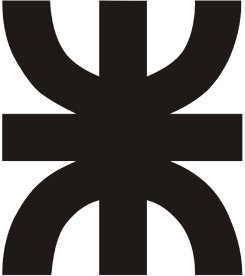


**TRABAJO PRACTICO INTEGRADOR: LEXER + PARSER - TRADUCTOR**

**ASIGNATURA: SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES**

****

**Cátedra:** SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES

**Grupo:** 5

**Carrera:** ISI A

**ALUMNOS:**

* **Gómez, Ricardo**
* **Paredes, Samuel**
* **Ramírez, Rocío V.**
* **Sánchez, Franco Damián**

**-2023-**

**Contenido**

[**INTRODUCCIÓN 3**](#_heading=h.gjdgxs)

[**OBJETIVO 3**](#_heading=h.30j0zll)

[**HERRAMIENTAS Y LENGUAJES 3**](#_heading=h.1fob9te)

[**DOC-BOOK 3**](#_heading=h.3znysh7)

[**XML 4**](#_heading=h.2et92p0)

[**HTML 4**](#_heading=h.tyjcwt)

[**Python 4**](#_heading=h.3dy6vkm)

[**GRAMÁTICA 5**](#_heading=h.1t3h5sf)

[**ANALIZADOR LEXICO (AL) 11**](#_heading=h.4d34og8)

[**Analizador léxico 11**](#_heading=h.2s8eyo1)

[**Tareas que realiza un analizador léxico 12**](#_heading=h.17dp8vu)

[**Palabras reservadas. 12**](#_heading=h.3rdcrjn)

[**Procedimiento del AL 12**](#_heading=h.26in1rg)

[**Errores detectables por el AL 12**](#_heading=h.lnxbz9)

[**Modo de Obtención del Intérprete 13**](#_heading=h.35nkun2)

[**Modo de Ejecución del Intérprete 13**](#_heading=h.44sinio)

[**ANALIZADOR SINTACTICO (AS) 13**](#_heading=h.2jxsxqh)

[**Analizador Sintáctico. 13**](#_heading=h.z337ya)

[**Tareas que realiza un AS 13**](#_heading=h.3j2qqm3)

[**Errores detectables por el AS 13**](#_heading=h.1y810tw)

[**Modo de Obtención del Intérprete 14**](#_heading=h.4i7ojhp)

[**Modo de Ejecución del Intérprete 14**](#_heading=h.2xcytpi)

[**CÓDIGO DE LEXER Y PARSER + TRADUCTOR DE XML A HTML 14**](#_heading=h.1ci93xb)

[**INTERFAZ GRÁFICA 14**](#_heading=h.3whwml4)

[**Modo de obtención de la Interfaz gráfica 15**](#_heading=h.2bn6wsx)

[**EJEMPLOS 15**](#_heading=h.qsh70q)

[**GLOSARIO 15**](#_heading=h.3as4poj)

[**CONCLUSIÓN 16**](#_heading=h.1pxezwc)

[**BIBLIOGRAFÍA Y REFERENCIAS WEB 16**](#_heading=h.49x2ik5)

# INTRODUCCIÓN

Para esta última instancia desarrollamos el Parser en donde a través de este documento dejamos asentado las distintas reglas y características fundamentales que debe cumplir, bajo una implementación desarrollada en el lenguaje de programación Python.

Para nosotros es importante explicar que a través de esta entrega final detallamos en primera instancia información del intérprete: cómo los componentes léxicos(tokens) y sentencias que nuestro Parser y Lexer analizarán del código fuente en XML(Docbook) ingresado, centrado en las directivas otorgadas por la Cátedra respecto a las características del lenguaje solicitado.

Resumiendo en este trabajo desarrollamos un analizador sintáctico para un conjunto de sentencias en Docbook, el cual devolverá las palabras reservadas (tokens) y verificará que la sintaxis sea válida, en caso de que esto no se de así, el programa nos indicará el error. Y en último lugar transformará algunas tokens de docbook a html.

# OBJETIVO

Utilizando el lenguaje de programación, Python, para el generador de lexer y parser se construirán los analizadores léxico (AL) y sintáctico (AS), que permitirán analizar, validar y transformar un subconjunto de etiquetas de la especificación DocBook/XML para la descripción de artículos.

# HERRAMIENTAS Y LENGUAJES

## DOC-BOOK

***DocBook*** es un lenguaje semántico que 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.

**<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.

## XML

***XML*** son las siglas en inglés de “Extensible Markup Language”, un lenguaje de marcas que define un conjunto de reglas de codificación de documentos en un formato que es a la vez legible por el hombre y legible por máquina.

## HTML

* HTML es un [lenguaje de marcado](https://definicion.de/lenguaje/) que se utiliza para el desarrollo de páginas de [Internet](https://definicion.de/internet/). Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.
* Se trata de un formato abierto que surgió a partir de las etiquetas SGML (Standard Generalized Markup Language). Concepto traducido generalmente como «Estándar de Lenguaje de Marcado Generalizado» y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto.
* Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de [etiquetas](https://definicion.de/etiqueta/), también llamadas tags, que permiten interconectar diversos conceptos y formatos.
* Para la escritura de este lenguaje, pues, se crean las etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos). (DEFINICION.DE)

## Python

Python es un lenguaje de alto nivel de programación (facilidad de utilizar para el programador) e interpretado, cuya filosofía hace hincapié en la legibilidad de su código, ya que es lo más parecido al lenguaje humano y se puede aprender de manera sencilla y rápida. Es muy utilizado para desarrollar aplicaciones de todo tipo y se trata de un lenguaje de programación multi-paradigma (resultado de integrar dos o más paradigmas en un mismo sistema). Es un lenguaje interpretado (no se traduce realmente a un formato legible por el ordenador en tiempo de ejecución), dinámico y multiplataforma (todas las herramientas necesarias están disponibles en todas las plataformas principales).

# GRAMÁTICA

A continuación se describen las producciones de la gramática:

**∑** → DOCTYPE\_ARTICULO **XML ARTICULO**

**DOCTYPE\_ACTICULO** → <!DOCTYPE article>

**XML** → <?xml version="**#NUM**" encoding="**#TEXT**"?>

**ARTICULO** → <article> **RAIZ** </article>

**RAIZ** → **INFO TITULO** op opSeccion

**RAIZ** → **INFO** op opSeccion

**RAIZ** → **TITULO** op opSeccion

**RAIZ** → op opSeccion

**RAIZ** → op

**RAIZ** → **INFO TITULO** op

**RAIZ** → **INFO TITULO** op

**RAIZ** → **INFO** op

**RAIZ** → **TITULO** op

op → **ITEMIZED\_LIST** | **IMPORTANTE**| **PARA** | **SIMPLE\_PARA** | **DIRECCION**  | **MEDIA\_OBJECT** | **INFORMAL\_TABLE** | **COMMENT** | **ABSTRACTO**

op → **ITEMIZED\_LIST** op | **IMPORTANTE** op| **PARA** op | **SIMPLE\_PARA** op | **DIRECCION**  op | **MEDIA\_OBJECT** op | **INFORMAL\_TABLE** op | **COMMENT** op | **ABSTRACTO** op

opSeccion → **SIMPLE\_SECCION** | **SECCION** | **SIMPLE\_SECCION** opSeccion | **SECCION** opSeccion

**SECCION** → <section> **RAIZ** </section>

**SIMPLE\_SECCION** → <simplesect> **INFO TITULO** op</simplesect>

**SIMPLE\_SECCION** → <simplesect> **INFO**  op</simplesect>

**SIMPLE\_SECCION** → <simplesect> **TITULO** op</simplesect>

**ITEMIZED\_LIST** → <itemizedlist> **LIST\_ITEM** </itemizedlist>

**ITEMIZED\_LIST** → <itemizedlist> **LIST\_ITEM** </itemizedlist> **ITEMIZED\_LIST**

**LIST\_ITEM** → <listitem> op </listitem> | <listitem> op </listitem> **LIST\_ITEM**

**IMPORTANTE** → <important> **TITULO** op </important>

**IMPORTANTE** → <important> op </important>

**URL** → **http|https|ftp|ftps):\/\/(www\.)?[-a-zA-Z0-9@:%.\_\+~#=]{1,256}\.[a-zA-Z0-9()]{1,6}\b([-a-zA-Z0-9()@:%\_\+.~#?&//=]\***

**TITULO** → <title> opTitulo </title>

**TITULO** → <title> opTitulo </title> **TITULO**

**opTitulo** → **#TEXT** | **EMPHASIS** | **LINK** | **EMAIL**

**opTitulo** → **#TEXT** opTitulo | **EMPHASIS**  opTitulo | **LINK** opTitulo | **EMAIL** opTitulo

**EMPHASIS** → <emphasis>opEmpashis</emphasis>

opEmpashis → **#TEXT** | **LINK** | **EMAIL** | **AUTHOR** | **COMMENT**

opEmpashis →**#TEXT** opEmpashis| **LINK** opEmpashis | **EMAIL** opEmpashis | **AUTHOR** opEmpashis | **COMMENT** opEmpashis

**LINK** → <link xlink:href=”**URL**”/>

**INFO** → <info> opInfo </info>

opInfo → **MEDIA\_OBJECT** | **ABSTRACTO** | **DIRECCION** | **AUTHOR** | **FECHA** | **DERECHOS**| **TITULO**

opInfo → **MEDIA\_OBJECT** opInfo | **ABSTRACTO** opInfo | **DIRECCION**  opInfo | **AUTHOR** opInfo | **FECHA** opInfo | **DERECHOS** opInfo | **TITULO** opInfo

**MEDIA\_OBJECT**  → <mediaobject>**INFO** opMedia</mediaobject>

**MEDIA\_OBJECT** → <mediaobject>**INFO** opMedia</mediaobject> **MEDIA\_OBJECT**

**MEDIA\_OBJECT**  → <mediaobject>opMedia</mediaobject>

**MEDIA\_OBJECT**  → <mediaobject>opMedia</mediaobject> **MEDIA\_OBJECT**

opMedia → **VIDEO\_OBJETO** | **IMAGE\_OBJECT**| **VIDEO\_OBJETO**  opMedia | **IMAGE\_OBJECT** opMedia

**VIDEO\_OBJETO**  → <videoobject> **INFO VIDEO\_DATA**</videoobject>

**VIDEO\_OBJETO** → <videoobject> **VIDEO\_DATA** </videoobject>

**VIDEO\_DATA** → <videodata fileref=”**URL**” />

**IMAGE\_OBJECT** → <imagedata> **INFO IMAGE\_DATA** </imagedata>

**IMAGE\_OBJECT** → <imagedata> **IMAGE\_DATA** </imagedata>

**IMAGE\_DATA** → <imagedata fileref=”**URL**” />

**ABSTRACTO** → <abstract> **TITULO** opPara</abstract>

**ABSTRACTO** → <abstract>opPara</abstract>

opPara → **PARA** | **SIMPLE\_PARA** | **PARA** opPara | **SIMPLE\_PARA** opPara

**SIMPLE\_PARA** → <simpara> OpSECL </simpara>

**PARA** → <para> opPara </para>

opPara → **#TEXT** | **EMPHASIS** | **LINK**| **EMAIL** | **AUTHOR** | **COMMENT** | **ITEMIZED\_LIST** | **IMPORTANTE** | **DIRECCION** | **MEDIA\_OBJECT** | **INFORMAL\_TABLE**

opPara → **#TEXT** opPara | **EMPHASIS** opPara | **LINK** opPara | **EMAIL** opPara | **AUTHOR** opPara | **COMMENT** | **ITEMIZED\_LIST** opPara | **IMPORTANTE** opPara | **DIRECCION**  opPara| **MEDIA\_OBJECT** opPara | **INFORMARL\_TABLE** opPara

**INFORMAL\_TABLE** → <informaltable **ATRIBUTO\_INFORMAL\_TABLE**> **MEDIA\_OBJECT** </informaltable>

**INFORMAL\_TABLE** → <informaltable **ATRIBUTO\_INFORMAL\_TABLE**> **TGRUPO** </informaltable>

**TGRUPO** → <tgroup **ATRIBUTOS\_GENERICOS**'> opTG </tgroup>

**TGRUPO** → <tgroup ATRIBUTOS\_GENERICOS'> opTG </tgroup> **TGRUPO**

opTG → **THEAD TFOOT TBODY**

opTG → **TBODY**

opTG → **THEAD TBODY**

opTG → **TFOOT TBODY**

**THEAD** → <thead>**ROW**</thead>

**TFOOT** → <tfoot>**ROW**</tfoot>

**TBODY** → <tbody>**ROW**</tbody>

**ROW** → <row> opRow </row> | <row> opRow </row> **ROW**

**opRow** → **ENTRY** | **ENTRYBL** | **ENTRY** opRow | **ENTRYBL** ROW

**ENTRY** → <entry>opEN</entry>

**opEN** → **#TEXT** | **ITEMIZED\_LIST** | **IMPORTANTE** | **PARA** | **SIMPLE\_PARA** | **MEDIA\_OBJECT** | **COMMENT** | **ABSTRACTO**

opEN → **#TEXT** opEN | **ITEMIZED\_LIST** opEN | **IMPORTANTE** opEN | **PARA** opEN | **SIMPLE\_PARA** opEN | **MEDIA\_OBJECT** opEN | **COMMENT** opEN | **ABSTRACTO** opEN

**ENTRYBL** → <entrytbl **ATRIBUTOS\_GENERICOS**>**THEAD** **TBODY**<entrytbl>

**ENTRYBL** → <entrytbl **ATRIBUTOS\_GENERICOS**>**TBODY**</entrytbl>

**DIRECCION** → <address> </address>

**DIRECCION** → <address>opAdress</address>

opAddress → **#TEXT** | **CALLE** | **CIUDAD** | **ESTADO**| **TELEFONO**| **EMAIL**

opAddress → **#TEXT** opAddress | **CALLE** opAddress | **CIUDAD** opAddress | **ESTADO** opAddress | **TELEFONO** opAddress | **EMAIL** opAddress

**CALLE** →<street>opDatos</street>

**CIUDAD** → <city> opDatos</city>

**ESTADO** → <state>opDatos</state>

**EMAIL** → <email>opDatos</email>

**TELEFONO** → <phone>opDatos</phone>

**AUTHOR** → <author>opAuthor</author>

opAuthor → **FIRSTNAME** | **SURNAME** | **FIRSTNAME** opAuthor | **SURNAME** opAuthor

**FIRSTNAME** →<firstname>opDatos</firstname>

**SURNAME** → <surname>opDatos</surname>

**FECHA** → <date>opDatos</date>

**DERECHOS** → <copyright>**YEAR** **HOLDER**</copyright>

**DERECHOS** → <copyright>**YEAR**</copyright>

**YEAR** → <year>opDatos</year> | <year>opDatos</year> **YEAR**

**HOLDER** → <holder>opDatos</holder> | <holder>opDatos</holder> **HOLDER**

opDatos → **#TEXT** | **LINK** | **EMPHASIS** | **COMMENT** | **#TEXT** opDatos | **LINK** opDatos | **EMPHASIS** opDatos| **COMMENT** opDatos

**COMMENT** → <comment>OpSECL</comment>

OpSECL → **#TEXT** | **EMPHASIS** | **LINK** | EMAIL | **AUTHOR** | **COMMENT**

OpSECL → **#TEXT** OpSECL | **EMPHASIS** OpSECL| **LINK** OpSECL| **EMAIL** OpSECL| **AUTHOR** OpSECL|

**ATRIBUTOS\_GENERICOS** → aling=”**VALOR**” | char=”**#TEXT**” | charoff=”**#NUM**” | colname=’**#TEXT**” | cols=”**#NUM**” | colsep=”**#NUM**” | nameed=”**#TEXT**” | namest=”**#TEXT**” | rowsep=”**#NUM**” | spanname=”**#NUM**” | tgroupstyle=”**#NUM**”

**VALOR** → center | char | justify | left | rigth

**ATRIBUTO\_INFORMAL\_TABLE** → colsep=”**#NUM**” | frame=”**VALOR\_F**” | label | orient=”valor\_orient” | pgwide=”#NUM” | rowsep=”#NUM” | rowheader=”**VALOR\_ROWHEADER**” | tabstyle | tocentry

**VALOR\_F** → above | all | below | border | bottom | box| hsides | lhs| none |rhs | sides | top | topbot | void | vsides

**VALOR\_ORIENT** → land | port

**VALOR\_ROWHEADER** → firstcol | headers | norowheader

# 

# ANALIZADOR LEXICO (AL)

## Analizador léxico

Un analizador Léxico es la primera tarea que realiza un compilador, a su vez es una subrutina del Analizador Sintáctico o Parser.

El lexer es un programa el cual una vez que ingresamos un archivo, lo convierte en una secuencia de caracteres. Luego, los agrupa en palabras con significado propio. Posteriormente, busca los componentes léxicos o palabras que componen el programa fuente, según reglas o patrones.

En otras palabras, el programa Lexer lee carácter por carácter , y de acuerdo a las reglas que definimos agrupa los caracteres y retorna los componentes léxicos que tienen sentido propio dentro del lenguaje. Los caracteres que no logra reconocer los marca como un error.

Veremos a continuación algunas definiciones:

* Un **TOKEN** o **COMPONENTE LÉXICO** es una unidad léxica indivisible con significado único dentro del lenguaje. Representa la salida del Analizador Léxico. Es la representación usada en el lenguaje.
* Un **PATRÓN** es una ***expresión regular*** que permite identificar unívocamente un tipo de token y referenciar al conjunto de todos los tokens que se ajustan a él. Es el conjunto de reglas asociadas a un token.
* Un **LEXEMA** es una cadena de caracteres que encaja con un patrón que describe un componente léxico, es decir, es como una instancia de un patrón. Es una palabra encontrada en el código fuente.
* **GRAMÁTICA**: una gramática consiste de un conjunto de reglas de producción, que pueden ser aplicadas en cualquier orden consistente, con el propósito de derivar un string terminal.

## Tareas que realiza un analizador léxico

Las tareas que realizará nuestro Analizador Léxico serán:

Reconocer los componentes léxicos en la entrada.

Eliminar los espacios en blanco, los caracteres de tabulación, los saltos de línea y de página y otros caracteres propios del dispositivo de entrada.

Eliminar los comentarios de entrada.

Detectar errores léxicos.

## Palabras reservadas.

En el caso del lenguaje sobre el cual nos toca realizar el Lexer (DockBook) las palabras reservadas, son consideradas como **Tokens**.

## Procedimiento del AL

Se aplica un método para reconocer los posibles patrones, es decir que, en este caso las expresiones regulares son útiles para describir formalmente los patrones de los tokens.

Se reconocen los tokens, es decir que, los autómatas finitos reconocen los lenguajes por medio de dichas expresiones regulares.

Se realizan acciones preestablecidas al reconocer el token correspondiente.

## Errores detectables por el AL

Algunos ejemplos de errores que detecta nuestro analizador léxico son:

* Caracteres ilegales en el programa fuente, es decir, caracteres inválidos, entre otros.
* Errores de ortografía en palabras reservadas. Por lo cual se puede decir que detecta etiquetas mal escritas y también utilización de símbolos no permitidos.

## Modo de Obtención del Intérprete

En esta etapa se definieron los Tokens basándonos en la Gramática, con la ayuda de la librería PLY en el lenguaje Python. Después de eso, modificamos el formato y añadimos los tokens específicos de la estructura DocBook para que sean reconocidos por nuestro Lexer. Posteriormente añadimos funciones específicas para una mejor ejecución del programa.

## Modo de Ejecución del Intérprete

Los pasos para la ejecución del intérprete son:

***Consideraciones:*** *El ejecutable debe estar en la misma carpeta del documento de prueba. El documento de prueba debe estar en una carpeta que se llama “test” {se lo hizo así, para que sea más directa la búsqueda del archivo}*

1. El usuario localiza la carpeta src, donde encontrará allí un archivo ejecutable de nombre Parser.exe
2. Posteriormente deberá seleccionar en un listado el archivo .xml a controlar.

***El intérprete logrará:***

⮚Funcionar correctamente tanto de forma interactiva como ejecutando las instrucciones desde los archivos de ejemplo.

⮚Indicar como salida si el análisis fue exitoso (archivo correctamente codificado, sin errores), en otro caso indicar los errores existentes (indicando tipo de error, número de línea y cadena que generó el error)

# ANALIZADOR SINTACTICO (AS)

## Analizador Sintáctico.

Un ***analizador sintáctico*** o **parser** es un programa que es parte de un compilador. La tarea del analizador es la descomposición y transformación de las entradas en un formato utilizable para su posterior procesamiento. Se analiza una cadena de instrucciones en un lenguaje de programación y luego se descompone en sus componentes individuales.

## Tareas que realiza un AS

Para analizar un texto, por ejemplo, los analizadores suelen utilizar un analizador léxico separado (llamado lexer), que descompone los datos de entrada en fichas (símbolos de entrada como palabras). Los Lexers son por lo general máquinas finitas, que siguen la gramática regular y por lo tanto aseguran un desglose adecuado. Los tokens obtenidos de esta manera sirven como caracteres de entrada para el analizador sintáctico.

## Modo de Obtención del Intérprete

La codificación del ***parser*** utiliza la misma librería (PLY) utilizada para el Lexer. De esta forma, adaptamos y definimos las reglas de gramática en DocBock.

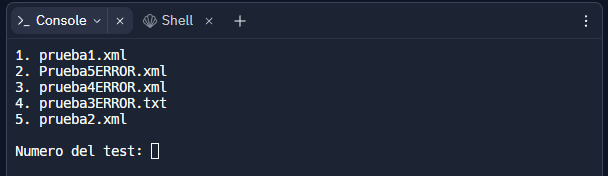
## Modo de Ejecución del Intérprete

Para ejecutar el analizador sintáctico, se lo puede ejecutar directamente a través del archivo parser.exe, y del mismo modo que el leer, debemos ingresar por teclado la dirección de donde se encuentran los archivos de prueba, en la cual luego de realizar el análisis léxico, nos da el resultado de si está correctamente ordenado. Además, como adicional, genera un archivo ‘pruebax.html’ donde se traduce los tags DocBook a HTML.

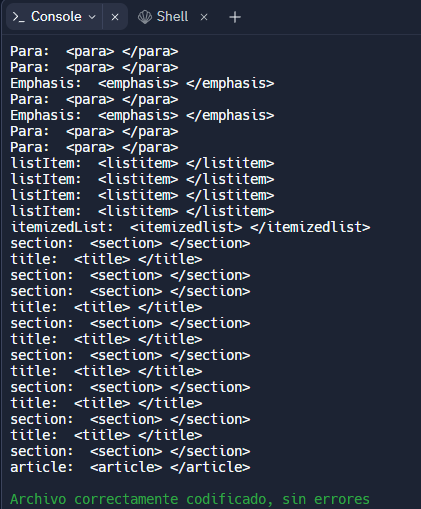
# EJEMPLOS DE IMPLEMENTACIÓN: LEXER-PARSER Y TRADUCCIÓN A HTML.

**1er:** ejecutamos el Parser.exe, lo cual nos va a abrir la consola con el listado de archivos que se encuentren en la carpeta test.

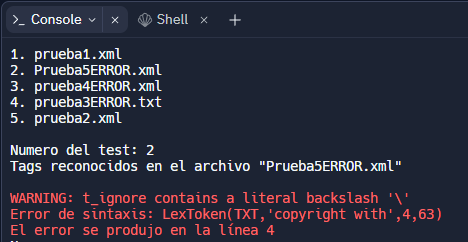
**2do:** Ingresemos el número de prueba.



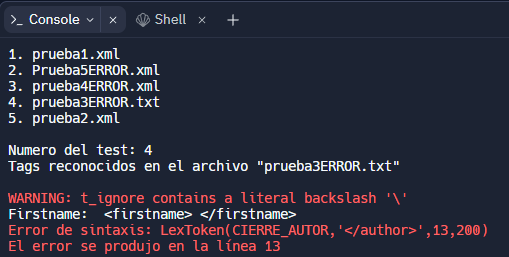
Este es un ejemplo de la prueba2.xml lo cual es un archivo sin errores léxicos ni sintácticos.



La prueba “Prueba5ERROR.xml” posee un error léxico en el token <copyrigth> que no posee atributos el atributo with=”” lo cual en esta prueba se lo introdujo a propósito.



Por último, en este ejemplo posee un error sintáctico ya que nos faltó escribir el token </author> que es el cierre de <author>.



# 

# 

# 

# GLOSARIO

* **LIBRERÍA:** Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.
* **INTÉRPRETE**: Intérprete o interpretador es un [programa informático](https://es.wikipedia.org/wiki/Programa_(computaci%C3%B3n)) capaz de analizar y ejecutar otros programas.
* **USUARIO**: Se entiende por usuario a un conjunto de permisos y de recursos asignados a un operador como parte de una red informática, y que bien puede ser una persona, un programa informático o un computador.
* **PLY(PYTHON):**Es una herramienta [de análisis](https://en.wikipedia.org/wiki/Parsing) escrita puramente en [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) . Es, en esencia, una reimplementación de [Lex](https://en.wikipedia.org/wiki/Lex_(software)) y [Yacc](https://en.wikipedia.org/wiki/Yacc) originalmente en [lenguaje](https://en.wikipedia.org/wiki/C_(programming_language)) C. Fue escrito por [David M. Beazley](https://en.wikipedia.org/wiki/David_M._Beazley) . PLY utiliza la misma técnica de análisis [LALR](https://en.wikipedia.org/wiki/LALR) que Lex y Yacc. También cuenta con amplias funciones [de depuración](https://en.wikipedia.org/wiki/Debugging) y notificación de errores.
* **ANÁLISIS**[**LALR**](https://en.wikipedia.org/wiki/LALR) o **Look-Ahead, Left-to-right**: Es un analizador de derivación por derecha.
* **ARCHIVO EJECUTABLE**: un ejecutable o archivo ejecutable, es tradicionalmente un [archivo binario](https://es.wikipedia.org/wiki/Archivo_binario), cuyo contenido se interpreta por el [ordenador](https://es.wikipedia.org/wiki/Ordenador) como un [programa](https://es.wikipedia.org/wiki/Programa_(computaci%C3%B3n)).

# CONCLUSIÓN

Durante el desarrollo del analizador léxico, no nos encontramos con mayores complicaciones debido a la utilización de la librería PLY.lex, librería que resultó de gran ayuda y con una documentación explicada a un nivel bastante amena para comprender en la página oficial.

Sin embargo, surgieron dificultades al trabajar en el desarrollo del parser. Tuvimos que enfrentarnos a la tarea de generar una gramática compleja y extensa, sumado a una falta de organización en el grupo en cuanto a la coordinación de tareas. Además, la escasa familiaridad con herramientas de control de versiones como GitHub incrementó las posibilidades de introducir errores en el código sin poder detectarlos eficientemente.

Por otro lado, debido al limitado tiempo de aprendizaje del lenguaje Python y a la falta de conocimientos en programación orientada a objetos (POO), paradigma funcional, el parser fue implementado con un enfoque monolítico y estructurado. Esta elección resulta desventajosa a la hora de mantener y escalar el código en el futuro.

# BIBLIOGRAFÍA Y REFERENCIAS WEB

* *DEFINICION.DE*. (s.f.). Obtenido de https://definicion.de/html/
* *DocBook 5.1: The Definitive Guide*. (s.f.). Obtenido de https://tdg.docbook.org/tdg/5.1/
* *DocBook Element Reference*. (s.d.). Récupéré sur https://tdg.docbook.org/tdg/5.1/ref-elements.html
* *Elements Article*. (s.f.). Obtenido de https://tdg.docbook.org/tdg/5.1/article.html
* Wikimedia Foundation, I. (s.f.). *Wikipedia.org*. Obtenido de https://en.wikipedia.org/wiki/Comparison\_of\_parser\_generators
* *XMLmind XML Editor*. (s.f.). Obtenido de http://www.xmlmind.com/xmleditor/