Universidad de Granada

Doble Grado en Ingeniería Informática y Matemáticas

Modelos de Computación

Práctica Lex/Flex

Autor:
Jesús Muñoz Velasco
Daniel Morán Sánchez

Curso 2024-2025

Índice

1. Introducción

El objetivo principal de esta práctica es la de resolver una traducción simple de un código en HTML a pdf utilizando LaTeXcomo lenguaje para realizar dicha transformación. Para ello se han seleccionado una serie de estructuras de HTML para transformarlas a su correspondiente código en LaTeX.

En esta implementación, para no complicar demasiado la traducción no se han tenido en cuenta el uso de estilos ni de bloques de código (ya que estos no serían implementables).

2. Estructuras

Para el desarrollo de este punto se ha usado la herramienta de Regex Vis para la creación de los diagramas.

Las principales estructuras de HTML que se han procesado con el programa realizado han sido los siguientes:

2.1. Bold/Black

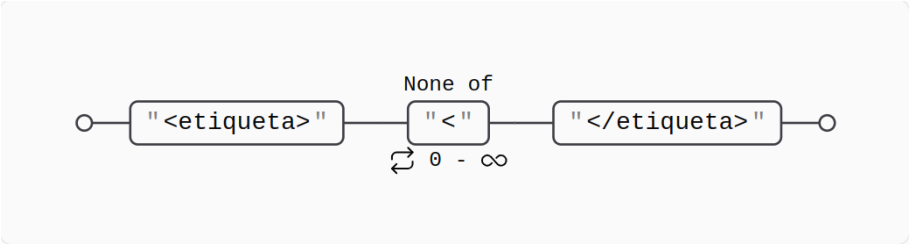
Estas etiquetas HTML ponen en negrita el texto que encierran. Un ejemplo de su uso en un documento HTML sería el siguiente:

```
Esto es una palabra en <strong>negrita</strong> y esta <b>otra</b>.
```

Las expresiones regulares que se han utilizado para la detección de estas etiquetas han sido:

```
    BOLD    \<strong\>[^\<]*\<\/strong\>
    BLACK    \<b\>[^\<]*\<\/b\>
```

De esta forma se reconocerían las etiquetas principales y el texto que encierran, el cual no puede contener el caracter `'<'`. Seguiría un diagrama similar al siguiente:



Donde `'etiqueta'` se cambiará por el nombre de la etiqueta en cada caso. Esta estructura se usará para el resto de etiquetas con un formato similar (ya que es propio de HTML) y por eso no se volverá a desarrollar.

Para su traducción a LaTeXse ha utilizado el entorno `textbf` el cual tiene el mismo resultado en este lenguaje. El ejemplo anterior traducido sería el siguiente:

```
Esto es una palabra en \textbf{negrita} y esta \textbf{otra}.
```

2.2. Italic

Esta etiqueta hace referencia al texto en cursiva. Sigue la misma estructura que antes. Un ejemplo de su uso en un documento HTML sería el siguiente:

```
Esto es una palabra en <em>cursiva</em>.
```

La expresión regular utilizada para su reconocimiento ha sido la siguiente:

```
    ITALIC    \<em\>[^\<]*\<\/em\>
```

Su traducción a LaTeXse ha implementado mediante el entorno `textit`. El ejemplo anterior traducido sería el siguiente:

```
Esto es una palabra en \texit{cursiva}.
```

2.3. Underlined

Esta etiqueta se usa para el texto subrayado. Un ejemplo de su uso en HTML sería el siguiente:

```
Esto es una palabra <u>subrayada</u>.
```

De nuevo, la expresión regular con la estructura de etiquetas sería la siguiente:

```
    UNDERLINED    \<u\>[^\<]*\<\/u\>
```

La implementación del subrayado se ha realizado en LaTeXmediante `underline`. El ejemplo anterior quedaría de la siguiente forma:

```
Esto es una palabra \underline{subrayada}.
```

2.4. Section, Subsection y Subsubsection

Estas etiquetas se utilizan para delimitar secciones, permitiendo usar una jerarquía. En HTML se implementan con las etiquetas `h1,...,h6`. En LaTeXsolo se permiten 3 niveles de jerarquía por lo que solo se han implementado `h1`, `h2` y `h3`. Con el resto de etiquetas se podría hacer algún tipo de gestión pero como dependería de la implementación deseada no se ha realizado ninguna. Un ejemplo del uso de estas etiquetas en un documento HTML sería el siguiente:

```
<h1>Esto es un título</h1>
<h2>Esto es un subtítulo</h2>
<h3>Esto es un subsubtítulo</h3>
```

Las expresiones regulares utilizadas para su detección han sido:

```
    SECTION    \<h1\>[^\<]*\<\/h1\>
    SUBSECTION    \<h2\>[^\<]*\<\/h2\>
    SUBSUBSECTION    \<h3\>[^\<]*\<\/h3\>
```

Si se quisiera agrupar `h3,...,h6`, se podría modificar esta última expresión a

```
    SUBSUBSECTION    \<h[3-6]\>[^\<]*\<\/h[3-6]\>
```

Aunque no garantizaría la corrección del código, es decir, no tendría por qué coincidir la etiqueta de apertura con la de cierre (podría admitir cadenas como por ejemplo `<h4>subsubtitulo</h5>`).

La traducción de estas etiquetas a LaTeXse ha realizado mediante `section`, `subsection` y `subsubsection` respectivamente. El ejemplo anterior quedaría de la siguiente forma:

```
\section{Esto es un título}
\subsection{Esto es un subtítulo}
\subsubsection{Esto es un subsubtítulo}
```

2.5. Paragraph

Esta estructura se utiliza para delimitar párrafos en HTML. Un ejemplo de su uso en HTML sería el siguiente:

```
<p>Esto es un párrafo.</p>
```

La expresión regular utilizada ha sido la siguiente:

```
    PARAGRAPH    \<p\>[^\<]*\<\/p\>
```

Su traducción se ha realizado simplemente añadiendo el texto sin ningún entorno ya que LaTeXes un lenguaje de procesamiento de texto y todo el texto sin etiqueta se interpreta como párrafo. De esta forma el ejemplo anterior traducido quedaría

```
Esto es un párrafo.
```

2.6. Image

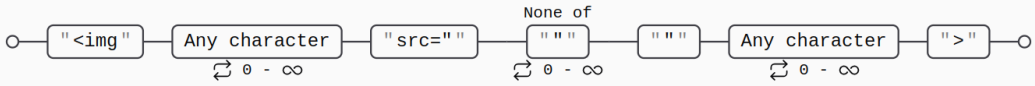
En HTML se utiliza la etiqueta `img` para insertar imágenes en el documento. En este caso sí se ha admitido algún parámetro adicional como el tamaño de la imagen (se ha considerado opcional). Un ejemplo de su uso en HTML sería el siguiente:

```
<img src="image.png" width="300" height="200">
```

Esto indica que la dirección de la imagen es `image.png` y que tiene unas dimensiones de `300 x 200 px`. Los parámetros relativos al tamaño son opcionales (pueden incluirse ambos, uno o ninguno y seguirá funcionando correctamente). De este modo, la expresión regular utilizada ha sido la siguiente:

```
    IMG     \<img.*src=\"[^\"]*\".*\>
```

Lo cual admite las cadenas que empiecen por `<img`, que sigan por cualquier cadena de caracteres (`.*`), después tienen que contener la cadena `src="` seguida de cualquier cadena que no incluya el caracter `"` y que se cierre con dicho caracter. Después vuelve a admitir cualquier cadena hasta que se cierre con `>`. Podemos verlo gráficamente con el siguiente diagrama:



Su traducción a LaTeXse ha realizado mediante `includegraphics` el cual admite parámetros opcionales, facilitando su adaptación. El ejemplo anterior quedaría de la siguiente forma:

```
\includegraphics[width=300px, height=200px]{image.png}
```

2.7. Link

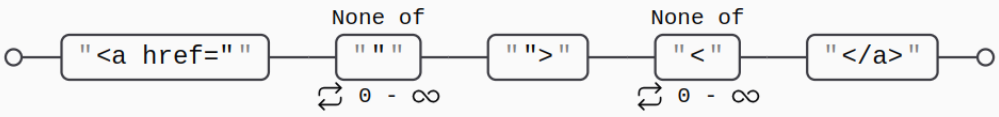
La estructura con etiqueta `a` permite en HTML insertar hiperenlaces a otras páginas. Su sintáxis incluye un parámetro `href` donde se indica la dirección de la página enlazada. Un ejemplo de su uso en un documento HTML sería el siguiente:

```
<a href="https://ejemplo.com">Esto es un enlace</a>
```

Sabiendo esto, la expresión regular empleada ha sido la siguiente:

```
    LINK    \<a\ href=\"[^\"]*\"\>[^\<]*\<\/a\>
```

Como se puede ver sigue una estructura mixta entre la de etiquetas (con la etiqueta `a` para cierre y apertura) y la de parámetros vista para imágenes. Su estructura se podría representar de la siguiente forma.



Para su implementación en LaTeXse ha empleado el comando `href` el cual admite tanto el enlace como el texto que se mostrará. El ejemplo anterior adaptado se verá de la siguiente forma:

```
\href{https://ejemplo.com"}{Esto es un enlace}
```

2.8. Newline

En HTML existe una etiqueta especial, `br` (break) que inserta un salto de línea. No requiere de una etiqueta de cierre y se podría usar de la siguiente forma.

```
Esto es una línea con salto al final.<br>
```

Su expresión regular es bastante simple ya que consiste en la detección literal de esta cadena.

```
    NEWLINE \<br\>
```

Su adaptación a LaTeXse ha implementado con el doble escape, `\\` el cual tiene un efecto similar. El ejemplo anterior quedaría de la siguiente forma:

```
Esto es una línea con salto al final.\\
```

2.9. Unordered List

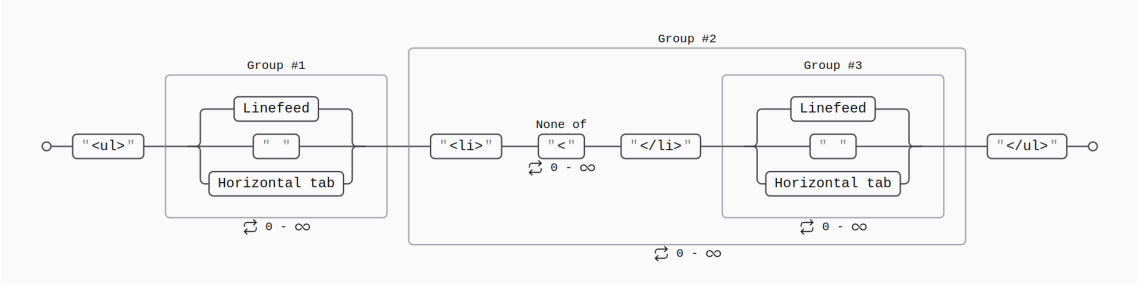
Esta estructura se utiliza en HTML para crear listas no ordenadas, es decir, cuyos elementos se encuentrar indentados por viñetas (no números). Su sintáxis en HTML consiste en una etiquetación (`ul`) para delimitar la lista y otra (`li`) para delimitar cada elemento de la lista. Una lista puede contener tantos elementos como se requiera. Un ejemplo de su uso en HTML sería el siguiente:

```
<ul>
    <li>Esto</li>
    <li>Es una lista</li>
    <li>No ordenada</li>
</ul>
```

Su expresión regular ha resultado en:

```
    ULIST    \<ul\>(\n|\ |\t)*(\<li\>[^\<]*\<\/li\>(\n|\ |\t)*)*\<\/ul\>
```

Lo cual tiene en cuenta la estructura que se ha mencionado anteriormente además de considerar los espacios en blanco, tabulaciones y saltos de línea (`\n|\ |\t`) que pueda haber entre las etiquetas. El diagrama resultante de esta expresión regular sería el siguiente:



Su traducción a LaTeXse ha implementado utilizando el entorno `itemize`, en el que los elementos están delimitados por el comando `item`. De esta forma, el ejemplo anterior quedaría de la siguiente forma:

```
\begin{itemize}
    \item Esto
    \item Es una lista
    \item No ordenada
\end{itemize}
```

2.10. Ordered List

Esta estructura es muy similar a la anterior y se utiliza en HTML para crear listas ordenadas, es decir, cuyos elementos están numerados. Su sintáxis es idéntica a la anterior a excepción de que la etiqueta que delimita la lista será `ol` en lugar de `ul`. Un ejemplo de su uso en HTML sería el siguiente:

```
<ol>
    <li>Esto</li>
    <li>Es una lista</li>
    <li>Ordenada</li>
</ol>
```

Aálogamente, su expresión regular será

```
    OLIST    \<ol\>(\n|\ |\t)*(\<li\>[^\<]*\<\/li\>(\n|\ |\t)*)*\<\/ol\>
```

Que no se desarrollará por ser igual a la anterior pero con el cambio ya explicado. Su implementación a LaTeXse ha realizado con el entorno `enumerate` el cual funciona exactamente igual que `itemize` pero numerando los items. Por tanto, el ejemplo anterior quedaría de la siguiente forma:

```
\begin{enumerate}
    \item Esto
    \item Es una lista
    \item Ordenada
\end{enumerate}
```

2.11. Table

La creación de tablas en HTML sigue una estructura algo compleja y se realiza a varios niveles. En primer lugar, una etiqueta `table` delimita por completo a la tabla. Cada fila de la tabla estará delimitada por la etiqueta `tr` (table row). Por último, cada celda dentro de cada fila podrá estar delimitada por las etiquetas `th` (cabecera) o `td`. De esta forma, un ejemplo de tabla con 3 columnas y 2 filas (siendo la primera la cabecera) que se podría hacer en HTML sería la siguiente:

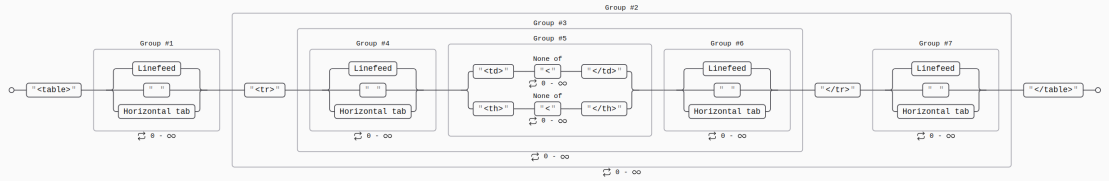
```
<table>
    <tr>
        <th>Esto</th>
        <th>Es la</th>
```

```
        <th>Cabecera</th>
    </tr>
    <tr>
        <td>Esto</td>
        <td>Es el resto</td>
        <td>de la tabla</td>
    </tr>
</table>
```

Su expresión regular utiliza la misma estrategia usada para las listas de gestión de espacios en blanco y por ello resulta un poco compleja de leer aunque con el diagrama que la representa se entenderá mejor. Se ha considerado la siguiente expresión (sin el salto de línea, que se ha puesto para su correcta visualización en este documento).

```
    TABLE    \<table\>(\n|\ |\t)*(\<tr\>((\n|\ |\t)*(\<td\>[^\<]*\<\/td>|\<th\>
             [^\<]*\<\/th>)(\n|\ |\t)*)*\<\/tr\>(\n|\ |\t)*)*<\/table\>
```

El diagrama que resulta es el siguiente (se recomienda hacer un poco de zoom para verlo mejor):



Su traducción a LaTeXse ha realizado mediante el entorno `tabular` en el cual se ha implementado un código en `c++` que permite tablas dinámicas, es decir, de cualquier numero de columnas (ya que en LaTeXse debe saber a priori). Esto se hace contando primero el número de elementos de la primera fila y asumiendo que el resto de la tabla tendrá dicho número de columnas. En LaTeXcada fila se representará como una línea acabada por salto de línea (\\) y las celdas se separarán horizontalmente con el separador `&`. Además, para las filas de encabezado (con etiqueta `th`) se ha insertado el texto en un entorno `strong`. Además se han añadido todos los bordes de la tabla (esto se podría cambiar pero se ha decidido que es lo que mejor representa una tabla). Para ello, entre cada fila y al principio y al final de la tabla se inserta el comando `hline` que añade los separadores horizontales. Los separadores verticales se especifican en las dimensiones de la tabla .De esta forma, la tabla anterior quedaría de la siguiente forma:

```
\begin{tabular}{|c|c|c|}
    \hline
    \textbf{Esto} & \textbf{Es la} & \textbf{Cabecera} \\
    \hline
    Esto & Es el resto & de la tabla \\
    \hline
\end{tabular}
```

2.12. Head

En HTML un documento se estructura en varias partes (estructura general del documento). Una de ellas es el `head` en el cual se especifican configuraciones varias como palabras claves para la búsqueda, idioma, enlaces a hojas de estilo, etc. Las configuraciones que nos interesan por poder ser pasadas a LaTeXson únicamente el título (con etiqueta `title`) y el autor (con etiqueta `author`). Esta sección en su totalidad está delimitada por la etiqueta `head`. De esta forma, un ejemplo de head de un documento HTML sería la siguiente:

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <meta name="author" content="Jesús y Dani">
    <style>
        table, th, td{
            border: 1px solid black;
            border-collapse: collapse;
        };
    </style>
</head>
```

Como puede contener cualquier etiqueta en su interior, la expresión regular utilizada ha sido mucho más general que las utilizadas hasta ahora resultando en

```
    \<head\>((.|\n)*?)\<\/head\>
```

Lo cual admite cualquier caracter incluyendo saltos de línea. De esta sección solo nos interesa detectar el autor y el título, lo cual se ha implementado en `c++` sin utilizar las utilidades de `flex` (por lo admitir realizar búsquedas recursivas y por la necesidad de eliminar el resto del contenido por ser inútil y causar fallos en la compilación del archivo LaTeXde salida). Por tanto, la adaptación a LaTeXdel ejemplo anterior sería únicamente lo siguiente:

```
\title{Document}
\author{Jesús y Dani}
```

2.13. Body

Al igual que la sección anterior, otra parte básica de la estructura general de un HTML es `body`. En él se incluyen los elementos que se visualizarán al procesar un archivo HTML. Está encerrado por la etiqueta `body` y en su interior se incluyen el resto de estructuras que se han mencionado hasta el momento. Es por ello que se hace de forma ligeramente distinta a los demás. Un ejemplo de uso de esta estructura en un documento HTML sería el siguiente:

```
<body>
    Esto es el cuerpo del documento.
</body>
```

Su expresión regular se ha trabajado de forma distinta ya que se va a emplear una sustitución literal. Se ha propuesto las siguientes expresiones:

```
    BODY_BEGIN    \<body\>
    BODY_END    \<\/body\>
```

De esta forma se buscan literalmente las etiquetas sin interferir lo que encierran y se han traducido a LaTeXutilizando la apertura y clausura del entorno `document` respectivamente. Así, el ejemplo anterior quedaría de la siguiente forma:

```
\begin{document}
    Esto es el cuerpo del documento.
\end{document}
```

Cabe destacar que esta estrategia se podría haber usado en otros casos y hubiese permitido la anidación de varias estructuras pero se ha decidido no hacerlo así para poder buscar expresiones regulares más complejas y que no fuesen literales.

2.14. Ignore

Por la naturaleza de los archivos HTML se incluyen campos que deben ser eliminados para la correcta compilación del archivo de salida en formato `tex`. Es por esto que se ha decidido ignorar (no pasar al documento final) las estructuras que se adecúen a la siguiente expresión regular:

```
    IGNORE  (\<!DOCTYPE\ html\>|\<html\ lang=\"[^\"]*\"\>|\<\/html>)
```

Con esto se consigue pasar finalmente a un documento LaTeXcon una sintáxis correcta. Estos campos que se ignoran son

- •) El indicador de que es un archivo HTML (`<!DOCTYPE html>`).
- •) La apertura de la etiqueta `html` que incluye `head` y `body` y que por tanto no tiene por qué pasar al documento final de ninguna forma especial, sino procesando como ya se ha mencionado las estructuras que contiene.
- •) El cierre de la etiqueta `html`.

3. Ejemplo de uso

Para probar el programa se ha realizado un documento HTML genérico llamado `fichero.html` que incluye todas las estructuras que se han trabajado en esta práctica para ver su correcto funcionamiento. El documento HTML contiene lo siguiente:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <meta name="author" content="Jesús y Dani">
```

```
    <style>
        table, th, td{
            border: 1px solid black;
            border-collapse: collapse;
        };
    </style>
</head>
<body>

<h1>Esto es un título</h1>
<h2>Esto es un subtítulo</h2>
<h3>Esto es un subsubtítulo</h3>

Aquí comienza un texto que puede contener palabras en <strong>negrita</strong>,
en <em>cursiva</em> y <u>subrayadas</u>.
Puede contener también <a href="https://ejemplo.com">enlaces</a>.
Además, se pueden empezar párrafos como el siguiente:

<p>Esto es un párrafo de un documento HTML que termina con un salto de línea.
</p><br>
Se pueden ver también imágenes como la siguiente:<br>

<img src="image.png" width="300" height="200"><br>

Además se incluye una lista no ordenada:
<ul>
    <li>Esto</li>
    <li>Es una lista</li>
    <li>No ordenada</li>
</ul>

Y una lista ordenada
<ol>
    <li>Esto</li>
    <li>Es una lista</li>
    <li>Ordenada</li>
</ol>

Se añade también una tabla con 3 columnas y cabecera

<table>
    <tr>
        <th>Esto</th>
        <th>Es la</th>
        <th>Cabecera</th>
    </tr>
    <tr>
        <td>Esto</td>
        <td>Es el resto</td>
        <td>de la tabla</td>
    </tr>
</table>

Se añade también una tabla con 2 columnas y sin cabecera

<table>
```

```
    <tr>
        <td>Esto es</td>
        <td>otra tabla</td>
    </tr>
    <tr>
        <td>Esto</td>
        <td>Es el resto</td>
    </tr>
</table>


</body>
</html>
```

Al visualizar este código HTML obtendríamos lo siguiente:

Esto es un título

Esto es un subtítulo

Esto es un subsubtítulo

Aquí comienza un texto que puede contener palabras en **negrita**, en *cursiva* y subrayadas. Puede contener también enlaces. Además, se pueden empezar párrafos como el siguiente:

Esto es un párrafo de un documento HTML que termina con un salto de línea.

Se pueden ver también imágenes como la siguiente:

Además se incluye una lista no ordenada:

- Esto
- Es una lista
- No ordenada

Y una lista ordenada

1. Esto
2. Es una lista
3. Ordenada

Se añade también una tabla con 3 columnas y cabecera

Esto	Es la	Cabecera
Esto	Es el resto	de la tabla

Se añade también una tabla con 2 columnas y sin cabecera

Esto es	otra tabla
Esto	Es el resto

Una vez realizado, podemos compilar el programa `html-to-latex.l` de la siguiente forma:

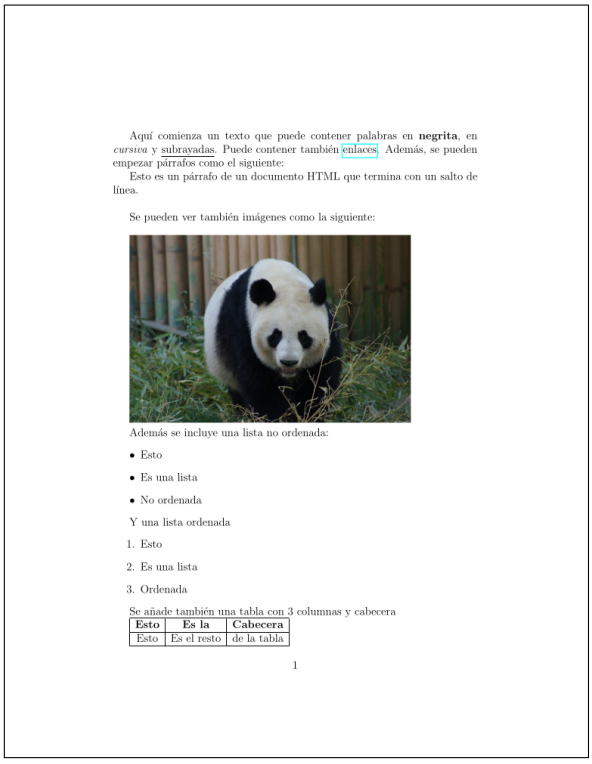
```
$ flex++ -o ./html-to-latex.yy.cc ./html-to-latex.l
$ g++ -o ./html-to-latex ./html-to-latex.yy.cc
$ ./html-to-latex ./fichero.html > ./salida.tex
$ pdflatex ./salida.tex
```

De esta forma se compila el archivo `flex` y se ejecuta sobre `fichero.html` dirigiendo la salida a un documento `salida.tex` que se compila para crear el pdf que se llamará `salida.pdf`. El documento `salida.tex` contendrá lo siguiente:

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\usepackage{hyperref}


\title{Document}
\author{Jesús y Dani}

\begin{document}


\section{Esto es un título}
\subsection{Esto es un subtítulo}
\subsubsection{Esto es un subsubtítulo}

Aquí comienza un texto que puede contener palabras en \textbf{negrita},
en \textit{cursiva} y \underline{subrayadas}.
Puede contener también \href{https://ejemplo.com"}{enlaces}.
Además, se pueden empezar párrafos como el siguiente:

Esto es un párrafo de un documento HTML que termina con un salto de línea.\\

Se pueden ver también imágenes como la siguiente:\\


\includegraphics[width=300px, height=200px]{image.png}

Además se incluye una lista no ordenada:
\begin{itemize}
    \item Esto
    \item Es una lista
    \item No ordenada
\end{itemize}


Y una lista ordenada
\begin{enumerate}
    \item Esto
    \item Es una lista
    \item Ordenada
\end{enumerate}


Se añade también una tabla con 3 columnas y cabecera

\begin{tabular}{|c|c|c|}
    \hline
```

```
    \textbf{Esto} & \textbf{Es la} & \textbf{Cabecera} \\
    \hline
    Esto & Es el resto & de la tabla \\
    \hline
\end{tabular}


Se añade también una tabla con 2 columnas y sin cabecera

\begin{tabular}{|c|c|}
    \hline
    Esto es & otra tabla \\
    \hline
    Esto & Es el resto \\
    \hline
\end{tabular}



\end{document}
```

Y el pdf resultante se vería de la siguiente forma:



Aquí comienza un texto que puede contener palabras en **negrita**, en *cursiva* y subrayadas. Puede contener también enlaces. Además, se pueden empezar párrafos como el siguiente:

Esto es un párrafo de un documento HTML que termina con un salto de línea.

Se pueden ver también imágenes como la siguiente:

Además se incluye una lista no ordenada:

- Esto
- Es una lista
- No ordenada

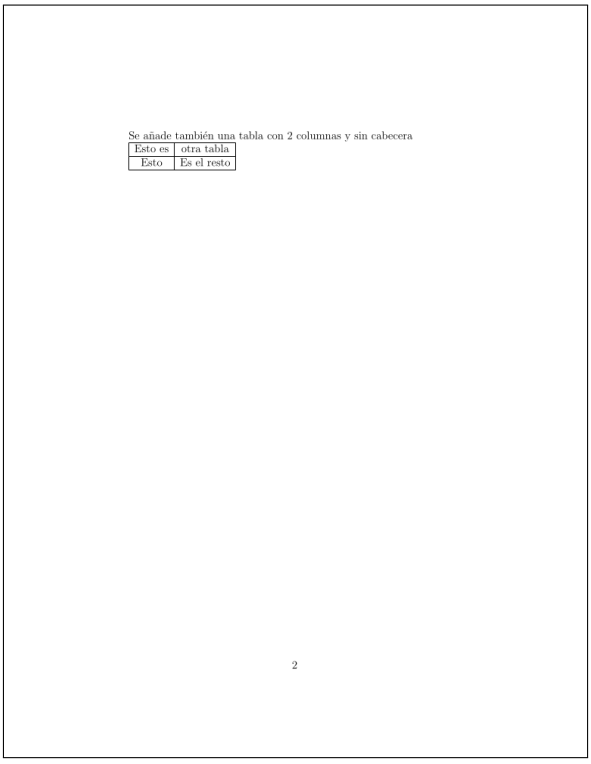
Y una lista ordenada

1. Esto
2. Es una lista
3. Ordenada

Se añade también una tabla con 3 columnas y cabecera

Esto	Es la	Cabecera
Esto	Es el resto	de la tabla

1



Se añade también una tabla con 2 columnas y sin cabecera

Esto es	otra tabla
Esto	Es el resto

2