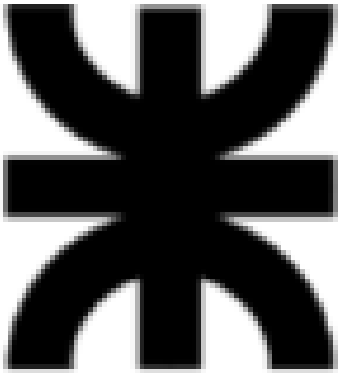
TRABAJO PRÁCTICO INTEGRADOR

Sintaxis y Semántica de los Lenguajes

Ingeniería en Sistemas de Información

Primer Cuatrimestre



UNIVERSIDAD TECNOLÓGICA NACIONAL

Integrantes:

- *GONZALEZ, Micaela*
- *MOLINAS GONZALEZ, Victor Adan*
- *TRESPALACIOS MARTINEZ, Carlos Daniel*

Año lectivo: 2023

Fecha de entrega: 30/04/23

Grupo: G3

ÍNDICE

INTRODUCCIÓN

El objetivo de este trabajo práctico integrador es aplicar los conocimientos teóricos y técnicas aprendidas sobre gramática, lenguajes de programación y procesamiento léxico y sintáctico. Se centra en la creación, diseño y procesamiento de lenguajes, específicamente en la gramática de un metalenguaje muy utilizado llamado XML, en su versión de DocBook.

Para lograr esto, se desarrollará un Lexer y Parser utilizando Python junto a la librería PLY. Estas herramientas permitirán analizar y verificar la gramática del XML de DocBook y generar como resultado un archivo HTML que refleje el contenido del XML procesado.

El Lexer se encargará de analizar el texto de entrada, identificando los diferentes tokens que conforman el XML de DocBook, como etiquetas, atributos y contenido de texto. El parser utilizará estos tokens para construir una estructura de datos que represente la jerarquía y relaciones entre los elementos del XML de DocBook.

El uso de la librería PLY simplifica la implementación del lexer y parser, ya que proporciona herramientas y funcionalidades específicas para el análisis léxico y sintáctico de lenguajes. Esto permite enfocarse en la definición de la gramática y reglas de procesamiento del XML de DocBook.

En resumen, este trabajo práctico integrador combina el uso de gramáticas, lenguajes de programación y procesamiento léxico y sintáctico para analizar, verificar y transformar el XML de DocBook en un formato HTML más adecuado para su visualización y uso en la web.

GRAMÁTICA

Símbolos Terminales :

- <?xml
- <!DOCTYPE article>
- version
- encoding
- <article>
- </article>
- <title>
- </title>
- <info>
- </info>
- <abstract>
- </abstract>
- <author>
- </author>
- <personname>
- </personname>
- <firstname>
- </firstname>
- <surname>
- </surname>
- <date>
- </date>
- <year>
- </year>
- <copyright>
- </copyright>
- <address>
- </address>
- <city>
- </city>
- <state>
- </state>
- <postcode>
- </postcode>
- <street>
- </street>
- <email>
- </email>
- <phone>

- </phone>
- <section>
- </section>
- <simplesect>
- </simplesect>
- <itemizedlist>
- </itemizedlist>
- <listitem>
- </listitem>
- <emphasis>
- </emphasis>
- <para>
- </para>
- <simpara>
- </simpara>
- <comment>
- </comment>
- <important>
- </important>
- <informaltable>
- </informaltable>
- <tgroup>
- </tgroup>
- <thead>
- </thead>
- <tfoot>
- </tfoot>
- <tbody>
- </tbody>
- <row>
- </row>
- <entry>
- </entry>
- <entrytbl>
- </entrytbl>
- <holder>
- </holder>
- <mediaobject>
- </mediaobject>
- <videoObject>
- </videoObject>
- <imageObject>
- </imageObject>

- <videodata

Simbolos No Terminales :

- TITLE
- ART
- INFO
- AUT
- PER
- FIR
- SUR
- SEC
- PARA

Tokens:

- URL: protocolo://dominio:puerto/ruta#fragmento
- TEXTO: Cadena de caracteres que no incluye los caracteres <,> ni &.
- CODING: Tipos de codificaciones (ISO-8859-1 /UTF-8 y otros tipos de codificaciones).
- RUTA: Directorio de archivos imagedata y videodata

PRODUCCIONES

Bloque Principal:

TOKEN INICIO:

Σ → <?xml version="NUMERO.NUMERO" encoding="CODING"> DOC ART

DOC→ <!DOCTYPE article>

ART→ <article> INFO TITLE ETIQUETAS </article>

ART→ <article> INFO ETIQUETAS </article>

ART→ <article> TITLE ETIQUETAS </article>

Casos generales :

ETIQUETAS → ETICOM ETISEC

ETIQUETAS → ETICOM

INFO:

INFO → <info> ETIQUETAINFO </info>

ETIQUETAINFO → AUTHOR ETIQUETAINFO

ETIQUETAINFO → DATE ETIQUETAINFO

ETIQUETAINFO → COPYRIGHT ETIQUETAINFO

ETIQUETAINFO → TITLE ETIQUETAINFO

ETIQUETAINFO → AUTHOR

ETIQUETAINFO → DATE

ETIQUETAINFO → COPYRIGHT

ETIQUETAINFO → TITLE

ETIQUETAINFO → *MEDIAOBJECT* ETIQUETAINFO

ETIQUETAINFO → *MEDIAOBJECT*

ETIQUETAINFO → *ABSTRACT* ETIQUETAINFO

ETIQUETAINFO → *ABSTRACT*

ETIQUETAINFO → *ADDRESS* ETIQUETAINFO

ETIQUETAINFO → *ADDRESS*

Section:

SECT→ <section> INFO TITLE ETICOM </section> SECT

SECT→ <section> INFO ETICOM </section> SECT

SECT→ <section> TITLE ETICOM </section> SECT

SECT→ <section> INFO TITLE ETICOM </section>

SECT→ <section> INFO ETICOM </section>

SECT→ <section> TITLE ETICOM </section>

BUCLE CON SIMPLE SECTIONS INTERNOS:

SECT→ <section> INFO TITLE ETICOM ETISEC </section> SECT

SECT→ <section> INFO ETICOM ETISEC </section> SECT
SECT→ <section> TITLE ETICOM ETISEC </section> SECT

SECT→ <section> INFO TITLE ETICOM ETISEC </section>
SECT→ <section> INFO ETICOM ETISEC </section>
SECT→ <section> TITLE ETICOM ETISEC </section>

ETISEC:

ETISEC→ SSEC
ETISEC→ SECT

SimpleSec:

SSEC→ <simplesect> INFO </simplesect>
SSEC→ <simplesect> INFO TITLE </simplesect>
SSEC→ <simplesect> INFO ETICOM </simplesect>
SSEC→ <simplesect> INFO TITLE ETICOM </simplesect>

SSEC→ <simplesect> INFO </simplesect>SSEC
SSEC→ <simplesect> INFO TITLE </simplesect>SSEC
SSEC→ <simplesect> INFO ETICOM </simplesect>SSEC
SSEC→ <simplesect> INFO TITLE ETICOM </simplesect> SSEC

ETIQUETAS COMPARTIDAS ART SECT SSEC IMPORTANT lisitem:

ETICOM → PARA ETICOM
ETICOM → PARA

ETICOM → SIMPARA ETICOM
ETICOM → SIMPARA

ETICOM → COMMENT ETICOM
ETICOM → COMMENT

ETICOM → INFTABLE ETICOM
ETICOM → INFTABLE

ETICOM → ITEMLIST ETICOM
ETICOM → ITEMLIST

ETICOM → IMPORTANT ETICOM
ETICOM → IMPORTANT

ETICOM → *MEDIAOBJECT ETICOM*
ETICOM → *MEDIAOBJECT*

ETICOM → *ABSTRACT ETICOM*
ETICOM → *ABSTRACT*

ETICOM → *ADDRESS ETICOM*
ETICOM → *ADDRESS*

ETIQUETAS COMPARTIDAS PARA:

ETICOMP → INFTABLE ETICOMP
ETICOMP → INFTABLE

ETICOMP → ITEMLIST ETICOMP
ETICOMP → ITEMLIST

ETICOMP → IMPORTANT ETICOMP
ETICOMP → IMPORTANT

ETICOMP → *MEDIAOBJECT ETICOMP*
ETICOMP → *MEDIAOBJECT*

ETICOMP → *ABSTRACT ETICOMP*
ETICOMP → *ABSTRACT*

ETICOMP → *ADDRESS ETICOMP*
ETICOMP → *ADDRESS*

ETICOMP → TEXTO ETICOMP
ETICOMP → TEXTO

ETICOMP → EMPHASIS ETICOMP
ETICOMP → EMPHASIS

ETICOMP → VIDIMALINK ETICOMP
ETICOMP → VIDIMALINK

ETICOMP → EMAIL ETICOMP
ETICOMP → EMAIL

ETICOMP → AUTHOR ETICOMP
ETICOMP → AUTHOR

ETICOMP → COMMENT ETICOMP
ETICOMP → COMMENT

ETIQUETAS COMPARTIDAS Street, City, State, Phone, Email, Date, Year ,Holder

ETICOM3 → TEXTO ETICOM3
ETICOM3 → TEXTO

ETICOM3 → VIDIMALINK ETICOM3
ETICOM3 → VIDIMALINK

ETICOM3 → EMPHASIS ETICOM3
ETICOM3 → EMPHASIS

ETICOM3 → COMMENT ETICOM3
ETICOM3 → COMMENT

ETIQUETAS COMPARTIDAS SimPara, Emphasis, Comment, Link

ETICOM4 → TEXTO ETICOM4
ETICOM4 → TEXTO

ETICOM4 → EMPHASIS ETICOM4
ETICOM4 → EMPHASIS

ETICOM4 → VIDIMALINK ETICOM4
ETICOM4 → VIDIMALINK

ETICOM4 → EMAIL ETICOM4
ETICOM4 → EMAIL

ETICOM4 → AUTHOR ETICOM4
ETICOM4 → AUTHOR

ETICOM4 → COMMENT ETICOM4
ETICOM4 → COMMENT

ETIQUETAS COMPARTIDAS PersonName

ETIAUT → FIRSTNAME ETIAUT
ETIAUT → FIRSTNAME

ETIAUT → SURNAME ETIAUT
ETIAUT → SURNAME

ETIQUETAS COMPARTIDAS Address

ETIADD → TEXTO ETIADD
ETIADD → TEXTO

ETIADD → STREET ETIADD
ETIADD → STREET

ETIADD → CITY ETIADD
ETIADD → CITY

ETIADD → STATE ETIADD
ETIADD → STATE

ETIADD → PHONE ETIADD
ETIADD → PHONE

ETIADD → EMAIL ETIADD
ETIADD → EMAIL

ETIQUETAS COMPARTIDAS entry

ETIENTRY → TEXTO ETIENTRY
ETIENTRY → TEXTO

ETIENTRY → ITEMLIST ETIENTRY
ETIENTRY → ITEMLIST

ETIENTRY → IMPORTANT ETIENTRY
ETIENTRY → IMPORTANT

ETIENTRY → PARA ETIENTRY
ETIENTRY → PARA

ETIENTRY → SIMPARA ETIENTRY
ETIENTRY → SIMPARA

ETIENTRY → MEDIAOBJECT ETIENTRY
ETIENTRY → MEDIAOBJECT

ETIENTRY → COMMENT ETIENTRY
ETIENTRY → COMMENT

ETIENTRY → ABSTRACT ETIENTRY
ETIENTRY → ABSTRACT

SIMPARA:

SIMPARA → <simpara> ETICOM4 </simpara>

EMPHASIS:

EMPHASIS → <emphasis> ETICOM4 </emphasis>

COMMENT:

COMMENT → <comment> ETICOM4 </comment>

VIDIMALINK: expresión regular que incluye las siguientes etiquetas**:**

LINK → <link xlink:href = "URI">ETICOM4

IMGD → <imagedata fileref="RUTA" />

VIDEOD → <videodata fileref="RUTA" />

ACLARACIÓN: Debido a constantes problemas al momento de generar la etiqueta de clausura **</link>**, se optó por no utilizarla.

PARA:

PARA→ <para> ETICOMP </para>

TITLE:

TITLE→ <title> TEXTO </title>

LIST:

ITEMLIST → <itemizedlist> LISITEM </itemizedlist>

LISTEM → <listitem> ETICOM </listitem> LISITEM

LISTEM → <listitem> ETICOM </listitem>

IMPORTANT:

IMPORTANT→ <important> TITLE ETICOM </important>

IMPORTANT→ <important> ETICOM </important>

DATA:

FIRSTNAME → <firstname> ETICOM3 </firstname>

SURNAME → <surname> ETICOM3 </surname>

STREET → <street> ETICOM3 </street>

CITY → <city> ETICOM3 </city>

STATE → <state> ETICOM3 </state>

PHONE → <phone> ETICOM3 </phone>

EMAIL → <email> ETICOM3 </email>

DATE → <date> ETICOM3 </date>

YEAR → <year> ETICOM3 </year> YEAR

YEAR → <year> ETICOM3 </year>

HOLDER → <holder> ETICOM3 </holder> HOLDER

HOLDER → <holder> ETICOM3 </holder>

ADDRESS → <address> ETIADD </address>
AUTHOR → <author> ETIAUT </author>

COPYRIGHT → <copyright> YEAR HOLDER</copyright>
COPYRIGHT → <copyright> YEAR </copyright>

MULTIMEDIA:

MEDIAOBJECT → <mediaobject> INFO OBJECT </mediaobject>
MEDIAOBJECT → <mediaobject> OBJECT </mediaobject>

VIDOEOBJ → <videoObject> INFO VIDIMALINK </videoObject>
VIDOEOBJ → <videoObject> VIDIMALINK </videoObject>
IMAGEOBJECT → <imageObject> INFO VIDIMALINK </imageObject>
IMAGEOBJECT → <imageObject> VIDIMALINK </imageObject>
OBJECT → VIDOEOBJ OBJECT | IMAGEOBJECT OBJECT
OBJECT → VIDOEOBJ | IMAGEOBJECT

TABLA:

INFTABLE → <informaltable> MEDIAOBJECT </informaltable>
INFTABLE → <informaltable> TGROUP </informaltable>

TGROUP → <tgroup> THEAD TFOOT TBODY </tgroup>
TGROUP → <tgroup> THEAD TBODY </tgroup>
TGROUP → <tgroup> TFOOT TBODY </tgroup>
TGROUP → <tgroup> TBODY </tgroup>

TGROUP → <tgroup> THEAD TFOOT TBODY </tgroup> TGROUP
TGROUP → <tgroup> THEAD TBODY </tgroup> TGROUP
TGROUP → <tgroup> TFOOT TBODY </tgroup> TGROUP
TGROUP → <tgroup> TBODY </tgroup> TGROUP

THEAD → <thead> ROW </thead>

TFOOT → <tfoot> ROW </tfoot>

TBODY → <tbody> ROW </tbody>

ROW → <row> ETIROW </row> ROW
ROW → <row> ETIROW </row>

ETIROW:

ETIROW → ENTRY ETIROW
ETIROW → ENTRYTBL ETIROW
ETIROW → ENTRYTBL
ETIROW → ENTRY

ENTRY → <entry> ETIENTRY </entry> ENTRY
ENTRY → <entry> ETIENTRY </entry>

ENTRYTBL → <entrytbl> THEAD TBODY </entrytbl> ENTRYTBL
ENTRYTBL → <entrytbl> TBODY </entrytbl> ENTRYTBL

ENTRYTBL → <entrytbl> THEAD TBODY </entrytbl>
ENTRYTBL → <entrytbl> TBODY </entrytbl>

ABSTRACT:

ABSTRACT → <abstract> TITLE ETIAUX </abstract>
ABSTRACT → <abstract> ETIAUX </abstract>

ETIAUX:

ETIAUX → PARA ETIAUX
ETIAUX → SIMPARA ETIAUX
ETIAUX → PARA
ETIAUX → SIMPARA

Analizador Léxico

El siguiente analizador léxico o Lexer está basado en el lenguaje Python con el uso de la librería **ply**. A su vez, trabaja con funciones con expresiones regulares tales como :

t_VIDIMALINK:

Esta función es utilizada para detectar las etiquetas de imagedata, videodata, y link respectivamente.

```
def t_VIDIMALINK(t):

r'<(?:imagedata|videodata|link)\s+(fileref|xlink:href)\s*=\s*"((ht
tp|ftp)(s)?://)?[^<>&]*([[^<>&]+:[^<>&]+?]?[[\\[^<>&]]+?[#[^<>&]+]
?])?"[/]?>'
    file.write("<a href = "+t.value)
    return (t)
```

La expresión regular también contiene otra definición interna de una URL, debido a que no fue posible llamar una función dentro de la expresión regular.
Coincide con cadenas con los siguientes formatos:

<videodata fileref="http://example.com/video.mp4" />
<imagedata fileref="http://example.com/image.jpg" />
<link xlink:href="http://example.com" >

t_INICIO:

```
def t_INICIO(t):
    r'<\?xml[\ ]+(version[\ ]*=[\ ]*\"\d.\d\" [\ ]+)?(encoding[\
]*=[\ ]*\" [\S]+ \"[\ ]*)? \?>'

    file.write("<!DOCTYPE html><html><head></head><body>")
    return (t)
```

Es utilizada para detectar el xml del docbook, con su respectiva versión, encoding o si se le quiere simplemente el tag: *<?xml version="1.0" encoding="UTF-8"?>*

t_DOCTYPE:

```
def t_DOCTYPE(t):
    r'<\!DOCTYPE[\s]+ (article|book|chapter)[\s]*>'
    file.write("<!DOCTYPE html><html><head></head><body>")
    return (t)
```

Esta función toma la etiqueta de Doctype que contiene el docbook:

<!DOCTYPE article>
<!DOCTYPE>

t_TEXTO:

```
def t_TEXTO(t):
    r'[^<>&]+'
    file.write(t.value)
    return (t)
```

Las funciones restantes definen los tags de apertura y clausura, por ejemplo:

Para la apertura:

```
def t_AABSTRACT(t):
    r'<abstract>'
    return (t)
```

Para la clausura:

```
def t_CABSTRACT(t):
    r'</abstract>'
    return (t)
```

t_error:

Esta función notifica por consola al usuario si un token no ha sido reconocido, especificando la etiqueta y la línea en la que se encuentra el error en el archivo.

```
def t_error(t):
    # print("Se encontró el siguiente token no reconocible ",
t.value[0])
    print("Token no reconocido: ", t.value + " [Linea:", t.lineno,
"]")
    t.lexer.skip(1)
```

Conversión a HTML

La conversión de las etiquetas de Docbook a HTML se genera en el momento que el Lexer reconoce la etiqueta específica, agregando el estilo si lo requiere.

```
def t_AARTICLE(t):
    r'<article>'
    file.write(
             '<div  style="background-color:  green;  color:  white;
font-size: 8pt;"><p>')
    return (t)
```

Analizador Sintáctico

El analizador sintáctico o Parser, al igual que el analizador léxico, está basado en el lenguaje Python, haciendo uso de la librería: *"**ply**"*.
Primeramente, el Docbook, ingresado por consola o por archivo, debe ser correcto. Una vez que esto se confirma mediante el Lexer, el Parser, para poder cumplir su función, importa los *Tokens* del Lexer y las etiquetas definidas en el archivo "Etiquetas.py" de la siguiente manera:

```
from Lexer import tokens
from etiquetas import *
```

Además, el Parser debe contar con la gramática correcta. Comenzando con la definición de la función que representa al símbolo distinguido (Σ):

```
def p_sigma(p):
    '''sigma : INICIO art
        | DOCTYPE art
        | INICIO DOCTYPE art'''
    print('EL DOCBOOK ES CORRECTO')
```

Las gramáticas se definen de la siguiente manera:

```
def p_info(p):
    '''info : AINFO etiquetainfo CINFO'''
```

donde los Terminales se diferencian en letras mayúsculas y los No Terminales en minúsculas.

Función *search_files*

Una vez ingresada la ruta o nombre de la carpeta, esta función busca el archivo XML que fue seleccionado por el usuario.

```
def search_files(folder, extension):
    files = []
    for dirpath, dirnames, filenames in os.walk(folder):
        for filename in filenames:
            if filename.endswith(extension):
                files.append(os.path.join(dirpath, filename))
    return files
```

Cuerpo del Parser

Esta sección del código se puede analizar como 2 subsecciones.

Subsección 1

La primer subsección incluye la inicialización de las variables "*entradaDoc*", "*entrada*" y "*salida*"

```
entradaDoc = ""
entra = ""
salida = "out.html"
```

Luego el condicional *if* que verifica si se ingresa un comando por consola.

```
if len(sys.argv) > 1:  # se ingresó un comando al ejecutar el
programa
    entra = sys.argv[1]
    f = open(entra, 'r')
    salida = entra.split('.')[0] + '.html'
    entradaDoc = f.read()
```

Subsección 2

Esta subsección cumple la función de dar al usuario las opciones de:

1. ingresar el contenido del archivo Docbook manualmente por consola.
2. importar un archivo por medio de ruta o nombre de la carpeta contenedora.

Además, en caso de que la ruta o carpeta no sea encontrada, el programa emite un mensaje indicando el error.

```
else:
    print("Ingrese 1 para ingresar un docbook por consola y 2 para
ingresarlo por archivo")
    opcion = int(input())
    if opcion == 1:
        print("Ingrese el código por teclado.")
        print("Para terminar, presione Ctrl+Z en Windows o Ctrl+D
en sistemas UNIX/Linux")
        while True:
            try:
                c = input('> ')
                entra += entra + c + '\n'
            except EOFError:
```

```
                break
    elif opcion == 2:
        while True:
            print("Ingrese el nombre/dirección de la carpeta:")
            folder = input()
            files = search_files(folder, ".xml")
            if len(files) == 0:
                print("No se encontraron archivos en la carpeta
especificada.")
                continue
            print("Archivos encontrados:")
            for i, file in enumerate(files):
                print(f"{i+1}. {file}")
            print("Ingrese el número del archivo que desea
seleccionar:")
            file_num = int(input())
            if file_num < 1 or file_num > len(files):
                print("Número de archivo inválido.")
                continue
            entra = files[file_num - 1]
            file = open(entra, "r", encoding="utf-8")
            salida = entra.split('.')[0] + '.html'
            entradaDoc = file.read()
            break
    else:
        print("Ingrese una opción válida.")
        quit()

result = parser.parse(entradaDoc)
```

Definición de etiquetas compartidas

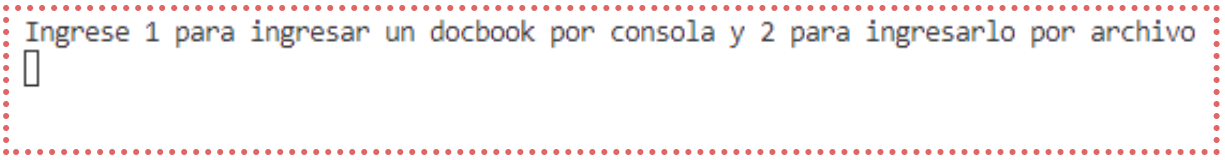
La gramática previamente establecida se construyó teniendo en cuenta el uso compartido de etiquetas. Por ejemplo, las etiquetas *FirstName, Surname, Street, City, State, Phone,Email, Date, Year, Holder* pueden hacer uso de las mismas etiquetas, las cuales son *Link, Emphasis* y *Comment*. Por lo tanto, con motivo de una mejor organización con estos conjuntos de etiquetas compartidas, se definen en un archivo llamado "Etiquetas" de extensión *.py*.

Ejemplo: etiqueta compartida

```
def p_eticom(p):
    '''eticom : para eticom
        | simpara eticom
        | comment eticom
        | inftable eticom
        | itemlist eticom
        | important eticom
        | mediaobject eticom
        | abstract eticom
        | address eticom
        | para
        | simpara
        | comment
        | inftable
        | itemlist
        | important
        | mediaobject
        | abstract
        | address'''
```

Ejecución del Analizador

Para iniciar el análisis del archivo Docbook, se debe ejecutar el archivo . Seguidamente se visualizará el siguiente mensaje:



Según la opción elegida:

1. ***ingresar un docbook por consola.***
   El usuario debe ingresar por teclado el contenido del archivo Docbook

```
Ingrese el código por teclado.
Para terminar, presione Ctrl+Z en Windows o Ctrl+D en sistemas UNIX/Linux
```

2. ***Ingresar un Docbook por archivo.***

   El usuario debe ingresar la ruta o nombre de la carpeta contenedora.

```
Ingrese el nombre/dirección de la carpeta:
```

   Luego el usuario debe elegir el archivo deseado ingresando el número del archivo correspondiente en la lista de archivos encontrados.

```
Ingrese el nombre/dirección de la carpeta:
prueba
Archivos encontrados:
1. prueba\prueba.xml
2. prueba\prueba2.xml
3. prueba\pruebafacil.xml
4. prueba\tfiejemplocampus.xml
Ingrese el número del archivo que desea seleccionar:
```

   El programa mostrará el siguiente mensaje en caso de ser correcto el archivo:

```
Ingrese el número del archivo que desea seleccionar:
1
EL DOCBOOK ES CORRECTO
```

Finalmente, se generará un archivo html en la carpeta “Prueba” como resultado de la conversión de archivos.

EJEMPLOS DE DOCBOOK

EJEMPLO 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE article>

<article>
  <info>
    <author>
      <firstname>John</firstname>
      <surname>Doe</surname>
    </author>
    <date>2023-06-13</date>
    <copyright>
      <year>2023</year>
      <holder>John Doe</holder>
    </copyright>
    <title>Sample Article</title>
  </info>
  <para>paraque</para>
  <section>
    <info>
      <title>Introduction</title>
    </info>
    <para>This is the introduction section.</para>
  </section>

  <section>
    <info>
      <title>Methods</title>
    </info>
    <para>This is the methods section.</para>
  </section>

  <section>
    <info>
      <title>Results</title>
    </info>
    <para>This is the results section.</para>
    <para>For more information, visit <link
xlink:href="http://www.example.com/videos/intro.mp4"> www.example.com.
</para>
  </section>
```

```
</article>
```

EJEMPLO 2

```
<!DOCTYPE article>
<article>
    <info>
        <title>El titulo del articulo</title>
        <author>
            <firstname>Juan</firstname>
            <surname>Perez</surname>
        </author>
    </info>
    <title>HOLA SOY UN TITULO</title>
    <important>
        <title>hola</title>
        <para>soyunparaffon</para>
    </important>
    <section>
        <title>Titulo para la seccion 1</title>
        <para>
            Esto es un parrafo
        </para>
        <para>
            Esto es un parrafo2222
        </para>
    </section>
</article>
```

CONCLUSIÓN

Durante la realización de este trabajo práctico integrador, nos enfrentamos a diversas dificultades, tanto en el aspecto creativo como en el técnico. Una de las principales dificultades surgió en la etapa de construcción de la gramática, donde nos encontramos con problemas de ambigüedad y recursividad infinita. Sin embargo, logramos abordar y resolver estas dificultades al redefinir cuidadosamente la gramática en el Parser, permitiéndonos reconstruir una versión mejorada.

El Lexer cumplió inicialmente su función correctamente, pero a medida que avanzábamos en el desarrollo del Parser, surgieron cambios necesarios en el Lexer para asegurar su correcto funcionamiento y la sincronización con el Parser.

La confección del Parser resultó ser un desafío que requirió un esfuerzo continuo hasta los últimos momentos del trabajo. Fue necesario analizar en detalle la gramática y redefinir las reglas de producción de manera precisa para lograr un análisis sintáctico efectivo.

A pesar de los desafíos enfrentados, estamos satisfechos con los resultados obtenidos y consideramos que este trabajo práctico fue una experiencia valiosa para fortalecer nuestros conocimientos en el lenguaje trabajado, porque es algo que no sabíamos cómo afrontar en un inicio pero a medida que íbamos haciendo y deshaciendo código aprendimos cómo sobrellevar todo.

Además, trabajando en equipo, aprendimos la importancia de la colaboración y la comunicación constante para superar los desafíos. Compartir ideas, discutir enfoques y apoyarnos mutuamente fue fundamental para lograr avances significativos en el proyecto.

BIBLIOGRAFÍA

DocBook 5.1: The Definitive Guide
https://www.dabeaz.com/ply/ply.html
https://ericknavarro.io/2020/03/15/26-Interprete-sencillo-utilizando-PLY/