



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA  
Escuela de Ingeniería Informática



Grado en Ingeniería Informática

<Computación>

Trabajo Final de Título

# LatexToHTMLOnLine

Conversor en línea de LATEX a HTML.

Autor: **Christian Brito Ramos**

Tutor: **Luís Álvarez León**

Tutor: **Nelson Monzón López**

<Julio, 2019>

## Resumen

En este Trabajo Fin de Grado se ha desarrollado un conversor en línea de formato LATEX a HTML. En este sentido, el usuario podrá elegir entre diferentes estilos de salida que se han configurado utilizando ficheros de estilo CSS. El conversor, implementado en C++, convierte a formato HTML los comandos básicos de LATEX incluyendo imágenes y vídeos. Se ha diseñado una implementación en línea (ver <http://www.ctim.es/LatexToHTML>) de este conversor usando las funcionalidades del servidor de DEMO's de IPOL (*Image Processing On Line*). De esta manera, cualquier usuario puede hacer uso de esta herramienta sin necesidad de descargar o instalar localmente el software asociado.

## Abstract

In this work an online LATEX to HTML converter is developed and implemented. The user can choose between different output styles that have been configured using CSS style files. The converter, implemented in C ++, converts the basic LATEX commands to HTML format including images and videos. An online implementation of this converter has been designed using the facilities of the IPOL (Image Processing On Line) DEMO's server (see <http://www.ctim.es/LatexToHTML>). In this way, any user can use this tool without downloading or installing the associated software locally.

# AGRADECIMIENTOS

---

Me gustaría expresar mi agradecimiento a todas aquellas personas que me han ayudado en esta larga travesía.

En primer lugar, a mis tutores de TFT D. Luis Álvarez León y D. Nelson Monzón López que me han guiado y aconsejado en esta última etapa.

A mis amigos y compañeros por brindarme su apoyo cuando más complicado era el camino.

Finalmente, a mi familia, por todos los ánimos que recibo a diario, las esperanzas que me dan y la fuerza que me transmiten. Agradecer en especial a mis padres, que me han ayudado a ser mejor cada día y levantarme cuando no podía.

Gracias.

*Christian Brito Ramos*

# ÍNDICE GENERAL

---

Resumen	III
Abstract	IV
Agradecimientos	V
Índice General	VI
Índice Figuras	VII
Índice Tablas	X
Capítulo 1: Introducción.	1
1.1 Objetivos Iniciales	2
1.2 Competencias Específicas Cubiertas.	2
1.3 Aportaciones.	3
Capítulo 2: Estado Actual del Tema.	4
2.1 Herramientas de conversión.	4
2.1.1 MathML (Mathematical Markup Language).	4
2.1.2 MathJax (Beautiful math in all browsers).	6
2.1.3 Online LATEX equation editor.	6
2.1.4 MathToWeb.	7
2.1.5 T <sub>T</sub> H: The TEX to HTML translator.	7
2.1.6 Tex4ht.	7
2.1.7 LaTeX2HTML.	7
2.1.8 LaTeXXML.	8
2.1.9 PlasTex.	8
2.1.10 Make4ht.	8
2.2 Alternativas a LatexToHTMLOnLine.	8
2.2.1 Alternativa 1: LaTeXXML.	9
2.2.2 Alternativa 2: The TEX to HTML Translator.	10
2.3 Evaluación.	11
Capítulo 3: Planificación del Proyecto.	12
3.1 Plan de Trabajo.	12
3.2 Metodología.	13
3.3 Presupuesto.	13
3.3.1 Costes de Personal.	14
3.3.2 Costes de Hardware.	14
3.3.3 Costes de Software.	14

3.3.4 Otros Costes.	15
3.3.5 Costes Totales.	15
Capítulo 4: Desarrollo de LatexToHTMLOnLine.	16
4.1 Estudio Previo.	16
4.1.1 Introducción a LATEX.	16
4.1.2 Estudio de Herramientas Existentes de Conversión LATEX/HTML.	17
4.1.3 Image Processing On Line.	18
4.2 Desarrollo del Conversor.	21
4.2.1 Comandos convertidos.	21
Cabecera LATEX.	21
Título Fecha y Autor.	21
Abstract (Resumen).	22
Índice.	23
Secciones del Documento.	26
Párrafos.	26
Negrita e Itálica.	27
Caracteres ASCII Extendidos.	27
Listas.	27
Tablas.	28
Imágenes y Vídeos.	30
Fórmulas Matemáticas.	31
Bibliografía.	33
Funciones LATEX no Implementadas.	33
4.2.2 Implementación en C++.	34
4.2.3 Demo OnLine.	36
Implementación de la DEMO.	37
4.3 Creación de Estilos.	41
4.3.1 Cascading Style Sheets (CSS).	41
4.4 Validación y Prueba.	42
4.4.1 Validación y Prueba “off-line”.	42
4.4.2 Validación “on-line”.	42
Capítulo 5: Conclusiones y Trabajo Futuro.	43
5.1 Conclusiones.	43
5.2 Trabajo Futuro.	44
Bibliografía.	45
ANEXO 1.	46

# ÍNDICE DE FIGURAS

---

<b>Figura 2.1:</b> Fórmula general para la obtención de raíces.	4
<b>Figura 2.2:</b> MathJax.	6
<b>Figura 2.3:</b> The TEX to HTML LOGO.	7
<b>Figura 2.4:</b> Librería digital de funciones matemáticas NIST.	9
<b>Figura 2.5:</b> Ejemplo de ejecución de The LATEX to HTML.	10
<b>Figura 4.1:</b> Ejemplo básico compilable en LATEX.	16
<b>Figura 4.2:</b> Resultado de ejecución de código LATEX.	17
<b>Figura 4.3:</b> Logo de la revista IPOL. Revista enfocada en la investigación reproducible.	18
<b>Figura 4.4:</b> Diagrama con los módulos principales de IPOL.	19
<b>Figura 4.5:</b> Ejemplo de la pantalla principal del “Panel de Control” en IPOL.	20
<b>Figura 4.6:</b> Código LATEX equivalente a título, fecha y autores.	21
<b>Figura 4.7:</b> Etiqueta HTML título página web.	22
<b>Figura 4.8:</b> Traducción LATEX/HTML de título, fecha y autor.	22
<b>Figura 4.9:</b> Ejemplo del comando “\abstract” en LATEX.	23
<b>Figura 4.10:</b> Ejemplo del comando “\abstract” en HTML.	23
<b>Figura 4.11:</b> “\tableofcontents” en código LATEX.	24
<b>Figura 4.12:</b> Resultado de “\tableofcontents” en código LATEX.	25
<b>Figura 4.13:</b> Resultado de “\tableofcontents” en código HTML.	26
<b>Figura 4.14:</b> Sustitución de las secciones.	26
<b>Figura 4.15:</b> Lista no numerada en LATEX.	27
<b>Figura 4.16:</b> Lista no numerada en HTML.	28
<b>Figura 4.17:</b> Tabla 5x3 en LATEX.	28
<b>Figura 4.18:</b> Resultado de una tabla en LATEX.	29
<b>Figura 4.19:</b> Resultado de una tabla HTML con CSS.	29
<b>Figura 4.20:</b> Resultado de una tabla en HTML.	30
<b>Figura 4.21:</b> Código LATEX para inyección de imagen.	31
<b>Figura 4.22:</b> Código HTML para inyección de imagen.	31
<b>Figura 4.23:</b> Código HTML para inyección de vídeo.	31
<b>Figura 4.24:</b> Librería MathJAX en HTML.	32
<b>Figura 4.25:</b> Código de fórmula matemática en LATEX.	32
<b>Figura 4.26:</b> Resultado de fórmula matemática en LATEX.	32
<b>Figura 4.27:</b> Declaración de zona bibliográfica en LATEX.	33
<b>Figura 4.28:</b> Declaración de zona bibliográfica en HTML.	33

<b>Figura 4.29:</b> Template HTML en español.	35
<b>Figura 4.30:</b> Página principal de la DEMO.	36
<b>Figura 4.31:</b> Zona de resultados dentro de la pantalla principal una vez terminada la ejecución.	37
<b>Figura 4.32:</b> Ejemplo de página HTML resultante tras usar el conversor online.	37
<b>Figura 4.33:</b> Página principal del “Panel de control” de la demo.	38
<b>Figura 4.34:</b> DDL de configuración de la DEMO.	39
<b>Figura 4.35:</b> Código JavaScript asociado a la demo online.	40
<b>Figura 4.36:</b> Script Python asociado a la DEMO online.	40

# ÍNDICE DE TABLAS

---

<b>Tabla 3.1:</b> Tabla que refleja las distintas fases del desarrollo del TFT.	13
<b>Tabla 3.2:</b> Tabla que refleja los costes mensuales del personal.	14
<b>Tabla 3.3:</b> Tabla que refleja los costes hardware del proyecto.	14
<b>Tabla 3.4:</b> Tabla que refleja los costes software del proyecto.	14
<b>Tabla 3.5:</b> Tabla que refleja otros costes mensuales del proyecto.	15
<b>Tabla 3.6:</b> Tabla que refleja los costes totales del proyecto durante el período de un año.	15

# INTRODUCCIÓN

---

LATEX es un lenguaje de edición de textos cuyo uso está muy extendido en trabajos científicos, especialmente en los de carácter técnico. Se utiliza para escribir artículos de revistas, comunicaciones a congresos, tesis doctorales y todo tipo de memorias. Por este motivo, muchos investigadores están familiarizados con este lenguaje y su uso está enormemente extendido en el ámbito científico y académico.

Relacionado con esto, la aparición de Internet ha facilitado enormemente la difusión de este tipo de publicaciones de una forma atractiva más allá de límites geográficos. En este sentido, consideramos relevante la elaboración de un sistema informático que permita convertir documentación redactada en LATEX a un lenguaje que un entorno web pueda interpretar. Gracias a esto únicamente sería necesario redactar el texto de la publicación en LATEX y se simplificaría su publicación en entornos web.

En el entorno científico, la revista *Image Processing On Line*<sup>1</sup> (IPOL) promueve un entorno de difusión de ciencia abierta y reproducible en el campo del procesamiento de señales (imagen, vídeo, audio, 3D). Su objetivo es presentar trabajos matemáticos cuyos resultados, además de aportar valor científico, puedan ser confirmados con exactitud. La interfaz web de demostración de IPOL se ha desarrollado con HTML5, CSS3 y JQuery y permite a los usuarios ejecutar algoritmos implementados en C++ u otros lenguajes en una interfaz ergonómica.

En este sentido, el objetivo de este Trabajo Fin de Título ha sido desarrollar un conversor en línea que traduce formato LATEX a código HTML convirtiendo tanto sus comandos básicos como insertando imágenes y vídeos que pertenezcan al documento original. El sistema en línea utiliza internamente un programa implementado en lenguaje C++ y el entorno de desarrollo DEMOS de IPOL que veremos en siguientes apartados. Además de la conversión, existen diferentes opciones que permiten la edición del aspecto del fichero de salida HTML. Esto se consigue mediante la selección de un fichero CSS que el usuario podrá adoptar a su elección entre diversas opciones de estilo.

Para el desarrollo del TFT, se ha realizado, en primer lugar, un estudio del editor de texto LATEX y sus comandos básicos. Seguidamente se estudia el problema de la conversión de código LATEX a HTML y cómo enfocar su desarrollo e implementación. Se diseñarán diferentes estilos de CSS para los resultados del conversor y finalmente se implementará el conversor en el entorno de IPOL.

Como resultado del desarrollo del proyecto, se espera tener una aplicación web sencilla que cumpla la funcionalidad de conversión y sea fácil de usar.

---

<sup>1</sup> <http://www.ipol.im/>

## 1.1. OBJETIVOS INICIALES.

A continuación, se exponen los objetivos a llevar a cabo en el desarrollo de LatexToHTMLOnLine.

1. Familiarización con el editor de texto LATEX y sus comandos básicos. Dada la gran variedad de comandos LATEX existentes se limitará el estudio a los comandos básicos necesarios en la creación de páginas DEMOS incluyendo la inserción de imágenes y vídeos. Se usará el compilador de LATEX *MikTeX* en un entorno Windows.
2. Desarrollo de ejemplo en LATEX a modo de prueba para el conversor y que incluya en el mismo documento todas las directrices LATEX que se implementarán en el conversor.
3. Se estudiará el problema de la conversión del código LATEX a lenguaje HTML y se implementará el conversor en lenguaje C++.
4. Desarrollo de diferentes hojas de estilo CSS para las páginas WEB's resultantes del conversor, incluyendo la selección del idioma (español o inglés).
5. Implementación del conversor en el entorno online de DEMOS *Image Processing On Line* [IPOL].

## 1.2. COMPETENCIAS ESPECÍFICAS CUBIERTAS.

### TEG01

*“Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sintetizen e integren las competencias adquiridas en las enseñanzas.”*

El presente Trabajo Final de Título junto con la memoria del mismo, su posterior presentación y defensa ante un tribunal, cubre la competencia anteriormente citada.

### ISO3

*“Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.”*

Esta competencia queda cubierta en el apartado [4.2 Desarrollo del conversor](#) donde se da solución al problema de la integración de las fórmulas LATEX en HTML con la tecnología [MathJax].

### 1.3. APORTACIONES.

En el contexto científico y académico se necesita, muy habitualmente, publicar en Internet la información de proyectos de investigación, tesis doctorales y, en general, todo tipo de memorias que previamente han sido editadas mediante el procesador de texto LATEX.

En este contexto, la aplicación web LatexToHTMLOnLine<sup>1</sup> desarrollada durante el presente Trabajo Fin de Grado, ofrece al usuario la posibilidad de elegir, de manera muy sencilla, entre diferentes estilos de salida, diferenciándose así de los demás conversores de LATEX a HTML. Su disponibilidad on-line y la interfaz del conversor facilitan un manejo sencillo a los usuarios potenciales.

Este proyecto no tiene un fin lucrativo sino que su principal objetivo es la aportación en un contexto científico/educativo, de una solución novedosa, versátil y sencilla de usar que permita al potencial usuario publicar de forma atractiva en internet documentos escritos en LATEX y por tanto no se considera, de momento, su comercialización.

---

<sup>1</sup> <http://www.ctim.es/LatexToHTML>

# ESTADO ACTUAL DEL TEMA

---

La visualización en línea de textos escritos en LATEX requiere su conversión manual a un lenguaje que los navegadores web puedan interpretar. Entonces, el objetivo principal de este trabajo es desarrollar un prototipo online que permita la conversión automática de un documento completo escrito en LATEX al formato HTML, incluyendo además diferentes estilos posibles de formateado para la visualización del texto en el navegador web. Por tanto, para la realización de este proyecto es necesario analizar las herramientas existentes en la literatura relacionadas con la conversión en línea de LATEX a HTML.

## 2.1. HERRAMIENTAS DE CONVERSIÓN.

En este capítulo se presenta una descripción de las herramientas existentes en la literatura relacionadas con la conversión en línea de LATEX a HTML.

### 2.1.1. MathML (Mathematical Markup Language)

MathML (ver la referencia [MathML]) es un lenguaje de marcado basado en XML cuyo objetivo es integrar fórmulas matemáticas en documentos web. MathML 1 apareció como recomendación del W3C en abril de 1998 siendo el primer lenguaje XML recomendado por el W3C.

Actualmente es parte de HTML5 y requiere que los navegadores implementen la interpretación de la sintaxis del lenguaje, que como se verá a continuación es bastante compleja. Actualmente FIREFOX incluye MathML pero otros navegadores conocidos como Microsoft EDGE o Google Chrome no. El hecho de que no funcione en todos los navegadores es una limitación importante. Por otro lado, la sintaxis de escritura de fórmulas matemáticas es completamente distinta a LATEX y mucho más compleja. Para ilustrar estas diferencias pondremos a continuación un sencillo ejemplo de fórmula matemática escrita utilizando ambos formatos

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

**Figura 2.1:** Fórmula general para la obtención de raíces.

Una fórmula matemática sencilla como puede ser la **Figura 2.1**, se escribe en LATEX como:

```
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
```

Sin embargo, la misma fórmula matemática en MathML se escribe de la siguiente forma:

```
<math>
  <mrow>
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mrow>
          <mo>-</mo>
          <mi>b</mi>
        </mrow>
        <mo>&PlusMinus;</mo>
        <msqrt>
          <mrow>
            <msup>
              <mi>b</mi>
              <mn>2</mn>
            </msup>
            <mo>-</mo>
            <mrow>
              <mn>4</mn>
              <mo>&InvisibleTimes;</mo>
              <mi>a</mi>
              <mo>&InvisibleTimes;</mo>
              <mi>c</mi>
            </mrow>
          </mrow>
        </msqrt>
      </mrow>
      <mrow>
        <mn>2</mn>
        <mo>&InvisibleTimes;</mo>
        <mi>a</mi>
      </mrow>
    </mfrac>
  </mrow>
</math>
```

Como puede apreciarse, la sintaxis de MathML es mucho más compleja que la de LATEX.

### 2.1.2. MathJax (Beautiful math in all browsers)



Figura 2.2: MathJax

Dos de las sociedades matemáticas más importantes de los Estados Unidos: la “American Mathematical Society” (AMS) y la “Society for Industrial and Applied Mathematics (SIAM)” han promovido el proyecto MathJax (ver la referencia [MathJax]). MathJax es una biblioteca JavaScript de código abierto que permite visualizar en la web fórmulas escritas directamente en la sintaxis de LATEX. El proyecto MathJax nació en 2009 como sucesor de una biblioteca JavaScript anterior, llamada jsMath.

Para utilizar MathJax en un documento web simplemente hay que añadir una directiva al principio del fichero HTML para que se genere una visualización de las fórmulas escritas directamente en formato LATEX. Concretamente se añade en la cabecera del documento HTML la directiva:

```
<script type="text/javascript" async  
src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/MathJax.js?con  
fig=TeX-MML-AM_CHTML" async>  
</script>
```

De esta manera, las fórmulas matemáticas escritas en LATEX se visualizan directamente en el navegador. Esta es la solución que se ha adoptado en este trabajo para visualizar en un navegador WEB las fórmulas matemáticas escritas en LATEX por ser una solución sencilla de implementar y que funciona además en cualquier navegador.

### 2.1.3. Online LATEX equation editor

Esta herramienta (ver [OLED]) permite convertir en línea fórmulas matemáticas sueltas escritas en formato LATEX. Permite exportar la salida en formato de imagen gif. La exportación a formato de imagen fue durante muchos años la manera habitual en la que las fórmulas matemáticas se incorporaban a los archivos web. Esta estrategia tiene la importante limitación que hay que definir “a priori” la resolución de la imagen y cuando se hace “zoom” sobre la página web, la fórmula se ve pixelada. Esta herramienta no está diseñada para convertir textos completos escritos en LATEX.

#### 2.1.4. MathToWeb

Esta herramienta (ver [MathToWeb]) permite convertir en línea fórmulas matemáticas sueltas escritas en formato LATEX al formato MathML mencionado anteriormente. Esta herramienta no está diseñada para convertir textos completos escritos en LATEX.

#### 2.1.5. The TEX To HTML Translator



Figura 2.3: The TEX to HTML LOGO.

La mayoría de los conversores del formato LATEX a HTML no trabajan en línea, es decir para su uso se requiere la instalación de un programa y su ejecución, normalmente usando la línea de comandos en LINUX. T<sub>T</sub>H (ver [T<sub>T</sub>H ]) es un conversor bastante conocido que funciona de esta manera. T<sub>T</sub>H produce documentos web compactos y de visualización rápida debido a que traduce las ecuaciones en lugar de convertirlas en imágenes.

#### 2.1.6. Tex4ht

Tex4ht (ver [TeX4ht]) es un conversor de LATEX a formatos SGML tales como HTML, MathML, OpenDocument y DocBook. Tex4ht no trabaja en línea y a diferencia de muchos conversores, TeX4ht no analiza la fuente LATEX, si no que lo utiliza para producir un archivo DVI que procesará a posteriori. Analiza el archivo DVI generado cuando se compila LATEX con las macros tex4hts incluidas. La configuración y la llamada a TeX4ht pueden ser bastante complicadas.

#### 2.1.7. LaTeX2HTML

LaTeX2HTML (ver [LaTeX2HTML]) es un paquete de script Perl que permite convertir código LATEX a HTML utilizando mapas de bits (imágenes). No trabaja en línea.

### **2.1.8. LaTeXXML**

LaTeXXML (ver [LTXML]) es un programa Perl que puede analizar la mayoría del código TeX, incluidas las definiciones de macros complicadas. Dispone de una versión on-line [OLXML] con la que se puede traducir LATEX a HTML5, pero con carencias en relación a imágenes, vídeos y estilos de salida. No está en las distribuciones estándar de TeX.

### **2.1.9. PlasTeX**

PlasTeX (ver [plasTeX]) es un marco de procesamiento de documentos LaTeX basado en Python. Puede generar múltiples formatos de salida (por ejemplo, HTML, DocBook, tBook, etc.). No funciona en línea.

### **2.1.10. Make4ht**

Make4ht (ver [Make4ht]), convierte LATEX a XML. Se ejecuta en línea de comando y no funciona en línea.

## **2.2. ALTERNATIVAS A LatexToHTMLOnline**

Como hemos podido comprobar en el apartado anterior, existen numerosas herramientas que traducen del lenguaje fuente LATEX a un lenguaje objeto determinado generalmente HTML.

Si es cierto que ninguna de las herramientas planteadas en el apartado anterior realiza una traducción completa del LATEX, como son [MathML], [MathJax], [OLED] y [MathToWeb], que solo traducen fórmulas matemáticas escritas en LATEX a un formato de imagen o a un lenguaje web. Otras herramientas como [TeX4ht], [LaTeXXML], [plasTeX] o [LaTeX2HTML], realizan las conversiones de un fichero LATEX completo, pero con ciertas deficiencias en cuestión de fórmulas, imágenes y vídeos.

Un pequeño subconjunto de los programas del apartado anterior 2.1 donde se describe el estado actual del tema, se pueden asemejar a la propuesta de este trabajo final de título.

### 2.2.1. Alternativa 1: LaTeXXML

Como se ha comentado en el capítulo anterior, el LATEX es ampliamente usado en trabajos de investigación, académicos y de otros intereses divulgativos. Un equipo del NIST (National Institute of Standards and Technology) intentaron desarrollar una librería digital de funciones matemáticas, pero identificaron la necesidad de traducir todos los documentos escritos en LATEX a algún lenguaje que pudiera interpretar un navegador web en este caso XML. Obteniendo dicha traducción, tendrían la ventaja de manipular, organizar y construir un sitio web con toda la información, obteniendo así una auténtica librería digital.

Según este equipo de investigadores, una auténtica librería digital debe centrarse en la semántica del material, por lo que tuvieron que convertir el material matemático en contenido y presentación MathML.

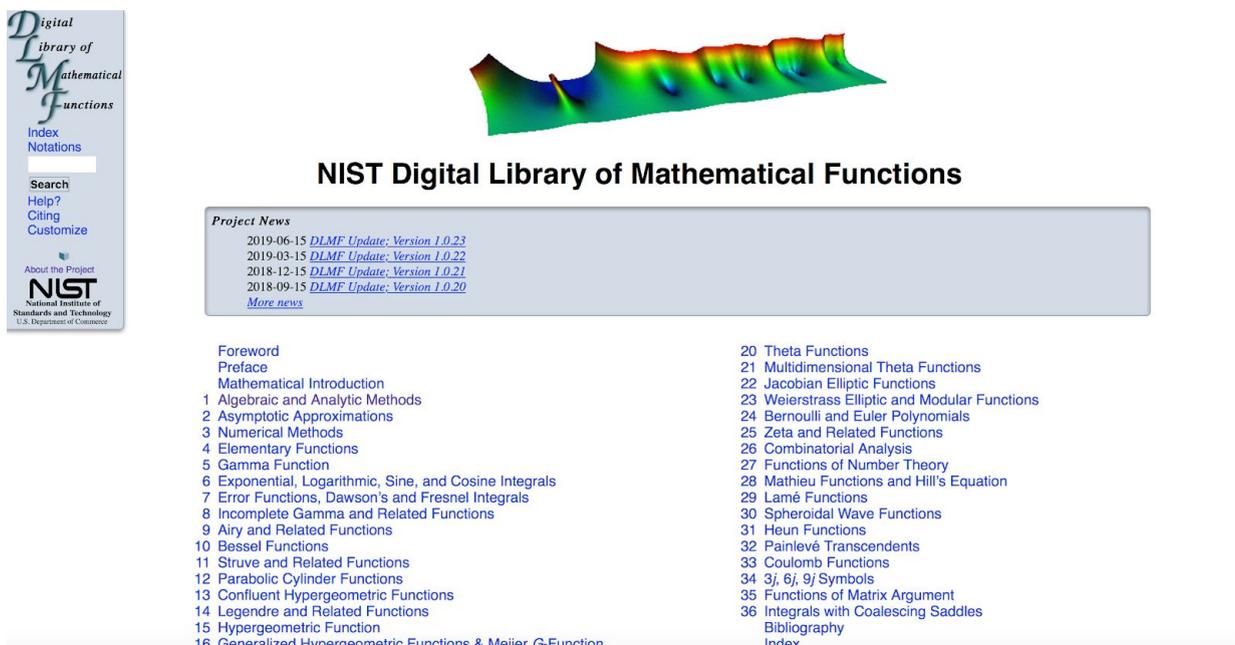


Figura 2.4: Librería digital de funciones matemáticas NIST.

Al no encontrar ninguna herramienta que lo permitiera en ese momento, decidieron crear el proyecto LaTeXXML de forma interna y así poder llevar a cabo NIST Digital Library of Mathematical Functions [NDLMF]. Ver **Figura 2.4**.

A pesar de ser XML su principal apuesta de traducción, LaTeXXML dispone de un postprocesador que convierte el XML resultante en lenguajes como HTML o XHTML, asemejándose así al objetivo de este trabajo final de título LatexToHTMLOnline.

Además, disponen de una interfaz online DEMO donde se puede ejecutar una pequeña versión de su algoritmo y obtener de forma inmediata el resultado por pantalla.

Por último, LaTeXML advierte a los potenciales usuarios de que el proyecto no está terminado, pues existen brechas debido a la gran cantidad de paquetes que ofrece LATEX.

### 2.2.2. Alternativa 2: The TEX to HTML translator

The Tex to HTML translator [T<sub>T</sub>H] es la herramienta más semejante. Permite una conversión completa del lenguaje LATEX a HTML, con algunas restricciones de comandos que no contempla. Traduce de manera literal las fórmulas y no existe conversión de ningún dato a imagen de forma que no se pierda información.

Se puede ver un ejemplo de formato de salida de [ejemplo](#) de ejecución de [T<sub>T</sub>H] en la siguiente figura.

HEAD
PREVIOUS

## Chapter 2

### Motion of Charged Particles in Fields

Plasmas are complicated because motions of electrons and ions are determined by the electric and magnetic fields but *also change* the fields by the currents they carry. For now we shall ignore the second part of the problem and assume that *Fields are Prescribed*. Even so, calculating the motion of a charged particle can be quite hard. Equation of motion:

$$\underbrace{m \frac{dv}{dt}}_{\text{Rate of change of momentum}} = q \underbrace{(\mathbf{E} + \mathbf{v} \wedge \mathbf{B})}_{\text{Lorentz Force}}$$

(2.1)

Have to solve this differential equation, to get position  $\mathbf{r}$  and velocity ( $\mathbf{v} = \dot{\mathbf{r}}$ ) given  $\mathbf{E}(\mathbf{r},t), \mathbf{B}(\mathbf{r},t)$ . Approach: Start simple, gradually generalize.

#### 2.1 Uniform B field, $\mathbf{E} = \mathbf{0}$ .

$$m \dot{\mathbf{v}} = q \mathbf{v} \wedge \mathbf{B}$$

(2.2)

##### 2.1.1 Qualitatively

in the plane perpendicular to  $\mathbf{B}$ :



**Figura 2.5:** Ejemplo de ejecución de The LATEX to HTML.

Como se ha comentado anteriormente, este proyecto no dispone de una interfaz on-line, principal diferenciación con respecto al presente trabajo final de título LatexToHTMLOnline.

## 2.3. EVALUACIÓN

A partir de las diferencias identificadas entre los diferentes conversores presentes en este apartado, y tras tener la oportunidad de evaluar aquellos proyectos más parecidos al presentado en este trabajo final de título LatexToHTMLOnLine, se destacan las siguientes observaciones:

La mayoría de los conversores de LATEX a HTML actualmente son gratis y muchos de ellos se encuentran sin mantenimiento.

El motivo es porque no hay un nicho de mercado específico que supla las necesidades de todo usuario que vaya a convertir código LATEX en HTML. Como ya he comentado con anterioridad, LATEX dispone de numerosas librerías dificultando así, crear un conversor que abarque todas las necesidades posibles de cualquier usuario. Es por ello, que hay entidades como la de [NIST] que crean su propio conversor para su tipo determinado de documentos y así ahorrarse posibles problemas de conversión empleando un conversor ajeno. Y es que sale más barato pagar a un equipo que cree un conversor de estas características, que pagar a un equipo para que transcriba a mano todos los documentos de un lenguaje a otro.

Entrando a diferenciar LatexToHTMLOnLine con los demás conversores, no se ha encontrado en la literatura ningún conversor que ofrezca las posibilidades que ofrece LatexToHTMLOnLine. Las novedades con respecto a los demás conversores radican en que el conversor únicamente se encuentra on-line con una interfaz sencilla y amigable, posibilita el resultado con plantillas en dos idiomas y por último, dispone de una amplia selección de estilos de tal manera que el usuario se pueda descargar el código html enlazado a un fichero CSS on-line que corresponde con el aspecto que ha elegido el usuario en la interfaz.

# PLANIFICACIÓN DEL PROYECTO

## 3.1. PLAN DE TRABAJO

En este apartado, se desglosa de forma detallada el trabajo realizado en el Trabajo Final de Título, ver **Tabla 3.1**. Consta de tres bloques ordenados secuencialmente y en total suman 300 horas de trabajo aproximadamente.

Fases	Duración estimada	Tareas
Estudio previo / Análisis	80 horas	Tarea 1.1: Estudio del editor de texto LATEX y la sintaxis de sus comandos básicos.
		Tarea 1.2: Estudio sobre las herramientas existentes para la conversión del código LATEX a HTML.
		Tarea 1.3: Estudio del entorno de programación de DEMOS online Image Processing On Line [IPOL].
Diseño / Desarrollo / Implementación	140 horas	Tarea 2.1: Diseño e implementación de un conversor de código LATEX a HTML.
		Tarea 2.2: Diseño de una colección de estilos css para las páginas webs resultantes del conversor.
		Tarea 2.3: Implementación del conversor en el entorno de DEMOS online de IPOL.
Evaluación / Validación / Prueba	20 horas	Tarea 3.1: Evaluación, validación y prueba del conversor off-line sin usar el entorno de DEMOS de IPOL. Se tomarán varios ficheros LATEX de ejemplo y se comprobará que el resultado de la conversión es correcto.
		Tarea 3.2: Evaluación, validación y prueba del conversor on-line en el entorno de DEMOS de IPOL. Se tomarán varios ficheros LATEX de ejemplo y se comprobará que el resultado de la conversión on-line es correcto.

Documentación / Presentación	60 horas	Tarea 4.1: Preparación de la documentación del proyecto y memoria final de acuerdo con el reglamento de TFT de la Escuela de Ingeniería Informática.
		Tarea 4.2: Preparación de la presentación oral del proyecto.

**Tabla 3.1:** Tabla que refleja las distintas fases del desarrollo del TFT.

### 3.2. METODOLOGÍA

Para abordar este Trabajo Final de Título planteado, se considera tarea fundamental comenzar con el estudio del editor de texto LATEX [LaTeX]. El estudio se centrará en los comandos básicos de dicho lenguaje y además tomaremos de referencia artículos de trabajos de investigación del Centro de Tecnologías de la Imagen [CTIM], para tener una idea general de los comandos más usados por los potenciales usuarios de la aplicación web resultante de este proyecto. Para esta primera parte del proyecto, se hace uso del compilador de LATEX MikTeX [MikTeX] en un entorno Windows 10.

El siguiente paso es el estudio del problema de la conversión de LATEX a HTML. Será necesario una correcta selección de los comandos que tengan un posible equivalente o semejante en el lenguaje de programación HTML que tenga un aspecto de visualización similar. Además, esta parte nos ayuda a descartar los comandos de complicada conversión o innecesarios para la finalidad de un prototipo como el que va a resultar este TFT. Se hace uso de HTML5 [HTML5] como versión de código para el lenguaje objeto.

Una de las características de este proyecto en diferencia con los ya existentes es la posibilidad de selección de diferentes estilos para el código HTML resultante del conversor, para cumplir con ello es necesaria la creación de diversas hojas de estilos CSS que darán forma a los resultados del conversor.

Previo al desarrollo del conversor, se ha de estudiar el entorno de programación de DEMOS de Image Processing On Line [IPOL] con la finalidad de implantar el conversor en la plataforma y así poder ofrecer la ventaja a los potenciales usuarios de no tener que instalar software.

### 3.3. PRESUPUESTO

A continuación, se expondrán los costes necesarios para la elaboración y culminación de este proyecto. Se dividen en: coste de personal, costes de hardware, costes de software y otros costes.

### 3.3.1. Costes de Personal

El coste mensual relativo al personal necesario para culminar el proyecto en un plazo de 6 meses se puede ver en la siguiente tabla.

Personal	Sueldo (€)
Jefe de proyecto	3.500 €
Programador	2.000 €
Diseñador	1.400 €
Total	6.900 €

**Tabla 3.2:** Tabla que refleja los costes mensuales del personal.

### 3.3.2. Costes de Hardware

En los costes de hardware vendrán incluidos todos los equipos necesario para el desarrollo del proyecto teniendo en cuenta el número de personas involucrado.

Equipo	Unidades	Precio unitario (€)	Total (€)
Ordenador de sobremesa	3	800 €	2.400 €
Monitor	3	100 €	300 €
Conjunto de teclado y ratón	3	30 €	90 €
Coste total			2.790 €

**Tabla 3.3:** Tabla que refleja los costes hardware del proyecto.

### 3.3.3. Costes de Software

En relación a los costes de software, todas las herramientas utilizadas son gratuitas.

Descripción	Coste (€)
MikTEX (compilador de LATEX)	0 €
g++ (compilador de C++)	0 €
Visual Studio Code	0 €
Servidor DEMO's IPOL	0 €
Google DOCS (redacción memoria)	0 €
Total	0 €

**Tabla 3.4:** Tabla que refleja los costes software del proyecto.

### 3.3.4. Otros costes

En este apartado se incluyen otros gastos mensuales necesarios como el alquiler de una oficina o el de suministros, que no entran en las categorías de costes descritas anteriormente.

Descripción	Coste (€)
Alquiler de oficina	1200 €
Gastos Oficina	200 €
Total	1.400 €

**Tabla 3.5:** Tabla que refleja otros costes mensuales del proyecto.

### 3.3.5. Costes totales

Los costes totales durante el desarrollo de LatexToHTMLOnLine se pueden ver en la siguiente tabla.

Costes	6 MESES (€)
Personal	41.400 €
Hardware	2.790 €
Software	0 €
Otros	8.400 €
Total	52.590 €

**Tabla 3.6:** Tabla que refleja los costes totales del proyecto durante el periodo de ejecución.

Además, a estos costes habría que añadir los gastos indirectos si los hubiera y los impuestos legales, en el caso de Canarias, el IGIC.

# DESARROLLO DE LatexToHTMLOnLine

---

Este capítulo recoge el procedimiento que se ha seguido para el desarrollo del conversor online LatexToHTMLOnLine.

## 4.1. ESTUDIO PREVIO

Para el desarrollo de LatexToHTMLOnLine ha sido necesario un estudio previo de dos materias fundamentales sobre las que se basa este proyecto final de título que son el LATEX e Image Processing On Line [IPOL].

### 4.1.1. Introducción a LATEX

LATEX es un sistema de edición de textos que está formado mayoritariamente por una serie de comandos del lenguaje TEX. En la **Figura 4.1** se puede ver un ejemplo sencillo en LATEX y su filosofía completamente distinta a los editores de texto convencionales que usan la estrategia “lo que ves es lo que obtienes”. Para obtener el resultado final de un documento escrito en LATEX es necesario compilar el código fuente donde se encuentran los comandos LATEX que definen el documento. La salida habitual de la compilación es un documento en formato PDF. En la **Figura 4.2** se muestra un ejemplo del resultado de la compilación de un documento LATEX.

```
\documentclass[12pt,a4paper]{article}

\begin{document}
\title{Ejemplo básico de documento Latex}
\author{Christian Brito Ramos \ \ ULPGC}

\maketitle
\abstract{
Ejemplo ilustrativo para la memoria del trabajo final de título LatexToHTMLOnLine
}

\end{document}
```

**Figura 4.1:** Ejemplo básico compilable en LATEX.

# Ejemplo básico de documento Latex

Christian Brito Ramos  
ULPGC

July 3, 2019

## Abstract

Ejemplo ilustrativo para la memoria del trabajo final de título  
LatexToHTMLOnLine

**Figura 4.2:** Resultado de ejecución de código LATEX.

Las posibilidades de LATEX utilizando comandos propios de TEX descritos en The Texbook [TheTexBook] es lo que hace a esta herramienta útil y práctica. La potencia de TEX unida a la facilidad de uso es lo que ha hecho de LATEX un lenguaje universalmente extendido entre el sector técnico y científico, llegando a ser un lenguaje predominante en artículos académicos y congresos. Una de las ventajas de LATEX es que la salida de la ejecución es siempre la misma independientemente del dispositivo o del sistema operativo.

LATEX sigue un sistema de código abierto que antes de ser estandarizado, permitió a numerosos usuarios crear nuevas utilidades de interés propio y que en muchas ocasiones se alejaban del verdadero propósito de su creación, llegando a surgir diversos dialectos LATEX incompatibles entre sí.

Tanto para el desarrollo del ejemplo de la **Figura 4.1**, como para los posteriores ficheros LATEX que servirán de testeo y validación del conversor, se utilizará TeXworks [TeXworks]. TexWorks es un editor multiplataforma que soporta la mayoría de los sistemas operativos y viene incluido al instalar el compilador de LATEX MikTeX [MikTeX].

Lo más importante de este apartado es la puesta en conocimiento que ha supuesto el estudio de este lenguaje, ya que será de vital importancia conocer tanto su funcionamiento como su ejecución. Además ha sido necesaria una vista general de los comandos LATEX existentes para poder disponer de buen juicio en el siguiente apartado donde se estudian las herramientas existentes de conversión de LATEX a HTML.

### 4.1.2. Estudio de herramientas existentes de conversión LATEX/HTML.

En el [Capítulo 2](#) se han expuesto las herramientas existentes que pueden tener semejanza con este Trabajo Final de Título, se puede concluir que hay numerosas herramientas que persiguen el mismo fin, pero ninguna se asemeja a las posibilidades que ofrece LatexToHTMLOnLine.

El mayor atractivo de este proyecto en comparación con los demás conversores es la interfaz que se proporciona junto con la posibilidad de selección de estilos CSS.

Si es cierto que la herramienta LaTeXML [LTXML] dispone de una herramienta on-line [OLXML], pero como se ha explicado en el apartado [2.2.1](#), es una herramienta que traduce LATEX a XML a pesar de que por otro lado dispongan de un post-procesado a HTML. Independientemente de que se pueda traducir LATEX a HTML, la interfaz on-line es poco intuitiva y existen opciones que concluyen en fallos de la página web. Se entiende que la versión on-line tenga defectos ya que el énfasis de LaTeXML se encuentra en la versión ejecutable local.

Como norma general, ninguno de los conversores estudiados da la opción a los usuarios de escoger algún tipo de estilo en la salida del documento.

Además, cabe destacar que todos los conversores llegan a la misma conclusión a la que hemos llegado tras el estudio previo de LATEX, y es que se hace muy costosa la programación de cada uno de los comandos LATEX existentes junto con todas las posibilidades que cada uno de ellos permite.

#### 4.1.3. *Image Processing On Line*

La revista científica *Image Processing Online* (IPOL), cuyo logo vemos en la **Figura 4.3**, es la piedra angular de una iniciativa que nace en el año 2010 de la mano del profesor Jean-Michel Morel y su equipo de colaboradores. IPOL promueve un entorno de difusión de ciencia abierta y reproducible en el campo del procesamiento de señales (imagen, vídeo, audio, 3D) con el objetivo de presentar trabajos cuyos resultados, además de aportar valor científico, puedan ser confirmados con exactitud. Esto lo consigue mediante una estrategia editorial que difiere de otros sistemas más tradicionales y por su entorno de experimentación online.



**Figura 4.3:** Logo de la revista IPOL. Revista enfocada en la investigación reproducible.

En ciertas ocasiones, las publicaciones científicas no siempre explican con claridad los detalles de su implementación, lo que dificulta poder reproducir de forma exacta los experimentos allí descritos. Este “oscurantismo” causa frustración y de alguna manera frena el

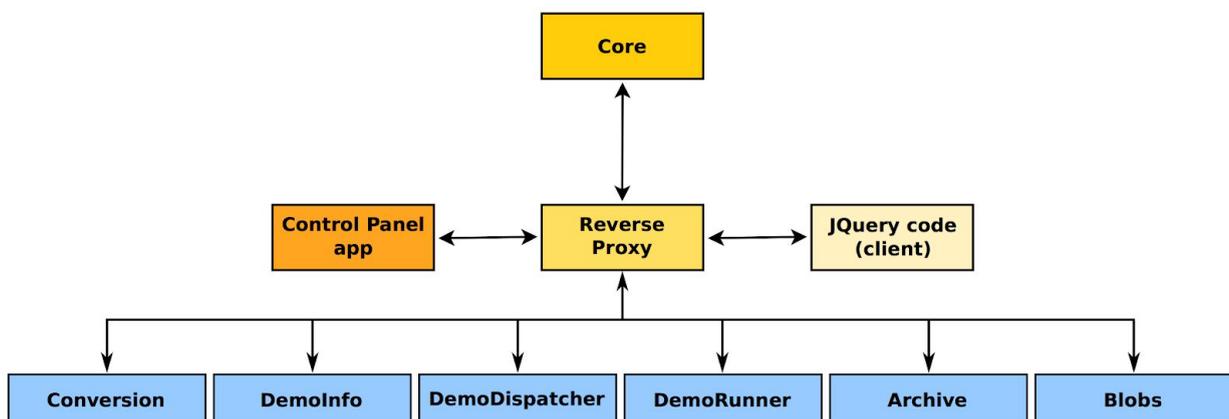
avance científico y, en algunos casos, incluso genera dudas acerca de la veracidad de los resultados publicados.

En este sentido, una publicación en IPOL supone presentar sus algoritmos mediante diagramas o una notación formal esquemática (pseudocódigo) que permitan su reproducibilidad. Además, los autores facilitan el código fuente que han desarrollado para su descarga online y verificación (protegido mediante licencia). Esto favorece publicar artículos científicos que además sean confiables.

El proceso de publicación se divide en dos etapas: primero, los revisores evalúan el interés científico, los experimentos y la reproducibilidad del trabajo; en segundo lugar, si esta evaluación es positiva, se publica una demo en línea que utiliza el código fuente original de los autores para poder experimentar con este de manera cómoda y sencilla desde cualquier navegador web. Las demos IPOL descargan, compilan y ejecutan los algoritmos entregados por los autores del artículo. Esto garantiza que la demo reproduce exactamente los resultados que los autores afirman conseguir.

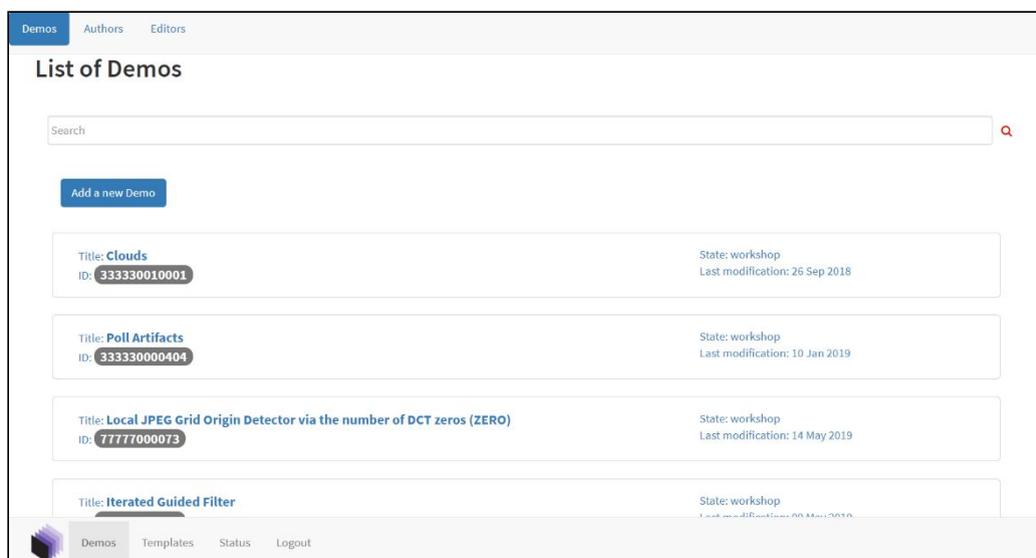
El sistema de demos IPOL supone una ventaja importante para la comunidad científica ya que permite probar un método incluso antes de considerar si se debe invertir más tiempo en estudiar un trabajo de investigación en profundidad. Las demos también pueden archivar sus experimentos si son realizados con datos propios de los usuarios, lo que facilita que investigadores localizados en sitios distantes puedan colaborar de forma remota y que usuarios externos experimenten con el sistema utilizando sus propios datos. Gracias a esto, un método puede extender sus objetivos originales más allá de las ideas iniciales de sus autores estudiando nuevos fenómenos.

El sistema de demos de IPOL está implementado como una plataforma web cuya arquitectura es un modelo “SOA” basado en microservicios entre distintos módulos que se comunican entre sí (**Figura 4.4**). El objetivo principal del sistema es crear fácilmente la mayor cantidad de demos en línea sin necesidad de invertir un esfuerzo muy grande integrando todas al sistema. Pueden ver más información en la publicación (Arévalo et al, 2017).



**Figura 4.4:** Diagrama con los módulos principales de IPOL.

Desde el punto de vista de la creación de demos, el sistema utiliza una aplicación web en *Django* que permite acceder a un menú con opciones para configurar los módulos disponibles. Esta aplicación se conoce como “Panel de Control”. Vemos un ejemplo en la siguiente figura.



**Figura 4.5:** Ejemplo de la pantalla principal del “Panel de Control” en IPOL.

La principal funcionalidad utilizada en el presente Trabajo de Fin de Grado es, evidentemente, la creación de demos. Esta depende en primera instancia de la elaboración de una DDL (Demo Description Lines) que configura las principales características de la demo. La DDL es una sintaxis abstracta, escrita en formato JSON (Notación de Objetos de JavaScript), que especifica las demos de IPOL. Su objetivo principal es simplificar lo más posible la creación de demostraciones describiéndolas sin la necesidad de escribir código en Python o HTML. Esto permite una rápida edición de demos para la revista. Las siguientes secciones describen cada una de las claves principales de la DDL:

- ❖ general: opciones generales;
- ❖ build: descargue y compile el código fuente;
- ❖ inputs: descripción de las entradas;
- ❖ params: descripción de los parámetros y control del usuario;
- ❖ run: script o binario que necesita ser llamado para la ejecución, junto con con sus parámetros;
- ❖ archive: qué parámetros y resultados serán almacenados en el historial de esa demo;
- ❖ results: los elementos que se mostrarán como resultados.

El panel de control de IPOL proporciona un editor JSON con un validador simple. La mayoría de los errores de sintaxis se detectan en tiempo real y son reportados por esta herramienta gráfica.

En distintas ocasiones, como por ejemplo en la demo desarrollada en este trabajo, es necesario utilizar *scripts* o datos auxiliares que den soporte a la demo.

## 4.2. DESARROLLO DEL CONVERTSOR

En este apartado se explica cómo se ha llevado a cabo el desarrollo del conversor, comenzando por una explicación de los comandos traducidos y posteriormente su implementación.

### 4.2.1. Comandos convertidos.

Como ya se ha explicado anteriormente, existen numerosos comandos LATEX y cada uno puede tener infinidad de variaciones. En este apartado veremos los comandos escogidos para implementar en la conversión.

Hasta este momento se ha podido tener la percepción de que cada comando LATEX tiene su equivalente en HTML y la realidad es que no. La conversión se realiza identificando el resultado que puede dar un comando en LATEX y buscar el equivalente o incluso los equivalentes en HTML. En la mayoría de las ocasiones, el resultado de la traducción de un comando LATEX involucra varias etiquetas HTML para obtener un resultado semejante.

Se debe tener en cuenta que en la mayoría de las ocasiones es necesaria la inclusión de clases e identificadores en las etiquetas HTML con la finalidad de que un fichero css le pueda aplicar estilo a cada una de las etiquetas.

### Cabecera LATEX

La cabecera de un documento LATEX incluye una gran cantidad de información como son los paquetes que utilizan para el formateado, el tamaño de la página, el tipo de letra de referencia, el título del documento, el autor y la fecha, etc.. En esta primera versión del conversor, de la cabecera del documento LATEX solo gestionaremos el título, el autor y la fecha.

### Título - Fecha - Autor

En la siguiente figura se muestra la sintaxis de LATEX para gestionar el título, la fecha y el autor del documento:

```
\title{LatexToHtmlOnline: Conversor en \'\{i\}nea de Latex a HTML}  
\author{Christian Brito Ramos \ \ ULPGC}  
\date{25 de Junio de 2019}
```

**Figura 4.6:** Código LATEX equivalente a título, fecha y autores.

El procedimiento que se ha seguido para crear la conversión, es siempre el mismo para cada uno de los comandos. En primer lugar buscamos una relación directa entre el comando LATEX que se va a convertir y un comando HTML. En caso de negativa, hay que generalizar una estructura de comandos que obtengan un resultado similar.

En el caso de ‘\title’, si tiene un equivalente en HTML, ver **Figura 4.7** . Pero el resultado obtenido no es el esperado, pues ‘<title></title>’ únicamente edita el título de la pestaña del navegador, por lo que no nos sirve como código objeto.

```
<title> LatexToHtmlOnLine (Christian Brito) </title>
```

**Figura 4.7:** Etiqueta HTML título página web.

En los casos de ‘\author’ y ‘\date’, tampoco tienen un equivalente en HTML, por lo tanto la única opción es crear una estructura de comandos que obtengan un resultado similar en HTML. El código objeto resultante de la **Figura 4.6** se puede ver en la siguiente figura.

```
<!-- title -->
<div id="paper_title"> LatexToHtmlOnLine: Conversor en línea de Latex a HTML</div><br>
<!-- Authors -->
<div id="adsection">
  <div id="authors">
    Christian Brito Ramos <br> UPGC
  </div><br>
<!-- date -->
  <div id="date">
    25 de Junio de 2019
  </div><br>
</div>
```

**Figura 4.8:** Traducción LATEX/HTML de título, fecha y autor.

El estilo de traducción que se ha podido ver para fecha, título y autor, va a ser muy común ya que es bastante complicado que coincidan los comandos LATEX con equivalentes HTML.

### **Abstract (Resumen)**

El ‘\abstract{’ es un comando LATEX empleado para escribir un resumen del artículo, ver **Figura 4.9**. No tiene un equivalente en HTML y por lo tanto se hace uso del potencial de los estilos CSS para obtener un resultado similar a LATEX.

## Resumen

El objetivo de este proyecto de TFT es implementar un prototipo de conversor en línea de un fichero en formato LATEX para escribir textos científicos a un fichero HTML que se pueda visualizar en un navegador web. Además se da la opción al usuario de elegir el estilo con el que sale formateado el texto HTML. La mayor parte de los conversores en línea que existen en el mercado están diseñados para convertir pequeños trozos de código LATEX (principalmente fórmulas), no admiten la conversión de un texto LATEX entero ni seleccionar estilos. Para conseguir los objetivos

**Figura 4.9:** Ejemplo del comando “\abstract” en LATEX.

Este comando tiene una peculiaridad en LATEX, y es que según la definición inicial del idioma del documento, muestra en la cabecera el título “RESUMEN”. Un posible resultado equivalente podría ser el de la **Figura 4.10**.

```
<!-- Abstract -->
<div id="abstract">
  <p> <b> Resumen </b> </p>
  <div class="justified">
    | | \abstract
  </div>
</div><br>
```

**Figura 4.10:** Ejemplo del comando “\abstract” en HTML.

## Índice

El siguiente paso es el tratamiento del índice. LATEX dispone del comando ‘\tableofcontents’ (ver [OVERLeaf]) que identifica automáticamente todas las secciones del documento y estructura las cabeceras de cada sección en una lista a modo de índice, ver la siguiente figura .

```

\documentclass{article}
\usepackage[utf8]{inputenc}

\title{Sections and Chapters}
\author{Gubert Farnsworth}
\date{ }

\begin{document}

\maketitle

\tableofcontents

\section{Introduction}

This is the first section.

Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Etiam lobortis facilisis sem. Nullam nec mi et
neque pharetra sollicitudin. Praesent imperdiet mi nec ante.
Donec ullamcorper, felis non sodales...

\addcontentsline{toc}{section}{Unnumbered Section}
\section*{Unnumbered Section}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilisissem. Nullam nec mi et neque pharetra
sollicitudin. Praesent imperdiet mi nec ante...

\section{Second Section}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Etiam lobortis facilisissem. Nullam nec mi et neque pharetra
sollicitudin. Praesent imperdiet mi nec ante...

\end{document}

```

**Figura 4.11:** “\tableofcontents” en código LATEX.

En la **Figura 4.12** se puede ver el resultado de la ejecución del código anterior. Además, aunque no se pueda apreciar, cada uno de los campos de la lista tiene un enlace a su respectiva sección.

Sections and Chapters

Gubert Farnsworth

<b>Contents</b>	
<b>1 Introduction</b>	<b>1</b>
<b>Unnumbered Section</b>	<b>1</b>
<b>2 Second Section</b>	<b>2</b>

**1 Introduction**

This is the first section.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

**Unnumbered Section**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue,

**Figura 4.12:** Resultado de “\tableofcontents” en código LATEX.

Como cabía de esperar, no existe ningún comando en HTML que genere automáticamente una sección “Índice”, por lo que una vez más hay que adecuar la forma en la que se obtiene la tabla completa del índice en base a las posibilidades que ofrece HTML, ver **Figura 4.13**. Los elementos y procedimientos necesarios para su construcción son:

- ❖ Crear una estructura con varios ‘<div></div>’ para encapsular el índice en el documento y en el futuro poder aplicar estilos. Además, internamente hay que diferenciar el título “Índice” del contenido.
- ❖ Será necesaria una lista numerada haciendo uso de ‘<ol></ol>’.
- ❖ Cada ítem de la lista se trata con ‘<li></li>’.
- ❖ Finalmente, para poder aplicar enlaces a cada una de las secciones del documento será necesario el uso de la etiqueta ‘<a></a>’ con su correspondiente ‘href’ referenciando a la sección marcada.

```

<!-- Sections -->
<div class="story">
  <div class="storyTitle">
    <h2 class="story"> Índice </h2>
  </div>
  <div class="partsIndex">
    <ol class="enumerate_index">
      <li class="indexList"><a class="index_link" href="#sec0"> Introducción </a></li>
      <li class="indexList"><a class="index_link" href="#sec1"> Gestión de párrafos, tipos de letra y caracteres ASCII extendido </a></li>
      <li class="indexList"><a class="index_link" href="#sec2"> Gestión de tablas y listas </a></li>
      <li class="indexList"><a class="index_link" href="#sec3"> Imágenes y vídeos </a></li>
      <li class="indexList"><a class="index_link" href="#sec4"> Fórmulas Matemáticas </a></li>
      <li class="indexList"><a class="index_link" href="#sec5"> Bibliografía y enlaces </a></li>
      <li class="indexList"><a class="index_link" href="#sec6"> Comentarios sobre otras funcionalidades de LATEX no implementadas </a></li>
    </ol>
  </div>
</div>
<br>

```

Figura 4.13: Resultado de “\tableofcontents” en código HTML.

### Secciones del documento

El tratamiento de las secciones tiene una conversión más directa, ya que solo es seguir un procedimiento iterativo de sustitución, ver **Figura 4.14**. Además, hay que tener en cuenta que cada etiqueta ‘<a></a>’ de cada sección, debe tener una referencia equivalente a la de su correspondiente ítem en el índice.



```

\section{Introducci'on}
<div class="sectionTitle"><h2><a name="sec0"> Introducción </a> </h2></div><br>

```

Figura 4.14: Sustitución de las secciones.

### Párrafos

La gestión de párrafos en LATEX es distinta a como lo gestiona HTML. Principalmente en LATEX, hay dos formas de identificar saltos de línea, ‘\newline’ y ‘\\’. Para ambos casos el criterio que se ha seguido es la sustitución de ambos comandos por la etiqueta HTML ‘<br>’.

## Negrita e Itálica

En el caso del texto en itálica o negrita, se puede realizar una conversión directa de LATEX a HTML.

En LATEX, la sentencia ‘`\bf texto en negrita`’ transforma el texto en negrita. El equivalente en HTML sería ‘`<b>texto en negrita</b>`’.

Lo mismo ocurre para el texto en itálica, siendo ‘`\em texto en itálica`’ en LATEX y ‘`<i>texto en itálica</i>`’ el equivalente en HTML.

## Caracteres ASCII extendido

Respecto a los acentos y otros caracteres especiales, si el documento LATEX ha sido realizado usando los paquetes de LATIN existentes aparecerán igual en el documento LATEX que en el HTML, sin embargo si no se usan los paquetes de LATIN cada caracter se escribe en LATEX usando un comando especial irreconocible por HTML, por ello se ha implementado una conversión de estos comandos LATEX a HTML.

En LATEX existen dos formas de poner acentos, siendo “`\’a`” un ejemplo para “`á`” y con “`\’{a}`” se obtiene el mismo resultado. En este caso simplemente hay que eliminar esas directivas y sustituirla directamente por el resultado.

## Listas

Tanto en LATEX como en HTML se diferencian dos tipos de listas, las listas numeradas y las no enumeradas. En este caso, el proceso de conversión es más complicado que los comandos anteriores, pues además de ser más larga la estructura a convertir, entra en juego la recursividad entre listas. En la **Figura 4.15** se puede ver un ejemplo de cómo sería una lista sin numerar en LATEX.

```
\begin{itemize}
\item Primer elemento de la lista
\item Segundo elemento de la lista
\item Tercer elemento de la lista
\item Cuarto elemento de la lista
\item Quinto elemento de la lista
\end{itemize}
```

**Figura 4.15:** Lista no numerada en LATEX.

El equivalente en HTML se puede ver en la **Figura 4.16**. El algoritmo de conversión reconoce en primer lugar una lista no numerada con el comando LATEX “\begin{itemize}” y en el caso de ser una lista numerada sería “\begin{enumerate}”.

Esta identificación sirve de preámbulo para inyectar la respectiva etiqueta HTML en el fichero resultado, siendo “<ol>” una lista numerada o “<ul>” una lista simple. Para el caso de cada uno de los ítems de la lista en LATEX “\item”, se usa el mismo comando tanto si es una “\enumerate” o una “\itemize”. Lo mismo ocurre en HTML, la etiqueta “<li>” cumple la misma función en ambos tipos de lista.

```
<div class="itemizeList">
  <ul class="itemizeList">
    <li class="itemizeList">Primer elemento de la lista</li>
    <li class="itemizeList">Segundo elemento de la lista</li>
    <li class="itemizeList">Tercer elemento de la lista</li>
    <li class="itemizeList">Cuarto elemento de la lista</li>
    <li class="itemizeList">Quinto elemento de la lista</li>
  </ul>
</div>
```

**Figura 4.16:** Lista no numerada en HTML.

### Tablas

El proceso de conversión de una tabla LATEX a HTML es similar al caso anterior, con la salvedad de que hay que recorrer cada una de las casillas para inyectar información. En la **Figura 4.17** se puede ver un ejemplo de una tabla en LATEX.

```
\begin{tabular}[t]{ccc}
Nombre & Apellidos & Edad \\
Juan & Garcia López & 22 \\
Pedro & Del Olmo & 43 \\
Agustín & Brito Perez & 30 \\
Laura & Pulido Ruiz & 11 \\
\end{tabular}
```

**Figura 4.17:** Tabla 5x3 en LATEX.

Nombre	Apellidos	Edad
Juan	Garcia López	22
Pedro	Del Olmo	43
Agustín	Brito Perez	30
Laura	Pulido Ruiz	11

**Figura 4.18:** Resultado de una tabla en LATEX.

El inconveniente de estructurar una tabla en LATEX es que la cabecera de la tabla no tiene porqué tener un formato especial (ver **Figura 4.18**), es decir, el creador del documento puede o no añadir distintivos a la cabecera de la tabla en forma de líneas con diferentes grosores. En nuestro caso al transformar la tabla a HTML utilizamos los códigos de estilo CSS para ofrecer al usuario diferentes posibilidades en el formateo de las tablas. De esta manera el propio usuario es el que va a escoger el fichero CSS que mejor se adapte a sus necesidades. Uno de los posibles resultados del conversor se puede ver en la siguiente figura.

<b>Nombre</b>	<b>Apellidos</b>	<b>Edad</b>
Juan	Garcia López	22
Pedro	Del Olmo	43
Agustín	Brito Perez	30
Laura	Pulido Ruiz	11

**Figura 4.19:** Resultado de una tabla HTML con CSS.

La estructura general en HTML para los ejemplos anteriores se puede ver en la **Figura 4.20**. Destacar que las etiquetas HTML tienen sus clases determinadas con la finalidad de que el fichero CSS pueda identificar los elementos.

```

<div class="tablaDiv">
  <table id="css_table">
    <tr class="spaceBox">
      <th> Nombre </th>
      <th> Apellidos </th>
      <th> Edad </th>
    </tr>
    <tr class="spaceBox">
      <td> Juan </td>
      <td> Garcia López </td>
      <td> 22 </td>
    </tr>
    <tr class="spaceBox">
      <td> Pedro </td>
      <td> Del Olmo </td>
      <td> 43 </td>
    </tr>
    <tr class="spaceBox">
      <td> Agustín </td>
      <td> Brito Perez </td>
      <td> 30 </td>
    </tr>
    <tr class="spaceBox">
      <td> Laura </td>
      <td> Pulido Ruiz </td>
      <td> 11 </td>
    </tr>
  </table>
</div>

```

Figura 4.20: Resultado de una tabla en HTML.

### Imágenes y vídeos

Se ha realizado una conversión básica de los comandos LATEX para visualizar imágenes y vídeos. Por defecto, los ficheros de imágenes y vídeos deben estar en la misma carpeta que el fichero HTML generado. Por tanto, al convertir en línea un fichero LATEX con imágenes y vídeos estos en general no se verán en el navegador de la aplicación en línea pues será necesario que el usuario se descargue el HTML generado y lo ponga en la carpeta donde están las imágenes y vídeos. Es decir, el conversor simplemente hace uso de la ruta que lee de cada una de las inserciones de imágenes o vídeos.

En cualquier caso, el usuario siempre tiene la opción de editar el fichero HTML generado y poner sus propias rutas para las imágenes o vídeos.

El tratamiento del comando de inserción de imagen en LATEX (ver **Figura 4.21**), se limita a identificar la ruta con el nombre de la imagen. En el momento de transformar a HTML (ver **Figura 4.22**), se inyecta la ruta en la etiqueta equivalente HTML.

```
\includegraphics [width=0.7]\linewidth]{imagen.jpg}
```

**Figura 4.21:** Código LATEX para inyección de imagen.

```

```

**Figura 4.22:** Código HTML para inyección de imagen.

Con respecto a las opciones que permite “\includegraphics”, no se tienen en cuenta. El formateo de las imágenes en HTML se realiza usando códigos de estilo CSS, dando la opción al usuario de elegir entre varias posibilidades. De esta manera, el usuario es el encargado de elegir el fichero CSS que mejor se adapte a sus necesidades.

Para el caso de los vídeos, cuyo comando LATEX es “\includemovie”, se traduce de la misma manera que en las imágenes. Únicamente se obtiene la ruta del vídeo proporcionada en el comando LATEX y se inyecta en la etiqueta HTML, ver **Figura 4.23**. De nuevo, el formateo en HTML se deja a los estilos CSS.

```
<video width="80%" controls ><source src=" penguinschasingbutterfly.mp4" > EL VIDEO DEBE ESTAR EN LA MISMA CARPETA QUE EL FICHERO HTML.</video>
```

**Figura 4.23:** Código HTML para inyección de vídeo.

## **Fórmulas matemáticas**

El LATEX tiene un potencial enorme para escribir todo tipo de fórmulas matemáticas. Sin embargo su conversión a HTML es un problema complejo. MathML (Mathematical Markup Language) es un lenguaje para escribir ecuaciones matemáticas cuyo objetivo es integrar fórmulas matemáticas en documentos web. Actualmente es parte de HTML5.

Las principales limitaciones de MathML son, por un lado, que actualmente no todos los navegadores interpretan correctamente su código y por otro lado que la sintaxis que utiliza es compleja y totalmente diferente a la de LATEX. Afortunadamente dos de las sociedades matemáticas más importantes de los Estados Unidos: la “American Mathematical Society” (AMS) y la “Society for Industrial and Applied Mathematics (SIAM)” han promovido el

proyecto Mathjax que es una plataforma de código abierto y JavaScript que permite visualizar en la web fórmulas escritas directamente en la sintaxis de LATEX.

```
<script type="text/javascript" async
  src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/MathJax.js?config=TeX-MML-AM_CHTML" async>
</script>
```

**Figura 4.24:** Librería MathJAX en HTML.

Para incorporar las librerías de MathJax simplemente hay que añadir una directiva al principio del fichero HTML para que se genere una visualización de las fórmulas en formato LATEX, ver **Figura 4.24**. Esta es la solución que se ha adoptado en este trabajo para visualizar en un navegador web las fórmulas matemáticas escritas en LATEX por ser una solución sencilla de implementar y que funciona además en cualquier navegador. Se puede ver un ejemplo de fórmula matemática LATEX en la **Figura 4.25**.

```
\begin{equation}
\label{fórmula}
\int_a^b f(x)dx = \frac{e^{x+1} + C_0}{\sqrt{xy^z} + \tan^2(y)} \left( 2\pi i + \frac{\gamma^\epsilon}{x} \right)^{\frac{2}{3}}
\end{equation}
```

**Figura 4.25:** Código de fórmula matemática en LATEX.

El resultado en el navegador se puede ver en la **Figura 4.26**. A la hora de escribir en HTML, simplemente hay que inyectar el código LATEX referente a la fórmula en el fichero HTML(ver Figura 21).

$$\int_a^b f(x)dx = \frac{e^{x+1} + C_0}{\sqrt{xy^z} + \tan^2(y)} \left( 2\pi i + \frac{\gamma^\epsilon}{x} \right)^{\frac{2}{3}}$$

**Figura 4.26:** Resultado de fórmula matemática en LATEX.

## Bibliografía

La gestión de la bibliografía es completamente distinta en LATEX y HTML. Se ha optado por una solución sencilla que traslada las etiquetas de los ítems bibliográficos escritos en LATEX a HTML. De la misma forma los enlaces a páginas web se convierten desde LATEX a HTML para su correcta interpretación por el navegador. La estructura de una sección bibliográfica en LATEX se puede ver en la siguiente figura.

```
\begin{thebibliography}{1}
\bibitem{La86} Leslie Lamport {\em LaTeX : A document Preparation System}. Addison-Wesley, 1986.
\bibitem{MathML} {\em Mathematical Markup Language (MathML) }. \url{https://en.wikipedia.org/wiki/MathML}, 2015.
\bibitem{MathJax} {\em Beautiful math in all browsers (MathJax) }. \url{https://www.mathjax.org/}, 2010.
\end{thebibliography}
```

**Figura 4.27:** Declaración de zona bibliográfica en LATEX.

La traducción a HTML se puede ver en la **Figura 4.28**. El algoritmo reemplaza las etiquetas correspondientes a los estilos de texto correspondientes, inserta saltos de línea donde corresponde con “<br>” y por último se llama a otra rutina para tratar las correspondientes direcciones WEB.

```
<div class="bibliog"><p class="pbiblio"><br>
<br>[La86] Leslie Lamport <i>LaTeX : A document Preparation System</i> . Addison-Wesley, 1986.<br>
<br>[MathML] <i>Mathematical Markup Language (MathML) </i> .<a href="https://en.wikipedia.org/wiki/MathML">https://en.wikipedia.org/wiki/MathML</a> , 2015.<br>
<br>[MathJax] <i>Beautiful math in all browsers (MathJax) </i> .<a href="https://www.mathjax.org/">https://www.mathjax.org/</a> , 2010.<br>
</p></div><br>
```

**Figura 4.28:** Declaración de zona bibliográfica en HTML.

Existen mucha literatura sobre el uso de LATEX, por ejemplo, [La86] , es una referencia muy usada para ello. Los proyectos que se describen en las referencias [MathML] y [MathJax] tienen por objetivo común la visualización por internet de fórmulas matemáticas como se ha visto anteriormente.

## Funciones LATEX no implementadas

Como se ha comentado a lo largo de este documento, el LATEX tiene una enorme variedad de funcionalidades y la consideración de todas excede en mucho el alcance de un TFT. Dentro de las cosas que no se han abordado caben destacar las siguientes:

- ❖ La asignación automática de números a las ecuaciones matemáticas y otras etiquetas del documento LATEX, así como las referencias cruzadas no se ha realizado.

- ❖ La conversión de los comandos LATEX se ha realizado sin contar con todas sus posibles opciones de configuración. La variedad en este sentido puede ser enorme.
- ❖ La conversión de estructuras anidadas complejas como por ejemplo tablas que contienen a su vez tablas puede no funcionar.
- ❖ En el proceso de conversión se han eliminado los comentarios en LATEX, así como algunos de los comandos LATEX que no se gestionan para que no aparezcan en el documento HTML.

La inclusión de nuevas funcionalidades de LATEX a este prototipo, como las arriba mencionadas u otras es un interesante trabajo futuro a realizar.

#### 4.2.2. Implementación en C++

Para la implementación en C++ se ha creado una librería llamada latex2html. En dicha librería se ha creado la clase “ArrayChar” para gestionar la conversión del texto LATEX a la sintaxis de HTML. Dicha conversión se hace de forma progresiva gestionando los comandos LATEX de forma independiente. De este modo se han creado procedimientos para gestionar las siguientes operaciones de conversión:

- ❖ Gestión de los acentos y ñes.
- ❖ Gestión de párrafos y saltos de línea.
- ❖ Gestión de los tipos de letra itálica y negrita.
- ❖ Gestión del título del documento.
- ❖ Gestión del autor del documento.
- ❖ Gestión de la fecha del documento.
- ❖ Gestión del abstract.
- ❖ Gestión del índice de contenidos del documento.
- ❖ Gestión de enlaces WEB.
- ❖ Gestión de lista no enumeradas.
- ❖ Gestión de listas numeradas.
- ❖ Gestión de las tablas.
- ❖ Gestión de fórmulas matemáticas.
- ❖ Gestión de imágenes y vídeos.
- ❖ Gestión de la bibliografía
- ❖ Eliminación de los comentarios en LATEX.
- ❖ Eliminación de algunos comandos LATEX no gestionados (para que no aparezcan en el documento HTML)

Para facilitar el procedimiento de conversión se utilizan plantillas de referencia para construir el fichero HTML. En la **Figura 4.29** se muestra un ejemplo de estos plantillas, en este caso formateando en el idioma español, en este template se utilizan las siguientes palabras claves para facilitar la gestión de la conversión:

- ❖ **##CSS##** : se utiliza para identificar y posicionar en el archivo HTML el fichero de estilos CSS elegido por el usuario.
- ❖ **\title** : se utiliza para fijar la localización en el archivo HTML del título del documento.
- ❖ **\author** : se utiliza para fijar la localización en el archivo HTML del autor del documento.

- ❖ `\date` : se utiliza para fijar la localización en el archivo HTML de la fecha del documento.
- ❖ `\abstract` : se utiliza para fijar la localización en el archivo HTML del resumen del documento.
- ❖ `\section` : se utiliza para fijar la localización en el archivo HTML del título del índice de contenidos del documento.
- ❖ `\title` : se utiliza para fijar la localización en el archivo HTML del título del documento.
- ❖ `\begin{document}` : se utiliza para fijar la localización en el archivo HTML del comienzo del texto del documento.

Las operaciones de conversión localizan estas palabras claves en el template y las sustituyen por lo que corresponda en cada caso. De esta forma el fichero HTML se va creando de forma progresiva atendiendo a todos los elementos de la sintaxis LATEX contemplada.

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <script type="text/javascript" async
    src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.5/MathJax.js?config=TeX-MML-AM_CHTML" async>
  </script>
  <title> LatexToHtmlOnLine (Christian Brito) </title>
  <link rel="stylesheet" href="http://ctim.ulpgc.es/css/##CSS##" type="text/css">
</head>
<body>
  <div id="layout_wrapper">
    <div id="layout_container">
      <!-- title -->
      <div id="paper_title"> \title</div><br>
      <!-- Authors -->
      <div id="adsection">
        <div id="authors">
          | \author
        </div><br>
        <!-- date -->
        <div id="date">
          | \date
        </div><br>
      </div>
      <!-- Abstract -->
      <div id="abstract">
        <p> <b> Resumen </b> </p>
        <div class="justified">
          | \abstract
        </div>
      </div><br>
      <!-- Sections -->
      <div class="story">
        <div class="storyTitle">
          | <h2 class="story"> Índice </h2>
        </div>
        <div class="partsIndex">
          <ol class="enumerate_index">
            | \section
          </ol>
        </div>
      </div>
      \begin{document}
    </div>
  </div>
</body>
</html>

```

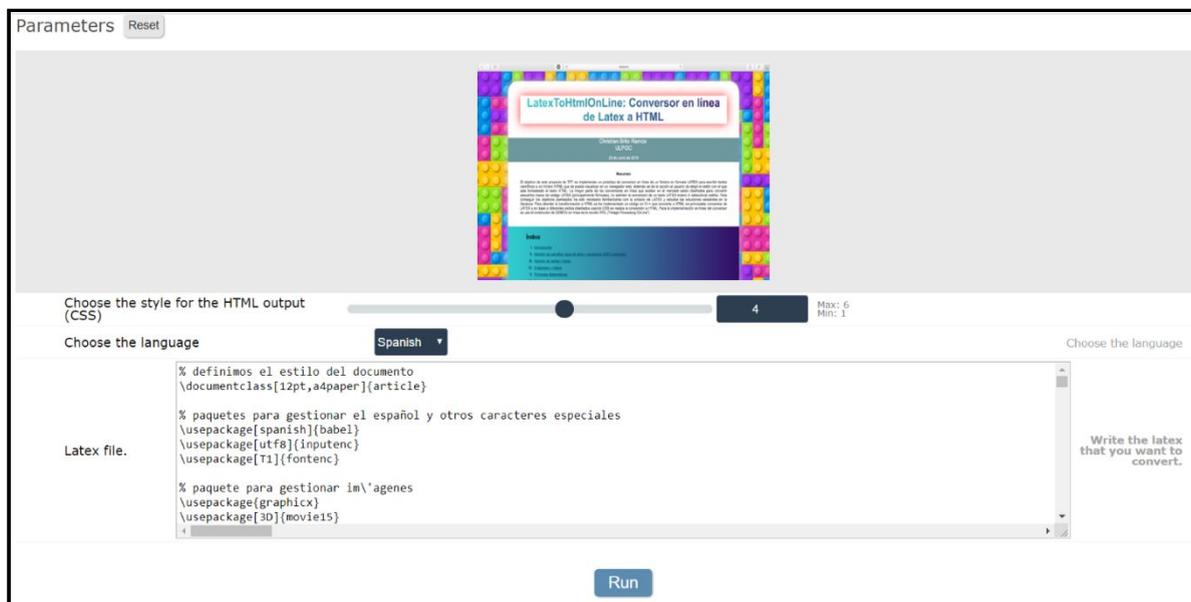
Figura 4.29: Template HTML en español.

### 4.2.3. Demo Online

#### ¿Cómo funciona la demo?

Entre los objetivos de este Trabajo de Fin de Grado se encuentra el desarrollo de una demo en línea integrada en el sistema IPOL. Comentamos aquí sus principales características.

En la **Figura 4.30** se observa la pantalla principal de la demo dónde los usuarios pueden escoger entre varios estilos CSS utilizando el slider correspondiente. Cada valor de ese slider está asociado a una representación visual que permite previsualizar cómo será el resultado. A continuación, la demo ofrece un parámetro para seleccionar el tipo de idioma a trabajar por el conversor. Finalmente, se ofrece un fichero LATEX por defecto en una caja de texto. Este fichero podrá ser modificado por el usuario aportando el código LATEX que desee para su web en HTML.



**Figura 4.30:** Página principal de la DEMO.

Una vez se seleccionan los parámetros, el usuario deberá pulsar RUN cuya acción producirá un fichero (output.html) que se podrá descargar mediante un click de botón o abrirlo directamente en el navegador tal y como se ve en la **Figura 4.31**. En la **Figura 4.32** se observa el resultado de utilizar una leve modificación del texto genérico de la demo, con lenguaje español y el estilo 4 dentro de los posibles CCS's.



**Figura 4.31:** Zona de resultados dentro de la pantalla principal una vez terminada la ejecución.



**Figura 4.32:** Ejemplo de página HTML resultante tras usar el conversor online.

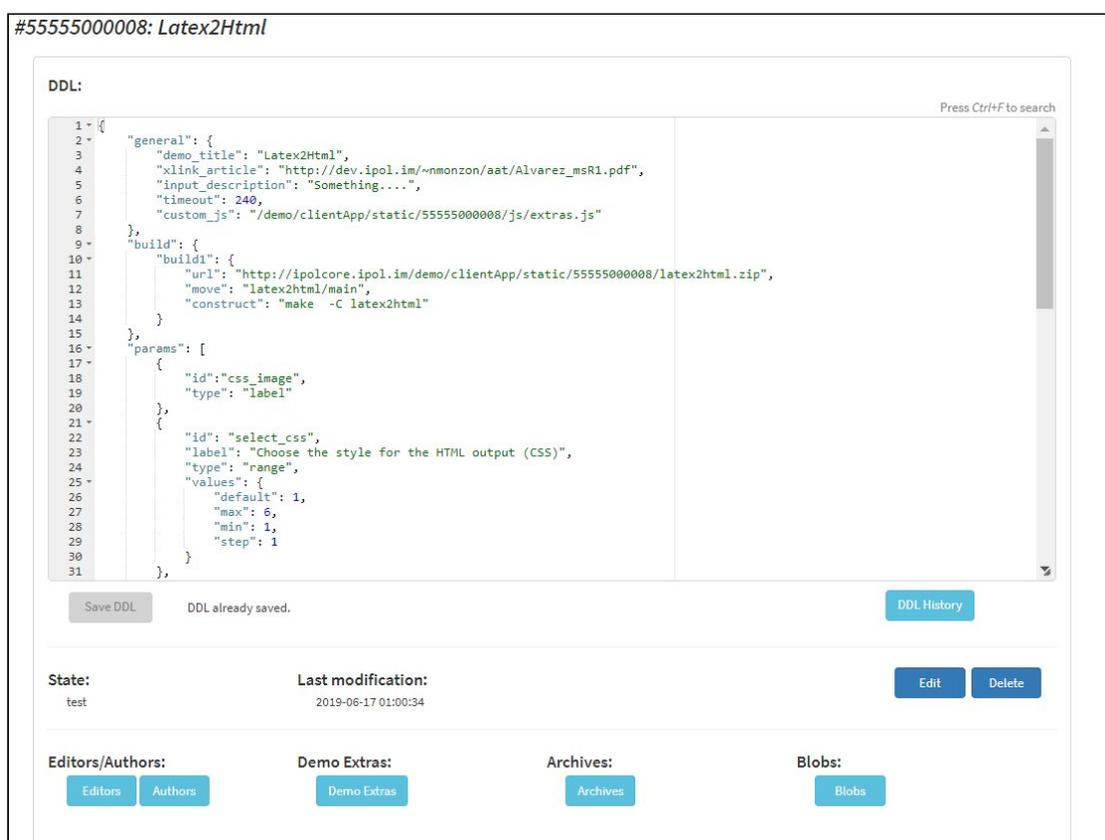
### **Implementación de la demo.**

A continuación describimos las características principales que hacen funcionar la demo online. Vemos en la **Figura 4.33** una captura de la ventana de configuración correspondiente en el panel de control donde se observa las distintas opciones para configurar la demo y el editor en formato JSON dónde se escribe la DDL.

Las demos IPOL se configuran mediante una DDL. En la **Figura 4.34** se muestra la realizada para esta demo en particular. En su sección general se indica su nombre, un enlace desde donde descargar este documento, el tiempo máximo de ejecución y la dirección donde la demo deberá encontrar el código Javascript que controla su interfaz principal. Vemos este código JavaScript en la **Figura 4.35**. Su principal función es asociar a la caja de texto con un fichero LATEX por defecto y asociar los eventos relacionados con el Slider de los CSS con sus representaciones visuales.

Por otro lado, el campo “build” indica dónde está el código en C++ implementado para ese Trabajo Fin de Grado y las instrucciones básicas que necesita la demo para poder compilar y almacenar internamente el ejecutable construido. El campo “params” describe los distintos parámetros que hace uso la demo tanto en su tipo, como rango de valores y etiquetas. El campo “run” indica que se llamará a un *script* auxiliar que recibirá los distintos parámetros en el orden indicado en esa línea. Este se encarga de preparar el entorno de ejecución necesario para posteriormente ejecutar el ejecutable en C++ mediante la línea de comandos requerida. Vemos el contenido de este script en la **Figura 4.36**.

Finalmente el campo “results” indica los resultados que debe visualizar mientras que “archive” indica la información a almacenar en el historial de la demo.



**Figura 4.33:** Página principal del “Panel de control” de la demo.

```

{
  "general": {
    "demo_title": "Latex2Html",
    "xlink_article": "http://dev.ipol.im/~nmonzon/TFT_Christian_Brito",
    "timeout": 240,
    "custom_js": "/demo/clientApp/static/5555500008/js/extras.js"
  },
  "build": {
    "build1": {
      "url":
"http://ipolcore.ipol.im/demo/clientApp/static/5555500008/latex2html.zip",
      "move": "latex2html/main",
      "construct": "make -C latex2html"
    }
  }
}

```

```

"params": [
  {
    "id": "css_image",
    "type": "label"
  },
  {
    "id": "select_css",
    "label": "Choose the style for the HTML output (CSS)",
    "type": "range",
    "values": {
      "default": 1,
      "max": 6,
      "min": 1,
      "step": 1
    }
  },
  {
    "comments": "Choose the language",
    "id": "language",
    "label": "Choose the language",
    "type": "selection_collapsed",
    "default_value": "0",
    "values": {
      "Spanish": "true",
      "English": "false"
    }
  },
  {
    "wrap": false,
    "height": 500,
    "type": "textarea",
    "id": "file",
    "label": "Latex file.",
    "comments": "<b>Write the latex that you want to convert.<b>"
  }
],
"run": "${demoextras}/run.py ${demoextras} $select_css $language ",
"results": [
  {
    "type": "file_download",
    "contents": {
      "html file": "output.html"
    },
    "label": "<b>Download the results:</b>"
  }
],
"archive": {
  "enable_reconstruct": true,
  "archive_always": true,
  "files": {
    "output.html": "Html file"
  },
  "info": {
    "version": "program version",
    "run_time": "run time"
  },
  "params": [
    "select_shape"
  ]
}
}

```

**Figura 4.34:** DDL de configuración de la DEMO.

```

1 demo_id="55555000008"
2
3 // read text from URL location
4 var request = new XMLHttpRequest();
5 request.open('GET', 'https://ipolcore.ipol.im/demo/clientApp/static/55555000008/extras/sample1.tex', true);
6 request.send(null);
7 request.onreadystatechange = function () {
8     if (request.readyState === 4 && request.status === 200) {
9         var type = request.getResponseHeader('Content-Type');
10         string_file = request.responseText;
11     }
12
13     document.getElementById("textarea_file").defaultValue = string_file;
14     updateParamsArrayValue("file", $('#textarea_file').val().replace("\r\n", '\n'));
15
16     css_head = "<img style='width: 312px;height: 174px;;margin:0 auto' src='static/"+demo_id+"/css/"
17     css_tail = ".png'></img>"
18
19     index = $('#range_select_css').val()
20     css = css_head + index + css_tail
21     document.getElementById('css_image').innerHTML = css;
22
23     $('#range_select_css').on('change input', function () {
24         index = $('#range_select_css').val()
25         css = css_head + index + css_tail
26         document.getElementById('css_image').innerHTML = css;
27     })
28
29     $('#number_select_css').on('change input', function () {
30         index = $('#number_select_css').val()
31         css = css_head + index + css_tail
32         document.getElementById('css_image').innerHTML = css;
33     })
34 }

```

Figura 4.35: Código JavaScript asociado a la demo online.

```

1 #!/usr/bin/env python
2 #-*- coding: utf-8 -*-
3 import os, sys
4 import json
5 import shutil
6 -----
7 if __name__ == '__main__':
8
9     demoextras_path = sys.argv[1]
10     select_shape = sys.argv[2]
11     language = sys.argv[3]
12
13     params = open('params.json', 'r')
14     parameters = params.read()
15     params.close()
16     params_dic = json.loads(parameters)
17     file_info = params_dic['file']
18
19     # Prepare input_file from the web interface
20     input_file = 'input.txt'
21     out = open(input_file, 'w')
22     out.write(file_info)
23     out.close()
24
25     css_folder = "{}/http/extras/css".format(demoextras_path)
26     comando = "cp -R {} {}".format(css_folder)
27     os.system(comando)
28
29     utils_folder = "{}/http/extras/utils".format(demoextras_path)
30     comando = "cp -R {} {}".format(utils_folder)
31     os.system(comando)
32
33     template_html_file = "{}/http/extras/template1ES.html".format(demoextras_path)
34     comando = "cp {} {}".format(template_html_file)
35     os.system(comando)
36
37     template_html_file = "{}/http/extras/template1EN.html".format(demoextras_path)
38     comando = "cp {} {}".format(template_html_file)
39     os.system(comando)
40
41     output_file="output.html"
42     string_for_execution = "main {} {} {} {}".format(select_shape, language, input_file, output_file)
43     os.system(string_for_execution)
44
45     if not (os.path.exists(output_file)):
46         error_message = "ERROR: There was a problem during the execution. The result file has not been obtained."
47         with open("demo_failure.txt", "w") as algo_info:
48             algo_info.write(string_for_execution)
49         algo_info.close()
50         sys.exit(0)

```

Figura 4.36: Script Python asociado a la DEMO online.

## 4.3. CREACIÓN DE ESTILOS.

Para la creación de estilos en el formateo de salida del fichero HTML generado por el conversor se han utilizado las hojas de estilo CSS (Cascading Style Sheets). La descripción que se presenta aquí sobre las hojas de estilo CSS se inspira en la referencia [CSS]. La gran ventaja del uso de las hojas de estilo CSS en nuestro proyecto es que permite separar completamente la generación del fichero HTML de la cuestión del formateo visual del documento HTML. Esto hace posible fácilmente ofrecer al usuario diferentes estilos de salida partiendo del mismo fichero HTML.

### 4.3.1. Cascading Style Sheets [CSS]

CSS (siglas en inglés de Cascading Style Sheets), en español "Hojas de estilo en cascada", es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en lenguajes como HTML, XHTMLo XML. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario.

Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS)

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

La separación del formato y el contenido hace posible presentar el mismo documento marcado en diferentes estilos para diferentes métodos de renderizado, como en pantalla, en impresión, en voz (mediante un navegador de voz o un lector de pantalla, y dispositivos táctiles basados en el sistema Braille. También se puede mostrar una página web de manera diferente dependiendo del tamaño de la pantalla o tipo de dispositivo. Los lectores pueden especificar una hoja de estilos diferente, como una hoja de estilos CSS guardado en su computadora, para sobrescribir la hoja de estilos del diseñador.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C). CSS tiene una sintaxis simple y usa un conjunto de palabras clave para especificar los nombres de varias propiedades de estilo.

Una hoja de estilos consiste en una serie de reglas. Cada regla, o conjunto de reglas consisten en uno o más selectores, y un bloque de declaración que determinan la representación visual del fichero HTML en el navegador.

## **4.4. VALIDACIÓN Y PRUEBA.**

Para la validación y prueba de los resultados se ha diseñado un documento de prueba en LATEX que contiene todas las funcionalidades de LATEX que el conversor va a tener en cuenta. Dicho documento se encuentra en el ANEXO I y es el documento que se utiliza por defecto en la implementación en línea del conversor en <http://www.ctim.es/LatexToHTML>.

### **4.4.1. Validación y prueba “off-line”**

Para la validación “off-line” se ha utilizado el documento arriba citado para ir verificando y depurando el código C++ que realiza la conversión. Para cada procedimiento de conversión de los comandos LATEX contemplados se ha verificado su correcto funcionamiento usando este ejemplo. En el caso de que un usuario introduzca un documento que contenga comandos LATEX no contemplados por nuestro conversor, simplemente dichos comandos aparecerán literalmente y sin convertir en el documento HTML generado. En ese caso el usuario deberá hacer una edición manual del documento HTML generado para tener en cuenta esos comandos.

Hay que tener en cuenta que este proyecto se ha realizado una conversión básica de los comandos LATEX, estructuras complejas en LATEX, como por ejemplo tablas que contienen dentro de sus celdas otras tablas pueden fallar. Por supuesto, tener en cuenta todas las posibles variantes de la sintaxis de LATEX está fuera del alcance de un proyecto de fin de grado.

### **4.4.2. Validación “on-line”.**

La validación “on-line” de la aplicación ha consistido en la verificación de que la interfaz de usuario de la DEMO de IPOL funciona correctamente, es decir que tanto la selección de ficheros de estilo CSS como la gestión del idioma elegido (español o inglés) funcionan correctamente y que el fichero HTML generado como salida a partir del texto LATEX incluido por el usuario corresponde correctamente a lo esperado siguiendo las especificaciones de la interfaz.

# CONCLUSIONES Y TRABAJO FUTURO

---

## 5.1. CONCLUSIONES

En este proyecto se ha elaborado un prototipo de conversor en línea de ficheros LATEX a HTML, operativo y disponible a través del enlace [www.ctim.es/LatexToHTML](http://www.ctim.es/LatexToHTML), muy sencillo de usar y que permite al usuario generar páginas web atractivas a través de la selección de hojas de estilo CSS. Esto ha sido posible gracias a la combinación de todo un abanico de tecnologías como son el LATEX, como editor universal de textos científicos, el lenguaje HTML como lenguaje estándar para la publicación de documentos en la web, las hojas de estilos CSS para dar formato al contenido de las páginas web, la programación en C++, para diseñar e implementar el conversor y finalmente el servidor de DEMO's de IPOL que permite la ejecución en línea del conversor sin necesidad de descargarse ningún software localmente.

Los resultados obtenidos por el proyecto, y en particular el prototipo de uso totalmente funcional que se ha implementado nos permite concluir que la idea y diseño original de este proyecto aporta novedades interesantes respecto a otros conversores existentes en la literatura por su facilidad de uso y por permitir al usuario configurar el formato de salida a través de las selección de hojas de estilo CSS dando lugar a un resultado visual más atractivo de la página web.

Dada la enorme implantación del LATEX como editor de textos científicos y la necesidad de publicar estos textos científicos en la web de forma atractiva, este proyecto atiende a una necesidad social principalmente en el ámbito científico/educativo.

## 5.2. TRABAJO FUTURO

Como trabajo futuro se plantea el perfeccionamiento de la herramienta de conversión para que tenga en cuenta una mayor cantidad de comandos LATEX, concretamente un trabajo futuro interesante a abordar sería la incorporación de los siguientes elementos al conversor:

1. La asignación automática de números a las ecuaciones matemáticas y otras etiquetas del documento LATEX, así como la gestión automática de las referencias a dichas dichas fórmulas y otras etiquetas.
2. Tener en cuenta en la conversión de un mayor número de comandos LATEX, así como sus opciones. La variedad en este sentido puede ser enorme.
3. La conversión de estructuras anidadas complejas como por ejemplo tablas que contienen a su vez tablas.
4. Ampliar los estilos CSS disponibles para que el usuario tenga una mayor posibilidad de selección de estilos.

# BIBLIOGRAFÍA

[Arévalo et al] Arévalo, M; Escobar, C; Monasse, P; Monzón, N; Colom, M; (2017). “The IPOL Demo System: A Scalable Architecture of Microservices for Reproducible Research”, Springer International Publishing, pages 3-16, DOI: 10.1007/978-3-319-56414-2\_1

- [CSS] Cascading Style Sheets, [https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada), 2019.
- [CTIM] Centro de Tecnologías de la Imagen, <http://ctim.ulpgc.es/site/>.
- [HTML5] <https://es.wikipedia.org/wiki/HTML5>.
- [IPOL] <https://www.ipol.im>.
- [LaTeX] <https://www.latex-project.org/>.
- [LTX2H] The LaTeX2HTML translator, <https://www.latex2html.org/>, 2018.
- [LTXML] A LaTeX to XML/HTML/MathML Converter, <https://dlnmf.nist.gov/LaTeXML/>, 2019.
- [MathML] Mathematical Markup Language (MathML), <https://es.wikipedia.org/wiki/MathML>, 2015.
- [MathJax] Beautiful math in all browsers (MathJax), <https://www.mathjax.org/>, 2010.
- [Make4ht] Make4ht – A build system for tex4ht, <https://ctan.org/pkg/make4ht>, 2019.
- [MikTeX] <https://miktex.org/>.
- [MToWeb] MathToWeb. Conversor online de ecuaciones en Latex a MathML, [http://www.mathtoweb.com/cgi-bin/mathtoweb\\_home.pl](http://www.mathtoweb.com/cgi-bin/mathtoweb_home.pl), 2013.
- [OLED] Online LATEX equation editor, <https://www.codecogs.com/latex/eqneditor.php>, 2015.
- [OXML] A LaTeX to XML/HTML Webserver, <https://latexml.mathweb.org/editor>, 2019.
- [plasTeX] plasTeX, <https://github.com/tiarno/plastex>, 2018.
- [OVERLeaf] [https://www.overleaf.com/learn/latex/Table\\_of\\_contents](https://www.overleaf.com/learn/latex/Table_of_contents), 2019.
- [TeX4ht] TeX4ht – Convert (La)TeX to HTML/XML, <https://ctan.org/pkg/tex4ht>, 2008.
- [TheTeXBook] <http://www.ctex.org/documents/shredder/src/texbook.pdf>, 1996.
- [T<sub>T</sub>H] T<sub>T</sub>H: the TEX to HTML translator, <http://hutchinson.belmont.ma.us/tth/>, 2017.
- [TeXworks] <http://www.tug.org/texworks/>, 2019.

# Ejemplo de documento LATEX utilizado para validación

---

A continuación se muestra el documento LATEX de ejemplo que ha sido diseñado para validar el correcto funcionamiento del conversor.

```
% definimos el estilo del documento
\documentclass[12pt,a4paper]{article}

% paquetes para gestionar el español y otros caracteres especiales
\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}

% paquete para gestionar imágenes
\usepackage{graphicx}
\usepackage[3D]{movie15}
\usepackage{float}

% paquete para gestionar enlaces web
\usepackage{hyperref}

% gestión de la zona impresa en la página
\oddsidemargin -1.0cm
\headsep -2.4cm
\textwidth=18.5cm
\textheight=26cm

% tamaño del salto entre párrafos
\parskip=3mm

% contador para numerar teoremas y lemas
\newtheorem{teorema}{Teorema}
\newtheorem{lema}{Lema}
\newtheorem{theorem}{Teorema}

% empieza el documento
\begin{document}

% definimos el título
\title{LatexToHtmlOnLine: Conversor en línea de Latex a HTML}
\author{Christian Brito Ramos \ \ ULPGC}
\date{25 de Junio de 2019}
% construimos el título
```

`\maketitle`

`% construimos la tabla de contenidos`

`\tableofcontents`

`\abstract{`

*El objetivo de este proyecto de TFT es implementar un prototipo de conversor en línea de un fichero en formato LATEX para escribir textos científicos a un fichero HTML que se pueda visualizar en un navegador web. Además se da la opción al usuario de elegir el estilo con el que sale formateado el texto HTML. La mayor parte de los conversores en línea que existen en el mercado están diseñados para convertir pequeños trozos de código LATEX (principalmente fórmulas), no admiten la conversión de un texto LATEX entero ni seleccionar estilos. Para conseguir los objetivos planteados ha sido necesario familiarizarse con la sintaxis de LATEX y estudiar las soluciones existentes en la literatura. Para abordar la transformación a HTML se ha implementado un código en C++ que convierte a HTML los principales comandos de LATEX y en base a diferentes estilos diseñados usando CSS se realiza la conversión a HTML. Para la implementación en línea del conversor se usa el constructor de DEMO's en línea de la revista IPOL ("Image Processing OnLine")*

`}`

`\section{Introducci'on}`

*Este documento, escrito originalmente en LATEX, tiene por objeto explicar los diferentes aspectos del formato LATEX que se han abordado durante el desarrollo del TFT y al mismo tiempo servir de documento de ejemplo para realizar la conversión a HTML. El LATEX tiene miles de comandos con múltiples opciones. En este proyecto se han abordado los comandos principales pues un análisis exhaustivo de todo el potencial de LATEX excede en mucho de los objetivos que se pueden abordar en un TFT. En particular ello implica que si el usuario utiliza comandos LATEX no contemplados en este trabajo, su conversión no se realizará correctamente. En cualquier caso, los errores que pueden surgir de la conversión de LATEX a HTML son, en general, fácilmente resolubles editando el documento HTML resultante.*

*En este documento se relacionan los aspectos que han sido abordados para la conversión de LATEX a HTML, pero la explicación detallada de como se ha realizado se encuentra en la memoria del TFT. El LATEX funciona como un lenguaje de programación con un compilador asociado que genera un fichero, por ejemplo un PDF, con el resultado de la compilación. En nuestro caso se ha utilizado el compilador MIKTEX. Todo el diseño del documento LATEX se realiza a partir de comandos. Por ejemplo, para convertir a HTML la organización básica del documento ha sido necesario abordar los siguientes aspectos:*

`\begin{itemize}`

`\item Gestionar el título del documento a través de su correspondiente comando LATEX.`

`\item Gestionar el autor (en LATEX, con frecuencia, la afiliación se gestiona junto con el nombre del autor y así se ha hecho en este trabajo)`

`\item Gestionar la fecha de realización del documento`

`\item Gestionar el resumen (abstract)`

`\item Gestionar el índice de contenidos. Este aspecto no es trivial pues hay que reconocer donde están las diferentes secciones del documento, a partir de ello construir el índice y además añadir enlaces en el índice para facilitar la navegación entre las diferentes secciones.`

`\end{itemize}`

*A continuación, organizado por secciones, se presentarán otros aspectos de LATEX que se han ido abordando a lo largo del desarrollo de este TFT*

`\section{Gestión de párrafos, tipos de letra y caracteres ASCII extendido}`

*La manera de gestionar los cambios de párrafo y de líneas es completamente distinto en LATEX y HTML por ello ha sido necesario realizar una conversión entre los dos lenguajes. El tipo de letra en LATEX se define de forma genérica para todo el texto al principio del documento. Además existen diversos comandos para gestionar su estilo en el interior del documento. En ese sentido se ha*

implementado la conversión a HTML de las letras en negrita (`\bf` ejemplo de texto en negrita) y en itálica (`\em` ejemplo de texto en itálica)

Respecto a los acentos y otros caracteres especiales, si el documento LATEX ha sido realizado usando los paquetes de LATIN existentes aparecerán igual en el documento LATEX que en el HTML, sin embargo si no se usan los paquetes de LATIN cada caracter se escribe en LATEX usando un comando especial irreconocible por HTML, por ello se ha implementado una conversión de estos comandos LATEX a HTML.

```
\section{Gestión de tablas y listas}
```

Para poder pasar las tablas y listas enumeradas y no enumeradas de LATEX a HTML ha sido necesario analizar cuidadosamente la sintaxis en ambos lenguajes (que son muy diferentes) y programar la conversión. A continuación se ponen algunos ejemplos de tablas y listas en LATEX para ilustrar su conversión posterior a HTML

```
\smallskip
```

Ejemplo de tabla:

```
\begin{tabular}[t]{ccc}
```

Nombre & Apellidos & Edad \\

Juan & Garcia López & 22 \\

Pedro & Del Olmo & 43 \\

Agustín & Brito Perez & 30 \\

Laura & Pulido Ruiz & 11 \\

```
\end{tabular}
```

```
\smallskip
```

Ejemplo de lista enumerada:

```
\begin{enumerate}
```

```
\item item 1
```

```
\item item 2
```

```
\item item 3
```

```
\end{enumerate}
```

```
\smallskip
```

Ejemplo de lista no enumerada:

```
\begin{itemize}
```

```
\item item 1
```

```
\item item 2
```

```
\item item 3
```

```
\end{itemize}
```

```
\section{Imágenes y vídeos}
```

Se ha realizado una conversión básica de los comandos LATEX para visualizar imágenes y vídeos. Los ficheros de imágenes y vídeos deben estar en la misma carpeta que el fichero HTML generado. Por tanto, al convertir en línea un fichero LATEX con imágenes y vídeos estos en general no se verán en el navegador de la aplicación en línea pues será necesario que el usuario se descargue el HTML generado y ponga en la carpeta donde están las imágenes y vídeos. A continuación se ponen ejemplos de inserción de imágenes y vídeos.

```
\medskip
```

Ejemplo de inserción de una imagen

```
\includegraphics[width=0.7\linewidth]{eii.jpg}
```

```
\medskip
```

Ejemplo de inserción de un vídeo

```
\begin{figure}[H]
```

```

\centering
\includemovie[
poster,
toolbar, %same as `controls'
text={\includegraphics[width=0.15\textwidth]{penguins.jpg}}
J{3cm}{2cm}{penguinschasingbutterfly.mp4}
\end{figure}

```

```
\section{Fórmulas Matemáticas}
```

El LATEX tiene un potencial enorme para escribir todo tipo de fórmulas matemáticas. Sin embargo su conversión a HTML es un problema complejo. MathML (Mathematical Markup Language) es un lenguaje para escribir ecuaciones matemáticas cuyo objetivo es integrar fórmulas matemáticas en documentos web. Actualmente es parte de HTML5. Las principales limitaciones de MathML son, por un lado, que actualmente no todos los navegadores interpretan correctamente su código y por otro lado que la sintaxis que utiliza es compleja y totalmente diferente a la de LATEX. Afortunadamente dos de las sociedades matemáticas más importantes de los Estados Unidos: la “American Mathematical Society” (AMS) y la “Society for Industrial and Applied Mathematics (SIAM)” han promovido el proyecto Mathjax que es una plataforma de código abierto y JavaScript que permite visualizar en la web fórmulas escritas directamente en la sintaxis de LATEX. Simplemente hay que añadir una directiva al principio del fichero HTML para que se genere una visualización de las fórmulas en formato LATEX. Esta es la solución que se ha adoptado en este trabajo para visualizar en un navegador web las fórmulas matemáticas escritas en LATEX por ser una solución sencilla de implementar y que funciona además en cualquier navegador. A continuación, a título ilustrativo, se escribe una fórmula matemática compleja :

```

\begin{equation}
\label{fórmula}
\int_a^b f(x)dx = \frac{e^{x+1} + C_0}{\sqrt{x^2y^z} + \tan^2(y)} \left( 2\pi i + \frac{\gamma^{\epsilon}}{x} \right)^{\frac{2}{3}}
\end{equation}

```

```
\section{Bibliografía y enlaces}
```

La gestión de la bibliografía es completamente distinta en LATEX y HTML. Se ha optado por una solución sencilla que traslada las etiquetas de los items bibliográficos escritos en LATEX a HTML. De la misma forma los enlaces a páginas web se convierten desde LATEX a HTML para su correcta interpretación por el navegador. A continuación se añaden frases con algunas citas de una pequeña bibliografía que incluye algunos enlaces web:

Existen mucha literatura sobre el uso de LATEX, por ejemplo, \cite{La86}, es una referencia muy usada para ello. Los proyectos que se describen en las referencias \cite{MathML} y \cite{MathJax} tienen por objetivo común la visualización por internet de fórmulas matemáticas.

```
\begin{thebibliography}{1}
```

```
\bibitem{La86} Leslie Lamport {\em LaTeX : A document Preparation System}. Addison-Wesley, 1986.
```

```
\bibitem{MathML} {\em Mathematical Markup Language (MathML)}.\url{https://en.wikipedia.org/wiki/MathML}, 2015.
```

```
\bibitem{MathJax} {\em Beautiful math in all browsers (MathJax)}. \url{https://www.mathjax.org/}, 2010.
```

```
\end{thebibliography}
```

`\section{Comentarios sobre otras funcionalidades de LATEX no implementadas}`

*Como se comentó al principio de este documento el LATEX tiene una enorme variedad de funcionalidades y la consideración de todas excede en mucho el alcance de un TFT. Dentro de las cosas que no se han abordado caben destacar las siguientes*

`\begin{enumerate}`

*\item La asignación automática de números a las ecuaciones matemáticas y otras etiquetas del documento LATEX, así como las referencias cruzadas no se ha realizado.*

*\item La conversión de los comandos LATEX se ha realizado sin contar con todas sus posibles opciones de configuración. La variedad en este sentido puede ser enorme.*

*\item La conversión de estructuras anidadas complejas como por ejemplo tablas que contienen a su vez tablas puede no funcionar.*

*\item En el proceso de conversión se han eliminado los comentarios en LATEX, así como algunos de los comandos LATEX que no se gestionan para que no aparezcan en el documento HTML.*

`\end{enumerate}`

*La inclusión de nuevas funcionalidades de LATEX a este prototipo, como las arriba mencionadas u otras es un interesante trabajo futuro a realizar.*

`\end{document}`

Por último, a título ilustrativo, en las páginas siguientes se muestra el resultado de la compilación de este documento LATEX utilizando el compilador MikTEX.

# LatexToHtmlOnLine: Conversor en línea de Latex a HTML

Christian Brito Ramos  
ULPGC

25 de Junio de 2019

## Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Gestión de párrafos, tipos de letra y caracteres ASCII extendido</b>	<b>2</b>
<b>3. Gestión de tablas y listas</b>	<b>2</b>
<b>4. Imágenes y vídeos</b>	<b>3</b>
<b>5. Fórmulas Matemáticas</b>	<b>3</b>
<b>6. Bibliografía y enlaces</b>	<b>4</b>
<b>7. Comentarios sobre otras funcionalidades de LATEX no implementadas</b>	<b>4</b>

## Resumen

El objetivo de este proyecto de TFT es implementar un prototipo de conversor en línea de un fichero en formato LATEX para escribir textos científicos a un fichero HTML que se pueda visualizar en un navegador web. Además se da la opción al usuario de elegir el estilo con el que sale formateado el texto HTML. La mayor parte de los conversores en línea que existen en el mercado están diseñados para convertir pequeños trozos de código LATEX (principalmente fórmulas), no admiten la conversión de un texto LATEX entero ni seleccionar estilos. Para conseguir los objetivos planteados ha sido necesario familiarizarse con la sintaxis de LATEX y estudiar las soluciones existentes en la literatura. Para abordar la transformación a HTML se ha implementado un código en C++ que convierte a HTML los principales comandos de LATEX y en base a diferentes estilos diseñados usando CSS se realiza la conversión a HTML. Para la implementación en línea del conversor se usa el constructor de DEMO's en línea de la revista IPOL ("Image Processing OnLine")

## 1. Introducción

Este documento, escrito originalmente en LATEX, tiene por objeto explicar los diferentes aspectos del formato LATEX que se han abordado durante el desarrollo del TFT y al mismo tiempo

servir de documento de ejemplo para realizar la conversión a HTML. El LATEX tiene miles de comandos con múltiples opciones. En este proyecto se han abordado los comandos principales pues un análisis exhaustivo de todo el potencial de LATEX excede en mucho de los objetivos que se pueden abordar en un TFT. En particular ello implica que si el usuario utiliza comandos LATEX no contemplados en este trabajo, su conversión no se realizará correctamente. En cualquier caso, los errores que pueden surgir de la conversión de LATEX a HTML son, en general, fácilmente resolubles editando el documento HTML resultante.

En este documento se relacionan los aspectos que han sido abordados para la conversión de LATEX a HTML, pero la explicación detallada de como se ha realizado se encuentra en la memoria del TFT. El LATEX funciona como un lenguaje de programación con un compilador asociado que genera un fichero, por ejemplo un PDF, con el resultado de la compilación. En nuestro caso se ha utilizado el compilador MIKTEX. Todo el diseño del documento LATEX se realiza a partir de comandos. Por ejemplo, para convertir a HTML la organización básica del documento ha sido necesario abordar los siguientes aspectos:

- Gestionar el título del documento a través de su correspondiente comando LATEX.
- Gestionar el autor (en LATEX, con frecuencia, la afiliación se gestiona junto con el nombre del autor y así se ha hecho en este trabajo)
- Gestionar la fecha de realización del documento
- Gestionar el resumen (abstract)
- Gestionar el índice de contenidos. Este aspecto no es trivial pues hay que reconocer donde están las diferentes secciones del documento, a partir de ello construir el índice y además añadir enlaces en el índice para facilitar la navegación entre las diferentes secciones.

A continuación, organizado por secciones, se presentarán otros aspectos de LATEX que se han ido abordando a lo largo del desarrollo de este TFT

## 2. Gestión de párrafos, tipos de letra y caracteres ASCII extendido

La manera de gestionar los cambios de párrafo y de líneas es completamente distinto en LATEX y HTML por ello ha sido necesario realizar una conversión entre los dos lenguajes. El tipo de letra en LATEX se define de forma genérica para todo el texto al principio del documento. Además existen diversos comandos para gestionar su estilo en el interior del documento. En ese sentido se ha implementado la conversión a HTML de las letras en negrita (**ejemplo de texto en negrita**) y en itálica (*ejemplo de texto en itálica*)

Respecto a los acentos y otros caracteres especiales, si el documento LATEX ha sido realizado usando los paquetes de LATIN existirán igual en el documento LATEX que en el HTML, sin embargo si no se usan los paquetes de LATIN cada caracter se escribe en LATEX usando un comando especial irreconocible por HTML, por ello se ha implementado una conversión de estos comandos LATEX a HTML.

## 3. Gestión de tablas y listas

Para poder pasar las tablas y listas enumeradas y no enumeradas de LATEX a HTML ha sido necesario analizar cuidadosamente la sintaxis en ambos lenguajes (que son muy diferentes) y programar la conversión. A continuación se ponen algunos ejemplos de tablas y listas en LATEX para ilustrar su conversión posterior a HTML

Ejemplo de tabla:

$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$

Ejemplo de lista enumerada:

1. item 1
2. item 2
3. item 3

Ejemplo de lista no enumerada:

- item 1
- item 2
- item 3

## 4. Imágenes y vídeos

Se ha realizado una conversión básica de los comandos LATEX para visualizar imágenes y vídeos. Los ficheros de imágenes y vídeos deben estar en la misma carpeta que el fichero HTML generado. Por tanto, al convertir en línea un fichero LATEX con imágenes y vídeos estos en general no se verán en el navegador de la aplicación en línea pues será necesario que el usuario se descargue el HTML generado y ponga en la carpeta donde están las imágenes y vídeos. A continuación se ponen ejemplos de inserción de imágenes y vídeos.

Ejemplo de inserción de una imagen



Ejemplo de inserción de un vídeo



## 5. Fórmulas Matemáticas

El LATEX tiene un potencial enorme para escribir todo tipo de fórmulas matemáticas. Sin embargo su conversión a HTML es un problema complejo. MathML (Mathematical Markup Language) es un lenguaje para escribir ecuaciones matemáticas cuyo objetivo es integrar fórmulas matemáticas en documentos web. Actualmente es parte de HTML5. Las principales limitaciones de MathML son, por un lado, que actualmente no todos los navegadores interpretan correctamente su código y por otro lado que la sintaxis que utiliza es compleja y totalmente diferente a la de LATEX. Afortunadamente dos de las sociedades matemáticas más importantes de los Estados Unidos: la “American

Mathematical Society” (AMS) y la “Society for Industrial and Applied Mathematics (SIAM)” han promovido el proyecto Mathjax que es una plataforma de código abierto y JavaScript que permite visualizar en la web fórmulas escritas directamente en la sintaxis de LATEX. Simplemente hay que añadir una directiva al principio del fichero HTML para que se genere una visualización de las fórmulas en formato LATEX. Esta es la solución que se ha adoptado en este trabajo para visualizar en un navegador web las fórmulas matemáticas escritas en LATEX por ser una solución sencilla de implementar y que funciona además en cualquier navegador. A continuación, a título ilustrativo, se escribe una fórmula matemática compleja :

$$\int_a^b f(x)dx = \frac{e^{x+1} + C_0}{\sqrt{x^{y^z} + \tan^2(y)}} \left( 2\pi i + \frac{\gamma^\epsilon}{x} \right)^{\frac{2}{3}} \quad (1)$$

## 6. Bibliografía y enlaces

La gestión de la bibliografía es completamente distinta en LATEX y HTML. Se ha optado por una solución sencilla que traslada las etiquetas de los items bibliográficos escritos en LATEX a HTML. De la misma forma los enlaces a páginas web se convierten desde LATEX a HTML para su correcta interpretación por el navegador. A continuación se añaden frases con algunas citas de una pequeña bibliografía que incluye algunos enlaces web:

Existen mucha literatura sobre el uso de LATEX, por ejemplo, [1], es una referencia muy usada para ello. Los proyectos que se describen en las referencias [2] y [3] tienen por objetivo común la visualización por internet de fórmulas matemáticas.

## Referencias

- [1] Leslie Lamport *LaTeX : A document Preparation System*. Addison-Wesley, 1986.
- [2] *Mathematical Markup Language (MathML)* .<https://en.wikipedia.org/wiki/MathML>, 2015.
- [3] *Beautiful math in all browsers (MathJax)* .<https://www.mathjax.org/>, 2010.

## 7. Comentarios sobre otras funcionalidades de LATEX no implementadas

Como se comentó al principio de este documento el LATEX tiene una enorme variedad de funcionalidades y la consideración de todas excede en mucho el alcance de un TFT. Dentro de las cosas que no se han abordado caben destacar las siguientes

1. La asignación automática de números a las ecuaciones matemáticas y otras etiquetas del documento LATEX, así como las referencias cruzadas no se ha realizado.
2. La conversión de los comandos LATEX se ha realizado sin contar con todas sus posibles opciones de configuración. La variedad en este sentido puede ser enorme.
3. La conversión de estructuras anidadas complejas como por ejemplo tablas que contienen a su vez tablas puede no funcionar.
4. En el proceso de conversión se han eliminado los comentarios en LATEX, así como algunos de los comandos LATEX que no se gestionan para que no aparezcan en el documento HTML.

La inclusión de nuevas funcionalidades de LATEX a este prototipo, como las arriba mencionadas u otras es un interesante trabajo futuro a realizar.