

# Índice

[**Índice 1**](#_3gtgayab5kce)

[**Introducción 3**](#_41y9git6r7w0)

[**Lenguaje de pseudocódigo 4**](#_t9jeim8hgbku)

[Introducción 4](#_6jqzkumrdyr7)

[Símbolos de la gramática 5](#_keh974c29n24)

[Etiquetas estructurales del documento 5](#_bhecerl9uify)

[Secciones 6](#_sitwmulncnz0)

[Etiquetas básicas de párrafo 6](#_u3qlmue25fl4)

[Imágenes y multimedia 7](#_vd745sgreyr)

[Listas 7](#_ilvpigr9ti6o)

[Tablas 7](#_y03mm0mecgol)

[Forma de Conversión a HTML 7](#_lmg5k1msbs11)

[**Gramática 10**](#_giz0sbhbw3v6)

[Símbolos Terminales 10](#_sqrqd4ygkrjo)

[Símbolos No terminales 14](#_hr2699jszgo2)

[Reglas de producción de la gramática 21](#_8vwpjwscqmuq)

[**Análisis Léxico 35**](#_m9u5mpaxoomy)

[HERRAMIENTAS 35](#_vr7zuofcu71v)

[CÓDIGO 36](#_8dv1ijoskjyf)

[**Análisis Sintáctico 43**](#_y2rsc88v4sa6)

[Herramientas 43](#_evq25fmn7b7u)

[Código 44](#_3wn0se45xq5y)

[**Funciones Auxiliares 62**](#_v64zyhiir10w)

[**Ejemplos 66**](#_z9830maxeg7q)

[**Conclusiones 67**](#_9q92vpht4dfn)

[**Solución de Errores 67**](#_xx0wbhio3b35)

[**Bibliografía o Referencias web 69**](#_gzt9cv9279pz)

# 

# Introducción

En este trabajo, se aborda el análisis, validación y transformación de un subconjunto de etiquetas del lenguaje de marcado DocBook/XML utilizando un generador de analizadores léxicos y sintácticos. El objetivo es construir una utilidad que reciba un archivo en formato XML y contenido DocBook, y determine si está bien construido según el estándar, identificando cualquier error presente. Además, se requiere transformar el contenido analizado en un documento HTML.

El proceso se divide en dos etapas fundamentales: el análisis léxico y el análisis sintáctico. El analizador léxico se encarga de convertir la secuencia de caracteres del programa en una secuencia lógica de tokens. Por su parte, el analizador sintáctico recibe los tokens generados y verifica si cumplen con la gramática definida para el subconjunto de etiquetas de DocBook/XML.

Para la implementación de esta solución, se ha utilizado el lenguaje de programación Python y la biblioteca PLY (Python Lex-Yacc). PLY proporciona las herramientas necesarias para generar analizadores léxicos y sintácticos de manera eficiente y sencilla. Al aprovechar estas capacidades, se logra una solución flexible y modular, capaz de analizar y validar documentos DocBook/XML, detectando errores y transformándolos en documentos HTML válidos.

En resumen, este trabajo presenta una solución que combina el uso de un generador de analizadores léxicos y sintácticos con el lenguaje de programación Python y la biblioteca PLY para construir un analizador léxico y sintáctico que permita analizar, validar y transformar un subconjunto de etiquetas de DocBook/XML en documentos HTML. La solución proporciona información sobre la validez del documento analizado y ofrece la posibilidad de generar diversas salidas según los criterios establecidos.

En la implementación de la solución, tuvimos diferentes problemas que resolvimos, explicados en la conclusión.

Para la implementación de la solución, utilizamos las siguientes herramientas externas, que se describe su funcionamiento en su apartado correspondiente, ya sea en el apartado de la documentación del analizador sintáctico o en la del analizador lexico.

Python 3.11.1

Bibliotecas:

PLY 3.11

Re

Tkinter

# Lenguaje de pseudocódigo

Esta son las características de DocType entregada por la cátedra de Sintaxis y Semántica de los Lenguajes de la Universidad Tecnológica Nacional de Resistencia en 2023, que nos permitió comprender la estructura del lenguaje de programación para poder llevar a cabo el trabajo integrador, para primero desarrollar la gramática, y luego generar el lexer con sus respectivos tokens que debía reconocer, para llegar al objetivo de poder codificar y ejecutar el parser, encargado del análisis sintáctico de un archivo o de un código escrito por el usuario.

## Introducción

Como lenguaje semántico que es, DocBook nos permite crear documentos en un formato neutro, independiente de la presentación. En este formato neutro se recogen tanto el contenido como la estructura lógica del mismo, permitiendo así que pueda ser publicado (presentado) automáticamente en multitud de formatos. Los documentos DocBook no describen ni la apariencia ni la presentación de sus contenidos, sino únicamente el sentido de dichos contenidos. Por ejemplo, en lugar de indicar exactamente cómo ha de visualizarse una determinada frase que es el título de un capítulo, DocBook simplemente indica que dicha frase es un título de capítulo. Un documento DocBook contiene básicamente texto. Ahora bien, hay texto que sirve para indicar elementos, describir el contenido y propiedades de la página. Es decir, el texto contiene lo que se denomina etiquetas o tags. Las etiquetas sirven para delimitar elementos de la página; por ejemplo, un párrafo es un elemento de la página, una tabla también. Incluso hay elementos que contienen otros elementos (las tablas constan de filas y las filas de celdas, por ejemplo). Las etiquetas son textos encerrados entre los signos de mayor y menor (< >). Cuando un navegador encuentra un texto así encerrado, entenderá que dentro de los símbolos menor y mayor lo que se indica es el inicio de un elemento. El inicio del elemento se marca con el nombre del elemento entre los símbolos < y >. La mayoría de etiquetas afectan a un determinado texto, el cual estará encerrado por las etiquetas. Por lo tanto, casi siempre, existe una etiqueta de apertura y otra de cierre, que se interpretará como el inicio y fin de un determinado elemento de la página. El cierre de una etiqueta se marca poniendo el símbolo /, seguido del nombre del elemento que estamos cerrando. Ejemplo:

<para>Texto normal <emphasis>texto destacado</emphasis></para>

Lo que indicamos en el ejemplo anterior es que un párrafo de texto contiene un elemento de tipo emphasis, que contiene el texto: texto destacado.

Atributos

Algunas etiquetas tienen atributos. Los atributos son propiedades de cada elemento a las que podemos asignar un valor, de modo que dicho valor varía el comportamiento del elemento. Estos atributos pueden definir, por ejemplo, la dirección del texto o el lenguaje usado dentro del elemento o información consultiva para su correcta interpretación, entre muchas otras. La forma de utilizar atributos es:

<elemento nombreDeAtributo1=”valor1” nombreDeAtributo2=”valor1” … >

A los atributos se les asignan valores que deben ir entre comillas dobles. El símbolo de igualdad (=) es obligatorio.

Tipos de datos

-Cadena: Estará compuesta por letras, números, signos de puntuación, caracteres especiales.

Aparecen dentro de los principales elementos , títulos, descripción, categoría,etc.

-URL: Los únicos caracteres permitidos en URL son letras, números, guión medio, guión bajo y punto. además de los caracteres reservados: # , /, :, &, ?, =

## Símbolos de la gramática

### Etiquetas estructurales del documento

**ARTICLE :=**((Info)?, (Title)?, ((ItemizedList|Important|Para|SimPara|Address|MediaObject|InformalTable|Comment| Abstract|)+, (SimpleSect\*|Section\*))

**SECTION:=** (Info)?, (Title)?,

((ItemizedList|Important|Para| SimPara|Address|MediaObject|InformalTable|Comment| Abstract|)+,

(SimpleSec**\***|Section\*))

### Secciones

**Section** := (Info?, (Title)?,  
 ((ItemizedList|Important|Para| SimPara|Address|MediaObject|InformalTable|Comment| Abstract|)+,   
(SimpleSec\*|Section\*))   
  
**SimpleSec** := (Info?, (Title)?,   
(ItemizedList|Important|Para|SimPara|Address|MediaObject| InformalTable|Comment| Abstract|)+

### Etiquetas básicas de párrafo

**Info :=** (MediaObject| Abstract|Address|Author|Date|Copyright|Title)+)

**Abstract :=** (Title?, (Para|SimPara)+   
 **Address :=** (#texto|Street|City|State|Phone|Email)\*   
 **Author :=** ((FirstName|Surname|)+ Copyright := (Year+,Holder\*)   
 **Title :=** (#texto|Emphasis|Link|Email)+)   
  
**SimPara, Emphasis, Comment, Link :=**   
((#texto|Emphasis|Link|Email|Author|Comment|)+)   
  
**Para:=**   
(#texto|Emphasis|Link|Email|Author|Comment|ItemizedList| |Important|Address|MediaObject|InformalTable)+  
  
 **Important :=**(Title?,((ItemizedList|Important|Para|SimPara| Address|MediaObject|InformalTable|Comment|Abstract)+)  
 FirstName,Surname, Street, City, State, Phone,Email, Date,Year,Holder := (#texto|Link|Emphasis| Comment|)+

### Imágenes y multimedia

**MediaObject :=**(Info?, (VideoObject|ImageObject), (VideoObject|ImageObject)\*, ) **VideoObject :=** (Info?,VideoData)   
**ImageObject :=** (Info?,ImageData)

### Listas

**ItemizedList :=** (ListItem+)

**ListItem :=** (ItemizedList|Important|Para|SimPara|Address|MediaObject| |InformalTable|Comment|Abstract)+

### Tablas

**InformalTable :=** (MediaObject+|TGroup+)  
 tgroup := (thead?,tfoot?,tbody) thead, tfoot, tbody := (row+)   
**row :=** ((entry| entrytbl)+)  
 **entrytbl :=** (thead?,tbody)   
**entry :=** (#texto| ItemizedList|Important|Para| SimPara|MediaObject| |Comment| Abstract|)+

## Forma de Conversión a HTML

| Descripción de Etiqueta | Tag en Doctype | Descripción de Transformación | Tag en HTML |
| --- | --- | --- | --- |
| Nombre de archivo | <!DOCTYPE ARTICLE> | Igual al archivo fuente XML pero con extensión HTML. | <!DOCTYPE html> |
| Título del documento | <Title>...</Title> | Encerrados entre tags <H1> | <H1>...</H1> |
| Título de secciones | <Section>...</Section>  <SimSection>...</SimSection> | encerrados entre tags <H2> | <H2>...</H2> |
| Etiquetas dentro de info | <Info>...</Info> | Párrafos con fondo verde , color de letra blanco y tamaño 8pts. | <Info> Contenido…  </Info> |
| Important | <Important>  ...  </Important> | Color de fondo rojo y todo contenido de texto dentro ,texto color blanco. | <Important>  Contenido…  </Important> |
| Etiquetas para y simpara | <Para>...</Para <SimPara>...</SimPara> | Se traducen como párrafos p | <p>...</p> |
| Etiqueta link | <Link>...</Link> | Se traduce como tag html a atributo xlink:href | <Link xlink:href= ”#texto”> </Link> |
| Tablas | <thead>...</thead>  <tfoot>...</tfoot>  <tbody>...</tbody> | Partes principales de una tabla:  **thead** → Sirve para indicar las filas que forman la cabecera de la tabla  **tfoot** → Indica el pie de la tabla  **tbody** → Indica el cuerpo de la tabla | <th> </th>  <td> </td>  <tr> </tr> |
| Lista ItemizedList | <itemizedlist>...</itemizedlist> | Deben traducirse como listas html , ul. | <ul> </ul> |
| Lista ListItem | <listitem>...</listitem> | Deben traducirse como listas html, li | <li> </li> |

# 

# Gramática

A partir del material de la cátedra especificado en el apartado anterior, llevamos a cabo la primera entrega del TPI especificando la gramática. Aunque luego al codificar el parser está cambio rotundamente debido a que al ponerla en práctica nos dimos cuenta de distintos errores, la gramática que se encuentra a continuación es la gramática pre cambiar la recursión por izquierda, debido a que eso se realizó en última instancia

## Símbolos Terminales

| Símbolos Terminales | Descripción |
| --- | --- |
| <info> | Abre la etiqueta de Info. |
| </info> | Cierra etiqueta de Info. |
| <title> | Abre la etiqueta de title. |
| </title> | Cierra etiqueta de title. |
| <ItemizedList> | Abre la etiqueta de ItemizedList. |
| </ItemizedList> | Cierra etiqueta de ItemizedList. |
| <Important> | Abre la etiqueta de Important. |
| </Important> | Cierra etiqueta de Important. |
| <Para> | Abre la etiqueta de Para. |
| </Para> | Cierra etiqueta de Para. |
| <SimPara> | Abre la etiqueta de SimPara. |
| </SimPara> | Cierra etiqueta de SimPara. |
| <MediaObject> | Abre la etiqueta de MediaObject. |
| </MediaObject> | Cierra etiqueta de MediaObject. |
| <InformalTable> | Abre la etiqueta de InformalTable. |
| </InformalTable> | Cierra etiqueta de InformalTable. |
| <Comment> | Abre la etiqueta de Comment. |
| </Comment> | Cierra etiqueta de Comment. |
| <Abstract> | Abre la etiqueta de Abstract. |
| </Abstract> | Cierra etiqueta de Abstract. |
| <SimpleSec> | Abre la etiqueta de SimpleSec. |
| </SimpleSec> | Cierra etiqueta de SimpleSec. |
| <Section> | Abre la etiqueta de Section. |
| </Section> | Cierra etiqueta de Section. |
| <Address> | Abre la etiqueta de Adress |
| </Address> | Cierra la etiqueta de Adress |
| <Author> | Abre la etiqueta de Author |
| </Author> | Cierra la etiqueta de Author |
| <Date> | Abre la etiqueta de Date |
| </Date> | Cierra la etiqueta de Date |
| <CopyRight> | Abre la etiqueta de Copyright |
| </CopyRight> | Cierra la etiqueta de Copyright |
| <Firstname> | Abre la etiqueta de Firstname |
| </Firstname> | Cierra la etiqueta de Firstname |
| <Surname> | Abre la etiqueta de Surname |
| </Surname> | Cierra la etiqueta de Surname |
| <texto> | Abre la etiqueta de texto |
| </texto> | Cierra la etiqueta de texto |
| <Street> | Abre la etiqueta de Street |
| </Street> | Cierra la etiqueta de Street |
| <Phone> | Abre la etiqueta de Phone |
| </Phone> | Cierra la etiqueta de Phone |
| <Email> | Abre la etiqueta de Email |
| </Email> | Cierra la etiqueta de Email |
| <Year> | Abre la etiqueta de Year |
| </Year> | Cierra la etiqueta de Year |
| <Holder> | Abre la etiqueta de Holder |
| </Holder> | Cierra la etiqueta de Holder |
| <Emphasis> | Abre la etiqueta de Emphasis |
| </Emphasis> | Cierra la etiqueta de Emphasis |
| <Link> | Abre la etiqueta de Link |
| </Link> | Cierra la etiqueta de Link |
| <Comment> | Abre la etiqueta de Comment |
| </Comment> | Cierra la etiqueta de Comment |
| <row> | Abre la etiqueta de row |
| </row> | Cierra la etiqueta de row |
| <entry> | Abre la etiqueta de entry |
| </entry> | Cierra la etiqueta de entry |
| <entrytbl> | Abre la etiqueta de entrytbl |
| </entrytbl> | Cierra la etiqueta de entrytbl |
| <mediaobject> | Abre la etiqueta de mediaobject |
| </mediaobject> | Cierra la etiqueta de mediaobject |
| <videoobject> | Abre la etiqueta de videoobject |
| </videoobject> | Cierra la etiqueta de videoobject |
| <imagenobject> | Abre la etiqueta de imagenobject |
| </imagenobject> | Cierra la etiqueta de imagenobject |
| <listitem> | Abre la etiqueta de listitem |
| </listitem> | Cierra la etiqueta de listitem |
| <tgroup> | Abre la etiqueta de tgroup |
| </tgroup> | Cierra la etiqueta de tgroup |
| <thead> | Abre la etiqueta de thead |
| </thead> | Cierra la etiqueta de thead |
| <tbody> | Abre la etiqueta de tbody |
| </tbody> | Cierra la etiqueta de tbody |
| <tfoot> | Abre la etiqueta de tfoot |
| </tfoot> | Cierra la etiqueta de tfoot |

## Símbolos No terminales

| Símbolos no Terminales | DESCRIPCIÓN |
| --- | --- |
| **ETIQUETAS para Estructura de Documento** | |
| DOCUMENT |  |
| ARTICLE | Permite abrir la etiqueta Info (y obliga a ir a Article1) o pasar directamente al no Terminal Article 1. |
| ARTCICLE1 | Permite abrir la etiqueta Title (y obliga a ir a Article2) o pasar directamente al no Terminal Article 2. |
| ARTICLE2 | Permite abrir todas las etiquetas estructurales (de forma alternativa), que pueden aparecer 1 o más veces. Tiene su caso recursivo que permite seguir abriendo dichas etiquetas, caso base, y una opción de ir al no terminal Article3section y otra de ir al no terminal Article3simsection. |
| ARTICLE3SECTION | Permite crear SECTION cero o mas veces, en el caso de ser cero, el no terminal ARTICLE2 crea el caso base de sus respectivos no terminales |
| ARTICLE3SIMSECTION | Permite crear SIMPLESEC cero o mas veces, en el caso de ser cero, el no terminal ARTICLE2 crea el caso base de sus respectivos no terminales. |
| SECTION | Permite abrir la etiqueta Info (y obliga a ir a SECTION1) o pasar directamente al no terminal SECTION1. |
| SECTION1 | Permite abrir la etiqueta Title (y obliga a ir a SECTION2) o pasar directamente al no Terminal SECTION2. |
| SECTION2 | Permite abrir todas las etiquetas estructurales (de forma alternativa), que pueden aparecer 1 o más veces. Tiene su caso recursivo que permite seguir abriendo dichas etiquetas, caso base, y una opción de ir al no terminal SECTION3SECTION y otra de ir al no terminal SECTION3SIMSECTION. |
| SECTION3SECTION | Permite crear SECTION cero o mas veces, en el caso de ser cero, el no terminal SECTION2 crea el caso base de sus respectivos no terminales |
| SECTION3SIMSECTION | Permite crear SIMPLESEC cero o mas veces, en el caso de ser cero, el no terminal SECTION2 crea el caso base de sus respectivos no terminales. |
| SIMPLESEC | Permite abrir la etiqueta Info (y obliga a ir a SIMPLESEC1) o pasar directamente al no Terminal SIMPLESEC1. |
| SIMPLESEC1 | Permite abrir la etiqueta Title (y obliga a ir a Article2) o pasar al no Terminal SIMPLESEC2. |
| SIMPLESEC2 | Permite abrir todas las etiquetas estructurales (de forma alternativa), que pueden aparecer 1 o más veces. Tiene su caso recursivo que permite seguir abriendo dichas etiquetas, caso base. En este caso, no se puede seguir creando etiquetas de secciones. |
| **ETIQUETAS BASICAS DE PÁRRAFO** | |
| INFO | Permite abrir las etiquetas que pueden aparecer como alternativas (se puede pensar como (A U B)\*sin incluir al string vacío), necesariamente se debe abrir como minimo una etiqueta. |
| ABSTRACT | Permite abrir la etiqueta de Title, que puede aparecer o no, pasa obligatoriamente al no terminal ABSTRACT2 |
| ABSTRACT2 | Permite abrir los no terminales PARA o SIMPARA una o muchas veces. |
| AUTHOR | Crea los no terminales FIRSTNAME o SURNAME una o muchas veces. |
| ADDRESS | Permite crear los no terminales text, STREET, CITY, STATE, PHONE, EMAIL una o muchas veces. |
| EMPTY |  |
| COPYRIGHT | Obligatoriamente crea una etiqueta de año y pasa o no al no terminal COPYRIGHT2. |
| COPYRIGHT2 | Permite abrir Holder cero o mas veces, en el caso de cero veces, el no terminal copyright hace el caso base de year |
| TITLE | Permite crear etiquetas, de manera alternativa pero obligatoriamente una etiqueta como mínimo debe aparecer, las etiquetas que generan son: Texto, Street, Phone y Email. |
| SIMPARA | Abre las etiquetas que permite crear 1 (como mínimo) o más veces distintas etiquetas, sin respetar orden. Las etiquetas que permiten abrir son: TEXTO, EMPHASIS, LINK, EMAIL, AUTHOR, COMMENT, con sus respectivos no terminales. |
| EMPHASIS | Abre las etiquetas que permite crear 1 (como mínimo) o más veces distintas etiquetas, sin respetar orden. Las etiquetas que permiten abrir son: TEXTO, EMPHASIS, LINK, EMAIL, AUTHOR, COMMENT, con sus respectivos no terminales. |
| COMMENT | Abre las etiquetas que permite crear 1 (como mínimo) o más veces distintas etiquetas, sin respetar orden. Las etiquetas que permiten abrir son: TEXTO, EMPHASIS, LINK, EMAIL, AUTHOR, COMMENT, con sus respectivos no terminales. |
| LINK | Abre las etiquetas que permite crear 1 (como mínimo) o más veces distintas etiquetas, sin respetar orden. Las etiquetas que permiten abrir son: TEXTO, EMPHASIS, LINK, EMAIL, AUTHOR, COMMENT, con sus respectivos no terminales. |
| PARA | Permite crear las distintas etiquetas básicas para un párrafo, que pueden venir 1 o más veces, sin respetar orden. |
| IMPORTANT | Permite crear una etiqueta de titulo que puede aparecer o no, obliga a ir al no terminal Important2. |
| IMPORTANT2 | Luego de haber creado o no un titulo, permite crear distintas etiquetas basicas que pueden venir 1 o más veces, sin respetar el orden (se podría pensar como (A U B)\* sin incluir al string vacio).. |
| FIRSTNAME | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| SURNAME | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| STREET | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| CITY | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| STATE | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| PHONE | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| EMAIL | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| DATE | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| YEAR | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |
| HOLDER | Permite crear distintas etiquetas, que pueden aparecer sin respetar el orden, pero obligatoriamente debe aparecer como minimo 1 vez una etiqueta (se podría pensar como (A U B)\*, pero sin incluir al string vacío). |
| MEDIAOBJECT | Permite crear o no info, pasa obligatoriamente a MEDIADATA |
| MEDIADATA | Crea obligatoriamente VIDEOOBJECT o MEDIAOBJECT, pasa obligatoriamente a MEDIADATA2 |
| MEDIADATA2 | Permite crear o no VIDEOOBJECT y MEDIAOBJECT cero o mas veces, en caso de ser cero, el no terminal MEDIADATA crea VIDEOOBJECT o MEDIAOBJECT sin pasar a MEDIADATA2 |
| VIDEOOBJECT | Permite crear o no info, pasa obligatoriamente a VIDEOOBJECT2 |
| IMAGEOBJECT | Permite crear o no info, pasa obligatoriamente a IMAGEOBJECT2 |
| VIDEOOBJECT2 | Crea obligatoriamente VIDEODATA |
| IMAGEOBJECT2 | Crea obligatoriamente IMAGENDATA |
| INFORMALTABLE | Crea obligatoriamente TABLEOBJECT o TABLEGROUP |
| TABLEOBJECT | Crea obligatoriamente MEDIAOBJECT una o mas veces |
| TABLEGROUP | Crea obligatoriamente TGROUP una o mas veces |
| TGROUP | Crea o no THEAD, pasa al no terminal TGROUP1 |
| TGROUP1 | Crea o no TFOOT, pasa al no terminal TGROUP2 |
| TGROUP2 | Crea obligatoriamente el terminal TBODY |
| THEAD | Crea obligatoriamente, una o más veces ROW |
| TFOOT | Crea obligatoriamente, una o mas veces ROW |
| TBODY | Crea obligatoriamente una o mas veces ROW |
| ROW | Crea obligatoriamente ENTRY o ENTRYTBL una o más veces |
| ENTRYTBL | Permite generar o no THEAD, pasa obligariamente a ENTRYTBL1 |
| ENTRYTBL1 | Genera el no terminal TBODY |
| ENTRY | Permite crear las no terminales text, ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, MEDIAOBJECT, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como mínimo 1 vez. |
| ITEMIZEDLIST | Crea el no terminal LISTITEM una o mas veces |
| LISTITEM | Permite crear las los no terminales ITEMIZEDLIST,IMPORTANT ,PARA,SIMPARA, ADDRESS, MEDIAOBJECT, INFORMALTABLE, COMMENT, ABSTRACT, que pueden aparecer sin respetar el orden, pero obligatoriamente alguna de ellas deben aparecer como minimo 1 vez. |

## Reglas de producción de la gramática

**<Article>ARTICLE</Article>**

**ARTICLE** → <Info>**INFO**</Info>**ARTICLE 1**

**ARTICLE** → **ARTICLE 1**

**ARTICLE1** → <title>**TITLE**</title>**ARTICLE 2**

**ARTICLE 1** → **ARTICLE 2**

**ARTICLE2**→ <ItemizedList>**ITEMIZEDLIST**</ItemizedList> **ARTICLE2**

**ARTICLE2**→ <Important>**IMPORTANT**</Important> **ARTICLE2**

**ARTICLE2** → <Para>**PARA**</Para> **ARTICLE2**

**ARTICLE2** → <SimPara>**SIMPARA**</SimPara> **ARTICLE2**

**ARTICLE2** → <Address>**ADDRESS**</Address> **ARTICLE2**

**ARTICLE2** → <MediaObject>**MEDIAOBJECT**</MediaObject> **ARTICLE2**

**ARTICLE2** → <InformalTable>**INFORMALTABLE**</InformalTable> **ARTICLE2**

**ARTICLE2** → <Comment>**COMMENT**</Comment> **ARTICLE2**

**ARTICLE2** → <Abstract>**ABSTRACT**</Abstract> **ARTICLE2**

**ARTICLE2**→ <ItemizedList>**ITEMIZEDLIST**</ItemizedList>

**ARTICLE2**→ <Important>**IMPORTANT**</Important>

**ARTICLE2** → <Para>**PARA**</Para>

**ARTICLE2** → <SimPara>**SIMPARA**</SimPara>

**ARTICLE2** → <Address>**ADDRESS**</Address>

**ARTICLE2** → <MediaObject>**MEDIAOBJECT**</MediaObject>

**ARTICLE2** → <InformalTable>**INFORMALTABLE**</InformalTable>

**ARTICLE2** → <Comment>**COMMENT**</Comment>

**ARTICLE2** → <Abstract>**ABSTRACT**</Abstract>

**ARTICLE2**→ <ItemizedList>**ITEMIZEDLIST**</ItemizedList> **ARTICLE3SECTION**

**ARTICLE2**→ <Important>**IMPORTANT**</Important> **ARTICLE3SECTION**

**ARTICLE2** → <Para>**PARA**</Para> **ARTICLE3SECTION**

**ARTICLE2** → <SimPara>**SIMPARA**</SimPara> **ARTICLE3SECTION**

**ARTICLE2** → <Address>**ADDRESS**</Address> **ARTICLE3SECTION**

**ARTICLE2** → <MediaObject>**MEDIAOBJECT**</MediaObject> **ARTICLE3SECTION**

**ARTICLE2** → <InformalTable>**INFORMAL TABLE**</InformalTable> **ARTICLE3SECTION**

**ARTICLE2** → <Comment>**COMMENT**</Comment> **ARTICLE3SECTION**

**ARTICLE2** → <Abstract>**ABSTRACT**</Abstract> **ARTICLE3SECTION**

**ARTICLE2**→ <ItemizedList>**ITEMIZEDLIST**</ItemizedList> **ARTICLE3SIMSECTION**

**ARTICLE2**→ <Important>**IMPORTANT**</Important> **ARTICLE3SIMSECTION**

**ARTICLE2** → <Para>**PARA**</Para> **ARTICLE3SIMSECTION**

**ARTICLE2** → <SimPara>**SIMPARA**</SimPara> **ARTICLE3SIMSECTION**

**ARTICLE2** → <Address>**ADDRESS**</Address> **ARTICLE3SIMSECTION**

**ARTICLE2** → <MediaObject>**MEDIAOBJECT**</MediaObject> **ARTICLE3SIMSECTION**

**ARTICLE2** → <InformalTable>**INFORMAL TABLE**</InformalTable> **ARTICLE3SIMSECTION**

**ARTICLE2** → <Comment>**COMMENT**</Comment> **ARTICLE3SIMSECTION**

**ARTICLE2** → <Abstract>**ABSTRACT**</Abstract>**ARTICLE3SIMSECTION**

**ARTICLE3SECTION** → <Section>**SECTION**</Section>**ARTICLE3SECTION**

**ARTICLE3** → <Section>**SECTION**</Section>

**ARTICLE3SIMSECTION** → <SimpleSect>**SIMPLESECT**</SimpleSect>**ARTICLE3SIMSECTION**

**ARTICLE3SIMSECTION** → <SimpleSect>**SIMPLESECT**</SimpleSect>

--------------------------------------------------------------------------------------------------------

**SECTION → <**Info**>INFO</**Info**>SECTION 1**

**SECTION → SECTION 1**

**SECTION1 → <**title**>TITLE</**title**>SECTION 2**

**SECTION 1 → SECTION 2**

**SECTION2→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList> **SECTION2**

**SECTION2→** <Important>**IMPORTANT**</Important> **SECTION2**

**SECTION2 →** <Para>**PARA**</Para> **SECTION2**

**SECTION2 →** <SimPara>**SIMPARA**</SimPara> **SECTION2**

**SECTION2 →** <Address>**ADDRESS**</Address> **SECTION2**

**SECTION2 →** <MediaObject>**MEDIAOBJECT**</MediaObject> **SECTION2**

**SECTION2 →** <InformalTable>**INFORMALTABLE**</InformalTable> **SECTION2**

**SECTION2 →** <Comment>**COMMENT**</Comment> **SECTION2**

**SECTION2 →** <Abstract>**ABSTRACT**</Abstract> **SECTION2**

**SECTION2→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList>

**SECTION2→** <Important>**IMPORTANT**</Important>

**SECTION2 →** <Para>**PARA**</Para**>**

**SECTION2 →** <SimPara>**SIMPARA**</SimPara>

**SECTION2 →** <Address>**ADDRESS**</Address>

**SECTION2 →** <MediaObject>**MEDIAOBJECT**</MediaObject>

**SECTION2 →** <InformalTable>**INFORMALTABLE**</InformalTable>

**SECTION2 →** <Comment>**COMMENT**</Comment>

**SECTION2 →** <Abstract>**ABSTRACT**</Abstract>

**SECTION2→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList>**SECTION3SECTION**

**SECTION2→** <Important>**IMPORTANT**</Important> **SECTION3SECTION**

**SECTION2 →** <Para>**PARA**</Para>**SECTION3SECTION**

**SECTION2 →** <SimPara>**SIMPARA</**SimPara> **SECTION3SECTION**

**SECTION2 →** <Address>**ADDRESS**</Address> **SECTION3SECTION**

**SECTION2 →** <MediaObject>**MEDIAOBJECT**</MediaObject> **SECTION3SECTION**

**SECTION2 →** <InformalTable>**INFORMALTABLE**</InformalTable> **SECTION3SECTION**

**SECTION2 →** <Comment>**COMMENT**</Comment>**SECTION3SECTION**

**SECTION2 →** <Abstract>**ABSTRACT**</Abstract>**SECTION3SECTION**

**SECTION2→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList>**SECTION3SIMSECTION**

**SECTION2→** <Important>**IMPORTANT**</Important> **SECTION3SIMSECTION**

**SECTION2 →** <Para>**PARA**</Para>**SECTION3SIMSECTION**

**SECTION2 →** <SimPara>**SIMPARA</**SimPara> **SECTION3SIMSECTION**

**SECTION2 →** <Address>**ADDRESS**</Address> **SECTION3SIMSECTION**

**SECTION2 →** <MediaObject>**MEDIAOBJECT**</MediaObject> **SECTION3SIMSECTION**

**SECTION2 →** <InformalTable>**INFORMALTABLE**</InformalTable> **SECTION3SIMSECTION**

**SECTION2 →** <Comment>**COMMENT**</Comment>**SECTION3SIMSECTION**

**SECTION2 →** <Abstract>**ABSTRACT**</Abstract>**SECTION3SIMSECTION**

**SECTION3SECTION →** <Section**>SECTION**</Section>**SECTION3SECTION**

**SECTION3SECTION →** <Section**>SECTION**</Section>

**SECTION3SIMSECTION→** <SimpleSect>**SIMPLESECT**</SimpleSect>**SECTION3SIMSECTION**

**SECTION3SIMSECTION→** <SimpleSect>**SIMPLESECT**</SimpleSect>

**-----------------------------------------------------------------------------------------------------------------------**

**SIMPLESEC →** <Info>**INFO**</Info>**SIMPLESEC1**

**SIMPLESEC → SIMPLESEC1**

**SIMPLESEC1 →** <title>**TITLE**</title>**SIMPLESEC2**

**SIMPLESEC1 → SIMPLESEC2**

**SIMPLESEC2→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList> **SIMPLESEC2**

**SIMPLESEC2→** <Important>**IMPORTANT**</Important> **SIMPLESEC2**

**SIMPLESEC2 →** <Para>**PARA**</Para> **SIMPLESEC2**

**SIMPLESEC2 →** <SimPara>**SIMPARA**</SimPara> **SIMPLESEC2**

**SIMPLESEC2 →** <Address>**ADDRESS**</Address> **SIMPLESEC2**

**SIMPLESEC2 →** <MediaObject>**MEDIAOBJECT**</MediaObject> **SIMPLESEC2**

**SIMPLESEC2 →** <InformalTable>**INFORMALTABLE</**InformalTable**> SIMPLESEC2**

**SIMPLESEC2 →** <Comment>**COMMENT**</Comment> **SIMPLESEC2**

**SIMPLESEC2 →** <Abstract>**ABSTRACT**</Abstract> **SIMPLESEC2**

**SIMPLESECT2→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList>

**SIMPLESECT2→** <Important>**IMPORTANT**</Important>

**SIMPLESECT2 →** <Para>**PARA**</Para>

**SIMPLESECT2 →** <SimPara>**SIMPARA**</SimPara>

**SIMPLESECT2 →** <Address>**ADDRESS**</Address>

**SIMPLESECT2 →** <MediaObject>**MEDIAOBJECT**</MediaObject>

**SIMPLESECT2 →** <InformalTable>**INFORMALTABLE**</InformalTable>

**SIMPLESECT2 →** <Comment>**COMMENT**</Comment>

**SIMPLESECT2 →** <Abstract>**ABSTRACT**</Abstract>

**--------------------------------------------------------------------------------------------------------------------------**

**INFO →** <MediaObject>**MEDIAOBJECT**</MediaObject>**INFO**

**INFO →**<Abstract>**ABSTRACT**</Abstract>**INFO**

**INFO →**<Address>**ADDRESS**</Address>**INFO**

**INFO →**<Author>**AUTHOR**</Author>**INFO**

**INFO →**<Date>**DATE**</Date>**INFO**

**INFO →**<CopyRight>**COPYRIGHT**</CopyRight>**INFO**

**INFO →**<Title>**TITLE**</Title>**INFO**

**INFO →** <MediaObject>**MEDIAOBJECT**</MediaObject>

**INFO →**<Abstract>**ABSTRACT**</Abstract>

**INFO →**<Address>**ADDRESS**</Address>

**INFO →**<Author>**AUTHOR**</Author>

**INFO →**<Date>**DATE**</Date>

**INFO →**<CopyRight>**COPYRIGHT**</CopyRight>

**INFO →**<Title>**TITLE**</Title>

**ABSTRACT→**<Title>**TITLE**</Title>**ABSTRACT2**

**ABSTRACT→ABSTRACT2**

**ABSTRACT2→**<para>**PARA**</para>**ABSTRACT2**

**ABSTRACT2→**<Simpara>**SIMPARA**</Simpara>**ABSTRACT2**

**ABSTRACT2→**<para>**PARA**</para>

**ABSTRACT2→**<Simpara>**SIMPARA**</Simpara>

**AUTHOR→**<FirstName>**FIRSTNAME**</FirstName>**AUTHOR**

**AUTHOR→**<surname>**SURNAME**</surname>**AUTHOR**

**AUTHOR→**<FirstName>**FIRSTNAME**</FirstName>

**AUTHOR→**<surname>**SURNAME**</surname>

**ADDRESS→**text **ADDRESS**

**ADDRESS→**<Street>**STREET**</Street>**ADDRESS**

**ADDRESS→**<City>**CITY**</City>**ADDRESS**

**ADDRESS→**<State>**STATE**</State>**ADDRESS**

**ADDRESS→**<Phone>**PHONE**</Phone>**ADDRESS**

**ADDRESS→**<Email>**EMAIL**</Email>**ADDRESS**

**ADDRESS→**text

**ADDRESS→**<Street>**STREET**</Street>

**ADDRESS→**<City>**CITY**</City>

**ADDRESS→**<State>**STATE**</State>

**ADDRESS→**<Phone>**PHONE**</Phone>

**ADDRESS→**<Email>**EMAIL**</Email>

**ADDRESS→**empty

**COPYRIGHT→**<year>**YEAR**</year>**COPYRIGHT**

**COPYRIGHT→**<year>**YEAR**</year>

**COPYRIGHT→**<year>**YEAR**</year>**COPYRIGHT2**

**COPYRIGHT2→**<holder>**HOLDER**</holder>**COPYRIGHT2**

**COPYRIGHT2→**<holder>**HOLDER**</holder>

**TITLE→** text **TITLE**

**TITLE→** <Emphasis>**EMPHASIS**</Emphasis>**TITLE**

**TITLE→** <Link xlink:href=”text”>**LINK**</Link>**TITLE**

**TITLE→** <Email>**EMAIL**</Email>**TITLE**

**TITLE→** text

**TITLE→** <Emphasis>**EMPHASIS**</Emphasis>

**TITLE→** <Link xlink:href=”text”>**LINK**</Link>

**TITLE→** <Email>**EMAIL**</Email>

**SIMPARA→** text **SIMPARA**

**SIMPARA→**<Emphasis>**EMPHASIS**</Emphasis>**SIMPARA**

**SIMPARA→**<Link xlink:href=”text”>**LINK**</Link>**SIMPARA**

**SIMPARA→**Email>**EMAIL**</Email>**SIMPARA**

**SIMPARA→**<Author>**AUTHOR**</Author>**SIMPARA**

**SIMPARA→**<Comment>**COMMENT**</Comment>**SIMPARA**

**SIMPARA→** text

**SIMPARA→**<Emphasis>**EMPHASIS**</Emphasis>

**SIMPARA→**<Link xlink:href=”text”>**LINK**</Link>

**SIMPARA→**Email>**EMAIL**</Email>

**SIMPARA→**<Author>**AUTHOR**</Author>

**SIMPARA→**<Comment>**COMMENT**</Comment>

**EMPHASIS→** text **EMPHASIS**

**EMPHASIS→**<Emphasis>**EMPHASIS**</Emphasis>**EMPHASIS**

**EMPHASIS→**<Link xlink:href=”text”>**LINK**</Link>**EMPHASIS**

**EMPHASIS→**Email>**EMAIL**</Email>**EMPHASIS**

**EMPHASIS→**<Author>**AUTHOR**</Author>**EMPHASIS**

**EMPHASIS→**<Comment>**COMMENT**</Comment>**EMPHASIS**

**EMPHASIS→** text

**EMPHASIS→**<Emphasis>**EMPHASIS**</Emphasis>

**EMPHASIS→**<Link xlink:href=”text”>**LINK**</Link>

**EMPHASIS→**Email>**EMAIL**</Email>

**EMPHASIS→**<Author>**AUTHOR**</Author>

**EMPHASIS→**<Comment>**COMMENT**</Comment>

**COMMENT→** text **COMMENT**

**COMMENT→**<Emphasis>**EMPHASIS**</Emphasis>**COMMENT**

**COMMENT→**<Link xlink:href=”text”>**LINK**</Link>**COMMENT**

**COMMENT→**Email>**EMAIL**</Email>**COMMENT**

**COMMENT→**<Author>**AUTHOR**</Author>**COMMENT**

**COMMENT→**<Comment>**COMMENT**</Comment>**COMMENT**

**COMMENT→**text

**COMMENT→**<Emphasis>**EMPHASIS**</Emphasis>

**COMMENT→**<Link xlink:href=”text”>**LINK**</Link>

**COMMENT→**Email>**EMAIL**</Email>

**COMMENT→**<Author>**AUTHOR**</Author>

**COMMENT→**<Comment>**COMMENT**</Comment>

**LINK→** text **LINK**

**LINK→**<Emphasis>**EMPHASIS**</Emphasis>**LINK**

**LINK→**<Link xlink:href=”text”>**LINK**</Link>**LINK**

**LINK→**Email>**EMAIL**</Email>**LINK**

**LINK→**<Author>**AUTHOR**</Author>**LINK**

**LINK→**<Comment>**COMMENT**</Comment>**LINK**

**LINK→** text

**LINK→**<Emphasis>**EMPHASIS**</Emphasis>

**LINK→**<Link xlink:href=”text”>**LINK**</Link>

**LINK→**Email>**EMAIL**</Email>

**LINK→**<Author>**AUTHOR**</Author>

**LINK→**<Comment>**COMMENT**</Comment>

**PARA→** text **PARA**

**PARA→** <Emphasis>**EMPHASIS**</Emphasis>**PARA**

**PARA→** <Link xlink:href=”text”>**LINK**</Link>**PARA**

**PARA→** Email>**EMAIL**</Email>**PARA**

**PARA→** <Link xlink:href=”text”>**LINK**</Link>**PARA**

**PARA→** <Comment>**COMMENT**</Comment>**PARA**

**PARA→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList>**PARA**

**PARA→** <Important>**IMPORTANT**</Important>**PARA**

**PARA→** <Address>**ADDRESS**</Address>**PARA**

**PARA→**<MediaObject>**MEDIAOBJECT**</MediaObject>**PARA**

**PARA→**<InformalTable>**INFORMALTABLE**</InformalTable>**PARA**

**PARA→** text

**PARA→** <Emphasis>**EMPHASIS**</Emphasis>

**PARA→** <Link xlink:href=”text”>**LINK**</Link>

**PARA→** Email>**EMAIL**</Email>

**PARA→** <Link xlink:href=”text”>**LINK**</Link>

**PARA→** <Comment>**COMMENT**</Comment>

**PARA→** <ItemizedList>**ITEMIZEDLIST**</ItemizedList>

**PARA→** <Important>**IMPORTANT**</Important>

**PARA→** <Address>**ADDRESS**</Address>

**PARA→**<MediaObject>**MEDIAOBJECT**</MediaObject>

**PARA→**<InformalTable>**INFORMALTABLE**</InformalTable>

**IMPORTANT→**<Title>**TITLE**</Title>**IMPORTANT2**

**IMPORTANT→IMPORTANT2**

**IMPORTANT2→**<ItemizedList>**ITEMIZEDLIST**</ItemizedList>**IMPORTANT2**

**IMPORTANT2→**<Important>**IMPORTANT**</Important>**IMPORTANT2**

**IMPORTANT2→**<para>**PARA**</para>**IMPORTANT2**

**IMPORTANT2→**<Simpara>**SIMPARA**</Simpara>**IMPORTANT2**

**IMPORTANT2→**<Address>**ADDRESS**</Address>**IMPORTANT2**

**IMPORTANT2→**<MediaObject>**MEDIAOBJECT**</MediaObject>**IMPORTANT2**

**IMPORTANT2→**<InformalTable>**INFORMALTABLE**</InformalTable>**IMPORTANT2**

**IMPORTANT2→**<Comment>**COMMENT**</Comment>**IMPORTANT2**

**IMPORTANT2→**<Abstract>**ABSTRACT**</Abstract>**IMPORTANT2**

**IMPORTANT2→**<ItemizedList>**ITEMIZEDLIST**</ItemizedList>

**IMPORTANT2→**<Important>**IMPORTANT**</Important>

**IMPORTANT2→**<para>**PARA**</para>

**IMPORTANT2→**<Simpara>**SIMPARA**</Simpara>

**IMPORTANT2→**<Address>**ADDRESS**</Address>

**IMPORTANT2→**<MediaObject>**MEDIAOBJECT**</MediaObject>

**IMPORTANT2→**<InformalTable>**INFORMALTABLE**</InformalTable>

**IMPORTANT2→**<Comment>**COMMENT**</Comment>

**IMPORTANT2→**<Abstract>**ABSTRACT**</Abstract>

**FirstName→** text **FirstName**

**FirstName→** <Link> xlink:href=”text”>**LINK**</Link>**FirstName**

**FirstName→** <Emphasis>**EMPHASIS**</Emphasis>**FirstName**

**FirstName→** <Comment>**COMMENT**</Comment>**FirstName**

**FirstName→** text

**FirstName→**<Link xlink:href=”text”>**LINK**</Link>

**FirstName→** <Emphasis>**EMPHASIS**</Emphasis>

**FirstName→** <Comment>**COMMENT**</Comment>

**Surname→** text **Surname**

**Surname→** <Link> xlink:href=”text”>**LINK**</Link>**Surname**

**Surname→** <Emphasis>**EMPHASIS**</Emphasis>**Surname**

**Surname→** <Comment>**COMMENT**</Comment>**Surname**

**Surname→** text

**Surname→**<Link xlink:href=”text”>**LINK**</Link>

**Surname→** <Emphasis>**EMPHASIS**</Emphasis>

**Surname→** <Comment>**COMMENT**</Comment>

**Street→** text **Street**

**Street→** <Link> xlink:href=”text”>**LINK**</Link>**Street**

**Street→** <Emphasis>**EMPHASIS**</Emphasis>**Street**

**Street→** <Comment>**COMMENT**</Comment>**Street**

**Street→** text

**Street→**<Link xlink:href=”text”>**LINK**</Link>

**Street→** <Emphasis>**EMPHASIS**</Emphasis>

**Street→** <Comment>**COMMENT**</Comment>

**City→** text **City**

**City→** <Link> xlink:href=”text”>**LINK**</Link>**City**

**City→** <Emphasis>**EMPHASIS**</Emphasis>**City**

**City→** <Comment>**COMMENT**</Comment>**City**

**City→** text

**City→**<Link xlink:href=”text”>**LINK**</Link>

**City→** <Emphasis>**EMPHASIS**</Emphasis>

**City→** <Comment>**COMMENT**</Comment>

**State→** text **State**

**State→** <Link> xlink:href=”text”>**LINK**</Link>**State**

**State→** <Emphasis>**EMPHASIS**</Emphasis>**State**

**State→** <Comment>**COMMENT**</Comment>**State**

**State→** text

**State→**<Link xlink:href=”text”>**LINK**</Link>

**State→** <Emphasis>**EMPHASIS**</Emphasis>

**State→** <Comment>**COMMENT**</Comment>

**Phone→** text **Phone**

**Phone→** <Link> xlink:href=”text”>**LINK**</Link>**Phone**

**Phone→** <Emphasis>**EMPHASIS**</Emphasis>**Phone**

**Phone→** <Comment>**COMMENT**</Comment>**Phone**

**Phone→** text

**Phone→**<Link xlink:href=”text”>**LINK**</Link>

**Phone→** <Emphasis>**EMPHASIS**</Emphasis>

**Phone→** <Comment>**COMMENT**</Comment>

**Email→** text **Email**

**Email→** <Link> xlink:href=”text”>**LINK**</Link>**Email**

**Email→** <Emphasis>**EMPHASIS**</Emphasis>**Email**

**Email→** <Comment>**COMMENT**</Comment>**Email**

**Email→** text

**Email→**<Link xlink:href=”text”>**LINK**</Link>

**Email→** <Emphasis>**EMPHASIS**</Emphasis>

**Email→** <Comment>**COMMENT**</Comment>

**Date→** text **Date**

**Date→** <Link> xlink:href=”text”>**LINK**</Link>**Date**

**Date→** <Emphasis>**EMPHASIS**</Emphasis>**Date**

**Date→** <Comment>**COMMENT**</Comment>**Date**

**Date→** text

**Date→**<Link xlink:href=”text”>**LINK**</Link>

**Date→** <Emphasis>**EMPHASIS**</Emphasis>

**Date→** <Comment>**COMMENT**</Comment>

**Year→** text **Year**

**Year→** <Link> xlink:href=”text”>**LINK**</Link>**Year**

**Year→** <Emphasis>**EMPHASIS**</Emphasis>**Year**

**Year→** <Comment>**COMMENT**</Comment>**Year**

**Year→** text

**Year→**<Link xlink:href=”text”>**LINK**</Link>

**Year→** <Emphasis>**EMPHASIS**</Emphasis>

**Year→** <Comment>**COMMENT**</Comment>

**Holder→** text **Holder**

**Holder→** <Link> xlink:href=”text”>**LINK**</Link>**Holder**

**Holder→** <Emphasis>**EMPHASIS**</Emphasis>**Holder**

**Holder→** <Comment>**COMMENT**</Comment>**Holder**

**Holder→** text

**Holder→**<Link xlink:href=”text”>**LINK**</Link>

**Holder→** <Emphasis>**EMPHASIS**</Emphasis>

**Holder→** <Comment>**COMMENT**</Comment>

**MEDIAOBJECT →**<Info>**INFO**</Info>**MEDIADATA**

**MEDIAOBJECT →MEDIADATA**

**MEDIADATA→** <videoobject>**VIDEOOBJECT**</videoobject> **MEDIADATA2**

**MEDIADATA→** <imageobject> **IMAGEOBJECT** </imageobject> **MEDIADATA2**

**MEDIADATA→** <videoobject>**VIDEOOBJECT**</videoobject>

**MEDIADATA→** <imageobject> **IMAGEOBJECT** </imageobject>

**MEDIADATA2→** <videoobject>**VIDEOOBJECT**</videoobject> **MEDIADATA2**

**MEDIADATA2→** <imageobject> **IMAGEOBJECT** </imageobject> **MEDIADATA2**

**MEDIADATA2→** <videoobject>**VIDEOOBJECT**</videoobject>

**MEDIADATA2→** <imageobject> **IMAGEOBJECT** </imageobject>

**VIDEOOBJECT→**<Info>**INFO**</Info> **VIDEOOBJECT2**

**VIDEOOBJECT→VIDEOOBJECT2**

**IMAGEOBJECT→**<Info>**INFO**</Info> **IMAGEOBJECT2**

**IMAGEOBJECT→ IMAGEOBJECT2**

**VIDEOOBJECT2→VIDEODATA**

**IMAGEOBJECT2→IMAGENDATA**

**IMAGENDATA→**<ImageData fileref=”text” />

**VIDEODATA→**<VideoData fileref=”text” />

**INFORMALTABLE → TABLEOBJECT**

**INFORMALTABLE → TABLEGROUP**

**TABLEOBJECT →** <mediaobject> **MEDIAOBJECT** </mediaobject> **TABLEOBJECT**

**TABLEOBJECT →** <imageobject> **MEDIAOBJECT** </imageobject>

**TABLEGROUP →** <tgroup>**TGROUP**</tgroup> **TABLEGROUP**

**TABLEGROUP →**<tgroup>**TGROUP**</tgroup>

**TGROUP→**<thead>**THEAD**</thead> **TGROUP1**

**TGROUP→TGROUP1**

**TGROUP1→**<tfoot>**TFOOT**</tfoot> **TGROUP2**

**TGROUP1→TGROUP2**

**TGROUP2→**<tbody>**TBODY**</tbody>

**THEAD →** <row>**ROW**</row>**THEAD**

**THEAD →** <row>**ROW**</row>

**TFOOT →** <row>**ROW**</row> **TFOOT**

**TFOOT →** <row> **ROW**</row>

**TBODY →** <row> **ROW**</row> **TBODY**

**TBODY →** <row> **ROW**</row>

**ROW →** <entry>**ENTRY**</entry> **ROW**

**ROW →** <entrytbl>**ENTRYTBL**</entrytbl> **ROW**

**ROW →** <entry>**ENTRY**</entry>

**ROW →** <entrytbl>**ENTRYTBL**</entrytbl>

**ENTRYTBL→** <thead>**THEAD**</thead> **ENTRYTBL1**

**ENTRYTBL→ ENTRYTBL1**

**ENTRYTBL1→**<tbody>**TBODY**</tbody>

**ENTRY→** text **ENTRY**

**ENTRY→**<ItemizedList>**ITEMIZEDLIST**</ItemizedList>**ENTRY**

**ENTRY→**<Important>**IMPORTANT**</Important>**ENTRY**

**ENTRY→**<para>**PARA**</para>**ENTRY**

**ENTRY→**<Simpara>**SIMPARA**</Simpara>**ENTRY**

**ENTRY→**<MediaObject>**MEDIAOBJECT**</MediaObject>**ENTRY**

**ENTRY→**<Comment>**COMMENT**</Comment>**ENTRY**

**ENTRY→**<Abstract>**ABSTRACT**</Abstract>**ENTRY**

**ENTRY→**text

**ENTRY→**<ItemizedList>**ITEMIZEDLIST**</ItemizedList>

**ENTRY→**<Important>**IMPORTANT**</Important>

**ENTRY→**<para>**PARA**</para>

**ENTRY→**<Simpara>**SIMPARA**</Simpara>

**ENTRY→**<MediaObject>**MEDIAOBJECT**</MediaObject>

**ENTRY→**<Comment>**COMMENT**</Comment>

**ENTRY→**<Abstract>**ABSTRACT**</Abstract>

**ITEMIZEDLIST→** <Listitem>**LISTITEM**</Listitem> **ITEMZEDLIST**

**ITEMIZEDLIST→** <Listitem>**LISTITEM**</Listitem>

**LISTITEM→**<ItemizedList>**ITEMIZEDLIST**</ItemizedList>**LISTITEM**

**LISTITEM→**<Important>**IMPORTANT**</Important>**LISTITEM**

**LISTITEM→**<para>**PARA**</para>**LISTITEM**

**LISTITEM→**<Simpara>**SIMPARA**</Simpara>**LISTITEM**

**LISTITEM→**<Address>**ADDRESS**</Address>**LISTITEM**

**LISTITEM→**<MediaObject>**MEDIAOBJECT**</MediaObject>**LISTITEM**

**LISTITEM→**<InformalTable>**INFORMALTABLE**</InformalTable>**LISTITEM**

**LISTITEM→**<Comment>**COMMENT**</Comment>**LISTITEM**

**LISTITEM→**<Abstract>**ABSTRACT**</Abstract>**LISTITEM**

**LISTITEM→**<ItemizedList>**ITEMIZEDLIST**</ItemizedList>

**LISTITEM→**<Important>**IMPORTANT**</Important>

**LISTITEM→**<para>**PARA**</para>

**LISTITEM→**<Simpara>**SIMPARA**</Simpara>

**LISTITEM→**<Address>**ADDRESS**</Address>

**LISTITEM→**<MediaObject>**MEDIAOBJECT**</MediaObject>

**LISTITEM→**<InformalTable>**INFORMALTABLE**</InformalTable>

**LISTITEM→**<Comment>**COMMENT**</Comment>

**LISTITEM→**<Abstract>**ABSTRACT**</Abstract>

# Análisis Léxico

En el primer paso de nuestro TPI, programamos el Lexer, que se encarga de leer el código fuente ingresado por texto o leído de un archivo externo, y descomponer en unidades; secuencia de caracteres que concuerdan con las palabras reservadas del lenguaje Doctype (si están escritos correctamente). Identifica y clasifica estos tokens, también ignora espacios en blanco, devuelve una secuencia de caracteres con todas las palabras reservadas encontradas en la cadena de texto.

En caso de que una cadena de caracteres no coincida con las palabras reservadas definidas, devuelve un error.

## HERRAMIENTAS

**Bibliotecas**

**PLY**

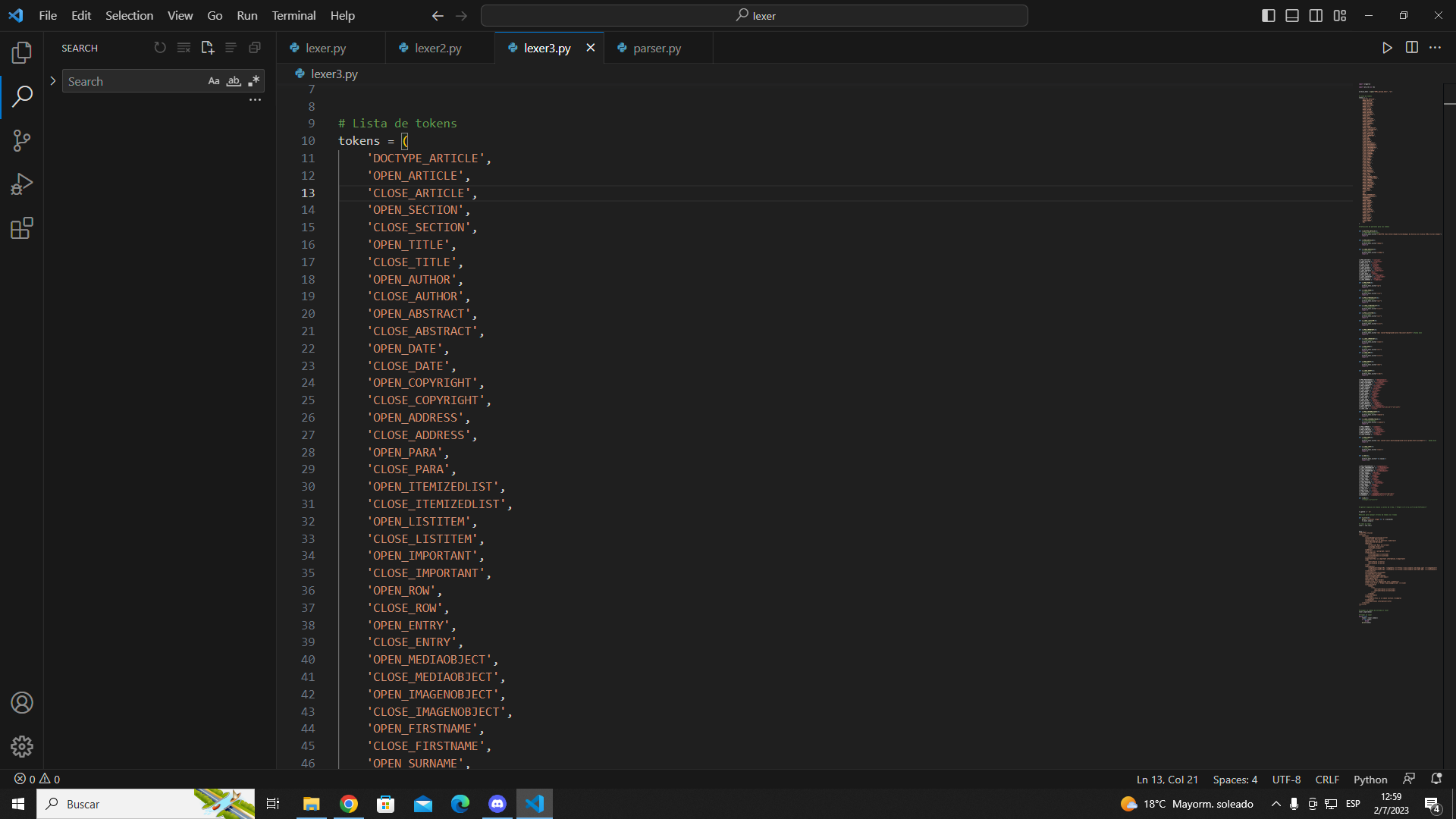
Python Lex-Yacc es una biblioteca en Python que facilita la implementación de analizadores léxicos (lexers) y analizadores sintácticos (parsers). Proporciona una forma conveniente de definir reglas léxicas y sintácticas en Python y generar automáticamente el código fuente para los analizadores.

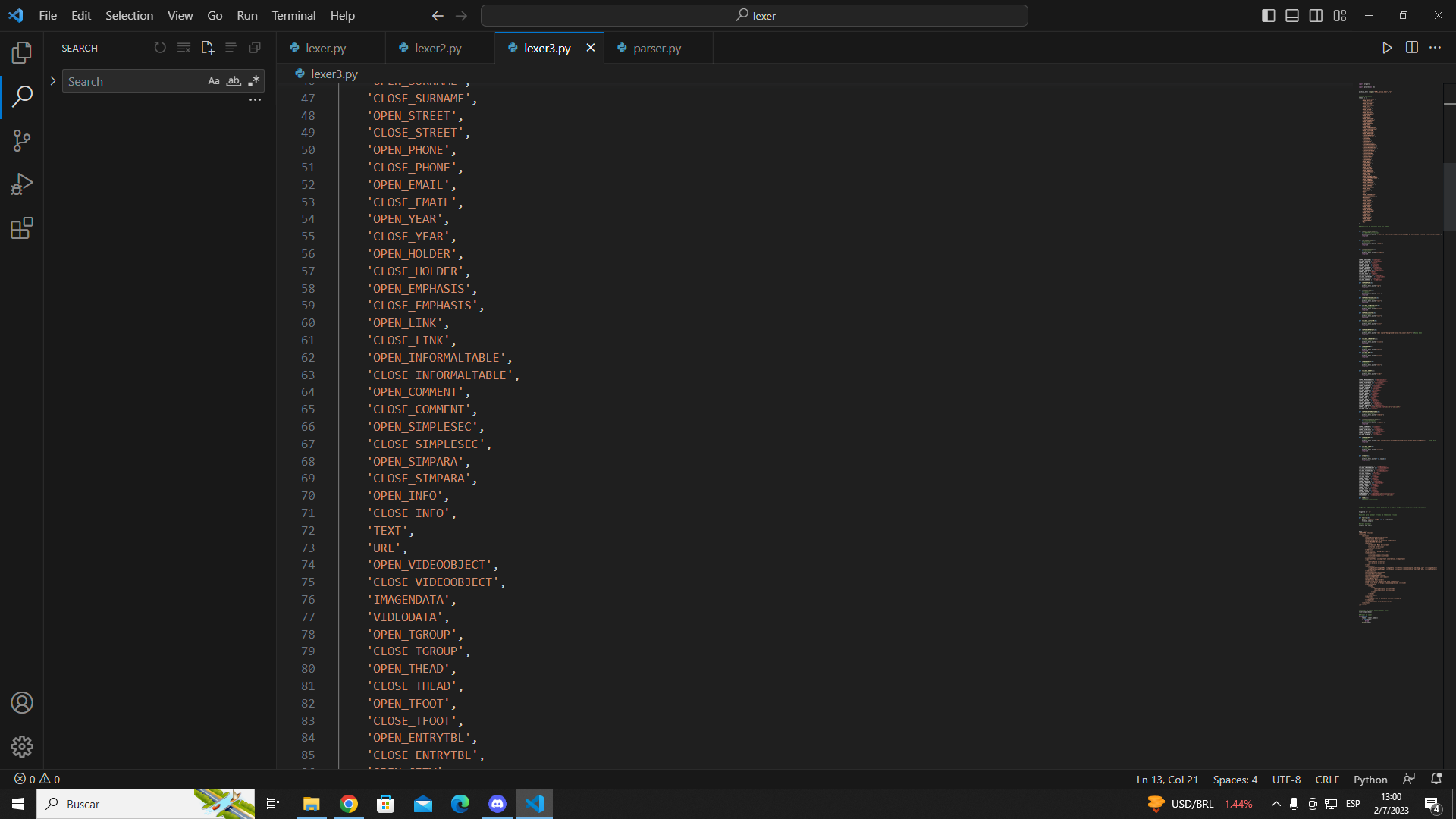
El analizador léxico de ply se encarga de escanear el código fuente, carácter por carácter, buscar coincidencias con los tokens definidos y generar los tokens correspondientes, como secuencia de caracteres que es el dato de entrada con la que trabaja el parser.

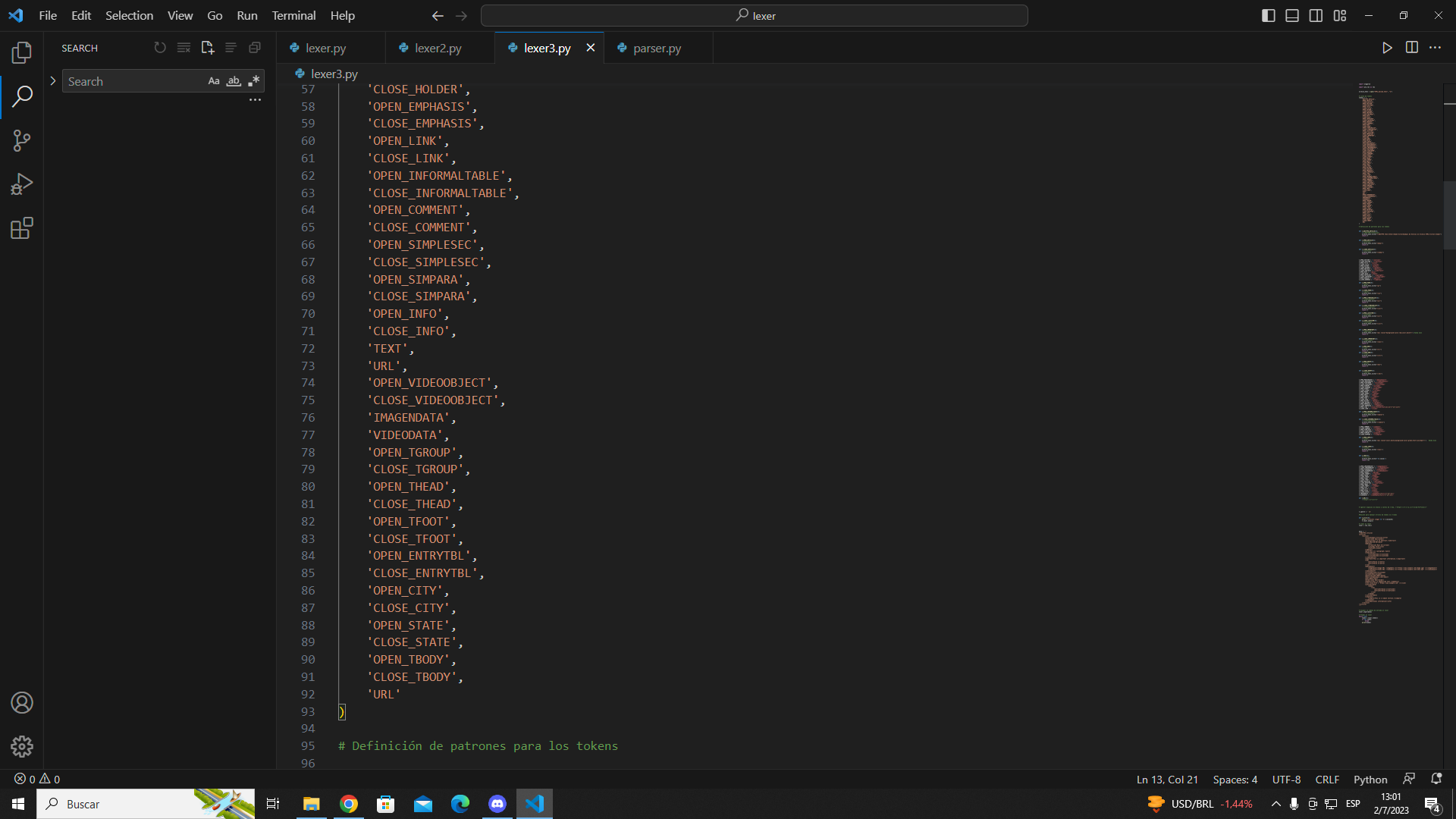
## CÓDIGO

1. **Definición de TOKENS**

Se definen los *tokens* que el analizador léxico reconocerá, cada *TERMINAL* definido en la gramática es un token. El lexer genera como salida una secuencia de caracteres con los tokens que encontró. Esta secuencia de salida utiliza el parser como datos de entrada para realizar el análisis sintáctico.





  
  
token: Un token es una unidad léxica que representa un componente individual en un lenguaje o gramática.

Se compone de caracteres o patrones que siguen una regla de formación específica.

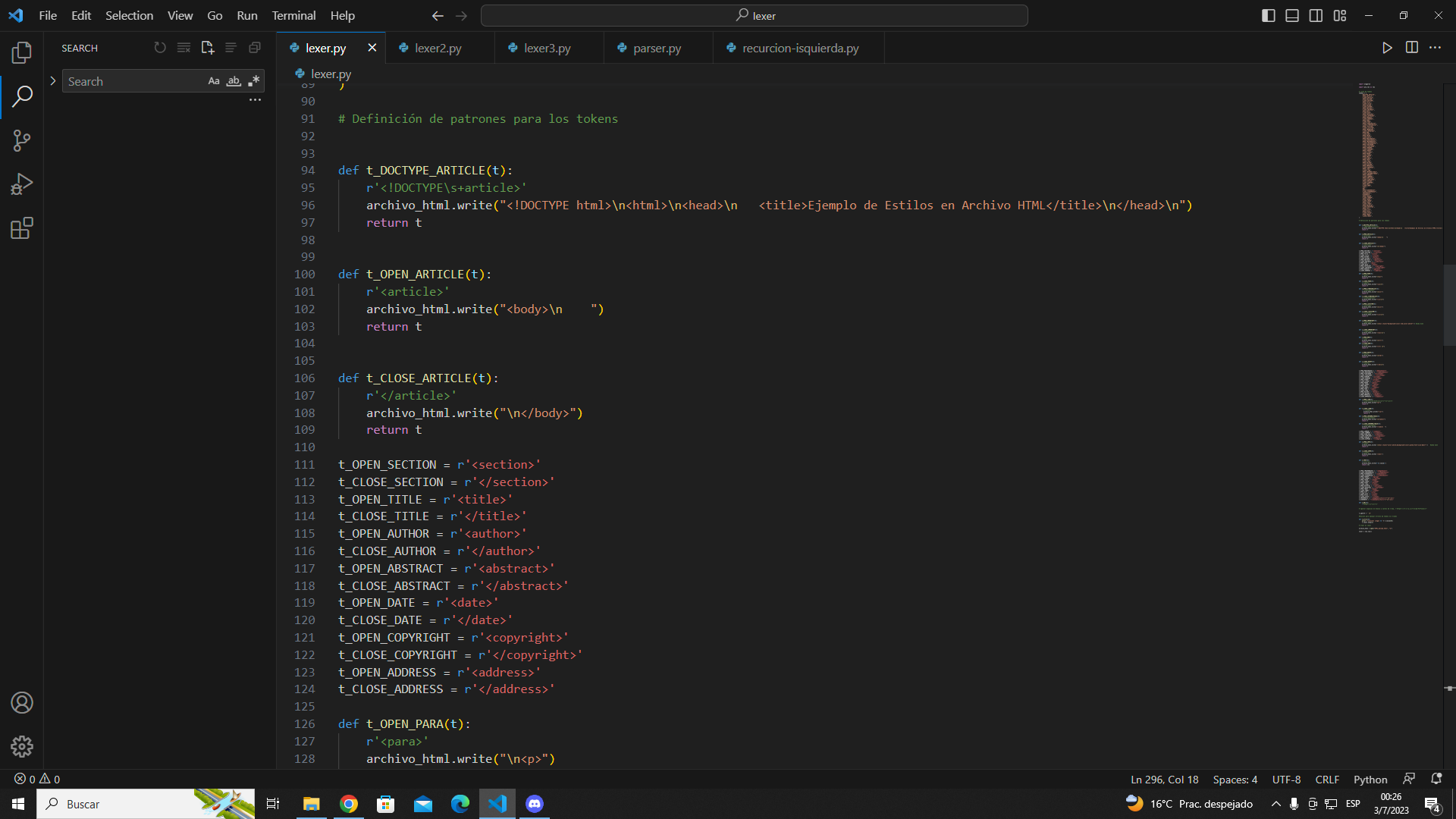
Ejemplo: En un lenguaje de programación, el token "if" representa una palabra clave para realizar una condición.

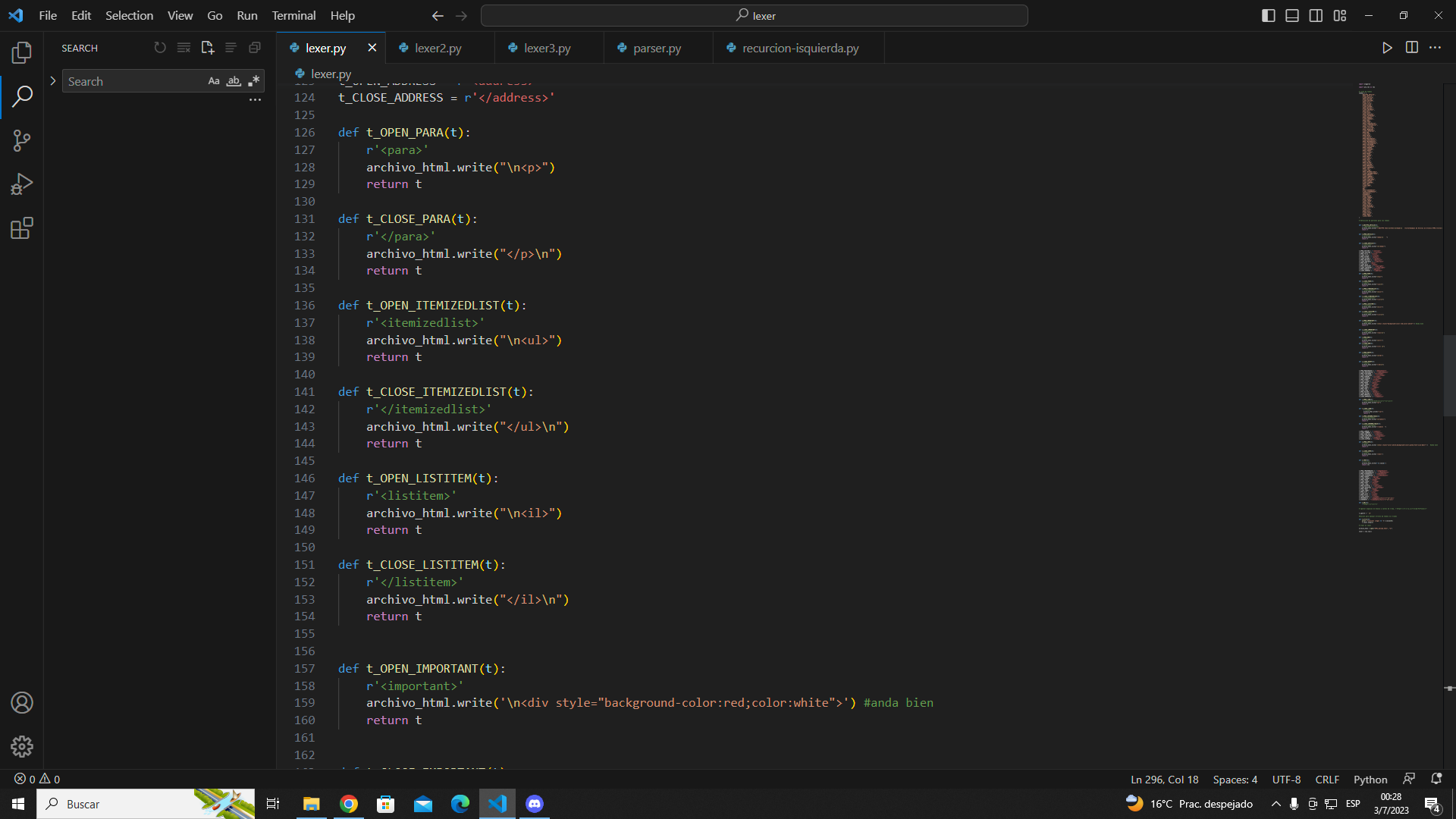
1. Definición de Patrones de los Tokens y Conversión a HTML

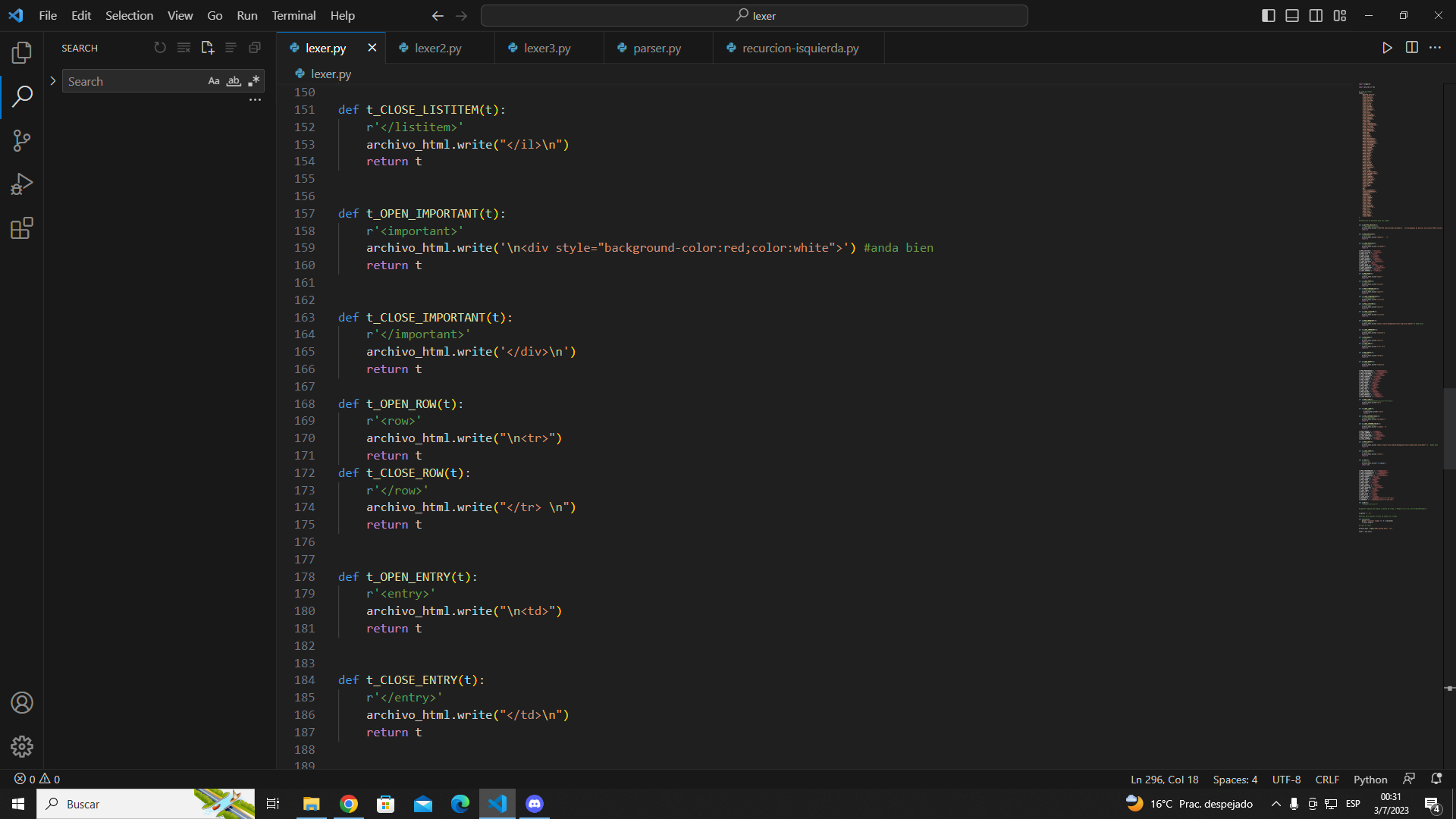
Cada NO TERMINAL definido en la gramática, está asociado a un TERMINAL. Al leer el código fuente que analiza, lee los terminales (etiquetas o expresiones regulares) y busca el no terminal asociado para generar el token.

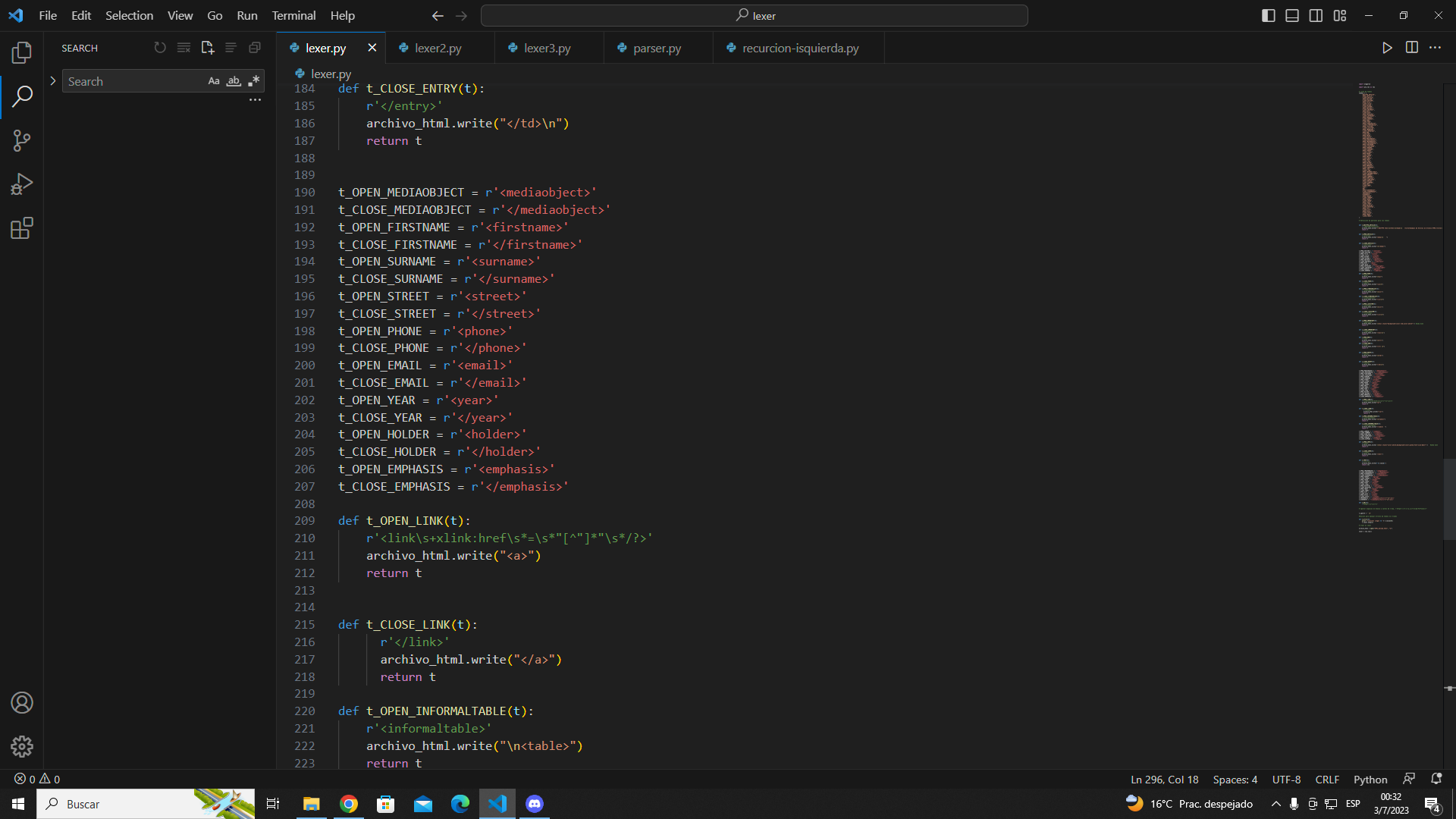
El lexer analiza caracter a caracter el código fuente y busca las coincidencias con las expresiones regulares o las etiquetas para producir los tokens correspondientes.

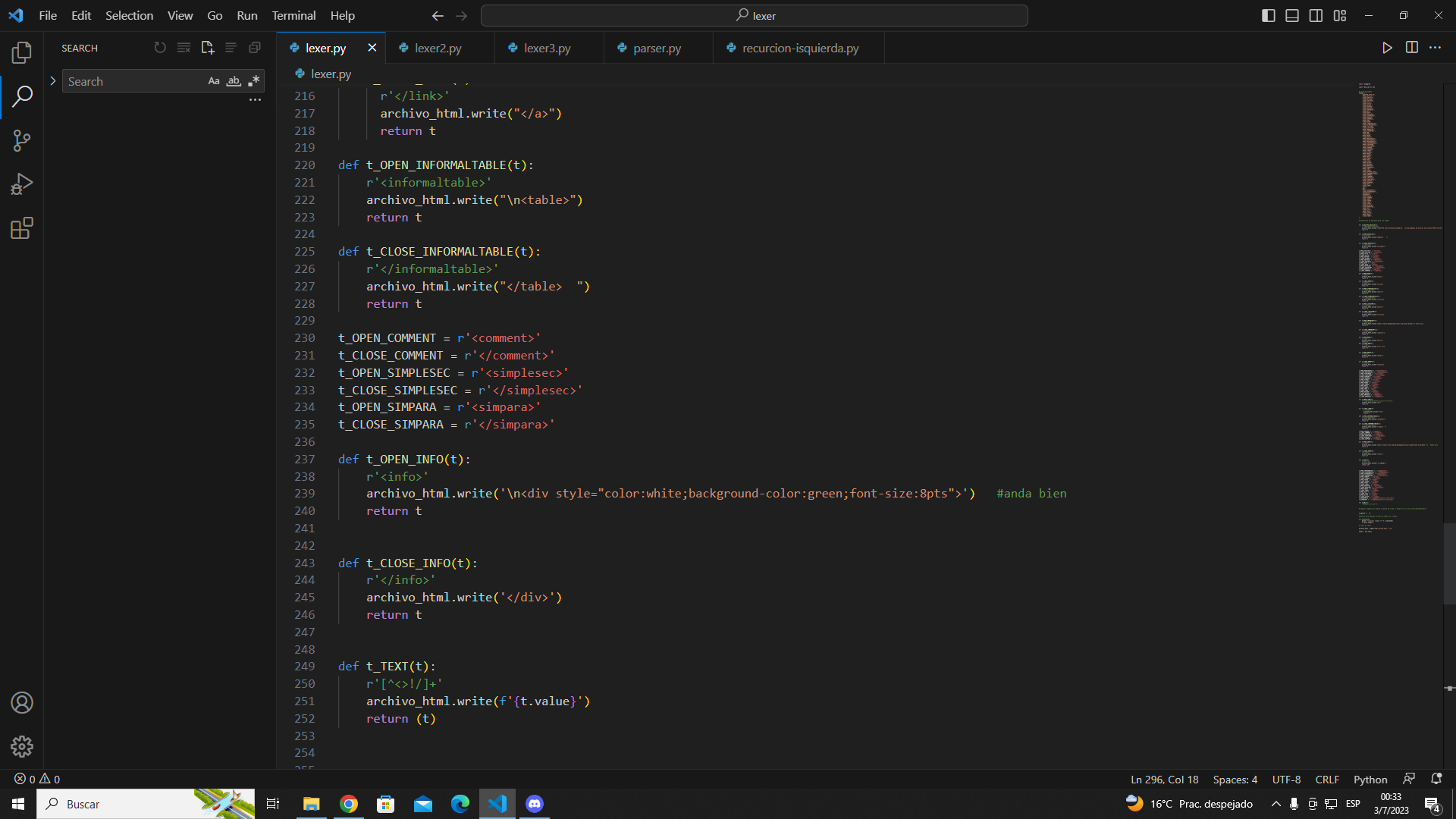
También, al detectar ciertas etiquetas, escribe su equivalente en un archivo de salida html con su formato.

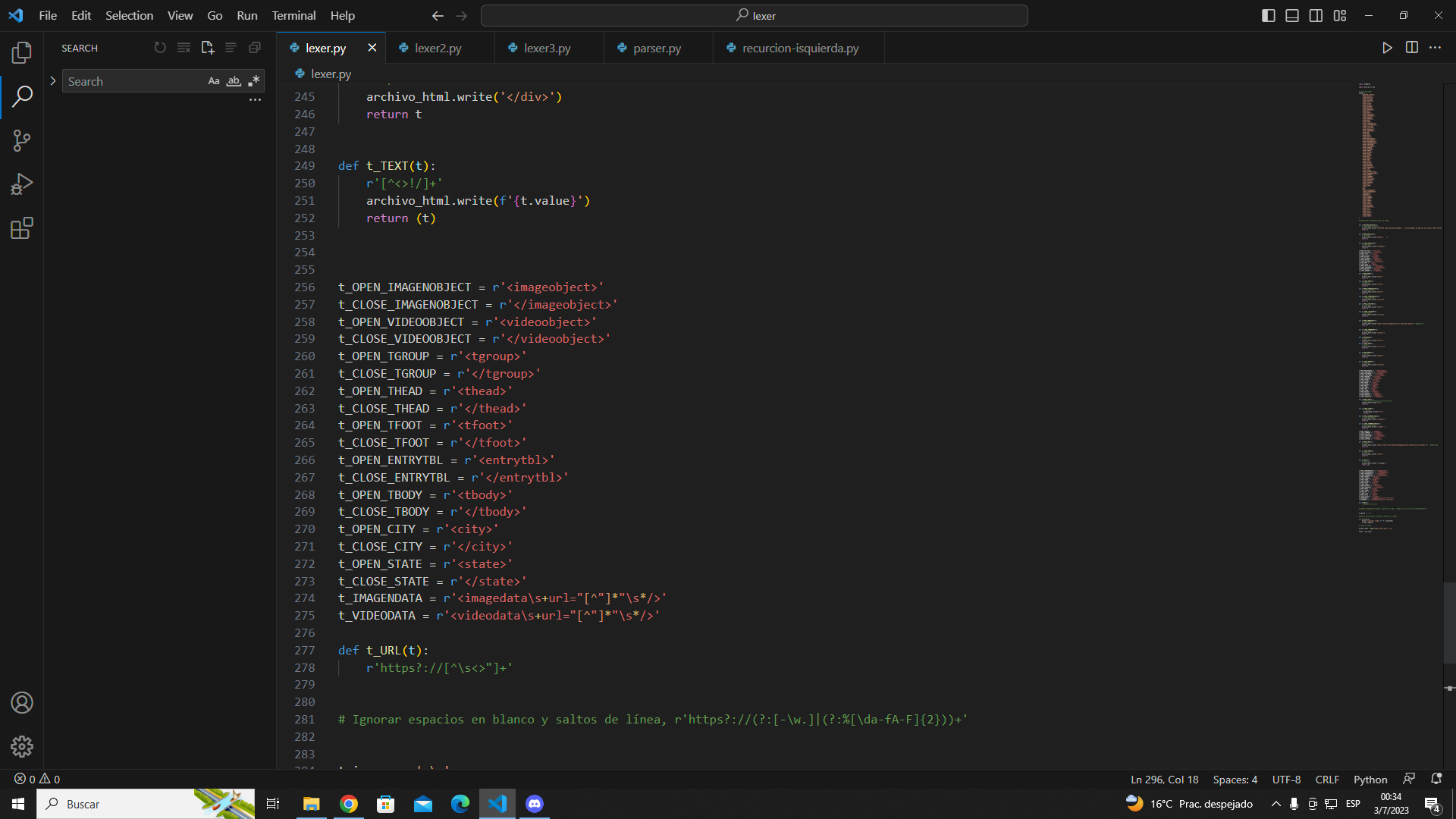












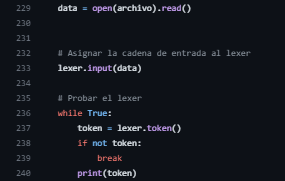
1. Generación del lexer



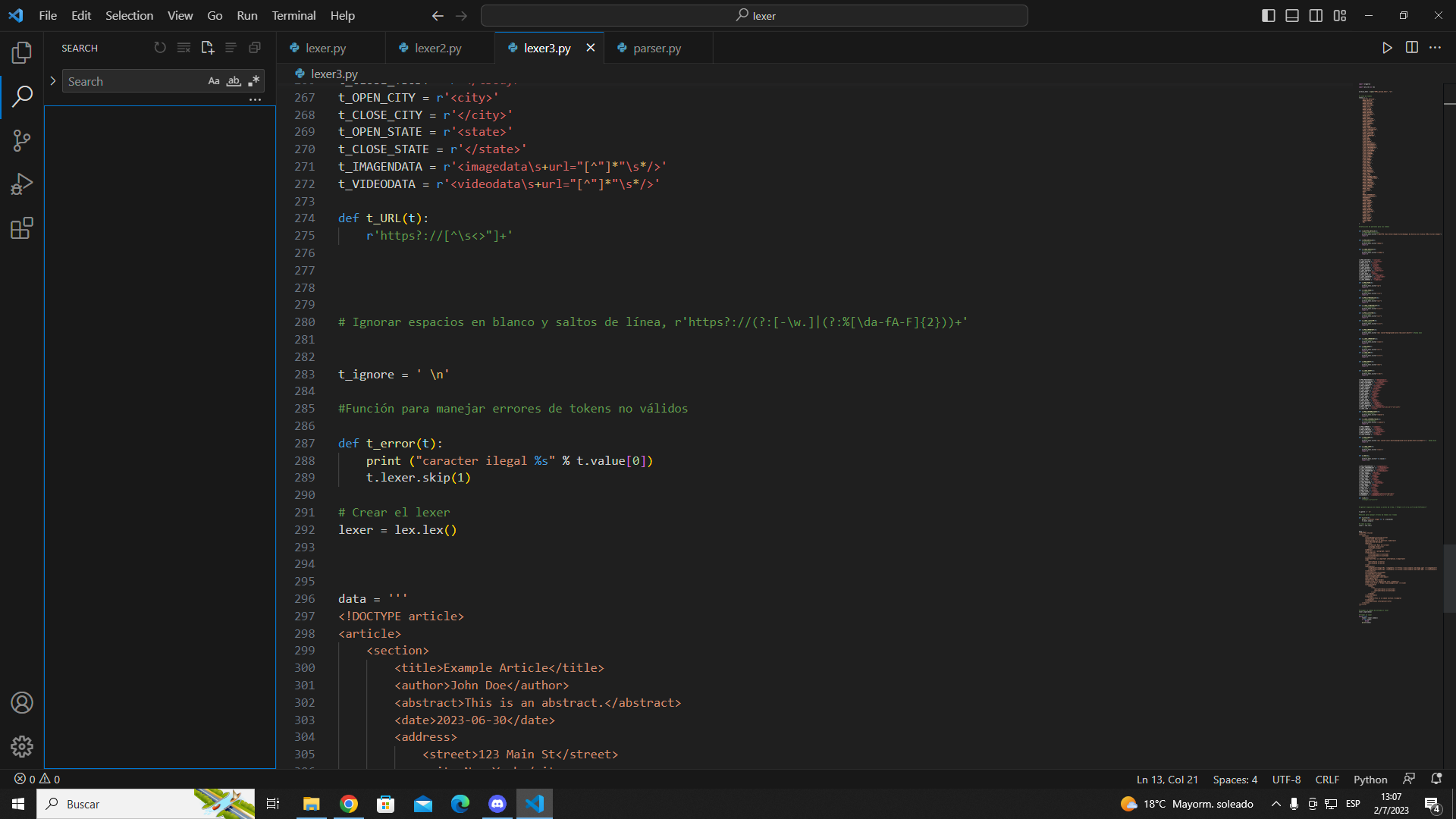
1. Analisis del codigo fuente y Generación de TOKENS

Se lee un archivo de entrada que es abierto desde la ejecución del parser, y se lo pasa como parámetro de entrada al lexer generado en el paso anterior. Cuando se encuentra con una coincidencia el lexer genera el token y se lo pasa al parser, guarda en la variable token, nombre, el texto correspondiente y la posición en el código fuente.

Sale del bucle cuando no encuentra más tokens, e imprime la variable token, con la información de cada token encontrado descrita anteriormente.



1. Manejo de errores léxicos  
   T\_error es una función auxiliar que se explica su funcionamiento en el apartado de Funciones Auxiliares. Proporciona una forma de manejar y reportar errores léxico.



# Análisis Sintáctico

En el segundo paso de nuestro TPI, programamos el parser, el cual llama al lexer mientras se ejecuta. El lexer fue programado en la segunda entrega y se encarga de reconocer la secuencia de tokens para que el parser pueda validar la estructura gramatical del archivo o cadena de texto ingresada. El parser se encarga de estructurar los tokens según las reglas gramaticales definidas.

## Herramientas

**Tkinter**

Tkinter es una biblioteca estándar de Python que se utiliza para crear interfaces gráficas de usuario (GUI, por sus siglas en inglés). Proporciona una interfaz de programación fácil de usar y multiplataforma para crear ventanas, botones, cajas de texto, menús, y otros elementos de una interfaz gráfica.

El funcionamiento básico de Tkinter se basa en la creación de una ventana principal y la adición de diferentes widgets (elementos de la interfaz) a esta ventana.

**Lexer**

**Tokens**

Se importan los tokens y el lexer generados a partir del archivo “lexer.py”. Corresponde al trabajo entregado en la segunda entrega donde se programó el lexer y se definió los tokens a reconocer. Se importa como lista que contiene los nombres de los tokens definidos.

**Yacc**

El módulo yacc de PLY es una herramienta para construir parser. Se obtiene acceso a las funciones y clases necesarias para generar un parser utilizando PLY. Se utiliza para generar el parser a partir de las reglas de gramática especificadas.

**Re**

En Python, la biblioteca re proporciona funciones y métodos para trabajar con expresiones regulares. Una expresión regular es una secuencia de caracteres que define un patrón de búsqueda en un texto**.**

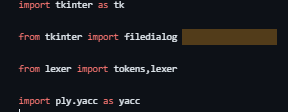
Utilizar expresiones regulares en Python implica importar el módulo re, definir un patrón de búsqueda utilizando una cadena de caracteres con caracteres literales y metacaracteres, y luego aplicar ese patrón a un texto utilizando las funciones y métodos proporcionados por el módulo re**.**

## Código

1. **Importación de Herramientas**

Se importan las herramientas descritas anteriormente.

1. Se importa el módulo tkinter.
2. Importa la función filedialog tkinter que se describe su funcionamiento en el apartado **Funciones Auxiliares.**
3. Del archivo lexer.py se importa como lista los tokens y el lexer generado
4. Se importa el módulo YACC de ply.



1. **Generación de Archivo de Salida (HTML)**

Se abre un archivo en modo de escritura que será la salida que crea el parser, haciendo la transformación del archivo doctype a HTML con el respectivo formato que debe tener cada token.



1. **Definición de Reglas Gramaticales y transformación a HTML**

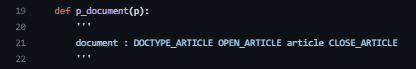
En PLY, se define una función con el nombre del token o no terminal que recibe como parámetro (p) que es una lista que contiene todos los tokens que va encontrando, y la acción a realizar (derivación) cuando encuentra ese no terminal.

Tokens:

**Start**



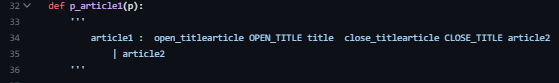
**Document**

****

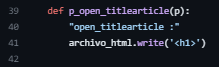
**Article**

****

**Article1**

****

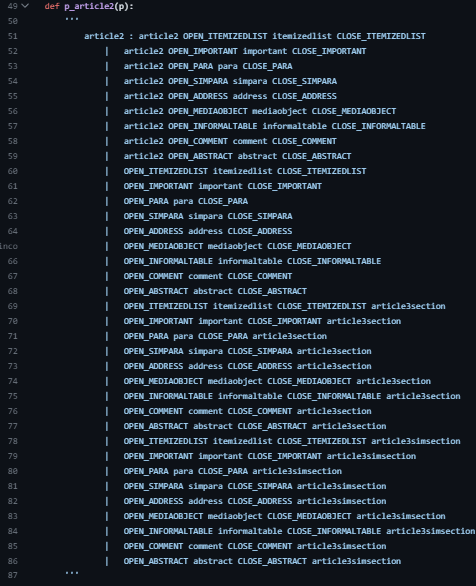
**Conversión a HTML**

****

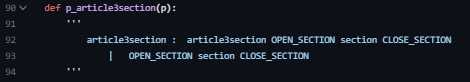
**Article2**



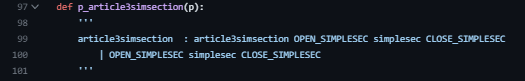
**Article3section**



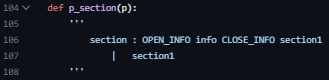
**Article3section**

****

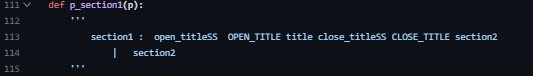
**Article3simsection**

****

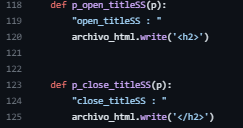
**Section**

.

**Section1**

****

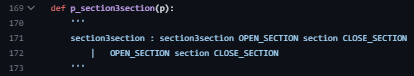
**Conversión a HTML**

****

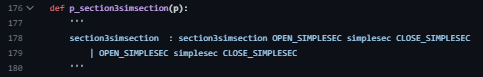
**Section2**

.

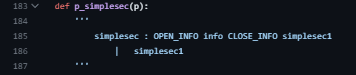
**Section3section**

****

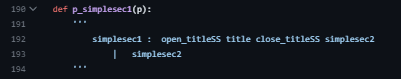
**Section3simsection**

****

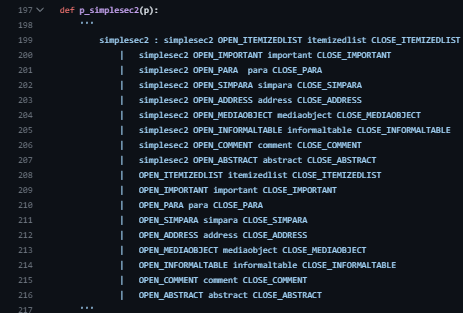
**Simplesec**

****

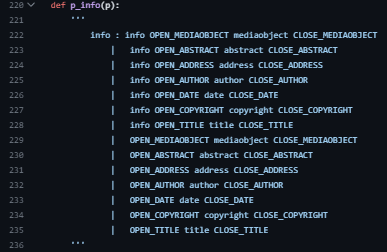
**Simplesec1**

. 

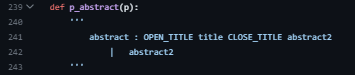
**Simplesec2**

****

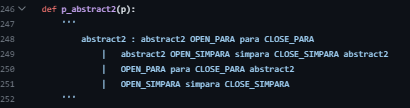
**Info**



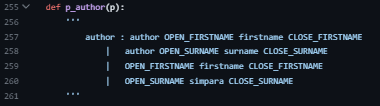
**Abstract**

****

**Abstract2**

.

**Author**

.

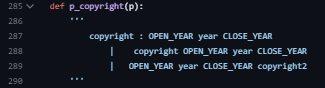
**Address**



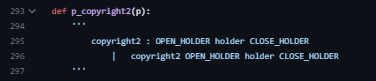
**Empty**

****

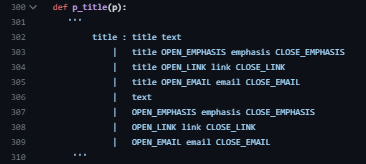
**Copyright**

****

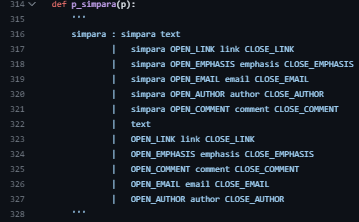
**Copyright2**

****

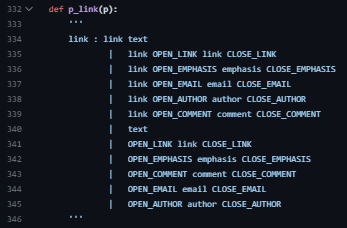
**Title**

****

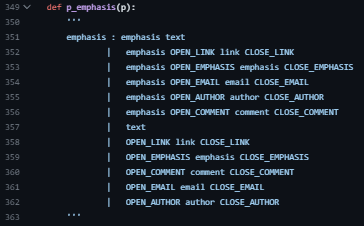
**Simpara**

****

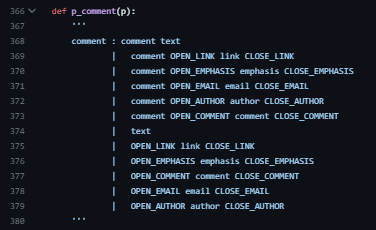
**Link**

****

**Emphasis**

****

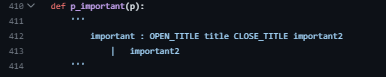
**Comment**

****

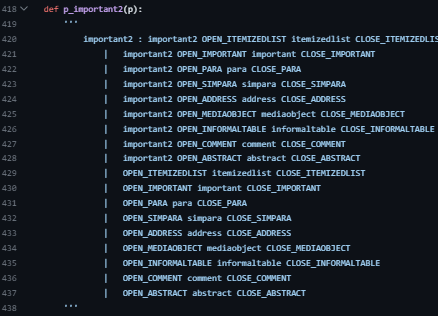
**Para**

****

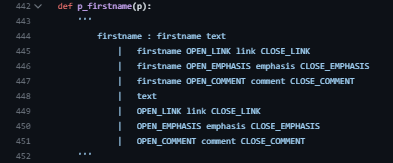
**Important**

****

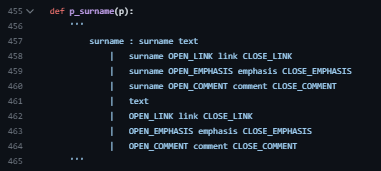
**Important2**

****

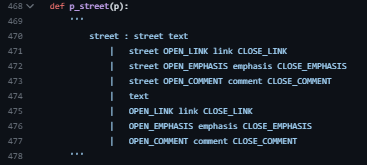
**Firstname**

****

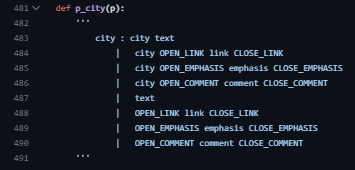
**Surname**

****

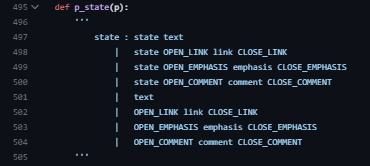
**Street**

****

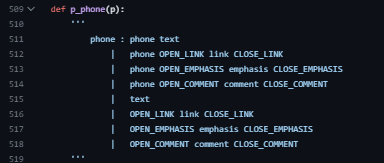
**City**

****

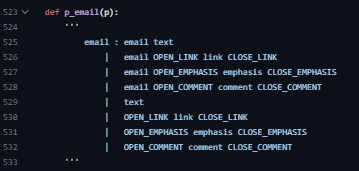
**State**

****

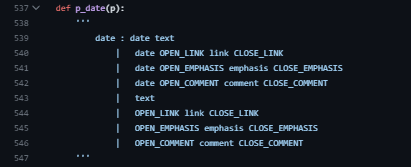
**Phone**

****

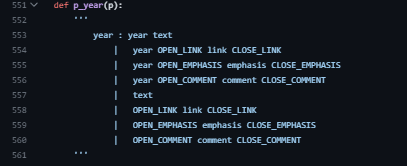
**Email**

****

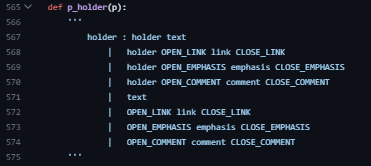
**Date**

****

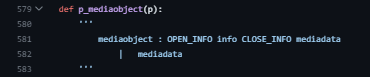
**Year**

****

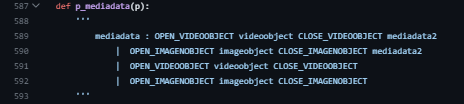
**Holder**

****

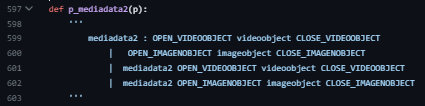
**Mediaobject**

****

**Mediadata**

****

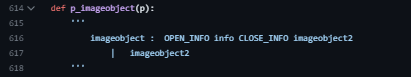
**Mediadata2**

****

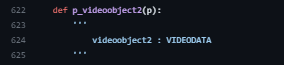
**Videobject**

****

**Imageobject**

****

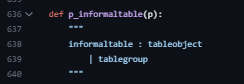
**Videoobject2**

****

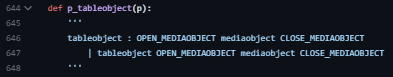
**Imageobject2**

****

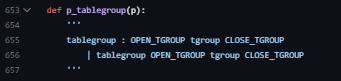
**Informaltable**

****

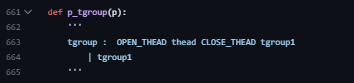
**Tableobject**

****

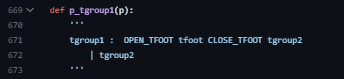
**Tablegroup**

****

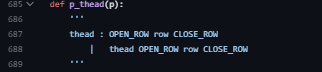
**Tgroup**

****

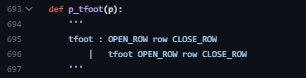
**Tgroup1**

****

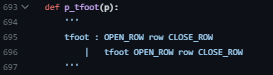
**Tgroup2**

****

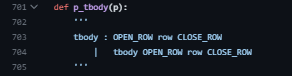
**Thead**

****

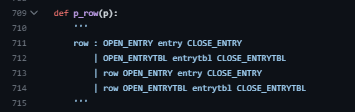
**Tfoot**

****

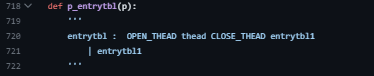
**Tbody**

****

**Trow**

****

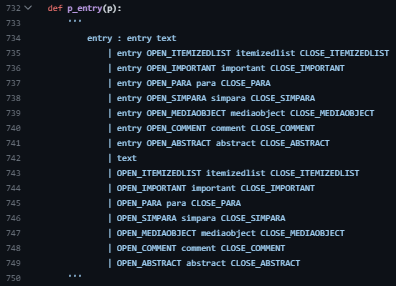
**entrytbl**

****

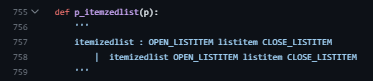
**entrytbl1**

****

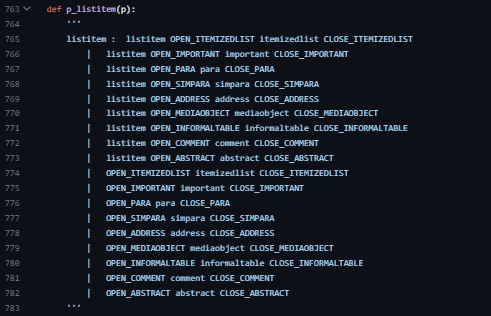
**entry**

****

**itemizedList**

****

**ListItem**

****

1. **Ejecución del parser de un Archivo Seleccionado**.

*Seleccionar archivo txt*

Utilizando la función **filedialog**, se permite abrir un cuadro de diálogo que permite al usuario seleccionar un archivo txt. Se guarda en una etiqueta, el nombre del archivo seleccionado y .pack es una función de tk que permite colocar el nombre del archivo en la interfaz,





*Función ejecutar\_parser*

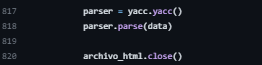
Se define la función ejecutar\_parser, donde se abre el archivo seleccionado en modo lectura y se lee su contenido, que luego será llamada desde manera interactiva desde un boton. Se pasa el contenido del archivo al lexer





*Generación de Parser*

Se crea el parser y se analiza la estructura gramatical del archivo mediante la función YACC de PLY. Luego se cierra el archivo HTML que se generó al ejecutar el parser.



*Ejecución de Parser*

Fuera de la función ejecutar\_parser, se crea un botón de ejecución que llama a la función ejecutar\_parser cuando se lo llama, descripta anteriormente



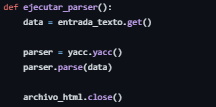
1. **Ejecución del parser desde modo interactivo - Analiza lo escrito**.

*Funcion mostrar\_ventana\_entrada*

Este código muestra una ventana emergente donde el usuario puede ingresar texto, y al hacer clic en el botón "Ejecutar", el texto se pasa al lexer y al parser. Se muestran mensajes en la ventana emergente según si se encontraron errores o si el texto se procesó correctamente.

**

*Ejecutar Parser*

Se define la función ejecutar\_parser dentro de la función mostrar\_ventana\_entrada:   
Dentro de ejecutar\_parser(), se obtiene el texto ingresado por el usuario mediante entrada\_texto.get().  
  


*Interfaz entrada de texto*

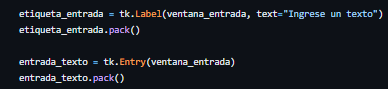
La interfaz para permitir al usuario ingresar un texto y ejecutarlo, está codificado de la siguiente forma:

*Generación ventana emergente*   
Se crea una nueva ventana emergente llamada ventana\_entrada utilizando la función Toplevel. Se establece el título de la ventana emergente como "Ventana de Entrada" utilizando el método title() y la geometría de la ventana emergente con el método de geometry de tk.



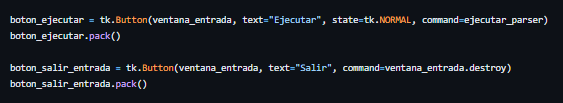
*Ingreso de Texto*

Se imprime en la interfaz el mensaje de ingrese un texto, utilizando los metodos label y pack de tk. Se crea un campo de entrada de texto dentro de la ventana emergente utilizando la función Entry. Este campo permite al usuario ingresar texto. Se vuelve a utilizar la función .pack para que la entrada del usuario se muestra en la interfaz.





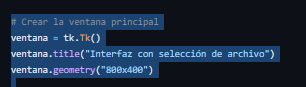
*Botón de ejecución*  
  
Se crean dos botones de interacción, un boton de ejecución quese llama la función ejecutar\_parser. También se crea un botón de salida (Button) dentro de la ventana emergente utilizando la función .Button. El botón muestra el texto "Salir". Al interactuar con el botón se cerrará la ventana emergente pasando como parámetro la función ventana\_entrada.destroy



1. Ventana Principal de la Interfaz

*Línea de Codigo ():Generar ventana principal*

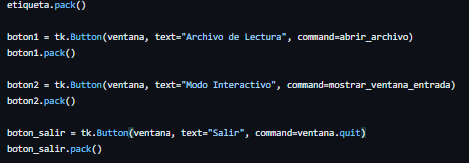
Se crea la ventana principal de la función llamando a la función .tk, se genera un título con la función .title y se dimensiona la ventana con la función .geometry.



*Generación de botones interactivos*

Se imprime un mensaje en la ventana, y se genera 3 botones interactivos que uno llama a la función abrir\_archivo, otro a la función mostrar\_ventana\_Entrad y otro botón que permite la salida de salida.





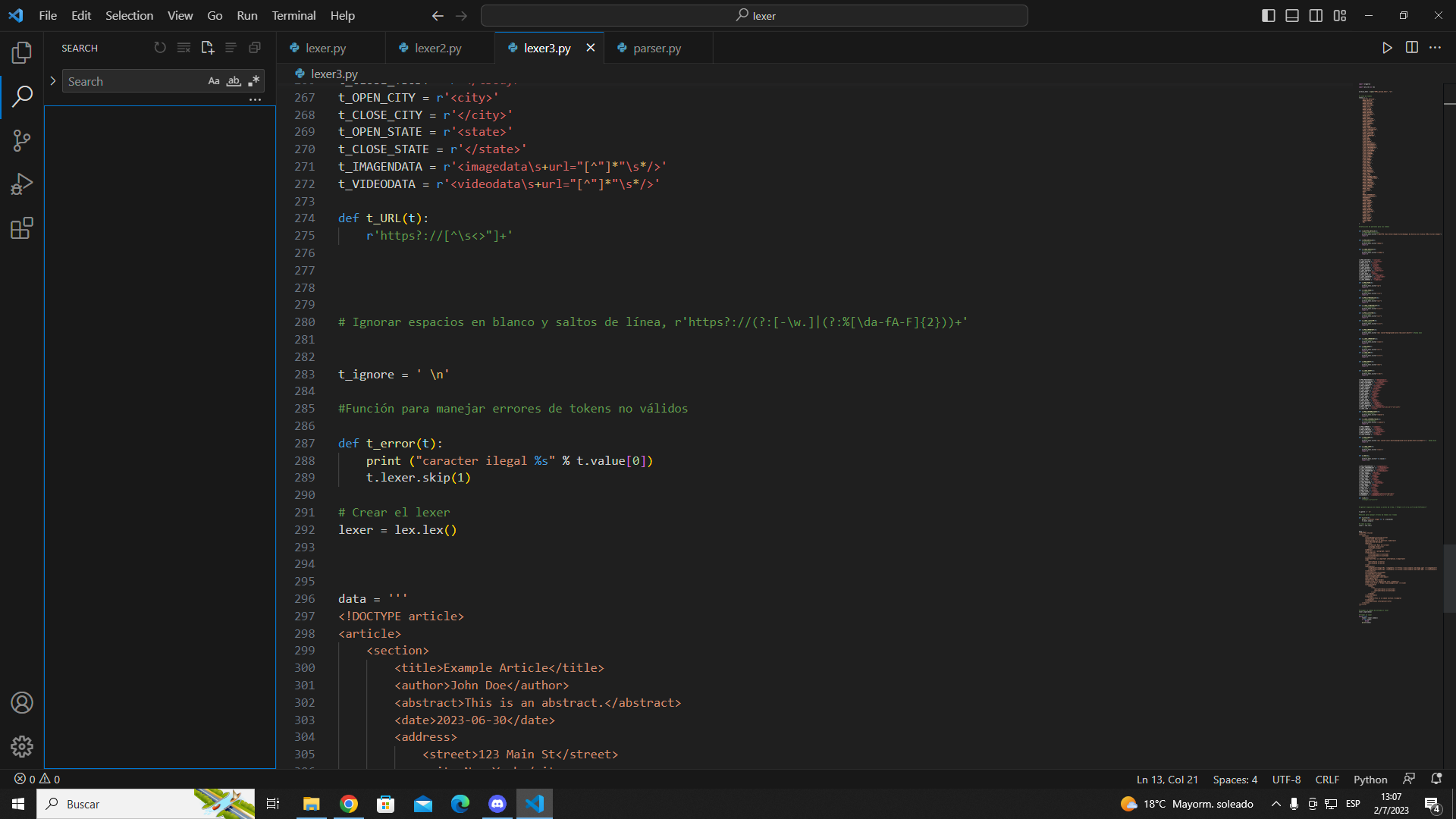
*Bucle de la aplicación*

Se inicializa el bucle principal de la interfaz gráfica, llamando a la función .mainloop se inicia un ciclo que espera interacción del usuario.



# Funciones Auxiliares

**Función t\_error**



La función t\_error(t) se utiliza en PLY (Python Lex-Yacc) para manejar errores léxicos durante el análisis léxico.

Cuando el analizador léxico encuentra un carácter o una secuencia de caracteres que no coincide con ninguna de las reglas definidas, se llama a esta función t\_error. Recibe como argumento t, que representa el token que causó el error.

En el código que proporcionaste, la función t\_error(t) simplemente imprime un mensaje que indica el carácter ilegal encontrado utilizando el valor del token t.value. Luego, llama a t.lexer.skip(1) para omitir ese carácter y avanzar al siguiente carácter en el análisis léxico.

**Función Filedialog**

****

La función filedialog es parte del módulo tkinter en Python y proporciona una interfaz gráfica para que los usuarios seleccionen archivos o directorios en sus sistemas de archivos locales. Esta función permite a los desarrolladores de Python crear ventanas emergentes de diálogo de archivos y directorios en sus aplicaciones.

La función filedialog proporciona varios métodos para abrir y guardar archivos, seleccionar directorios y manejar diferentes tipos de archivos.Los métodos utilizados fueron los siguientes:

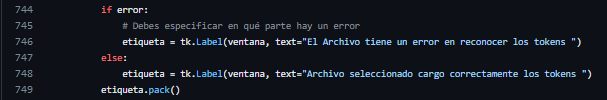
**Filedialog.askopenfilename**

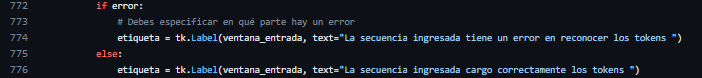
Abre una ventana de diálogo que permite al usuario seleccionar un archivo existente para abrir. Recibe como parámetros:

****

**tk.Label**

La función tk.Label es parte del módulo tkinter en Python y se utiliza para crear un widget de etiqueta en una interfaz gráfica de usuario (GUI). Un widget de etiqueta se utiliza para mostrar texto o imágenes estáticas en una ventana de la aplicación.

****

****

****

****

**variable.pack()**

La función pack() es uno de los métodos de administración de geometría en tkinter que se utiliza para colocar widgets en una ventana o en otro widget contenedor. Su objetivo principal es organizar y distribuir los widgets de manera automática dentro del espacio disponible. variable.Pack coloca la etiqueta creada con el .label en la interfaz.

Ejemplos en el código:

****

****

****

****

****

****

****

****

**tk.Button**

Es una clase en el módulo tkinter de Python que se utiliza para crear botones en una interfaz gráfica de usuario (GUI). Los botones son componentes interactivos que los usuarios pueden hacer clic para realizar acciones en una aplicación, los argumentos que toma en nuestro código son los de:

Ejemplos en el código:

****

****

****

****

****

****

**tk.TopLevel**

La clase tk.TopLevel en el módulo tkinter de Python se utiliza para crear ventanas emergentes adicionales, también conocidas como ventanas secundarias o ventanas de nivel superior, en una interfaz gráfica de usuario (GUI).

A diferencia de la ventana principal creada con tk.Tk(), que representa la ventana principal de la aplicación, tk.TopLevel se utiliza para crear ventanas adicionales que pueden contener otros widgets y proporcionar funcionalidades específicas.

Utilización en el código:

****

**variable.Title**

variable.title("texto") se utiliza para establecer el título de la ventana representada por variable a "texto"

Utilización en el código:

****

****

**variable.Geometry**

Es una forma válida de establecer la geometría (tamaño y posición) de una ventana representada por variable. El ancho y el alto se especifican en píxeles, y posx y posy son las coordenadas de la esquina superior izquierda de la ventana en la pantalla.

Utilización en el código:

****

****

**variable.Entry**

Entry, que es un widget de entrada de texto en tkinter. Se utiliza para permitir al usuario ingresar y editar texto en la interfaz gráfica.

Ejemplos en el código:

****

**tk.Tk**

Se utiliza para crear una ventana principal en tkinter. Tk es una clase en tkinter que representa la ventana principal de la aplicación.

Utilización en el código:

****

**.bind**

Se utiliza para asociar un evento específico con una función de controlador. Cuando ocurre el evento especificado, se ejecuta la función de controlador.  
Utilización en el código:

****

**.mainloop.**

Es un método que se llama en una instancia de Tk para iniciar el bucle principal de la aplicación. El bucle principal permite que la interfaz gráfica responda a eventos y se actualice de manera continua.

Utilización en el código:

****

# Ejemplos

Ejemplo 1

<!DOCTYPE article>

<article>

<info>

<title>El titulo del articulo</title>

<author>

<firstname>Juan</firstname>

<surname>Perez</surname>

</author>

</info>

<mediaobject>

<imageobject><imagedata url="https://www.example.com/image.jpg" /></imageobject>

<videoobject><videodata url="https://www.example.com/video.mp4" /></videoobject>

</mediaobject>

<simplesec>

<info>

<address>

<phone>123-456-7890</phone>

<email>john@example.com</email>

</address>

<copyright>

<year>2023</year>

<holder>John Doe</holder>

</copyright>

</info>

<simpara>

<link xlink:href = "https://www.example.com/" >haga Click aqui</link>

<author>

<firstname>John</firstname>

<surname>Doe</surname>

</author>

<emphasis>This is emphasized text.</emphasis>

</simpara>

<informaltable>

<tgroup>

<tbody>

<row>

<entry>Value 1</entry>

<entry>Value 2</entry>

</row>

</tbody>

</tgroup>

</informaltable>

</simplesec>

</article>

Ejemplo 2

<!DOCTYPE article>

<article>

<title>Ejemplo TPI</title>

<comment>Cualquier sugerencia de mejora es bienvenida</comment>

<section>

<title>

<link xlink:href = "https://www.example.com/" >Click aqui</link>

</title>

<para>Titulo seccion1</para>

<section>

<comment>Seccion de prueba</comment>

</section>

</section>

</article>

# 

# Conclusiones

En conclusión, hemos logrado llegar a una solución que cumple con los objetivos descritos anteriormente. A través de la implementación de analizadores léxicos y sintácticos utilizando el generador PLY, pudimos realizar el análisis, validación y transformación de un subconjunto de etiquetas DocBook/XML. Además, solucionamos posibles problemas como la complejidad de la gramática, el manejo de errores, la optimización y el rendimiento, así como el mantenimiento y actualizaciones del sistema.

Este trabajo nos ha brindado una valiosa oportunidad para aplicar los conceptos aprendidos en la materia de Sintaxis y Semántica de los Lenguajes. Hemos podido comprender y aplicar los fundamentos teóricos en un contexto práctico, lo que ha fortalecido nuestra comprensión de la materia y nos ha permitido adquirir habilidades prácticas en el análisis de lenguajes de programación.

Además, este proyecto nos ha permitido familiarizarnos con el proceso de construcción de analizadores léxicos y sintácticos, así como desarrollar nuestras habilidades de resolución de problemas y diseño de soluciones eficientes.

En resumen, esta experiencia ha sido enriquecedora tanto desde el punto de vista académico como práctico. Hemos logrado implementar una solución que cumple con los requisitos planteados, superando los posibles desafíos y mejorando nuestra comprensión de la sintaxis y semántica de los lenguajes de programación.

# Solución de Errores

Uno de los grandes problemas que enfrentamos fue el de la comunicación: cada integrante del grupo tenía horarios diferentes y era muy impráctico el manejo de las distintas actualizaciones al código pasandonos primero por whats app y luego por drive.

La solución a este problema fue aprender github, donde se convirtió en nuestra plataforma para almacenar y compartir nuestro código, donde cada uno trabajaba desde su propia ramas luego de dividirnos las tareas o comunicarnos las distintas funcionalidades/errores que debíamos solucionar.

Luego, una vez a la semana nos reunimos para probar y fusionar nuestras versiones y avances realizados. Esta experiencia nos enseñó la importancia de utilizar herramientas adecuadas para facilitar la comunicación y la colaboración en equipo, mejorando significativamente la eficiencia y el flujo de trabajo en nuestro proyecto.

La forma en la que asignamos tareas fue utilizando el método de tablero Kanva en la plataforma Notion, dividimos las tareas en 3: Agregar Funcionalidad, Corregir Errores y HECHO. En la puesta en común semanal, luego de una lluvia de ideas agregaremos las tareas clasificadas y cada iba marcando el progreso de cada una de ellas.

En cuanto a código, en la creación del lexer el principal problema fue la definición de la

url del lexer. La solución fue “a fuerza bruta”, probando cambiando reiteradas veces la url, los ejemplos de url, hasta que conseguimos la expresión regular obtenida que funcionó.

Originalmente, habíamos implementado la gramática utilizando recursión por derecha, pero generaba distintos errores de recursión infinita. Al leer sobre la documentación de ply, descubrimos que este trabajaba sobre recursión por izquierda, por lo que cambiamos toda la gramática a recursión por izquierda y nos dio resultado solucionado nuestro problema de recursión por derecha.

Y el problema que más tardamos en solucionar fue la conversión a HTML, que siempre nos llevaba a un error: primero no reconocía los formatos que debía transformar, luego, escribía en el archivo de salida doble, hasta que lo trabajamos la conversión desde el lexer, debido a que no requería un análisis sintáctico salvo la etiqueta title, que dependía de donde se abría para pasar al archivo de salida como <h1> o <h2>, esa fue la manera que nos permitió que emita en formato y sin errores el archivo de salida.

# 

# Bibliografía o Referencias web

● DocBook 5.1: The Definitive Guide: https://tdg.docbook.org/tdg/5.1/

● DocBook Element Reference: https://tdg.docbook.org/tdg/5.1/ref-elements.html

● Elements Article: https://tdg.docbook.org/tdg/5.1/article.html

● XMLmind XML Editor: http://www.xmlmind.com/xmleditor/

● https://ply.readthedocs.io/en/latest/ply.html#parsing-basics

● <https://www.slideshare.net/dabeaz/writing-parsers-and-compilers-with-ply>

● http://www.dalkescientific.com/writings/NBN/parsing\_with\_ply.html

● [Curso Python](https://www.youtube.com/watch?v=nKPbfIU442g)

● Curso Github

● [Curso de HTML5 desde CERO (Completo)](https://www.youtube.com/watch?v=kN1XP-Bef7w&t=7274s)

● [Git y Github | Curso Práctico de Git y Github Desde Cero](https://www.youtube.com/watch?v=HiXLkL42tMU)