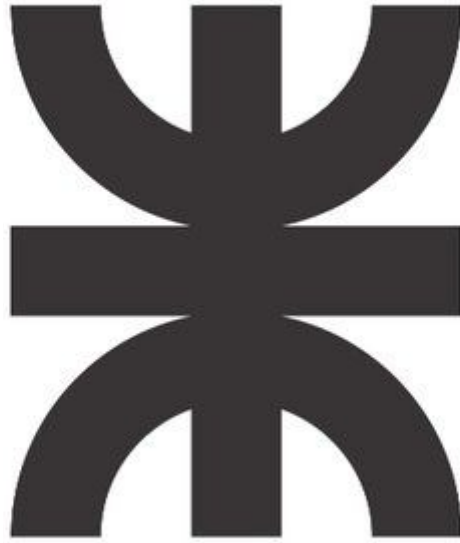


# INTÉRPRETE HTML



***INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN***

***SINTAXIS Y SEMÁNTICA DEL LENGUAJE***

**U.T.N. F.R.Re.**

PRIMER CUATRIMESTRE  
CURSO ACADÉMICO: 2020

## **Integrantes:**

- *Acosta, Matías.*
- *Carrizo, Juan Cruz.*
- *Ríos, Lucas Martín.*
- *Zoloaga, Gonzalo.*

# ***Índice***

<b>INTRODUCCIÓN</b>	<b>3</b>
<b>COMPONENTES LÉXICOS</b>	<b>4</b>
ATRIBUTOS	5
Tokens	5
PLANTILLA BÁSICA DE CÓDIGO HTML	6
<b>PRODUCCIONES DE GRAMÁTICA</b>	<b>7</b>
INICIO	7
LISTAS	9
TABLAS	10
<b>DERIVACIÓN</b>	<b>11</b>
<b>ESPECIFICACIONES DE PROGRAMA</b>	<b>12</b>
<b>ANALIZADOR LÉXICO</b>	<b>13</b>
<b>ANALIZADOR SINTÁCTICO</b>	<b>13</b>
<b>Modo de uso</b>	<b>14</b>
Para análisis mediante digitación manual:	15
Para análisis mediante archivo:	16
<b>CONCLUSIÓN</b>	<b>17</b>
<b>BIBLIOGRAFÍA Y FUENTES</b>	<b>17</b>

# INTRODUCCIÓN

El HTML (Hyper Text Markup Language) es el lenguaje con el que se escriben las páginas web. Es un lenguaje de hipertexto, es decir, un lenguaje que permite escribir texto de forma estructurada, y que está compuesto por etiquetas, que marcan el inicio y el fin de cada elemento del documento.

Un documento hipertexto no sólo se compone de texto, puede contener imágenes, sonido, vídeos, etc., por lo que el resultado puede considerarse como un documento multimedia.

Los documentos HTML deben tener la extensión html o htm, para que puedan ser visualizados en los navegadores (programas que permiten visualizar las páginas web).

Los navegadores se encargan de interpretar el código HTML de los documentos, y de mostrar a los usuarios las páginas web resultantes del código interpretado.

En el presente trabajo se procederá a explicar el desarrollo de un analizador léxico-sintáctico de HTML, el cual se utiliza para verificar si los strings utilizados son correctos y corresponden al lenguaje de programación utilizado.

El trabajo fue realizado mediante un archivo compartido Docs de Google y manteniendo una comunicación mediante la plataforma Zoom para mejor comprensión y sinergia de los integrantes en momentos de alerta por la enfermedad Covid-19 que se ha transformado en pandemia inhabilitando el contacto entre personas para no propagarlo.

Lo primero a realizar para el analizador léxico-sintáctico es una producción gramatical de los terminales y no terminales que poseerá nuestro programa y por medio de ellos como si fuera un máquina tipo acceptor dar notificaciones de si cumple o no con la gramática de HTML.

El objetivo de este trabajo es poder elaborar un intérprete de lenguaje HTML utilizando los conocimientos y estrategias adquiridas en clases, la idea es poder diferenciar bien un símbolo terminal de un símbolo no terminal y que estos deban seguir un determinado orden y jerarquía

## COMPONENTES LÉXICOS

ETIQUETAS CON CIERRE	APERTURA	CIERRE
DOCTYPE		
HTML	<html>	</html>
HEAD	<head> <ul style="list-style-type: none"><li>&lt;meta&gt;</li><li>&lt;title&gt;</li></ul>	</head> <ul style="list-style-type: none"><li>&lt;/meta&gt;</li><li>&lt;/title&gt;</li></ul>
BODY	<body>	</body>
PÁRRAFOS	<p>	</p>
ETIQUETAS BÁSICAS DE PÁRRAFO	<strong> <em> <mark> <hr> <&nbsp;   > <img>   <&nbsp;   >	</strong> </em> </mark> // // // // //
TÍTULOS	<h1>;<h2>;<h3>;<h4>;<h5>;<h6>;	</h1>;</h2>;</h3>;</h4>;</h5>;</h6>;
LISTAS	Item dentro de lista: <li>	</li>
	ORDENADA: <ol>	</ol>
	NO ORDENADA: <ul>	</ul>
SECCIONES	<section>	</section>
TABLAS	<table>	</table>
	Cabecera de tabla: <thead> Pie de tabla: <tfoot> Cuerpo de tabla: <tbody> Filas: <tr> Columnas: <th>	</thead> </tfoot> </tbody> Filas: </tr> Columnas: </th>
CONTENEDOR GENÉRICO	<div>	</div>
ENLACES	<a>	</a>

## 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.

<b>ATRIBUTOS</b>	<b><u>Globales:</u></b> class, id <b><u>Específicos:</u></b> <html>→ lang <ol> → type <li> → value <a>→ href, target <img>→ src, alt, width height
------------------	--

## Tokens

Texto.

Idioma.

URL.

Num (numero).

Atributo.

## PLANTILLA BÁSICA DE CÓDIGO HTML

---

```
<!DOCTYPE html>
<html lang="es"> <!-- Etiqueta Inicial Documento WEB -->

  <head> <!-- Inicio del Encabezado -->
    <meta charset="UTF-8" /> <!-- Set de Caracteres -->
    <title>Aquí va el Título del Documento WEB</title>
  </head>

  <body> <!-- Inicio del cuerpo -->

    <!-- Aquí va el contenido del cuerpo -->

  </body> <!-- Fin del cuerpo -->

</html> <!-- Etiqueta Final del Documento WEB -->
```

## PRODUCCIONES DE GRAMÁTICA

### INICIO

$\Sigma \rightarrow \text{INIC}$

Palabra reservada

$\text{INIC} \rightarrow \text{<!doctype html>LENG}$

$\text{LENG} \rightarrow \text{<html lang =idioma>CAB CUER</html>}$

### Encabezado

### CABEZA

$\text{CAB} \rightarrow \text{<head>F TIT</head>}$

$\text{F} \rightarrow \text{<meta charset = 'UTF - 8' />J|<meta charset = 'UTF - 8' />}$

$\text{J} \rightarrow \text{<meta name=texto content= texto />J|<meta name=texto content=texto />|}$

$\text{TIT} \rightarrow \text{<title>texto</title>}$



### CUERPO

$\text{CUER} \rightarrow \text{|<body>SENTENCIA</body>|}$

### SENTENCIAS

$\text{SENTENCIA} \rightarrow \text{PARR|TITU|SECCIÓN|ENLA|LIST|IMGG|DIVV}$

$\text{|TAB|<hr>SENTENCIA|<br>SENTENCIA|
|<hr>}$

## ETIQUETAS DE PÁRRAFO

**Q**→<strong>texto</strong>**Q**|<em>texto</em>**Q**|<mark>texto</mark>**Q**|&nbsp;**Q**|  
<strong>texto</strong>|<em>texto</em>|<mark>texto</mark>|&nbsp;|texto.

## DIVISIÓN

**DIVV**→ <div**ATG**>**SENTENCIA**</div>|<div>**SENTENCIA**</div>

## IMÁGENES

**IMGG** → <img **ATG****ATIMG**> |<img **ATIMG**>

**ATIMG**→ src URL | src URL widthAtri s heightAtri | srcURL widthAtri heightAtri |  
altAtri srcURL | altAtri src URL widthAtri | altAtri srcURL widthAtri heightAtri | altAtri  
srcURL heightAtri

## SECCIÓN

**SECCIÓN**→|<section**ATG**>**SENTENCIA**</section>|<section>**SENTENCIA**</section>|

## PÁRRAFOS

**PARR**→|<p**ATG**>**Q**</p>**SENTENCIA**|<p**ATG**>**Q**</p>|<p**ATG**>**SENTENCIA**</p>**SENT**  
**ENCIA**|<p**ATG**>**SENTENCIA**</p>|<p>**Q**</p>**SENTENCIA**|<p>**Q**</p>|<p>**SENTENCI**  
**A**</p>**SENTENCIA**|<p>**SENTENCIA**</p>

## TÍTULOS

**TITU**→<h1**ATG**>texto</h1>**SENTENCIA**|<h1**ATG**>texto</h1>|  
<h2**ATG**>texto</h2>**SENTENCIA**|<h2**ATG**>texto</h2>|  
<h3**ATG**>texto</h3>**SENTENCIA**|<h3**ATG**>texto</h3>|  
<h4**ATG**>texto</h4>**SENTENCIA**|<h4**ATG**>texto</h4>|  
<h5**ATG**>texto</h5>**SENTENCIA**|<h5**ATG**>texto</h5>|  
<h6**ATG**>texto</h6>**SENTENCIA**|<h6**ATG**>texto</h6>|  
<h1>texto</h1>**SENTENCIA**|<h1>texto</h1>|  
<h2>texto</h2>**SENTENCIA**|<h2>texto</h2>|  
<h3>texto</h3>**SENTENCIA**|<h3>texto</h3>|  
<h4>texto</h4>**SENTENCIA**|<h4>texto</h4>|  
<h5>texto</h5>**SENTENCIA**|<h5>texto</h5>|  
<h6>texto</h6>**SENTENCIA**|<h6>texto</h6>|



## LISTAS

**LIST** → `<olATGLATE>U</ol>SENTENCIA|<olATGLATE>U</ol>|`  
`<olATG>U</ol>SENTENCIA|<olATG>U</ol>|<ol>U</ol>|`  
`<ol>U</ol>SENTENCIA|<olLATE>U</ol>SENTENCIA|<olLATE>U</ol>|`  
`<ulATG>U</ul>SENTENCIA|<ulATG>U</ul>|`  
`<ul>U</ul>SENTENCIA|<ul>U</ul>`

**U** → `<li>Q</li>U|<li>Q</li>|<li>SENTENCIA</li>|<li>SENTENCIA</li>U|`  
`<liS>Q</li>U|<liS>Q</li>|<liS>SENTENCIA</li>|<liS>SENTENCIA</li>U|`

## LISTAS ATRIBUTO ESPECÍFICO

- **OL**  
**LATE** → type = Text|Num
- **LI**  
**S** → value = num

## ENLACES

**ENLA** → `<a=ATGATR>Q</a>SENTENCIA|<a=ATR>Q</a>`  
`<a=ATR>Q</a>SENTENCIA|`  
**ATR** → href url| href url target texto

## ATRIBUTOS GLOBALES

**ATG** → id = texto|class = texto| id = texto class = texto |  
class = texto id = texto

## TABLAS

**TAB**→<table BORDER="1"><caption>Q</caption>**CABEZA PIE CUERPO**</table>  
|<table><caption>Q</caption>**CABEZA PIE CUERPO**</table>|  
|<table>**CABEZA PIE CUERPO**</table>|  
|<table BORDER = "1"><caption>Q</caption>**PROD**</table>  
|<table><caption>Q</caption>**PROD**</table>  
|<table BORDER ="1">**PROD**</table>  
|<table>**PROD**</table>

*Hay tres elementos HTML que sirven para diferenciar las tres partes principales de una tabla, son:*

**CABEZA**→<thead>**PROD**</thead>

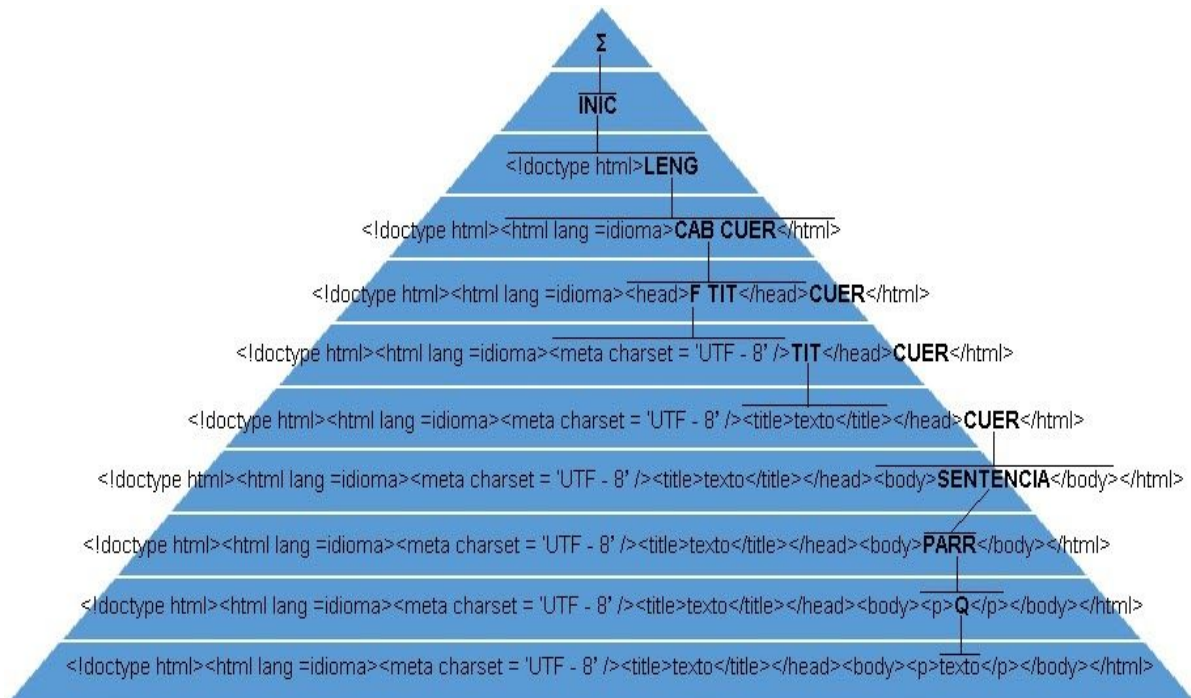
**PIE**→ <tfoot>**PROD**</tfoot>

**CUERPO**→ <tbody>**PROD**</tbody>

**GLO**→ |<td>Q</td>**GLO**|<td>Q</td>|<td>**SENTENCIA**</td>**GLO**|  
|<td>**SENTENCIA**</td>|<th>Q</th>**GLO**|<th>Q</th>|<th>**SENTENCIA**</th>|  
|**GLO**|<th>**SENTENCIA**</th>|

**PROD**→ <tr>**GLO**</tr>**PROD**|<tr>**GLO**</tr>

## DERIVACIÓN



# ESPECIFICACIONES DE PROGRAMA

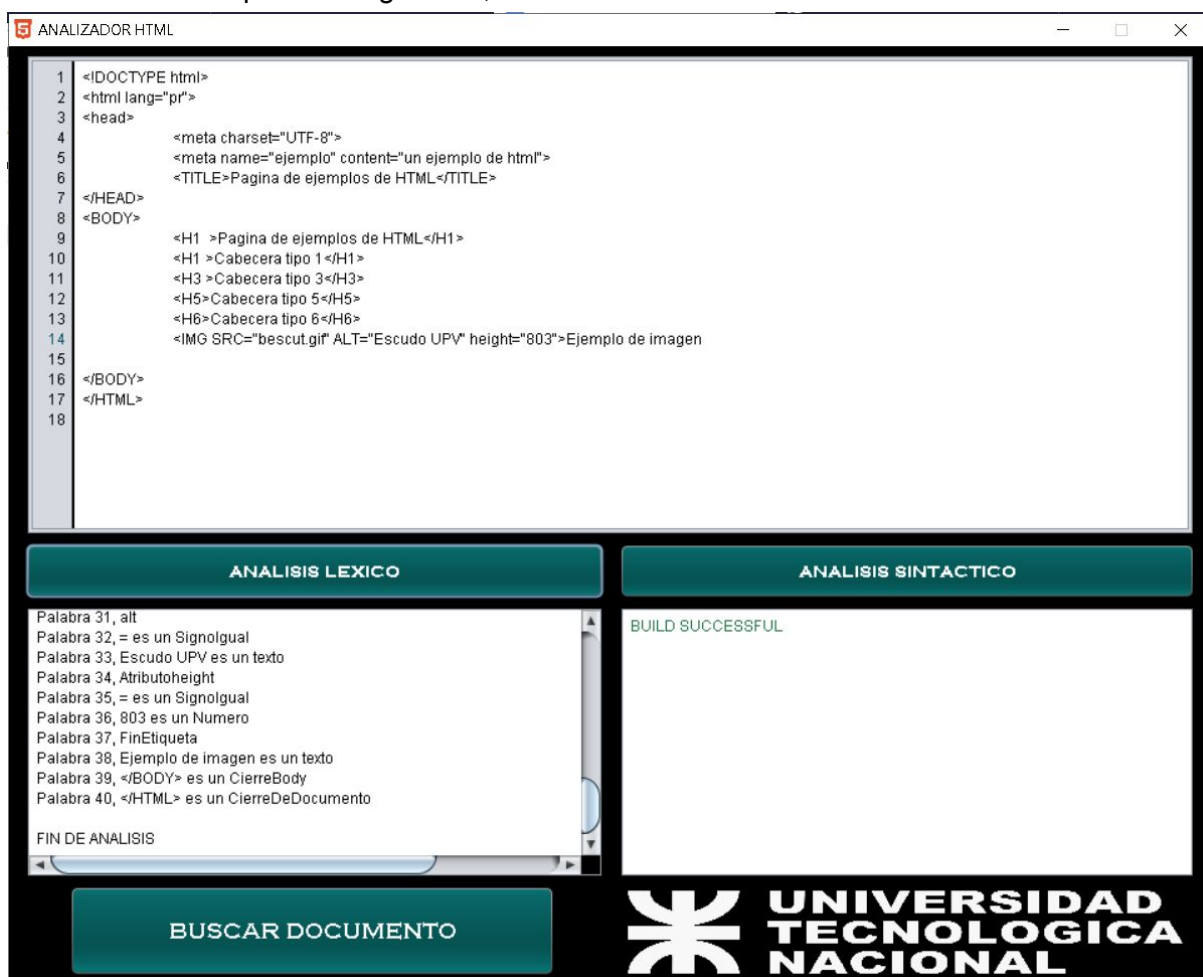
El programa lo realizamos en el software llamada NetBeans 8.2 con sus extensiones en Java y en el sistema operativo Windows 10 de 64 bits.

Para el diseño creamos un interfaz sencilla y amigable para el usuario.

Sus respectivas funciones están en el archivo src->código->"FrmPrincipal", en el cual se detalla lo que hace de acuerdo a la llegada de lo introducido.

En caso de no poseer el programa para ejecutarlo, sea con NetBeans o un ".jar", utilizamos al programa Launch4j para hacerlo un ".exe" con una versión mínima de Java de 1.8 en el cual se lo puede utilizar de la misma manera como en las anteriores aplicaciones mencionadas.

Para realizar pruebas, el usuario puede ingresar palabras cualesquiera y se devolverá lo referido al tipo de análisis que se seleccione. También, se puede ingresar todo un archivo ".txt" o ".html" que contenga texto, solo se lo debe buscar.



*Imágen del programa en ejecución.*

## ANALIZADOR LÉXICO

En esta etapa utilizamos un generador de analizadores léxicos llamado “JFLEX”, que posee una estructura sencilla para definir las reglas léxicas que mediante esta librería se nos facilita el desarrollo del programa.

El programa realiza las mismas funciones independientemente de la manera que se los ejecute. Este cuenta con un analizador en el cual se puede digitar el código y se lo analiza mediante el botón “ANÁLISIS LÉXICO” en caso de ya poseerlo se lo busca mediante el botón “BUSCAR DOCUMENTO”.

### Estructura:

Se creó una Aplicación Java en la cual definimos las reglas léxicas en el File en src->código->”Lexer.flex”. Luego en un JavaEnum en src->código->”Tokens” definimos las respectivas salidas ante la entrada del código.

## ANALIZADOR SINTÁCTICO

En esta etapa utilizamos un generador de analizadores sintácticos llamado “JavaCup” y su complemento “JavaCup11a”, que posee una estructura sencilla para definir las reglas sintácticas exigidas por el lenguaje que mediante esta librería se nos facilita el desarrollo del programa.

El programa realiza las mismas funciones independientemente de la manera que se los ejecute. Este cuenta con un analizador en el cual se puede digitar el código y se lo analiza mediante el botón “ANÁLISIS SINTÁCTICO” en caso de ya poseerlo se lo busca mediante el botón “BUSCAR DOCUMENTO”.

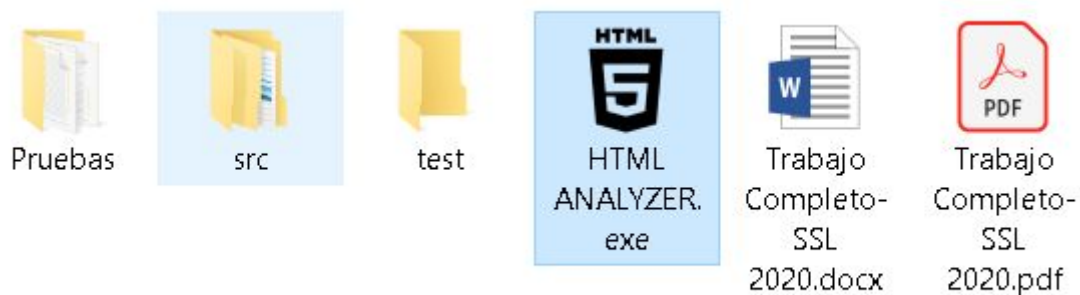
### Estructura:

Se creó una Aplicación Java en la cual definimos aquellos símbolos que deberá reconocer el orden en el lenguaje, esto se encuentra en el File src->código->”LexerCup.flex” y las reglas sintácticas en el File en src->código->”Syntax.cup”.

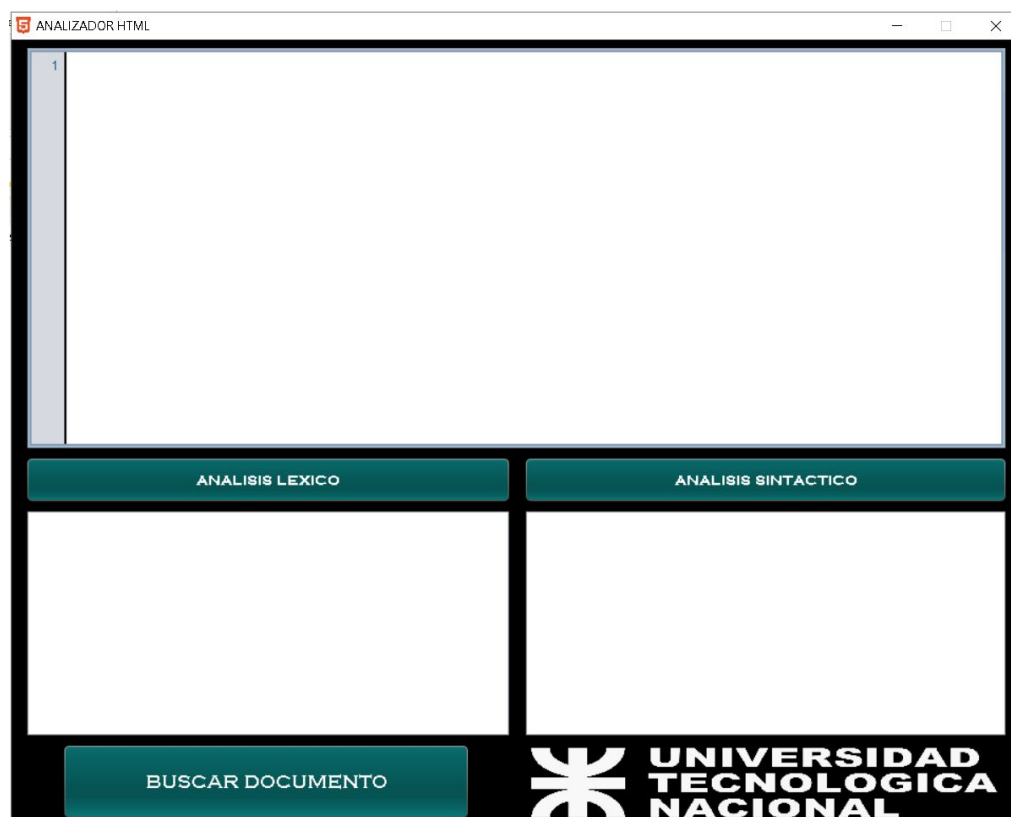
# Modo de uso

A pesar de ser bastante intuitivo y amigable el software procederemos a explicar su uso luego de haber hecho la extracción del archivo “.zip”:

1° Ingresamos en el ejecutable “AnalizadorLéxico.exe” mostrado a continuación:

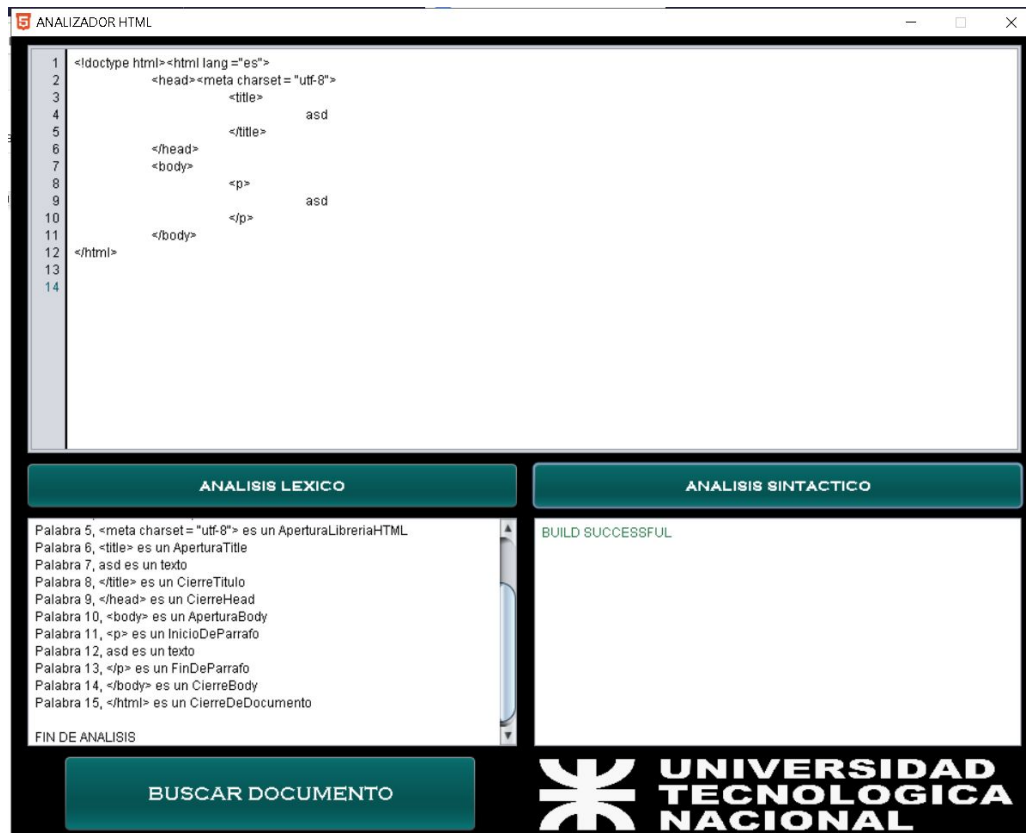


2°Una vez abierto el programa podemos optar por introducir la cadena de código manualmente o mediante un archivo “.txt” o “.html”.



Para análisis mediante digitación manual:

2.1° Para analizar mediante la introducción manual debe digitar el código en el cuadro más pequeño y luego presionar en “ANALISIS...” siendo ... el tipo, sintáctico o léxico.

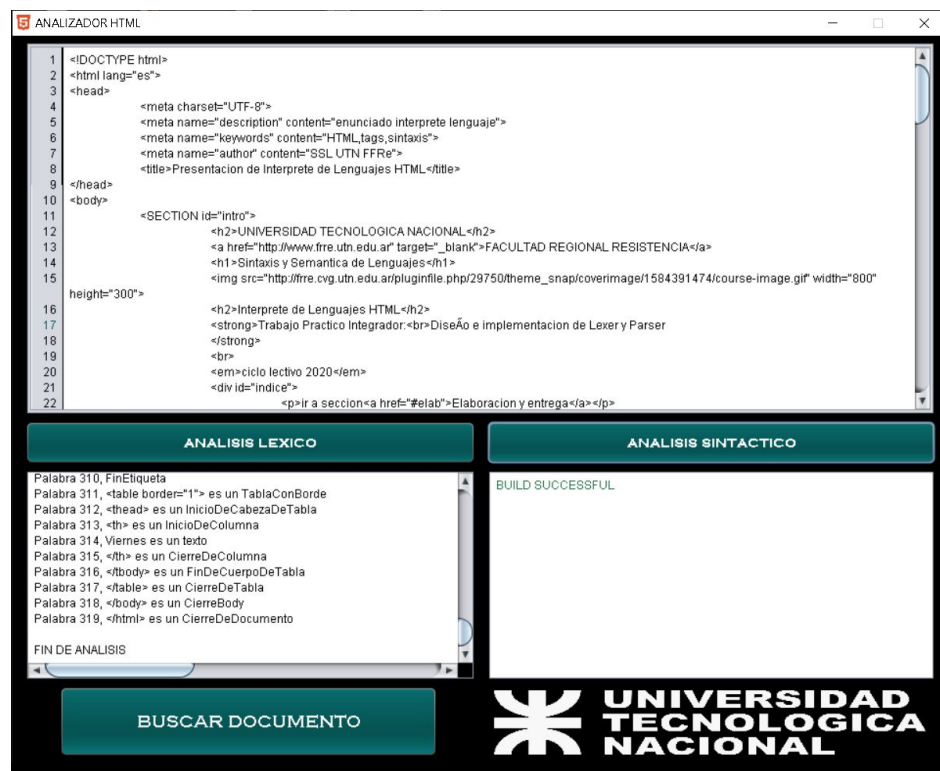


Para análisis mediante archivo:

2.2° Para analizar mediante un archivo “.txt” o “.html” debemos presionar en el botón “BUSCAR DOCUMENTO” y luego buscar el archivo que deseemos.

Anexo: Dejamos unos archivos de prueba en la carpeta “Prueba” para probarlos.

Luego seleccionamos o escribimos el nombre del archivo y le damos en abrir para que se nos analice





# CONCLUSIÓN

En el trabajo presentado se vieron todos los pasos elementales previos a la creación de un analizador léxico sintáctico y su desarrollo final, aplicando todos los conocimientos adquiridos en la materia “Sintaxis y Semántica de los lenguajes” respectivos a la temática. Con ello se logró concluir en un programa de análisis léxico-sintáctico, llevando más allá de lo aprendido en clases para la producción del mismo, sea programación en lenguajes, sea planteando interfaces de interacción usuario-máquina con múltiples formas de uso.

Concluido el trabajo se puede destacar de apreciaciones personales que sirvió para adentrarse en lo profundo de la materia llevando el saber más allá y encontrando fines prácticos para la materia.

## BIBLIOGRAFÍA Y FUENTES

- Apuntes brindados por la cátedra.
- JFlex | Analizador léxico con Java (explicación paso a paso). (17/2/19 ) Charles. Youtube. Recuperado de <https://youtu.be/5mlRn2yEe8>
- JCup y JFlex | Analizador sintáctico con Java (explicación paso a paso). (29/5/19 ) Charles. Youtube. Recuperado de <https://youtu.be/4Z6Tnit810Y>