reconocidas de un archivo de entrada

```
def reservadas(self, reservada):
```

Contiene las palabras reservadas del lenguaje javascript y las envia para pintar en la entrada

```
def reservadasHTML(self, reservada):
```

Contiene las palabras reservadas del lenguaje javascript y las envia para pintar en la entrada

```
def reservadasCSS(self, reservada):
```

Contiene las palabras reservadas del lenguaje javascript y las envia para pintar en la entrada

def printSalida(self):

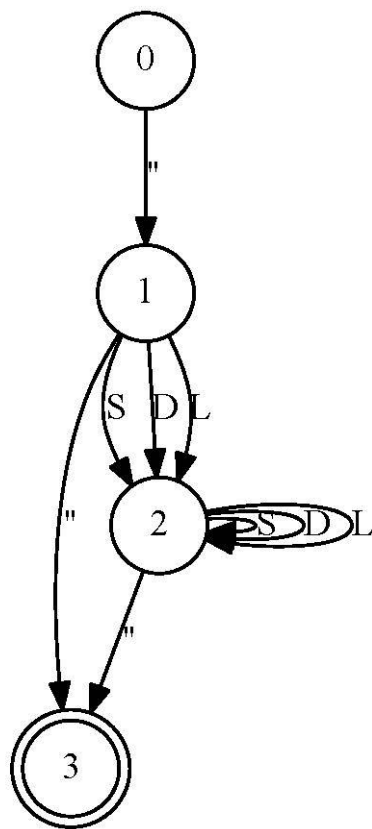
ejecuta funciones al finalizar el analisis

def terminar(self):

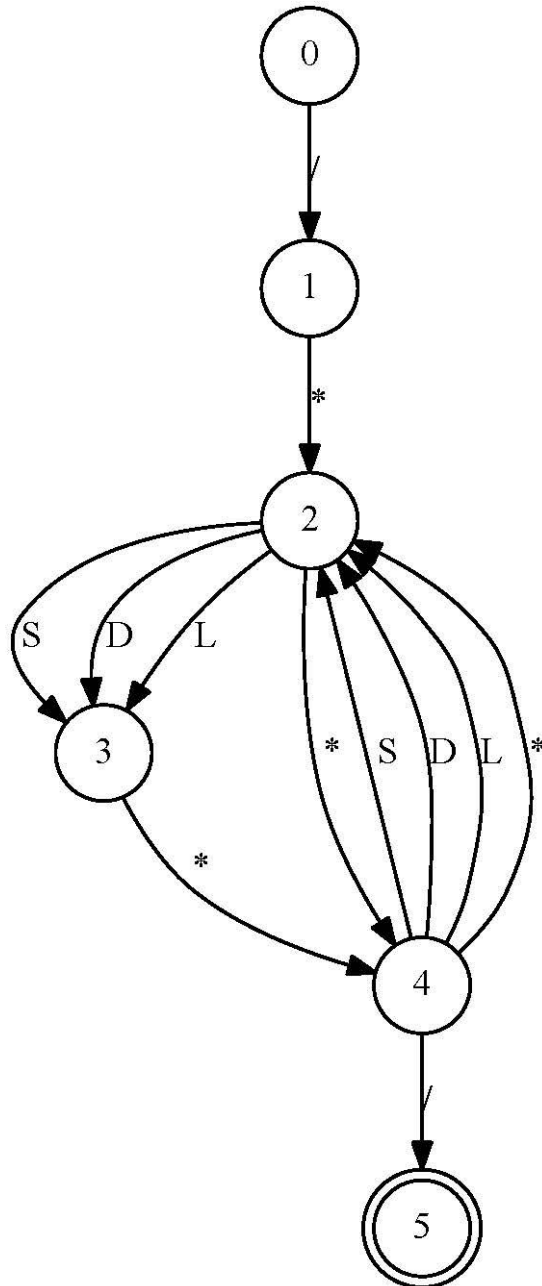
Cierra la ventana creada con la librería tkinder

Expresiones reconocidas:

Expresion: " (S|D|L)* " Reconocida



Expresion: "/" "*" (L|D|S)* (*)+ "/" Reconocida



Expresion: "VAR" [ID]=([ID]|["cadena"]|[num]) Reconocida

