# Memoria del proyecto. (Versión pública)

27 de abril de 2021

# Índice de figuras

2.1. Imagen descriptiva de Scrum. . . . . 6
2.2. Primera imagen del diagrama de Gantt . . . . . 8
2.3. Segunda imagen del diagrama de Gantt . . . . . 8
2.4. Tercera imagen del diagrama de Gantt . . . . . 9
2.5. Cuarta imagen del diagrama de Gantt . . . . . 9
2.6. Quinta imagen del diagrama de Gantt . . . . . 9

3.1. Los primeros 7 casos de uso . . . . . 29
3.2. Los siguientes casos de uso . . . . . 30
3.3. Modelo conceptual de datos . . . . . 40

4.1. Diagrama de despliegue . . . . . 43
4.2. Arquitectura lógica de la plataforma . . . . . 44
4.3. Wireframe para el estado inicial . . . . . 47
4.4. Wireframe para la conversión a PDF . . . . . 48
4.5. Wireframe para la conversión a lenguaje intermedio . . . . . 49
4.6. Wireframe para el menú . . . . . 50
4.7. Wireframe para el menú en móvil . . . . . 51
4.8. Página inicial del prototipo de alta fidelidad. . . . . 52

4.9. Página de ancho reducido del prototipo de alta fidelidad. . . . 53

4.10. Página con menú del prototipo de alta fidelidad. . . . . 54

6.1. Prueba completa de validador HTML5 de W3C. . . . . 68

6.2. Resultado del validador de CSS3 de W3C. . . . . 68

8.1. Ratón por encima del desplegable de los títulos. . . . . 78

8.2. Resultado de cambiar la línea a un nivel de título. . . . . 79

8.3. Ratón encima del botón de lista . . . . . 80

8.4. Lista insertada adecuadamente . . . . . 81

8.5. Ratón encima del botón de enlaces . . . . . 82

8.6. Ratón encima del botón de enlaces . . . . . 83

8.7. Ventana emergente que pide una URL al usuario . . . . . 83

8.8. Ratón encima del botón de insertar tabla. . . . . 84

8.9. El diálogo emergente de alto y ancho. . . . . 85

8.10. La tabla nueva completamente insertado . . . . . 86

8.11. Ratón encima del botón de insertar imagen. . . . . 87

8.12. Diálogo emergente del sistema de archivos del dispositivo. . . 88

8.13. Diálogo emergente para texto alternativo de la imagen. . . . . 88

8.14. Logo de la UCA mostrándose adecuadamente. . . . . 89

8.15. Ratón encima del botón de insertar contenedor. . . . . 90

8.16. Diálogo emergente para escribir la clase del contenedor. . . . 91

8.17. Contenedor visible con el color gris que separa los bloques entre sí. . . . . 92

8.18. Ratón encima del botón de la metaetiqueta . . . . . 93

8.19. Metaetiqueta insertada en el texto como se puede ver en la fuente de letra distinta en la línea. . . . . 94

8.20. Ratón encima del botón de crear PDF. . . . . 95

8.21. PDF con el texto del editor añadido. . . . . 96

8.22. Ratón encima del botón de la conversión al lenguaje intermedio. 97

8.23. Conversión del editor visual a modo código. . . . . 98

8.24. Conversión del editor en modo código a modo visual. . . . . 99

8.25. Ratón encima del botón de compartir de Nibis. . . . . 100

8.26. Pantalla inicial para compartir. . . . . 101

8.27. Pantalla para mostrar el enlace que el usuario puede compartir. 102

# Índice de cuadros


2.1. Costes de los recursos del proyecto . . . . . 12

2.2. Costes totales del proyecto . . . . . 12

2.3. Riesgos posibles en el proyecto . . . . . 13

3.1. RF-01 (Creación de documentos PDF a partir del texto introducido) . . . . . 16

3.2. RF-02 (Conversión de HTML a lenguaje intermedio.) . . . . . 16

3.3. RF-03 (Conversión de lenguaje intermedio a HTML.) . . . . . 16

3.4. RF-04 (Conversión de lenguaje intermedio a HTML.) . . . . . 17

3.5. RF-05 (Creación de tablas en el editor.) . . . . . 17

3.6. RF-06 (Inserción de imágenes en el editor.) . . . . . 17

3.7. RF-07 (Inserción de enlaces en el editor.) . . . . . 18

3.8. RF-08 (Inserción de listas desordenadas en el editor.) . . . . . 18

3.9. RF-09 (Inserción de títulos en el editor.) . . . . . 18

3.10. RF-10 (Inserción de contenedores en el editor.) . . . . . 19

3.11. RF-11 (Incrustar código.) . . . . . 19

3.12. RF-12 (Texto sin comprobar.) . . . . . 20

3.13. RF-13 (Guardar texto del editor.) . . . . . 21

3.14. RF-14 (Cargar texto del editor.) . . . . . 21

3.15. RF-15 (Cargar editor a partir de archivo guardado.) . . . . . 22
3.16. RNF-01 (Usabilidad.) . . . . . 23
3.17. RNF-02 (Portabilidad.) . . . . . 24
3.18. RNF-03 (Eficiencia.) . . . . . 24
3.19. RNF-04 (Seguridad.) . . . . . 25
3.20. RNF-05 (Interoperabillidad.) . . . . . 25
3.21. RNF-06 (Mantenibilidad.) . . . . . 26
3.22. RINF-01 (Texto en HTML.) . . . . . 27
3.23. RINF-02 (Texto en lenguaje intermedio.) . . . . . 27
3.24. Usuario . . . . . 28
3.25. Tiempo . . . . . 28
3.26. CU-01 (Caso de uso para insertar texto literal) . . . . . 31
3.27. CU-02 (Caso de uso para cargar documento) . . . . . 32
3.28. CU-03 (Caso de uso para insertar tabla) . . . . . 33
3.29. CU-04 (Caso de uso para insertar imagen) . . . . . 34
3.30. CU-05 (Caso de uso para insertar enlace) . . . . . 35
3.31. CU-06 (Caso de uso para insertar lista) . . . . . 35
3.32. CU-07 (Caso de uso seleccionar archivo) . . . . . 36
3.33. CU-08 (Caso de uso para convertir a HTML) . . . . . 36
3.34. CU-09 (Caso de uso para convertir a lenguaje intermedio) . . 37
3.35. CU-10 (Caso de uso para exportar a PDF) . . . . . 38
3.36. CU-11 (Caso de uso para insertar código) . . . . . 38
3.37. CU-12 (Caso de uso para guardar texto del editor) . . . . . 39
3.38. CU-13 (Caso de uso para cargar texto del editor) . . . . . 39

6.1. PF-01 (Nueva línea.) . . . . . 60
6.2. PF-02 (Eliminar línea.) . . . . . 60
6.3. PF-03 (Insertar tabla.) . . . . . 60
6.4. PF-04 (Insertar enlace.) . . . . . 61
6.5. PF-05 (Insertar lista.) . . . . . 61
6.6. PF-06 (Insertar contenedor.) . . . . . 61
6.7. PF-07 (Usar la metaetiqueta) . . . . . 62
6.8. PF-08 (Usar la etiqueta de incrustar código) . . . . . 62
6.9. PF-09 (Insertar un título) . . . . . 62
6.10. PF-10 (Conversiones del editor vacío) . . . . . 63
6.11. PF-11 (Conversión del editor con una sola línea) . . . . . 64
6.12. PF-12 (Conversión del editor con una sola línea) . . . . . 64
6.13. PF-13 (Conversión de una imagen vacía) . . . . . 64
6.14. PF-14 (Conversión de una lista dentro de una lista) . . . . . 65
6.15. PF-15 (Conversión de una lista dentro de una lista) . . . . . 65
6.16. PF-16 (Conversión de una lista dentro de una lista) . . . . . 65
6.17. PF-17 (Inserción de una tabla por atajo de teclado) . . . . . 66
6.18. PF-18 (Inserción de una tabla por atajo de teclado) . . . . . 66
6.19. PF-19 (Inserción de una tabla por atajo de teclado) . . . . . 66
6.20. PF-20 (Inserción de un contenedor por atajo de teclado) . . . 67
6.21. PF-21 (Inserción de la metaetiqueta por atajo de teclado) . . 67
6.22. PF-22 (Inserción del código incrustado por atajo de teclado) . 67

# Índice general


**I Prolegómeno** **1**

**1. Introducción** **2**

1.1. Motivación . . . . . 2

1.2. Objetivos . . . . . 3

1.3. Mercado Actual . . . . . 3

1.4. Organización del documento . . . . . 4

**2. Planificación** **5**

2.1. Metodología de desarrollo . . . . . 5

2.2. Planificación del proyecto . . . . . 6

2.2.1. Primera iteración . . . . . 6

2.2.2. Segunda iteración . . . . . 6

2.2.3. Tercera iteración . . . . . 7

2.2.4. Cuarta iteración . . . . . 7

2.2.5. Quinta Iteración . . . . . 7

2.2.6. Sexta iteración . . . . . 7

2.2.7. Séptima iteración . . . . . 7

2.2.8. Documentación . . . . . 7

2.2.9. Diagrama de Gantt . . . . . 8
2.3. Organización . . . . . 10
2.3.1. Roles y Responsabilidad . . . . . 10
2.3.2. Recursos y Herramientas . . . . . 10
2.4. Costes . . . . . 11
2.5. Gestión de riesgos . . . . . 12

**II Desarrollo** **14**

**3. Análisis del sistema** **15**

3.1. Catálogo de requisitos . . . . . 15
3.1.1. Requisitos funcionales . . . . . 15
3.1.2. Requisitos no funcionales . . . . . 23
3.1.3. Requisitos de información . . . . . 27
3.2. Casos de uso . . . . . 28
3.2.1. Actores . . . . . 28
3.2.2. Diagramas de caso de uso . . . . . 29
3.2.3. Descripción de los casos de uso . . . . . 31
3.3. Modelo conceptual de datos . . . . . 40
3.3.1. Entidades del modelo conceptual de datos . . . . . 40

**4. Diseño del sistema** **42**

4.1. Arquitectura del sistema . . . . . 42
4.1.1. Arquitectura física . . . . . 42
4.1.2. Arquitectura lógica . . . . . 43

4.2. Patrones de diseño . . . . . 45
4.2.1. Tooltip . . . . . 45
4.2.2. Atajos de teclado . . . . . 45
4.2.3. Arquitectura dirigida por eventos . . . . . 45
4.2.4. Convenciones de código . . . . . 45
4.3. Diseño de la interfaz de usuario . . . . . 46
4.3.1. Wireframes . . . . . 46
4.3.2. Prototipo de alta fidelidad . . . . . 52

**5. Implementación del sistema** **55**
5.1. Entorno de desarrollo . . . . . 55
5.1.1. Emacs . . . . . 55
5.1.2. Herramientas adicionales . . . . . 56

**6. Pruebas del sistema** **58**
6.1. Entorno de pruebas . . . . . 58
6.1.1. Entorno de pruebas del servidor . . . . . 58
6.1.2. Entorno de prueba del cliente . . . . . 58
6.1.3. Herramientas utilizadas . . . . . 59
6.1.4. Pruebas funcionales . . . . . 59
6.2. Calidad de producto . . . . . 68
6.3. Evaluación con usuarios . . . . . 68
6.3.1. Tareas del usuario . . . . . 69
6.3.2. Cuestionario post-test . . . . . 69
6.3.3. Resultados obtenidos . . . . . 70

**III Epílogo** **72**

**7. Manual de instalación e explotación** **73**

7.1. Introducción . . . . . 73

7.2. Requisitos previos . . . . . 73

7.2.1. Inventario de componentes . . . . . 74

7.3. Procedimiento de instalación . . . . . 74

7.3.1. Instalación de la API . . . . . 74

7.3.2. Instalación de servidor TogetherJS . . . . . 75

7.3.3. Inicio de la API . . . . . 75

7.3.4. Configuración del cliente . . . . . 76

7.3.5. Configuración automatizada . . . . . 76

**8. Manual de usuario** **77**

8.1. Características . . . . . 77

8.2. Requisitos previos . . . . . 77

8.3. Uso del sistema . . . . . 78

8.3.1. Insertar título en el editor . . . . . 78

8.3.2. Insertar lista en el editor . . . . . 79

8.3.3. Insertar enlace en el editor . . . . . 82

8.3.4. Insertar tabla en el editor . . . . . 84

8.3.5. Insertar imagen en el editor . . . . . 87

8.3.6. Insertar contenedor en el editor . . . . . 90

8.3.7. Uso de la metaetiqueta en el editor . . . . . 93

8.3.8. Exportar a PDF . . . . . 95

8.3.9. Conversión a lenguaje intermedio . . . . . 97
8.3.10. Compartir el documento . . . . . 100

**9. Conclusiones** **103**

9.0.1. Objetivos alcanzados . . . . . 103
9.0.2. Lecciones aprendidas . . . . . 104
9.0.3. Trabajo futuro . . . . . 105

**IV Bibliografía** **107**

**V Licencia** **111**

**GNU Free Documentation License** **112**

1. APPLICABILITY AND DEFINITIONS . . . . . 112
2. VERBATIM COPYING . . . . . 114
3. COPYING IN QUANTITY . . . . . 114
4. MODIFICATIONS . . . . . 115
5. COMBINING DOCUMENTS . . . . . 117
6. COLLECTIONS OF DOCUMENTS . . . . . 117
7. AGGREGATION WITH INDEPENDENT WORKS . . . . . 118
8. TRANSLATION . . . . . 118
9. TERMINATION . . . . . 119
10. FUTURE REVISIONS OF THIS LICENSE . . . . . 119
11. RELICENSING . . . . . 119
ADDENDUM: How to use this License for your documents . . . . 120

# Agradecimientos

A mis padres porque sin su apoyo diario y detalles no sería ni un cuarto de lo que soy hoy. Saber aguantar tantas sobremesas que versaban de los problemas de este proyecto tiene mérito.

A mi hermano porque ha sido y es mi profesor, consejero y atento oyente de todo lo que me pasa. Al que se le ha echado en falta todos estos meses que ha estado fuera. Se nota que estaba acostumbrado a las buenas compañías.

A Pablo por ser el mejor cuidador de pollos y la única persona capaz de crear la paradoja de estar más cerca que muchas personas que veo diariamente a pesar de vivir lejos.

A todos los amigos que he conocido estos años, los que son capaces de hacerte entrar a un examen con una risa imposible de ocultar. Sin ellos no hubiera merecido la pena estudiar esta carrera.

A Manuel y Salva por ayudarme a llevar a cabo esta idea y por animarme a hacer un proyecto tan complejo como este.

Sin olvidarme de todas las personas que a través de distintas organizaciones y desde la mayor humanidad me han enseñado que la comunidad del software libre reune a gente siempre dispuesta a ayudar.

Una última mención a Dani, Emilio, Pablo, Fer, entre otros los cuales participaron en las pruebas con usuarios y que pusieron a prueba tanto al sistema como a mi.

## Resumen

Actualmente se requieren documentos para todo tipo de tareas. Ya sean estos un Curriculum Vitae, una queja formal, un trámite para la administración pública, documentación de proyectos y manuales de usuarios. A pesar de esta situación, no hay herramientas para generar documentos de tipo profesional para dispositivos móviles. Por otro lado, hay múltiples herramientas para los sistemas de escritorio que pueden ser demasiado complejas para personas con conocimientos básicos.

Nibis es una plataforma web con un editor de texto visual (WYSIWYG) accesible desde cualquier dispositivo que sirve para generar documentos de caracter profesional o informal de forma sencilla. Para ello han participado usuarios en el desarrollo del proyecto para comprobar la usabilidad de la herramienta.



Palabras clave: Conversión, Editor de texto, Colaborativo, Multiplataforma, Web, Lenguaje de marcas

# Índice general

# Índice de cuadros

# Índice de figuras

# Parte I

# Prolegómeno

# Capítulo 1

# Introducción

Desde el principio de la década los servicios web están tomando un claro protagonismo. No solo permiten un acceso ubicuo a las herramientas, también han mejorado la accesibilidad y han acercado a los usuarios lo que antes solo unos pocos podían conseguir.

## 1.1. Motivación

La motivación principal era hacer más agradable y sencillo para el usuario el uso de lenguajes de etiquetados complejos tales como LaTeX. Esto favorece que los usuarios con poca costumbre a estas herramientas se adapten a ellas mucho antes y puedan centrarse en generar documentos de mayor calidad en el mismo o en menor tiempo.

A su vez, un usuario medio no puede prestar todo el tiempo necesario para aprender LaTeX, ya que este tiene muchas etiquetas y tiene gran cantidad de problemas respecto a la usabilidad y la accesibilidad para todas las personas que pueden necesitar hacer un documento con las posibilidades que permite LaTeX[2].

Uno de los problemas más usuales de escribir estos documentos usando LaTeX es utilizar todas las herramientas necesarias para que sea fácil de usar. Por ello este proyecto integra todas las utilidades necesarias para poder escribir. Desde el editor hasta el proceso de conversión automatizado.

## 1.2. Objetivos

Esta sección resume los principales objetivos del proyecto de Nibis. Los objetivos son

- Simplificar el proceso de creación de un documento para ámbitos profesionales y universitarios así como apuntes rápidos.
- Hacer un editor lo más accesible y usable posible
- Optimizar y mejorar las conversiones entre los lenguajes de marcas propuestos
- Promover el uso de herramientas de código abierto y aprender más sobre las que sean necesarias para este proyecto
- Conocer más de cerca el funcionamiento interno de parte de los navegadores e intérpretes de Javascript.
- Servir de base para un editor mucho más completo compatible con dispositivos de gama baja.

## 1.3. Mercado Actual

Si se enfoca como un procesador de texto más, hay gran variedad de paquetes ofimáticos y servicios web que tienen una inmensa cantidad de funcionalidades y de gran calidad. Entre ellos tenemos a Microsoft Office e iWork. Si tenemos en cuenta los objetivos de este proyecto podemos centrarnos en otros:

- Overleaf es la plataforma más usada para realizar documentos profesionales y técnicos. Tiene una buena documentación y permite compartir plantillas y documentos fácilmente. Sin embargo, también tiene sus deficiencias como la imposibilidad de hacer el documento sin conexión o una interfaz algo más sobrecargada de lo recomendable.
- Lyx es la aplicación que se suele recomendar para las personas que no tienen conocimientos previos sobre LaTeX o que encuentran algunas dificultades para conocerlo. Es muy útil para personas con poca experiencia pero no es tan práctico como Overleaf y el usuario necesita tener todas las dependencias del programa que rondan los 2GB de almacenamiento.

Respecto a Overleaf, Nibis proporciona un sistema que no necesita una conexión a Internet para seguir funcionando. Esto es una ventaja para usuarios con problemas de conexión y reduce la carga extra del sistema debido a la continua comunicación con el servidor central.

Frente a Lyx, Nibis presenta un sistema multiplataforma tanto en escritorio como en dispositivos móviles. No requiere instalar ninguna dependencia en el sistema del usuario más allá de un navegador y posee una interfaz que ha sido probada con usuarios.

## 1.4. Organización del documento

En este apartado se describirán brevemente cada sección de la memoria.

- Introducción. Este capítulo contiene los motivos principales por los que este proyecto se ha realizado y una comparación con los ya existentes.
- Planificación. El capítulo abarca la metodología y gestión de los recursos para el proyecto.
- Análisis del sistema. El capítulo recoge las descripciones de los requisitos funciones y no funcionales, requisitos de información, reglas de negocio del sistema y los actores implicados en el sistema.
- Diseño de sistema En este punto se expone el diseño que se ha planificado para el sistema de los diferentes componentes y la interfaz del usuario.
- Implementación del sistema. En este punto se detalla el desarrollo que se ha llevado a cabo para el proyecto.
- Pruebas del sistema. En este punto se exponen las pruebas realizadas al sistema para garantizar su correcto funcionamiento.
- Manual de instalación y explotación. En este punto se detalla como se debe instalar toda la plataforma
- Manual de usuario. En el manual de usuario se explica detalladamente como se debe utilizar el sistema, además se hace también una recopilación de las principales funcionalidades del sistema.
- Conclusiones: En este último punto se expone una reflexión sobre el desarrollo de este proyecto, las posibles mejoras futuro a desarrollar, las lecciones aprendidas y las habilidades adquiridas.

# Capítulo 2

# Planificación

## 2.1. Metodología de desarrollo

La metodologia seguida en este proyecto ha sido la metodología ágil.

Esta metodología consiste en la subdivisión del proyecto en fases o iteraciones. Tras cada iteración se obtendrá un software funcional que podrá ser probado. En cada iteración se detallan unos objetivos pendientes que hay que cumplir y junto a los tutores se proponen los objetivos de la siguiente iteración después de terminar la anterior[14].

Esta metodología es una generalización de una gran variedad de ellas como XP [5] o Scrum [6]. Se ha elegido Scrum por la familiaridad que se tiene tras tantos años usando esta metodología. Las reuniones permiten comprobar el estado del proyecto y pone de manifiesto los fallos principales.

Las iteraciones del proyecto comienzan tras una breve conversación con los tutores en la que se determinan las tareas pendientes de realizar. Para comenzar una iteración hace falta terminar la anterior. Esto permite gestionar correctamente las tareas e incluso limitar el número de tareas máximas y considerar todas las tareas como necesarias. Las iteraciones pueden variar de carga de trabajo dependiendo de las necesidades, favoreciendo el trabajo en los momentos más productivos y reduciéndolo cuando no sea posible. El tiempo entre cada reunión dependía de las tareas ya que puede encontrarse un problema o una deuda técnica, pero como mínimo era necesario una reunión cada dos semanas.

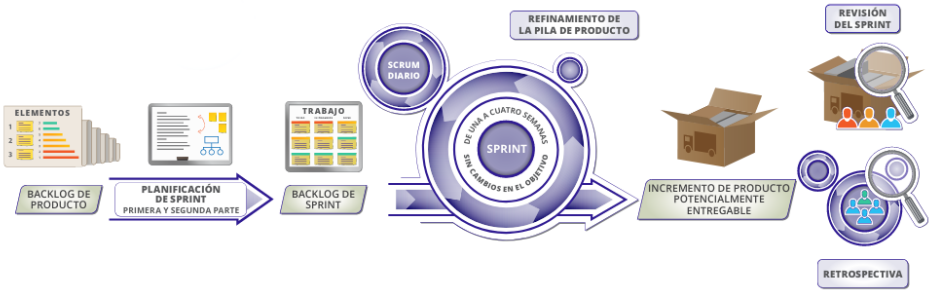



Figura 2.1: Imagen descriptiva de Scrum.

## 2.2. Planificación del proyecto

El proyecto tiene una duración estimada de 7 meses, desde Octubre de 2020 hasta Abril de 2021. Durante este tiempo se ha trabajado en varias iteraciones para tener un producto funcional al final de cada una de ellas.

Cada una de estas iteraciones comienza con un intercambio de mensajes, normalmente correos, entre el alumno y los tutores del proyecto para concretar los objetivos a cumplir durante la iteración, así como los requisitos que el sistema debe cumplir al final de la iteración. A continuación describimos las diferentes iteraciones.

### 2.2.1. Primera iteración

Durante esta primera iteración fue el diseño inicial de todo el sistema y la toma de contacto con el tutor y el cotutor. Esta iteración requirió una profunda investigación sobre todas las posibilidades que permitía el navegador y recopilar las herramientas que serían imprescindibles para todo el desarrollo.

### 2.2.2. Segunda iteración

Definición de los requisitos funcionales principales para el sistema, comprobación de la iteración anterior, definición esquemática de las distintas conversiones a partir del documento original.

### 2.2.3. Tercera iteración

Tras la investigación durante las iteraciones anteriores se empieza a implementar el editor y se definen las funciones del mismo. Se comprueban las definiciones de la anterior iteración y se desarrollan los primeros algoritmos de conversión.

### 2.2.4. Cuarta iteración

Definición más precisa e investigación sobre la API REST del proyecto. Comprobación del estado del editor e indagar sobre las mejoras de usabilidad que necesite. Se incorporan los algoritmos pendientes del proyecto.

### 2.2.5. Quinta Iteración

Desarrollo de la API REST y optimización de los algoritmos para dispositivos con pocos recursos hardware. Se introduce la posibilidad de plantillas para el usuario.

### 2.2.6. Sexta iteración

Selección de funcionalidades principales y sus pruebas pertinentes en QUnit. Se mejora de la interfaz gráfica para hacerla más vistosa y usable en móviles.

### 2.2.7. Séptima iteración

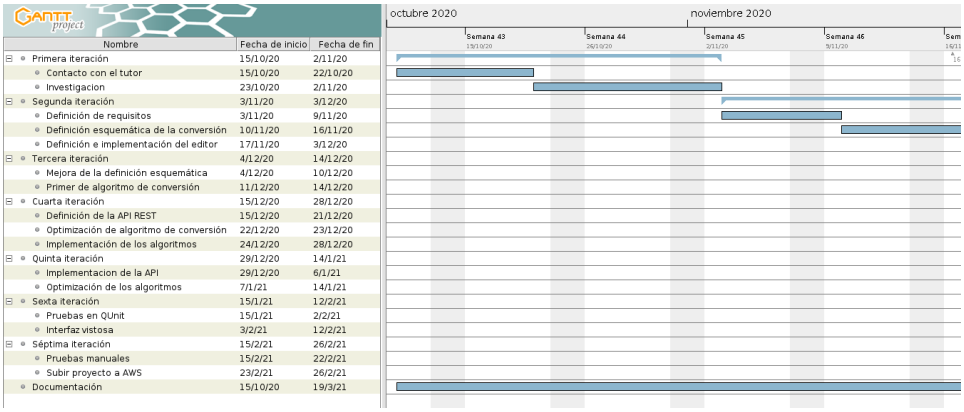
Todo el proyecto desarrollado es subido a Amazon Web Services y se comprueban las distintas funcionalidades del proyecto.

### 2.2.8. Documentación

Durante la fase de documentación del proyecto se ha escrito y revisado el presente documento, donde se ha dejado constancia de las distintas fases de desarrolllo del proyecto y el resultado de las mismas.

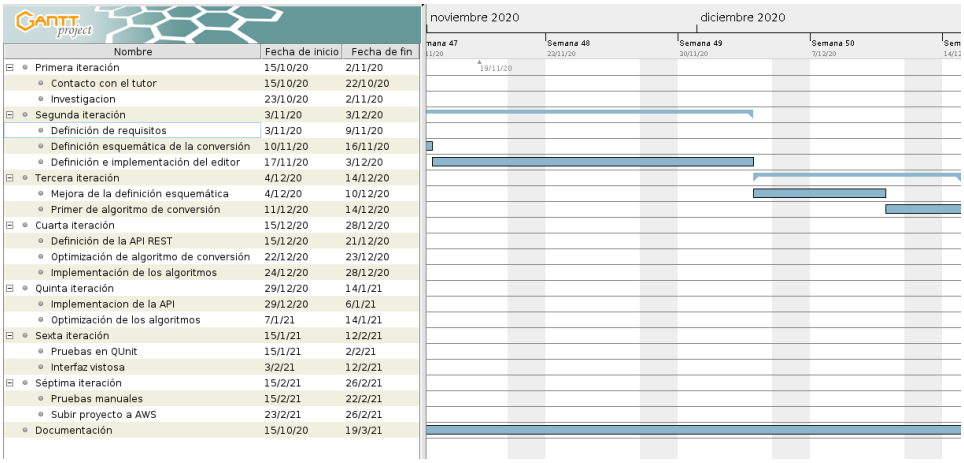
### 2.2.9. Diagrama de Gantt

Puede verse en las imágenes 2.2 - 2.6




| Nombre | Fecha de inicio | Fecha de fin |
|---|---|---|
| Primera iteración | 15/10/20 | 2/11/20 |
| Contacto con el tutor | 15/10/20 | 22/10/20 |
| Investigacion | 23/10/20 | 2/11/20 |
| Segunda iteración | 3/11/20 | 3/12/20 |
| Definición de requisitos | 3/11/20 | 9/11/20 |
| Definición esquemática de la conversión | 10/11/20 | 16/11/20 |
| Definición e implementación del editor | 17/11/20 | 3/12/20 |
| Tercera iteración | 4/12/20 | 14/12/20 |
| Mejora de la definición esquemática | 4/12/20 | 10/12/20 |
| Primer de algoritmo de conversión | 11/12/20 | 14/12/20 |
| Cuarta iteración | 15/12/20 | 28/12/20 |
| Definición de la API REST | 15/12/20 | 21/12/20 |
| Optimización de algoritmo de conversión | 22/12/20 | 23/12/20 |
| Implementación de los algoritmos | 24/12/20 | 28/12/20 |
| Quinta iteración | 29/12/20 | 14/1/21 |
| Implementacion de la API | 29/12/20 | 6/1/21 |
| Optimización de los algoritmos | 7/1/21 | 14/1/21 |
| Sexta iteración | 15/1/21 | 12/2/21 |
| Pruebas en QUnit | 15/1/21 | 2/2/21 |
| Interfaz vistosa | 3/2/21 | 12/2/21 |
| Séptima iteración | 15/2/21 | 26/2/21 |
| Pruebas manuales | 15/2/21 | 22/2/21 |
| Subir proyecto a AWS | 23/2/21 | 26/2/21 |
| Documentación | 15/10/20 | 19/3/21 |

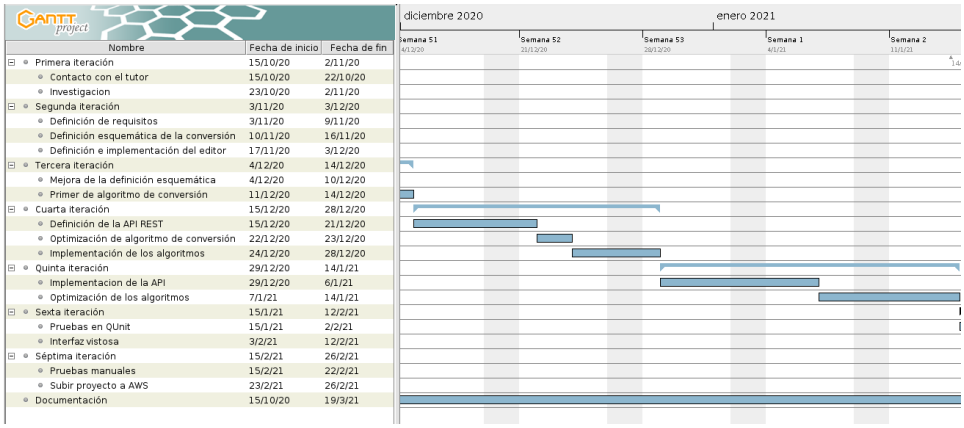
Figura 2.2: Primera imagen del diagrama de Gantt




| Nombre | Fecha de inicio | Fecha de fin |
|---|---|---|
| Primera iteración | 15/10/20 | 2/11/20 |
| Contacto con el tutor | 15/10/20 | 22/10/20 |
| Investigacion | 23/10/20 | 2/11/20 |
| Segunda iteración | 3/11/20 | 3/12/20 |
| Definición de requisitos | 3/11/20 | 9/11/20 |
| Definición esquemática de la conversión | 10/11/20 | 16/11/20 |
| Definición e implementación del editor | 17/11/20 | 3/12/20 |
| Tercera iteración | 4/12/20 | 14/12/20 |
| Mejora de la definición esquemática | 4/12/20 | 10/12/20 |
| Primer de algoritmo de conversión | 11/12/20 | 14/12/20 |
| Cuarta iteración | 15/12/20 | 28/12/20 |
| Definición de la API REST | 15/12/20 | 21/12/20 |
| Optimización de algoritmo de conversión | 22/12/20 | 23/12/20 |
| Implementación de los algoritmos | 24/12/20 | 28/12/20 |
| Quinta iteración | 29/12/20 | 14/1/21 |
| Implementacion de la API | 29/12/20 | 6/1/21 |
| Optimización de los algoritmos | 7/1/21 | 14/1/21 |
| Sexta iteración | 15/1/21 | 12/2/21 |
| Pruebas en QUnit | 15/1/21 | 2/2/21 |
| Interfaz vistosa | 3/2/21 | 12/2/21 |
| Séptima iteración | 15/2/21 | 26/2/21 |
| Pruebas manuales | 15/2/21 | 22/2/21 |
| Subir proyecto a AWS | 23/2/21 | 26/2/21 |
| Documentación | 15/10/20 | 19/3/21 |

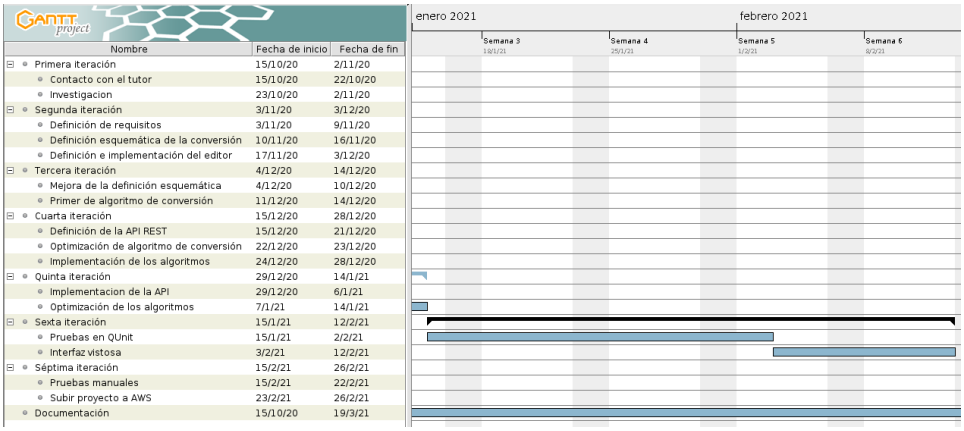
Figura 2.3: Segunda imagen del diagrama de Gantt

Es necesario explicar adecuadamente algunas tareas realizadas. En el caso de investigación, se refiere a la investigación sobre el uso de distintos editores o crear uno propio. Esta investigación se aprovecha durante todo el proyecto ya que al final se usó uno propio. Para el caso de interfaz vistosa, indica que la interfaz ya estaba realizada como un prototipo de baja fidelidad (véase en el Capítulo 4.3) pero faltaban colores, diseño adaptativo y añadir ventanas emergentes.



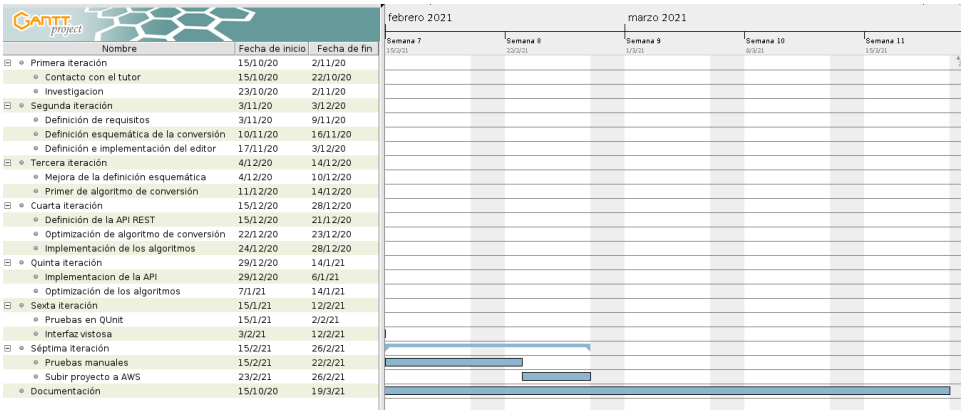

| Nombre | Fecha de inicio | Fecha de fin |
|---|---|---|
| Primera iteración | 15/10/20 | 2/11/20 |
| Contacto con el tutor | 15/10/20 | 22/10/20 |
| Investigacion | 23/10/20 | 2/11/20 |
| Segunda iteración | 3/11/20 | 3/12/20 |
| Definición de requisitos | 3/11/20 | 9/11/20 |
| Definición esquemática de la conversión | 10/11/20 | 16/11/20 |
| Definición e implementación del editor | 17/11/20 | 3/12/20 |
| Tercera iteración | 4/12/20 | 14/12/20 |
| Mejora de la definición esquemática | 4/12/20 | 10/12/20 |
| Primer de algoritmo de conversión | 11/12/20 | 14/12/20 |
| Cuarta iteración | 15/12/20 | 28/12/20 |
| Definición de la API REST | 15/12/20 | 21/12/20 |
| Optimización de algoritmo de conversión | 22/12/20 | 23/12/20 |
| Implementación de los algoritmos | 24/12/20 | 28/12/20 |
| Quinta iteración | 29/12/20 | 14/1/21 |
| Implementacion de la API | 29/12/20 | 6/1/21 |
| Optimización de los algoritmos | 7/1/21 | 14/1/21 |
| Sexta iteración | 15/1/21 | 12/2/21 |
| Pruebas en QUnit | 15/1/21 | 2/2/21 |
| Interfaz vistosa | 3/2/21 | 12/2/21 |
| Séptima iteración | 15/2/21 | 26/2/21 |
| Pruebas manuales | 15/2/21 | 22/2/21 |
| Subir proyecto a AWS | 23/2/21 | 26/2/21 |
| Documentación | 15/10/20 | 19/3/21 |

Figura 2.4: Tercera imagen del diagrama de Gantt




| Nombre | Fecha de inicio | Fecha de fin |
|---|---|---|
| Primera iteración | 15/10/20 | 2/11/20 |
| Contacto con el tutor | 15/10/20 | 22/10/20 |
| Investigacion | 23/10/20 | 2/11/20 |
| Segunda iteración | 3/11/20 | 3/12/20 |
| Definición de requisitos | 3/11/20 | 9/11/20 |
| Definición esquemática de la conversión | 10/11/20 | 16/11/20 |
| Definición e implementación del editor | 17/11/20 | 3/12/20 |
| Tercera iteración | 4/12/20 | 14/12/20 |
| Mejora de la definición esquemática | 4/12/20 | 10/12/20 |
| Primer de algoritmo de conversión | 11/12/20 | 14/12/20 |
| Cuarta iteración | 15/12/20 | 28/12/20 |
| Definición de la API REST | 15/12/20 | 21/12/20 |
| Optimización de algoritmo de conversión | 22/12/20 | 23/12/20 |
| Implementación de los algoritmos | 24/12/20 | 28/12/20 |
| Quinta iteración | 29/12/20 | 14/1/21 |
| Implementacion de la API | 29/12/20 | 6/1/21 |
| Optimización de los algoritmos | 7/1/21 | 14/1/21 |
| Sexta iteración | 15/1/21 | 12/2/21 |
| Pruebas en QUnit | 15/1/21 | 2/2/21 |
| Interfaz vistosa | 3/2/21 | 12/2/21 |
| Séptima iteración | 15/2/21 | 26/2/21 |
| Pruebas manuales | 15/2/21 | 22/2/21 |
| Subir proyecto a AWS | 23/2/21 | 26/2/21 |
| Documentación | 15/10/20 | 19/3/21 |

Figura 2.5: Cuarta imagen del diagrama de Gantt




| Nombre | Fecha de inicio | Fecha de fin |
|---|---|---|
| Primera iteración | 15/10/20 | 2/11/20 |
| Contacto con el tutor | 15/10/20 | 22/10/20 |
| Investigacion | 23/10/20 | 2/11/20 |
| Segunda iteración | 3/11/20 | 3/12/20 |
| Definición de requisitos | 3/11/20 | 9/11/20 |
| Definición esquemática de la conversión | 10/11/20 | 16/11/20 |
| Definición e implementación del editor | 17/11/20 | 3/12/20 |
| Tercera iteración | 4/12/20 | 14/12/20 |
| Mejora de la definición esquemática | 4/12/20 | 10/12/20 |
| Primer de algoritmo de conversión | 11/12/20 | 14/12/20 |
| Cuarta iteración | 15/12/20 | 28/12/20 |
| Definición de la API REST | 15/12/20 | 21/12/20 |
| Optimización de algoritmo de conversión | 22/12/20 | 23/12/20 |
| Implementación de los algoritmos | 24/12/20 | 28/12/20 |
| Quinta iteración | 29/12/20 | 14/1/21 |
| Implementacion de la API | 29/12/20 | 6/1/21 |
| Optimización de los algoritmos | 7/1/21 | 14/1/21 |
| Sexta iteración | 15/1/21 | 12/2/21 |
| Pruebas en QUnit | 15/1/21 | 2/2/21 |
| Interfaz vistosa | 3/2/21 | 12/2/21 |
| Séptima iteración | 15/2/21 | 26/2/21 |
| Pruebas manuales | 15/2/21 | 22/2/21 |
| Subir proyecto a AWS | 23/2/21 | 26/2/21 |
| Documentación | 15/10/20 | 19/3/21 |

Figura 2.6: Quinta imagen del diagrama de Gantt

## 2.3. Organización

A continuación se expone un análisis de la organización utilizada en este proyecto, así como la relación entre los roles del proyecto, el software y hardware utilizados para el desarrollo del mismo.

### 2.3.1. Roles y Responsabilidad

Se pueden diferenciar dos roles principales, director de proyecto y desarrollador. El tutor y cotutor de este proyecto han ejercido el rol de director de proyecto y el alumno ha ejercido el rol de desarrollador.

Se realizan reuniones entre los integrantes del proyecto, al final de cada iteración para comprobar que se han cumplido los requisitos establecidos y se definen los requisitos a cumplir en la siguiente iteración.

En caso de que surja alguna duda en alguna de las iteraciones que pueda paralizar el desarrollo del proyecto, la comunicación por correo entre el director del proyecto y el desarrollador puede darse a mitad de la iteración para poder alcanzar los requisitos establecidos para el final de la iteración.

El director de proyecto es el encargado de orientar al desarollador a lo largo del proceso, así como la revisión final de la aplicación y de la documentación.

El desarrollador del proyecto se ha encargado del diseño y desarrollo de la aplicación, así como de la documentación.

### 2.3.2. Recursos y Herramientas

- Software
    - Node.js: es un entorno multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript.
    - PDF.JS: Librería para cargar archivos PDF directamente en el navegador.
    - JQuery: Librería para manipular elementos HTML y CSS de forma mucho sencilla.
    - JSZIP: Librería encargada de cargar y guardar el documento en un zip.

    - TeXlive: Entorno necesario para la conversión de LaTeX a PDF
    - Chromium: Navegador web principal para el desarrollo.
    - Emacs: es un editor de texto con una gran cantidad de funciones, es parte del proyecto GNU.
    - Firefox: Navegador web secundario para el desarrollo para comprobar compatibilidades con otros motor

- Hardware
    - Ordenador de sobremesa
        - Intel i7-2200K
        - 8GB de RAM DDR3
        - Memoria SSD 256GB
        - Gráfica Nvidia Geforce GTX 550 Ti
    - Móvil
        - Qualcomm Snapdragon 700
        - 2GB DDR2
        - Android 10

Uno de los recursos más importante de este proyecto ha sido la documentación proporcionada por los desarrolladores de los navegadores. Tanto Google[3] como Mozilla[4] mantienen unas referencias y guía con distintos enfoques para facilitar su comprensión a usuarios novatos. Estos proyectos alojan la información de nuevos estándares y funcionalidades resumidas para evitar leer todos los borradores y estándares que van evolucionando año a año. El proyecto usa una gran cantidad de módulos para poder realizar todas las funciones necesarias, en este párrafo se indican todas las dependencias: Node[17](express[19], cors[18], morgan[20], tree-kill[21]), PDFjs[22], JQuery[24], Filesaver[25], NPM[26], TogetherJS[23], XeLaTeX[27], JSZip[28], JS-Beautify[29] y Qunit[30].

## 2.4. Costes

En esta parte dejaremos constancia del coste que requiere realizar un proyecto como este teniendo en cuenta los recursos utilizados para la realización del mismo.

Durante el desarrollo de un producto software se hace uso de una serie de recursos para llegar a satisfacer los requisitos a cumplir por el producto. Estos recursos pueden ser materiales o humanos.

Si tenemos en cuenta el servidor de Amazon Web Services, el coste ronda los 30-40€ mensuales, si suponemos que incluye un dominio y soporte técnico. Este sistema solo se usará para el producto final, entonces solo se añadirá como gasto al último mes. Se necesitará pagar mensualmente para mantener el servicio a partir de su publicación. Aunque teniendo en cuenta que el coste dependerá del tráfico que genere la plataforma.

| Coste de los recursos | |
|---|---|
| Sofware | 0€ |
| Hardware | 734€ |
| Coste total | 734€ |

Cuadro 2.1: Costes de los recursos del proyecto

En lo que respecta a los costes humanos, un desarrollador junior suele tener un sueldo medio entorno a unos 1000-1200€ netos al mes, teniendo en cuenta que el proyecto ha tomado unos 7-8 meses, obtendríamos un coste medio aproximado de unos 9100€ de costes humanos.

Puede ver reflejado todos estos cálculos en la tabla de recursos (Cuadro 2.1) y la tabla de costes totales (Cuadro 2.2).

| Coste total del proyecto | |
|---|---|
| Coste material | 764€ |
| Coste recursos humanos | 9100€ |
| Coste total | 9864€ |

Cuadro 2.2: Costes totales del proyecto

## 2.5. Gestión de riesgos

Durante el desarrollo de un software hay una serie de riesgos que pueden llegar a darse y que deben afrontarse. Estos riesgos podrán darse con una probabilidad: muy baja, baja, media, alta o muy alta. Estos son los riesgos a los que este proyecto puede llegar a tener que enfrentarse tal como se muestra en la tabla 2.3.

| ID | Riesgo | Impacto | Probabilidad | Solución |
|---|---|---|---|---|
| R-01 | Dificultad en el aprendizaje de alguna herramienta. | Retraso en el desarrollo de varias semanas a meses | Alta | Modificar la planificación del proyecto |
| R-02 | Aumento en la complejidad del proyecto | Retraso sustancial de todo el proyecto | Medio | Redefinir la planificación original |
| R-03 | Cambio en las interfaces o en las herramientas usadas | Inutilización de parte del proyecto | Bajo | Recuperar versión original o modificar planificación |
| R-04 | Pérdida del proyecto | Una pérdida parcial o total, afectaría al proyecto gravemente | Bajo | Uso de git y copias de seguridad |
| R-05 | Avería en el hardware | Imposibilidad de hacer nuevos cambios | Muy baja | Un equipo de repuesto |

Cuadro 2.3: Riesgos posibles en el proyecto

# Parte II

# Desarrollo

# Capítulo 3

# Análisis del sistema

## 3.1. Catálogo de requisitos

En este apartado se describirán los diferentes requisitos del sistema, incluyendo los requisitos funcionales, los requisitos no funcionales, los requisitos de información y las reglas de negocio.

### 3.1.1. Requisitos funcionales

Los requisitos funcionales de este sistema se detallan a continuación en las tablas 3.1 - 3.14. En este apartado se indican los requisitos básicos para el sistema, este podrá cumplir con otros incorporados durante el desarrollo para mejorar el conjunto de la plataforma. Las primeras nueve tablas son las funcionalidades imprescindibles del proyecto. Sin estas funcionalidades el usuario no podría incorporar ninguna etiqueta que haga agradable la lectura del documento ni generar documentos PDF. Esto dejaría al proyecto con un editor de poca utilidad y generaría documentos poco legibles.

| RF-01 | Creación de PDFs a partir del editor |
|---|---|
| Descripción | El sistema deberá permitir al usuario crear PDFs en el sistema |
| Prioridad | Alta |
| Observaciones | El usuario deberá introducir un documento válido. |

Cuadro 3.1: RF-01 (Creación de documentos PDF a partir del texto introducido)

| RF-02 | Conversión de HTML a lenguaje intermedio. |
|---|---|
| Descripción | El sistema deberá permitir al usuario convertir al lenguaje intermedio. |
| Prioridad | Alta |
| Observaciones | El usuario deberá introducir un documento válido. |

Cuadro 3.2: RF-02 (Conversión de HTML a lenguaje intermedio.)

| RF-03 | Conversión de lenguaje intermedio a HTML. |
|---|---|
| Descripción | El sistema deberá permitir al usuario convertir a HTML. |
| Prioridad | Alta |
| Observaciones | El usuario deberá introducir un documento válido. |

Cuadro 3.3: RF-03 (Conversión de lenguaje intermedio a HTML.)

| RF-04 | Selección de plantillas |
|---|---|
| Descripción | El sistema deberá permitir al usuario cambiar la plantilla para crear los PDFs. |
| Prioridad | Media |
| Observaciones | El usuario deberá introducir una plantilla válida. |

Cuadro 3.4: RF-04 (Conversión de lenguaje intermedio a HTML.)

| RF-05 | Creación de tablas |
|---|---|
| Descripción | El sistema deberá permitir al usuario crear tablas en el editor HTML. |
| Prioridad | Alta |
| Observaciones | El usuario deberá introducir unas dimensiones válidas. |

Cuadro 3.5: RF-05 (Creación de tablas en el editor.)

| RF-06 | Inserción de imágenes |
|---|---|
| Descripción | El sistema deberá permitir al usuario insertar imágenes en el editor HTML. |
| Prioridad | Alta |
| Observaciones | El usuario deberá introducir una imagen y texto alternativo válidos. |

Cuadro 3.6: RF-06 (Inserción de imágenes en el editor.)

| RF-07 | Inserción de enlaces |
|---|---|
| Descripción | El sistema deberá permitir al usuario insertar enlaces en el editor HTML. |
| Prioridad | Alta |
| Observaciones | El usuario deberá introducir un enlace válido. |

Cuadro 3.7: RF-07 (Inserción de enlaces en el editor.)

| RF-08 | Inserción de lista desordenada |
|---|---|
| Descripción | El sistema deberá permitir al usuario insertar listas desordenadas en el editor HTML. |
| Prioridad | Alta |
| Observaciones | El usuario deberá introducir el primer elemento de una lista desordenada. |

Cuadro 3.8: RF-08 (Inserción de listas desordenadas en el editor.)

| RF-09 | Inserción de título |
|---|---|
| Descripción | El sistema deberá permitir al usuario insertar títulos en el editor HTML. |
| Prioridad | Alta |
| Observaciones | El usuario deberá seleccionar un texto válido |

Cuadro 3.9: RF-09 (Inserción de títulos en el editor.)

Las siguientes tablas son funcionalidades que están pensadas para usuarios avanzados. Insertar código LaTeX, usar contenedores y añadir texto sin comprobar son funciones nuevas que los usuarios desconocen de otros sistemas, por ello pueden ser más dificil de comprender y poco utilizadas.

| RF-10 | Inserción de contenedores |
|---|---|
| Descripción | El sistema deberá permitir al usuario insertar contenedores en el editor HTML. |
| Prioridad | Media |
| Observaciones | El usuario deberá seleccionar un texto válido |

Cuadro 3.10: RF-10 (Inserción de contenedores en el editor.)

| RF-11 | Incrustar código |
|---|---|
| Descripción | El sistema deberá permitir al usuario insertar código de LaTeX en el editor HTML. |
| Prioridad | Media |
| Observaciones | El usuario deberá seleccionar un texto válido |

Cuadro 3.11: RF-11 (Incrustar código.)

| RF-12 | Insertar texto sin comprobar |
|---|---|
| Descripción | El sistema deberá permitir al usuario insertar texto sin comprobar en el editor HTML. |
| Prioridad | Media |
| Observaciones | El usuario deberá seleccionar un texto válido |

Cuadro 3.12: RF-12 (Texto sin comprobar.)

Estas últimas tablas muestran las formas que tiene el sistema para recuperar el documento si se cierra el navegador inesperadamente o se borra por parte del usuario y quiere recuperarlo.

| RF-13 | Guardar texto del editor |
|---|---|
| Descripción | El sistema deberá permitir al usuario guardar el texto del editor HTML. |
| Prioridad | Baja |
| Observaciones | |

Cuadro 3.13: RF-13 (Guardar texto del editor.)

| RF-14 | Cargar texto del editor |
|---|---|
| Descripción | El sistema deberá permitir al usuario cargar el texto del editor HTML. |
| Prioridad | Baja |
| Observaciones | Hace falta haber guardado antes. |

Cuadro 3.14: RF-14 (Cargar texto del editor.)

| RF-15 | Cargar documento a partir de archivo guardado |
|---|---|
| Descripción | El sistema deberá permitir al usuario cargar el texto del editor HTML a partir de archivo |
| Prioridad | Baja |
| Observaciones | Hace falta haber guardado antes. |

Cuadro 3.15: RF-15 (Cargar editor a partir de archivo guardado.)

### 3.1.2. Requisitos no funcionales

Los requisitos no funcionales que se han de cumplir en este sistema se detallan a continuación en las tablas 3.16 - 3.21.

| RNF-01 | Usabilidad |
|---|---|
| Descripción | Para garantizar el requisito no funcional de usabilidad se han de cumplir las siguientes Heurísticas de Nielsen:<br>▪ *Visibilidad del estado del sistema:* El sistema indicará la posición del cursor, llevará al usuario al elemento modificado y mostrará si se está convirtiendo a PDF para saber qué esta haciendo el sistema<br>▪ *Utilizar el mismo lenguaje que el usuario:* Elementos visuales y textuales que remarquen la utilidad de las herramientas<br>▪ *Prevención de errores:* El sistema notificará a los usuarios los errores e intentará simplificarlos para su fácil comprensión<br>▪ *Diseño minimalista:* El sistema tiene gran cantidad de funcionalidades pero el estado inicial busca ser de una carga cognitiva menor que el sistema completo con todas sus funcionalidades. |

Cuadro 3.16: RNF-01 (Usabilidad.)

| RNF-02 | Portabilidad |
|---|---|
| Descripción | Para garantizar el requisito no funcional de portabilidad se garantizará:<br>▪ La API está centrada para su funcionamiento en una distribución Linux pero ha de funcionar en un Windows Server.<br>▪ El cliente puede ejecutarse en cualquier sistema operativo que permita la ejecución de un navegador que permita Javascript ES5. |

Cuadro 3.17: RNF-02 (Portabilidad.)

| RNF-03 | Eficiencia |
|---|---|
| Descripción | Para garantizar el requisito no funcional de eficiencia se garantizará:<br>▪ Que el sistema responda con fluidez a las acciones que realice el usuario sin derivar en errores o retrocesos en la aplicación. |

Cuadro 3.18: RNF-03 (Eficiencia.)

| RNF-04 | Seguridad |
|---|---|
| Descripción | Para garantizar el requisito no funcional de seguridad se garantizará:<br>▪ Todas las conexiones con el sistema se harán bajo TLS v1.3 para evitar que se pueda interponer un actor malintencionado.<br>▪ Sólo se comparte información con los servidores cuando es necesario, en el caso de compartir el archivo y de convertirlo a PDF.<br>▪ En el caso de la conversión, nada más obtenido el resultado se elimina los archivos.<br>▪ Se comprueba todo el texto HTML insertado para evitar la inserción de scripts malintencionado. |

Cuadro 3.19: RNF-04 (Seguridad.)

| RNF-05 | Interoperabilidad |
|---|---|
| Descripción | Para garantizar el requisito no funcional de interoperabilidad se garantizará:<br>▪ Que el sistema pueda utilizarse desde sistemas externos.<br>▪ Todos los protocolos usados en este sistema son públicos y bien conocidos. |

Cuadro 3.20: RNF-05 (Interoperabillidad.)

| RNF-06 | Mantenibilidad |
|---|---|
| Descripción | Para garantizar el requisito no funcional de mantenibilidad se garantizará:<br>▪ Se ha separado adecuadamente las distintas partes del sistema para que pueda ser reutilizado, mejorado y actualizado. Si hiciera falta cambiar algún componente solo habría que modificar el fichero oportuno. |

Cuadro 3.21: RNF-06 (Mantenibilidad.)

### 3.1.3. Requisitos de información

Los requisitos de información del sistema se detallan a continuación en las tablas 3.22 y 3.23

| RINF-01 | Editor HTML |
|---|---|
| Descripción | ▪ Texto almacenado en el propio editor<br>▪ Menú modificado para cada editor. |

Cuadro 3.22: RINF-01 (Texto en HTML.)

| RINF-02 | Editor de lenguaje intermedio |
|---|---|
| Descripción | ▪ Texto almacenado en el propio editor<br>▪ Menú modificado para cada editor. |

Cuadro 3.23: RINF-02 (Texto en lenguaje intermedio.)

## 3.2. Casos de uso

### 3.2.1. Actores

El sistema está enfocado a un usuario objetivo que es el que puede hacer uso de todas las funciones del sistema como escribir en el editor, cargar un documento y otras tantas. El actor usuario (véase tabla 3.24) es el que representa en los casos de uso a todas las personas que pueden utilizar la plataforma.

El tiempo (véase tabla 3.25) representa al actor que inicia las tareas planificadas ya que estas tareas se lanzarán en segundo plano sin que el usuario pulse ningún botón[32]. La única tarea planificada que tiene el sistema es la de guardado de automático del editor. Esta tarea almacena en el navegador el documento para recuperarlo en otro momento si hace falta, tal como se describe en el caso de uso 12 en la tabla 3.37.

Se decidieron estos actores porque el objetivo principal es el usuario y que pueda tener la mayor cantidad de funcionalidades posibles para tener una experiencia agradable con el editor. El actor tiempo y la tarea planificada son necesarios para que el usuario no tenga que estar continuamente guardando el documento porque le llevará a tener una gran cantidad de archivos de guardado en su ordenador que al poco tiempo quedarán incompletos.

| Actor | Usuario |
|---|---|
| Descripción | Usuario de la plataforma que aprovecha las funcionalidades |

Cuadro 3.24: Usuario

| Actor | Tiempo |
|---|---|
| Descripción | Tiempo actual para el cálculo del tiempo transcurrido para guardar el estado del cliente por si el usuario cierra el navegador. |

Cuadro 3.25: Tiempo

### 3.2.2. Diagramas de caso de uso

Se han separado los casos de uso para que sean más fácil de ver. Son las dos siguientes imágenes 3.1 y 3.2. Se han organizando los casos de uso de una manera diferente a los requisitos, se ha preferido por similitud en su definiciones, de ahí que los casos de uso que requieren usar include o intervienen otros actores estén en la misma imagen.

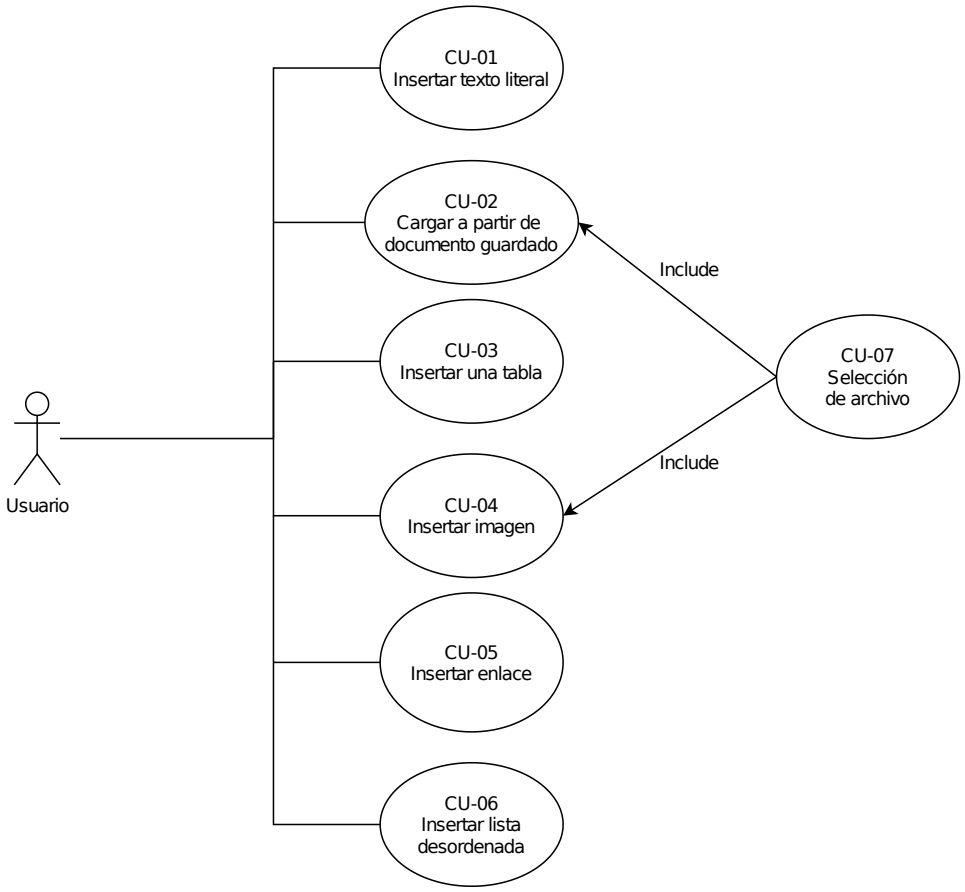



Figura 3.1: Los primeros 7 casos de uso

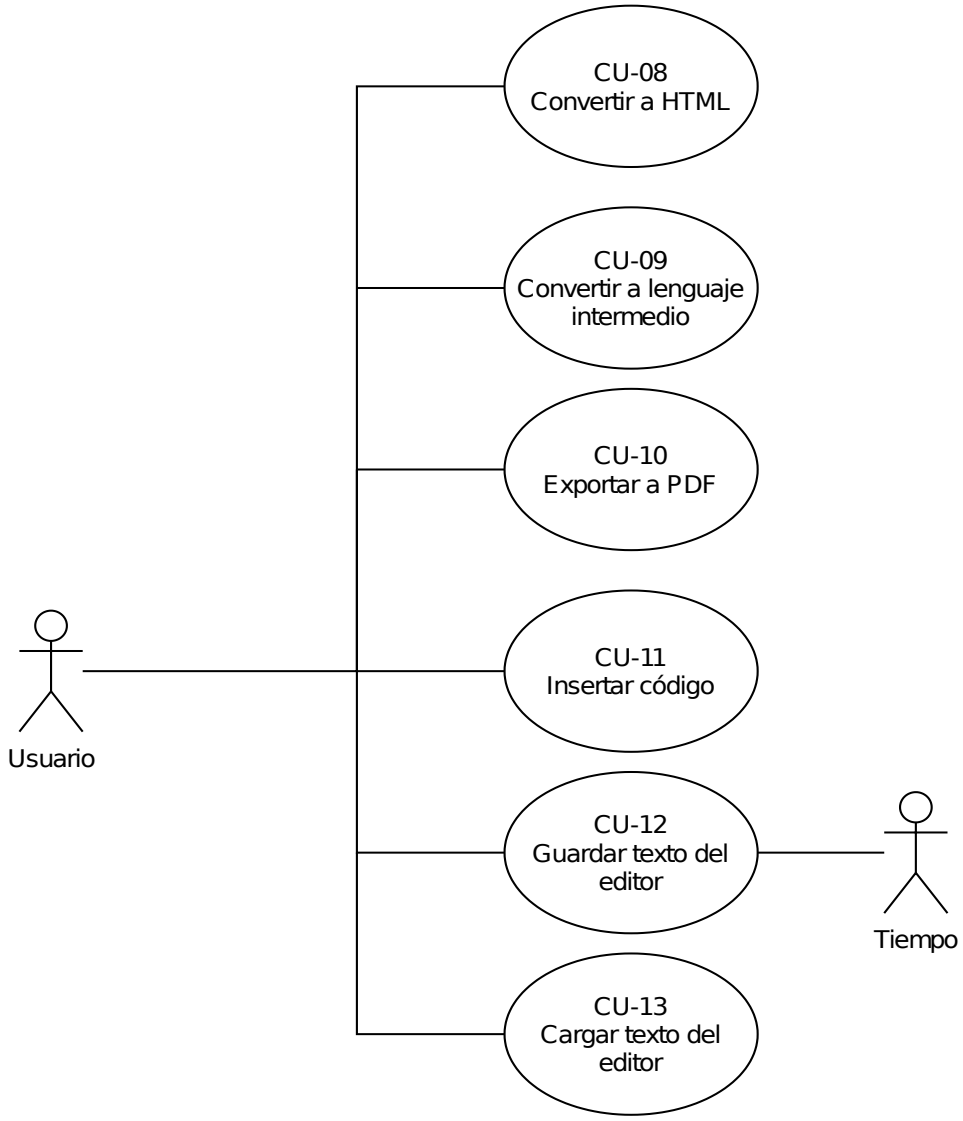



Figura 3.2: Los siguientes casos de uso

### 3.2.3. Descripción de los casos de uso

| CU-01 | Insertar texto literal |
|---|---|
| Requisitos | Requisito funcional 3.12 |
| Precondiciones | Tener texto seleccionado |
| Postcondiciones | 1. El usuario selecciona el texto a convertir<br>2. El usuario activa la opción de mantener el texto sin convertir<br>3. El sistema modifica el texto seleccionado y le cambia el tipo de letra para que el usuario pueda visualizarlo<br>4. El usuario comprueba que esta acción ocurre satisfactoriamente. |
| Escenario alternativo | 3a. El usuario no ha seleccionado texto<br>3a.1 El sistema verifica que el usuario<br>no ha seleccionado texto<br>y no realiza acción alguna |

Cuadro 3.26: CU-01 (Caso de uso para insertar texto literal)

| CU-02 | Cargar a partir de archivo de guardado |
|---|---|
| Requisitos | Requisito funcional 3.15 y caso de uso 3.33 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de cargar archivo<br>2. El sistema activa la selección de archivo (include caso de uso 8)<br>3. El sistema comprueba el archivo selecionado<br>4. El sistema realiza la carga adecuadamente |
| Escenario alternativo | 2a. El usuario cancela la selección<br>2a.1 El sistema finaliza el caso de uso<br>3a. El sistema verifica el archivo<br>3a.1 El sistema cancela la acción por archivo erróneo |

Cuadro 3.27: CU-02 (Caso de uso para cargar documento)

| CU-03 | Insertar tabla |
|---|---|
| Requisitos | Requisito funcional 3.5 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de tabla<br>2. El sistema muestra una ventana emergente donde pide ancho y alto de la tabla<br>3. El usuario insertar los valores<br>4. El sistema comprueba e inserta la tabla |
| Escenario alternativo | 3a. El usuario cierra la ventana<br>3a.1 El sistema finaliza el caso de uso<br>4a. El sistema verifica los datos aportados<br>4a.1 El sistema mantiene la ventana<br>abierta hasta que el usuario aporte datos válidos |

Cuadro 3.28: CU-03 (Caso de uso para insertar tabla)

| CU-04 | Insertar imagen |
|---|---|
| Requisitos | Requisito funcional 3.6 y caso de uso 3.33 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de insertar imagen<br>2. El sistema activa la selección de archivo (include caso de uso 8)<br>3. El sistema comprueba el archivo selecionado<br>4. El sistema inserta la imagen |
| Escenario alternativo | 2a. El usuario cancela la selección<br>2a.1 El sistema finaliza el caso de uso<br>3a. El sistema verifica el archivo<br>3a.1 El sistema cancela la acción por archivo erróneo |

Cuadro 3.29: CU-04 (Caso de uso para insertar imagen)

| CU-05 | Insertar enlace |
|---|---|
| Requisitos | Requisito funcional 3.7 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de enlace<br>2. El sistema muestra una ventana emergente donde pide el enlace<br>3. El usuario introduce el enlace<br>4. El sistema comprueba e inserta la enlace en el editor |
| Escenario alternativo | 3a. El usuario cierra la ventana<br>3a.1 El sistema finaliza el caso de uso<br>4a. El sistema verifica los datos aportados<br>4a.1 El sistema mantiene la ventana<br>abierta hasta que el usuario aporte datos válidos |

Cuadro 3.30: CU-05 (Caso de uso para insertar enlace)

| CU-06 | Insertar lista desordenada |
|---|---|
| Requisitos | Requisito funcional 3.8 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de lista desordenada<br>2. El sistema comprueba la posición seleccionada<br>3. inserta la lista |
| Escenario alternativo | 2a. El usuario no ha seleccionado posición<br>2a.1 El sistema utiliza la última posición del editor |

Cuadro 3.31: CU-06 (Caso de uso para insertar lista)

| CU-07 | Selección de archivo |
|---|---|
| Requisitos | |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El sistema muestra un listado de archivos y una lista de carpetas disponibles<br>2. El usuario selecciona un elemento y pulsa seleccionar<br>3. El sistema comprueba que es una dirección válida<br>4. El sistema selecciona el archivo y finaliza el caso de uso |
| Escenario alternativo | 2a. El usuario cancela la selección<br>2a.1 El sistema finaliza el caso de uso |

Cuadro 3.32: CU-07 (Caso de uso seleccionar archivo)

| CU-08 | Convertir a HTML |
|---|---|
| Requisitos | Requisito funcional 3.3 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de convertir<br>2. El sistema obtiene el texto del editor<br>3. El sistema convierte el texto<br>4. El sistema muestra el texto convertido a HTML. |
| Escenario alternativo | 2a. El texto del editor tiene un error<br>2a.1 El sistema muestra el error sintáctico por pantalla |

Cuadro 3.33: CU-08 (Caso de uso para convertir a HTML)

Las siguientes descripciones tratan de los casos de uso que se entienden que los usuarios utilizan menos veces. Aún así el sistema debe de proporcionarlas para que el proyecto esté completo.

| CU-09 | Convertir a lenguaje intermedio |
|---|---|
| Requisitos | Requisito funcional 3.2 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de convertir<br>2. El sistema obtiene el texto del editor<br>3. El sistema convierte el texto<br>4. El sistema muestra el texto convertido a lenguaje intermedio. |
| Escenario alternativo | 2a. El texto del editor tiene un error<br>2a.1 El sistema muestra el error sintáctico por pantalla |

Cuadro 3.34: CU-09 (Caso de uso para convertir a lenguaje intermedio)

| CU-10 | Exportar a PDF |
|---|---|
| Requisitos | Requisito funcional 3.1 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de exportar a PDF<br>2. El sistema obtiene el texto del editor<br>3. El sistema convierte el texto a LaTeX<br>4. El sistema convierte a PDF.<br>5. El sistema muestra el PDF satisfactoriamente |
| Escenario alternativo | 2a. El texto del editor tiene un error<br>2a.1 El sistema muestra el error sintáctico por pantalla<br>3a El sistema detecta un error en la conversión a PDF<br>3a.1 El sistema finaliza el caso de uso mostrando el error |

Cuadro 3.35: CU-10 (Caso de uso para exportar a PDF)

| CU-11 | Insertar código |
|---|---|
| Requisitos | Requisito funcional 3.11 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de insertar código<br>2. El sistema convierte la línea seleccionada a código insertado |
| Escenario alternativo | 2a. El usuario no ha seleccionado línea<br>2a.1 El sistema utiliza la última línea del editor |

Cuadro 3.36: CU-11 (Caso de uso para insertar código)

| CU-12 | Guardar texto del editor |
|---|---|
| Requisitos | Requisito funcional 3.15 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de guardar estado<br>2. El sistema guarda el estado del editor |
| Escenario alternativo | 1a El tiempo indica que hay que guardar el estado<br>1a.1 El sistema inicia el guardado |

Cuadro 3.37: CU-12 (Caso de uso para guardar texto del editor)

| CU-13 | Cargar texto del editor |
|---|---|
| Requisitos | Requisito funcional 3.15 |
| Precondiciones | Sistema correctamente iniciado |
| Postcondiciones | 1. El usuario selecciona el botón de cargar estado<br>2. El sistema carga el estado del editor |
| Escenario alternativo | |

Cuadro 3.38: CU-13 (Caso de uso para cargar texto del editor)

## 3.3. Modelo conceptual de datos

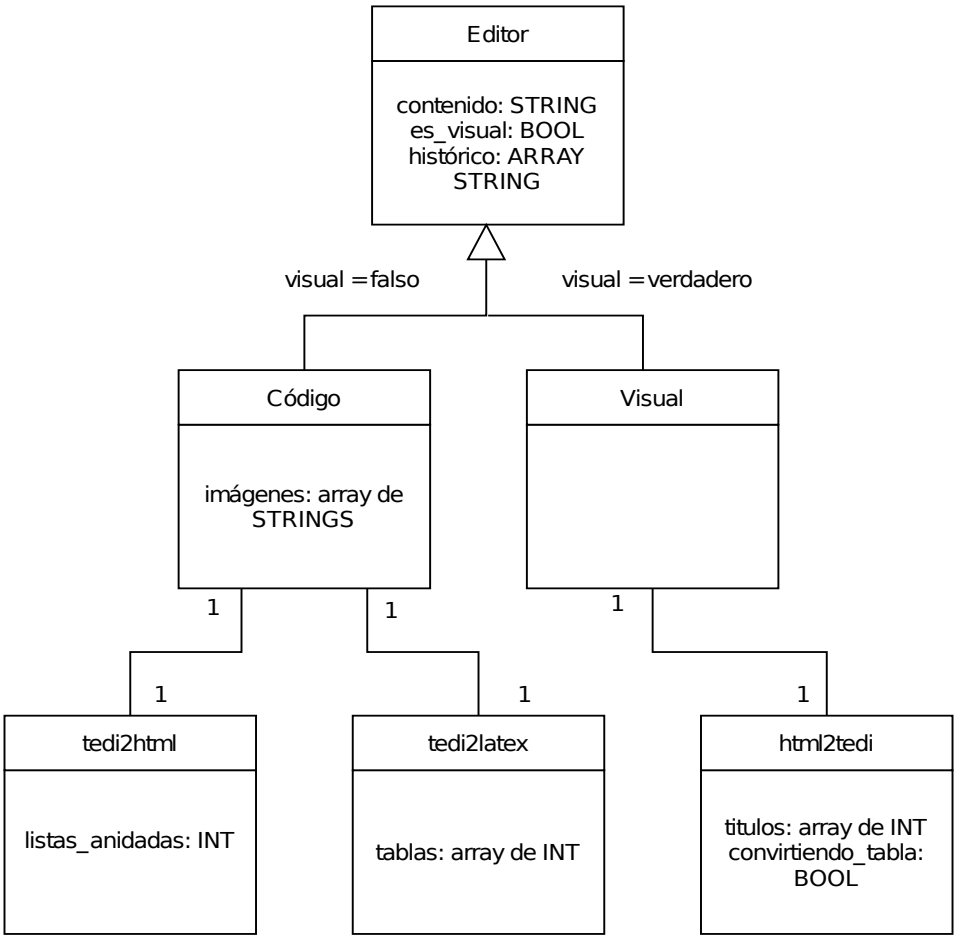



Figura 3.3: Modelo conceptual de datos

### 3.3.1. Entidades del modelo conceptual de datos

Las entidades presentes en el modelo conceptual de datos en la figura 3.3 se detallan a continuación:

- Editor: Clase que modela de forma abstracta los dos tipos de editores presentes en el sistema.
  - Código: Especialización del editor para su versión de sólo código, sin etiquetas HTML, pensado para mostrar el lenguaje intemedio.

    - Visual: Especialización del editor para su versión de sólo código, con etiquetas HTML, pensado para mostrar de forma visual lo que representan las etiquetas del lenguaje intermedio.

- tedi2html: Clase que encapsula la conversión de lenguaje intermedio a HTML.
- tedi2latex: Clase que encapsula la conversión de lenguaje intermedio a LaTeX.
- html2tedi: Clase que encapula la conversión de HTML a lenguaje intermedio.

# Capítulo 4

# Diseño del sistema

## 4.1. Arquitectura del sistema

En esta sección se describe la arquitectura física y lógica del sistema y se especificará la infraestructura tecnológica necesaria para el funcionamiento de la aplicación.

### 4.1.1. Arquitectura física

En esta sección de describen los elementos hardware que conforman la arquitectura física del sistema. Así obtenemos una visión más detallada de la estructura interna del sistema. Para ello, se describen los requisitos necesarios para su funcionamiento adecuado. La arquitectura física contempla las conexiones de los dispositivos del cliente con un servidor (o varios) centrales en el que se delegan ciertas funciones y de los que se obtiene el código a ejecutar. Por ello, para este proyecto tiene sentido usar el modelo cliente-servidor.

- El servidor requerirá cumplir ciertos requisitos para poder funcionar correctamente.
  - Este equipo deberá usar Windows, Linux o MacOS.
  - El equipo necesita un servidor web como Apache o Nginx para publicar el cliente web.
  - El equipo usa NodeJS para la API.

- El cliente requerirá un navegador web. Siendo mínimo un navegador capaz de ejecutar Javascript de versión ES5. Como recomendado, las últimas versiones de los navegadores más conocidos como Firefox y Chrome.

En la figura 3.3 se muestra el diagrama de despliegue del sistema. Este diagrama esquematiza la arquitectura del sistema como el despliegue de los distintos componentes necesarios para la plataforma. Con el podemos apreciar como la conexión entre el cliente y el servidor es de tres formas distintas, mediante la API, mediante websockets para compartir el documento con otros usuarios y con HTTP para acceder a la página. El sistema operativo del servidor puede ser cualquiera de los principales MacOS, Windows o Linux. La plataforma utiliza un software distinto para cada conexión, en el caso de websockets es controlado por TogetherJS mediante un servidor NodeJS, para el caso de HTTP es un servidor web como Apache y para el caso de la API es otra aplicación de NodeJS con diferentes dependencias de la de TogetherJS.

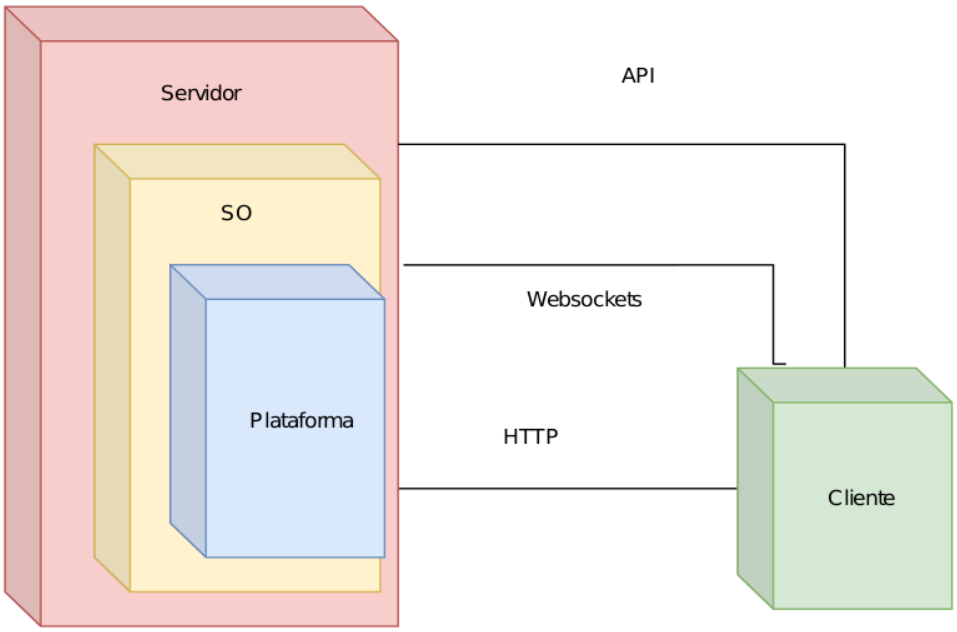



Figura 4.1: Diagrama de despliegue

### 4.1.2. Arquitectura lógica

La arquitectura de diseño especifica la forma en que los artefactos software interactúan entre sí para lograr el comportamiento deseado en el sistema.

El patrón arquitectónico utilizado para el desarrollo de la aplicación es la arquitectura por capas [11], encontramos las siguientes:

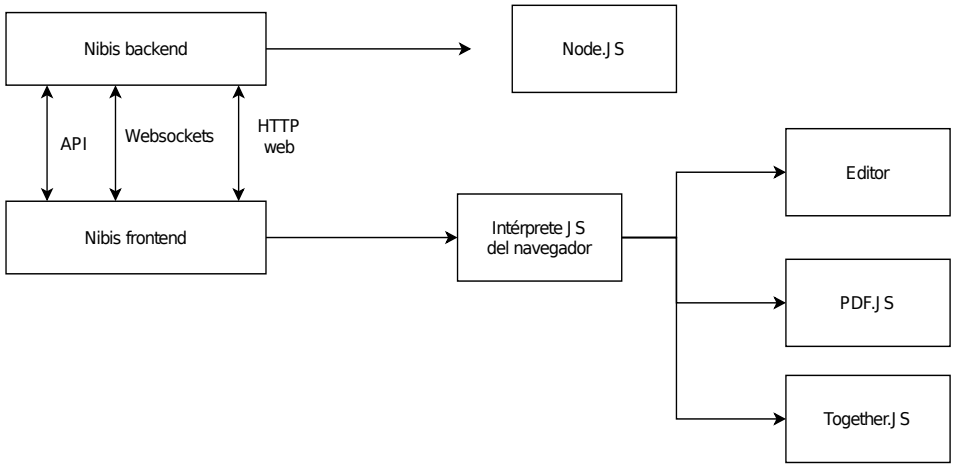



Figura 4.2: Arquitectura lógica de la plataforma

- Capa de presentación: Es toda la interfaz de usuario. Esta interfaz se ha hecho con HTML5 y CSS3. Aprovechando las últimas tecnologías de los navegadores pero siempre manteniendo compatibilidad con navegadores no tan actualizados.
- Capa de servicios: Conforma los servicios de Nibis, en este apartado se encuentra la API REST que depende de NodeJS.
- Capa de persistencia: No hay capa de persistencia. Todo el sistema ejecuta las peticiones y las envía en el momento, no se almacena información sobre el usuario en el servidor más allá de un posible sistema de protección de DDoS.

## 4.2. Patrones de diseño

Un patrón de diseño propone, explica y evalúa una solución para un problema de diseño que se da con mucha frecuencia a la hora de hacer una aplicación[13].

### 4.2.1. Tooltip

Es un elemento común de interfaz de usuario, en el cual, cuando se pasa el ratón por encima un elemento o componente muestra un texto explicativo de las funciones o de una abreviatura. El tooltip se seguirá mostrando mientras el usuario mantenga el ratón encima del elemento. Se pueden ver algunos ejemplos en el apartado 8.3

### 4.2.2. Atajos de teclado

Un atajo de teclado es una serie de una o varias teclas que invocan una acción en un programa. Puede haber sido introducido por el usuario, por el sistema operativo o por una aplicación para su uso[9]. En Nibis todas las etiquetas tienen un atajo asociado para poder usar el editor solo con teclado.

### 4.2.3. Arquitectura dirigida por eventos

La arquitectura dirigida por eventos es un patrón de la arquitectura software basada en la producción, detección, consumo y reacciones a eventos.[**?**] Un evento puede ser definido como un cambio significativo de estado. Todo el proyecto está pensado teniendo en mente la respuesta a eventos. Salvo unas funciones reducidas, el resto es una respuesta a un evento. El editor inserta una lista cuando el usuario pulsa el botón pertinente, si el usuario pulsa la barra espaciadora se añade un espacio en el elemento adecuado...

### 4.2.4. Convenciones de código

Las convenciones de código son importantes para el mantenimiento y comprensión del código fuente de un sistema. Es por esto que se ha decidido seguir en la medida de lo posible una serie de convenciones de código, para que la modificación del código fuente no se haga pesada y las secciones de

código sean fácilmente reconocibles. Estas son las convenciones que se han seguido en la medida de lo posible durante el desarrollo del proyecto:

- Variables: Para nombrar variables se usará la mayor cantidad de información posible sin sobrecargar al posible lector. Ya que la falta de contexto suele obligar a releer partes de códigos y esto es una pérdida de tiempo en el desarrollo y en las modificaciones.
  - Entre instrucciones de un mismo método que sirvan para propósitos distintos.
  - Entre estructuras condicionales y de repetición.
  - Entre métodos de una misma clase.
- Nombre de funciones: Los nombres de las funciones se harán al estilo C, así se pueden diferenciar de las funciones del propio lenguaje. Esta nomenclatura se hace separando las palabras por guiones bajos.

## 4.3. Diseño de la interfaz de usuario

En esta sección se detallarán los prototipos utilizados para diseñar la interfaz del sistema. Además se definirá el comportamiento de las pantallas más relevantes y las acciones que se disparan cuando el usuario haga uso de la interfaz. Este sistema se compone de una pantalla, en esa pantalla ocurre todo y a petición de los eventos de los usuarios añadirá u ocultará los elementos opcionales.

### 4.3.1. Wireframes

Los wireframes[12] describen el contenido de una página web y sus prioridades relativas. Son útiles para preveer la funcionalidad y el posible comportamiento de las distintas pantallas, o comunmente, distintas plantillas. El propósito es conocer las funcionalidades y el comportamiento. Ayuda a comenzar el desarrollo de la interfaz y aprovechar al máximo la interfaz desde el principio.

Tal como se puede observar en la ilustración 4.3 la interfaz a simple vista tiene las funcionalidades del editor visible en la pantalla para poder interactuar con ellas sin necesidad de pulsar varios apartados. Como objetivo la interfaz debe adaptarse a la pantalla para centrar al usuario en el editor y no en otros elementos.

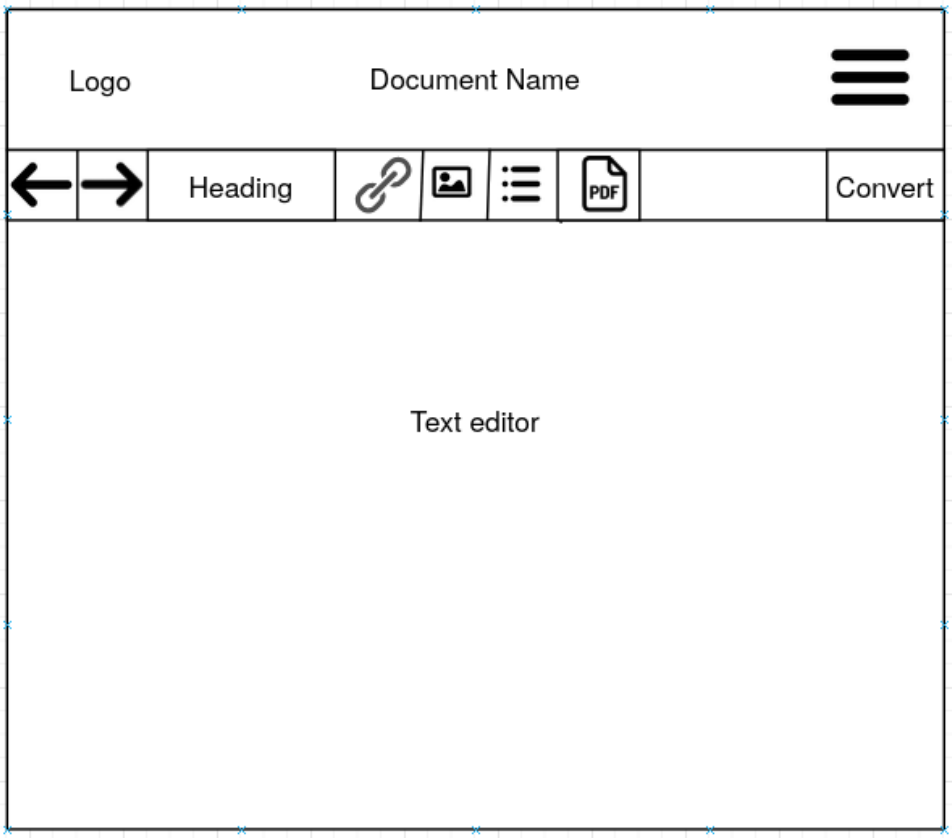



Figura 4.3: Wireframe para el estado inicial

En la ilustración 4.4 la interfaz se adapta para incorporar el visor de PDF. La plataforma obtiene el PDF a partir del texto del editor y al dividir la zona de escritura en dos, es más fácil revisar el documento que se acaba de crear.

de una o varias tareas concretas

A la hora de convertir al lenguaje intermedio y cambiar de editor, las funcionalidades se reducen ya que no es un editor WYSIWYG. Este pasa a mostrar lo que antes se mostraba tal como se verá en el documento como una previsualización a modo código, donde ya el usuario escribe las etiquetas por si mismo (véase Figura 4.5).

También se intentó organizar desde el principio una interfaz para dispositivos de pantallas con ancho reducido.

Los wireframe son una de las posibilidades de prototipado de baja fidelidad. Los prototipos de Baja Fidelidad implementan aspectos generales

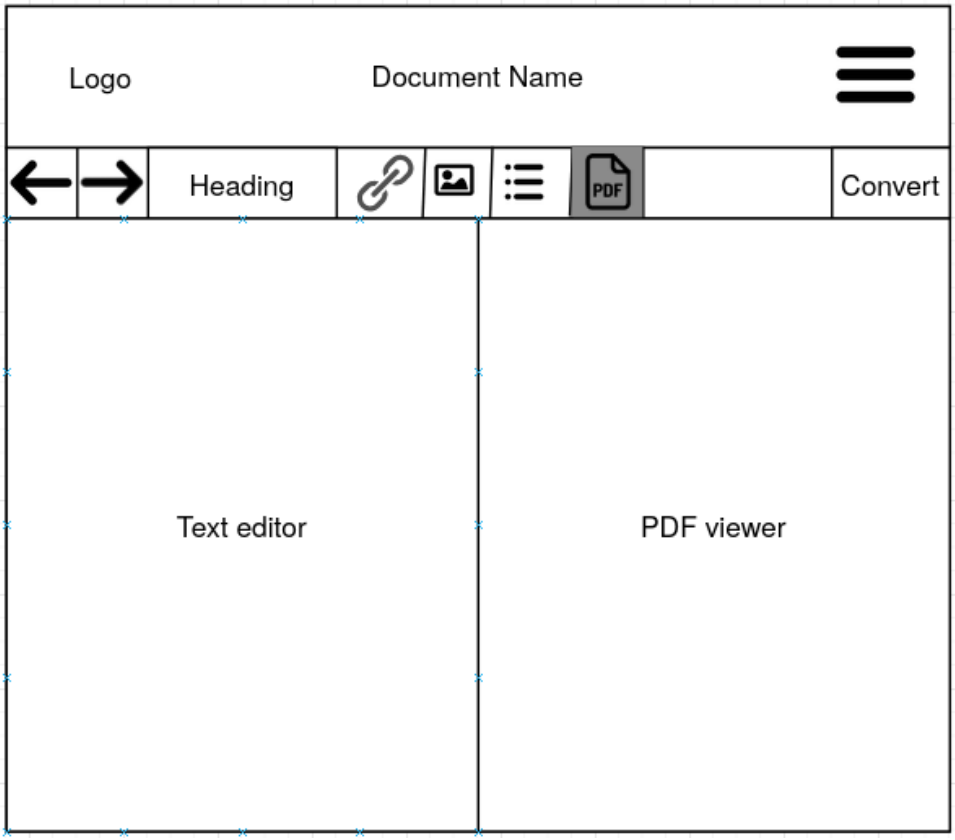



Figura 4.4: Wireframe para la conversión a PDF

del sistema sin entrar en detalles. Permiten abarcar un espectro mayor de la interacción a realizar. Al partir de este prototipo se puede mejorar poco a poco, incluyendo imágenes, combinación de colores, mejoras y nuevas secciones. Es la base en la que empezar a construir una interfaz solida y usable.

Como se pueden observar en las figuras, faltan muchos botones que se incorporaría finalmente en el proyecto. Cuando se hizo el prototipo, no se pensó en algunos aspectos que durante el desarrollo se observaron de suma importancia. Entre ellos son las ventanas emergentes para escribir ciertos valores necesarias para las funciones de cada botón, los elementos del menú desplegable y la sección de los títulos. En la parte final del desarrollo se añadió una página que explicara de forma simple qué es el proyecto y de qué herramientas depende.

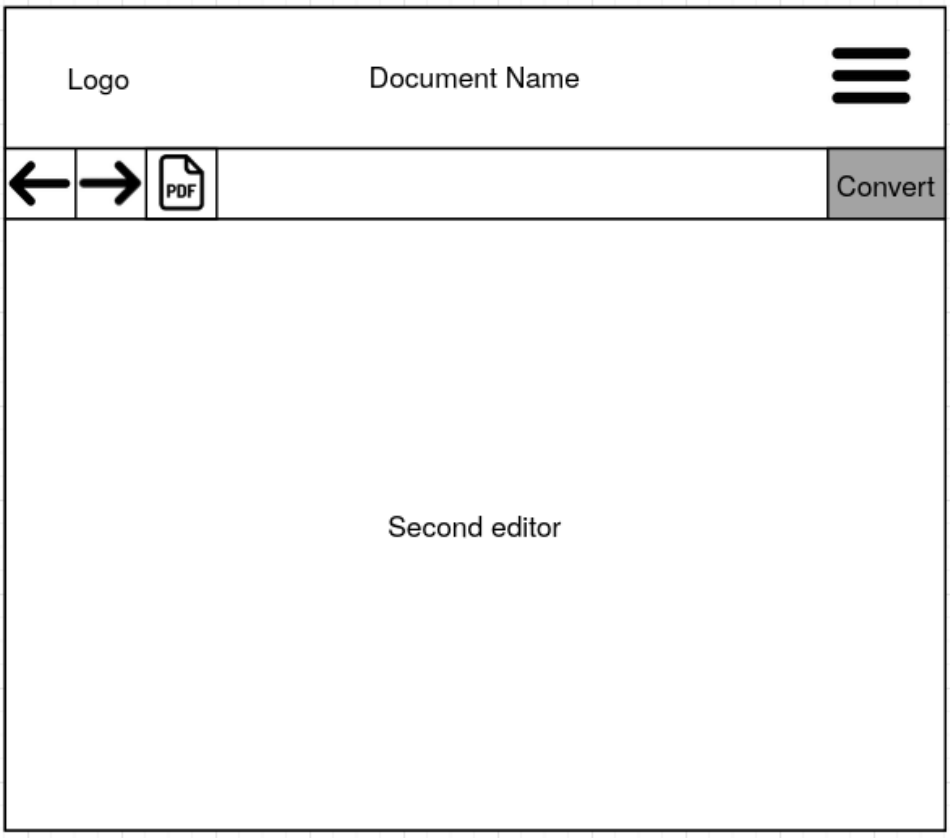



Figura 4.5: Wireframe para la conversión a lenguaje intermedio

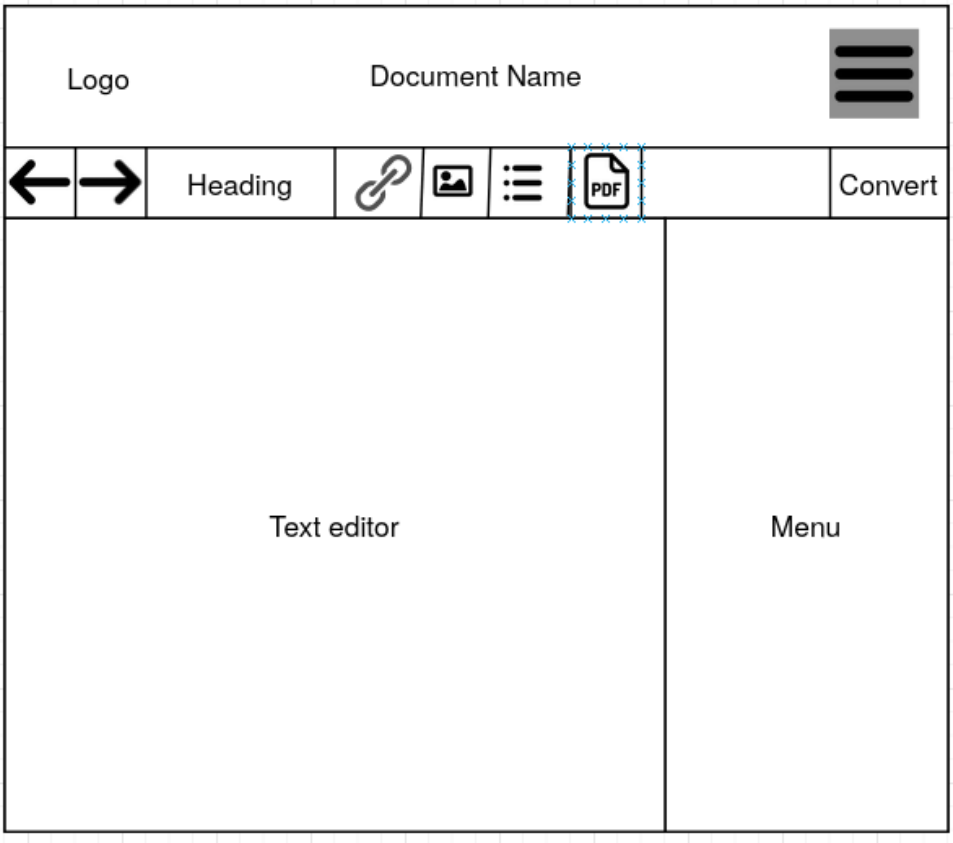



Figura 4.6: Wireframe para el menú

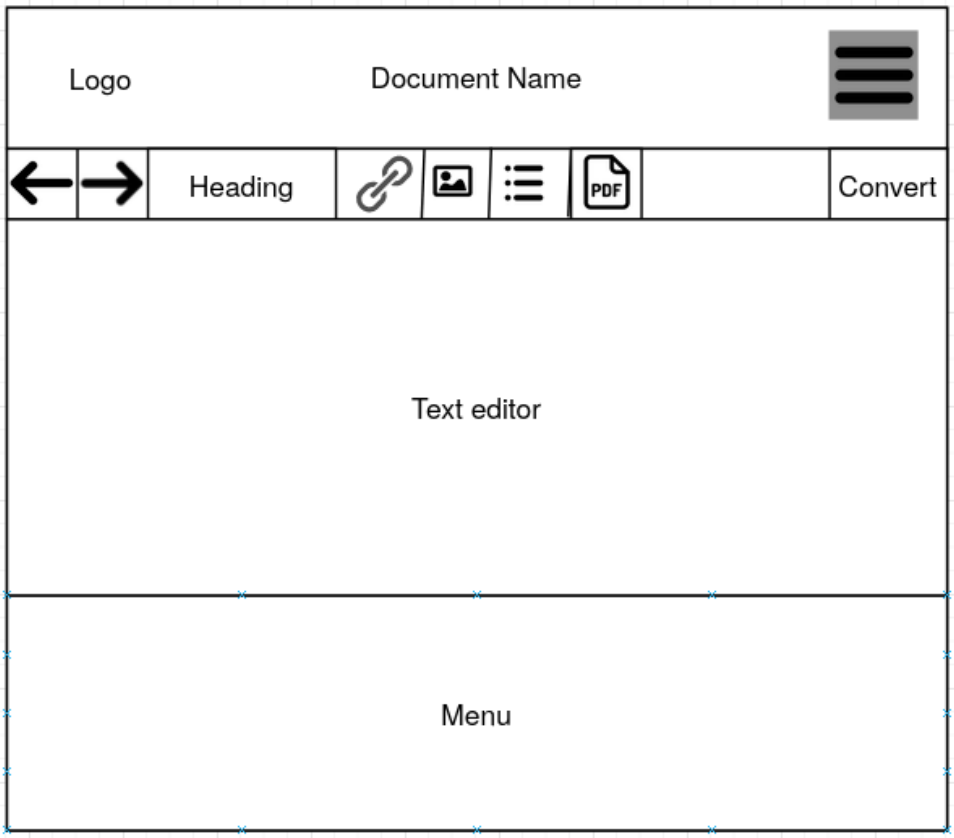



Figura 4.7: Wireframe para el menú en móvil

### 4.3.2. Prototipo de alta fidelidad

Tras haber realizado un prototipo que cumpla con todas las requisitos y estar enfocado al público objetivo, se puede empezar a construir el prototipo de alta fidelidad. Con los prototipos de Alta Fidelidad se representan aspectos más precisos. Sirven, por ejemplo, para detallar el proceso interactivo global de una o varias tareas concretas[13]. Para este prototipo no se usan unas imágenes simples para detallar las posibilidades. Este es el sistema que puede llegar a ser el final. Esta será la verdadera base de la interfaz. Por eso, se realizó directamente en HTML5 y CSS3[2.2].

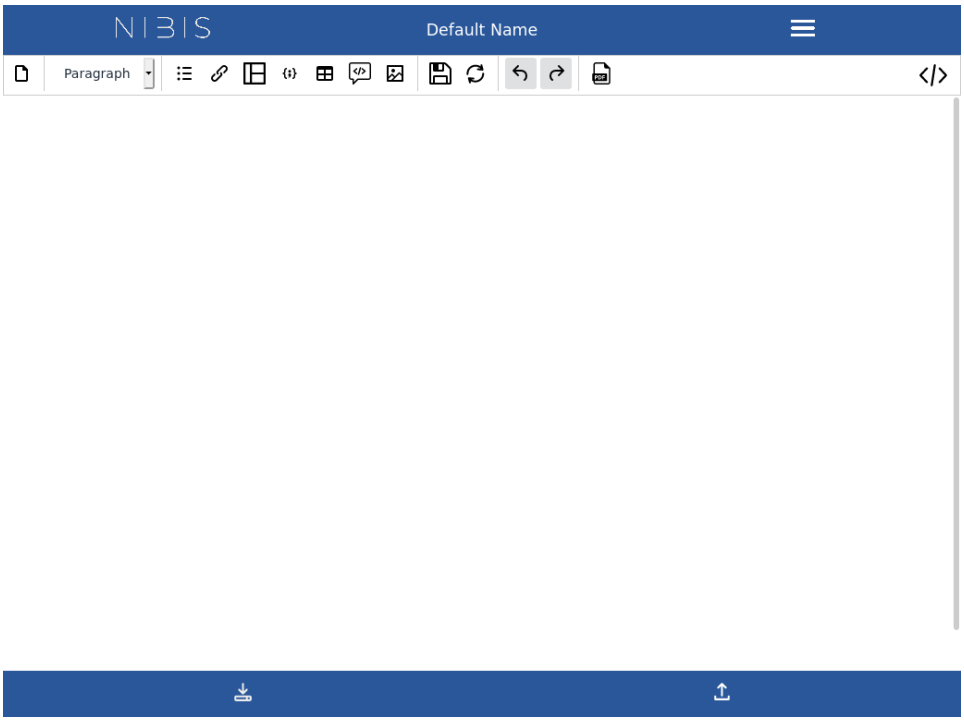



Figura 4.8: Página inicial del prototipo de alta fidelidad.

Se pueden ver claras mejoras desde el prototipo de baja fidelidad. Este ya añade todos los botones del editor salvo el botón de colaborar en el documento. Durante el desarrollo, se acabó añadiendo la organización para las pantallas de tamaño reducido mostrada en los wireframes. Antes, esta pantalla reducía su tamaño pero acababa por dificultar la escritura y lectura. Los elementos también se organizaron según utilidad y uso durante la escritura de un documento.

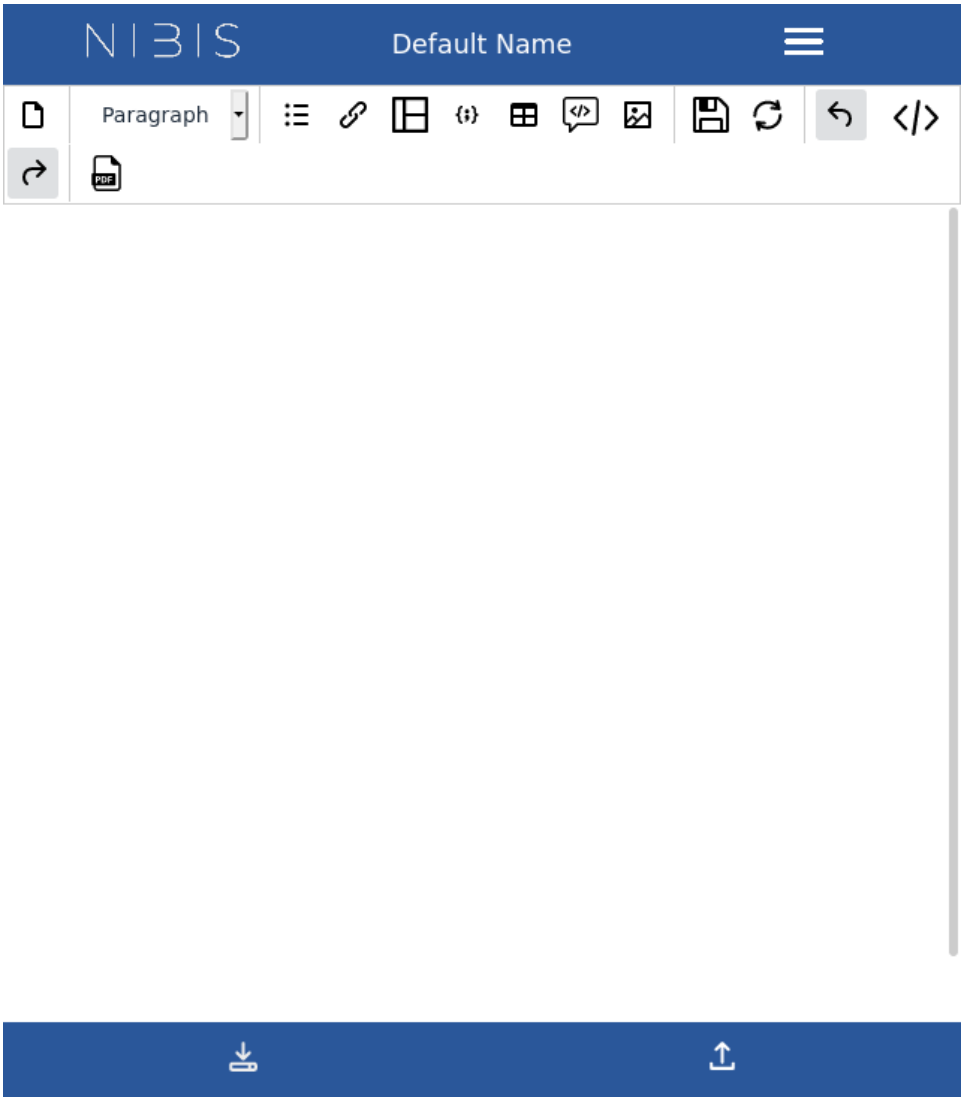



Figura 4.9: Página de ancho reducido del prototipo de alta fidelidad.

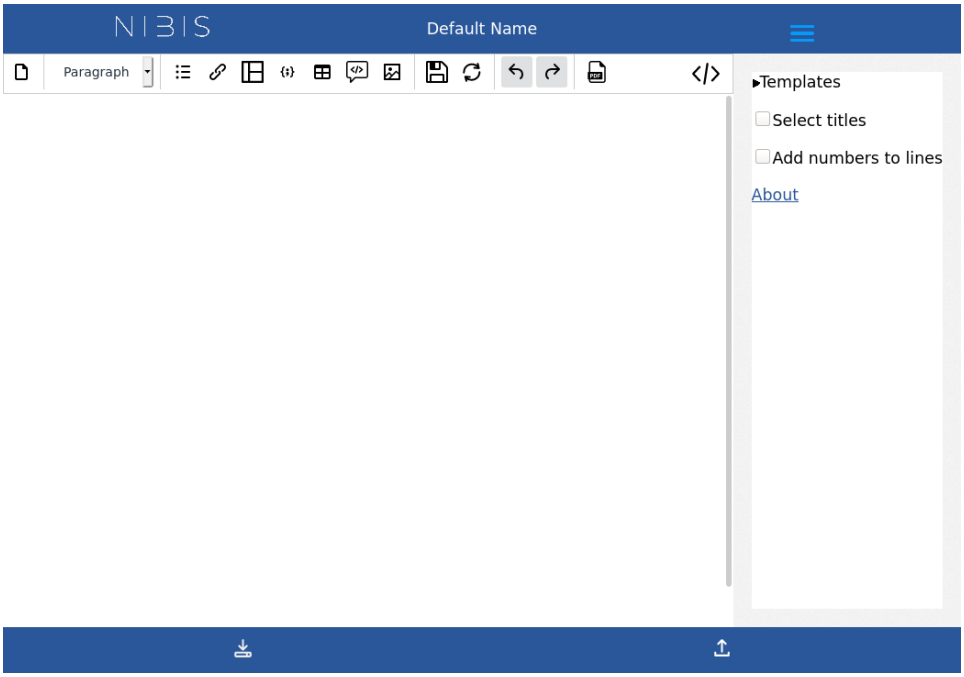



Figura 4.10: Página con menú del prototipo de alta fidelidad.

# Capítulo 5

# Implementación del sistema

## 5.1. Entorno de desarrollo

El desarrollo del proyecto se ha realizado mediante unos entornos de desarrollo explicados en las siguientes secciones.

### 5.1.1. Emacs

Emacs es un IDE[1] software libre desarrollado en Emacs Lisp y C. Tiene gran cantidad de herramientas y complementos que están algunos integrados en el propio software o que se pueden integrar fácilmente. Se ha primado Emacs frente a otros como Visual Studio por su comodidad para trabajar con LaTeX ya que, desde el principio, ha hecho falta entender bien el lenguaje para poder sustituirlo correctamente. Este editor tiene integrada una terminal Bash que nos permite lanzar mientras escribimos tareas al servidor de pruebas, trabajar con el sistema de ficheros o incluso programar tareas de limpieza o de publicación de código en Git. Aunque sería perfectamente posible usar otro editor, sin embargo, se ha utilizado este por costumbre y conocimiento sobre él.

[1] Entorno de desarrollo integrado

### 5.1.2. Herramientas adicionales

#### Notabug

[Este párrafo ha sido corregido para poner el repositorio original y eliminar servidores privativos]

Para descargar el código es necesario instalar Git en un ordenador y ejecutar en la consola lo siguiente:

```
git clone https://notabug.org/alkeon/Nibis
```

#### Node.js

es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript. Se prefirió este entorno por conocimiento previo, gracias a la asignatura de Programación en Internet, y debido a la popularidad del mismo que permite tener herramientas modernas y sencillas de usar.

#### PDF.JS

Librería para cargar archivos PDF directamente en el navegador. Librería realizada por Mozilla que permite cargar un PDF con elementos HTML, se encuentra incluída por defecto en su navegador y varios ejemplos en su página para poder utilizarlo en todos los navegadores con intérprete de Javascript moderno.

#### JQuery

jQuery es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

### TexLive

Distribución para el lenguaje TeX que contiene macros, fuentes y paquetes relacionados con TeX y LaTeX. Es la distribución por defecto en la mayoría de distribuciones Linux. Se escogió por su facilidad para automatizarse frente a sus contrapartes de Windows y MacOS. Dentro de esta distribución se usa XeTeX que permite el uso de UTF-8 para permitir otros idiomas que estándares anteriores no permiten.

### Chromium

Principalmente se ha usado Chromium que es la versión de código abierto de Google Chrome. Ambos usan el motor de renderizado Blink. Es de los navegadores más usado tanto en móvil como en escritorio. Principalmente por su uso, se prefirió este sobre Firefox aunque también se ha comprobado con este para tener mayor compatibilidad. Dispone de aislamiento de procesos, multiproceso, plugins para mejorar la experiencia y WebRTC.

### Together.JS

Librería de Javascript que permite añadir colaboración en la página web mediante WebSockets. Este librería detecta los cambios en el DOM de la página web y por defecto envía todos los cambios a un servidor central que enviará los cambios a todos los que estén suscritos a esa conexión WebSocket. Tiene una gran cantidad de funciones que hemos tenido que reducir ya que incluso permite ver un vídeo de Youtube simultáneamente. En Nibis usamos el servidor de TogetherJs de la empresa JSFiddle.

Durante el desarrollo se comprobó si era adecuado mantener un servidor propio. Al principio se planteó esta idea debido a que Mozilla abandonó el proyecto y su servidor oficial. Al final todas las modificaron se hicieron de lado del cliente y por ello se prefirió el servidor de JSFiddle. Aún así si fuera necesario, al ser software libre tan solo hay que montarlo en un servidor y cambiar la dirección principal que se encuentra en la página principal del proyecto.

# Capítulo 6

# Pruebas del sistema

La gran mayoría de pruebas han sido manuales. Esto tiene sentido cuando el proyecto usa herramientas de colaboración, conversión a PDF y está centrado en la usabilidad. Las pruebas permiten determinar adecuadamente si una conversión es adecuada, si las tablas se insertan adecuadamente o saber cuántas líneas permite como máximo. Sin hacer sombra a las pruebas automatizadas que muestran si al menos un mínimo de las funciones se realicen.

## 6.1. Entorno de pruebas

### 6.1.1. Entorno de pruebas del servidor

Se ha estado cambiando de servidor para poner a prueba y bajo estrés el proyecto realizado. El servidor en el que más tiempo se ha probado es

- Procesador: Intel(R) Atom(TM) CPU N280 1.66GHz
- Memoria RAM: 2GB
- Sistema operativo: Debian GNU/Linux 10

### 6.1.2. Entorno de prueba del cliente

Para entorno móvil:

- Modelo: LGE Nexus 5
- Sistema operativo: Android 10
- Procesador: Qualcomm Snapdragon 700
- Memoria RAM: 2GB

Para entorno de escritorio:

- Intel i7-2200K
- 8GB de RAM DDR3
- Memoria SSD 256GB
- Gráfica Nvidia Geforce GTX 550 Ti

### 6.1.3. Herramientas utilizadas

El software utilizado para las pruebas es QUnit. Este software es una herramienta sencilla pero a la vez muy vistosa y con una interfaz que te ayuda a solucionar los errores, mostrando la entrada recibida, la esperada y su diferencia.

Esta herramienta permite realizar pruebar unitarias aunque se ha aprovechado para determinar la cohesión y la integración de las distintas partes del proyecto.

### 6.1.4. Pruebas funcionales

Las pruebas funcionales comprueban la mayoría de los requisitos de forma automatizada. Salvo para las funciones de guardado y cargado que dependen de las funciones del navegador y pueden producir incompatibilidades de los tests. Estas pruebas están pensadas para ejecutarse en Firefox y se pueden comprobar en Chrome pero este último ordena los atributos de una manera diferente a Firefox.

Se han separado en tres apartados: Pruebas de editor, Pruebas de atajos de teclado y Pruebas de integración. Todas las pruebas tienen tres comprobaciones

Las pruebas de editor son 6.10 a 6.16 con ellas se comprueba el funcionamiento del editor sin conversiones intermedias. Estas pruebas son imprescindibles para que el editor funcione para las tareas más esenciales.

| | |
|---|---|
| PF-1 | Escribir una nueva línea |
| Descripción | Comprobamos la respuesta del editor al pulsar Intro |
| Entrada | Editor sin texto |
| Salida | Una nueva línea en el editor |

Cuadro 6.1: PF-01 (Nueva línea.)

| | |
|---|---|
| PF-2 | Eliminar una línea |
| Descripción | Comprobamos la respuesta del editor a pulsar Retroceso |
| Entrada | Editor con dos líneas |
| Salida | Editor con una sola línea |

Cuadro 6.2: PF-02 (Eliminar línea.)

| | |
|---|---|
| PF-3 | Crear una tabla |
| Descripción | Comprobamos si el editor crea correctamente las tablas |
| Entrada | Editor sin texto |
| Salida | Editor con una tabla 3x3 en la que se puede escribir. |

Cuadro 6.3: PF-03 (Insertar tabla.)

| | |
|---|---|
| PF-4 | Crear un enlace |
| Descripción | Comprobamos si el editor crea correctamente los enlaces |
| Entrada | Editor con el texto 'ejemplo' |
| Salida | Editor con un enlace que tiene escrito ejemplo y te lleva a https://example.org/ |

Cuadro 6.4: PF-04 (Insertar enlace.)

| | |
|---|---|
| PF-5 | Crear una lista |
| Descripción | Comprobamos si el editor crea correctamente las listas |
| Entrada | Editor con el texto 'ejemplo' |
| Salida | Editor con una lista |

Cuadro 6.5: PF-05 (Insertar lista.)

| | |
|---|---|
| PF-6 | Crear un contenedor |
| Descripción | Comprobamos si el editor crea correctamente los contenedores |
| Entrada | Editor con el texto 'texto incluido al contenedor' |
| Salida | Editor con un contenedor con el texto anterior insertado y la clase del contenedor como 'class' |

Cuadro 6.6: PF-06 (Insertar contenedor.)

| | |
|---|---|
| PF-7 | Usar la metaetiqueta |
| Descripción | Comprobamos si el editor crea correctamente los apartados 'code' y la metaetiqueta en el lenguaje intermedio. |
| Entrada | Editor con el texto 'ejemplo' |
| Salida | Editor con la metaetiqueta para no convertir la línea |

Cuadro 6.7: PF-07 (Usar la metaetiqueta)

| | |
|---|---|
| PF-8 | Incustrar código |
| Descripción | Comprobamos si el editor crea correctamente los div correspondientes y las etiquetas adecuadas |
| Entrada | Editor con el texto LaTeX |
| Salida | Editor con la etiqueta de incrustar código para no convertir la línea |

Cuadro 6.8: PF-08 (Usar la etiqueta de incrustar código)

| | |
|---|---|
| PF-9 | Insertar un título |
| Descripción | Comprobamos si el editor crea correctamente los títulos de nivel 1 |
| Entrada | Editor con el texto 'título' |
| Salida | Editor con un título de nivel 1 con el texto 'título' |

Cuadro 6.9: PF-09 (Insertar un título)

Las pruebas de integración son 6.10 a reftab:pf16. Estas pruebas verifican el funcionamiento del editor y de las conversiones intermedias. De ahí que se haya nombrado como pruebas de integración ya que el desarrollo de la conversiones no se hice a la par que el desarrollo del editor, estos fueron integrados cuando ya eran funcionales ambos. Por último se comprueban

| | |
|---|---|
| PF-10 | Conversión del editor vacío |
| Descripción | Comprueba la conversión del editor vacío. Necesario porque durante el desarrollo ha habido ocasiones que el editor vacio generaba errores. |
| Entrada | Editor sin texto |
| Salida | El mismo editor vacío |

Cuadro 6.10: PF-10 (Conversiones del editor vacío)

que los atajos de teclado funcionan. Los atajos de teclado son una funcionalidad extra que agilizan el uso del sistema pero no son una parte principal del desarrollo, aún así es conviente comprobarlos.

| | |
|---|---|
| PF-11 | Conversión del editor con una sola línea |
| Descripción | Comprueba la conversión completa de una línea. |
| Entrada | Conversión del editor con una sola línea |
| Salida | Resultado de la conversión al lenguaje intermedio y reconvertir a HTML. |

Cuadro 6.11: PF-11 (Conversión del editor con una sola línea)

| | |
|---|---|
| PF-12 | Conversión del editor con varias líneas |
| Descripción | Comprueba la conversión completa con varias líneas. |
| Entrada | Editor con varias líneas |
| Salida | Resultado de la conversión al lenguaje intermedio y reconvertir a HTML. |

Cuadro 6.12: PF-12 (Conversión del editor con una sola línea)

| | |
|---|---|
| PF-13 | Conversión de una imagen vacía |
| Descripción | Comprueba la conversión completa de una imagen. |
| Entrada | Editor con varias líneas |
| Salida | Resultado de la conversión al lenguaje intermedio y reconvertir a HTML. |

Cuadro 6.13: PF-13 (Conversión de una imagen vacía)

| | |
|---|---|
| PF-14 | Conversión de una lista dentro de una lista |
| Descripción | Comprueba la conversión completa de una lista anidada. |
| Entrada | Editor con varias listas anidadas. |
| Salida | Resultado de la conversión al lenguaje intermedio y reconvertir a HTML. |

Cuadro 6.14: PF-14 (Conversión de una lista dentro de una lista)

| | |
|---|---|
| PF-15 | Uso de la metaetiqueta en parte de una línea |
| Descripción | Comprueba la conversión completa de una metaetiqueta sin salto de línea. |
| Entrada | Editor con una etiqueta code en parte de la línea. |
| Salida | Editor con la misma línea tras sucesivas conversiones |

Cuadro 6.15: PF-15 (Conversión de una lista dentro de una lista)

| | |
|---|---|
| PF-16 | Uso de la metaetiqueta en una línea completa |
| Descripción | Comprueba la conversión completa de una metaetiqueta con salto de línea. |
| Entrada | Editor con una etiqueta code en una línea completa |
| Salida | Editor con la misma línea tras sucesivas conversiones |

Cuadro 6.16: PF-16 (Conversión de una lista dentro de una lista)

| | |
|---|---|
| PF-17 | Inserción de una tabla<br>por atajo de teclado |
| Descripción | Comprueba la conversión completa<br>de una tabla<br>mediante un atajo de teclado. |
| Entrada | Editor en estado inicial |
| Salida | Editor con una tabla 3x3 |

Cuadro 6.17: PF-17 (Inserción de una tabla por atajo de teclado)

| | |
|---|---|
| PF-18 | Inserción de un enlace<br>por atajo de teclado |
| Descripción | Comprueba la conversión completa<br>de un enlace<br>mediante un atajo de teclado. |
| Entrada | Editor en estado inicial |
| Salida | Editor con un enlace |

Cuadro 6.18: PF-18 (Inserción de una tabla por atajo de teclado)

| | |
|---|---|
| PF-19 | Inserción de una lista<br>por atajo de teclado |
| Descripción | Comprueba la conversión completa<br>de una lista<br>mediante un atajo de teclado. |
| Entrada | Editor en estado inicial |
| Salida | Editor con una lista |

Cuadro 6.19: PF-19 (Inserción de una tabla por atajo de teclado)

| | |
|---|---|
| PF-20 | Inserción de un contenedor por atajo de teclado |
| Descripción | Comprueba la conversión completa de un contenedor mediante un atajo de teclado. |
| Entrada | Editor con el texto 'ejemplo' |
| Salida | Editor con un contenedor con texto 'ejemplo' y clase 'class' |

Cuadro 6.20: PF-20 (Inserción de un contenedor por atajo de teclado)

| | |
|---|---|
| PF-21 | Inserción de la metaetiqueta por atajo de teclado |
| Descripción | Comprueba la conversión completa de la metaetiqueta mediante un atajo de teclado. |
| Entrada | Editor con el texto 'ejemplo' |
| Salida | Editor con la metaetiqueta con texto 'ejemplo' |

Cuadro 6.21: PF-21 (Inserción de la metaetiqueta por atajo de teclado)

| | |
|---|---|
| PF-22 | Inserción de código incrustado por atajo de teclado |
| Descripción | Comprueba la conversión completa de código incrustado mediante un atajo de teclado. |
| Entrada | Editor con el texto 'ejemplo' |
| Salida | Editor con el código incrustado con texto 'ejemplo' |

Cuadro 6.22: PF-22 (Inserción del código incrustado por atajo de teclado)

## 6.2. Calidad de producto

Este sistema ha de cumplir con los estándares de la W3C para HTML y CSS con unos mínimos de calidad en el producto. Por ello, durante el desarrollo se usaron distintas herramientas de análisis estático de software.

Para el caso de HTML y CSS se usaron las herramientas disponibles por la propia W3C que son de acceso gratuito y de gran utilidad para comprobar distintos navegadores. Nu HTML Checker se usó para comprobar que el código HTML de todo el sistema cumple con el estándar HTML5 presentado en 2008. A su vez, W3C CSS Validator para CSS3 que comenzó a publicarse en 1999. Estas comprobaciones se realizaron antes de subir el proyecto a la plataforma, pero antes de presentarse se volverá a pasar.

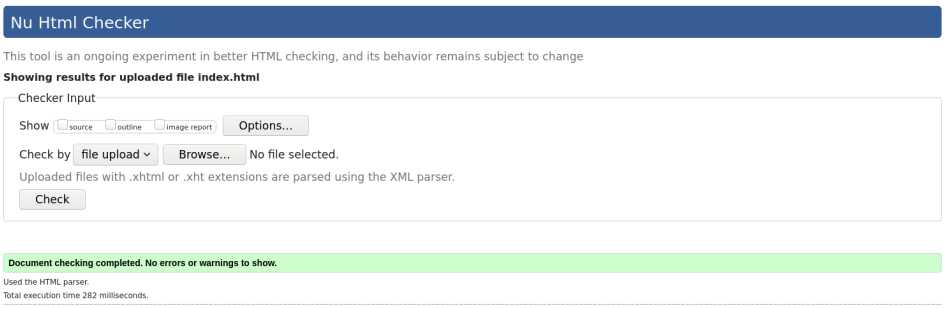



Figura 6.1: Prueba completa de validador HTML5 de W3C.


Figura 6.2: Resultado del validador de CSS3 de W3C.

## 6.3. Evaluación con usuarios

El test con usuarios es una prueba de usabilidad enmarcada en el enfoque de diseño centrado en el usuario. Consiste en la evaluación de un sitio web a partir de la observación, expresión verbal e interacción de los usuarios

mientras ejecutan las tareas propias de un producto o servicio interactivo.[15] En este proyecto la evaluación con usuarios ha sido la base para todo el desarrollo. Conforme se iban introduciendo funcionalidades complejas o que requieren de la respuesta de un usuario se han aumentado el número de evaluaciones con usuarios. El software ha ido evolucionando conforme los distintos usuarios encontraban problemas para utilizar el proyecto.

Las evaluaciones consistían en dos posibles escenarios. El usuario solo tiene que realizar uno, puede copiar un documento que contenía distintas imágenes, tablas, enlaces, referencias bibliográficas que aprovechaban todas las funcionalidades o podían probar el editor a su manera y ritmo.

El primer escenario es uno mucho más controlado que centra al participante en conseguir un objetivo, pero obliga al usuario a realizar unas tareas en un orden que quizás no es el más adecuado. El segundo es más cercano a la situación de un usuario que aprende de forma más natural para qué vale cada herramienta.

### 6.3.1. Tareas del usuario

Todos los tests con usuarios han sido en remoto debido a la situación sanitaria en la que nos encontramos. El usuario deberá compartir pantalla del ordenador para observar las distintas funcionalidades que va utilizando y cuáles no necesita. Esta evaluación se realiza tanto en un ordenador como en un móvil o tablet. Para terminar, el cuestionario post-test se realizará dentro de la plataforma. Al usuario se le envía un documento para cargarlo en Nibis y responder a las preguntas mientras lo usa.

### 6.3.2. Cuestionario post-test

Tal como se ha explicado en la introducción de esta sección esta evaluación se ha ido realizando desde el primer prototipo hasta el proyecto final. Por ende, este cuestionario ha ido incrementando el número de preguntas por los resultados de evaluaciones anteriores y de nuevas funcionalidades que necesitan mejorarse[16].

- Versión de escritorio
  - ¿Qué te ha parecido la prueba?
  - ¿Qué no comprendiste?
  - ¿Qué le añadirías al proyecto?

    - ¿Qué eliminarías al proyecto?
    - ¿Tienes alguna opinión o comentario que añadir?
    - ¿Has encontrado algún error?
    - ¿Qué eliminarías?
    - ¿Alguna opinión o comentario que añadir?
- Versión móvil
    - ¿Has encontrado algún error?
    - ¿Qué añadirías o mejorarías?

Estas preguntas las respondía cada usuario tras realizar la evaluación. Servían de refuerzo para las mejoras, eras unas preguntas más de opinión para saber la valoración del usuario.

Este método y cada prueba realizada al sistema han sido las bases con las que se ha construido este sistema.

### 6.3.3. Resultados obtenidos

Todo el proyecto ha sido probado por al menos 15 personas distintas que el rango de edad ronda los 19-50 años. Cada respuesta apuntaba a un aspecto distinto en el sistema. En el caso de algunos usuarios llegaron a colapsar la plataforma o incluso bloquear el navegador por ciertas funcionalidades incompletas o comportamientos inesperados.

La gran mayoría se centraban en el editor y en las conversiones entre el lenguaje intermedio a HTML. Los errores encontrados por los usuarios han sido:

- El editor permitía duplicar imágenes y tablas aprovechando la implementación drag and drop de los navegadores que no está finalizada en los contenedores contenteditable. Esto permitía insertar tablas dentro de tablas que llevarían al navegador a bloquearse.
- El editor permitía escribir fuera de los contenedores lo que incumplía la propia definición del programa y eliminaba las líneas escritas fueras de los contenedores.
- El usuario podía pegar texto dentro de texto, imágenes e incluso de los botones para eliminar las imágenes. Este fallo aprovechaba la implementación drag and drop y la de selección de texto que no son compatibles en Firefox con Blink.

- A la hora de insertar los elementos, la selección no detectaba correctamente el elemento en el que se encontraba porque usaba métodos experimentales.
- El botón de PDF no indicaba el estado en el que estaba la conversión, por ello los usuarios volvían a pulsarlo una y otra vez hasta bloqueaban el servidor. Ahora indica el estado cambiando el color de fondo y no permite ser pulsado mientras está convirtiendo una petición.
- Los atajos de teclado CTRL-C y CTRL-V provocaban que los usuarios anidaran elementos dentro de si mismos, entrada que el conversor no podía procesar.
- El atajo CTRL-V antes permitía incorporar elementos de otras páginas webs, esto suponía un problema serio porque incorporaba estilos extraños en la página o incluso scripts de terceros que empeoraban el rendimiento.
- Un error en la conversión traducía – como elemento sin importar en la posición o si está inicializada la lista.
- El interlineado entre los elementos de las listas era incorrecto, lo detectaron los usuarios al introducir textos de más de 10 líneas.
- Antes los enlaces solo funcionaban si el usuario seleccionaba un texto. Esto se observó que era poco usable porque todos los usuarios esperaban que si no se seleccionaba, se escribiera el propio enlace como texto del enlace.
- Reestructurar los botones por la utilidad de los mismos para los usuarios finales. Antes la botonera del editor estaba ordenada según se fueron implementando.

# Parte III

# Epílogo

# Capítulo 7

# Manual de instalación e explotación

## 7.1. Introducción

En este capítulo se explicará el proceso de instalación y explotación de esta plataforma.

## 7.2. Requisitos previos

Los requisitos hardware necesarios para la instalación y uso del servidor del sistema son los siguientes:

- Un ordenador con conexión a Internet.
- Al menos 2GB de RAM.

Los requisitos software son:

- NodeJS, junto a NPM.
- TexLive completo con todos sus complementos para XeLaTeX y todas las macros necesarias para usar LaTeX.
- eps2pdf para poder usar imágenes vectoriales en formato eps.

- Si se quiere usar la configuración automatizada hace falta Bash instalado.
- Un servidor web configurado ya sea este Apache o Nginx.

Los requisitos del cliente son:

- Un navegador actualizado, puede ser uno basado en Chromium o Firefox.
- Al menos 2GB de RAM.
- Procesador multinúcleo

### 7.2.1. Inventario de componentes

El sistema se compone de tres componentes, los cuales dos se pueden y deben instalar en el mismo servidor.

- La aplicación servidor de Nibis
- La aplicación servidor de TogetherJS
- La aplicación cliente de Nibis

## 7.3. Procedimiento de instalación

Tal y como se comentó en el Capítulo 5, el código fuente del servidor y del cliente de Nibis está alojado en GitHub, para descargarlo y compilarlo debemos hacer lo siguiente.

Si descargamos del git el archivo sh.sh, se puede usar la opción automatizada que se explicará en el apartado de Inicio de las aplicaciones.

### 7.3.1. Instalación de la API

Se detallarán las instrucciones para instalar la plataforma para los sistemas que no tienen instalado Bash. El código tanto del cliente como del servidor se encuentran en https://notabug.org/alkeon/Nibis.

```
git clone https://github.com/a-castilla/Nibis
```

```
cd Nibis/api
```

```
npm install
```

### 7.3.2. Instalación de servidor TogetherJS

En la plataforma usamos un servidor montado por JSFiddle, pero se debe poder mantener si falla.

```
git clone git://github.com/mozilla/togetherjs.git
```

```
    cd togetherjs
```

```
npm install
```

```
npm install -g grunt-cli
```

Estas son las instrucciones proporcionadas por Mozilla para TogetherJS. De este sistema necesitaremos la IP para introducirla en la configuración del cliente.

### 7.3.3. Inicio de la API

Para ejecutar la API tal solo hará falta escribir

```
node index.js
```

Esto lanzará la API y se encontrará esperando en el puerto indicado. Por defecto 8080.

### 7.3.4. Configuración del cliente

El cliente necesita cierta información del servidor para funcionar. Las configuraciones más importantes del cliente son:

- Dirección IP de la API escrita en el fichero ip.js, que se puede cambiar en la variable 'address'
- Dirección IP o dominio en 'index.html' en la variable de configuración `TogetherJSConfig_hubBase` , cambiando este valor, te permite usar otro servidor de TogetherJS o Hub como lo conoce la documentación de Mozilla.

### 7.3.5. Configuración automatizada

El script de shell permite simplificar el proceso anterior para que tan solo escribiendo la IP, descargue si hace falta el proyecto completo y le configure la IP personalizada para la API. Solo hace falta escribir en la terminal

```
bash sh.sh IP
```

Paso a paso realizará lo anterior mucho más rápido y sencillo para el usuario.

# Capítulo 8

# Manual de usuario

El objetivo principal de Nibis es crear documentos profesionales para particulares o empresas. Así pues Nibis proporciona a los usuarios una plataforma mediante la cual generar documentos y editarlos de una manera sencilla.

## 8.1. Características

La funcionales que ofrecen Nibis son:

- Creación de documentos a petición del usuario
- Un editor que requiere pocos recursos para funcionar
- Conversión a un lenguaje de marcado para guardados y funciones propias del usuario
- Añadir enlaces, títulos, imágenes directamente en el navegador sin necesidad de un servidor tipo cloud
- Uso de plantillas para permitir distintos tipos de documentos
- Compartir y editar en tiempo real el documento mediante TogetherJS

## 8.2. Requisitos previos

Cualquier dispositivo que tenga un navegador actualizado como los basados en Chromium o Firefox.

## 8.3. Uso del sistema

En esta sección se va a describir todos los aspectos necesarios para una utilización efectiva y eficiente del sistema por parte de los usuarios, para ello se van a explicar las funcionalidades más relevantes de Nibis.

### 8.3.1. Insertar título en el editor

En el panel de funciones del editor podemos encontrar un desplegable con el texto 'Paragraph' en la parte superior del editor. Está pensado para insertar un título en la posición donde el usuario esté escribiendo.

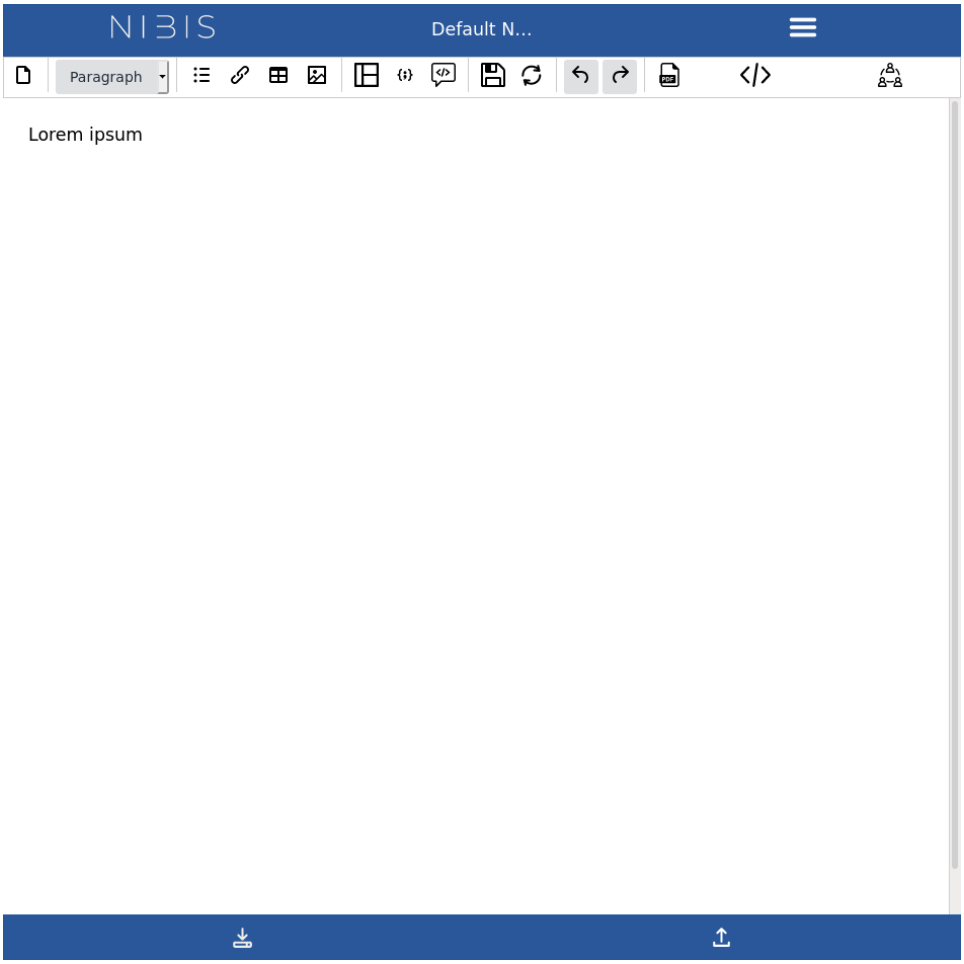



Figura 8.1: Ratón por encima del desplegable de los títulos.

El desplegable contendrá distintas opciones de títulos. Si se elige una

opción cambiará la línea al nivel de título seleccionado.

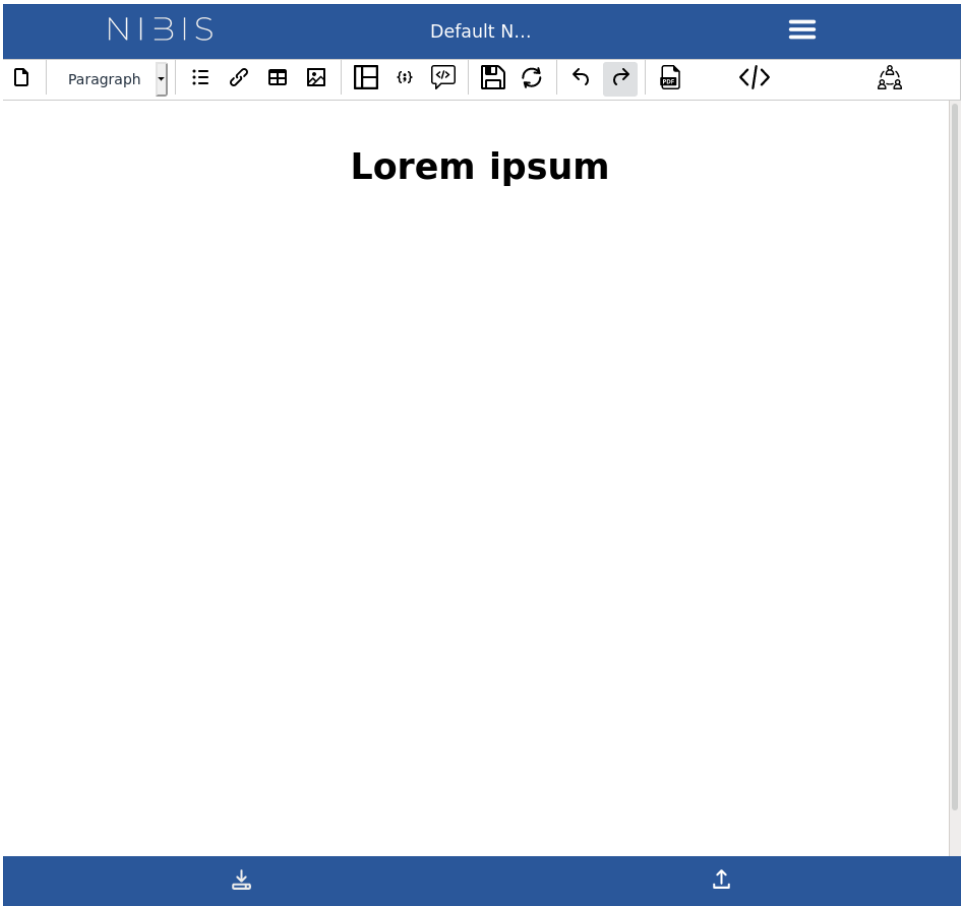



Figura 8.2: Resultado de cambiar la línea a un nivel de título.

### 8.3.2. Insertar lista en el editor

En el panel de funciones del editor podemos encontrar un botón con tres líneas horizontales en la parte superior del editor. Está pensado para insertar una lista en la posición donde el usuario esté escribiendo.

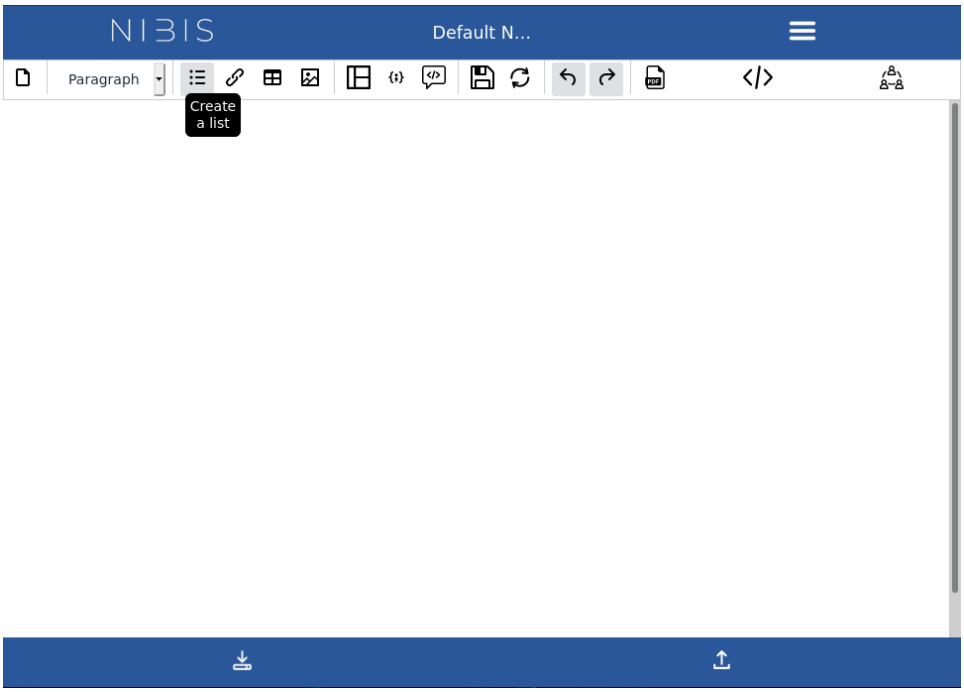



Figura 8.3: Ratón encima del botón de lista

Este botón insertará una lista en la posición que tenga seleccionada el usuario. La posición seleccionada se cambiará a la lista y también la parte visible del editor para que el usuario pueda ver la nueva lista incorporada y estar preparado para escribir directamente.

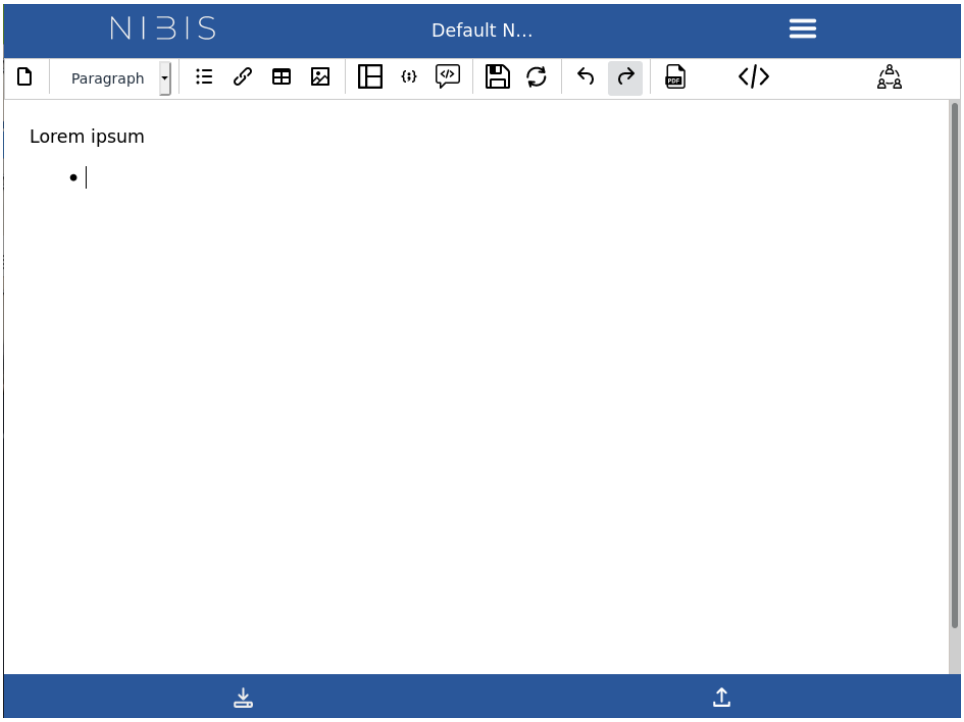



Figura 8.4: Lista insertada adecuadamente

### 8.3.3. Insertar enlace en el editor

En el panel de funciones del editor podemos encontrar un botón con dos eslabones de una cadena. Este botón nos permite crear enlaces en el editor.

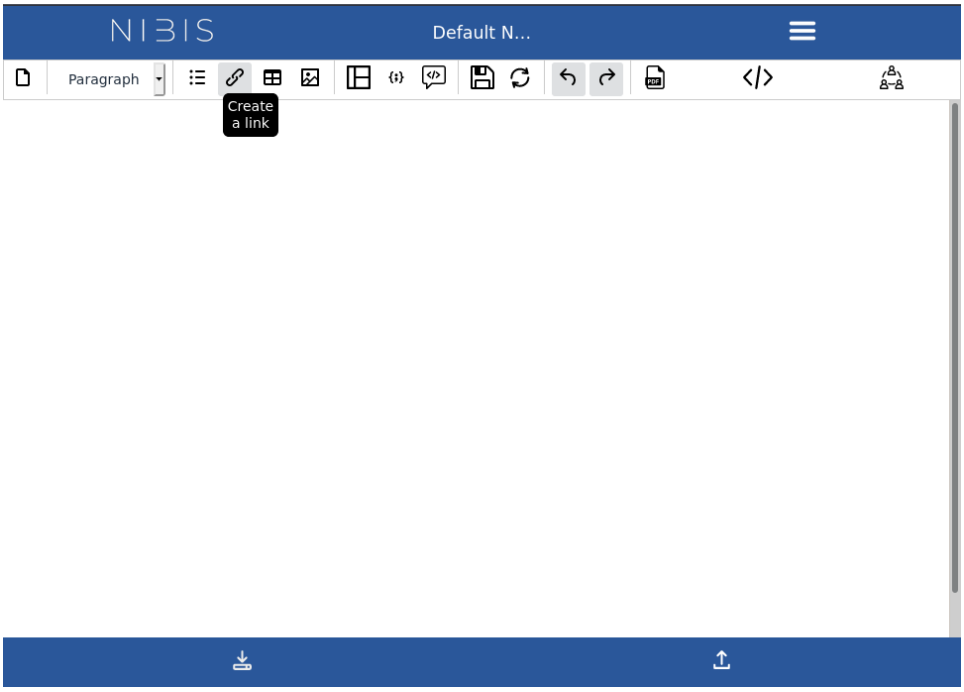



Figura 8.5: Ratón encima del botón de enlaces

Este enlace se añadirá en la posición seleccionada del editor o en su defecto en la última línea. Antes el usuario deberá escribir el enlace en un diálogo emergente.

Si no escribe el protocolo se supondrá por defecto HTTPS. Si tenía resaltado text, se le añadirá a ese texto el enlace, si no se pondrá el enlace como texto al hipervínculo.

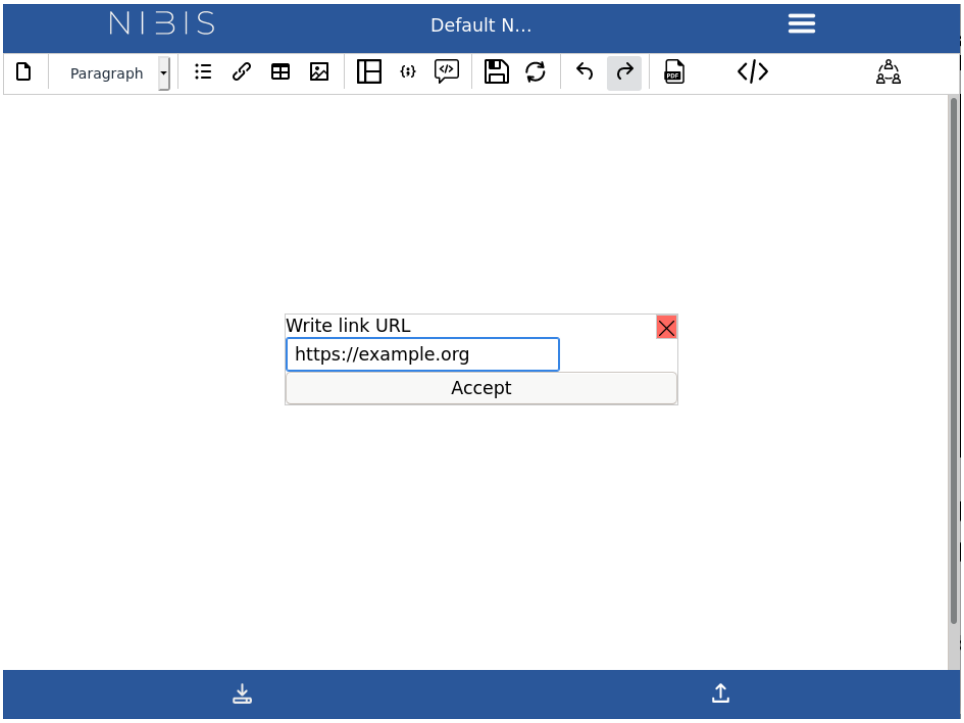



Figura 8.6: Ratón encima del botón de enlaces

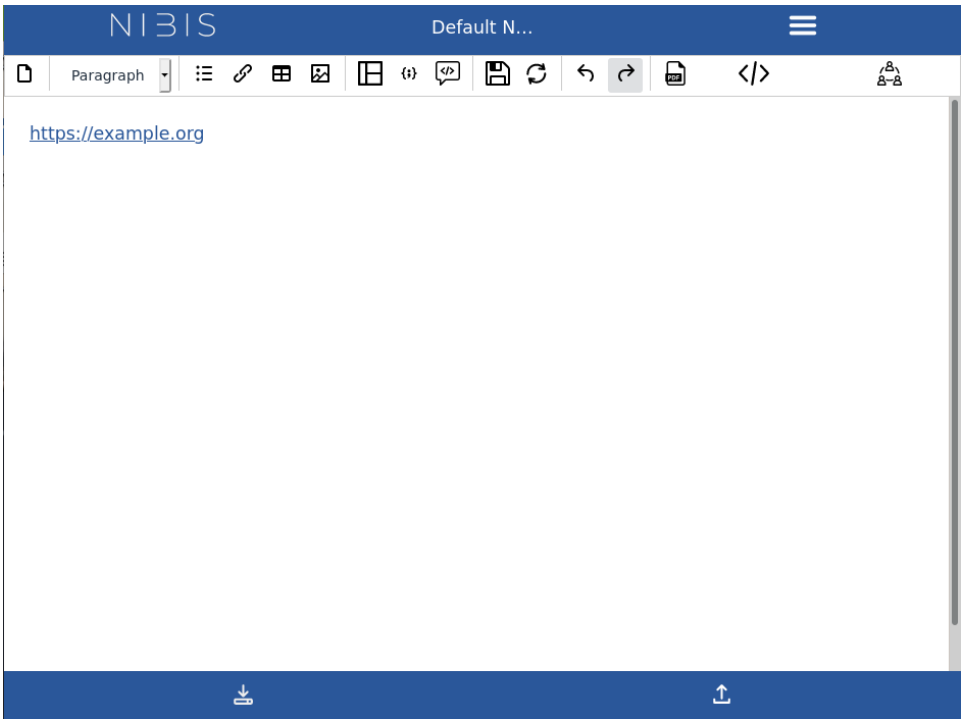



Figura 8.7: Ventana emergente que pide una URL al usuario

### 8.3.4. Insertar tabla en el editor

En el panel de funciones del editor podemos encontrar un botón con una tabla de 2x2. Las tablas se encuentran limitadas hasta 19x19. Está pensado para insertar una tabla en la posición donde el usuario esté escribiendo.

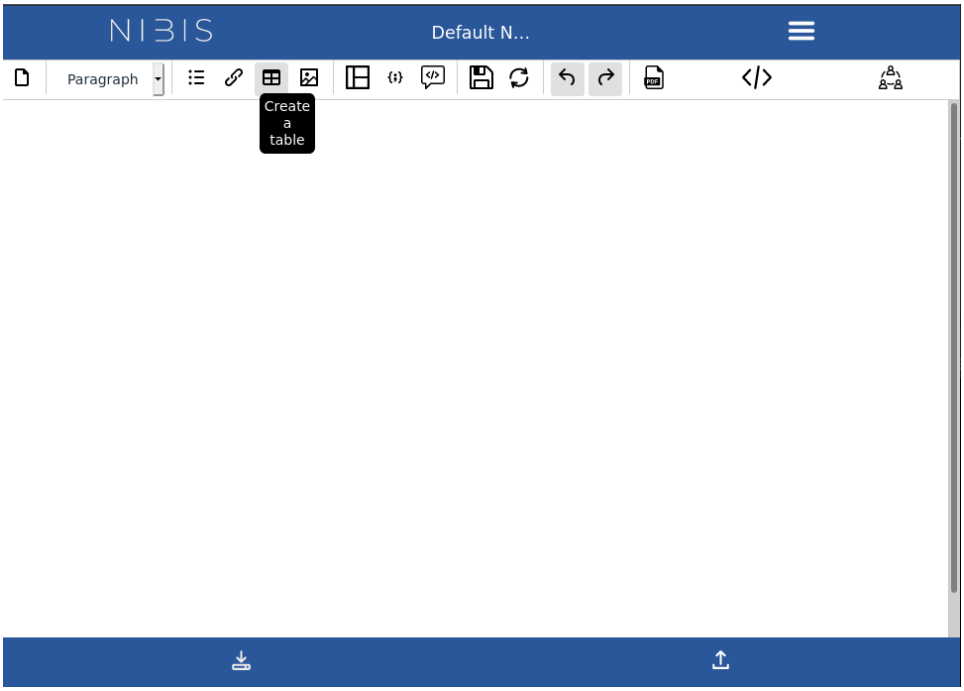



Figura 8.8: Ratón encima del botón de insertar tabla.

Tras seleccionar el botón saldrá un diálogo emergente que pregunta por el ancho y alto de la tabla. La tabla se generará dependiendo de estos valores introducidos por el usuario. Si no son correctos no dejará continuar.

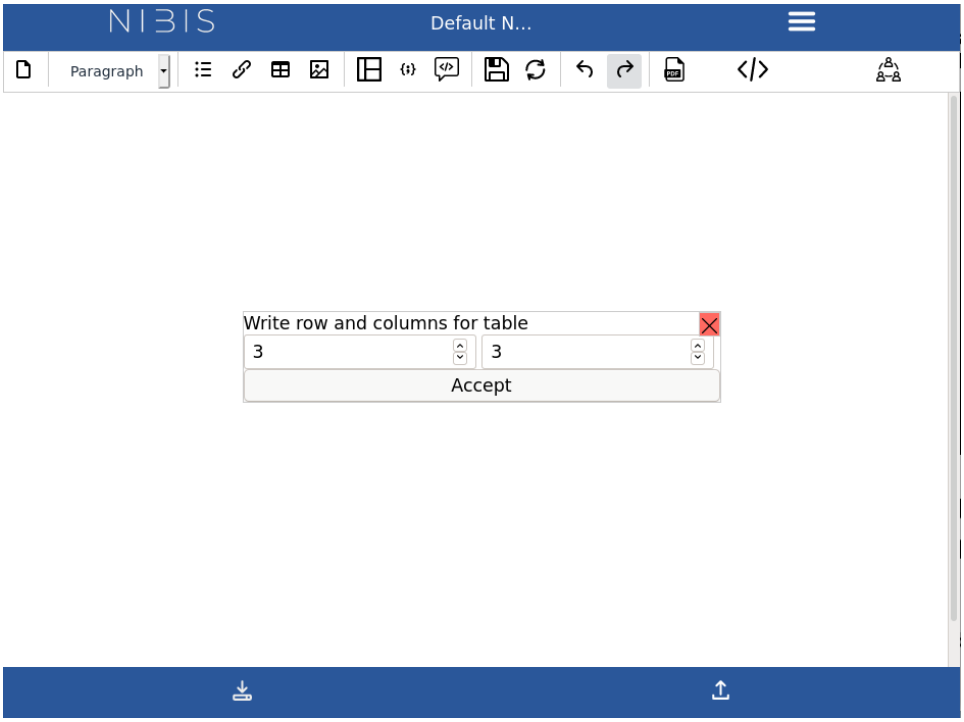



Figura 8.9: El diálogo emergente de alto y ancho.

Tras cumplimentar los campos y el usuario acepta se generará una tabla del tamaño asignado. En este caso de un tamaño de 3x3.

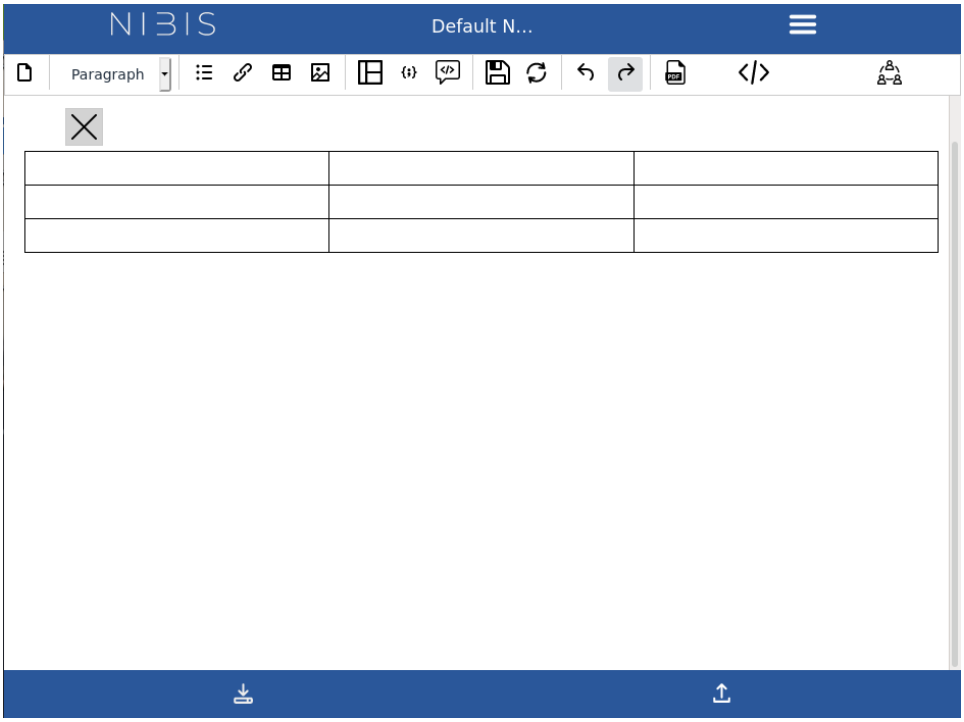



Figura 8.10: La tabla nueva completamente insertado

### 8.3.5. Insertar imagen en el editor

En el panel de funciones del editor podemos encontrar un botón con el icono de una imagen con un paisaje.

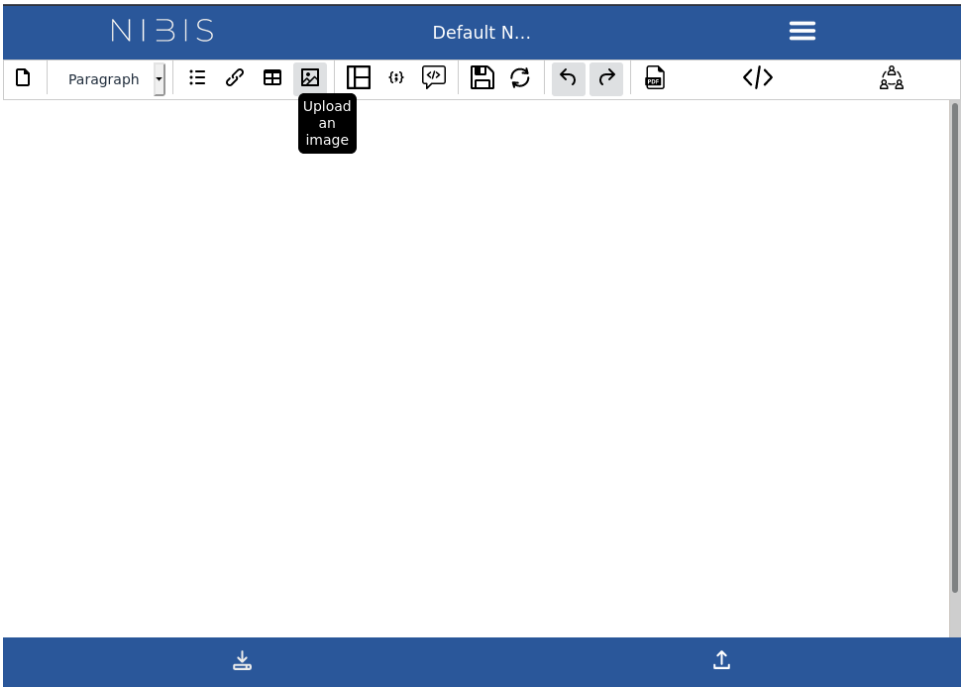



Figura 8.11: Ratón encima del botón de insertar imagen.

Tras pulsar sobre el botón aparecerá una ventana nueva que pide al usuario seleccionar la imagen de su sistema de ficheros. Esta ventana depende del sistema operativo del dispositivo y de la configuración del mismo.

Tras seleccionar el fichero saldrá un diálogo emergente que pide al usuario escribir el texto alternativo para la imagen que se mostrará como leyenda en el PDF.

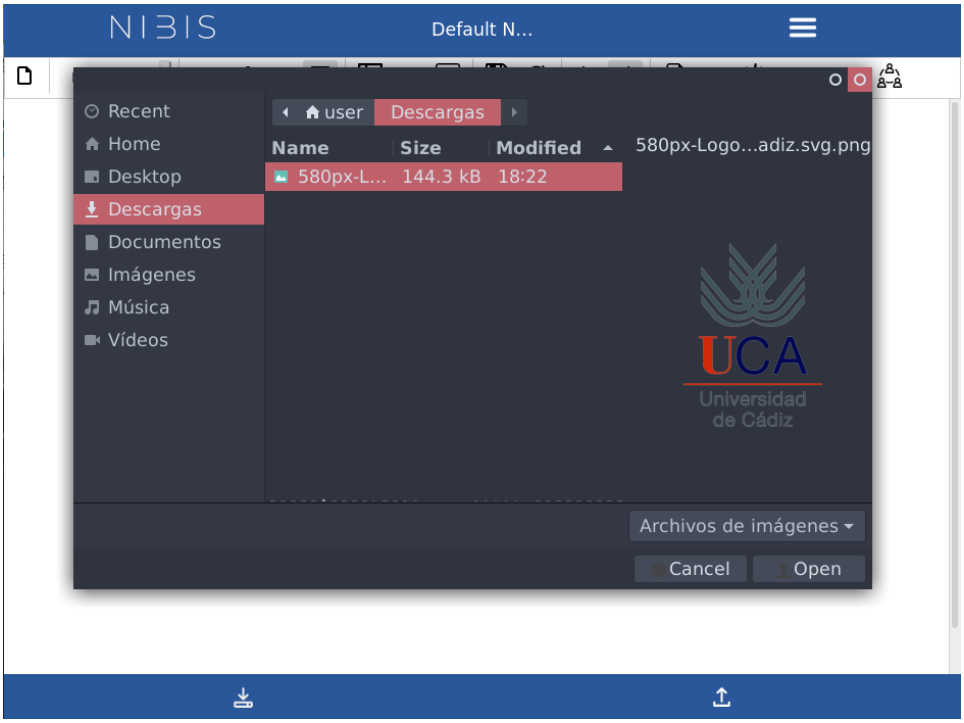



Figura 8.12: Diálogo emergente del sistema de archivos del dispositivo.

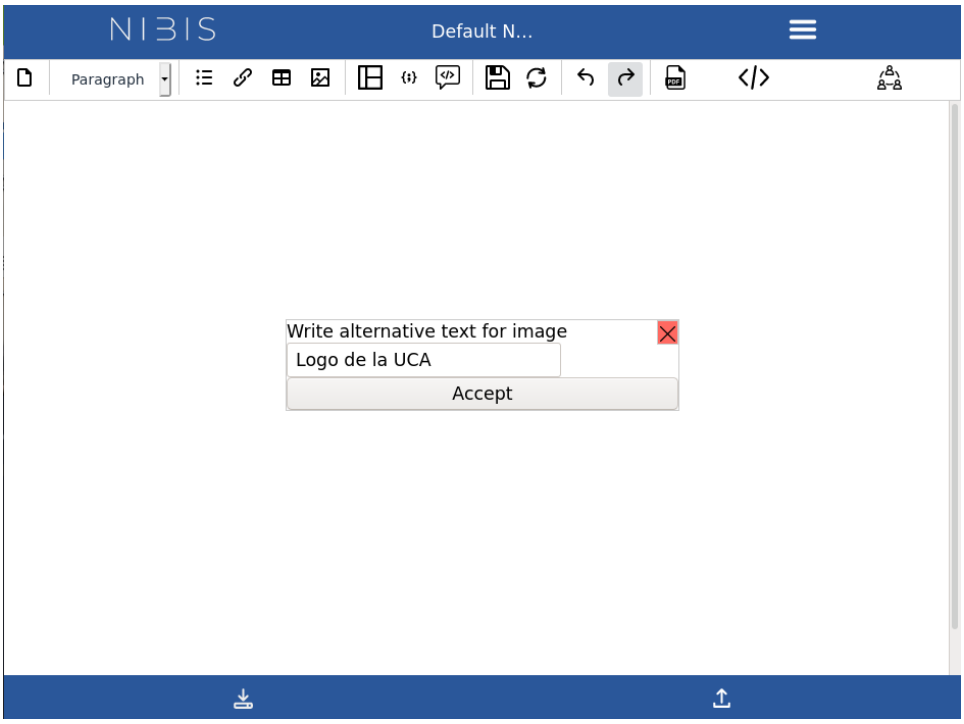



Figura 8.13: Diálogo emergente para texto alternativo de la imagen.

Tras cumplimentar los campos y el usuario acepta se insertará la imagen indicada, con el texto alternativo y un botón para eliminar la imagen.


Figura 8.14: Logo de la UCA mostrándose adecuadamente.

### 8.3.6. Insertar contenedor en el editor

En el panel de funciones del editor podemos encontrar un botón con el icono de tres bloques organizados. Está pensado para insertar un contenedor en la posición donde el usuario esté escribiendo.

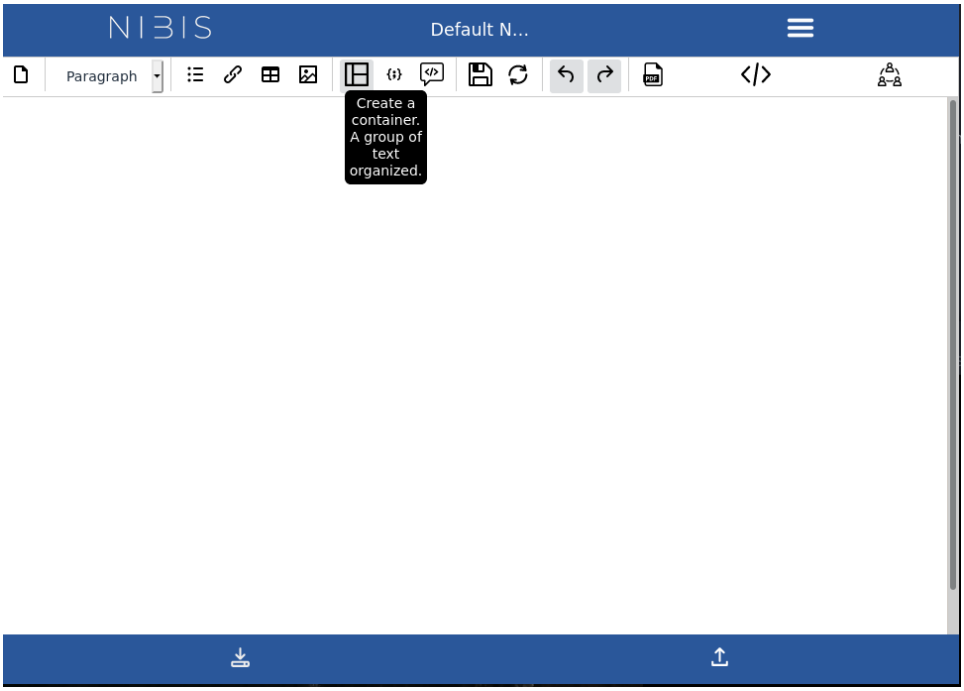



Figura 8.15: Ratón encima del botón de insertar contenedor.

Tras seleccionar el botón saldrá un diálogo emergente que pregunta por la clase de este contenedor. La clase del contenedor es un identificativo de este grupo de texto.

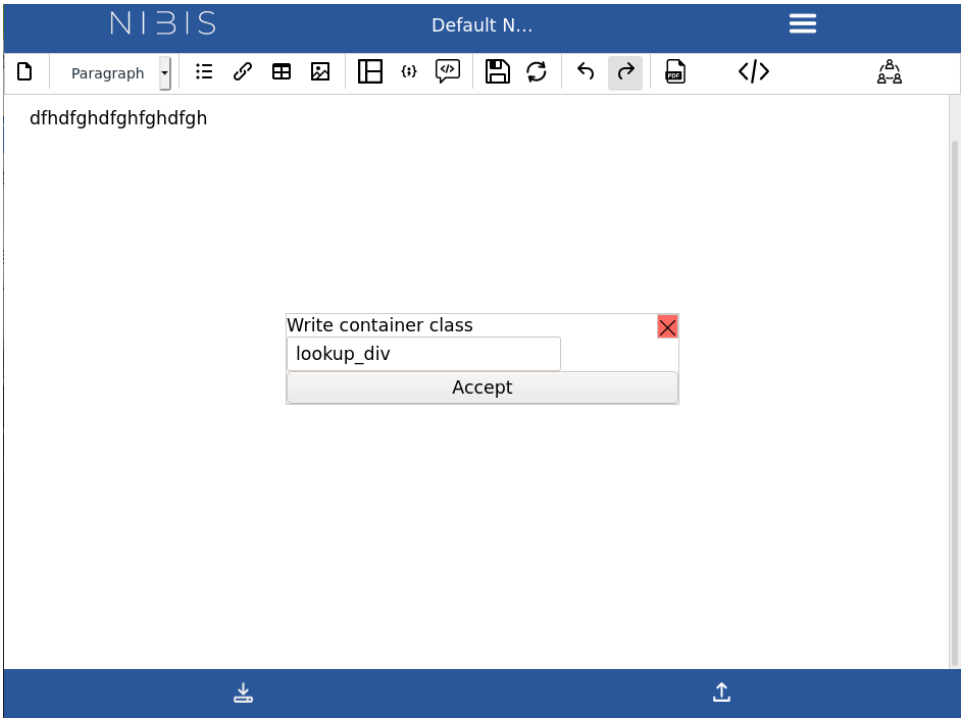



Figura 8.16: Diálogo emergente para escribir la clase del contenedor.

Tras cumplimentar el campo y el usuario acepta se insertará el contendor indicado.

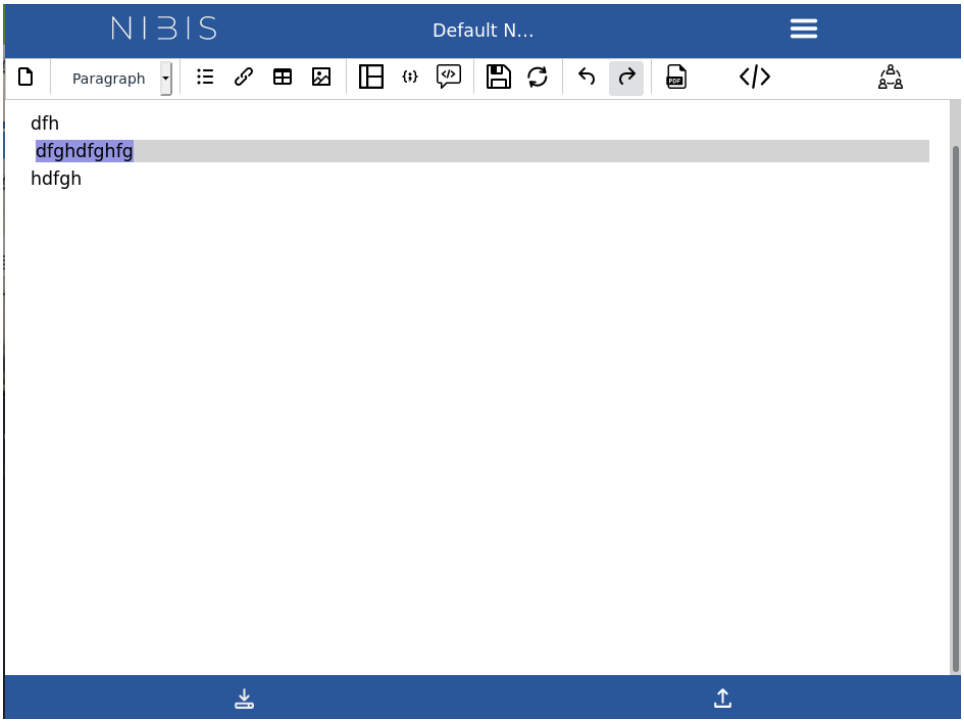



Figura 8.17: Contenedor visible con el color gris que separa los bloques entre sí.

### 8.3.7. Uso de la metaetiqueta en el editor

En el panel de funciones del editor podemos encontrar un botón con el icono de unas llaves que contiene un punto y coma dentro. Este símbolo representa a la metaetiqueta que se puede usar para escribir ciertos caracteres especiales de HTML y LaTeX.

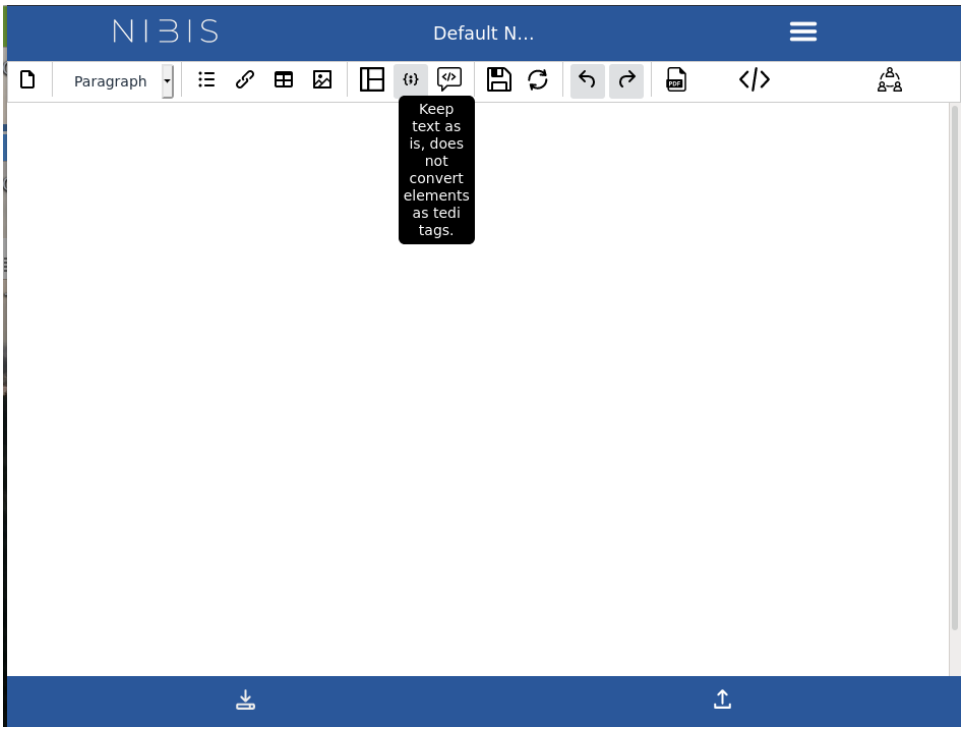



Figura 8.18: Ratón encima del botón de la metaetiqueta

Esta funcionalidad solo sirve para textos seleccionados porque puede convertir una línea completa o solo parte del texto en la misma línea.

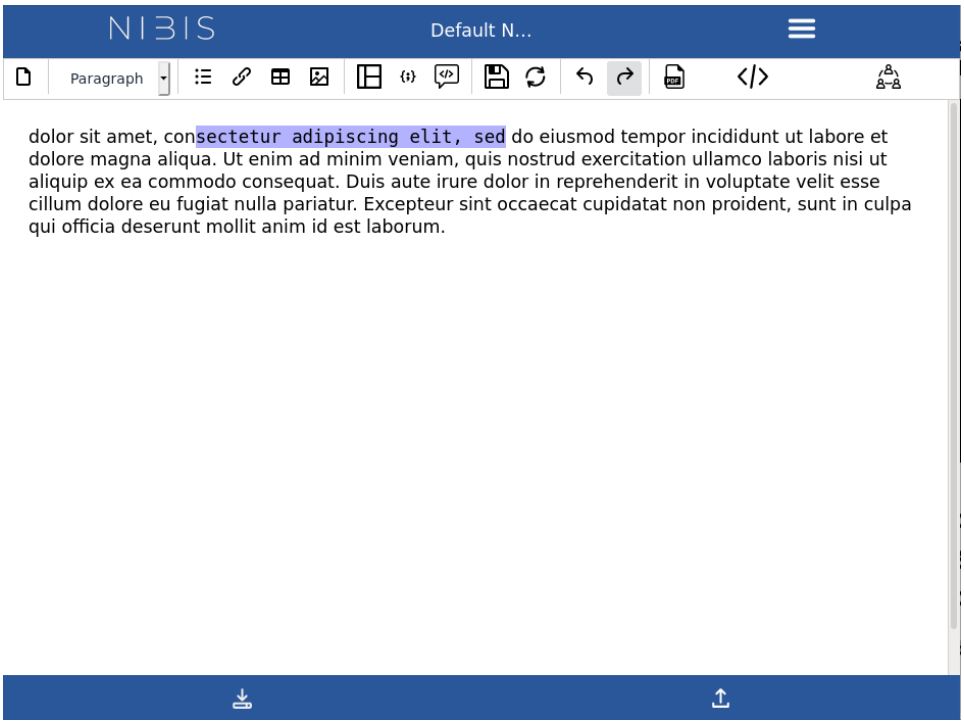



Figura 8.19: Metaetiqueta insertada en el texto como se puede ver en la fuente de letra distinta en la línea.

### 8.3.8. Exportar a PDF

En el panel de funciones del editor podemos encontrar un botón con el icono de un documento PDF. Este botón nos permite exportar directamente a PDF. Mientras se esté convirtiendo, aparecerá sombreado el botón para mostrar que no ha terminado la operación.

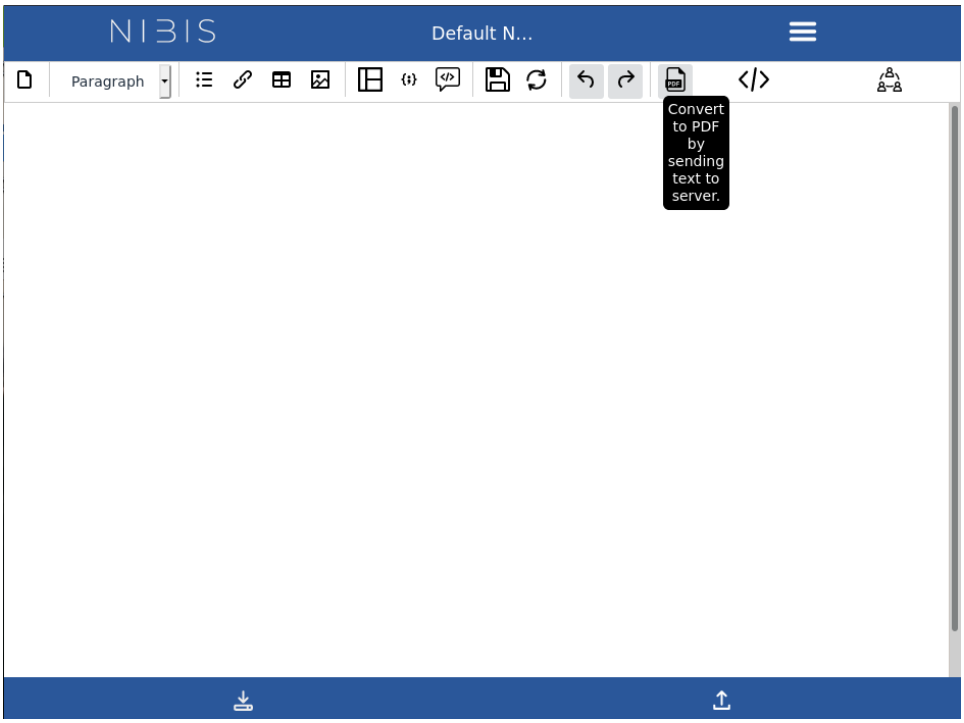



Figura 8.20: Ratón encima del botón de crear PDF.

Si el resultado es satisfactorio, se mostrará el PDF directamente en la interfaz diviéndola en dos tal como se muestra en la imagen.

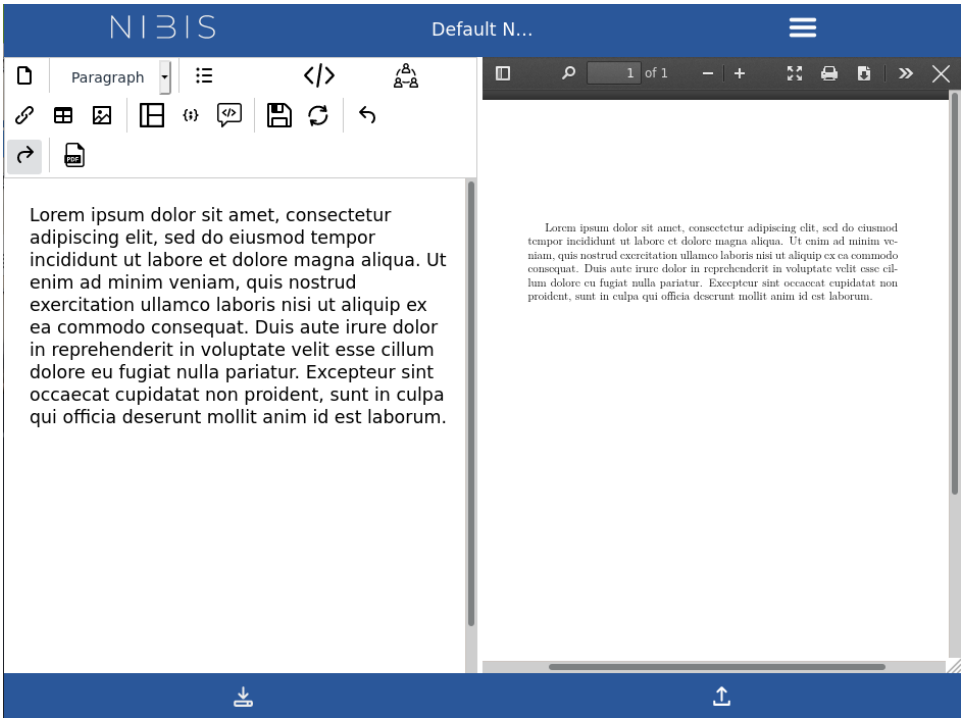



Figura 8.21: PDF con el texto del editor añadido.

### 8.3.9. Conversión a lenguaje intermedio

En el panel de funciones del editor podemos encontrar un botón con una barra inclinada y los caracteres <>. Este botón convierte entre el editor que muestra los elementos de una manera más agradable y la otra tal como se escribe cada elemento en el lenguaje intermedio.

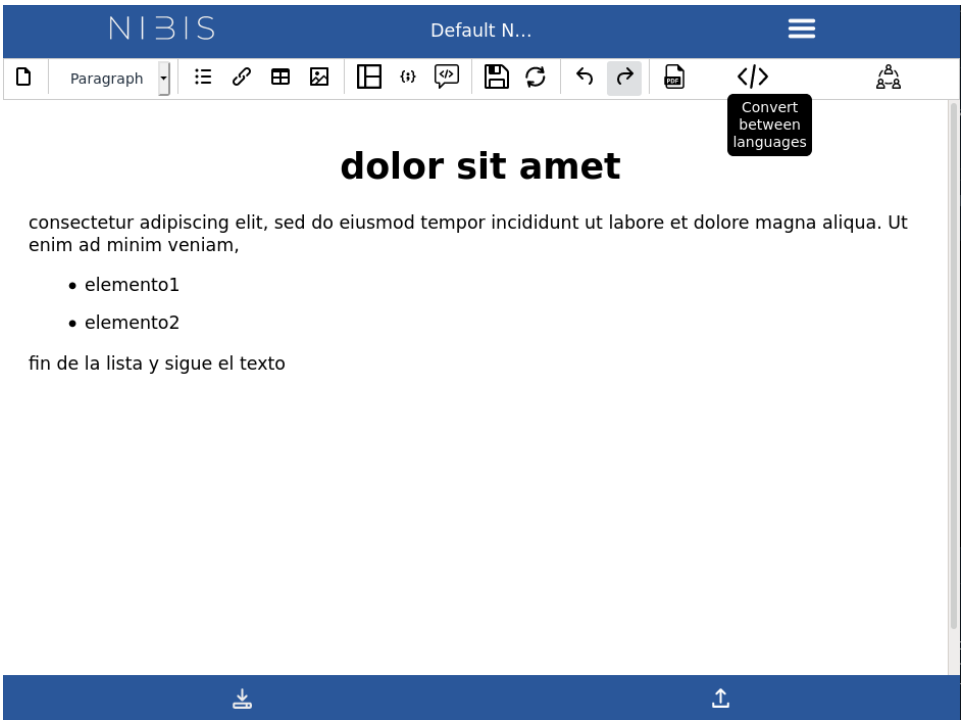



Figura 8.22: Ratón encima del botón de la conversión al lenguaje intermedio.

Tras pulsarlo se convierte el texto al lenguaje intermedio y se cambia el editor mostrando tal como se escriben en vez de como se ven.

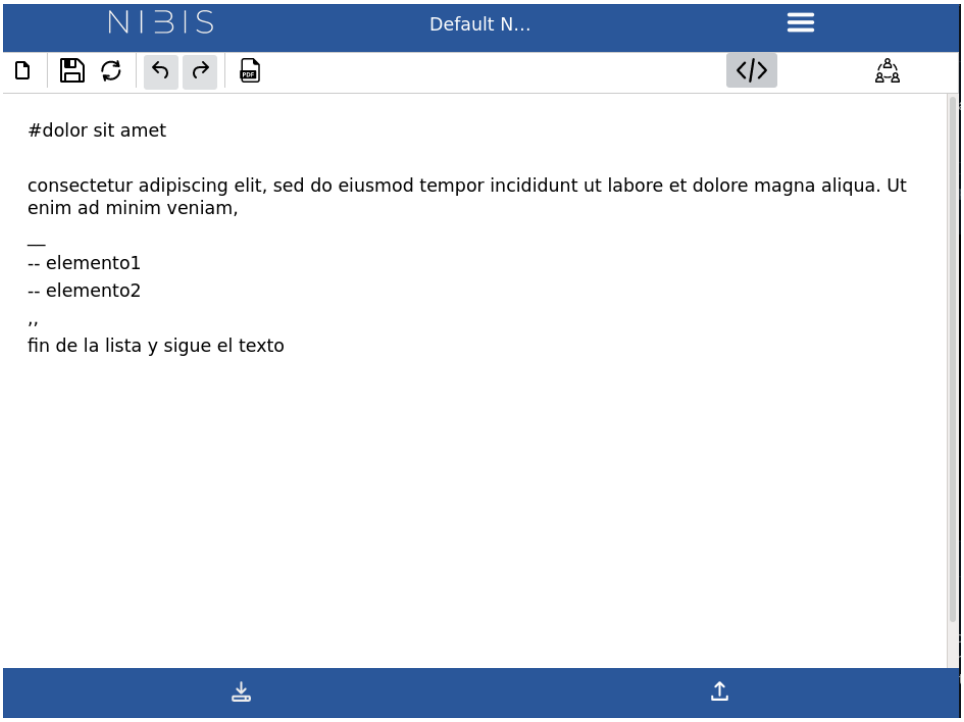



Figura 8.23: Conversión del editor visual a modo código.

En este editor se pueden añadir más etiquetas, eliminarlas y observar las conversiones entre las distintas etiquetas. Si se vuelve a pulsar el botón de conversión, se recupera el editor visual con los nuevos cambios.

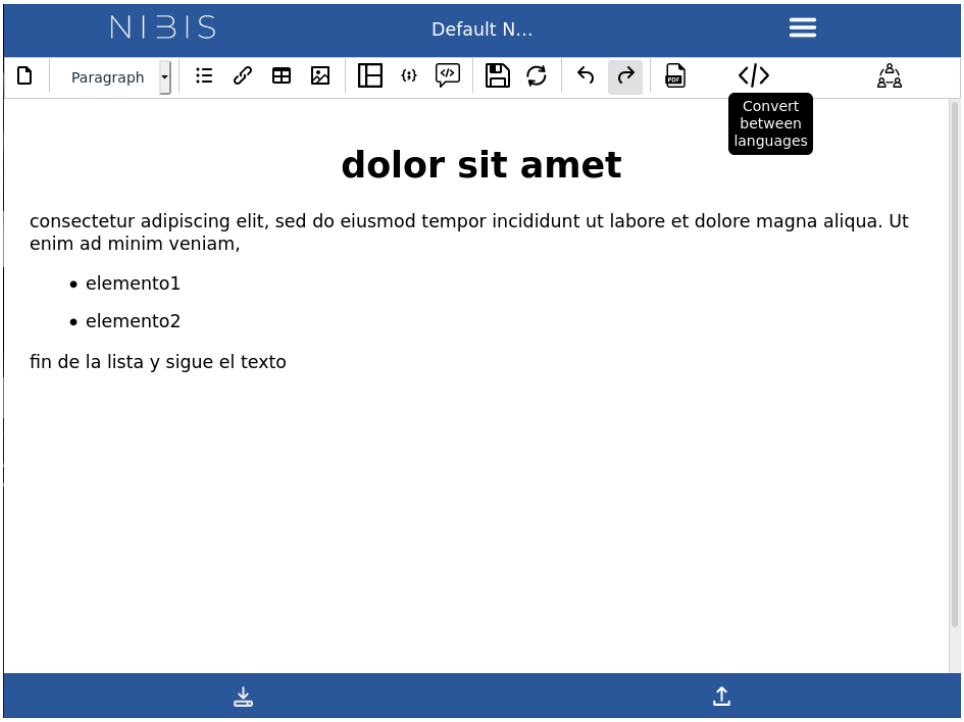



Figura 8.24: Conversión del editor en modo código a modo visual.

### 8.3.10. Compartir el documento

En el panel de funciones del editor podemos encontrar un botón con un icono con la representación simple de tres personas conectadas por tres líneas. Esta función permite compartir el documento en tiempo real y escribir sincronizados.

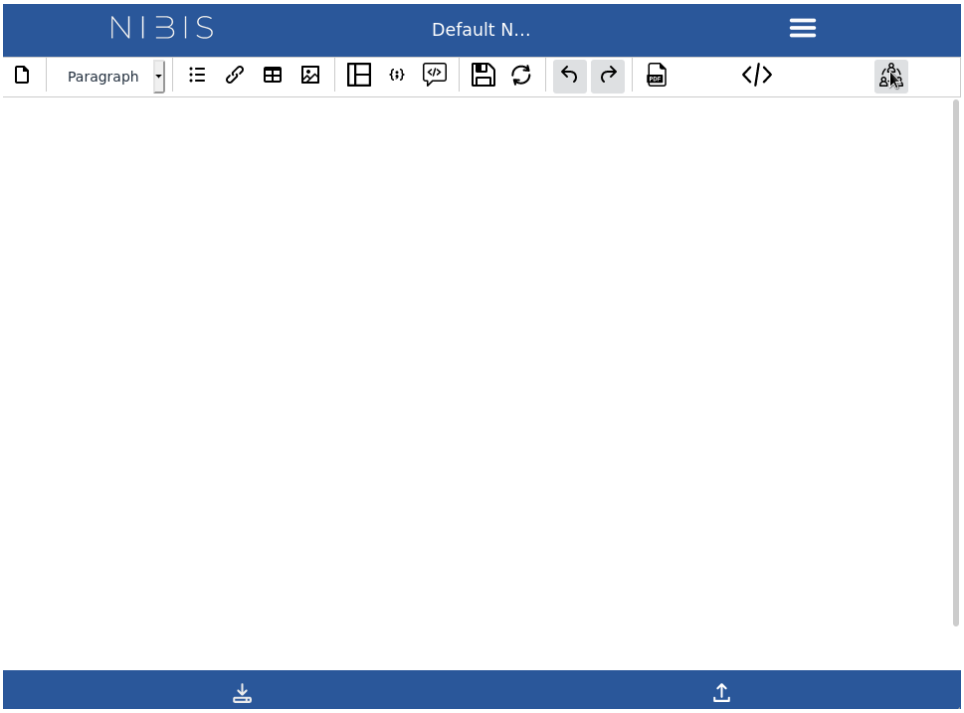



Figura 8.25: Ratón encima del botón de compartir de Nibis.

Tras pulsar el botón, aparecerá una pantalla de bienvenida en el idioma del navegador del usuario en el que explica cómo funciona esta funcionalidad en profundidad para el usuario.

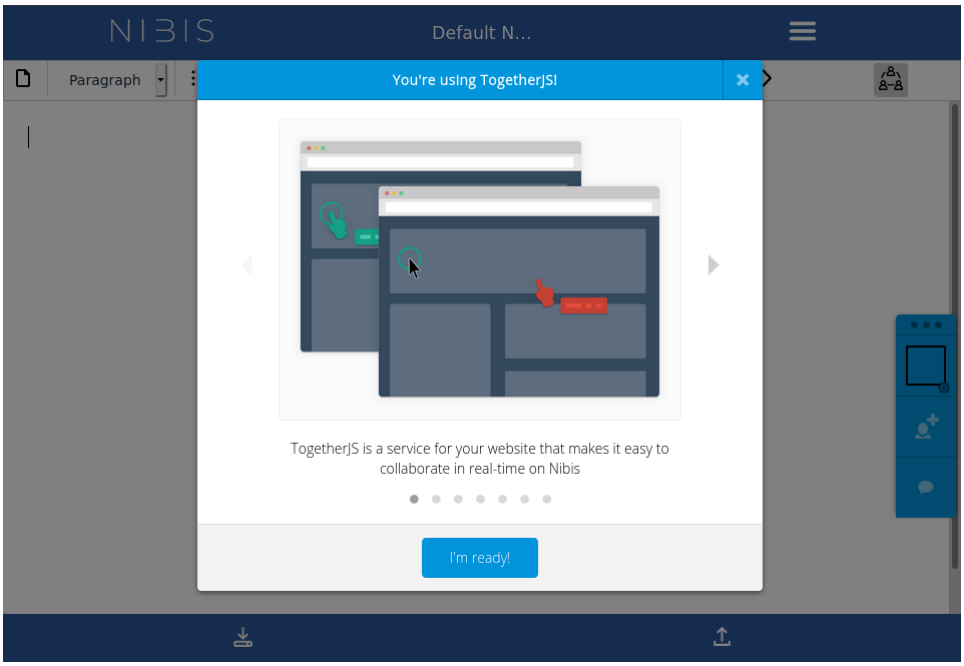



Figura 8.26: Pantalla inicial para compartir.

Si el usuario selecciona el botón de 'Estoy listo', podrá obtener el enlace con el que puede invitar a otros. Con esta invitación otros usuarios pueden leer y editar el texto.

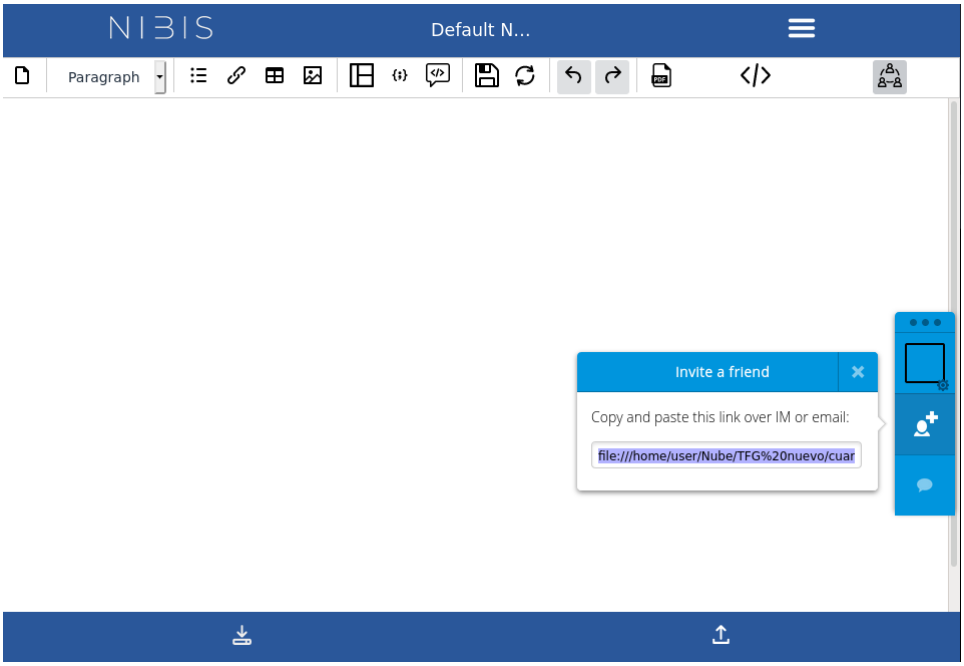



Figura 8.27: Pantalla para mostrar el enlace que el usuario puede compartir.

# Capítulo 9

# Conclusiones

En este capítulo se expondrán los objetivos que se han alcanzado en este proyecto, lo que hemos aprendido durante el desarrollo de este y por último se indicarán posibles cambios o mejoras que se podrían añadir al proyecto para mejorar su rendimiento, así como trabajos futuros que podrían utilizar nuestro proyecto como base.

### 9.0.1. Objetivos alcanzados

Este proyecto tiene como objetivo principal crear una plataforma que simplifique el uso de LaTeX para la gran mayoría de usuarios. Este objetivo principal ha dado como resultado una investigación inicial sobre los posibles editores que se pueden utilizar para este objetivo, investigación durante el desarrollo sobre la interacción de los usuarios con la interfaz y un producto final con un conjunto de funciones necesarias para realizar un documento.

Al usar plantillas y poder incorporar etiquetas LaTeX este proyecto permite crear documentos de tipo profesional. Esto se ha comprobado haciendo la propia documentación del proyecto en el editor. Ha ayudado mucho a su comprobación y la mejora sustancial del proyecto completo.

El producto final requiere automatizar la generación de un PDF para que el usuario lo tenga sin necesidad de conocer cómo funcionan LaTeX ni TexLive. Para ello se planteó primero una conversión con Javascript para hacerlo del lado del cliente. Esto era una tarea más compleja como para hacerse con este enfoque y se prefirió hacer una API REST con la que se automatizó mediante Node. Esto reduce la carga computacional del proyecto, pasándolo al servidor.

Al ser un editor WYSIWYG, el usuario previsualiza lo que ha escrito directamente en el propio navegador antes que estar continuamente exportando a PDF y si lo necesita puede ver el código en el otro modo que tiene la plataforma. Esto ha ayudado a los usuarios a entender cómo funciona la plataforma y lo hace más usable.

Todas las herramientas usadas son software libre y están bien documentadas. Esto último ha sido de gran utilidad ya que a pesar de usar herramientas han requerido modificaciones, personalizaciones y configuraciones para este proyecto en específico.

El proceso de conversión se ha optimizado y precisado. Durante todo el desarrollo se han ido detectando conversiones incorrectas o mejorables. Entonces, este proyecto ha ayudado a tener resultados más sanos y fiables.

### 9.0.2. Lecciones aprendidas

Este Trabajo Final de Grado ha supuesto una parte clave de mi aprendizaje en el grado. Ha puesto a prueba todo lo que he aprendido durante los estudios.

En cuanto al aprendizaje más teórico, ha puesto a prueba todos los campos que se desarrollan durante el grado. Es verdad que en algunas asignaturas he tenido prácticas de Javascript pero no cumpliendo un estándar ni buscando la compatibilidad entre navegadores.

La parte fundamental y más complicada del proyecto ha sido el editor. Este apartado tal como está implementado aprovecha el rendimiento de los navegadores actuales. Para esto ha hecho falta conseguir una base de documentos y referencias para comprender cómo el navegador trabaja con algunas herramientas como la selección, el DOM[1] y expresiones regulares. Con este proyecto he aprendido y revisado ciertas tecnologías, como JQuery, que en los últimos años de la carrera había dejado de lado. No solo me ha servido para demostrar mis capacidades con este proyecto, también durante las prácticas extracurriculares que estoy haciendo.

La documentación es otro punto clave donde he repasado ciertas campos que durante el grado he podido no prestar tanto interés o que no valoraba por desconocimiento. Este trabajo demuestra que la planificación y organización es necesaria para realizar cualquier producto de mayor envergadura.

Por último, me ha hecho replantear ciertos formas de trabajo que he ido

[1] Modelo en Objetos para la Representación de Documentos

adoptando durante estos años. En los últimos meses me ha dejado más que claro que por muchas horas se dedique a una tarea, es igual de importante el descanso y el tiempo a estar con familiares y amigos, a pesar de la situación actual.

### 9.0.3. Trabajo futuro

He observado continuamente que se podían añadir nuevas funcionales para mejorar los distintos apartados. A pesar de mi curiosidad y ganas de comprender cómo mejorarlo algunas funcionalidades no se han podido implementar por falta de tiempo y conocimiento.

- Nuevas funciones que tiene otros editores y suites ofimáticas. Algunos usuarios indicaron durante las pruebas funciones como interlineado, sangría,
- Nuevas plantillas que aprovechen el proyecto en sus múltiples variantes. A pesar de centrarse en documentos profesionales, se pueden realizar presentaciones, texto en varias columnas, cartas, documentos legales y documentos que cumplen con estándares del IEEE.
- Incorporar conversiones a nuevos lenguajes como Markdown y reStructuredText. Esto favorecerá que usuarios que ya sus documentos estén hechos en estos lenguajes puedan aprovechar el potencial de la plataforma.
- Mejorar la opción para compartir documentos. Tal como está ahora mismo envia demasiada información, algo ineficiente y que lo ralentiza para dispositivos no tan modernos. Si se enviara solo la posición y la nueva línea se ahorraría todo esto.
- Generar una variedad de tutoriales y guías para que el usuario aproveche al máximo todas las funcionalidades de la plataforma.
- Aplicación de escritorio y para móvil con Electron aprovecharía que la plataforma está casi completamente pensada para ser sin mucha complicación una aplicación.

Cabe destacar que realizar una conversión completa entre lenguajes que no comprueban su validez antes de ejecutarse. Por eso tiene sentido realizar validaciones de todo código HTML realizado. XHTML es un ejemplo de hacer necesaria la validez de un lenguaje para poder ejecutarse. Debido a este problema ciertas conversiones pueden dar resultados incorrectos. Esto

es un problema inherente al trabajar con HTML y se ha intentado evitar en todo el proyecto y mediante limitaciones en el uso de etiquetas, eliminar todas las posibles entradas de una etiqueta script de HTML ' `<script>` '.

# Parte IV

# Bibliografía

# Bibliografía


[1] https://www.census.gov/content/dam/Census/library/publications/2018/acs/ACS-39.pdf *U.S Gov.* (2005).

[2] LaTeX is NOT Easy: Creating Accessible Scientific Documents with R Markdown (2019) http://scholarworks.csun.edu/handle/10211.3/210398

[3] Chrome Developers, Chrome docs (2021) https://developer.chrome.com/

[4] Mozilla and individual contributors, Mozilla Developer Network (2021) https://developer.mozilla.org

[5] Ibrahim, Muhammad and Aftab, Shabib and Ahmad, Munir and Iqbal, Ahmed and Khan, Bilal and Iqbal, Muhammad and Sabri, Nouh and Ihnaini, Baha "Presenting and Evaluating Scaled Extreme Programming Process Model" Computer Science Review 39 (2021) 100314 https://cutt.ly/pkUYMQi

[6] Marcelo Morandini,Thiago Adriano Coleti,Edson Oliveira Jr,Pedro Luiz Pizzigatti Corrêa "Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams" Computer Science Review 39 (2021) 100314 https://www.redalyc.org/pdf/849/84933912009.pdf

[7] Richard Stallman, The GNU Operating System and the Free Software Movement (2021) https://www.gnu.org/

[8] Tooltip definition, Microsoft Academic, 2021 https://academic.microsoft.com/v2/detail/73556201

[9] Shortcut definition, Microsoft Academic, 2021 https://academic.microsoft.com/v2/detail/2781174843

[10] Editing Task Force about standardized web editors, W3C, 2021 https://w3c.github.io/editing/

[11] Ian Sommerville. Software Engineering. Addison-Wesley, Harlow, England, 9 edition, 2010.

[12] Dan M. Brown “Communicating Design Developing Web Site Documentation for Design and Planning” Second edition p. 167 , 2010

[13] Universitat de Lleida, “Curso de Interacción Persona-Ordenador Basado en el Modelo de Proceso de la Ingeniería de la usabilidad y de la accessibilidad (MPIu+a)”, 2021 https://mpiua.invid.udl.cat

[14] Pierre Bourque and Richard E. Fairley, editors. SWEBOK: Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, Los Alamitos, CA, version 3.0 edition, 2014.

[15] Amaia Calvo-Fernández Rodríguez Sergio Ortega Santamaría Alicia Valls Saez “Métodos de evaluación con usuario” p.17 (2011)

[16] Amaia Calvo-Fernández Rodríguez Sergio Ortega Santamaría Alicia Valls Saez “Métodos de evaluación con usuario” p.46 (2011)

[17] OpenJS Foundation, NodeJS https://nodejs.org (2021)

[18] ExpressJS, cors https://github.com/expressjs/cors (2021)

[19] ExpressJS, ExpressJS https://github.com/expressjs/express (2021)

[20] ExpressJS, Morgan https://github.com/expressjs/morgan (2021)

[21] Peter Krumins, Tree-kill https://github.com/pkrumins/node-tree-kill/ (2021)

[22] Mozilla Foundation, PDF.js https://github.com/mozilla/pdf.js/ (2021)

[23] JSFiddle and contributors, TogetherJS https://github.com/mozilla/pdf.js/ (2021)

[24] OpenJS Foundation and jQuery contributors, JQuery https://jquery.com/ (2021)

[25] Eli Grey, Filesaver https://github.com/eligrey/FileSaver.js (2021)

[26] OpenJS Foundation, NPM https://github.com/npm/cli (2021)

[27] Tex Users Group, XeLaTeX/XeTex https://ctan.org/pkg/xetex (2021)

[28] Stuart Knightley, David Duponchel, Franz Buchinger, António Afonso, JSZip https://github.com/Stuk/jszip (2021)

[29] Einar Lielmanis, Liam Newman, and contributors. JS-Beautify https://github.com/beautify-web/js-beautify (2021)

[30] OpenJS Foundation, QUnit https://qunitjs.com/ (2021)

[31] Event driven programming definition, Microsoft Academic, 2021 https://academic.microsoft.com/v2/detail/77362995

[32] UML 2 Use Case Diagramming Guidelines, Scott W. Ambler, 2021 http://www.agilemodeling.com/style/useCaseDiagram.htm

# Parte V

# Licencia

# GNU Free Documentation License



### Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others. This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software. We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions

stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law. A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language. A "**Secondary Section**" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them. The "**Invariant Sections**" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none. The "**Cover Texts**" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words. A "**Transparent**" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**". Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only. The "**Title Page**" means, for a printed book, the

title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text. The "**publisher**" means any person or entity that distributes copies of the Document to the public. A section "**Entitled XYZ**" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "**Acknowledgements**", "**Dedications**", "**Endorsements**", or "**History**".) To "**Preserve the Title**" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition. The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3. You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying

with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects. If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages. If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public. It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the

Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles. You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard. You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one. The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers. The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work. In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents

released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects. You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document. If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail. If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License. However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation. Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice. Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `https://www.gnu.org/licenses/`. Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

## 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provi-

des prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site. "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization. "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document. An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008. The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

## ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation. If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.