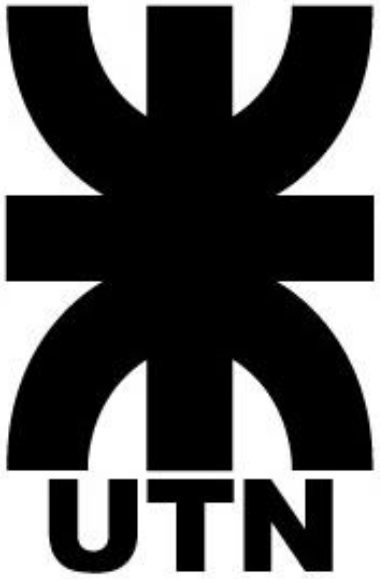

SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

TRABAJO PRÁCTICO INTEGRADOR

Diseño e implementación de Lexer y Parser y Traductor de Lenguaje RSS

Grupo: N.º 4

Integrantes:

- AGUIRRE, Camilo
- BIANCIOTTO, Joaquín
- COLOMBO, Matías Julián
- MARAIN, Yoel Mario

Carrera: Ingeniería en Sistemas de Información

Comisión: ISI A

Primer Cuatrimestre

Curso Académico: 2023

UNIVERSIDAD TECNOLOGICA NACIONAL

FACULTAD REGIONAL RESISTENCIA

Fecha y Lugar de presentación: 04/06/2023. Resistencia, Chaco

ÍNDICE

1. INTRODUCCION:

Un analizador léxico (o *lexer*) es una parte esencial de un compilador o intérprete que se encarga de descomponer el código fuente en una secuencia de elementos más pequeños llamados *tokens*. Estos tokens son unidades léxicas que representan los componentes individuales del lenguaje de programación, como palabras clave, identificadores, operadores, números y símbolos.

El lexer toma el código fuente como entrada y realiza un escaneo caracter por caracter, identificando y clasificando los diferentes elementos léxicos. Utiliza reglas definidas previamente para reconocer patrones y formar los tokens correspondientes.

Para la realización de este trabajo, optamos por utilizar **Python** debido a las siguientes razones:

- **Sintaxis clara y legible:** Python se destaca por su sintaxis simple y fácil de leer, lo que facilita la comprensión y escritura de código, manteniendo también un código más limpio y organizado.
- **Aprendizaje eficiente:** Python tiene una curva de aprendizaje suave y cuenta con una gran comunidad, lo que facilita la obtención de recursos de aprendizaje en línea y documentación clara.
- **Amplia disponibilidad de bibliotecas y módulos:** Python cuenta con una gran cantidad de bibliotecas y módulos disponibles que facilitan la tarea de implementar funcionalidades avanzadas.

Llevamos a cabo el proyecto en una plataforma web de desarrollo colaborativo llamada **GitHub**, esta proporciona control automático de versiones, lo que permite realizar un seguimiento de los cambios realizados en el proyecto a lo largo del tiempo, facilita la colaboración en equipo, ofrece herramientas de seguimiento de problemas y solicitudes de extracción.

2. LEXER:

2.1 Módulos utilizados:

PLY (Python Lex-Yacc) es una biblioteca de análisis léxico y sintáctico. Proporciona las herramientas necesarias para construir analizadores personalizados basados en las técnicas de análisis léxico y sintáctico LEX y YACC utilizadas tradicionalmente en otros lenguajes. Sin embargo, a diferencia de LEX y YACC, que están escritos en C, PLY está escrito en Python y aprovecha las características del lenguaje y la facilidad de uso que ofrece.

Nosotros utilizamos el módulo 'ply.lex', el cual proporciona herramientas necesarias para definir y ejecutar reglas de análisis léxico, es decir, para reconocer tokens en un flujo de texto.

Para esto, lo importamos, de la siguiente forma:

```
import ply.lex as lex
```

Además, importamos el módulo 're' para realizar coincidencias de expresiones regulares.

```
import re
```

El módulo 'codecs', para trabajar con codificaciones de caracteres.

```
import codecs
```

El módulo 'os', para realizar operaciones relacionadas con el sistema operativo.

```
import os
```

Y el módulo 'sys', para acceder a funcionalidades específicas del intérprete de Python.

```
import sys
```

2.2 Listas:

Definimos 2 listas importantes, la primera será una lista vacía llamada 'error_caracter_ilegal' para almacenar caracteres ilegales encontrados durante el análisis léxico.

```
error_caracter_ilegal=[]
```

Además, definimos una lista llamada 'tokens' que contiene los nombres de los tokens reconocidos por el analizador léxico.

2.2.1 Definición de tokens:

```
tokens = [ 'DT1','DT2','APERTURA_ARTICLE', 'CIERRE_ARTICLE' , 'APERTURA_PARA',
'CIERRE_PARA', 'TEXTO', 'APERTURA_INFO' , 'CIERRE_INFO' , 'APERTURA_TITLE' ,
'CIERRE_TITLE' , 'APERTURA_ITEMIZEDLIST', 'CIERRE_ITEMIZEDLIST',
'APERTURA_IMPORTANT' , 'CIERRE_IMPORTANT' , 'APERTURA_SIMPARA', 'CIERRE_SIMPARA'
, 'APERTURA_ADDRESS' , 'CIERRE_ADDRESS' , 'APERTURA_MEDIAOBJECT' ,
'CIERRE_MEDIAOBJECT' , 'APERTURA_INFORMALTABLE' , 'CIERRE_INFORMALTABLE' ,
'APERTURA_COMMENT' , 'CIERRE_COMMENT' , 'APERTURA_ABSTRACT' , 'CIERRE_ABSTRACT'
, 'APERTURA_SECTION' , 'CIERRE_SECTION' , 'APERTURA_SIMPLESECT' ,
'CIERRE_SIMPLESECT' , 'APERTURA_EMPHASIS' , 'CIERRE_EMPHASIS' , 'APERTURA_LINK'
, 'CIERRE_LINK' ,  'APERTURA_FIRSTNAME' , 'CIERRE_FIRSTNAME' ,
'APERTURA_SURNAME' , 'CIERRE_SURNAME' , 'APERTURA_STREET' , 'CIERRE_STREET' ,
'APERTURA_CITY' , 'CIERRE_CITY' , 'APERTURA_STATE' , 'CIERRE_STATE' ,
```

```
'APERTURA_PHONE' , 'CIERRE_PHONE' , 'APERTURA_EMAIL' , 'CIERRE_EMAIL' ,
'APERTURA_DATE' , 'CIERRE_DATE' , 'APERTURA_YEAR' , 'CIERRE_YEAR' ,
'APERTURA_HOLDER' , 'CIERRE_HOLDER', 'APERTURA_IMAGEDATA' ,
'APERTURA_VIDEOOBJECT' , 'CIERRE_VIDEOOBJECT' , 'APERTURA_IMAGENOBJECT' ,
'CIERRE_IMAGENOBJECT' , 'APERTURA_VIDEODATA', 'APERTURA_LISTITEM' ,
'CIERRE_LISTITEM' , 'APERTURA_TGROUP' , 'CIERRE_TGROUP' , 'APERTURA_THEAD' ,
'CIERRE_THEAD' , 'APERTURA_TFOOT' , 'CIERRE_TFOOT' , 'APERTURA_TBODY' ,
'CIERRE_TBODY' , 'APERTURA_ROW' , 'CIERRE_ROW' , 'APERTURA_ENTRY' ,
'CIERRE_ENTRY' , 'APERTURA_ENTRYTBL' , 'CIERRE_ENTRYTBL' , 'APERTURA_AUTHOR' ,
'CIERRE_AUTHOR' , 'ERROR_1' , 'ERROR_2' , 'ERROR_3' , 'newline' ]
```

2.3 Expresiones regulares:

Ahora planteamos las expresiones regulares de cada token definido en la lista anterior mediante funciones. Estas funciones se ejecutan cuando se encuentra una coincidenciay las utilizamos para realizar operaciones de análisis léxico.

```
t_APERTURA_ARTICLE = r'<article>'
t_CIERRE_ARTICLE = r'</article>'
t_APERTURA_SIMPARA = r'<simpara>'
t_CIERRE_SIMPARA = r'</simpara>'
t_APERTURA_ADDRESS = r'<address>'
t_CIERRE_ADDRESS = r'</address>'
t_APERTURA_MEDIAOBJECT = r'<mediaobject>'
t_CIERRE_MEDIAOBJECT = r'</mediaobject>'
t_APERTURA_COMMENT = r'<comment>'
t_CIERRE_COMMENT = r'</comment>'
t_APERTURA_ABSTRACT = r'<abstract>'
t_CIERRE_ABSTRACT = r'</abstract>'
t_APERTURA_SECTION = r'<section>'
t_CIERRE_SECTION = r'</section>'
t_APERTURA_SIMPLESECT = r'<simplesect>'
t_CIERRE_SIMPLESECT = r'</simplesect>'
t_APERTURA_EMPHASIS = r'<emphasis>'
t_CIERRE_EMPHASIS = r'</emphasis>'
t_APERTURA_AUTHOR = r'<author>'
t_CIERRE_AUTHOR = r'</author>'
t_APERTURA_FIRSTNAME = r'<firstname>'
t_CIERRE_FIRSTNAME = r'</firstname>'
t_APERTURA_SURNAME = r'<surname>'
t_CIERRE_SURNAME = r'</surname>'
t_APERTURA_STREET = r'<street>'
t_CIERRE_STREET = r'</street>'
t_APERTURA_CITY = r'<city>'
t_CIERRE_CITY = r'</city>'
t_APERTURA_STATE = r'<state>'
t_CIERRE_STATE = r'</state>'
t_APERTURA_PHONE = r'<phone>'
t_CIERRE_PHONE = r'</phone>'
t_APERTURA_EMAIL = r'<email>'
t_CIERRE_EMAIL = r'</email>'
t_APERTURA_DATE = r'<date>'
t_CIERRE_DATE = r'</date>'
t_APERTURA_YEAR = r'<year>'
t_CIERRE_YEAR = r'</year>'
```

```
t_APERTURA_HOLDER = r'<holder>'
t_CIERRE_HOLDER = r'</holder>'
t_APERTURA_VIDEOOBJECT = r'<videoobject>'
t_CIERRE_VIDEOOBJECT = r'</videoobject>'
t_APERTURA_IMAGENOBJECT = r'<imagenobject>'
t_CIERRE_IMAGENOBJECT = r'</imagenobject>'
t_APERTURA_TGROUP = r'<tgroup>'
t_CIERRE_TGROUP = r'</tgroup>'
t_APERTURA_THEAD = r'<thead>'
t_CIERRE_THEAD = r'</thead>'
t_APERTURA_TFOOT = r'<tfood>'
t_CIERRE_TFOOT = r'</tfood>'
t_APERTURA_TBODY = r'<tbody>'
t_CIERRE_TBODY = r'</tbody>'
t_APERTURA_ENTRYTBL = r'<entrytbl>'
t_CIERRE_ENTRYTBL = r'</entrytbl>'
t_ERROR_1 = r'<[\w]+>'
t_ERROR_2 = r'<[\w]+\s[\w]+=*[\w"]+\s*/*>'
t_ERROR_3 = r'</[\w]+>'
```

2.4 Funciones:

Definimos una función 't_ignore' que especifica los caracteres que deben ser ignorados por el lexer, como espacios en blanco o tabulaciones.

```
t_ignore = ' \t'
```

Definimos una función 't_error' que maneja los errores de caracteres no reconocidos.

```
def t_error(t):
  print("caracter ilegal %s" % t.value[0])
  t.lexer.skip(1)
```

Definimos una función 't_newline' que cuenta el número de nuevas líneas en el código fuente para realizar un seguimiento de los números de línea.

```
def t_newline(t):
  r'\n+'
  t.lexer.lineno += len(t.value)
```

Creamos el lexer llamando a la función lex.lex(). Esto inicializa el lexer con las reglas y funciones definidas previamente.

```
lexer = lex.lex()
```

3. Conversión a HTML:

Otra de las funcionalidades que tiene nuestro trabajo es la de traducir el documento, generando un archivo de texto HTML, transformando algunas etiquetas XML en etiquetas HTML.

Apertura de archivo:

Primero establecimos los indicadores o flags para abrir un archivo en el modo deseado.

```
flags = os.O_RDWR | os.O_CREAT
```

- **os.O_RDWR:** Este flag indica que el archivo se abrirá en modo lectura y escritura. Permite tanto leer como escribir en el archivo.
- **os.O_CREAT:** Este flag indica que se creará el archivo si no existe. Si el archivo ya existe, este flag no tiene ningún efecto.

Utilizamos el operador de bits OR (|) para combinar ambos flags en un solo valor. Al hacer esto, se obtiene un valor que puede pasarse como argumento al abrir el archivo, indicando que se desea abrir el archivo en modo lectura/escritura y crearlo si no existe.

Luego, abrimos un archivo llamado 'archivo.html' en modo escritura y codificado en UTF-8 para almacenar el resultado del análisis léxico.

```
arch= open("src/html_generados/archivo.html","w",flags,encoding="utf-8")
```

Funciones de conversión:

```
def t_DT1(t):
      r'<[!]DOCTYPE\sarticle>'
      arch.write("<!DOCTYPE html>")
def t_TEXTO (t):
    r'[\w._%+?¿¡!()"|°¬$&={}\#@*-]+'  #falta ver caracteres especiales
    arch.write(f'{t.value} ')
    return (t)
def t_APERTURA_PARA(t):
      r'<para>'
      arch.write("<p>")
      return(t)
def t_CIERRE_PARA(t):
      r'</para>'
      arch.write("</p>")
      return(t)
def t_APERTURA_TITLE(t):
      r'<title>'
      arch.write("<h1>")
      return(t)
def t_CIERRE_TITLE(t):
      r'</title>'
      arch.write("</h1>")
      return(t)
def t_APERTURA_INFO(t):
      r'<info>'
      arch.write('<div style="color:white;background-color:green;font-
size:8pts"><p>')
      return(t)
def t_CIERRE_INFO(t):
      r'</info>'
      arch.write('</p></div>')
      return(t)
def t_APERTURA_IMPORTANT(t):
      r'<important>'
      arch.write('<div style="background-color:red;color:white">')
      return(t)
def t_CIERRE_IMPORTANT(t):
      r'</important>'
```

```
      arch.write('</div>')
      return(t)
def t_APERTURA_IMAGEDATA (t):
      r'<imagedata=fileref=""^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      return(t)
def t_APERTURA_VIDEODATA (t):
      r'<videodata=fileref="^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      return(t)
def t_APERTURA_LINK (t):
      r'<link = xlink:href ="^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      arch.write(f'<a href="{t.value}">esto es un link</a>')
      return (t)
def t_APERTURA_INFORMALTABLE(t):
      r'<informaltable>'
      arch.write("<table>")
      return (t)
def t_CIERRE_INFORMALTABLE(t):
      r'</informaltable>'
      arch.write("</table>")
      return (t)
def t_APERTURA_ROW(t):
      r'<row>'
      arch.write("<tr>")
      return (t)
def t_CIERRE_ROW(t):
      r'</row>'
      arch.write("</tr>")
      return (t)
def t_APERTURA_ENTRY(t):
      r'<entry>'
      arch.write("<td>")
      return (t)
def t_CIERRE_ENTRY(t):
      r'</entry>'
      arch.write("</td>")
      return (t)
def t_APERTURA_ITEMIZEDLIST(t):
      r'<itemizedlist>'
      arch.write("<ul>")
      return (t)
def t_CIERRE_ITEMIZEDLIST(t):
      r'</itemizedlist>'
      arch.write("</ul>")
      return (t)
def t_APERTURA_LISTITEM(t):
      r'<listitem>'
      arch.write("<il>")
      return (t)
def t_CIERRE_LISTITEM(t):
      r'</listitem>'
      arch.write("</il>")
      return (t)
```

4. Implementación:

Ahora podemos utilizar el Lexer creado para analizar texto.

Para esto, lo primero que debemos hacer es importar nuestro lexer:

```
from lexer import lexer
```

También importaremos los módulos 'os' y 'sys', que proporcionan funciones relacionadas con el sistema operativo y la funcionalidad del intérprete de Python, respectivamente.

```
import os
import sys
```

Creamos una función llamada 'borrarPantalla', que se utiliza para borrar la salida en la pantalla. La implementación depende del sistema operativo en el que se está ejecutando el programa. En sistemas tipo Unix (como Linux y macOS), se usa el comando **clear** para borrar la pantalla. En sistemas Windows, se usa el comando **cls**.

```
def borrarPantalla():
    if os.name == "posix":
        os.system("clear")
    elif os.name == "ce" or os.name == "nt" or os.name == "dos":
        os.system("cls")
```

Y vamos a darle al usuario la opción de elegir si desea ingresar datos manualmente o si desea analizar un archivo:

```
print("Hola este es el analizador Lexico")
print("1 para ingresar datos a mano\n2 si quiere cargar datos desde un
archivo\n")
op = input()
errores = []
```

4.1 Ingreso manual:

Si el usuario elige la opción "1", se realiza un ingreso manual de datos. El programa solicita al usuario que ingrese el texto a analizar. Luego, se pasa la cadena de entrada al analizador léxico lexer y se generan los tokens correspondientes. Si se detecta un error léxico, se muestra un mensaje de error. Después de cada análisis, se le pregunta al usuario si desea continuar o terminar el programa.

```
if op == "1":
    borrarPantalla()
    while True:
        print("ingrese lo que quiere analizar")
        cadena = input()
        lexer.input(cadena)
        while True:
            tok = lexer.token()
            if not tok:
                break
            if tok.type == "ERROR_1" or tok.type == "ERROR_2" or tok.type ==
"ERROR_3":
                print(f"error lexico en linea {tok.lineno} ({tok.value})")
            else:
```

```
                print(tok)
        print("desea continuar?\n1 para continuar\n0 para terminar")
        eleccion = input()
        if eleccion == "0":
            break
        borrarPantalla()
```

4.2 Ingreso por archivo:

Si el usuario elige la opción "2", el programa realiza un ingreso de datos a través de un archivo. El programa muestra una lista de archivos en el directorio "prueba/" y pide al usuario que elija un archivo para leer. El archivo seleccionado se abre y su contenido se pasa al analizador léxico lexer. Se generan los tokens correspondientes y se muestra la salida. Si se encuentra un error léxico, se muestra un mensaje de error. Además, hay una sección del código que renombra un archivo llamado "archivo.html" a un nombre nuevo basado en el archivo de entrada.

```
elif op == "2":
      n = 0

      ejemplo_dir = 'prueba/'                   #elegir el archivo
      with os.scandir(ejemplo_dir) as ficheros:
            ficheros = [fichero.name for fichero in ficheros if
fichero.is_file()]    #ficheros es una lista con los archivos de la carpeta
prueba
      for i in ficheros:

            print(f"{n+1}: {ficheros[n]}")
            n +=1
      print("elegi el archivo")
      op2 = input()
      if int(op2) <= n:
            ruta = ficheros[int(op2)-1]
            with open(f"prueba/{ruta}","r",encoding="utf-8") as maestro:
                  print(f"hola abri el archivo: {ruta}")
                  lexer.input(maestro.read())
                  while True:
                        tok = lexer.token()

                        if not tok:break
                        print(tok)
                  cambio = ruta.replace(".xml","")

os.rename("src/html_generados/archivo.html",f"src/html_generados/{cambio}.html"
)
      else:
            print("numero invalido")
```

Si ninguna de las opciones anteriores es seleccionada, se muestra el mensaje "volve a empezar".

```
else:
    print("volve a empezar")
```

5. Historial de cambios:

1) Mayo 22

- Primera prueba de código generador de Lexer

lexer.txt

```
import ply.lex as lex
import re
import codecs
import os
import sys


tokens = [ 'APERTURAARTICULO', 'CIERREARTICULO' , 'APERTURAPARRAFO',
'CIERREPARRAFO', 'TEXTO'
  ]

t_ignore = '\t '   #nose que hace pero vi en varios, creo q ignora espacios en
blanco o tabulacion
t_APERTURAARTICULO = r'<article>'
t_CIERREARTICULO = r'</article>'
t_APERTURAPARRAFO = r'<para>'
t_CIERREPARRAFO = r'</para>'



def t_TEXTO (t):
    r'[a-zA-Z][a-zA-Z0-9]*'  #falta ver caraxteres especiales
    return (t)
def t_error(t):
 print ("caracter ilegal '%s'" % t.value[0])
 t.lexer.skip(1)

print ("Hola este es el analizador Lexico")
print ("Ingrese el codigo a analizar \n")
cadena = ''
while True:
 cad = input()
 cadena = cadena+cad+ '\n'
 break
 if not cadena: continue
 print ('\n')
lexer = lex.lex()
#ciclo para mostrar tokens
lexer.input(cadena)
while True:
    tok = lexer.token()
    if not tok : break
    print (tok)
```

2) Mayo 24

- Adición de etiquetas básicas de párrafo

lexer.txt

```
import ply.lex as lex
import re
import codecs
import os
import sys


tokens = [ 'APERTURA_ARTICLE', 'CIERRE_ARTICLE' , 'APERTURA_PARA',
'CIERRE_PARA', 'TEXTO',
          'APERTURA_INFO' , 'CIERRE_INFO' , 'APERTURA_TITLE' , 'CIERRE_TITLE' ,
'APERTURA_ITEMIZEDLIST',
          'CIERRE_ITEMIZEDLIST', 'APERTURA_IMPORTANT' , 'CIERRE_IMPORTANT' ,
'APERTURA_SIMPARA',
          'CIERRE_SIMPARA' , 'APERTURA_ADDRESS' , 'CIERRE_ADDRESS' ,
'APERTURA_MEDIAOBJECT' , 'CIERRE_MEDIAOBJECT' ,
          'APERTURA_INFORMALTABLE' , 'CIERRE_INFORMALTABLE' , 'APERTURA_COMMENT'
, 'CIERRE_COMMENT' ,
          'APERTURA_ABSTRACT' , 'CIERRE_ABSTRACT' , 'APERTURA_SECTION' ,
'CIERRE_SECTION' , 'APERTURA_SIMPLESECT' ,
          'CIERRE_SIMPLESECT' , 'APERTURA_EMPHASIS' , 'CIERRE_EMPHASIS' ,
'APERTURA_LINK' , 'CIERRE_LINK' ,
          'APERTURA_FIRSTNAME' , 'CIERRE_FIRSTNAME' , 'APERTURA_SURNAME' ,
'CIERRE_SURNAME' , 'APERTURA_STREET' ,
          'CIERRE_STREET' , 'APERTURA_CITY' , 'CIERRE_CITY' , 'APERTURA_STATE' ,
'CIERRE_STATE' , 'APERTURA_PHONE' ,
          'CIERRE_PHONE' , 'APERTURA_EMAIL' , 'CIERRE_EMAIL' , 'APERTURA_DATE' ,
'CIERRE_DATE' , 'APERTURA_YEAR' ,
          'CIERRE_YEAR' , 'APERTURA_HOLDER' , 'CIERRE_HOLDER'
  ]

t_ignore = '\t '   #nose que hace pero vi en varios, creo q ignora espacios en
blanco o tabulacion
t_APERTURA_ARTICLE = r'<article>'
t_CIERRE_ARTICLE = r'</article>'
t_APERTURA_PARA = r'<para>'
t_CIERRE_PARA = r'</para>'
t_APERTURA_INFO = r'<info>'
t_CIERRE_INFO = r'</info>'
t_APERTURA_TITLE = r'<title>'
t_CIERRE_TITLE = r'</title>'
t_APERTURA_ITEMIZEDLIST = r'<itemizedlist>'
t_CIERRE_ITEMIZEDLIST = r'</itemizedlist>'
t_APERTURA_IMPORTANT = r'<important>'
t_CIERRE_IMPORTANT = r'</important>'
t_APERTURA_SIMPARA = r'<simpara>'
t_CIERRE_SIMPARA = r'</simpara>'
t_APERTURA_ADDRESS = r'<address>'
t_CIERRE_ADDRESS = r'</address>'
t_APERTURA_MEDIAOBJECT = r'<mediaobject>'
t_CIERRE_MEDIAOBJECT = r'</mediaobject>'
t_APERTURA_INFORMALTABLE = r'<informaltable>'
t_CIERRE_INFORMALTABLE = r'</informaltable>'
t_APERTURA_COMMENT = r'<comment>'
```

```
t_CIERRE_COMMENT = r'</comment>'
t_APERTURA_ABSTRACT = r'<abstract>'
t_CIERRE_ABSTRACT = r'</abstract>'
t_APERTURA_SECTION = r'<section>'
t_CIERRE_SECTION = r'</section>'
t_APERTURA_SIMPLESECT = r'<simplesect>'
t_CIERRE_SIMPLESECT = r'</simplesect>'
t_APERTURA_EMPHASIS = r'<emphasis>'
t_CIERRE_EMPHASIS = r'</emphasis>'
t_APERTURA_LINK = r'<link>'
t_CIERRE_LINK = r'</link>'
t_APERTURA_FIRSTNAME = r'<firstname>'
t_CIERRE_FIRSTNAME = r'</firstname>'
t_APERTURA_SURNAME = r'<surname>'
t_CIERRE_SURNAME = r'</surname>'
t_APERTURA_STREET = r'<street>'
t_CIERRE_STREET = r'</street>'
t_APERTURA_CITY = r'<city>'
t_CIERRE_CITY = r'</city>'
t_APERTURA_STATE = r'<state>'
t_CIERRE_STATE = r'</state>'
t_APERTURA_PHONE = r'<phone>'
t_CIERRE_PHONE = r'</phone>'
t_APERTURA_EMAIL = r'<email>'
t_CIERRE_EMAIL = r'</email>'
t_APERTURA_DATE = r'<date>'
t_CIERRE_DATE = r'</date>'
t_APERTURA_YEAR = r'<year>'
t_CIERRE_YEAR = r'</year>'
t_APERTURA_HOLDER = r'<holder>'
t_CIERRE_HOLDER = r'</holder>'


def t_TEXTO (t):
    r'[a-zA-Z][a-zA-Z0-9]*'  #falta ver caraxteres especiales
    return (t)
def t_error(t):
 print ("caracter ilegal '%s'" % t.value[0])
 t.lexer.skip(1)

print ("Hola este es el analizador Lexico")
print ("Ingrese el codigo a analizar \n")
cadena = ''
while True:
 cad = input()
 cadena = cadena+cad+ '\n'
 break
 if not cadena: continue
 print ('\n')
lexer = lex.lex()
#ciclo para mostrar tokens
lexer.input(cadena)
while True:
    tok = lexer.token()
```

```
    if not tok : break
    print (tok)
```

3) Mayo 26

- Creación de carpetas finales del trabajo
- Implementación de lectura de archivos xml
- Interactividad con el usuario
- Creación de archivo de prueba
- Primer planteo de conversión a html
- Creación de archivo html con el mismo nombre que el xml
- Adición de nuevas etiquetas

lexer.py

```
import ply.lex as lex
import re
import codecs
import os
import sys
error_caracter_ilegal=[]

tokens = [ 'DT1','DT2','APERTURA_ARTICLE', 'CIERRE_ARTICLE' , 'APERTURA_PARA',
'CIERRE_PARA', 'TEXTO',
          'APERTURA_INFO' , 'CIERRE_INFO' , 'APERTURA_TITLE' , 'CIERRE_TITLE' ,
'APERTURA_ITEMIZEDLIST',
          'CIERRE_ITEMIZEDLIST', 'APERTURA_IMPORTANT' , 'CIERRE_IMPORTANT' ,
'APERTURA_SIMPARA',
          'CIERRE_SIMPARA' , 'APERTURA_ADDRESS' , 'CIERRE_ADDRESS' ,
'APERTURA_MEDIAOBJECT' , 'CIERRE_MEDIAOBJECT' ,
          'APERTURA_INFORMALTABLE' , 'CIERRE_INFORMALTABLE' , 'APERTURA_COMMENT'
, 'CIERRE_COMMENT' ,
          'APERTURA_ABSTRACT' , 'CIERRE_ABSTRACT' , 'APERTURA_SECTION' ,
'CIERRE_SECTION' , 'APERTURA_SIMPLESECT' ,
          'CIERRE_SIMPLESECT' , 'APERTURA_EMPHASIS' , 'CIERRE_EMPHASIS' ,
'APERTURA_LINK' , 'CIERRE_LINK' ,
          'APERTURA_FIRSTNAME' , 'CIERRE_FIRSTNAME' , 'APERTURA_SURNAME' ,
'CIERRE_SURNAME' , 'APERTURA_STREET' ,
          'CIERRE_STREET' , 'APERTURA_CITY' , 'CIERRE_CITY' , 'APERTURA_STATE' ,
'CIERRE_STATE' , 'APERTURA_PHONE' ,
          'CIERRE_PHONE' , 'APERTURA_EMAIL' , 'CIERRE_EMAIL' , 'APERTURA_DATE' ,
'CIERRE_DATE' , 'APERTURA_YEAR' ,
          'CIERRE_YEAR' , 'APERTURA_HOLDER' , 'CIERRE_HOLDER',
'APERTURA_IMAGEDATA' , 'APERTURA_VIDEOOBJECT' ,
          'CIERRE_VIDEOOBJECT' , 'APERTURA_IMAGENOBJECT' , 'CIERRE_IMAGENOBJECT'
, 'APERTURA_VIDEODATA', 'APERTURA_LISTITEM' ,
          'CIERRE_LISTITEM' , 'APERTURA_TGROUP' , 'CIERRE_TGROUP' ,
'APERTURA_THEAD' , 'CIERRE_THEAD' , 'APERTURA_TFOOT' ,
          'CIERRE_TFOOT' , 'APERTURA_TBODY' , 'CIERRE_TBODY' , 'APERTURA_ROW' ,
'CIERRE_ROW' , 'APERTURA_ENTRY' ,
          'CIERRE_ENTRY' , 'APERTURA_ENTRYTBL' , 'CIERRE_ENTRYTBL'
```

```
  ]

t_ignore = '\t '   #nose que hace pero vi en varios, creo q ignora espacios en
blanco o tabulacion


t_APERTURA_ARTICLE = r'<article>'
t_CIERRE_ARTICLE = r'</article>'
t_APERTURA_ITEMIZEDLIST = r'<itemizedlist>'
t_CIERRE_ITEMIZEDLIST = r'</itemizedlist>'
t_APERTURA_SIMPARA = r'<simpara>'
t_CIERRE_SIMPARA = r'</simpara>'
t_APERTURA_ADDRESS = r'<address>'
t_CIERRE_ADDRESS = r'</address>'
t_APERTURA_MEDIAOBJECT = r'<mediaobject>'
t_CIERRE_MEDIAOBJECT = r'</mediaobject>'
t_APERTURA_INFORMALTABLE = r'<informaltable>'
t_CIERRE_INFORMALTABLE = r'</informaltable>'
t_APERTURA_COMMENT = r'<comment>'
t_CIERRE_COMMENT = r'</comment>'
t_APERTURA_ABSTRACT = r'<abstract>'
t_CIERRE_ABSTRACT = r'</abstract>'
t_APERTURA_SECTION = r'<section>'
t_CIERRE_SECTION = r'</section>'
t_APERTURA_SIMPLESECT = r'<simplesect>'
t_CIERRE_SIMPLESECT = r'</simplesect>'
t_APERTURA_EMPHASIS = r'<emphasis>'
t_CIERRE_EMPHASIS = r'</emphasis>'
t_APERTURA_LINK = r'<link>'
t_CIERRE_LINK = r'</link>'
t_APERTURA_FIRSTNAME = r'<firstname>'
t_CIERRE_FIRSTNAME = r'</firstname>'
t_APERTURA_SURNAME = r'<surname>'
t_CIERRE_SURNAME = r'</surname>'
t_APERTURA_STREET = r'<street>'
t_CIERRE_STREET = r'</street>'
t_APERTURA_CITY = r'<city>'
t_CIERRE_CITY = r'</city>'
t_APERTURA_STATE = r'<state>'
t_CIERRE_STATE = r'</state>'
t_APERTURA_PHONE = r'<phone>'
t_CIERRE_PHONE = r'</phone>'
t_APERTURA_EMAIL = r'<email>'
t_CIERRE_EMAIL = r'</email>'
t_APERTURA_DATE = r'<date>'
t_CIERRE_DATE = r'</date>'
t_APERTURA_YEAR = r'<year>'
t_CIERRE_YEAR = r'</year>'
t_APERTURA_HOLDER = r'<holder>'
t_CIERRE_HOLDER = r'</holder>'
t_APERTURA_VIDEOOBJECT = r'<videoobject>'
t_CIERRE_VIDEOOBJECT = r'</videoobject>'
t_APERTURA_IMAGENOBJECT = r'<imagenobject>'
t_CIERRE_IMAGENOBJECT = r'</imagenobject>'
t_APERTURA_LISTITEM = r'<listitem>'
```

```
t_CIERRE_LISTITEM = r'</listitem>'
t_APERTURA_INFORMALTABLE = r'<informaltable>'
t_CIERRE_INFORMALTABLE = r'</informaltable>'
t_APERTURA_TGROUP = r'<tgroup>'
t_CIERRE_TGROUP = r'</tgroup>'
t_APERTURA_THEAD = r'<thead>'
t_CIERRE_THEAD = r'</thead>'
t_APERTURA_TFOOD = r'<tfood>'
t_CIERRE_TFOOD = r'</tfood>'
t_APERTURA_TBODY = r'<tbody>'
t_CIERRE_TBODY = r'<tbody>'
t_APERTURA_ROW = r'<row>'
t_CIERRE_ROW = r'</row>'
t_APERTURA_ENTRY = r'<entry>'
t_CIERRE_ENTRY = r'</entry>'
t_APERTURA_ENTRYTBL = r'<entrytbl>'
t_CIERRE_ENTRYTBL = r'</entrytbl>'


arch= open("src/archivo.html","w",encoding="utf-8")

#funciones
def t_DT1(t):
      r'<[!]DOCTYPE\sarticle>'
      arch.write("<!DOCTYPE html>")
def t_TEXTO (t):
    r'[a-zA-Z][a-zA-Z0-9]*'  #falta ver caraxteres especiales
    arch.write(f'{t.value} ')
    return (t)
def t_error(t):

 print ("caracter ilegal %s" % t.value[0])
 t.lexer.skip(1)
def t_APERTURA_PARA(t):
      r'<para>'
      arch.write("<p>")
      return(t)
def t_CIERRE_PARA(t):
      r'</para>'
      arch.write("</p>")
      return(t)
def t_APERTURA_TITLE(t):
      r'<title>'
      arch.write("<h1>")
      return(t)
def t_CIERRE_TITLE(t):
      r'</title>'
      arch.write("</h1>")
      return(t)
def t_APERTURA_INFO(t):
      r'<info>'
      arch.write('<div style="color:white;background-color:green;font-
size:8pts"><p>')   #anda bien
      return(t)
def t_CIERRE_INFO(t):
```

```
      r'</info>'
      arch.write('</p></div>')
      return(t)
def t_APERTURA_IMPORTANT(t):
      r'<important>'
      arch.write('<div style="background-color:red;color:white">') #anda bien
      return(t)
def t_CIERRE_IMPORTANT(t):
      r'</important>'
      arch.write('</div>')
      return(t)
def t_APERTURA_IMAGEDATA (t):
      r'<imagedata =
fileref="^(https|ftp|http|ftps):\/\/([^\s\/$.?#]+\.[^\s\/$.?#]+)(:\d+)?(\/[^\s$?
#]*)?(#[^\s]*)?$">'
      return(t)
def t_APERTURA_VIDEODATA (t):
      r'<videodata=
fileref="^(https|ftp|http|ftps):\/\/([^\s\/$.?#]+\.[^\s\/$.?#]+)(:\d+)?(\/[^\s$?
#]*)?(#[^\s]*)?$">'
      return(t)
def t_APERTURA_LINK (t):
      r'link = xlink:href
="^(https|ftp|http|ftps):\/\/([^\s\/$.?#]+\.[^\s\/$.?#]+)(:\d+)?(\/[^\s$?#]*)?(#
[^\s]*)?$"'
      return (t)



lexer = lex.lex()
contador = 0
```

main.py

```
from lexer import lexer
import os
import sys
print ("Hola este es el analizador Lexico")
print ("Ingrese 1 si quiere ingresar datos a mano y 2 si quiere desde un archivo
de prueba \n")
op = input()

if op == "1":                                                  #ingreso manual
      print("ingrese lo que quiere analizar")
      cadena = input()
      lexer.input(cadena)
      while True:
            tok = lexer.token()

            if not tok : break
            print (tok)

elif op == "2":                                          #ingreso por archivo
      print("todavia no esta listo")
      n = 0
```

```
      ejemplo_dir = 'prueba/'                   #elegir el archivo
      with os.scandir(ejemplo_dir) as ficheros:
            ficheros = [fichero.name for fichero in ficheros if
fichero.is_file()]    #ficheros es una lista con los archivos de la carpeta
prueba
      for i in ficheros:

            print(f"{n+1}: {ficheros[n]}")
            n +=1
      print("elegi el archivo")
      op2 = input()
      if int(op2) <= n:
            ruta = ficheros[n-1]
            with open(f"prueba/{ruta}","r",encoding="utf-8") as maestro: #esto
ya funciona para cualquier fichero en prueba/
                  print("hola abri el archivo jejej")
                  lexer.input(maestro.read())
                  while True:
                        tok = lexer.token()

                        if not tok:break
                        print(tok)
                  cambio = ruta.replace(".xml","")
                  os.rename("src/archivo.html",f"src/{cambio}.html" )
      else:
            print("numero invalido")
else:
      print("volve a empezar")
```

nombreprueba.xml

```
<!DOCTYPE article>
<article>
<info>
<title>El titulo del articulo</title>
<author>
<firstname>Juan</firstname>
<surname>Perez</surname>
</author>
</info>
<sect1>
<title>Titulo para la seccion 1</title>
<para>
<important>
Esto es un parrafo
</important>
</para>
<para>
Otro parrafo.
</para>
</sect1>
</article>
```

nombreprueba.html

```
<!DOCTYPE html><div style="color:white;background-color:green;font-
size:8pts"><p><h1>El titulo del articulo </h1>author Juan Perez author
</p></div>sect1 <h1>Titulo para la seccion </h1><p><div style="background-
color:red;color:white">Esto es un parrafo </div></p><p>Otro parrafo </p>sect1
```

4) Mayo 27

- Traducciones de etiquetas de links, tablas y listas
- Creación de carpetas para los archivos html
- Arreglo del problema que había al abrir archivos
- Primera prueba de las tablas
- Se arregló la generación de archivos html
- Se generaron más archivos de prueba

lexer.py

```
import ply.lex as lex
import re
import codecs
import os
import sys
error_caracter_ilegal=[]

tokens = [ 'DT1','DT2','APERTURA_ARTICLE', 'CIERRE_ARTICLE' , 'APERTURA_PARA',
'CIERRE_PARA', 'TEXTO',
          'APERTURA_INFO' , 'CIERRE_INFO' , 'APERTURA_TITLE' , 'CIERRE_TITLE' ,
'APERTURA_ITEMIZEDLIST',
          'CIERRE_ITEMIZEDLIST', 'APERTURA_IMPORTANT' , 'CIERRE_IMPORTANT' ,
'APERTURA_SIMPARA',
          'CIERRE_SIMPARA' , 'APERTURA_ADDRESS' , 'CIERRE_ADDRESS' ,
'APERTURA_MEDIAOBJECT' , 'CIERRE_MEDIAOBJECT' ,
          'APERTURA_INFORMALTABLE' , 'CIERRE_INFORMALTABLE' , 'APERTURA_COMMENT'
, 'CIERRE_COMMENT' ,
          'APERTURA_ABSTRACT' , 'CIERRE_ABSTRACT' , 'APERTURA_SECTION' ,
'CIERRE_SECTION' , 'APERTURA_SIMPLESECT' ,
          'CIERRE_SIMPLESECT' , 'APERTURA_EMPHASIS' , 'CIERRE_EMPHASIS' ,
'APERTURA_LINK' , 'CIERRE_LINK' ,
          'APERTURA_FIRSTNAME' , 'CIERRE_FIRSTNAME' , 'APERTURA_SURNAME' ,
'CIERRE_SURNAME' , 'APERTURA_STREET' ,
          'CIERRE_STREET' , 'APERTURA_CITY' , 'CIERRE_CITY' , 'APERTURA_STATE' ,
'CIERRE_STATE' , 'APERTURA_PHONE' ,
          'CIERRE_PHONE' , 'APERTURA_EMAIL' , 'CIERRE_EMAIL' , 'APERTURA_DATE' ,
'CIERRE_DATE' , 'APERTURA_YEAR' ,
          'CIERRE_YEAR' , 'APERTURA_HOLDER' , 'CIERRE_HOLDER',
'APERTURA_IMAGEDATA' , 'APERTURA_VIDEOOBJECT' ,
          'CIERRE_VIDEOOBJECT' , 'APERTURA_IMAGENOBJECT' , 'CIERRE_IMAGENOBJECT'
, 'APERTURA_VIDEODATA', 'APERTURA_LISTITEM' ,
          'CIERRE_LISTITEM' , 'APERTURA_TGROUP' , 'CIERRE_TGROUP' ,
'APERTURA_THEAD' , 'CIERRE_THEAD' , 'APERTURA_TFOOT' ,
          'CIERRE_TFOOT' , 'APERTURA_TBODY' , 'CIERRE_TBODY' , 'APERTURA_ROW' ,
'CIERRE_ROW' , 'APERTURA_ENTRY' ,
          'CIERRE_ENTRY' , 'APERTURA_ENTRYTBL' , 'CIERRE_ENTRYTBL'
```

```
  ]

t_ignore = '\t '   #nose que hace pero vi en varios, creo q ignora espacios en
blanco o tabulacion


t_APERTURA_ARTICLE = r'<article>'
t_CIERRE_ARTICLE = r'</article>'
t_APERTURA_SIMPARA = r'<simpara>'
t_CIERRE_SIMPARA = r'</simpara>'
t_APERTURA_ADDRESS = r'<address>'
t_CIERRE_ADDRESS = r'</address>'
t_APERTURA_MEDIAOBJECT = r'<mediaobject>'
t_CIERRE_MEDIAOBJECT = r'</mediaobject>'
t_APERTURA_COMMENT = r'<comment>'
t_CIERRE_COMMENT = r'</comment>'
t_APERTURA_ABSTRACT = r'<abstract>'
t_CIERRE_ABSTRACT = r'</abstract>'
t_APERTURA_SECTION = r'<section>'
t_CIERRE_SECTION = r'</section>'
t_APERTURA_SIMPLESECT = r'<simplesect>'
t_CIERRE_SIMPLESECT = r'</simplesect>'
t_APERTURA_EMPHASIS = r'<emphasis>'
t_CIERRE_EMPHASIS = r'</emphasis>'
t_APERTURA_FIRSTNAME = r'<firstname>'
t_CIERRE_FIRSTNAME = r'</firstname>'
t_APERTURA_SURNAME = r'<surname>'
t_CIERRE_SURNAME = r'</surname>'
t_APERTURA_STREET = r'<street>'
t_CIERRE_STREET = r'</street>'
t_APERTURA_CITY = r'<city>'
t_CIERRE_CITY = r'</city>'
t_APERTURA_STATE = r'<state>'
t_CIERRE_STATE = r'</state>'
t_APERTURA_PHONE = r'<phone>'
t_CIERRE_PHONE = r'</phone>'
t_APERTURA_EMAIL = r'<email>'
t_CIERRE_EMAIL = r'</email>'
t_APERTURA_DATE = r'<date>'
t_CIERRE_DATE = r'</date>'
t_APERTURA_YEAR = r'<year>'
t_CIERRE_YEAR = r'</year>'
t_APERTURA_HOLDER = r'<holder>'
t_CIERRE_HOLDER = r'</holder>'
t_APERTURA_VIDEOOBJECT = r'<videoobject>'
t_CIERRE_VIDEOOBJECT = r'</videoobject>'
t_APERTURA_IMAGENOBJECT = r'<imagenobject>'
t_CIERRE_IMAGENOBJECT = r'</imagenobject>'
t_APERTURA_TGROUP = r'<tgroup>'
t_CIERRE_TGROUP = r'</tgroup>'
t_APERTURA_THEAD = r'<thead>'
t_CIERRE_THEAD = r'</thead>'
t_APERTURA_TFOOT = r'<tfood>'
t_CIERRE_TFOOT = r'</tfood>'
t_APERTURA_TBODY = r'<tbody>'
```

```
t_CIERRE_TBODY = r'<tbody>'
t_APERTURA_ENTRYTBL = r'<entrytbl>'
t_CIERRE_ENTRYTBL = r'</entrytbl>'


arch= open("src/html_generados/archivo.html","w",encoding="utf-8")

#funciones
def t_DT1(t):
      r'<[!]DOCTYPE\sarticle>'
      arch.write("<!DOCTYPE html>")
def t_TEXTO (t):
    r'[a-zA-Z][a-zA-Z0-9]*'  #falta ver caraxteres especiales
    arch.write(f'{t.value} ')
    return (t)
def t_error(t):

 #print ("caracter ilegal %s" % t.value[0])
 t.lexer.skip(1)
def t_APERTURA_PARA(t):
      r'<para>'
      arch.write("<p>")
      return(t)
def t_CIERRE_PARA(t):
      r'</para>'
      arch.write("</p>")
      return(t)
def t_APERTURA_TITLE(t):
      r'<title>'
      arch.write("<h1>")
      return(t)
def t_CIERRE_TITLE(t):
      r'</title>'
      arch.write("</h1>")
      return(t)
def t_APERTURA_INFO(t):
      r'<info>'
      arch.write('<div style="color:white;background-color:green;font-
size:8pts"><p>')   #anda bien
      return(t)
def t_CIERRE_INFO(t):
      r'</info>'
      arch.write('</p></div>')
      return(t)
def t_APERTURA_IMPORTANT(t):
      r'<important>'
      arch.write('<div style="background-color:red;color:white">') #anda bien
      return(t)
def t_CIERRE_IMPORTANT(t):
      r'</important>'
      arch.write('</div>')
      return(t)
#def t_APERTURA_IMAGEDATA (t):
      #r'<imagedata =
fileref="^(https|ftp|http|ftps):\/\/([^\s\/$.?#]+\.[^\s\/$.?#]+)(:\d+)?(\/[^\s$?
```

```
#]*)?(#[^\s]*)?$">'
      #return(t)
#def t_APERTURA_VIDEODATA (t):
      #r'<videodata=
fileref="^(https|ftp|http|ftps):\/\/([^\s\/$.?#]+\.[^\s\/$.?#]+)(:\d+)?(\/[^\s$?
#]*)?(#[^\s]*)?$">'
      #return(t)
#def t_APERTURA_LINK (t):
      #r'link = xlink:href
="^(https|ftp|http|ftps):\/\/([^\s\/$.?#]+\.[^\s\/$.?#]+)(:\d+)?(\/[^\s$?#]*)?(#
[^\s]*)?$?"'
      #arch.write(f'<a href="{t.value}">esto es un link</a>')
      #return (t)
def t_APERTURA_INFORMALTABLE(t):
      r'<informaltable>'
      arch.write("<table>")
      return (t)
def t_CIERRE_INFORMALTABLE(t):
      r'</informaltable>'
      arch.write("</table>")
      return (t)
def t_APERTURA_ROW(t):                              #un problema con esto es
que en html todos son tr y se diferencian adentro usando
      r'<row>'                                      #<th></th> para los
encabezados y pies de la tabla
      arch.write("<tr>")
      return (t)
def t_CIERRE_ROW(t):
      r'</row>'
      arch.write("</tr>")
      return (t)
def t_APERTURA_ENTRY(t):
      r'<entry>'
      arch.write("<td>")
      return (t)
def t_CIERRE_ENTRY(t):
      r'</entry>'
      arch.write("</td>")
      return (t)
def t_APERTURA_ITEMIZEDLIST(t):
      r'<itemizedlist>'
      arch.write("<ul>")
      return (t)
def t_CIERRE_ITEMIZEDLIST(t):
      r'</itemizedlist>'
      arch.write("</ul>")
      return (t)
def t_APERTURA_LISTITEM(t):
      r'<listitem>'
      arch.write("<il>")
      return (t)
def t_CIERRE_LISTITEM(t):
      r'</listitem>'
      arch.write("</il>")
      return (t)
```

```
lexer = lex.lex()
contador = 0
```

main.py

```
from lexer import lexer
import os
import sys
print ("Hola este es el analizador Lexico")
print ("Ingrese 1 si quiere ingresar datos a mano y 2 si quiere desde un archivo
de prueba \n")
op = input()

if op == "1":                                                    #ingreso manual
      print("ingrese lo que quiere analizar")
      cadena = input()
      lexer.input(cadena)
      while True:
            tok = lexer.token()

            if not tok : break
            print (tok)

elif op == "2":                                          #ingreso por archivo
      print("todavia no esta listo")
      n = 0

      ejemplo_dir = 'prueba/'                    #elegir el archivo
      with os.scandir(ejemplo_dir) as ficheros:
            ficheros = [fichero.name for fichero in ficheros if
fichero.is_file()]    #ficheros es una lista con los archivos de la carpeta
prueba
      for i in ficheros:

            print(f"{n+1}: {ficheros[n]}")
            n +=1
      print("elegi el archivo")
      op2 = input()
      if int(op2) <= n:
            ruta = ficheros[int(op2)-1]
            with open(f"prueba/{ruta}","r",encoding="utf-8") as maestro: #esto
ya funciona para cualquier fichero en prueba/
                  print(f"hola abri el archivo: {ruta}")
                  lexer.input(maestro.read())
                  while True:
                        tok = lexer.token()

                        if not tok:break
                        print(tok)
                  cambio = ruta.replace(".xml","")

os.rename("src/html_generados/archivo.html",f"src/html_generados/{cambio}.html"
)
      else:
            print("numero invalido")
```

```
else:
      print("volve a empezar")
```

ejLista.xml

```
<itemizedlist>
    <listitem>
      <para>Hal Computer Systems y O'Reilly &amp;amp; Associates, de
        1991 a 1994</para>
    </listitem>
    <listitem>
      <para>El grupo Davenport, de 1994 a 1998.</para>
    </listitem>
    <listitem>
      <para>El grupo <acronym>OASIS</acronym> de 1998 hasta hoy.</para>
    </listitem>
  </itemizedlist>
```

ejLista.html

```
<ul><il><p>Hal Computer Systems y O Reilly amp Associates de a
</p></il><il><p>El grupo Davenport de a </p></il><il><p>El grupo acronym OASIS
acronym de hasta hoy </p></il></ul>
```

tablasprueba.xml

```
<!DOCTYPE article>
<article>
    <title>Archivo de prueba sobre tablas </title>

    <abstract>
            <para>Se vera a continuacion una tabla de puntos sobre del top 5 de
la liga de futbol argentino <emphasis> campeonatado 2023 </emphasis>, a modo de
demostracion </para>
    </abstract>

    <informaltable>
        <tgroup>
            <thead>
                <row>
                    <entry>
                        <important>
                            <title> CAMPEONATO ARGENTINO 2023 </title>
                        </important>
                    </entry>
                </row>
            </thead>
            <tbody>
                <row>
                    <entry>
                        <important>
                            <title> EQUIPO </title>
                            <title> PUNTOS </title>
                        </important>
                    </entry>
                </row>
```

```
                <row>
                    <entry>
                        <para>  River plate</para>
                        <para> 40 </para>
                    </entry>
                </row>
                <row>
                    <entry>
                        <para> San lorenzo</para>
                        <para> 35 </para>
                    </entry>
                </row>
                <row>
                    <entry>
                        <para> talleres </para>
                        <para> 31 </para>
                    </entry>
                </row>
                <row>
                    <entry>
                        <para> Estudiantes </para>
                        <para> 31 </para>
                    </entry>
                </row>
                <row>
                    <entry>
                        <para> Rosario central </para>
                        <para> 30 </para>
                    </entry>
                </row>
            </tbody>
        </tgroup>
    </informaltable>

    <section>
        <para> final del archivo de prueba de tablas </para>
    </section>
</archivo>
```

archprueba.xml

```
<!DOCTYPE article>
<article>
    <info>
        <title>Archivo de prueba presentacion del grupo </title>

        <abstract>
            <title>integrantes </title>
            <para>A continuacion se mostraran los nombres de los distintos
integrantes del grupo </para>
        </abstract>

        <author>
            <firtname>Camilo</firtname>
            <surname>Aguirre </surname>
        </author>
```

```
        <author>
            <firtname> Joaquin </firtname>
            <surname> Bianciotto </surname>
        </author>
        <author>
            <firtname> Julian </firtname>
            <surname> Colombo </surname>
        </author>
        <author>
            <firtname> Yoel </firtname>
            <surname> Maraim </surname>
        </author>
    </info>

    <comment> correo electronico de contacto <email> grupo4@mail.com </email>
</comment>

    <section>
        <title> Las 3 entregas parciales del tpi son las siguiente: </title>

        <itemizedlist>
            <lisitem>
                <para>1er entrega: Documentacia del proyecto y la gramatica a
generar, <emphasis> el 23 de abril </emphasis> </para>
            </listitem>

            <lisitem>
                <para>2da entrega: Presentacion del lexer que reconozca los
tokens del leguaje, <emphasis> el 4 de junio </emphasis> </para>
            </listitem>

            <lisitem>
                <para>3er entrega (entrega final): presentacion del tpi
completo, lexer y parse, incluyendo toda la presentacion del tpi antes de la
clase con una exposicion hora de no menos de 20 minutos <emphasis> el 2 junio
</emphasis> </para>
            </listitem>
        </itemizedlist>

        <para>Con esto se concluye el archivo de prueba, donde se experimento
con distinitas etiquetas anidadas, como asi tmb listas</para>
    </section>
</article>
```

5) Mayo 28

- Se añadieron archivos de error

Docu_error_listas.txt

```
El lexer a sido sometido a los siguientes errores del archivo "LISTAS_ERROR.xml"

linea 21 -> se abre una etiqueta <itemizedlist> dentro de la etiqueta <listitem>

linea 22 -> se cierra la etiqueta </para> antes de cerrar la etiqueta de
contenia dentro </emphasis>
```

```
linea 27 -> se abre la etiqueta <date> dentro de la etiqueta <emphasis>
```

Docu_error_tablas.txt

```
El lexer a sido sometido a los siguientes errores del archivos
"TABLAR_ERROR.xml"

linea 13 -> se abre la etiqueta <entry> pero nunca se cierra

linea 20 -> directamente se escribe una etiqueta <para> despues de <row> sin
haber estado antes la etiqueta <entry>

linea 29 -> se abre la etiqueta <tfood> sin antes cerrar la etiqueta </tbody>

linea 42 -> error en la palabra, se escribe <setion> en vez de <section>
```

LISTAS.xml

```
<!DOCTYPE article>
<article>
    <info>
        <title>Archivo de prueba presentacion del grupo </title>

        <abstract>
            <title>integrantes </title>
            <para>A continuacion se mostraran los nombres de los distintos
integrantes del grupo </para>
        </abstract>

        <author>
            <firtname>Camilo</firtname>
            <surname>Aguirre </surname>
        </author>
        <author>
            <firtname> Joaquin </firtname>
            <surname> Bianciotto </surname>
        </author>
        <author>
            <firtname> Julian </firtname>
            <surname> Colombo </surname>
        </author>
        <author>
            <firtname> Yoel </firtname>
            <surname> Maraim </surname>
        </author>
    </info>

    <comment> correo electronico de contacto <email> grupo4@mail.com </email>
</comment>

    <section>
        <title> Las 3 entregas parciales del tpi son las siguiente: </title>

        <itemizedlist>
            <listitem>
```

```
                <para>1er entrega: Documentacia del proyecto y la gramatica a
generar, <emphasis> el 23 de abril </emphasis> </para>
            </listitem>

            <listitem>
                <para>2da entrega: Presentacion del lexer que reconozca los
tokens del leguaje, <emphasis> el 4 de junio </emphasis> </para>
            </listitem>

            <listitem>
                <para>3er entrega (entrega final): presentacion del tpi
completo, lexer y parse, incluyendo toda la presentacion del tpi antes de la
clase con una exposicion hora de no menos de 20 minutos <emphasis> el 2 junio
</emphasis> </para>
            </listitem>
        </itemizedlist>

        <para>Con esto se concluye el archivo de prueba, donde se experimento
con distinitas etiquetas anidadas, como asi tmb listas</para>
    </section>
</article>
```

LISTAS_ERROR.xml

```
<!DOCTYPE article>
<article>
    <info>
        <title>Entregas tpi</title>

        <abstract>
            <title>integrantes </title>
            <para>Integrantes del grupo n°4</para>
        </abstract>


    </info>

    <comment> correo electronico de contacto <email> grupo4@mail.com </email>
</comment>

    <section>
        <title> Las 3 entregas parciales del tpi son las siguiente: </title>

        <itemizedlist>
            <listitem>
                <itemizedlist>
                    <para>1er entrega: Documentacia del proyecto y la gramatica
a generar, <emphasis> el 23 de abril </para> </emphasis>
                </itemizedlist>
            </listitem>

            <lisitem>
                <para>2da entrega: Presentacion del lexer que reconozca los
tokens del leguaje, <emphasis> <date> el 4 de junio </date> </emphasis> </para>
            </listitem>
```

```
            <listitem>
                <para>3er entrega (entrega final): presentacion del tpi
completo, lexer y parse, incluyendo toda la presentacion del tpi antes de la
clase con una exposicion hora de no menos de 20 minutos <emphasis> el 2 junio
</emphasis> </para>
            </listitem>
        </itemizedlist>

        <para>Con esto se concluye el archivo de prueba, donde se experimento
con distinitas etiquetas anidadas, como asi tmb listas</para>
    </section>
</article>
```

TABLAS_ERROR.xml

```
<!DOCTYPE article>
<article>
    <title>Archivo de prueba sobre tablas </title>

    <abstract>
            <para>Se vera a continuacion una tabla de puntos sobre del top 5 de
la liga de futbol argentino <emphasis> campeonatado 2023 </emphasis>, a modo de
demostracion </para>
    </abstract>

    <informaltable>
        <tgroup>
            <tbody>
                <row>
                    <entry>
                        <important>
                            <title> EQUIPO </title>
                            <title> PUNTOS </title>
                        </important>
                <row>
                <row>
                        <para>  River plate</para>
                        <para> 40 </para>
                </row>
                <row>
                    <entry>
                        <para> San lorenzo</para>
                        <para> 35 </para>
                    </entry>
                </row>
            <tfood>
            </tbody>
                <row>
                    <entry>
                        <important>
                            <title> CAMPEONATO ARGENTINO 2023 </title>
                        </important>
                    </entry>
                </row>
            </tfood>
        </tgroup>
```

```
    </informaltable>

    <setion>
        <para> final del archivo de prueba de tablas </para>
    </section>
</archivo>
```

6) Mayo 31

- Modificación de la expresión regular de las URL

lexer.py

```
import ply.lex as lex
import re
import codecs
import os
import sys
error_caracter_ilegal=[]

tokens = [ 'DT1','DT2','APERTURA_ARTICLE', 'CIERRE_ARTICLE' , 'APERTURA_PARA',
'CIERRE_PARA', 'TEXTO',
          'APERTURA_INFO' , 'CIERRE_INFO' , 'APERTURA_TITLE' , 'CIERRE_TITLE' ,
'APERTURA_ITEMIZEDLIST',
          'CIERRE_ITEMIZEDLIST', 'APERTURA_IMPORTANT' , 'CIERRE_IMPORTANT' ,
'APERTURA_SIMPARA',
          'CIERRE_SIMPARA' , 'APERTURA_ADDRESS' , 'CIERRE_ADDRESS' ,
'APERTURA_MEDIAOBJECT' , 'CIERRE_MEDIAOBJECT' ,
          'APERTURA_INFORMALTABLE' , 'CIERRE_INFORMALTABLE' , 'APERTURA_COMMENT'
, 'CIERRE_COMMENT' ,
          'APERTURA_ABSTRACT' , 'CIERRE_ABSTRACT' , 'APERTURA_SECTION' ,
'CIERRE_SECTION' , 'APERTURA_SIMPLESECT' ,
          'CIERRE_SIMPLESECT' , 'APERTURA_EMPHASIS' , 'CIERRE_EMPHASIS' ,
'APERTURA_LINK' , 'CIERRE_LINK' ,
          'APERTURA_FIRSTNAME' , 'CIERRE_FIRSTNAME' , 'APERTURA_SURNAME' ,
'CIERRE_SURNAME' , 'APERTURA_STREET' ,
          'CIERRE_STREET' , 'APERTURA_CITY' , 'CIERRE_CITY' , 'APERTURA_STATE' ,
'CIERRE_STATE' , 'APERTURA_PHONE' ,
          'CIERRE_PHONE' , 'APERTURA_EMAIL' , 'CIERRE_EMAIL' , 'APERTURA_DATE' ,
'CIERRE_DATE' , 'APERTURA_YEAR' ,
          'CIERRE_YEAR' , 'APERTURA_HOLDER' , 'CIERRE_HOLDER',
'APERTURA_IMAGEDATA' , 'APERTURA_VIDEOOBJECT' ,
          'CIERRE_VIDEOOBJECT' , 'APERTURA_IMAGENOBJECT' , 'CIERRE_IMAGENOBJECT'
, 'APERTURA_VIDEODATA', 'APERTURA_LISTITEM' ,
          'CIERRE_LISTITEM' , 'APERTURA_TGROUP' , 'CIERRE_TGROUP' ,
'APERTURA_THEAD' , 'CIERRE_THEAD' , 'APERTURA_TFOOT' ,
          'CIERRE_TFOOT' , 'APERTURA_TBODY' , 'CIERRE_TBODY' , 'APERTURA_ROW' ,
'CIERRE_ROW' , 'APERTURA_ENTRY' ,
          'CIERRE_ENTRY' , 'APERTURA_ENTRYTBL' , 'CIERRE_ENTRYTBL'
  ]

t_ignore = '\t '   #nose que hace pero vi en varios, creo q ignora espacios en
blanco o tabulacion


t_APERTURA_ARTICLE = r'<article>'
```

```
t_CIERRE_ARTICLE = r'</article>'
t_APERTURA_SIMPARA = r'<simpara>'
t_CIERRE_SIMPARA = r'</simpara>'
t_APERTURA_ADDRESS = r'<address>'
t_CIERRE_ADDRESS = r'</address>'
t_APERTURA_MEDIAOBJECT = r'<mediaobject>'
t_CIERRE_MEDIAOBJECT = r'</mediaobject>'
t_APERTURA_COMMENT = r'<comment>'
t_CIERRE_COMMENT = r'</comment>'
t_APERTURA_ABSTRACT = r'<abstract>'
t_CIERRE_ABSTRACT = r'</abstract>'
t_APERTURA_SECTION = r'<section>'
t_CIERRE_SECTION = r'</section>'
t_APERTURA_SIMPLESECT = r'<simplesect>'
t_CIERRE_SIMPLESECT = r'</simplesect>'
t_APERTURA_EMPHASIS = r'<emphasis>'
t_CIERRE_EMPHASIS = r'</emphasis>'
t_APERTURA_FIRSTNAME = r'<firstname>'
t_CIERRE_FIRSTNAME = r'</firstname>'
t_APERTURA_SURNAME = r'<surname>'
t_CIERRE_SURNAME = r'</surname>'
t_APERTURA_STREET = r'<street>'
t_CIERRE_STREET = r'</street>'
t_APERTURA_CITY = r'<city>'
t_CIERRE_CITY = r'</city>'
t_APERTURA_STATE = r'<state>'
t_CIERRE_STATE = r'</state>'
t_APERTURA_PHONE = r'<phone>'
t_CIERRE_PHONE = r'</phone>'
t_APERTURA_EMAIL = r'<email>'
t_CIERRE_EMAIL = r'</email>'
t_APERTURA_DATE = r'<date>'
t_CIERRE_DATE = r'</date>'
t_APERTURA_YEAR = r'<year>'
t_CIERRE_YEAR = r'</year>'
t_APERTURA_HOLDER = r'<holder>'
t_CIERRE_HOLDER = r'</holder>'
t_APERTURA_VIDEOOBJECT = r'<videoobject>'
t_CIERRE_VIDEOOBJECT = r'</videoobject>'
t_APERTURA_IMAGENOBJECT = r'<imagenobject>'
t_CIERRE_IMAGENOBJECT = r'</imagenobject>'
t_APERTURA_TGROUP = r'<tgroup>'
t_CIERRE_TGROUP = r'</tgroup>'
t_APERTURA_THEAD = r'<thead>'
t_CIERRE_THEAD = r'</thead>'
t_APERTURA_TFOOT = r'<tfood>'
t_CIERRE_TFOOT = r'</tfood>'
t_APERTURA_TBODY = r'<tbody>'
t_CIERRE_TBODY = r'<tbody>'
t_APERTURA_ENTRYTBL = r'<entrytbl>'
t_CIERRE_ENTRYTBL = r'</entrytbl>'


arch= open("src/html_generados/archivo.html","w",encoding="utf-8")
```

```
#funciones
def t_DT1(t):
      r'<[!]DOCTYPE\sarticle>'
      arch.write("<!DOCTYPE html>")
def t_TEXTO (t):
    r'[a-zA-Z][a-zA-Z0-9]*'  #falta ver caraxteres especiales
    arch.write(f'{t.value} ')
    return (t)
def t_error(t):

 #print ("caracter ilegal %s" % t.value[0])
 t.lexer.skip(1)
def t_APERTURA_PARA(t):
      r'<para>'
      arch.write("<p>")
      return(t)
def t_CIERRE_PARA(t):
      r'</para>'
      arch.write("</p>")
      return(t)
def t_APERTURA_TITLE(t):
      r'<title>'
      arch.write("<h1>")
      return(t)
def t_CIERRE_TITLE(t):
      r'</title>'
      arch.write("</h1>")
      return(t)
def t_APERTURA_INFO(t):
      r'<info>'
      arch.write('<div style="color:white;background-color:green;font-
size:8pts"><p>')   #anda bien
      return(t)
def t_CIERRE_INFO(t):
      r'</info>'
      arch.write('</p></div>')
      return(t)
def t_APERTURA_IMPORTANT(t):
      r'<important>'
      arch.write('<div style="background-color:red;color:white">') #anda bien
      return(t)
def t_CIERRE_IMPORTANT(t):
      r'</important>'
      arch.write('</div>')
      return(t)
def t_APERTURA_IMAGEDATA (t):
      r'<imagedata=fileref=""^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      return(t)
def t_APERTURA_VIDEODATA (t):
      r'<videodata=fileref="^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      return(t)
def t_APERTURA_LINK (t):
      r'link = xlink:href ="^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
```

```
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      arch.write(f'<a href="{t.value}">esto es un link</a>')
      return (t)
def t_APERTURA_INFORMALTABLE(t):
      r'<informaltable>'
      arch.write("<table>")
      return (t)
def t_CIERRE_INFORMALTABLE(t):
      r'</informaltable>'
      arch.write("</table>")
      return (t)
def t_APERTURA_ROW(t):                                  #un problema con esto es
que en html todos son tr y se diferencian adentro usando
      r'<row>'                                          #<th></th> para los
encabezados y pies de la tabla
      arch.write("<tr>")
      return (t)
def t_CIERRE_ROW(t):
      r'</row>'
      arch.write("</tr>")
      return (t)
def t_APERTURA_ENTRY(t):
      r'<entry>'
      arch.write("<td>")
      return (t)
def t_CIERRE_ENTRY(t):
      r'</entry>'
      arch.write("</td>")
      return (t)
def t_APERTURA_ITEMIZEDLIST(t):
      r'<itemizedlist>'
      arch.write("<ul>")
      return (t)
def t_CIERRE_ITEMIZEDLIST(t):
      r'</itemizedlist>'
      arch.write("</ul>")
      return (t)
def t_APERTURA_LISTITEM(t):
      r'<listitem>'
      arch.write("<il>")
      return (t)
def t_CIERRE_LISTITEM(t):
      r'</listitem>'
      arch.write("</il>")
      return (t)
lexer = lex.lex()
contador = 0
```

7) Junio 3

- Implementación de control de errores de escritura
- Implementación de control de línea en la que aparece un token
- Adición de etiquetas

- Implementación de control de caracteres especiales
- Adición de los errores correspondientes a los nuevos controles
- Cambios estéticos
- Implementación de función para borrar la salida en la pantalla
- Arreglo de errores correspondientes a la apertura de los archivos, incorporación de flags

lexer.py

```
import ply.lex as lex
import re
import codecs
import os
import sys
error_caracter_ilegal=[]

tokens = [ 'DT1','DT2','APERTURA_ARTICLE', 'CIERRE_ARTICLE' , 'APERTURA_PARA',
'CIERRE_PARA', 'TEXTO',
          'APERTURA_INFO' , 'CIERRE_INFO' , 'APERTURA_TITLE' , 'CIERRE_TITLE' ,
'APERTURA_ITEMIZEDLIST',
          'CIERRE_ITEMIZEDLIST', 'APERTURA_IMPORTANT' , 'CIERRE_IMPORTANT' ,
'APERTURA_SIMPARA',
          'CIERRE_SIMPARA' , 'APERTURA_ADDRESS' , 'CIERRE_ADDRESS' ,
'APERTURA_MEDIAOBJECT' , 'CIERRE_MEDIAOBJECT' ,
          'APERTURA_INFORMALTABLE' , 'CIERRE_INFORMALTABLE' , 'APERTURA_COMMENT'
, 'CIERRE_COMMENT' ,
          'APERTURA_ABSTRACT' , 'CIERRE_ABSTRACT' , 'APERTURA_SECTION' ,
'CIERRE_SECTION' , 'APERTURA_SIMPLESECT' ,
          'CIERRE_SIMPLESECT' , 'APERTURA_EMPHASIS' , 'CIERRE_EMPHASIS' ,
'APERTURA_LINK' , 'CIERRE_LINK' ,
          'APERTURA_FIRSTNAME' , 'CIERRE_FIRSTNAME' , 'APERTURA_SURNAME' ,
'CIERRE_SURNAME' , 'APERTURA_STREET' ,
          'CIERRE_STREET' , 'APERTURA_CITY' , 'CIERRE_CITY' , 'APERTURA_STATE' ,
'CIERRE_STATE' , 'APERTURA_PHONE' ,
          'CIERRE_PHONE' , 'APERTURA_EMAIL' , 'CIERRE_EMAIL' , 'APERTURA_DATE' ,
'CIERRE_DATE' , 'APERTURA_YEAR' ,
          'CIERRE_YEAR' , 'APERTURA_HOLDER' , 'CIERRE_HOLDER',
'APERTURA_IMAGEDATA' , 'APERTURA_VIDEOOBJECT' ,
          'CIERRE_VIDEOOBJECT' , 'APERTURA_IMAGENOBJECT' , 'CIERRE_IMAGENOBJECT'
, 'APERTURA_VIDEODATA', 'APERTURA_LISTITEM' ,
          'CIERRE_LISTITEM' , 'APERTURA_TGROUP' , 'CIERRE_TGROUP' ,
'APERTURA_THEAD' , 'CIERRE_THEAD' , 'APERTURA_TFOOT' ,
          'CIERRE_TFOOT' , 'APERTURA_TBODY' , 'CIERRE_TBODY' , 'APERTURA_ROW' ,
'CIERRE_ROW' , 'APERTURA_ENTRY' ,
          'CIERRE_ENTRY' , 'APERTURA_ENTRYTBL' ,
'CIERRE_ENTRYTBL','APERTURA_AUTHOR','CIERRE_AUTHOR','ERROR_1','ERROR_2','ERROR_3
','newline'
  ]

t_ignore = '\t '   #nose que hace pero vi en varios, creo q ignora espacios en
blanco o tabulacion
```

```
t_APERTURA_ARTICLE = r'<article>'
t_CIERRE_ARTICLE = r'</article>'
t_APERTURA_SIMPARA = r'<simpara>'
t_CIERRE_SIMPARA = r'</simpara>'
t_APERTURA_ADDRESS = r'<address>'
t_CIERRE_ADDRESS = r'</address>'
t_APERTURA_MEDIAOBJECT = r'<mediaobject>'
t_CIERRE_MEDIAOBJECT = r'</mediaobject>'
t_APERTURA_COMMENT = r'<comment>'
t_CIERRE_COMMENT = r'</comment>'
t_APERTURA_ABSTRACT = r'<abstract>'
t_CIERRE_ABSTRACT = r'</abstract>'
t_APERTURA_SECTION = r'<section>'
t_CIERRE_SECTION = r'</section>'
t_APERTURA_SIMPLESECT = r'<simplesect>'
t_CIERRE_SIMPLESECT = r'</simplesect>'
t_APERTURA_EMPHASIS = r'<emphasis>'
t_CIERRE_EMPHASIS = r'</emphasis>'
t_APERTURA_AUTHOR = r'<author>'
t_CIERRE_AUTHOR = r'</author>'
t_APERTURA_FIRSTNAME = r'<firstname>'
t_CIERRE_FIRSTNAME = r'</firstname>'
t_APERTURA_SURNAME = r'<surname>'
t_CIERRE_SURNAME = r'</surname>'
t_APERTURA_STREET = r'<street>'
t_CIERRE_STREET = r'</street>'
t_APERTURA_CITY = r'<city>'
t_CIERRE_CITY = r'</city>'
t_APERTURA_STATE = r'<state>'
t_CIERRE_STATE = r'</state>'
t_APERTURA_PHONE = r'<phone>'
t_CIERRE_PHONE = r'</phone>'
t_APERTURA_EMAIL = r'<email>'
t_CIERRE_EMAIL = r'</email>'
t_APERTURA_DATE = r'<date>'
t_CIERRE_DATE = r'</date>'
t_APERTURA_YEAR = r'<year>'
t_CIERRE_YEAR = r'</year>'
t_APERTURA_HOLDER = r'<holder>'
t_CIERRE_HOLDER = r'</holder>'
t_APERTURA_VIDEOOBJECT = r'<videoobject>'
t_CIERRE_VIDEOOBJECT = r'</videoobject>'
t_APERTURA_IMAGENOBJECT = r'<imagenobject>'
t_CIERRE_IMAGENOBJECT = r'</imagenobject>'
t_APERTURA_TGROUP = r'<tgroup>'
t_CIERRE_TGROUP = r'</tgroup>'
t_APERTURA_THEAD = r'<thead>'
t_CIERRE_THEAD = r'</thead>'
t_APERTURA_TFOOT = r'<tfood>'
t_CIERRE_TFOOT = r'</tfood>'
t_APERTURA_TBODY = r'<tbody>'
t_CIERRE_TBODY = r'</tbody>'
t_APERTURA_ENTRYTBL = r'<entrytbl>'
t_CIERRE_ENTRYTBL = r'</entrytbl>'
t_ERROR_1 = r'<[\w]+>'
```

```
t_ERROR_2 = r'<[\w]+\s[\w]+=*[\w"]+\s*/*>'
t_ERROR_3 = r'</[\w]+>'
flags = os.O_RDWR | os.O_CREAT
arch= open("src/html_generados/archivo.html","w",flags,encoding="utf-8")

#funciones
def t_newline(t):
    r'\n+'
    t.lexer.lineno += len(t.value)
def t_DT1(t):
      r'<[!]DOCTYPE\sarticle>'
      arch.write("<!DOCTYPE html>")
def t_TEXTO (t):
    r'[\w._%+?¿¡!()"|°¬$&={}\#@*-]+'  #falta ver caraxteres especiales
    arch.write(f'{t.value} ')
    return (t)
def t_error(t):

 print ("caracter ilegal %s" % t.value[0])
 t.lexer.skip(1)
def t_APERTURA_PARA(t):
      r'<para>'
      arch.write("<p>")
      return(t)
def t_CIERRE_PARA(t):
      r'</para>'
      arch.write("</p>")
      return(t)
def t_APERTURA_TITLE(t):
      r'<title>'
      arch.write("<h1>")
      return(t)
def t_CIERRE_TITLE(t):
      r'</title>'
      arch.write("</h1>")
      return(t)
def t_APERTURA_INFO(t):
      r'<info>'
      arch.write('<div style="color:white;background-color:green;font-
size:8pts"><p>')   #anda bien
      return(t)
def t_CIERRE_INFO(t):
      r'</info>'
      arch.write('</p></div>')
      return(t)
def t_APERTURA_IMPORTANT(t):
      r'<important>'
      arch.write('<div style="background-color:red;color:white">') #anda bien
      return(t)
def t_CIERRE_IMPORTANT(t):
      r'</important>'
      arch.write('</div>')
      return(t)
def t_APERTURA_IMAGEDATA (t):
      r'<imagedata=fileref=""^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
```

```
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      return(t)
def t_APERTURA_VIDEODATA (t):
      r'<videodata=fileref="^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      return(t)
def t_APERTURA_LINK (t):
      r'<link = xlink:href ="^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-
Z]{2,3}(:[a-zA-Z0-9 ]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$">'
      arch.write(f'<a href="{t.value}">esto es un link</a>')
      return (t)
def t_APERTURA_INFORMALTABLE(t):
      r'<informaltable>'
      arch.write("<table>")
      return (t)
def t_CIERRE_INFORMALTABLE(t):
      r'</informaltable>'
      arch.write("</table>")
      return (t)
def t_APERTURA_ROW(t):                                  #un problema con esto es
que en html todos son tr y se diferencian adentro usando
      r'<row>'                                          #<th></th> para los
encabezados y pies de la tabla
      arch.write("<tr>")
      return (t)
def t_CIERRE_ROW(t):
      r'</row>'
      arch.write("</tr>")
      return (t)
def t_APERTURA_ENTRY(t):
      r'<entry>'
      arch.write("<td>")
      return (t)
def t_CIERRE_ENTRY(t):
      r'</entry>'
      arch.write("</td>")
      return (t)
def t_APERTURA_ITEMIZEDLIST(t):
      r'<itemizedlist>'
      arch.write("<ul>")
      return (t)
def t_CIERRE_ITEMIZEDLIST(t):
      r'</itemizedlist>'
      arch.write("</ul>")
      return (t)
def t_APERTURA_LISTITEM(t):
      r'<listitem>'
      arch.write("<il>")
      return (t)
def t_CIERRE_LISTITEM(t):
      r'</listitem>'
      arch.write("</il>")
      return (t)

lexer = lex.lex()
```

main.py

```
from lexer import lexer
import os
import sys
def borrarPantalla(): #Borra lo ya escrito en pantalla
      if os.name == "posix":
            os.system ("clear")
      elif os.name == "ce" or os.name == "nt" or os.name == "dos":
            os.system ("cls")
print ("Hola este es el analizador Lexico")
print ("1 para ingresar datos a mano\n2 si quiere cargar datos desde un
archivo\n")
op = input()
errores = []
if op == "1":                                                  #ingreso manual
      borrarPantalla()
      while True: #ciclo para ingresar datos hasta que eleccion sea 0
            print("ingrese lo que quiere analizar")
            cadena = input()
            lexer.input(cadena)
            while True:
                  tok = lexer.token()

                  if not tok : break
                  if tok.type == "ERROR_1" or tok.type == "ERROR_2" or tok.type
== "ERROR_3":
                        print(f"error lexico en linea {tok.lineno}
({tok.value})")
                  else:
                        print(tok)
            print("desea continuar?\n1 para continuar\n0 para terminar")
            eleccion = input()
            if eleccion == "0": break
            borrarPantalla()

elif op == "2":                                          #ingreso por archivo
      n = 0
      dir = 'prueba/'                    #elegir el archivo
      with os.scandir(dir) as ficheros:
            print(type(ficheros))
            ficheros = [fichero.name for fichero in ficheros if
fichero.is_file()]    #ficheros es una lista con los archivos de la carpeta
prueba
      for j in ficheros:
            if ".txt" in j:
                  ficheros.remove(f"{j}")
      for i in ficheros:

            print(f"{n+1}: {ficheros[n]}")
            n +=1
      print("elegi el archivo para leer")
      op2 = input()
      borrarPantalla()
      if int(op2) <= n:
```

```
            ruta = ficheros[int(op2)-1]
            with open(f"prueba/{ruta}","r",encoding="utf-8") as maestro: #esto
ya funciona para cualquier fichero en prueba/
                  print(f"abierto archivo: {ruta}")
                  lexer.input(maestro.read())
                  while True:
                        tok = lexer.token()

                        if not tok:break
                        if tok.type == "error":
                              errores.append(tok.value)
                        if tok.type == "ERROR_1" or tok.type == "ERROR_2" or
tok.type == "ERROR_3":
                              print(f"error lexico en linea {tok.lineno}
({tok.value})")
                        else:
                              print(tok)
                  cambio = ruta.replace(".xml","")

os.rename("src/html_generados/archivo.html",f"src/html_generados/{cambio}.html"
)

      else:
            print("numero invalido")
else:
      print("volve a empezar")
```

TABLAS_ERROR.xml

```
<!DOCTYPE article>
<article>
    <title>Archivo de prueba sobre tablas </title>

    <abstract>
            <para>Se vera a continuacion una tabla de puntos sobre del top 5 de
la liga de futbol argentino <emphasis> campeonatado 2023 </emphasis>, a modo de
demostracion </para>
    </abstract>

    <informaltable>
        <tgroup>
            <tbody>
                <row>
                    <entry>
                        <important>
                            <title> EQUIPO </title>
                            <title> PUNTOS </title>
                        </important>
                </row>
                <row>
                        <para>  River plate</para>
                        <para> 40 </para>
                </row>
                <row>
                    <entry>
                        <para> San lorenzo</para>
```

```
                        <para> 35 </para>
                    </entry>
                </row>
            <tfood>
            </tbody>
                <row>
                    <entry>
                        <important>
                            <title> CAMPEONATO ARGENTINO 2023 </title>
                        </important>
                    </entry>
                </row>
            </tfood>
        </tgroup>
    </informaltable>

    <setion>
        <para> final del archivo de prueba de tablas </para>
    </sectionmm>
</archivo>
```

TABLAS_ERROR.html

```
<!DOCTYPE html><h1>Archivo de prueba sobre tablas </h1><p>Se vera a continuacion
una tabla de puntos sobre del top 5 de la liga de futbol argentino campeonatado
2023 a modo de demostracion </p><table><tr><td><div style="background-
color:red;color:white"><h1>EQUIPO </h1><h1>PUNTOS </h1></div></tr><tr><p>River
plate </p><p>40 </p></tr><tr><td><p>San lorenzo </p><p>35
</p></td></tr><tr><td><div style="background-
color:red;color:white"><h1>CAMPEONATO ARGENTINO 2023
</h1></div></td></tr></table><p>final del archivo de prueba de tablas </p>
```

ejLista.html

```
<ul><il><p>Hal Computer Systems y O Reilly &amp Associates de 1991 a 1994
</p></il><il><p>El grupo Davenport de 1994 a 1998. </p></il><il><p>El grupo
OASIS /acronym de 1998 hasta hoy. </p></il></ul>
```

nombreprueba.html

```
<!DOCTYPE html><div style="color:white;background-color:green;font-
size:8pts"><p><h1>El titulo del articulo </h1>Juan Perez </p></div><h1>Titulo
para la seccion 1 </h1><p><div style="background-color:red;color:white">Esto es
un parrafo </div></p><p>Otro parrafo. </p>
```

LISTAS.xml

```
<!DOCTYPE article>
<article>
    <info>
        <title>Archivo de prueba presentacion del grupo </title>

        <aabstract>
            <title>integrantes </title>
            <para>A continuacion se mostraran los nombres de los distintos
integrantes del grupo </para>
        </abstract>
```

```
        <author>
            <firtname>Camilo</firtname>
            <surname>Aguirre </surname>
        </author>
        <author>
            <firtname> Joaquin </firtname>
            <surname> Bianciotto </surname>
        </author>
        <author>
            <firtname> Julian </firtname>
            <surname> Colombo </surname>
        </author>
        <author>
            <firtname> Yoel </firtname>
            <surname> Maraim </surname>
        </author>
    </info>

    <comment> correo electronico de contacto <email> grupo4@mail.com </email>
</comment>

    <section>
        <title> Las 3 entregas parciales del tpi son las siguiente: </title>

        <itemizedlist>
            <listitem>
                <para>1er entrega: Documentacia del proyecto y la gramatica a
generar, <emphasis> el 23 de abril </emphasis> </para>
            </listitem>

            <listitem>
                <para>2da entrega: Presentacion del lexer que reconozca los
tokens del leguaje, <emphasis> el 4 de junio </emphasis> </para>
            </listitem>

            <listitem>
                <para>3er entrega (entrega final): presentacion del tpi
completo, lexer y parse, incluyendo toda la presentacion del tpi antes de la
clase con una exposicion hora de no menos de 20 minutos <emphasis> el 2 junio
</emphasis> </para>
            </listitem>
        </itemizedlist>

        <para>Con esto se concluye el archivo de prueba, donde se experimento
con distinitas etiquetas anidadas, como asi tmb listas</para>
    </section>
</article>
```

LISTAS.html

```
<!DOCTYPE html><div style="color:white;background-color:green;font-
size:8pts"><p><h1>Archivo de prueba presentacion del grupo </h1><h1>integrantes
</h1><p>A continuacion se mostraran los nombres de los distintos integrantes del
grupo </p>Camilo firtname Aguirre author Joaquin firtname Bianciotto author
Julian firtname Colombo author Yoel firtname Maraim author </p></div>correo
```

```
electronico de contacto grupo4 mail com <h1>Las entregas parciales del tpi son
las siguiente </h1><ul><il><p>er entrega Documentacia del proyecto y la
gramatica a generar el de abril </p></il><il><p>da entrega Presentacion del
lexer que reconozca los tokens del leguaje el de junio </p></il><il><p>er
entrega entrega final presentacion del tpi completo lexer y parse incluyendo
toda la presentacion del tpi antes de la clase con una exposicion hora de no
menos de minutos el junio </p></il></ul><p>Con esto se concluye el archivo de
prueba donde se experimento con distinitas etiquetas anidadas como asi tmb
listas </p>
```

Docu_error_tablas.txt.html

```
El lexer a sido sometido a los siguientes errores del archivos TABLAR ERROR xml
linea se abre la etiqueta <td>pero nunca se cierra linea directamente se escribe
una etiqueta <p>despues de <tr>sin haber estado antes la etiqueta <td>linea se
abre la etiqueta sin antes cerrar la etiqueta tbody linea error en la palabra se
escribe setion en vez de
```

LISTAS_ERROR.html

```
<!DOCTYPE html><div style="color:white;background-color:green;font-
size:8pts"><p><h1>Entregas tpi </h1><h1>integrantes </h1><p>Integrantes del
grupo n </p></p></div>correo electronico de contacto grupo4 mail com <h1>Las
entregas parciales del tpi son las siguiente </h1><ul><il><ul><p>er entrega
Documentacia del proyecto y la gramatica a generar el de abril
</p></ul></il>lisitem <p>da entrega Presentacion del lexer que reconozca los
tokens del leguaje el de junio </p></il><il><p>er entrega entrega final
presentacion del tpi completo lexer y parse incluyendo toda la presentacion del
tpi antes de la clase con una exposicion hora de no menos de minutos el junio
</p></il></ul><p>Con esto se concluye el archivo de prueba donde se experimento
con distinitas etiquetas anidadas como asi tmb listas </p>
```