



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS

LFP

MANUAL TECNICO

Nombre: Angel Francisco Sique Santos

Carnet:

2	0	2	0	–	1	2	0	3	9
---	---	---	---	---	---	---	---	---	---

Contenido

Contenido

Introducción	3
1. Objetivo.....	3
2. Requisitos del programa.	3
Opciones del programa	4
Pestaña principal	4
Archivo	4
Analizador léxico	4
Analizador sintáctico	4
Comandos.....	5
Realizar acción.....	6
Generar HTML	6
Expresión regular	6
AFD	7
Gramática libre de contexto.....	8

Introducción

La Liga Bot es un robot creado para facilitar la manipulación de una gran base de datos de partidos, de modo que hace más amigable buscar y filtrar entre estos datos.

1. Objetivo.

Crear un robot para una manipulación de datos sobre todos los partidos de La Liga, torneo español de fútbol más sencilla y amigable a la vista

2. Requisitos del programa.

- 200mb de disco duro.
- 512mb de RAM
- Windows 7
- Java 8
- Equipo Intel Pentium o superior

Opciones del programa

Pestaña principal

Para crear la interfaz se usó la librería PySimpleGUI.

```
import PySimpleGUI as sg
```

```
errores =
sg.theme('DarkBlue2')

layout = [
    [sg.Output(size=(110, 20), font=('Helvetica 10'),text_color='white'),[sg.Button('REPORTE DE \nERRORES', button_color=('DarkTurquoise', 'Black'))
    ,sg.Button('LIMPIAR LOG\n DE ERRORES', button_color=('DarkTurquoise', 'Black')),sg.Button('REPORTE DE \nTOKENS', button_color=('DarkTurquoise', 'Black'))
    ,sg.Button('LIMPIAR LOG\n DE TOKENS', button_color=('DarkTurquoise', 'Black')),sg.Button('MANUAL DE\n USUARIO', button_color=('DarkTurquoise', 'Black'))
    ,sg.Button('MANUAL TECNICO', button_color=('DarkTurquoise', 'Black'))]],
    [sg.Multiline(size=(70, 5), enter_submits=False, key='-QUERY-', do_not_clear=False,text_color='white'),
    sg.Button('ENVIAR', button_color=('DarkTurquoise', 'Black'), bind_return_key=True,expand_x=True,expand_y=True)]]

window = sg.Window('La Liga Bot', layout, font=('Helvetica', ' 13'), default_button_element_size=(8,2), use_default_focus=False)
```

Se crea un layout donde se define el tamaño de la ventana y de los botones que incluye.

Archivo

La librería Pandas facilita la lectura del csv donde están todos los datos, solo se debe poner la ruta y ya se tiene acceso a todos los datos.

```
datosPartidos = pd.read_csv("C:/Users/Angel/Desktop/VSCode/Carpeta para Github/[LFP]Proyecto2_202012039/LFP_PY2_202012039/LaLigaBot-LFP.csv")
```

Analizador léxico

Para el analizador léxico se hicieron 5 estados.

```
def analizar(self, cadena):
    cadena = cadena + '$'
    self.listaErrores = []
    self.listaTokens = []
    self.i = 0
    while self.i < len(cadena):
        if self.estado == 0:
            self.s0(cadena[self.i],cadena)
        elif self.estado == 1:
            self.s1(cadena[self.i],cadena)
        elif self.estado == 2:
            self.s2(cadena)
        elif self.estado == 4:
            self.s4(cadena)
        elif self.estado == 6:
            self.s6(cadena[self.i])

        self.i += 1
```

Analizador sintáctico

Para el analizador sintáctico, estas son las palabras reservadas que definen qué hacer.

```
def INICIO(self):
    # Observar el primer elemento para
    # decidir a donde ir
    temporal = self.observarToken()
    if temporal is None: ...
    elif temporal.tipo == 'reservada_RESULTADO': ...
    elif temporal.tipo == 'reservada_JORNADA': ...
    elif temporal.tipo == 'reservada_PARTIDOS': ...
    elif temporal.tipo == 'reservada_TABLA': ...
    elif temporal.tipo == 'reservada_TOP': ...
    elif temporal.tipo == 'reservada_GOLES': ...
    elif temporal.tipo == 'reservada_ADIOS': ...
    else: ...
```

Comandos

Cuando se reconoce una palabra reservada se llama una función a corda a la palabra reservada encontrada. La manera en que se valida que el comando esté bien escrito es con unos if donde se va analizando token por token hasta encontrar todos los tokens en el orden que tiene que estar con los tokens que tenga que tener.

```
def resultadoDePartido(self):
    # Comando para obtener resultado de un partido

    # Sacar token --- se espera reservada_RESULTADO
    token = self.sacarToken()
    if token.tipo == 'reservada_RESULTADO':
        # Sacar otro token --- se espera cadena
        token = self.sacarToken()
        if token is None: ...
        elif token.tipo == "cadena":
            # Se guarda el equipo
            equipo2 = token.lexema
            # Sacar otro token --- se espera reservada_VS
            token = self.sacarToken()
            if token is None: ...
            elif token.tipo == "reservada_VS":
                # Sacar otro token --- se espera cadena
                token = self.sacarToken()
                if token is None: ...
                elif token.tipo == "cadena":
                    # Se guarda el equipo
                    equipo1 = token.lexema
                    # Sacar otro token --- se espera reservada_TEMPORADA
                    token = self.sacarToken()
                    if token is None: ...
                    elif token.tipo == "reservada_TEMPORADA":
                        # Sacar otro token --- se espera menorQue
                        token = self.sacarToken()
                        if token is None: ...
                        elif token.tipo == "menorQue":
                            # Sacar otro token --- se espera año
                            token = self.sacarToken()
```

```
# Sacar otro token --- se espera año
token = self.sacarToken()
if token is None: ...
elif token.tipo == "numero":
    # Se guarda el primer año
    fecha1 = token.lexema
    # Sacar otro token --- se espera guion
    token = self.sacarToken()
    if token is None: ...
    elif token.tipo == "guion":
        # Sacar otro token --- se espera guion
        token = self.sacarToken()
        if token is None: ...
        elif token.tipo == "numero":
            # Se guarda el primer año
            fecha2 = token.lexema
            # Sacar otro token --- se espera mayorQue
            token = self.sacarToken()
            if token is None: ...
            elif token.tipo == "mayorQue": ...
            else: ...
```

Realizar acción

Al verificar que todo está bien, se llama una función que realiza la acción que se define en el análisis sintáctico.

```
def obtenerResultadoDePartido(equipo1,equipo2,fecha1,fecha2):
    datos = (datosPartidos[(datosPartidos['Equipo1'] == equipo1) & (datosPartidos['Equipo2'] == equipo2) & (datosPartidos['Fecha'] == fecha1) & (datosPartidos['Fecha'] == fecha2)])
    tmpEquipo1 = datos.iloc[0,3]
    tmpEquipo2 = datos.iloc[0,4]
    tmpGol1 = datos.iloc[0,5]
    tmpGol2 = datos.iloc[0,6]
    print('--->El resultado de este partido fue: {} {} - {} {}'.format(tmpEquipo1,tmpGol1,tmpEquipo2,tmpGol2))
```

Generar HTML

Todos los HTML se generan con la misma función, donde se le envía la tabla en HTML y el nombre del archivo.

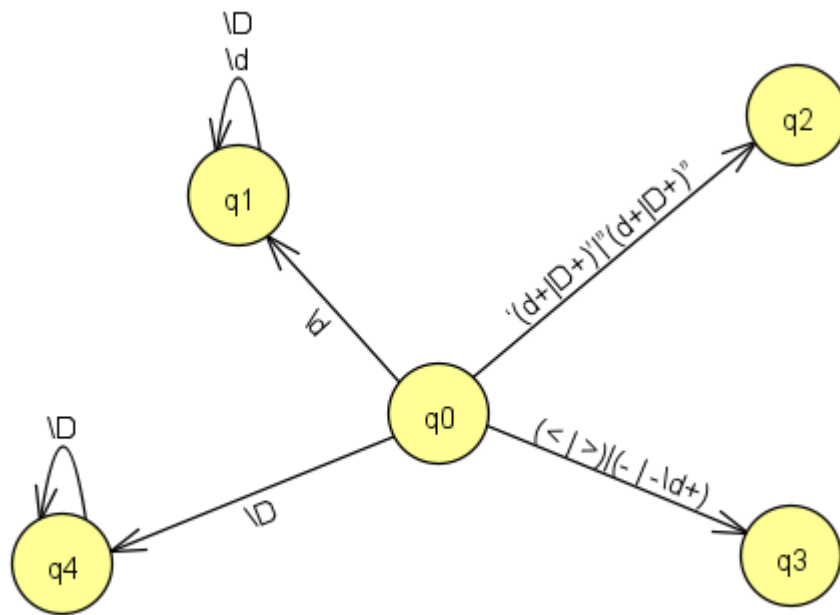
```
def imprimir(archivo,tabela):
    """Imprime una tabla en HTML"""
    strARCHIVO = open(archivo+'.html','w')
    strHTML = '''<!DOCTYPE html>
    <html>
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    <title>''' + archivo + '''</title>
    </head>
    <style>
    table, th, td {
    border: 1px solid rgb(31, 31, 31);
    border-collapse: collapse;
    background-color: #000000;
    color: #00CED1;
    }
    th:nth-child(even),td:nth-child(even) {
    background-color: #494949;
    }
    </style>
    <body bgcolor="DarkSlateGray">
    <center>
    '''
    strHTML += tabela
    strHTML += '''
    </center>
    </body>
    </html>'''
    strARCHIVO.write(strHTML)
    strARCHIVO.close()
    webbrowser.open(archivo+'.html')
```

Expresión regular

El analizador tiene 5 estados.

- En el estado 1 la expresión es: `\d+`
- En el estado 2 la expresión es: `'(d+|D+)'|'(d+|D+)'`
- En el estado 3 la expresión es: `\D+`
- En el estado 4 la expresión es: `(< | >)|(- | -\d+)`

AFD



Gramática libre de contexto

S ::= COMANDO

COMANDO ::= LISTADECOMANDOS

LISTADECOMANDOS ::= PR_RESULTADO EQUIPO PR_VS EQUIPO PR_TEMPORADA AÑO

LISTADECOMANDOS ::= PR_JORNADA DOS_DIGITOS PR_TEMPORADA AÑO

LISTADECOMANDOS ::= PR_JORNADA DOS_DIGITOS PR_TEMPORADA AÑO ARCHIVO

LISTADECOMANDOS ::= PR_GOLES CONDICION_EQUIPO EQUIPO PR_TEMPORADA AÑO

LISTADECOMANDOS ::= PR_TABLA PR_TEMPORADA AÑO ARCHIVO

LISTADECOMANDOS ::= PR_TABLA PR_TEMPORADA AÑO

LISTADECOMANDOS ::= PR_PARTIDOS EQUIPO PR_TEMPORADA AÑO LISTAOPCIONALES

LISTADECOMANDOS ::= PR_TOP CONDICION_TOP PR_TEMPORADA AÑO

LISTADECOMANDOS ::= PR_TOP CONDICION_TOP PR_TEMPORADA AÑO CANTIDAD_TOP

LISTADECOMANDOS ::= PR_ADIOS

EQUIPO ::= comilla cadena comilla

AÑO ::= menorQue CUATRO_DIGITOS guion CUATRO_DIGITOS mayorQue

CUATRO_DIGITOS ::= numero_de_cuatro_digitos

DOS_DIGITOS ::= numero_de_dos_digitos

CONDICION_EQUIPO ::= LISTACONDICION_EQUIPO

LISTACONDICION_EQUIPO ::= PR_LOCAL

LISTACONDICION_EQUIPO ::= PR_VISITANTE

LISTACONDICION_EQUIPO ::= PR_TOTAL

ARCHIVO ::= -f cadena

LISTAOPCIONALES ::= ARCHIVO LISTAOPCIONALES

LISTAOPCIONALES ::= -ji numero LISTAOPCIONALES

LISTAOPCIONALES ::= -jf numero LISTAOPCIONALES

LISTAOPCIONALES ::= vacio

CONDICION_TOP ::= PR_SUPERIOR

CONDICION_TOP ::= PR_INFERIOR

CANTIDAD_TOP ::= -n numero