



Universidad Tecnológica Nacional
Facultad Regional Resistencia
Departamento de Ingeniería en Sistemas de Información

Sintaxis y Semántica del Lenguaje

Trabajo Práctico Integrador: Diseño e implementación de Lexer y Parser

Ciclo lectivo: 2020

Equipo docente:

Director de Cátedra

Nombre y Apellido: Ing. Gabriela P. TOMASELLI

Docentes

Ing. Rodrigo VIGIL

Ing. Nicolas G. Tortosa

Ing. Juliana Torre

e-mail: rodrigovigil@gmail.com

e-mail: nicotortosa@gmail.com

e-mail: julitorre025@gmail.com



Contenido

1. Introducción	3
Competencias	3
Objetivo	3
2. Elaboración y entrega	3
3. Características de lenguaje de HTML	4
Introducción	4
Atributos	4
Componentes léxicos o tokens. estructura básica de un documento HTML	5
Etiquetas para estructura del documento	5
Etiquetas básicas de párrafo	6
Listas	6
Tablas	6
Enlaces	7
Imágenes	8
Atributos globales	9
· Atributos específicos	9
Texto a mostrar	10
4. Control de errores	10
5. Modos de ejecución del intérprete	10
· Modo interactivo	10
· Ejecución desde un archivo	10
6. Documentación del trabajo	11
7. Criterios de evaluación	12
Documentación y Presentación: 40 %	12
Funcionamiento del intérprete (software): 60 %	12
8. Referencias de ayuda	12
9. Archivos de ejemplo	13
Hola Mundo	13
Enunciado resumido TFI	13



1. Introducción

Competencias

El presente trabajo práctico integrador pretende desarrollar las siguientes “competencias de la asignatura”:

Especificar, proyectar y desarrollar software. Comprender conceptos y procedimientos de las gramáticas libres de contexto y gramáticas regulares para especificar la sintaxis de los lenguajes de programación.

Capacidad para conocer los fundamentos teóricos de los lenguajes de programación e intérpretes y las técnicas de procesamiento léxico y sintáctico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de un intérprete de lenguajes.

Objetivo

Se debe elaborar un intérprete de lenguaje HTML (lenguaje de marcación de hipertexto) utilizando algún lenguaje de programación o un generador de lexer y parser.

Para ello deberán construir los analizadores léxico y sintáctico. El analizador lexicográfico es un módulo que recibe una secuencia de caracteres que componen el programa a analizar y lo convierte lógicamente en una secuencia de tokens. El analizador sintáctico recibe la secuencia de tokens que le entrega el analizador lexicográfico y verifica que la secuencia pueda ser generada por la gramática del lenguaje.

Hay dos grupos de herramientas que se pueden usar para generar los analizadores:

1) Se utilizan expresiones regulares y autómatas finitos para el análisis lexicográfico y la técnica LALR para el análisis sintáctico. Ejemplos de esto son lex y yacc, que generan código C o C++, o JLex y CUP, que generan código Java. flex y bison son implementaciones libres y gratuitas de lex y yacc y se pueden conseguir en <http://www.gnu.org/>.

2) El otro utiliza la técnica LL(k) tanto para el análisis léxico como para el sintáctico, generando parsers descendentes recursivos. Ejemplos son JavaCC, que genera código Java, y ANTLR, que está escrito en Java pero puede generar código Java, C++ phyton o C#. ANTLR se puede conseguir en <http://www.antlr.org/>.

https://en.wikipedia.org/wiki/Comparison_of_parser_generators

2. Elaboración y entrega

- Modo de realización del trabajo

- o El trabajo se debe realizar de forma grupal en grupos de hasta cuatro (4) alumnos, debiendo cada integrante desarrollar una parte del trabajo durante la presentación final.

- o Cada grupo estará representado por un delegado, encargado de subir las entregas parciales y el trabajo final al campus virtual, de enviar consultas o dudas, etc. A su vez cada grupo tendrá un docente designado como tutor encargado del seguimiento del trabajo, responder inquietudes y la evaluación final.

- Presentaciones.

Se prevén tres instancias de presentación:

- . 1er entrega: Primera documentación del proyecto y gramática a generar.

Viernes 24 de Abril de 2020

- . 2da entrega: Presentación del lexer o scanner que reconozca los tokens del lenguaje.

Viernes 27 de Mayo de 2020

- . 3er entrega final: Presentación de trabajo completo. Lexer y parser, incluye la presentación del trabajo ante la clase con una exposición de no más de 20 minutos.-

Lunes 29 de Junio de 2020

- Modo de entrega final

- o Un archivo comprimido deberá ser “subido” a la tarea de la plataforma de “moodle”.

- o Dicho archivo comprimido deberá contener:



directorio *doc*. Documentación del trabajo (véase el apartado no 6)
directorio *src*. Archivos fuente del proyecto (flex, bison, archivos de C (".c", ".h"))
directorio *bin*. Archivos binarios listos para ser ejecutados. (lexer y scanner)
directorio *prueba*. Archivos de ejemplo de html con la extensión ".html"

3. Características de lenguaje de HTML

Introducción

Un documento HTML 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 se 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 >.

Ejemplo:

```
<strong>
```

La etiqueta strong sirve para indicar texto importante (se suele mostrar en negrita)

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:

```
<p>Texto normal <strong>texto negrita</strong></p>
```

Lo que indicamos en el ejemplo anterior es que un párrafo de texto contiene un elemento de tipo strong, que contiene el texto: *texto negrita*. El navegador reaccionará mostrando ese texto con un tipo de letra más fuerte. Es decir:

Texto normal **texto negrita**

Ya se ha comentado que un elemento puede contener dentro más elementos, por lo que entre la apertura y el cierre de una etiqueta puede haber más etiquetas. Una norma importante es que siempre debemos de cerrar primero las etiquetas abiertas más tarde.

Es decir, **este código no es correcto**.

```
Texto normal <strong><em>texto negrita y cursiva </strong></em>
```

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.

Ejemplo:



```
<p lang="es">
```

Componentes léxicos o tokens. *estructura básica de un documento HTML*

Etiquetas para estructura del documento

Para crear una página web usando HTML 5 la estructura básica es la siguiente:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" >
    <title></title>
  </head>
  <body>
    <p>cuerpo de la página</p>
  </body>
</html>
```

Donde el significado de los elementos es el siguiente:

- La etiqueta **DOCTYPE** indica el tipo de HTML que estamos utilizando (concretamente HTML 5).
- etiqueta **html**. Todo documento HTML está encerrado dentro de la apertura y el cierre de esa etiqueta. Marca el principio y fin del mismo (marca el elemento raíz de un documento HTML).
 - Se le debe añadir el atributo lang para indicar el idioma en el que está escrito el documento.
- **head**. Encierra las etiquetas de cabecera, las cuales proporcionan información que el navegador debe tener en cuenta para el contenido del documento aparezca de forma correcta y también contiene elementos que proporcionan información a los buscadores. se encuentra entre la etiqueta *html* y la etiqueta *body*.

Los elementos habituales que contiene son:

- etiquetas **meta**. Que sirven para indicar aspectos de funcionamiento de la página web como, cabeceras del protocolo http, indicaciones sobre cada cuánto caduca la página, palabras clave para buscar la página en los navegadores, codificación de caracteres de la página,... En este caso sólo se usa la etiqueta meta obligatoria que indica que estamos codificando el texto usando el formato utf-8 de Unicode.

```
<meta charset="UTF-8">
<meta name="description" content="enunciado interprete lenguaje">
<meta name="keywords" content="HTML,tags,sintaxis">
<meta name="author" content="SSL UTN FFRé">
<meta name="robots" content="index,nofollow" />
```

- etiqueta **title**. El título de la página, que los navegadores suelen colocar en la barra de título de la ventana en la que se muestra la página.
- cuerpo. Encerrado en la etiqueta **body**, encierra el contenido en sí de la página web. Lo que el navegador muestra por pantalla es lo que se coloca en este apartado.
- **Espacios en blanco**: El texto dentro de las páginas web no respeta los espacios en blanco ni tabuladores que coloquemos en el código, a la hora de mostrar el contenido por pantalla. Solo se considera el primer espacio en blanco, el resto se eliminan.
- **section** : Define una sección dentro del documento, puede contener los mismos tags que la sección body.



- **div** : Contenedor genérico para bloques de texto. se pueden colocar todo tipo de etiquetas (tablas, párrafos, imágenes,...).

Etiquetas básicas de párrafo

todo texto debe ir dentro de una etiqueta que indique que tipo de texto es :

- Etiqueta **p** : todo texto dentro de etiquetas **p** es considerado párrafo normal.
- Etiquetas **h1** a **h6** : texto dentro de estas etiquetas es considerado títulos. **h1** es de primer nivel , **h2** de segundo, etc.
- Etiqueta **br** : realiza un salto de línea, este elemento no tiene cierre.
- Etiqueta **hr** : similar al anterior , además de salto de línea , dibuja línea horizontal. no necesita cierre.
- Etiqueta **strong** : Texto importante, énfasis fuerte.
- Etiqueta **em** : texto enfatizado, normalmente se muestra en cursiva.
- Etiqueta **mark** : Texto resaltado. sobre el texto se aplica un fondo vivo, similar a marcador.

Listas

Las listas permiten crear párrafos agrupados y alineados mediante símbolos como viñetas o números para facilitar la lectura y organización.

- Etiqueta **ul** : engloba listas con viñetas, no ordenada.
 - Etiqueta **ol** : engloba listas numéricas, ordenadas.
 - Etiqueta **li** : ítem dentro de la lista.
- *Observación:* Es posible meter una lista dentro de otra, incluso mezclando.

```
<p>Lista de la compra</p>
```

```
<ol>
<li>
  No alcohólicas
  <ul>
    <li>Agua</li>
  </ul>
</li>
<li>
  Alcohólicas
  <ul>
    <li>Vino</li>
    <li>Cerveza</li>
  </ul>
</li>
</ol>
```

Tablas

Una tabla es un conjunto de datos presentados en forma de filas y columnas.

Cada tabla está asociado a un elemento **table**, dentro de este elemento se indican las filas mediante el elemento **tr** y dentro de cada fila se indican las celdas mediante elementos **td**.

```
<table>
<tr>
  <th>&nbsp;</th>
  <th>Jueves</th>
  <th>Viernes</th>
</tr>
```



```
<tr>
  <th>18:30</th>
  <td>Practica Sintaxis</td>
  <td>Fisica</td>
</tr>
<tr>
  <th>21:30</th>
  <td>Practica Sintaxis</td>
  <td>Mat Discreta</td>
</tr>
</table>
```

- Celdas de cabecera: Es muy habitual que las tablas muestren datos y que estos posean celdas que sirvan para describirles. Esas celdas se consideran de cabecera y se marcan con **th**.
- agrupación de filas: Hay tres elementos HTML que sirven para diferenciar las tres partes principales de una tabla, son:
 - **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

```
<table border="1" rules="groups">
  <caption>Ventas por secciones</caption>
  <thead>
    <tr>
      <td>&nbsp;</td>
      <td>Hardware</td>
      <td>Software</td>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Total</th>
      <th>25000</th>
      <th>22000</th>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <th>Enero</th>
      <td>12000</td>
      <td>15000</td>
    </tr>
    <tr>
      <th>Febrero</th>
      <td>13000</td>
      <td>9000</td>
    </tr>
  </tbody>
</table>
```

Enlaces

La etiqueta de enlaces permiten colocar un texto (u otro elemento, como una imagen) resaltado de forma especial, de modo que cuando se le hace clic, nos llevará al destino de la URL.



- Etiqueta **a** : crea enlaces a texto u otros recursos. El atributo **href** permite indicar la URL a la que se realiza el salto.
- Los enlaces pueden ser absolutos, locales relativos, o internos dentro del propio documento.

Nota: Una URL (**Uniform Resource Location**) es la dirección concreta de un recurso en Internet.

La sintaxis completa de una URL es la siguiente:

protocolo://dominio:puerto/ruta#fragmento

Se detalla el significado de todos los elementos

- El **protocolo o esquema de red**. Hace referencia al nombre de protocolo de red necesario para poder alcanzar el recurso al que hace referencia la URL. Protocolos habituales son:
 - **http://** (para recursos de la web)
 - **https://** (para recursos de la web contenidos en un servidor seguro)
 - **ftp://** (recursos contenidos en un servidor de ficheros)
 - **ftps://** (recursos contenidos en un servidor de ficheros seguro)

Tras el protocolo se indican dos puntos, tras los cuales normalmente se indican dos barras para indicar la máquina.

- **Nombre de dominio**. Nombre completo en Internet de la máquina (o la red) que posee el recurso en forma de nombre de dominio. no distinguen entre mayúsculas y minúsculas.
- **Puerto**. Opcional. Puerto por el que se debe conectar con el servidor para obtener el recurso. Si no se indica (que es lo habitual) se toma el puerto por defecto. Por ejemplo en *http* el puerto por defecto es el 80. Si queremos usar uno en particular se indica tras el servidor poniendo dos puntos y el puerto.
- **Ruta**. Opcional. Indica el recorrido dentro de la máquina remota que hay que hacer a través de los directorios para llegar al recurso que queremos. Se pone después del servidor. Ejemplos:
 - */index.html* Accede a la página index.html situada en el directorio raíz.
 - */imagenes/paisajes/foto001.jpg* Accede a la imagen foto001.jpg dentro del directorio paisajes dentro, a su vez, del directorio imagenes.
- **Localizador interno**. Opcional. Va detrás del símbolo # y sirve para indicar un identificador (que puede ser más o menos complejos) que permita localizar o seleccionar una parte concreta del recurso destino de la URL.
- Los únicos caracteres permitidos en URL son letras, números, guión medio, guión bajo y punto. además de los caracteres reservados: #, /, :

Imágenes

Las imágenes se colocan mediante la etiqueta **img**. El atributo **src** indica la URL (relativa o absoluta) a la imagen. Ejemplo:



img es una etiqueta sin cierre, la imagen se coloca directamente en la posición de esta etiqueta. Se la considera como texto al maquetarla. De modo que aparece entre el texto si ponemos la etiqueta entre el texto.

Atributos globales

- **CLASS:** El nombre de una clase, o una lista de nombres de clases separados por espacios. Este atributo es utilizado en conjunto con hojas de estilo e indica al navegador qué clases se supone asignarán atributos presentacionales al elemento.
 - o **Ejemplo de uso:** `<p class="referencias"> lorem ipsum </p>`
- **ID:** Un identificador para el elemento. Este identificador debe ser único en el documento y puede ser utilizado para referirse al elemento desde vínculos, scripts, definiciones de estilo u otros atributos.
 - o **Ejemplo de uso:** `<h1 id="clave"> lorem ipsum </h1>`

Atributos específicos

1. ELEMENTO ``:

- a. **TYPE:** El tipo de símbolos a usar en las viñetas. Existen cinco posibles valores:
 - 1: enteros (1, 2, 3...).
 - a: letras minúsculas (a, b, c...).
 - A: letras mayúsculas (A, B, C...).
 - i: números romanos en minúsculas (i, ii, iii...).
 - I: números romanos en mayúsculas (I, II, III...).

2. ELEMENTO ``:

- a. **VALUE:** Un entero indicando el valor ordinal del ítem en la lista.

3. ELEMENTO `<a>`:

- a. **HREF:** La dirección (representada por un URI) del recurso enlazado. Este atributo es el corazón del vínculo, e indica a los navegadores a dónde deben dirigirse cuando el vínculo es activado.
- b. **TARGET:** Un valor que especifica dónde debería abrirse el recurso enlazado.
 - i. `_blank`: el vínculo se abrirá en una nueva ventana.
 - ii. `_parent`: el vínculo se abrirá en el contexto padre inmediato.
 - iii. `_self`: el vínculo se abrirá en el mismo contexto que contiene al vínculo.
 - iv. `_top`: el vínculo se abrirá en el más alto contexto de navegación (el contexto más exterior que contiene al vínculo).

4. ELEMENTO ``:

- a. **ALT:** Una línea de texto que provee contenido equivalente para aquellas situaciones en que las imágenes no pueden ser vistas (usuarios con discapacidades



visuales, navegadores sin soporte, configuraciones en que las imágenes están deshabilitadas, etc.).

- b. **SRC:** El [URI](#) del recurso de imagen. Este atributo es obligatorio ya que apunta al recurso que será, de hecho, incrustado en el documento.
- c. **WIDTH:** Un número de píxeles que indica el ancho que la imagen tomará cuando sea incrustada en el documento.
- d. **HEIGHT:** Un número de píxeles que indica la altura que la imagen tomará cuando sea incrustada en el documento.

Texto a mostrar

Características

Estarán compuestos por una serie de letras mayúsculas, minúsculas, dígitos y caracteres especiales. Todos compatibles con el estándar UTF-8.

No se distinguirá entre mayúsculas ni minúsculas.

4. Control de errores

El intérprete deberá controlar toda clase de errores:

• Léxicos:

- tags mal escritos.
- Utilización de símbolos no permitidos.
- Etc.

• Sintácticos:

- Apertura y cierre de tags correctos
- Falta de Tags obligatorios
- Anidamiento de tags.
- Tags con atributos incompatibles o falta de atributos obligatorios
- Etc.

- Observación : Se valorará la utilización de “reglas de producción de control de errores” que no generen conflictos.

. Semánticos

o atributos u tags mal agrupados

• De ejecución

o archivo de entrada inexistente o con una extensión incorrecta.
o Etc.

5. Modos de ejecución del intérprete

El intérprete se podrá ejecutar de dos formas diferentes:

• Modo interactivo

o Se ejecutarán las instrucciones tecleadas desde un terminal de texto
ilh.exe
> ...

o Se utilizará el carácter de fin de archivo para terminar la ejecución: **Control + D**

• Ejecución desde un archivo

o Se interpretarán las etiquetas a partir de un archivo pasado como argumento desde la línea de comandos

o El archivo deberá tener la extensión “.html”
ilh.exe ejemplo.html



6. Documentación del trabajo

Se deberá elaborar un documento de texto con las siguientes características:

- Portada
 - o Título del trabajo desarrollado
 - o Nombre y apellidos de las personas que forman el grupo
 - o Nombre de la asignatura
 - o Nombre de la carrera
 - o Primer cuatrimestre
 - o Curso académico: 2020
 - o Universidad y regional
 - o Lugar y fecha
- Índice
 - o Las páginas deberán estar numeradas.
- Introducción
 - o Breve descripción del trabajo realizado y de las partes del documento.
 - o Descripción de cómo se implementó la solución.
 - o Información y requerimientos de software para ejecutar y recompilar el tp (versiones de compiladores, herramientas, plataforma, etc).
- Información sobre el intérprete
 - o Se corresponde con el apartado no 3 de este documento
 - # Componentes léxicos
 - # Sentencias
 - o *Observación: Si se ha ampliado el lenguaje del intérprete entonces se deberá indicar en este apartado.*
- Gramática
 - o Descripción de cada regla de la gramática libre de contexto a utilizar.
 - # Símbolos de la gramática
 - # Símbolos terminales (componentes léxicos)
 - # Símbolos no terminales
 - # Reglas de producción de la gramática
 - # Se valorará la inclusión de gráficos explicativos.
- Análisis léxico
 - o Descripción del archivo del lexer para definir y reconocer los componentes léxicos.
- Análisis sintáctico:
 - o Descripción del archivo del scanner utilizado para definir la gramática libre de contexto
- Funciones auxiliares
 - o Se deben indicar y describir las funciones auxiliares que se hayan codificado.
- Modo de obtención del intérprete
 - o Nombre y descripción de cada archivo utilizado
- Modo de ejecución del intérprete
 - o Interactiva
 - o A partir de un archivo
- Ejemplos
 - o Al menos se deben proporcionar dos ejemplos.
 - o Se valorará la cantidad, originalidad y complejidad de los ejemplos propuestos.
 - o También se puede incluir el ejemplo propuesto por el profesor y/o otros grupos.
- Conclusiones:
 - o Reflexión sobre el trabajo realizado.
 - o Puntos fuertes y puntos débiles del intérprete desarrollado.



- Bibliografía o referencias web.
- Anexos

o Se podrían incluir aquellos anexos que se consideren oportunos para mejorar la calidad de la documentación

7. Criterios de evaluación

Documentación y Presentación: 40 %

Se tendrá en cuenta lo indicado en el apartado no 6.

- o El código elaborado deberá estar documentado.
- o Se valorará la inclusión de gráficos o figuras.
- o Se valorará la cantidad, originalidad y complejidad de los ejemplos propuestos.
- o También se valorará:
 - # la acentuación,
 - # la corrección ortográfica
 - # la calidad y claridad de la redacción.
- o Conocimiento que cada integrante del grupo aporte durante la exposición final del trabajo.

Funcionamiento del intérprete (software): 60 %

- o La gramática diseñada no podrá generar conflictos.
 - # Esta condición es imprescindible para aprobar el trabajo de prácticas.
- o El intérprete deberá:
 - # funcionar correctamente tanto de forma interactiva como ejecutando las instrucciones de los archivos de ejemplo.
 - # en particular, deberá ejecutar correctamente el ejemplo propuesto por el grupo y dos ejemplos más: uno al azar de los presentados por otros grupos y otro ejemplo elaborado por la cátedra.
- o Se valorará
 - # la completitud del intérprete.
 - # La calidad en el diseño del lenguaje y la gramática.
 - # El control de errores.
 - # La ampliación del lenguaje del intérprete.

Observación:

Además, se valorará la asistencia a clase de prácticas y la resolución de dificultades encontradas durante la elaboración del trabajo.

8. Referencias de ayuda

Direcciones en Internet que nos ayudan a utilizar los distintos elementos HTML5 (incluso nos dicen su compatibilidad). Las más importantes son :

- w3schools. Contiene información al día sobre HTML, CSS, JavaScript y otras tecnologías web. Está en inglés pero es sencilla y bien estructurada. La ayuda de las etiquetas de HTML está en la dirección: <http://www.w3schools.com/tags/>
- Mozilla Developer Network. La página oficial de los desarrolladores del navegador Mozilla Firefox. Contiene información de todos los elementos HTML a medida que aparecen, aunque no estén del todo estandarizados, indicando su compatibilidad con los navegadores y si ya es estándar o no. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
- World Wide Web Consortium (W3C). Referencia oficial de la documentación de todos los estándares web. Se estructura de manera muy formal, pero tiene la ventaja de que esta documentación es la que se refiere al estándar. Sobre HTML podemos acudir a la dirección: <https://www.w3.org/TR/html5/>



9. Archivos de ejemplo

Hola Mundo

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" >
    <title>HOLA MUNDO</title>
  </head>
  <body>
    <SECTION id="principal">
      <p>hola mundo 2020</p>
    </section>
  </body>
</html>
```

Enunciado resumido TFI

El archivo html esta disponible en el recurso compartido Drive.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" >
    <meta name="description" content="enunciado interprete lenguaje">
    <meta name="keywords" content="HTML,tags,sintaxis">
    <meta name="author" content="SSL UTN FFRé">
    <title>Presentacion de Interprete de Lenguajes HTML</title>
  </head>
  <body>
    <SECTION id="intro">
      <h2>UNIVERSIDAD TECNOLÓGICA NACIONAL</h2><h2> <a href="http://www.frre.utn.edu.ar"
      target="_blank">FACULTAD REGIONAL RESISTENCIA</a></H2>
      <h1>Sintaxis y Semántica de Lenguajes</h1>
      
      <h2>Interprete de Lenguajes HTML</h2>
      <strong>Trabajo Práctico Integrador:<br>Diseño e implementación de Lexer y Parser</strong>
      <br>
      <em> ciclo lectivo 2020</em>
      <div id="indice">
        <p>ir a seccion <a href="#elab">Elaboracion y entrega</a></p>
```



```
<p>ir a seccion <a href="#tags">Tags</a></p>
<p>ir a seccion <a href="#horarios">Horarios</a></p>

</div>
</section>
<h3>Objetivo</h3>
<p>elaborar un intérprete de lenguaje HTML (lenguaje de marcación de hipertexto) utilizando
algún lenguaje de programación o un generador de lexer y parser.</p>
<hr>
<section id="entrega">
  <h3 id="elab">Elaboracion y Entrega</h3>
  <ul>
    <li><strong>Modo de realización del trabajo</strong>
      <p>El trabajo se debe realizar de forma grupal en grupos de hasta
<strong>cuatro (4)</strong> alumnos, donde cada integrante deberá presentar una parte del
trabajo durante la presentación final.</p>
      <p>Cada grupo estará representado por un delegado, encargado de subir
las entregas parciales y el trabajo final al campus virtual, enviará consultas o dudas,etc.</p>
    </li>
    <li><strong>Presentaciones</strong>
      <p>Se preveen tres instacias de presentación:</p>
      <ol type="I">
        <li>1er entrega: Primera documentación del proyecto y gramática a
generar. <mark>Viernes 24 de Abril de 2020</mark></li>
        <li>2da entrega: Presentación del lexer o scanner que reconozca los
tokens del lenguaje. <mark>Viernes 27 de Mayo de 2020</mark> </li>
        <li>3er entrega final: Presentacion de trabajo completo. Lexer y
parser, incluye la presentación del trabajo ante la clase con una exposición de no más de 20
minutos. <mark><strong>Lunes 29 de Junio de 2020</strong></mark></li>
      </ol>
    </li>
  </ul>
</section>
<hr>
<section id="tags">
  <h3 id="elab">Principales Etiquetas</h3>
  <ul>
    <li><strong>Estructura documento</strong>
      <p>Describen la estructura general del documento:</p>
      <ol type="a">
        <li>DOCTYPE</li>
        <li>html </li>
        <li>
          head </li>
        <li>title</li>
        <li>body</li>
        <li>section</li>
        <li>div</li>
      </ol>
    </li>
    <li><strong>Etiquetas de parrafo</strong>
      <p>Describen texto:</p>
      <ol type="a">
        <li>p</li>
      </ol>
    </li>
  </ul>
</section>
```



```
<li>h1 a h2 </li>
<li>                                br    </li>
<li>  hr</li>
      <li> strong</li>
<li>em</li>
<li>mark</li>
</ol>
</li>
<br>
<li><strong>Listas</strong>
  <p> Las listas permiten crear párrafos agrupados y alineados, como este
:)</p>
  <ol type="a">
    <li>ul</li>
    <li>ol </li>
    <li>  li                                </li>
  </ol>
</li>
<br>
<li><strong>Tablas</strong>
  <p>Una tabla es un conjunto de datos presentados en forma de filas y
columnas.</p>
  <ol type="a">
    <li>table</li>
    <li>tr</li>
    <li>                                td    </li>
    <li>  th</li>
      <li> thead</li>
    <li>tfoot</li>
    <li>tbody</li>
  </ol>
</li>
<br>
<li><strong>Enlaces</strong>
  <p>La etiqueta de enlaces permiten colocar un texto (u otro elemento, como
una imagen) resaltado de forma especial, de modo que cuando se le hace clic, nos llevará al destino
de la URL.</p>
  <p>La sintaxis completa de una URL es la siguiente: <MARK>
protocolo://dominio:puerto/ruta#fragmento</mark></p>
  </li>
<li><strong>Imagenes</strong>
  <p>Las imágenes se colocan mediante la etiqueta img. El atributo src indica
la URL (relativa o absoluta) a la imagen</p>
  </li>
</ul>

</section>
<hr>
<section id="horarios">
<table border="1">
  <h3>Horarios de Sintaxis</h3>
  <thead>
```



```
<tr>
<td>2.1</td>
<th>Practica</th>
<th>Teoria</th>
</tr>
</thead>
```

```
<tbody>
<tr>
<th>Jueves</th>
<td>18:30</td>
<td></td>
</tr>
<tr>
<th>Viernes</th>
<td>21:30</td>
<td></td>
</tr>
<tr>
<th>Miercoles</th>
<td></td>
<td>14:15</td>
</tr>
</tbody>
```

```
</table>
</section>
```

```
</body>
```

```
</html>
```