

Grado en Ingeniería Informática  
Ingeniería del Software

Trabajo de Fin de Grado

---

**Desarrollo de una aplicación para el análisis de sentencias judiciales utilizando la línea de productos software WacLine**

---

Autor

*Aitor Diaz de Otazu Hernando*

2020



Grado en Ingeniería Informática  
Ingeniería del Software

Trabajo de Fin de Grado

---

**Desarrollo de una aplicación para el análisis de  
sentencias judiciales utilizando la línea de  
productos software WacLine**

---

Autor

*Aitor Diaz de Otazu Hernando*

Directora

Arantza Irastoza



---

## **Resumen**

---

Este Trabajo de Fin de Grado ha sido desarrollado utilizando una tecnología muy poco extendida, las Líneas de Producto de Software (SPL), en concreto la SPL WacLine desarrollada por Haritz Medina, dentro del grupo de investigación ONEKIN. Concretamente, en este Trabajo de Fin de Grado se ha desarrollado una aplicación para el análisis de sentencias judiciales utilizando dicha SPL como medio para generarla. Para ello se han diseñado e implementado nuevas características que aportan las funcionalidades necesarias para la aplicación deseada. Por último, dichas características se han incluido en la arquitectura de la SPL para poder construir a partir de WacLine la aplicación DOCAL que es el objetivo de este Trabajo de Fin de Grado.



---

## Agradecimientos

---

En primer lugar, me gustaría agradecer a mi tutora, Arantza Irastorza, la oportunidad de realizar este Trabajo de Fin de Grado y poder trabajar en un proyecto basado en las Líneas de Producto de Software al nivel de un proyecto real. Agradecerle también el apoyo y la atención recibida durante el desarrollo de todo el proyecto aun pasando por una época tan inesperada como una pandemia global con todo lo que eso implica.

En segundo lugar, agradecer a Haritz Medina, desarrollador de la SPL WacLine y miembro del grupo de investigación ONEKIN, por aportarme toda la ayuda referente a WacLine que he necesitado durante el aprendizaje de la arquitectura de la SPL y el desarrollo del proyecto.

Por último, agradecer a mi familia y amigos de toda la vida por estar siempre ahí y apoyarme y ayudarme en cualquier cosa siempre que lo he necesitado. También agradecer a todos los compañeros de universidad que ahora son parte indispensable de mi vida, en concreto agradecerle a Asier Barrasa el acompañarme durante prácticamente todos estos años y a Gorka Arce por acompañarme los dos últimos cursos además de en la realización de este Trabajo de Fin de Grado. Tengo claro que sin mi familia y amigos serian impensables todos los proyectos que llevo a cabo en mi vida, simplemente gracias.





---

# Índice general

---

<b>Índice general</b>	<b>IX</b>
<b>Índice de figuras</b>	<b>XIV</b>
<b>Índice de tablas</b>	<b>XVII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos y planificación del trabajo</b>	<b>3</b>
2.1. Antecedentes . . . . .	3
2.2. Objetivos . . . . .	4
2.3. Alcance . . . . .	5
2.4. Estructura de descomposición de trabajo . . . . .	6
2.4.1. Fase de Gestión de Conocimiento (GC) . . . . .	7
2.4.2. Fase de Gestión (G) . . . . .	8
2.4.3. Desarrollo del proyecto (DP) . . . . .	9
2.4.4. Documentación (D) . . . . .	10
2.5. Planificación temporal . . . . .	11
2.6. Herramientas . . . . .	14
2.6.1. GitHub y Git . . . . .	14

2.6.2.	Time Machine	14
2.6.3.	Overleaf y LaTeX	14
2.6.4.	TeamGantt	15
2.6.5.	Draw.io	15
2.6.6.	Visual Studio Code	15
2.6.7.	Google Chrome	16
2.6.8.	pure::variants	16
2.6.9.	Eclipse	16
2.6.10.	Node.js y ECMAScript	16
2.7.	Gestión de Riesgos	17
2.7.1.	Dificultad compaginando el proyecto con las prácticas	17
2.7.2.	Dificultad en el aprendizaje de las herramientas y tecnologías necesarias	17
2.7.3.	Dificultad en el entendimiento de WacLine SPL	18
2.7.4.	Planificación incorrecta	18
2.7.5.	Problemas técnicos del software utilizado	19
2.7.6.	Problemas técnicos en el hardware utilizado	19
2.7.7.	Pandemia global	20
2.7.8.	Problemas de índole personal	20
2.8.	Plan de Calidad	21
2.8.1.	Producto	21
2.8.2.	Memoria	21
<b>3.</b>	<b>SPL y Pure::Variants</b>	<b>23</b>
3.1.	SPL	23
3.1.1.	Diagrama de características	26
3.2.	Pure::Variants	28

---

<b>4. Presentación de las características y organización de WacLine</b>	<b>31</b>
4.1. Arquitectura de las Anotaciones Web	31
4.2. Características de WacLine	33
4.3. Estructura del código de WacLine	36
<b>5. Módulo gestor de documentos</b>	<b>41</b>
5.1. Análisis de las interfaces de las bases de datos	41
5.2. Requisitos del módulo gestor de documentos	46
<b>6. Análisis de Requisitos</b>	<b>49</b>
6.1. Requisitos funcionales	49
6.2. Requisitos no funcionales	51
6.3. Casos de uso	51
6.3.1. Crear Sesión	52
6.3.2. Renombrar Sesión	53
6.3.3. Eliminar sesión	54
6.3.4. Transferir sesión	55
6.3.5. Crear anotación	56
6.3.6. Eliminar anotación	56
6.3.7. Comentar anotación	57
6.3.8. Enlazar anotación	57
6.3.9. Mostrar enlaces de la anotación	58
6.3.10. Añadir tema	58
6.3.11. Modificar tema	59
6.3.12. Eliminar tema	60
6.3.13. Añadir código	61
6.3.14. Modificar código	62

6.3.15. Eliminar código . . . . .	63
6.3.16. Exportar anotaciones . . . . .	64
6.3.17. Generar documento Word . . . . .	65
6.3.18. Generar Codebook . . . . .	65
6.3.19. Seleccionar Servidor de anotaciones . . . . .	66
<b>7. Diseño e implementación de la nueva arquitectura de WacLine</b>	<b>67</b>
7.1. Nuevo diagrama de características . . . . .	67
7.2. Característica <i>Session</i> . . . . .	69
7.3. Característica <i>Linking</i> . . . . .	74
7.4. Característica <i>DocExporter</i> . . . . .	77
<b>8. Descripción de la aplicación DOCAL</b>	<b>81</b>
8.1. Selección de características . . . . .	81
8.2. Aspectos de usabilidad de la aplicación DOCAL . . . . .	83
<b>9. Seguimiento y Control</b>	<b>85</b>
9.1. Problemas encontrados durante el desarrollo del TFG . . . . .	85
9.2. Desviaciones . . . . .	87
<b>10. Conclusiones</b>	<b>89</b>
<b>Anexos</b>	
<b>A. Plantilla módulo export</b>	<b>93</b>
<b>B. Diseño de las interfaces</b>	<b>99</b>
B.1. Sidebar . . . . .	100
B.2. Alertas . . . . .	103

---

B.2.1. Alerta de aviso . . . . .	104
B.2.2. Alerta de confirmación . . . . .	104
B.2.3. Alerta de propósito general . . . . .	106
<b>C. Implementación de las características</b>	<b>107</b>
<b>Bibliografía</b>	<b>115</b>

---

## Índice de figuras

---

2.1. Diagrama E.D.T . . . . .	7
2.2. Diagrama de Gantt por meses - Primera parte . . . . .	13
2.3. Diagrama de Gantt por meses - Segunda parte . . . . .	13
3.1. Diferencia de relación calidad/precio entre las SPL y la producción individual . . . . .	24
3.2. Proceso de desarrollo de una SPL [Eichberg, 2017] . . . . .	25
3.3. Ejemplo de un diagrama de características . . . . .	27
3.4. Definición de características de WacLine . . . . .	28
3.5. Selección de las características de una variante . . . . .	29
3.6. <i>Family Model</i> de DOCAL . . . . .	30
3.7. Ejemplo de condiciones de Pure::Variants . . . . .	30
4.1. Mapa de concepto de WacLine . . . . .	32
4.2. Diagrama de Funcionalidades - WacLine . . . . .	33
4.3. Servidores de Anotaciones - WacLine . . . . .	34
4.4. Target - WacLine . . . . .	34
4.5. Purposes - WacLine . . . . .	35
4.6. Operaciones - WacLine . . . . .	36
4.7. Estructura de ficheros - WacLine . . . . .	37

---

4.8. Estructura de la carpeta Scripts - WacLine . . . . .	38
4.9. Función encargada de procesar el evento <i>UpdateSession</i> . . . . .	39
4.10. Ejemplo de punto de variación en el código . . . . .	39
5.1. Buscador Aranzadi . . . . .	42
5.2. Resultado de la búsqueda - Aranzadi . . . . .	42
5.3. Visualización de documentos - Aranzadi . . . . .	43
5.4. Buscador Tirant Online . . . . .	44
5.5. Resultado de la búsqueda - Tirant . . . . .	44
5.6. Buscador Cendoj . . . . .	45
5.7. Resultado de la búsqueda - Cendoj . . . . .	45
5.8. Visualizador de documentos - Cendoj . . . . .	46
6.1. Modelo de Casos de Uso . . . . .	52
6.2. Opción <i>Create new Session</i> . . . . .	53
6.3. Formulario de creación de sesiones . . . . .	53
6.4. Menú contextual de las sesiones . . . . .	53
6.5. Formulario para renombrar la sesión actual . . . . .	54
6.6. Formulario de confirmación para eliminar la sesión . . . . .	54
6.7. Formulario de confirmación para transferir la sesión . . . . .	55
6.8. Formulario para seleccionar a qué sesión transferir la información . . . . .	55
6.9. Proceso de creación de una anotación . . . . .	56
6.10. Menú contextual de las anotaciones . . . . .	56
6.11. Formulario de comentarios . . . . .	57
6.12. Menú de enlaces de una anotación . . . . .	58
6.13. Visualización de anotaciones enlazadas . . . . .	58
6.14. Formulario de creación de temas . . . . .	59

6.15. Menú contextual del tema . . . . .	60
6.16. Formulario de modificación de temas . . . . .	60
6.17. Confirmación para eliminar un tema . . . . .	61
6.18. Formulario de creación de códigos . . . . .	62
6.19. Menú contextual de los códigos . . . . .	63
6.20. Formulario modificación de códigos . . . . .	63
6.21. Confirmación para eliminar un tema . . . . .	64
6.22. Opción de exportar a documento Word . . . . .	64
6.23. Confirmación para generar el <i>codebook</i> por defecto . . . . .	65
6.24. Selección del servidor de anotaciones . . . . .	66
7.1. Nuevo diagrama de características de WacLine . . . . .	68
7.2. Sistema de ficheros de la característica <i>Session</i> . . . . .	69
8.1. Selección de características . . . . .	82
8.2. Selección del servidor de anotaciones . . . . .	83
B.1. Captura de pantalla de la interfaz de usuario . . . . .	100
B.2. Sección ToolBox del Sidebar . . . . .	101
B.3. Sección Codebook del Sidebar . . . . .	102
B.4. Sección de sesiones del Sidebar . . . . .	103
B.5. Ejemplo de alerta de aviso . . . . .	104
B.6. Ejemplo de alerta de confirmación . . . . .	105
B.7. Ejemplo de alerta de error . . . . .	105
B.8. Ejemplo de alerta de propósito general . . . . .	106



---

## Índice de tablas

---

2.1. Dedicaciones estimadas en horas para cada tarea . . . . .	12
7.1. Tabla de relaciones entre las nuevas características y casos de uso . . . . .	68
7.2. Tabla de funcionalidades implementadas en la característica <i>Session</i> . . . . .	69
7.3. Tabla de funcionalidades implementadas en la característica <i>Linking</i> . . . . .	74
7.4. Tabla de funcionalidades implementadas en la característica <i>DocExporter</i> . . . . .	77
9.1. Desviaciones en las horas definidas para los paquetes de trabajo . . . . .	88



# 1. CAPÍTULO

---

## Introducción

---

La evolución de la tecnología y la aparición de la web ha sido una revolución en todos los ámbitos, incluido en la anotación de documentos con la aparición de las anotación web.

Este hecho ha generado una gran cantidad de opciones que aportan herramientas para la anotación de documentos online, eliminando la necesidad de imprimir y anotar dichos documentos como se ha llevado haciendo toda la vida.

En el dominio judicial, aunque se ha contabilizado un gran aumento en la cantidad de herramientas y bases de datos que han aparecido, los profesionales del sector carecen de una herramienta que les permita realizar los análisis de documentos fácilmente eliminando la necesidad de imprimir dichos documentos y anotarlos a mano.

Además las Líneas de Producto de Software (SPL) cada vez están más extendidas y su utilización en gran variedad de dominios es cada vez más evidente. Estas SPL nos permiten generar aplicaciones específicas para un dominio en donde tenemos unas funcionalidades comunes y unas funcionalidades variables dependiendo de las necesidades reduciendo los costes y tiempos de desarrollo y mejorando la calidad de las aplicaciones.

El Trabajo de Fin de grado (TFG a partir de este momento) esta centrado en la extensión de una SPL para producir la aplicación que cubre las necesidades de los juristas en el análisis de documentos. Durante este TFG se hace un análisis de dichas necesidades y se explica la estructura de una SPL y en concreto la SPL WacLine que ha sido la utilizada para llevar acabo dicha aplicación. Para ello se han desarrollado las características necesarias y se han incluido en la arquitectura de WacLine, seleccionando después el conjunto de características adecuado para que el producto resultante cumpla los objetivos del TFG.

El contenido de esta memoria está dividido en 10 capítulos, organizados de esta manera:

- Capítulo 2: Se presentan los antecedentes, los objetivos del TFG, el alcance del proyecto y los paquetes de trabajo en los que se ha dividido este TFG. También se presenta la planificación del proyecto, las herramientas utilizadas y los riesgos identificados.
- Capítulo 3: Se presentan las SPL y sus características. Además de explicar el funcionamiento de la herramienta *Pure::Variants*.
- Capítulo 4: Se explican las características y la arquitectura de la SPL WacLine utilizada en este TFG, desarrollada por Haritz Medina dentro del grupo de investigación ONEKIN<sup>1</sup>.
- Capítulo 5: Se analiza la viabilidad del *módulo gestor de documento*.
- Capítulo 6: Se indican los requisitos del proyecto y se exponen los casos de uso del mismo.
- Capítulo 7: Se expone el diseño e implementación de las nuevas características y su inclusión en la arquitectura de WacLine.
- Capítulo 8: Se detalla la selección de características y los aspectos de usabilidad más importantes de la aplicación resultante de este TFG.
- Capítulo 9: Se muestran los problemas surgidos durante el desarrollo además del Seguimiento y Control realizado durante el proyecto.
- Capítulo 10: Se presentan las conclusiones tanto a nivel técnico como a nivel personal del proyecto y se exponen mejoras para la extensión de la SPL WacLine.

---

<sup>1</sup>Grupo ONEKIN:<https://onekin.org>

## 2. CAPÍTULO

---

### Objetivos y planificación del trabajo

---

En este capítulo se presenta el planteamiento del proyecto inicial elaborado para el correcto desarrollo del mismo. Se describen y analizan los objetivos, definiendo el alcance del proyecto y la planificación para llevarlo a cabo.

También se indican las herramientas necesarias para el desarrollo del proyecto y se identifican los posibles riesgos que pueden ocurrir durante la elaboración del proyecto, detallando además, un plan de mitigación para paliar las posibles consecuencias.

#### 2.1. Antecedentes

La investigación y análisis de documentos jurídicos es el proceso que usan juristas y docentes para determinar qué leyes se aplican a unos hechos determinados que se desean estudiar. Para ello necesitan analizar los hechos presentados en los casos, determinar el problema legal y procesar la información legal disponible relacionada con el problema que han identificado.

Este último paso, a pesar de los avances tecnológicos disponibles, se siguen realizando de forma manual. Aunque es verdad que la búsqueda de documentos se realiza de forma digital, el análisis se realiza de una forma rudimentaria complicando la recopilación de datos y la organización de las anotaciones realizadas al trabajar con documentos físicos. De esta forma, las posibilidades de que se pases por alto anotaciones importantes, o la posibilidad de que se pierda algún documento se pierda son notablemente mayores al

trabajar de forma física. Además realizar una recopilación de todas las anotaciones se convierte en un proceso muy tedioso.

El grupo de investigación ONEKIN de la Universidad del País Vasco (UPV/EHU), específicamente Haritz Medina, está desarrollando una Línea de Producto de Software (SPL) llamada WacLine<sup>1</sup> en la que han unificado tres herramientas relacionadas con el ámbito de las anotaciones web<sup>2</sup>. WacLine nos servirá como punto de partida para desarrollar nuestra aplicación para la cual tendremos que desarrollar nuevos módulos y adaptar los actuales para realizar una perfecta integración.

En este Trabajo de Fin de Grado (TFG en adelante) se intentará crear una solución al problema real al que se enfrentan los docentes y juristas a la hora de analizar sentencias judiciales.

Para ello, se desarrollará una aplicación para el navegador donde los usuarios podrán analizar y recopilar toda la información deseada sin necesidad de lidiar con el análisis de documentos físicos. La extensión Chrome ofrecerá una interfaz sencilla donde el usuario podrá observar, crear y/o modificar los criterios de anotaciones como de una rúbrica se tratase. También dispondrán de un sistema de guardado donde los usuarios podrán elegir si guardar sus anotaciones en local o en su lugar, en la plataforma de anotaciones [hypothes.is](https://hypothes.is)<sup>3</sup>.

Tanto los módulos desarrollados en este TFG como el producto final, tienen el objetivo de formar parte de dicha SPL. Por lo que se presentará el estado actual de la SPL WacLine y la relación con los módulos a desarrollar.

## 2.2. Objetivos

El objetivo principal de este proyecto es la creación de una herramienta basada en anotaciones web que permita a los docentes y juristas analizar resoluciones judiciales o cualquier otro tipo de jurisprudencia. Se implementarán 3 módulos que complementen la línea de productos WacLine.

Para ello se implementará un *módulo gestor de documentos*. Realizará búsquedas en diferentes bases de datos dependiendo de la configuración seleccionada por el usuario, dispondrá de un formulario con criterios de búsqueda. También dispondrá de la opción de

<sup>1</sup>WacLine: <https://www.onekin.github.com/WacLine>

<sup>2</sup>Anotaciones Web: <https://www.w3.org/TR/annotation-model/>

<sup>3</sup>Hypothesis: [hypothes.is](https://hypothes.is)

poder elegir en qué base de datos, de entre las configuradas previamente, desea el usuario realizar la búsqueda. Por último, este módulo, daría la opción de guardar los documentos obtenidos como resultado de la búsqueda.

También se implementará un *módulo de enlazamiento de anotaciones*, que le permitirá al usuario enlazar anotaciones con otras anotaciones realizadas previamente.

Se desarrollará un *módulo generador de informes GoogleDocs* a partir de plantillas definidas específicamente para este proyecto, donde el usuario podrá visualizar todas las anotaciones realizadas de una forma sencilla y clara con referencias a los documentos anotados como a las propias anotaciones.

Como objetivo dentro del proyecto se encuentra el aprendizaje del desarrollo de una extensión para el navegador Chrome. Durante el proyecto se estudiará el funcionamiento de los diferentes aspectos de una extensión Chrome debido a que los módulos que se implementarán usarán funcionalidades propias del navegador.

Por último, se definirán e implementarán casos de prueba para asegurar que los nuevos módulos no interaccionan los ya implementados en el proyecto WacLine.

Entre los objetivos secundarios, destaca el aprendizaje y uso de herramientas para el desarrollo de proyectos SPL como pure::variants<sup>4</sup>, las cuales cada vez irán cobrando más importancia ya que disminuye los costes de producción al desarrollar un único proyecto y nos otorga flexibilidad a la hora de extender y producir nuevas aplicaciones que compartan funcionalidades con otras.

Por último, otro de los objetivos secundarios es demostrar las competencias necesarias para definir el alcance y la realización de un proyecto. Para ello será necesario identificar las tareas y los riesgos que puedan surgir a lo largo del proyecto para poder abordarlos correctamente y completar el proyecto con éxito.

## 2.3. Alcance

Para el desarrollo de una adecuada planificación, es necesario definir correctamente el alcance del proyecto. Para ello es necesario identificar los procesos necesarios para que el proyecto incluya todo el trabajo requerido para completarlo con éxito.

Con motivo del ya comentado uso de la SPL WacLine, basada en anotaciones web, y

---

<sup>4</sup>pure::variants: <https://www.pure-systems.com/products/pure-variants-9.html>

realizado por el grupo de investigación ONEKIN, se excluyen del alcance del proyecto la implementación de las funcionalidades correspondientes al almacenamiento de las anotaciones, definición de criterios de anotación y anotaciones web.

Sin embargo, alguno de los módulos que sí serán desarrollados extenderán los módulos anteriormente mencionados además de implementar funcionalidades específicas para este proyecto. Se comprobará la compatibilidad con los módulos ya implementados en WacLine.

Este proyecto cuenta con ciertas limitaciones impuestas por WacLine, como el uso de la herramienta `pure::variants` para definir la SPL.

Queda fuera del alcance de este proyecto realizar una evaluación de las funcionalidades con usuarios reales para ver su usabilidad. Además de realizar el oportuno tratamiento de datos en caso de que los usuarios finales usen la aplicación para almacenar datos de carácter personal.

## 2.4. Estructura de descomposición de trabajo

Para el desarrollo del proyecto, al definir la implementación por módulos, se ha optado por una metodología en la que se pueda organizar el trabajo de manera que se puedan observar resultados y realizar pruebas de manera incremental y continua.

Para ello, se ha adoptado un ciclo de vida incremental, creando así en distintas iteraciones, versiones de los módulos las cuales aporten nuevas funcionalidades y mejoras con respecto a las versiones anteriores.

Este ciclo de vida nos permitirá analizar cualquier cambio realizado en los módulos para comprobar su correcto funcionamiento y así ir alcanzando los distintos objetivos definidos en el proyecto.

A continuación se describirán todos los paquetes de trabajo y sus tareas explicando la finalidad de cada una de ellas. Estos paquetes y tareas se resumen en la figura 2.1. Consta de dos niveles: el primero hace referencia a las fases del proyecto mientras que en el segundo se encuentra cada paquete de trabajo asociado a cada fase correspondiente.



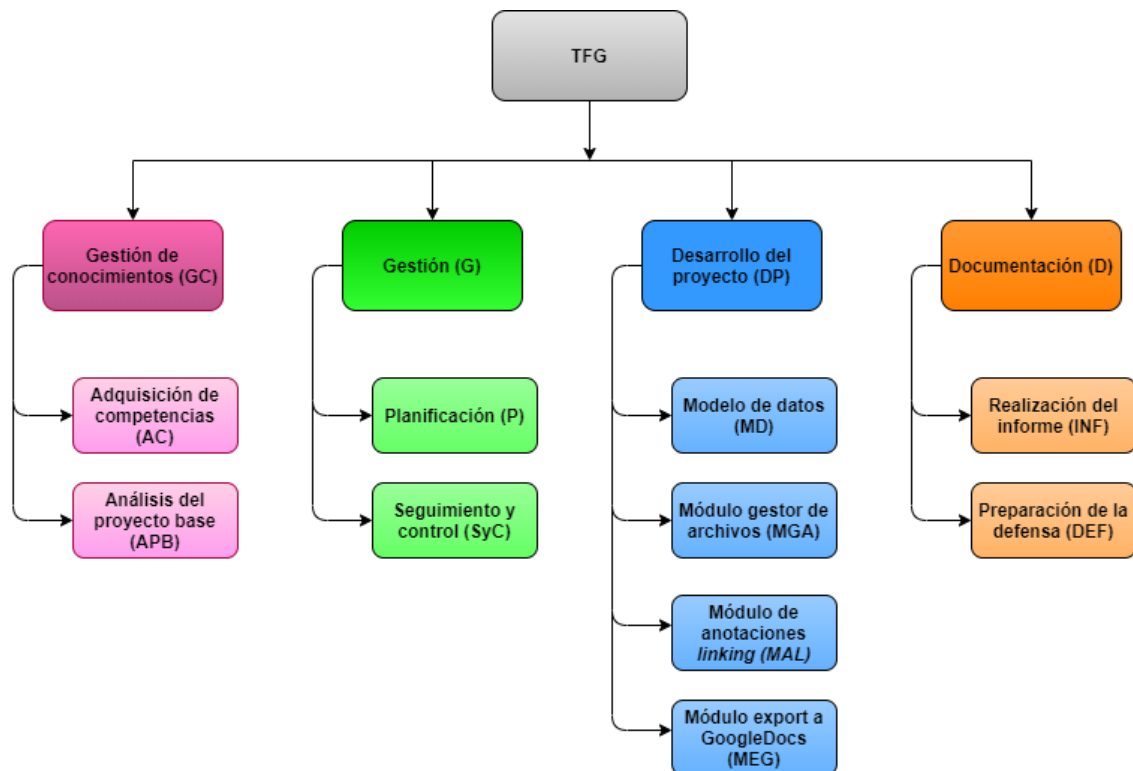


Figura 2.1: Diagrama E.D.T

#### 2.4.1. Fase de Gestión de Conocimiento (GC)

En el paquete de trabajo *Adquisición de Competencias (AD)* incluirá las tareas necesarias relativas a la adquisición de conocimientos para el correcto desarrollo del proyecto. Las tareas son:

- AC.1: Conocer y alcanzar un conocimiento intermedio sobre el desarrollo de proyectos SPL en la herramienta pure::variants.
- AC.2: Conocer y alcanzar un conocimiento intermedio sobre la arquitectura y la implementación de extensiones para el navegador Chrome de la empresa tecnológica Google.
- AC.3: Conocer y alcanzar un conocimiento intermedio sobre el lenguaje JavaScript en su versión ES6 y la programación orientada a eventos.
- AC.4: Conocer y alcanzar un conocimiento intermedio sobre el diseño e implementación de plantillas personalizadas en la suite de Google, concretamente para la aplicación GoogleDocs.

- AC.5: Conocer y alcanzar un conocimiento intermedio sobre la tecnología Node.js utilizada para convertir nuestro proyecto de ES6 a ES5.
- AC.6: Conocer y alcanzar conocimientos básicos sobre las herramientas secundarias necesarias para el desarrollo del proyecto, incluyendo librerías o programas concretos utilizados en momentos puntuales.

En el paquete de trabajo *Análisis de WacLine SPL (APB)* incluirá las tareas necesarias relativas al estudio de la arquitectura e implementación del proyecto de partida. Dichas tareas son:

- APB.1: Analizar la estructura de WacLine SPL para entender su funcionamiento y anotar las dudas para resolverlas con la persona que desarrolló el proyecto.
- APB.2: Resolver las dudas relativas a la estructura de WacLine SPL.

#### 2.4.2. Fase de Gestión (G)

El paquete de trabajo de *Planificación (P)* incluirá las tareas de planificación inicial, así como aquellas tareas necesarias para mantener una adecuada planificación. Dichas tareas son:

- P.1: Tareas relacionadas con la identificación de requisitos, toma de decisiones iniciales, análisis de información, resolución de dudas e identificación de los riesgos que podrían surgir durante la realización del proyecto.
- P.2: Planificación inicial orientada a la preparación del entorno de desarrollo del proyecto.
- P.3: Actualización, si fuera necesaria, de la planificación inicial.

El paquete de trabajo *Seguimiento y Control (SyC)* agrupará las tareas necesarias para garantizar el correcto desarrollo del proyecto, controlando así la planificación inicial con lo finalmente ocurrido para así poder gestionar las modificaciones. Dichas tareas son:

- SyC.1: Reuniones con la tutora a lo largo del desarrollo del proyecto.
- SyC.2: Elaboración de un documento donde se refleje el estado de cada tarea pendiente.

- SyC.3: Configuración de la aplicación Jiffy<sup>5</sup> para controlar el tiempo dedicado a cada paquete de trabajo.
- SyC.4: Recopilación de información relevante sobre el desarrollo del proyecto.
- SyC.5: Contraste de la información recopilada con la planificación inicial para identificar las desviaciones más significativas y los riesgos emergentes.

### 2.4.3. Desarrollo del proyecto (DP)

El paquete de trabajo *Modelo de datos (MD)* incluirá las tareas referentes al análisis del modelo de datos de las anotaciones web y de los documentos de jurisprudencia. Dichas tareas son:

- MD.1: Entender el funcionamiento de las anotaciones web y las recomendaciones de la W3C para su diseño e implementación debido a su uso en el proyecto de partida.
- MD.2 Entender cómo se estructuran los documentos de jurisprudencia para desarrollar una estructura adecuada de las plantillas de GoogleDocs.

El paquete de trabajo *Módulo gestor de archivos (MGA)* incluirá las tareas necesarias para el desarrollo del módulo encargado de gestionar la descarga y almacenamiento de los documentos de jurisprudencia. Dichas tareas son:

- MGA.1: Análisis de las diferentes casuísticas que podemos encontrar en los formularios de acceso a las bases de datos.
- MGA.2 Implementar el formulario de consulta y la posterior presentación de los resultados obtenidos en las diferentes bases de datos.
- MGA.3: Implementar la descarga y la visualización de los diferentes documentos descargados.

El paquete de trabajo *Módulo de anotaciones linking (MAL)* agrupará las tareas necesarias para la implementación del módulo encargado de la funcionalidad de asociar anotaciones entre ellas. Dichas tareas son:

---

<sup>5</sup>Jiffy: <https://jiffy.nu>

- MAL.1: Análisis de la actual implementación de las anotaciones, para no generar conflictos con las nuevas implementaciones.
- MAL.2: Implementación de las anotaciones linking entre documentos, controlando que no se generan conflictos con la anterior implementación de las anotaciones web.

El paquete de trabajo *Módulo export a GoogleDocs (MEG)* incluirá las tareas necesarias para implementar el módulo encargado del procesamiento de las anotaciones realizadas y el posterior volcado a GoogleDocs usando una plantilla personalizada.

- MEG.1: Diseño e implementación de una plantilla compatible con GoogleDocs que se ajuste a los criterios de marcaje definidos por el usuario.
- MEG.2: Implementar la automatización del volcado de datos a GoogleDocs después del procesamiento de las anotaciones recopiladas por el usuario.

#### 2.4.4. Documentación (D)

El paquete de trabajo *Realización del informe (INF)* agrupará todas las tareas necesarias para la realización del informe del trabajo de fin de grado. Dichas tareas son:

- INF.1: Preparar y configurar el entorno LaTeX para el desarrollo de la memoria.
- INF.2: Desarrollo de la memoria y el póster para la exhibirlo en la facultad.

El paquete de trabajo *Preparación de la Defensa (DEF)* incluirá las tareas necesarias para preparar la defensa del trabajo de fin de grado. Dichas tareas son:

- DEF.1: Crear el documento para la presentación, que servirá de apoyo visual en la defensa del proyecto.
- DEF.2: Preparar la defensa repasando y mejorando, en caso de que sea necesario, el documento de presentación creado.

## 2.5. Planificación temporal

Una vez definido el alcance, es importante desarrollar una planificación donde se indiquen las fechas de entrega y plazos previstos para la realización del proyecto, procurando así que esta sea lo más realista posible.

En la tabla 2.1 se indican las horas de dedicación estimadas para cada una de las tareas detalladas anteriormente. También se han agrupado las horas totales de cada paquete de trabajo definido en el E.D.T así como las horas estimadas en para el proyecto entero.

Usando las estimaciones de dedicaciones entre tareas, en la figura 2.2 y 2.3 se presenta el Diagrama de Gantt, donde se definen los distintos periodos en los cuales se van a realizar cada tarea como los hitos más significativos.

Para definir los periodos de las tareas hay que ser consciente de la situación actual ya que se tendrá que desarrollar en paralelo con el desarrollo de las prácticas en empresa.

Hasta el día 13 de Abril, coincidiendo con la semana santa, el proyecto se desarrollará a la vez que las prácticas en empresa, por lo que la dedicación semanal se verá reducida a 4 horas en consecuencia. A partir de dicho día, se dedicarán completamente las 8 horas de trabajo al proyecto aumentando la dedicación sustancialmente. Además se usarán algún que otro día festivo para recuperar en parte la dedicación consumida por las prácticas.

Tareas	Estimación de horas
Proyecto Fin de Grado	319
<b>Fase de Gestión de Conocimiento (GC)</b>	<b>65</b>
Adquisición de Competencias (AC)	25
AC.1: pure::variants	5
AC.2: Extensión Chrome	3
AC.3: JS ES6	5
AC.4: Plantillas GoogleDocs	5
AC.5: Node.js	2
AC.6: Herramientas secundarias	5
Análisis del proyecto base (APB)	40
APB.1: Arquitectura base	30
APB.2: Dudas	10
<b>Fase de Gestión (G)</b>	<b>47</b>
Planificación (P)	19
P.1: Requisitos	2
P.2: Entorno de desarrollo y planificación	14
P.3: Actualización de planificación	3
Seguimiento y Control (SyC)	28
SyC.1: Reuniones	10
SyC.2: Dedicaciones	5
SyC.3: Entorno de control de dedicaciones	2
SyC.4: Recopilación de información	5
SyC.5: Contraste de información	6
<b>Fase Desarrollo del proyecto (DP)</b>	<b>155</b>
Modelo de datos (MD)	20
MD.1: Modelo Anotaciones Web	10
MD.2: Estructura jurisprudencia	10
Módulo gestor de archivos (MGA)	55
MGA.1: Análisis	15
MGA.2: Búsqueda	20
MGA.3: Descarga	20
Módulo de anotaciones linking (MAL)	50
MAL.1: Análisis	20
MAL.2: Implementación	30
Módulo export a GoogleDocs (MEG)	30
MEG.1: Plantilla	10
MEG.2: Procesamiento de datos	20
<b>Fase Documentación (D)</b>	<b>52</b>
Realización del informe (INF)	42
INF.1: Entorno LaTeX	2
INF.2: Desarrollo memoria	40
Preparación de la defensa (DEF)	10
DEF.1: Presentación	4
DEF.2: Revisión	6

**Tabla 2.1:** Dedicaciones estimadas en horas para cada tarea

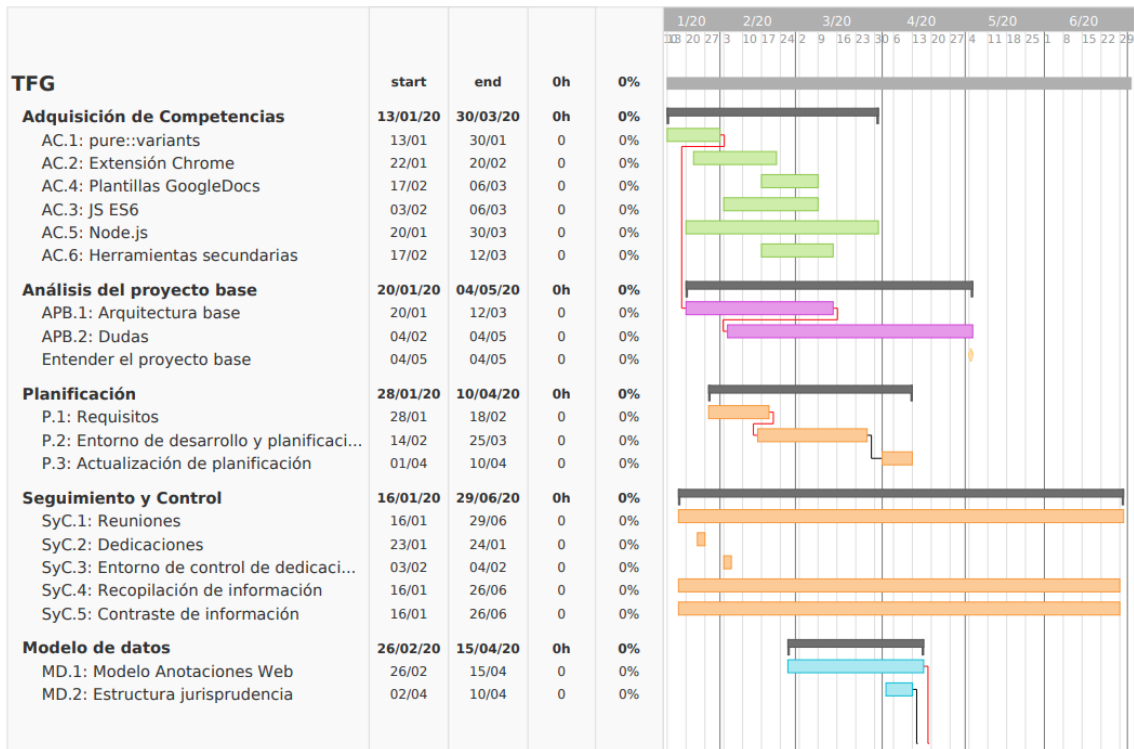


Figura 2.2: Diagrama de Gantt por meses - Primera parte

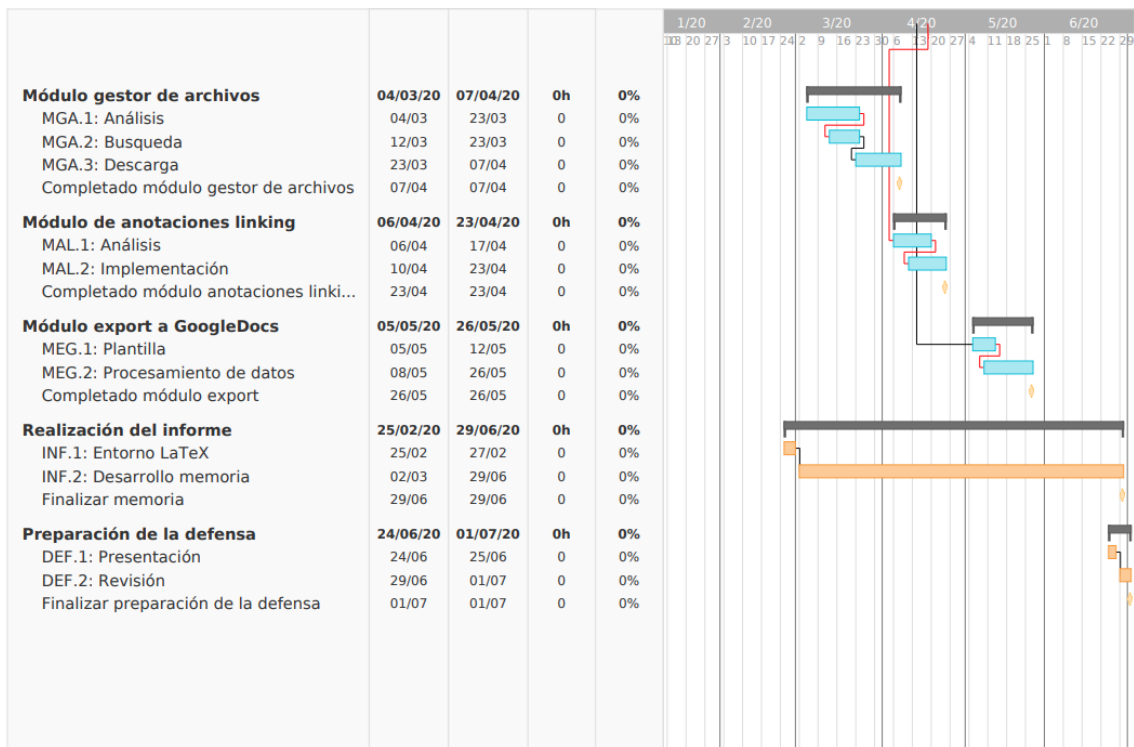


Figura 2.3: Diagrama de Gantt por meses - Segunda parte

## 2.6. Herramientas

En este apartado se presentan las herramientas y tecnologías que se van a utilizar para el desarrollo del proyecto.

### 2.6.1. GitHub y Git

GitHub<sup>6</sup> se trata de una plataforma propiedad de Microsoft que permite el almacenamiento de proyectos utilizando la tecnología Git.

Git<sup>7</sup> se trata de una tecnología que nos permite hacer un control de versiones de nuestro código pudiendo ver en todo momento los cambios realizados entre una versión y otra muy útil en desarrollos cooperativos.

La plataforma se usará para tener disponible en todo momento una copia de seguridad y una versión estable del proyecto además de llevar el control de versiones utilizando la herramienta Git.

### 2.6.2. Time Machine

Time Machine es un software diseñado por Apple para realizar copias de seguridad. Como se puede suponer se utilizará este software para realizar copias incrementales todos los días en un disco duro extraíble.

### 2.6.3. Overleaf y LaTeX

LaTeX<sup>8</sup> es un sistema de composición de textos que facilita la creación de documentos técnicos y científicos. Se utilizará esta tecnología para el desarrollo del proyecto debido a las facilidades que aporta a la hora de diseñar la estructura del proyecto y la modularización de la memoria.

Para usar esta tecnología se usará la plataforma Overleaf<sup>9</sup> la cual nos permitirá realizar el documento online de una forma clara y sencilla pudiendo visualizar el documento resul-

---

<sup>6</sup>GitHub: <https://github.com/>

<sup>7</sup>Git: <https://git-scm.com/>

<sup>8</sup>LaTeX: <https://www.latex-project.org/>

<sup>9</sup>Overleaf: <https://https://es.overleaf.com/>



tante. Además al tratarse de una plataforma online, nos da la ventaja de poder acceder al documento desde cualquier dispositivo y tener siempre una copia disponible.

#### 2.6.4. TeamGantt

TeamGantt<sup>10</sup> es una plataforma online que nos permite definir fácilmente Diagramas de Gantt y compartirlos con más gente de una forma intuitiva.

Se usará la plataforma para la definición del Diagrama de Gantt del proyecto como para realizar el seguimiento del mismo.

#### 2.6.5. Draw.io

Draw.io<sup>11</sup> se trata de una herramienta OpenSource que permite crear una gran variedad de diagramas.

Esta herramienta será utilizada para la creación de los diagramas del proyecto a excepción del diagrama de Gantt.

#### 2.6.6. Visual Studio Code

Visual Studio Code<sup>12</sup> es un editor de texto muy extendido y desarrollado por Microsoft para los principales sistemas operativos. Incluye soporte para la depuración, control de versiones con Git, finalización inteligente de código, fragmentos y refactorización de código.

Este será el principal editor de código debido a su interfaz potente y minimalista y a la facilidad de uso. Nos permitirá una mayor productividad a la hora de movernos, refactorizar y desarrollar nuestro código.

---

<sup>10</sup>TeamGantt: <https://www.teamgantt.com/>

<sup>11</sup>Draw.io: <https://www.draw.io/>

<sup>12</sup>VSCoode: <https://code.visualstudio.com/>

### 2.6.7. Google Chrome

Google Chrome<sup>13</sup> es el navegador web más utilizado y está desarrollado por Google. Lo usaremos para utilizar nuestra extensión resultante del proyecto.

### 2.6.8. pure::variants

pure::variants<sup>14</sup> es un framework, que trabaja sobre el editor de texto Eclipse, el cual nos permite diseñar e implementar Líneas de Producto de Software (SPL). Se usará este framework para definir y generar nuestra aplicación.

### 2.6.9. Eclipse

Eclipse<sup>15</sup> es un editor de código compuesto por herramientas OpenSource muy conocido desarrollado por Eclipse Foundation.

Debido a la exclusiva disponibilidad de la tecnología pure::variants para este editor de código será necesario para la implementación de la SPL y la posterior generación de la aplicación.

### 2.6.10. Node.js y ECMAScript

Node.js<sup>16</sup> es un entorno de ejecución multiplataforma OpenSource. Se utiliza en el backend de las aplicaciones y está basado en el lenguaje de programación ECMAScript y orientado a eventos.

ECMAScript es una definición del lenguaje realizada por ECMA International la cual implementa el lenguaje de programación JavaScript. Para este proyecto se usará la versión ES6 del lenguaje y la plataforma Node.js para gestionar las dependencias del código y generar e código de la extensión final.

---

<sup>13</sup>Chrome: [https://www.google.com/intl/es\\_es/chrome/](https://www.google.com/intl/es_es/chrome/)

<sup>14</sup>pure::variants: <https://www.pure-systems.com/products/pure-variants-9.html>

<sup>15</sup>Eclipse: <https://www.eclipse.org/downloads/>

<sup>16</sup>Node.js: <https://nodejs.org/es/>

## 2.7. Gestión de Riesgos

En todo proyecto importante pueden aparecer diversos riesgos que hagan peligrar nuestra planificación. Identificar estos riesgos nos ayuda a llevar una mejor gestión de ellos teniendo la capacidad de prevenirlos en mayor medida así como de mitigarlos de una forma eficaz.

A continuación se identificarán los posibles riesgos para este proyecto, especificando también, las medidas adoptadas para la mitigación de estos riesgos o los planes de prevención adquiridos para evitar dichos riesgos.

### 2.7.1. Dificultad compaginando el proyecto con las prácticas

El comienzo del TFG coincide con el comienzo de las prácticas en empresa, por lo que durante los 3 primeros meses del proyecto, este deberá estar compaginado con las prácticas. Esto implica el riesgo de no ser capaz de compaginar correctamente ambas actividades.

- **Probabilidad:** media.
- **Impacto:** medio.
- **Consecuencias:** retraso en el cumplimiento de la planificación inicial.
- **Prevención:** elaborar la planificación inicial realista teniendo en cuenta los tres primeros meses de carga extra de trabajo.
- **Plan de mitigación:** reservar sábados, domingos y días festivos para recuperar las horas previstas y que no se hayan podido invertir durante la semana.

### 2.7.2. Dificultad en el aprendizaje de las herramientas y tecnologías necesarias

La gran mayoría de tecnologías y herramientas usadas para la elaboración del TFG no se han utilizado anteriormente, necesitando un periodo de aprendizaje. Esto implica el riesgo de tener más dificultad de la esperada a la hora de usar dichas herramientas lo que implica un tiempo de aprendizaje mayor de lo esperado.

- **Probabilidad:** bajo.
- **Impacto:** medio.
- **Consecuencias:** retraso en el desarrollo de funcionalidades del proyecto dificultando así el cumplimiento de los plazos previstos inicialmente.
- **Prevención:** reservar más tiempo al paquete de trabajo referido a la adquisición de conocimientos, teniendo en cuenta la cantidad tecnologías nuevas que se utilizan.
- **Plan de mitigación:** adelantar las tareas que no dependan de estas nuevas tecnologías intentando causar el menor desajuste en la planificación. Replanificar si lo anterior no es posible.

### 2.7.3. Dificultad en el entendimiento de WacLine SPL

Debido a que este proyecto extiende la SPL WacLine, desarrollado por otra persona, implica la adaptación de nuestra forma de trabajar a la estructura del proyecto.

- **Probabilidad:** alta.
- **Impacto:** alto.
- **Consecuencias:** retraso en el desarrollo de funcionalidades del proyecto dificultado así el cumplimiento de los plazos previstos inicialmente.
- **Prevención:** reservar más tiempo al paquete de trabajo referido al entendimiento del proyecto base estando en contacto vía email y realizando las reuniones presenciales posibles con el anterior desarrollador para resolver las dudas surgidas referentes al proyecto.
- **Plan de mitigación:** adelantar las tareas que no dependan de la implementación de los módulos en el proyecto base. Replanificar si lo anterior no es posible.

### 2.7.4. Planificación incorrecta

En todo proyecto de cierta envergadura, donde es necesaria la realización de una planificación para llevarlo a cabo, existe el riesgo de que dicha planificación no se ajuste correctamente.

- **Probabilidad:** media.
- **Impacto:** alto.
- **Consecuencias:** retraso en el cumplimiento de los plazos previstos.
- **Prevención:** realizar una planificación realista asignando mayores plazos a las tareas disponiendo así de mayor margen para los errores.
- **Plan de mitigación:** replanificar los plazos destinados a cada tarea intentando reducir, en lo mejor posible, la fecha de entrega final.

#### 2.7.5. Problemas técnicos del software utilizado

El proyecto depende de tecnologías ajenas que permiten tanto el desarrollo como su posterior utilización, lo que implica el riesgo de tener problemas con dichas tecnologías.

- **Probabilidad:** baja.
- **Impacto:** medio.
- **Consecuencias:** retrasos en el cumplimiento de ciertas tareas, incumpliendo así los plazos de las mismas.
- **Prevención:** realizar un mantenimiento y un control de versiones de las tecnologías utilizadas.
- **Plan de mitigación:** adelantar la realización de otras tareas mientras se soluciona el problema y replanificar si de no ser posible lo anterior.

#### 2.7.6. Problemas técnicos en el hardware utilizado

El proyecto depende del hardware en el que se está desarrollando y posteriormente utilizado, lo que implica el riesgo de tener problemas técnicos con dicho hardware.

- **Probabilidad:** baja.
- **Impacto:** alto.

- **Consecuencias:** retrasos en el cumplimiento de ciertas tareas, incumpliendo así los plazos de las mismas.
- **Prevención:** realizar un mantenimiento del hardware utilizado como añadir equipos redundantes donde sea posible asegurando el funcionamiento de las herramientas necesarias en dichos equipos, además de de realizar copias de seguridad para proporcionar un respaldo de todos los progresos realizados en el proyecto.
- **Plan de mitigación:** remplazar el hardware dañado por otro, ya se dispone de otro PC con todas las herramientas instaladas y actualizadas. En último caso, adquirir nuevo hardware si no supera un presupuesto de 100.

### 2.7.7. Pandemia global

Debido a la expansión del virus SARS-CoV-2 y la enfermedad que provoca, el COVID-19, corre el riesgo de una pandemia global. Esto implica el confinamiento durante la cuarentena para evitar en lo posible el contagio.

- **Probabilidad:** alta.
- **Impacto:** medio.
- **Consecuencias:** imposibilitar las reuniones presenciales tanto con la tutora como con el desarrollador de WacLine SPL, implicando así posibles retrasos en la planificación.
- **Prevención:** realizar las reuniones de forma telemática o comunicarse por email.
- **Plan de mitigación:** replanificar procurando tener el menor impacto posible en la fecha de finalización del proyecto. Se utilizarán los días festivos para adelantar tareas.

### 2.7.8. Problemas de índole personal

Durante el periodo de desarrollo del proyecto pueden surgir problemas inesperados de índole personal.

- **Probabilidad:** baja.

- **Impacto:** medio.
- **Consecuencias:** retraso en el cumplimiento de los plazos previstos en ciertas tareas.
- **Prevención:** no se puede prever ya que depende del problema.
- **Plan de mitigación:** replanificar en consecuencia con los problemas surgidos.

## 2.8. Plan de Calidad

Como cabe de esperar del desarrollo de un proyecto de estas características, es necesario definir los requisitos mínimos de calidad con los que contará. En este apartado se detallan los requisitos mínimos de calidad que deberán cumplir tanto el producto como la memoria.

### 2.8.1. Producto

- **Mantenibilidad:** El producto deberá estar implementado en módulos bien definidos y siguiendo los criterios observados en la SPL WacLine. Del mismo modo, se usarán patrones de diseños para la optimización de la mantenibilidad de los módulos.
- **Funcionalidad:** El producto deberá cumplir con los requisitos funcionales anteriormente definidos y los nuevos módulos implementados no deberán crear conflictos con los módulos ya implementados. Para ello se realizarán una serie de test con los que se podrá comprobar lo ya mencionado.
- **Usabilidad:** El producto deberá ser usable en el navegador Google Chrome independientemente del sistema operativo que use el usuario.

### 2.8.2. Memoria

- **Referencias:** La memoria deberá tener toda la información actualizada y bien referenciada.

- **Extensión:** Atendiendo a las memorias publicadas en ADDI<sup>17</sup> y las proporcionadas por la tutora y otros compañeros, se ha creído conveniente establecer la extensión de la memoria entre las 70 y 100 páginas.

---

<sup>17</sup><https://addi.ehu.es/handle/10810/13140>



## 3. CAPÍTULO

---

### SPL y Pure::Variants

---

Como hemos visto en épocas anteriores, antes de la producción en cadena, los productos se hacían individualmente por lo que no había dos iguales. Después de la revolución industrial, con la producción en cadena, se consiguió reducir los costes de producción sustancialmente. Esto fue posible debido a crear masivamente el mismo producto por partes. Empezaron a darse cuenta que cada cliente tenía necesidades distintas, por lo que empezaron a diseñar características variables para los productos, haciendo así que se adaptaran a esas necesidades.

#### 3.1. SPL

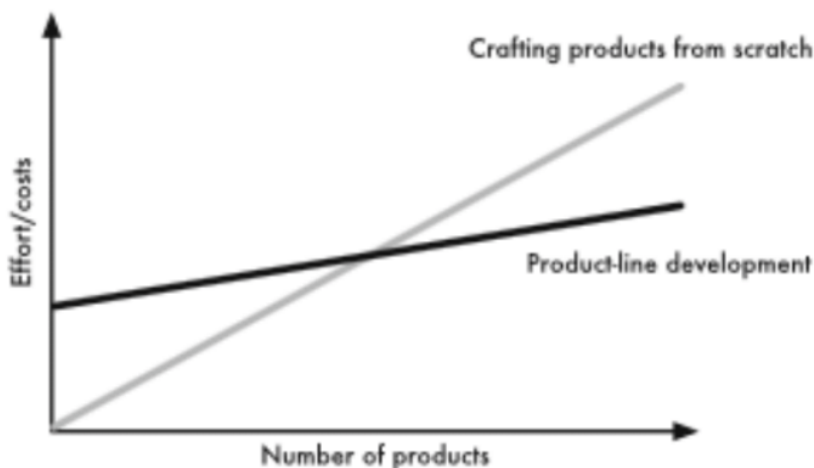
Las Líneas de Producto del Software (SPL) [Eichberg, 2017], nos ofrecen las mismas opciones que las líneas de producción en cadena comentadas anteriormente. Gracias a ellas, en vez de desarrollar una aplicación o un sistema desde cero, nos permiten crear productos que se adapten a las necesidades de cada cliente. Resumiendo, las Líneas de Producto del Software nos permiten, teniendo un núcleo común, crear esos módulos variables que nos permitan cumplir las necesidades específicas.

El proceso de las SPL, se puede comparar a la producción automovilística. Un cliente cuando encarga un vehículo, dispondrá de un paquete de funcionalidades básicas. Pero después, cada cliente tiene la opción de seleccionar opciones extras que completen el producto, cumpliendo así, las necesidades específicas del usuario.

Entre las ventajas que nos ofrecen las SPL, estas son las más importantes [Eichberg, 2017]:

- **Reducción de costes.** Al no tener que implementar un producto desde cero para cada nuevo cliente, ahorran en costes de diseño y desarrollo.
- **Hecho a medida.** Una de las grandes ventajas de las SPL es la gran facilidad para adaptarse a las necesidades específicas del cliente creando los módulos necesarios.
- **Mejora de la calidad.** Al crear los productos en cadena, se pueden establecer mejores controles de calidad y estándares de producción.

En la siguiente figura podemos observar la diferencia de relación calidad/precio entre la producción de software en cadena y la producción individual.



**Figura 3.1:** Diferencia de relación calidad/precio entre las SPL y la producción individual

Un modelo de SPL está definido por características. Estas características son utilizadas para diferenciar los productos resultantes de la SPL. El objetivo de las SPL es organizar y combinar estas características para conseguir un producto hecho a medida que cumpla los requisitos del usuario. Para entender este proceso mejor, a continuación se explicarán algunas definiciones:

- **Características.** Una característica es una funcionalidad concreta de un sistema de software. En las SPL se utilizan estas características para definir las partes comunes y variables de los productos. Además, estas características deben seguir el mismo proceso de planificación, desarrollo y pruebas, además de los ciclos de vida, que sigue cualquier aplicación.

- Producto.** Un producto de una SPL es el resultado de la combinación de características comunes y variables. A su vez, las características seleccionadas para el producto tienen que ser compatibles entre ellas por lo que deberán cumplirse las restricciones propias de cada característica.

Las SPL están definidas por la relación entre las características y los productos que pueden ser creados (Variantes). Al proceso de selección se le llama **selección de características** (feature selection).

Por otra parte, como hemos comentado antes las características pueden tener dependencias o restricciones.

En el desarrollo de una SPL, el **análisis del dominio** tiene una gran importancia a la hora de definir las características de una SPL. Esto nos permitirá definir aquellas características comunes a todos los productos que se engloban en el dominio. Sin embargo, carecería de sentido realizar este proceso si solo se va a desarrollar un único producto.

El proceso de **análisis de variabilidad** en cambio, se encarga de analizar las necesidades específicas de cada cliente para definir las funcionalidades propias de ese producto.

A continuación se detalla el proceso de desarrollo de una SPL.

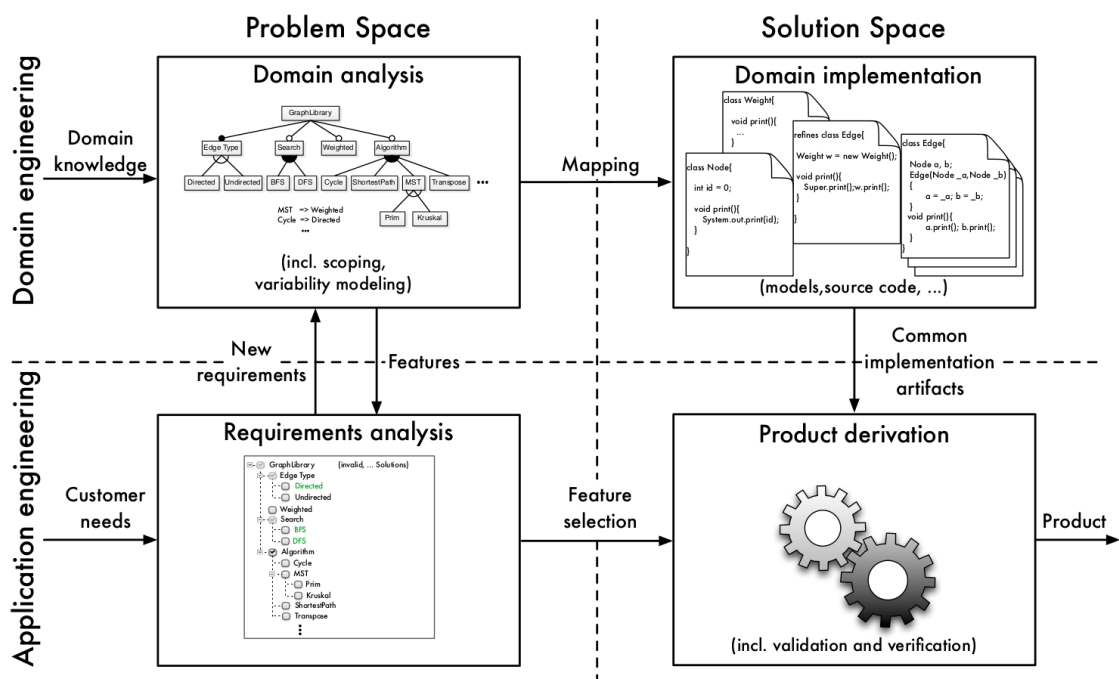


Figura 3.2: Proceso de desarrollo de una SPL [Eichberg, 2017]

El espacio de problemas se refiere a las etapas donde se toman en cuenta todos los proble-

mas identificados en el dominio con el que se está trabajando. Una vez identificados, se analizan y se introducen el análisis del dominio.

En el espacio de soluciones se diseña la implementación del dominio. En este momento de definen las características comunes y variables, la estructura del código y los productos que se van a poder generar.

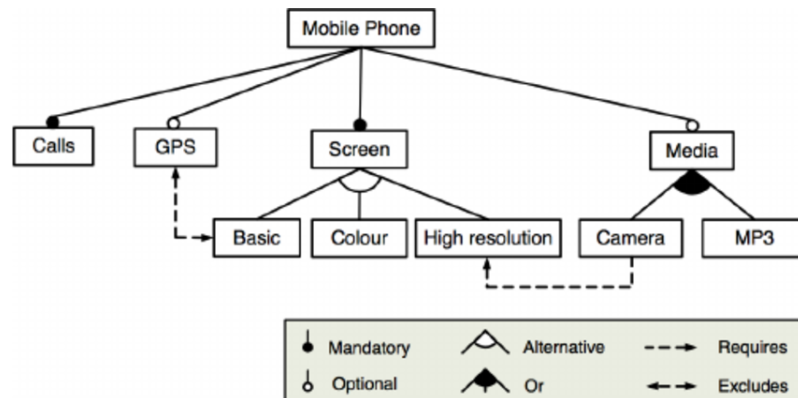
Estos espacios a su vez, se dividen en dos grupos dependiendo de los procesos comentados anteriormente (Ver figura 3.2).

- **Análisis del dominio.** En este apartado se utiliza todo el conocimiento del dominio para generar un **diagrama de características** el cual comentaremos en la siguiente sección.
- **Análisis de requisitos.** En esta etapa se analizarán los requisitos propios de cada característica. Además se analizan los casos especiales de cada característica variable que pueda surgir para definir las dependencias de cada módulo.
- **Implementación del dominio.** En el apartado referente a la implementación se mapean todas las características del dominio con sus respectivas dependencias. Además en esta etapa, se desarrollan todos los módulos los cuales se usaran para la creación de los productos.
- **Generación del producto.** En esta última etapa se seleccionan las características para cada producto cumpliendo las dependencias entre módulos.

La selección de qué características generan un producto se realiza una única vez. De esta forma se eliminan los costes derivados con el diseño de los productos cada vez que se quiere realizar uno.

### 3.1.1. Diagrama de características

En un diagrama de características se definen todas las características identificadas en el dominio. A su vez, estas características se pueden pertenecer a otras características formando así un diagrama estilo árbol el cual dispone de una notación específica para definir las dependencias o restricciones que tiene nuestro dominio. En la figura 3.3 podemos ver un ejemplo de un diagrama de características de un teléfono móvil.



**Figura 3.3:** Ejemplo de un diagrama de características

Como se puede ver en la figura anterior, se distinguen cuatro tipos de notaciones:

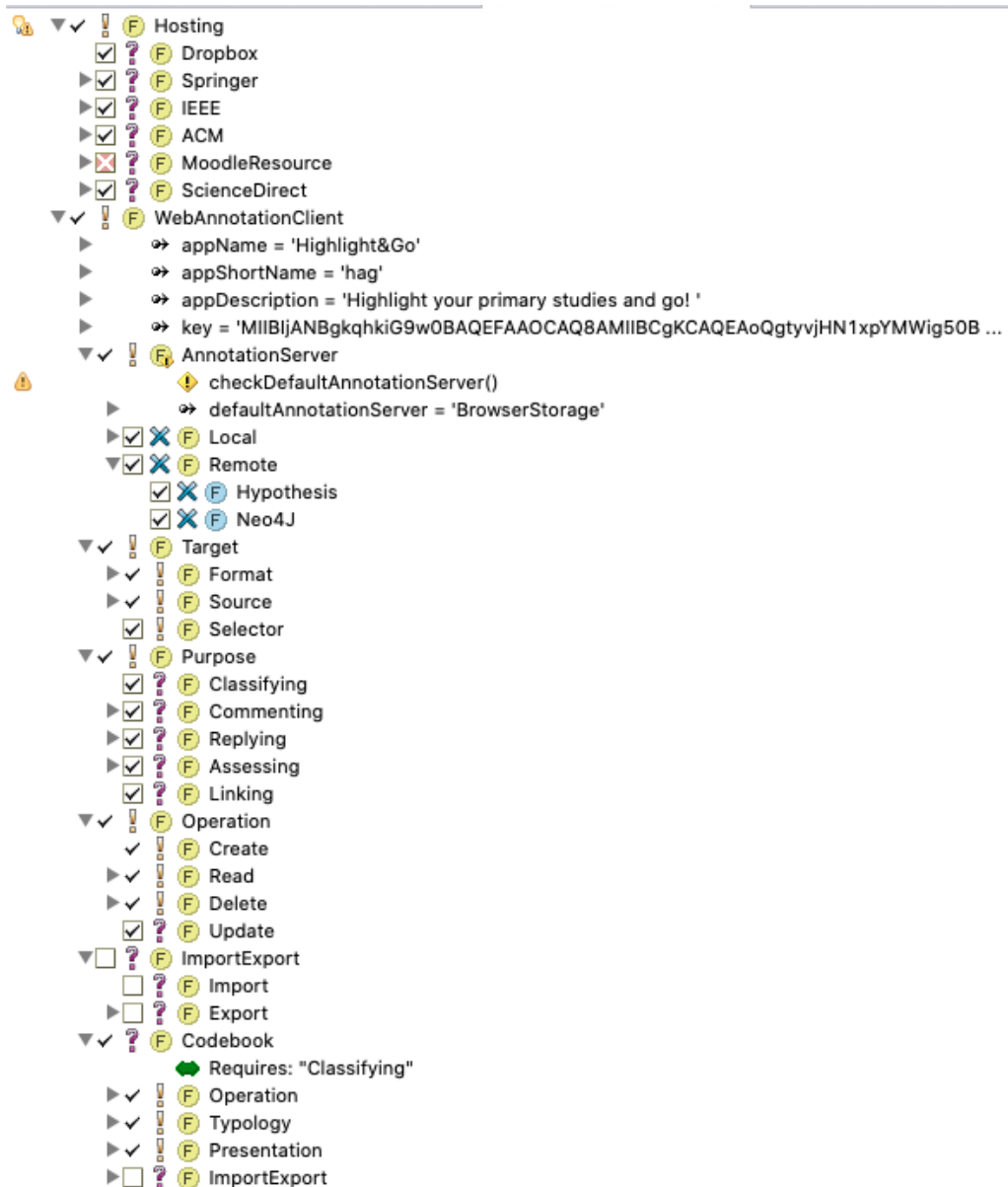
- **Mandatory.** Este tipo de anotación nos indica que la característica es **obligatoria** y deberá estar incluida en todos los productos de la SPL.
- **Optional.** La notación **Optional** nos indica que la característica es opcional. Es decir, puede incluirse o no en los productos de la SPL según la necesidad del cliente.
- **Alternative.** La notación **Alternative** nos indica que solo puede elegirse una característica del grupo para el producto. Es decir, si se selecciona una característica las demás no pueden incluirse en el producto resultante.
- **Or.** Esta última notación nos indica que las opciones del grupo son compatibles, con lo que se pueden elegir tantas como se deseen. Por otro lado, por lo menos tiene que elegirse una característica del grupo para que pertenezca a el producto final.

Además de las notaciones comentadas, en algunos tipos de diagramas de características se pueden observar las dependencias o las restricciones de las características (Ver figura 3.3).

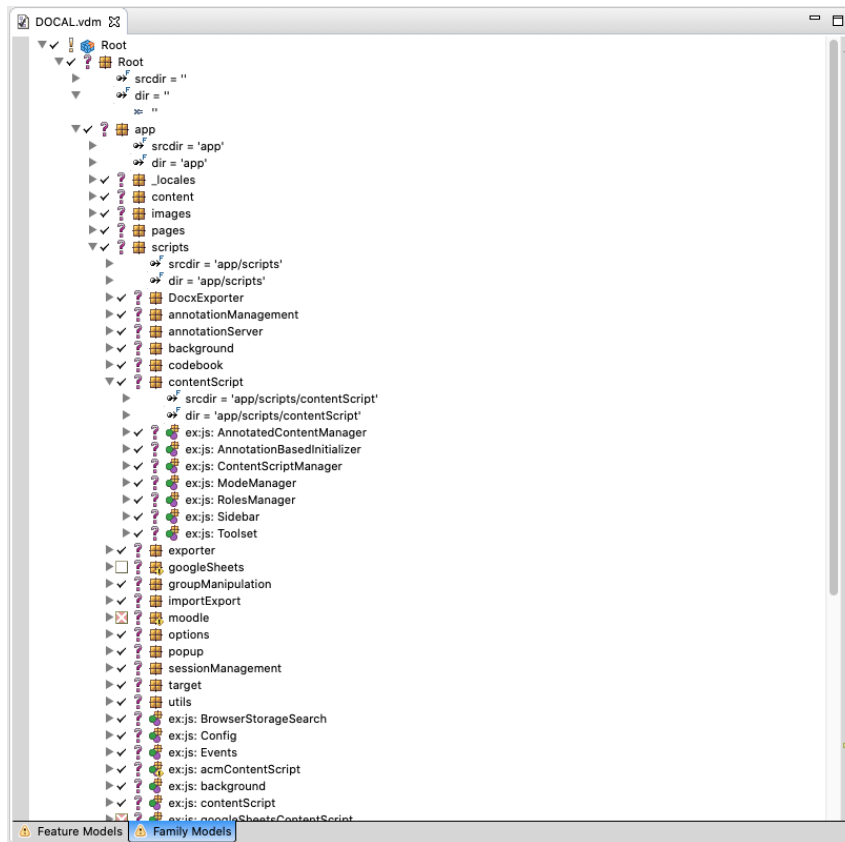
Según este diagrama, un teléfono móvil siempre puede realizar llamadas y tendrá una pantalla que solo podrá ser básica, a color o de alta resolución. Opcionalmente tendrá la característica de GPS y de multimedia, las cuales podrán ser la cámara y/o el MP3. En caso de tener GPS, la pantalla que tenga el dispositivo móvil deberá ser la pantalla básica. Sin embargo, si el dispositivo tiene la opción de cámara deberá tener una pantalla de alta resolución. Con esto podemos inferir que, según este diagrama de características, un dispositivo móvil no puede disponer de GPS y cámara al mismo tiempo ya que cada característica requiere un tipo de pantalla diferente.



A su vez, estas variantes disponen de su propio *Feature Model* con las características seleccionadas y un *Family Model* el cual dispondrá toda la información referida al mapeo de los archivos de nuestra SPL. Es decir, en el *Family Model* podemos ver todos los archivos que incluirá nuestro producto (Ver figura 3.6).



**Figura 3.5:** Selección de las características de una variante



**Figura 3.6:** *Family Model* de DOCAL

Por último, en Pure::variants, a diferencia de otras herramientas para desarrollar SPL, se basa en *anotaciones*. Esto nos permite establecer condiciones en las partes de código correspondientes a las diferentes características. De esta forma podemos desarrollar nuestro código de forma habitual, sin necesidad de optar por arquitecturas de código complejas y de difícil mantenibilidad.

En la figura 3.7 se pueden observar estas *anotaciones*. En el caso de la figura, se hace referencia a la funcionalidad de ordenar los códigos de marcado. Dependiendo de la característica que se seleccione, el producto resultante tendrá únicamente el código referente al método de ordenamiento seleccionado.

```

288 // PVSCl:IFCOND(Alphabetical, LINE)
289 themes.sort((a, b) => a.name.localeCompare(b.name))
290 // PVSCl:ENDCOND
291 // PVSCl:IFCOND(Number, LINE)
292 themes.sort((a, b) => parseFloat(a.name) - parseFloat(b.name))
293 // PVSCl:ENDCOND
294 // PVSCl:IFCOND(Date, LINE)
295 themes.sort((a, b) => a.createdDate - b.createdDate)
296 // PVSCl:ENDCOND

```

**Figura 3.7:** Ejemplo de condiciones de Pure::Variants



## 4. CAPÍTULO

---

### Presentación de las características y organización de WacLine

---

Una vez explicado qué es una SPL y la herramienta que se utiliza para desarrollarla en este TFG, se explicará la estructura de la SPL WacLine.

WacLine, como se ha comentado en secciones anteriores, es la SPL de referencia que se ha utilizado para el desarrollo de este TFG. En este capítulo se detallará la arquitectura y los módulos ya implementados en WacLine los cuales servirán como base del producto resultante de este TFG.

#### 4.1. Arquitectura de las Anotaciones Web

WacLine es la SPL desarrollada por Haritz Medina dentro del grupo de investigación Onekin en la UPV/EHU. Esta SPL trabaja en el dominio de las anotaciones web. Por ese motivo, la arquitectura de WacLine se ha diseñado siguiendo las recomendaciones sobre anotaciones web de la W3C<sup>1</sup>.

En la figura 4.1 se resumen los conceptos más importantes que maneja WacLine, siguiendo las directrices principales de la W3C. Así, el sistema de anotaciones web consta de dos partes: el servidor y el cliente de anotaciones.

---

<sup>1</sup><https://www.w3.org/TR/annotation-model/>

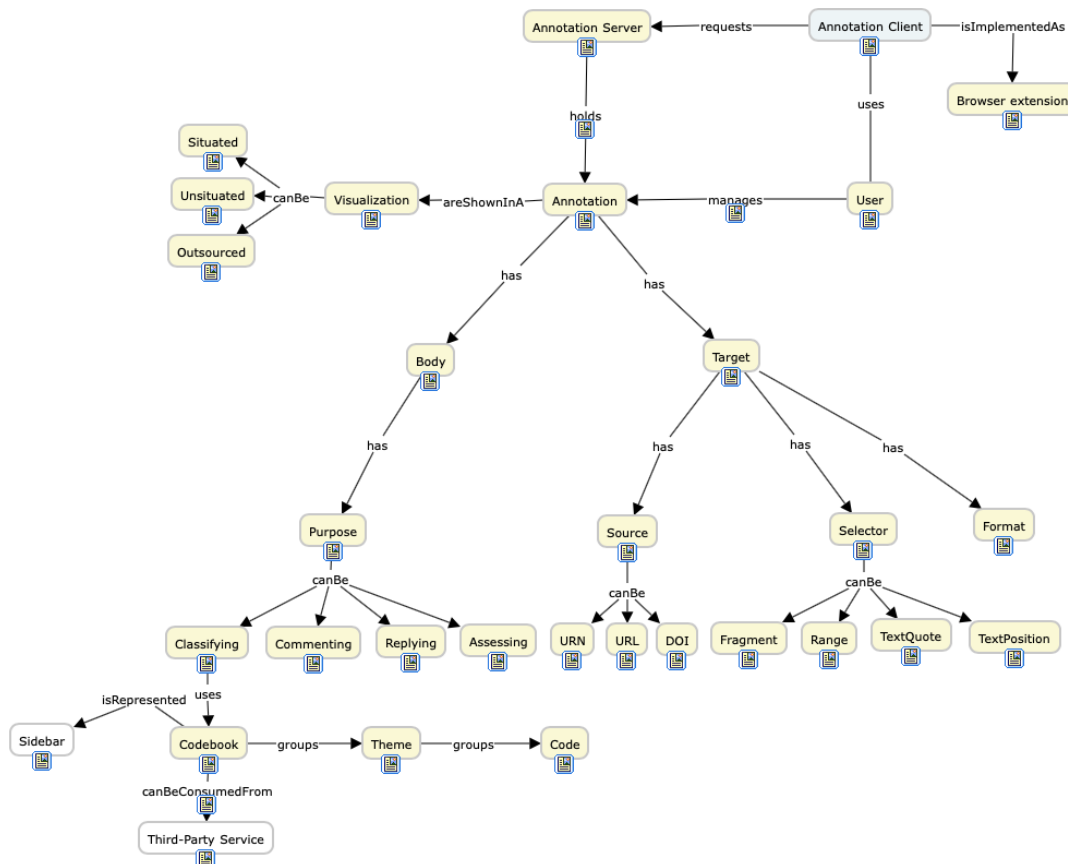


Figura 4.1: Mapa de concepto de WacLine

El servidor de anotaciones será el encargado de guardar y organizar internamente las anotaciones, además de, en la mayoría de servidores, gestionar los usuarios.

El cliente de anotaciones sin embargo, es el encargado de realizar dichas anotaciones y permitir al usuario que pueda interactuar con ellas. Según el modelo de la W3C, las anotaciones están compuestas de dos partes importantes: El *Body* y el *Target*.

El apartado *Body* es el encargado de almacenar todos los datos relacionados con el propósito de esa anotación. Es decir, en el *Body* se almacenan: los comentarios de la anotación, el código de marcado (*codebook*) al que pertenecen, las calificaciones...

En cambio el *Target* es el encargado de crear la anotación en sí. Este consta de tres partes:

- El apartado **Source**: Este apartado es el encargado de almacenar la dirección URI del documento donde se encuentra la anotación. Para los documentos PDF se utiliza el campo URN, para los documentos html alojados en internet la URL, y DOI para los documentos de investigación publicados.

- El **Selector**: Es el encargado de almacenar la información del contenido de la anotación. Se almacena la posición concreta en la que la anotación se encuentra dentro del documento, el texto anotado junto con el texto adyacente para proporcionar contexto.
- El apartado **Format**: Proporciona información sobre el formato del documento donde se encuentra la anotación. Dependiendo de la información recopilada en este apartado se utilizan diferentes herramientas para visualizar el contenido, como por ejemplo, un visualizador de documentos PDF.

Tras ver los conceptos principales, en la siguiente sección se presentará el diseño de las características principales que se han tenido en cuenta en la implementación de WacLine.

## 4.2. Características de WacLine

En la figura 4.2 podemos ver el diagrama de características el cual incluye todas las opciones disponibles en WacLine. WacLine está compuesto por un total de 85 características, de las cuales 23 son del tipo *Mandatory*, 11 del tipo *Alternative*, 40 del tipo *Optional* y 12 del tipo *OR*. Estas características cuentan con sus respectivas restricciones lo que nos hace una idea de la envergadura de la SPL.

A continuación se explicarán las características principales que han influido directamente en el TFG al ser características comunes a todas los productos.

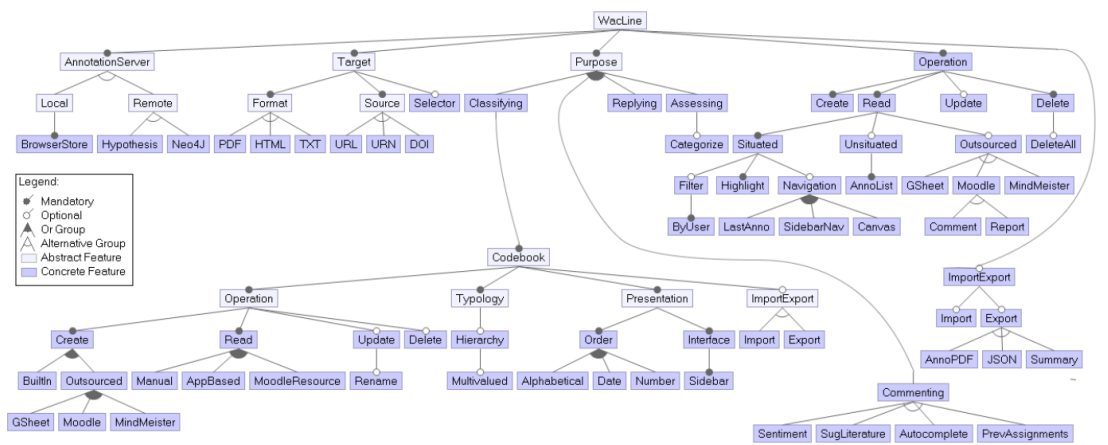
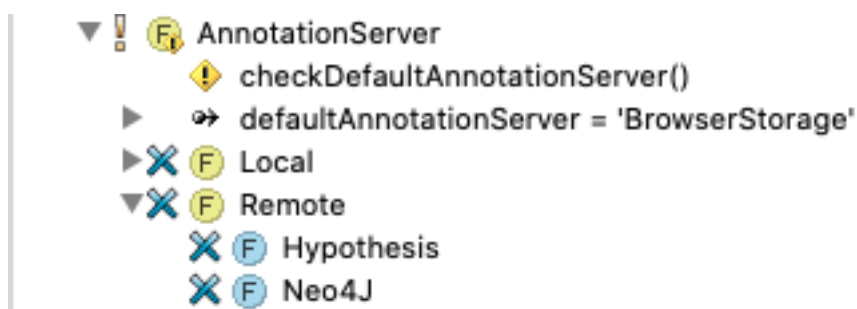


Figura 4.2: Diagrama de Funcionalidades - WacLine

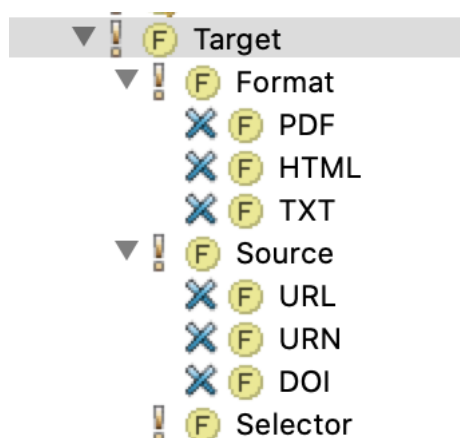
Las variantes generadas por WacLine son clientes de anotaciones, es decir, necesita de servidores externos para gestionar las anotaciones de los usuarios. Aún así, WacLine da la opción al usuario de elegir entre servidores de anotaciones externos como Hypothesis<sup>2</sup> o Neo4J<sup>3</sup> pero también da la posibilidad de utilizar un servidor local (Ver figura 4.3).

Como podemos ver en la imagen se tratan de características tipo *OR*, es decir, las aplicaciones resultantes de WacLine deben incluir al menos una característica del grupo. En el capítulo 3 se han definido los diferentes tipos de características.



**Figura 4.3:** Servidores de Anotaciones - WacLine

En la figura 4.4 podemos ver el apartado *Target* anteriormente comentado. En WacLine tenemos la opción de procesar documentos PDF, HTML y TXT. Como podemos observar el apartado *Format* y el apartado *Source* son grupos *OR* donde se pueden seleccionar un tipo o más de documentos. El *Selector* en cambio, es *Mandatory* lo que nos indica que es obligatorio seleccionarla.



**Figura 4.4:** Target - WacLine

<sup>2</sup>Hypothesis: <https://web.hypothes.is>

<sup>3</sup>Neo4J: <https://neo4j.com>

Como podemos observar en la figura 4.5, las acciones que podemos realizar con las anotaciones (*Purposes*) son las vistas en el diagrama de características (Figura 4.2). Estas acciones son del tipo *Optional* lo que quiere decir que puede incluirse o no y al contrario que con las de tipo *OR*, puede que no haya ninguna seleccionada. También podemos ver como las características de *Replying* y *Assessing* cuentan con una restricción la cual nos advierte de que es necesario en módulo *Commenting* para seleccionar alguna de las dos funcionalidades debido a que las dos características hacen uso de la funcionalidad aportada por *Commenting*.

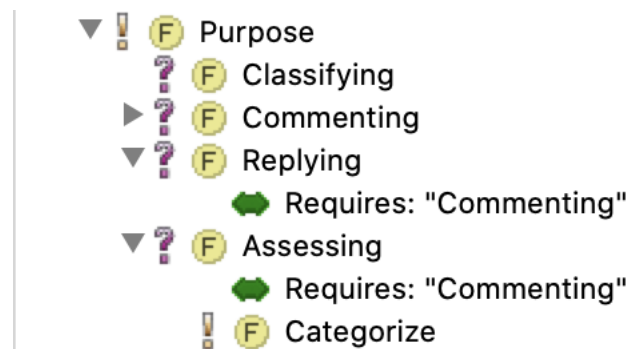
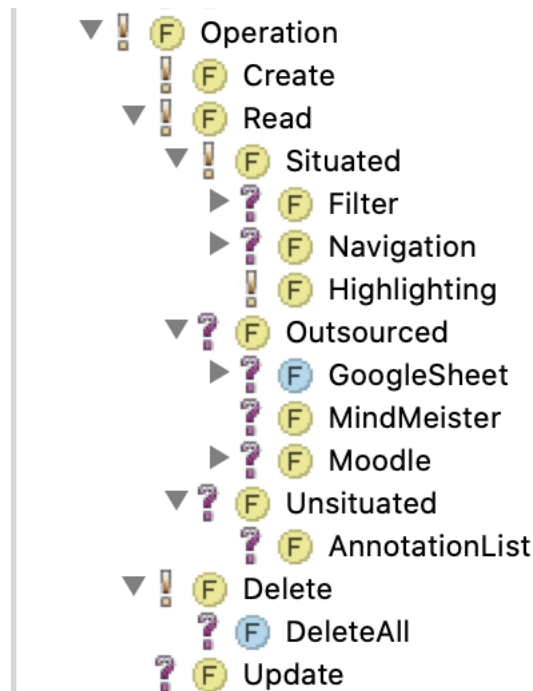


Figura 4.5: Purposes - WacLine

Por último, en la figura 4.6, se pueden observar las operaciones disponibles para las anotaciones. Como podemos ver las operaciones de *Create*, *Read* y *Delete* son obligatorias en cualquier variante que se realice.



**Figura 4.6:** Operaciones - WacLine

A la hora de crear una variante concreta, con la característica *Read* se indica que esta variante tendrá la funcionalidad para resaltar anotaciones en el documento (todas las variantes o productos lo tendrán). Además se podrá elegir entre otras funcionalidades como el filtrado de anotaciones o la navegación entre anotaciones. Además podremos utilizar fuentes externas para importar nuestras anotaciones, por ejemplo desde un documento de cálculo de Google, donde tengamos nuestras anotaciones guardadas, desde la plataforma MindMeister de mapas conceptuales o desde la plataforma Moodle. Esta última se utiliza en la variante Mark&Go de WacLine. Usa la plataforma Moodle para generar rúbricas que se utilizarán como libros de códigos para realizar las anotaciones y la evaluación del documento.

### 4.3. Estructura del código de WacLine

Como hemos comentado en secciones anteriores, WacLine es una SPL de extensiones para el navegador Chrome. Esto hace que siga una estructura donde podemos apreciar tres grupos de ficheros: (Ver figura 4.7).

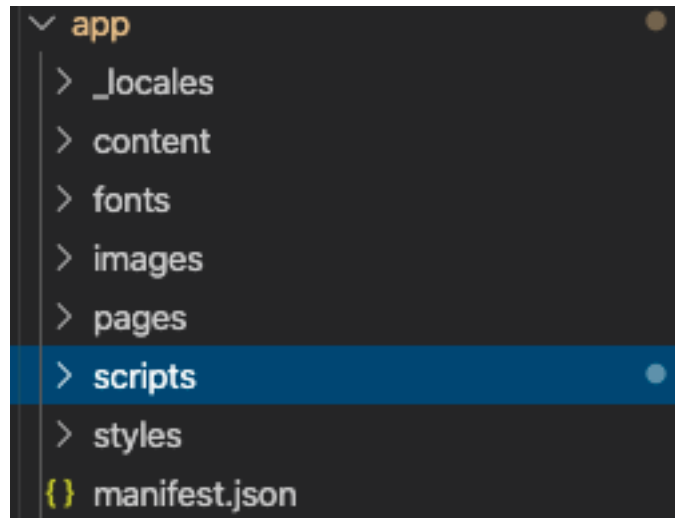
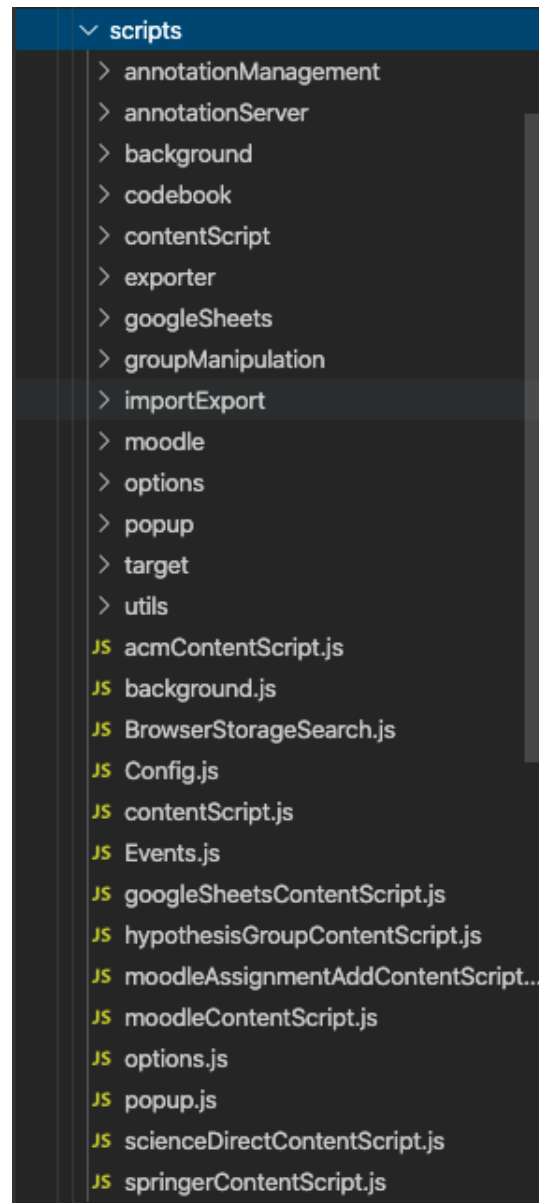


Figura 4.7: Estructura de ficheros - WacLine

- El apartado **Frontend** el cual corresponde a las carpetas: *content*, *fonts*, *images* y *styles*. En estas carpetas se almacenan todos los documentos como html, imágenes, css...
- El apartado **Backend** donde se encuentra toda la lógica de la línea de productos. Como podemos observar en la figura 4.8, en WacLine se ha optado por seguir una estructura orientada al diagrama de características. Esto nos permite gestionar el código y escalarlo fácilmente en un futuro.
- El archivo **Manifest**, el cual define toda la información de la extensión además de indicarle al navegador los archivos que tiene que cargar para que la extensión funcione y los permisos que necesitará nuestra aplicación.

Dado que WacLine produce extensiones que trabajan sobre un navegador y procesan documentos en diferentes formatos, está orientada a eventos. Esto quiere decir que toda la lógica de la aplicación se estructura en eventos los cuales se accionan y procesan. Estos eventos desencadenan otra serie de eventos a su vez, ejecutando las funciones necesarias. Por ejemplo, en la figura 4.9, se puede observar la función encargada de tratar el evento *UpdateSession*. En este caso, la función procesa el evento y a continuación, lanza el evento *SessionUpdated* para que la función correspondiente actualice la información, en la interfaz de usuario, relacionada con la sesión actualizada.



**Figura 4.8:** Estructura de la carpeta Scripts - WacLine



```

updateSessionEventHandler () {
  return (event) => {
    let session = event.detail.session
    window.abwa.annotationServerManager.client.updateAnnotation(
      session.id,
      session.toAnnotation(),
      (err, session) => {
        if (err) {
          Alerts.errorAlert({ text: 'Unexpected error, unable to update annotation' })
        } else {
          session = Session.deserialize(session)
          LanguageUtils.dispatchCustomEvent(Events.sessionUpdated, { session: session })
        }
      })
  }
}

```

**Figura 4.9:** Función encargada de procesar el evento *UpdateSession*

Por último, WacLine está compuesto por alrededor de cien puntos de variación, lo que nos proporciona una idea de la envergadura de la SPL. Estos puntos de variación van desde condiciones simples, como la selección de una características, a condiciones más complejas donde se evalúan la existencia de varias características además de poder encontrar puntos de variación anidados. En la figura 4.10 podemos observar un ejemplo donde en la condición se evalúa si solo se ha seleccionado un tipo de servidor de anotaciones. Si es así, se anidan puntos de variación para cada tipo de servidor. En caso de haberse seleccionado más de un servidor de anotaciones, al generar la variante se añadirá el código correspondiente al apartado *PVSCL:ELSECOND*.

```

// PVSCL:IFCOND(AnnotationServer->pv:SelectedChildren()->pv:Size()==1, LINE)
// PVSCL:IFCOND(Hypothesis, LINE)
this.AnnotationServerClientManager = new HypothesisClientManager()
// PVSCL:ENDCOND
// PVSCL:IFCOND(BrowserStorage, LINE)
this.AnnotationServerClientManager = new BrowserStorageManager()
// PVSCL:ENDCOND
this.AnnotationServerClientManager.init(() => {--
})
// PVSCL:ELSECOND
chrome.runtime.sendMessage({ scope: 'annotationServer', cmd: 'getSelectedAnnotationServer' },
})
// PVSCL:ENDCOND

```

**Figura 4.10:** Ejemplo de punto de variación en el código



## 5. CAPÍTULO

---

### Módulo gestor de documentos

---

En este capítulo se presentará el análisis de las interfaces de las bases de datos seleccionadas y los requisitos del *módulo gestor de documentos*. Además se detallarán los problemas detectados que dieron lugar a desechar el desarrollo del mismo.

#### 5.1. Análisis de las interfaces de las bases de datos

Las interfaces web de las tres bases de datos a utilizar, Aranzadi, Cendoj, Tirant, tienen diferencias importantes. A continuación se analizarán en detalle.

##### Aranzadi

Thomson Reuters Aranzadi<sup>1</sup> es una base de datos de documentos jurídicos perteneciente a la empresa Thomson Reuters. Para acceder a este servicio hay que hacer uso de la biblioteca de la UPV/EHU, acceder a la sección de 'bases de datos y recursos' y seleccionar Aranzadi. Una vez seleccionado nos proporcionará un par usuario/contraseña con el que podremos acceder a su propio buscador.

Como podemos observar en la figura 5.1, algunos campos del formulario de búsqueda, como el rango de fecha, se diferencia del resto teniendo que introducir mediante una lista desplegable el día, mes y año.

---

<sup>1</sup><http://www.aranzadidigital.es/maf/app/authentication/signon?legacy>

The screenshot shows the Aranzadi search interface. At the top, there is a navigation bar with 'Personalizar', 'Alertas', 'Utilidades', and 'Mis Búsquedas'. Below this, the main search area includes a search bar and a 'Buscar' button. The interface is divided into several sections:

- Filters:** Marginal, Texto, Resumen, Voces, Disposiciones examinadas, Artículo (Todos), Tribunal, Fecha (desde/hasta), ECLI (ECLI:ES:APGU:2014:500), Tipo de Resolución, Número de Resolución, Número de Recurso, Jurisdicción/Procedimiento, Ponente, Relevancia.
- Buttons:** Buscar, Borrar, Traspasar consulta a Doct. Adm.
- Right Sidebar:** 'Tesoro de Jurisprudencia' with categories like Constitucional, Civil, Penal, Contencioso-Administrativo, Social, Militar, and CONFLICTOS DE JURISDICCION Y DE COMPETENCIA. It also includes 'Buscar Jurisprudencia sobre una norma' and 'Acuerdos del Tribunal Supremo'.

Figura 5.1: Buscador Aranzadi

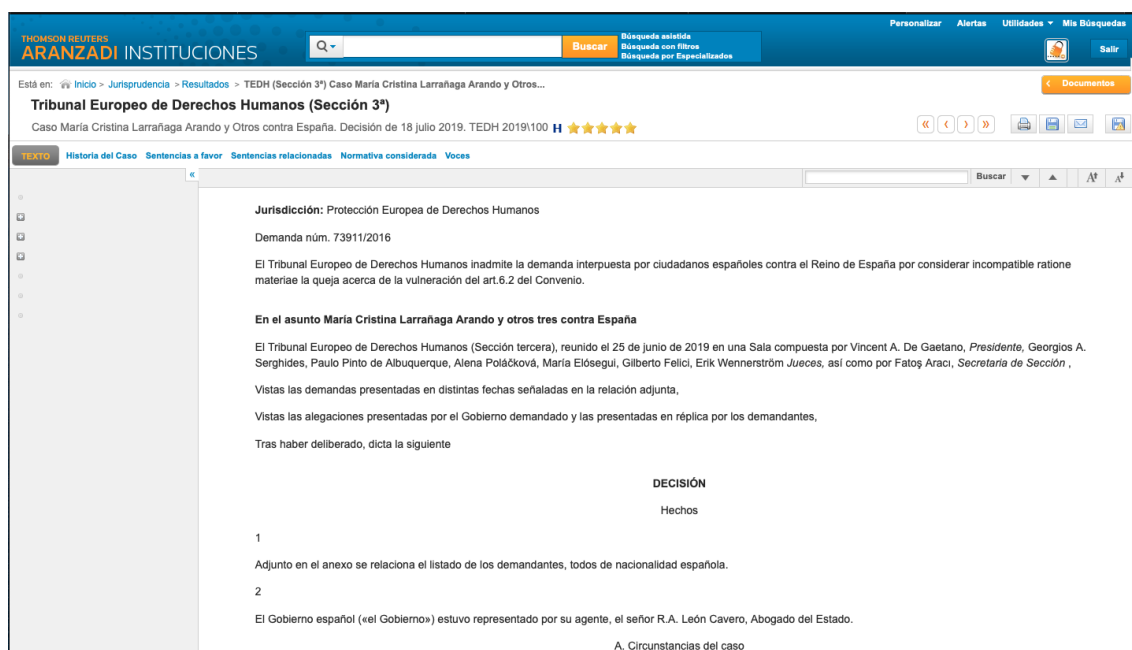
Una vez realizada la búsqueda con los criterios deseados, se nos muestra una lista con los documentos con un breve resumen de su contenido y un enlace para poder visualizarlo dentro de su página web (ver figura 5.2).

The screenshot shows the search results page for 'Jurisprudencia (4000)'. The page is organized into several sections:

- Search Bar:** Includes a search bar, 'Buscar' button, and options for 'Búsqueda avanzada', 'Búsqueda con filtros', and 'Búsqueda por Especializados'.
- Navigation:** 'Inicio > Jurisprudencia > Resultados'.
- Left Sidebar:** 'TERMINOS BUSCADOS' (Tribunal: Tribunal Europeo de Derechos Humanos, Tribunal General de la Unión Europea, Tribunal Constitucional), 'NAVEGACIÓN EN LOS RESULTADOS' (Jurisprudencia (4000)), 'BUSCAR EN LOS RESULTADOS' (Buscar), 'RESULTADOS CLASIFICADOS' (Área, Tribunal, Fecha), 'RESULTADOS' (Jurisprudencia 4000).
- Main Results:** A list of 5 results, each with a checkbox, title, and summary.
  - 1. Caso Ahmadov contra Azerbalyán (5 stars). Summary: DERECHO AL RESPETO A LA VIDA PRIVADA Y FAMILIAR: Injerencias de los poderes públicos: ciudadanía: denegación de documento de identidad azarbayano a ciudadano ruso que entró en el país hace 30 años como estudiante, con permiso temporal, por incumplimiento del requisito de figurar como residente en el momento de entrar en vigor la ley de ciudadanía... Más.
  - 2. Caso María Cristina Larrañaga Arando y Otros contra España (5 stars). Summary: TRIBUNAL EUROPEO DE DERECHOS HUMANOS: Jurisdicción y competencia 1: rationae materiae: en materia de presunción de inocencia: indemnización a las víctimas del GAL y de grupos ultraderechistas españoles: denegación por parte de las autoridades nacionales de las demandas de indemnización adicional por el fallecimiento de sus familiares en atentados t... Más.
  - 3. Caso Karmele Martínez Aguirre y Otros contra España (5 stars). Summary: TRIBUNAL EUROPEO DE DERECHOS HUMANOS: Jurisdicción y competencia 1: rationae materiae: en materia de presunción de inocencia: indemnización a las víctimas del GAL y de grupos ultraderechistas españoles: denegación por parte de las autoridades nacionales de las demandas de indemnización adicional por el fallecimiento de sus familiares en atentados t... Más.
  - 4. Caso María Carmen Forcadell i Lluís y otros contra España (5 stars). Summary: LIBERTAD DE REUNIÓN: Límites: «necesarios en una sociedad democrática»: caso Independencia de Cataluña: suspensión cautelar por el TC de la celebración de un pleno del Parlamento de Cataluña de 9 de octubre de 2017 para que el Presidente de la Generalidad de Cataluña Puigdemont hiciera una valoración de los resultados del referéndum del 1 de octubre... Más.
  - 5. Caso Stern Taulats y Roura Capellera contra España (5 stars). Summary: LIBERTAD DE EXPRESIÓN: Jurisdicción y proceso penal: delito de injurias contra la Corona: condena a quince meses de prisión, con privación del derecho de sufragio pasivo durante el tiempo de condena y al pago de la mitad de las costas del
- Right Sidebar:** 'Relacionados' with a list of related documents and their summaries.

Figura 5.2: Resultado de la búsqueda - Aranzadi

Como podemos ver en la figura 5.3, la visualización de los documentos no está en formato pdf sino que se visualiza en un formato propio. Tenemos la opción de descargarlo mediante la opción disponible en la parte superior derecha después de rellenar el pertinente formulario.



The screenshot displays the Aranzadi legal database interface. At the top, there is a navigation bar with the Thomson Reuters logo and the text 'ARANZADI INSTITUCIONES'. A search bar is located in the center, with a 'Buscar' button and options for 'Búsqueda asistida', 'Búsqueda con filtros', and 'Búsqueda por especialidades'. On the right side of the navigation bar, there are links for 'Personalizar', 'Alertas', 'Utilidades', and 'Mis Búsquedas', along with a 'Salir' button.

The main content area shows the following information:

- Está en: Inicio > Jurisprudencia > Resultados > TEDH (Sección 3ª) Caso María Cristina Larrañaga Arando y Otros...
- Tribunal Europeo de Derechos Humanos (Sección 3ª)
- Caso María Cristina Larrañaga Arando y Otros contra España. Decisión de 18 julio 2019. TEDH 2019/100 H ★★★★★

Below this, there are navigation tabs: 'TEXTO', 'Historia del Caso', 'Sentencias a favor', 'Sentencias relacionadas', 'Normativa considerada', and 'Voces'. The main text area contains the following content:

**Jurisdicción:** Protección Europea de Derechos Humanos

**Demanda núm.** 73911/2016

El Tribunal Europeo de Derechos Humanos inadmite la demanda interpuesta por ciudadanos españoles contra el Reino de España por considerar incompatible *ratione materiae* la queja acerca de la vulneración del art.6.2 del Convenio.

**En el asunto María Cristina Larrañaga Arando y otros tres contra España**

El Tribunal Europeo de Derechos Humanos (Sección tercera), reunido el 25 de junio de 2019 en una Sala compuesta por Vincent A. De Gaetano, *Presidente*, Georgios A. Serghides, Paulo Pinto de Albuquerque, Alena Poláková, María Elósegui, Gilberto Felici, Erik Wennerström *Jueces*, así como por Fatoş Araci, *Secretaria de Sección*,

Vistas las demandas presentadas en distintas fechas señaladas en la relación adjunta,

Vistas las alegaciones presentadas por el Gobierno demandado y las presentadas en réplica por los demandantes,

Tras haber deliberado, dicta la siguiente

**DECISIÓN**

Hechos

1

Adjunto en el anexo se relaciona el listado de los demandantes, todos de nacionalidad española.

2

El Gobierno español («el Gobierno») estuvo representado por su agente, el señor R.A. León Cavero, Abogado del Estado.

A. Circunstancias del caso

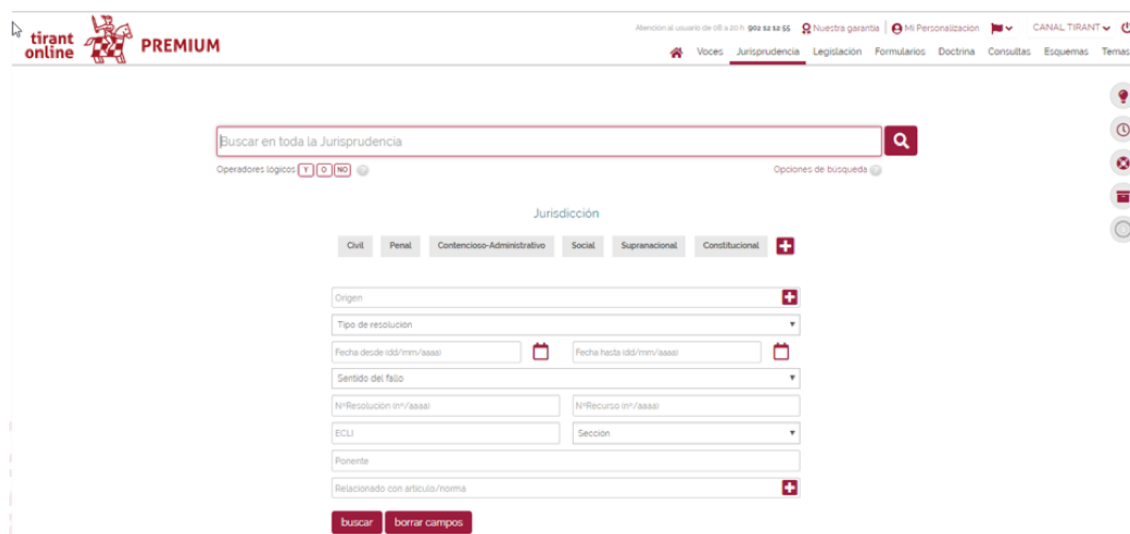
**Figura 5.3:** Visualización de documentos - Aranzadi

## Tirant Online

Tirant Online<sup>2</sup> es una base de datos jurídica de ámbito general creada inicialmente como biblioteca de la Universidad de Valencia. Se puede acceder desde el portal de la biblioteca de la universidad siempre que estés en la red de la UPV/EHU o puedes solicitar una clave en la misma página. Una vez conseguidas las credenciales de acceso podremos realizar la búsqueda utilizando el buscador online o con su propia aplicación de escritorio.

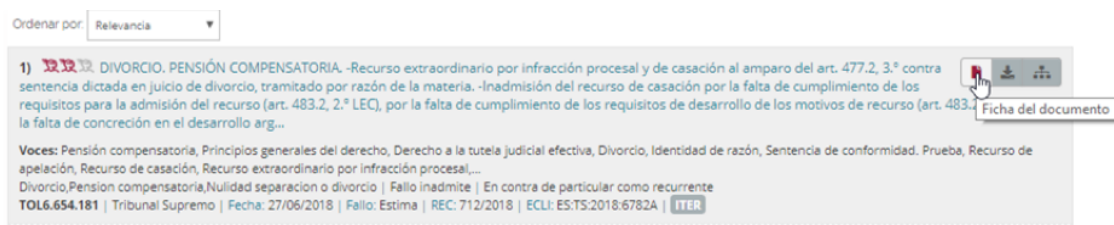
Como podemos observar en la figura 5.4, en este caso la búsqueda de documentos se realiza mediante el uso de "tags" que se añaden mediante un botón en vez de utilizando un desplegable como se utilizan en otras bases de datos analizadas.

<sup>2</sup><https://www.tirantonline.com/tol/>



**Figura 5.4:** Buscador Tirant Online

En este caso, desde el resultado de la búsqueda podemos descargar el documento desde la opción de descarga habilitada en la parte superior derecha del documento (ver figura 5.5).



**Figura 5.5:** Resultado de la búsqueda - Tirant

## Cendoj

El Centro de Documentación Judicial (Cendoj)<sup>3</sup> es el órgano técnico del Consejo General del Poder Judicial que se encarga de la publicación oficial de la jurisprudencia, así como de las demás competencias en el ámbito de la documentación y de los servicios de gestión del conocimiento.

A diferencia de las anteriores no es necesaria la autenticación del usuario para realizar las búsquedas y nos proporciona una gran cantidad de opciones de búsqueda (ver figura 5.6).

<sup>3</sup><http://www.poderjudicial.es/search/indexAN.jsp>

The screenshot shows the CENDOJ search interface. On the left, there are several filter sections: 'Jurisdicción' (Todas), 'Tipo res.' (Todas), 'Tipo de órgano' (Todas), 'Sección', and 'Localización' (Todas). Below these are fields for 'Nº Resolución', 'Nº ROJ', 'Fecha resolución' (Desde/Hasta), 'Nº Recurso', 'ECLI' (ECLI:ES:TS:2014:3877), and 'Idioma' (Todos). There is also a 'Búsqueda por texto libre' field with options 'Y', 'O', 'No', and 'Próximo', and a 'Ponente' field. At the bottom left, there are options for 'Resultados' (10) and 'Ordenar por' (Fecha Resol. [9..0]), along with checkboxes for 'Historio del Tribunal Supremo', 'Resoluciones de interés casacional', 'Resoluciones de actualidad', and 'Resoluciones de interés jurídico (estudios TS)'. On the right, there are sections for 'Resoluciones más consultadas' (listing STS 2761/2019, STS 600/2020, and ATS 2446/2020), 'Sentencias por tema' (listing various legal topics like 'Real Decreto alarma sanitaria Covid-19', 'derecho mercantil', etc.), and 'Últimas sentencias del Tribunal Supremo' (listing STS 812/2020, STS 808/2020, STS 778/2020, and STS 776/2020).

Figura 5.6: Buscador Cendoj

El resultado de la búsqueda de documentos no nos proporciona ninguna forma de descargarlos. A su vez, el visualizador de documentos que nos proporciona es un visualizador pdf embebido en la propia web. El cual no nos da la opción de descargar el documento (ver figuras 5.7 y 5.8)

The screenshot shows the search results page in CENDOJ. At the top, it says 'Buscador > Resultados' and '183714 resultados (0 máximo documentos a recuperar 200)'. There are navigation options for 'Ver' (10), 'Fecha Resol.[9..0]', 'Página' (1 de 20). The results are listed in a table-like format with the following entries:

- ROJ: STSJ AR 224/2020 - ECLI:ES:TSJAR:2020:224**  
Nº de Resolución: 151/2020 - Tipo Órgano: Tribunal Superior de Justicia, Sala de lo Contencioso - Municipio: Zaragoza - Ponente: JUAN JOSE CARBONERO REDONDO - Nº Recurso: 112/2020 - Fecha: 30/04/2020 - Tipo Resolución: Sentencia
- ROJ: SJSO 31/2020 - ECLI:ES:JSO:2020:31**  
Nº de Resolución: 123/2020 - Tipo Órgano: Juzgado de lo Social - Municipio: Zaragoza - Ponente: ITZIAR MARIA OCHOA CABELLO - Nº Recurso: 521/2018 - Fecha: 27/04/2020 - Tipo Resolución: Sentencia  
RESUMEN: La magistrada del juzgado de Social nº 2 de Zaragoza, estima en su sentencia la demanda interpuesta por la Tesorería General del la Seguridad Social contra Roofoods Spain SL (Deliveroo) reconociendo así que existía una relación laboral de los riders con la empresa Roofoods Spain SL (Deliveroo). Apoya sus razonamientos en la amplia jurisprudencia existente y manifiesta su concordancia con los criterios marcados en las sentencias dictadas por el Juzgado de lo Social nº 19 de Madrid y el Tribunal Superior de Justicia de Madrid referidas a los riders empleados por Deliveroo.
- ROJ: AJSO 54/2020 - ECLI:ES:JSO:2020:54A**  
Nº de Resolución: 20/2020 - Tipo Órgano: Juzgado de lo Social - Municipio: Huesca - Ponente: EDUARDO JOSE BERNUES MATEOS - Nº Recurso: 215/2020 - Fecha: 21/04/2020 - Tipo Resolución: Auto
- ROJ: STSJ AR 276/2020 - ECLI:ES:TSJAR:2020:276**  
Nº de Resolución: 122/2020 - Tipo Órgano: Tribunal Superior de Justicia, Sala de lo Contencioso - Municipio: Zaragoza - Ponente: JUAN CARLOS ZAPATA HIJAR - Nº Recurso: 210/2018 - Fecha: 25/03/2020 - Tipo Resolución: Sentencia

Figura 5.7: Resultado de la búsqueda - Cendoj

The screenshot displays the CENDOJ (Centro de Documentación Judicial) interface. At the top left, the logo 'CENDOJ' is visible. The top right corner lists various judicial bodies: TRIBUNAL SUPREMO, AUDIENCIA NACIONAL, TRIBUNAL SUPERIOR DE JUSTICIA, AUDIENCIA PROVINCIAL, and TRIBUNALES MILITARES Y UNIPERSONALES. The main content area shows the document details for 'Roj: STSJ AR 224/2020 - ECLI: ES:TSJAR:2020:224'. Below this, the following information is provided: Id Cendoj: 50297330012020100022, Órgano: Tribunal Superior de Justicia. Sala de lo Contencioso, Sede: Zaragoza, Sección: 1, Fecha: 30/04/2020, N° de Recurso: 112/2020, N° de Resolución: 151/2020, Procedimiento: Derecho de reunión, Ponente: JUAN JOSE CARBONERO REDONDO, and Tipo de Resolución: Sentencia. At the bottom, the text reads 'SECCION Nº 1 DE LA SALA DE LO CONTENCIOSO-ADMINISTRATIVO DEL TRIBUNAL SUPERIOR DE JUSTICIA DE ARAGON' and 'C/ Coso, 1, Zaragoza Zaragoza'. A large 'FALCI' watermark is visible across the bottom right of the document details.

Figura 5.8: Visualizador de documentos - Cendoj

## 5.2. Requisitos del módulo gestor de documentos

El módulo gestor de documentos a desarrollar en este proyecto debería realizar una búsqueda generalizada en las diferentes bases de datos descritas anteriormente en la sección 5.1, y recuperar todos los resultados de las búsquedas. El usuario debería tener la opción de seleccionar los documentos deseados y el módulo tendría que encargarse de descargar dichos documentos en formato pdf.

Además el módulo tendría que ser generalizable para otras bases de datos por lo que el usuario tendría la opción de elegir, en principio, en qué bases de datos desearía buscar los documentos.

Analizando las interfaces de base de datos anteriores se observan cuatro problemas principales: diferencias y dependencia de varios sistemas de autenticación, falta de una API disponible que facilite la realización de búsquedas en sus propias bases de datos, la carga



dinámica de diferentes campos en algunos formularios y la variabilidad (en algunos casos ausencia) de opciones para la descarga de documentos.

En las bases de datos de Aranzadi y Tirant es necesario el acceso mediante la plataforma de la UPV/EHU con lo que sería necesario implementar un sistema de autenticación en la plataforma GAUR. Después de obtener las credenciales para la base de datos, sería necesario implementar un sistema de inicio de sesión específico para cada base de datos.

En el caso de la base de datos Tirant, el acceso tanto a la página web como a la aplicación tiene que hacerse desde la red de la UPV/EHU o adquiriendo unas credenciales desde la tienda de Tirant Online. Esto limitaría el uso de la aplicación a los equipos conectados a la red de la UPV/EHU o a los profesionales que hayan adquirido una cuenta de pago en Tirant Online.

Podemos observar que ninguna de las bases de datos descritas anteriormente proporcionan una API con la que poder realizar las búsquedas de los documentos. Por ese motivo, para implementar un módulo que busque en las diferentes bases de datos sería necesario usar una técnica llamada Web Scraping<sup>4</sup> la cual permite recoger el documento html de la página y procesar los campos simulando una búsqueda manual por un usuario real. El problema de utilizar web scraping para realizar las búsquedas es que ello nos obliga a realizar un análisis individual para cada web y la implementación personalizada para dicha web. En caso de que la plataforma decida modificar algún aspecto del buscador o de protocolo de búsqueda, podría ser que la funcionalidad realizada deje de funcionar.

En el caso del formulario de Tirant Online la selección de muchos campos se hacen de forma dinámica utilizando botones lo que nos impide obtener las opciones disponibles para ese campo en concreto al cargar la página. Por ejemplo, en el formulario de Tirant Online (ver figura 5.4), el campo de jurisdicción no dispone de un desplegable con todas las opciones disponibles. En su defecto, dispone de un icono que al seleccionarlo carga una ventana emergente con las opciones disponibles y se añaden dinámicamente en el formulario.

Además en los diferentes formularios los elementos disponibles para realizar la consulta varían dependiendo de los valores elegidos en diferentes campos. Generando el problema de identificar las incompatibilidades de las diferentes opciones en los campos del formulario. Debido a que no disponemos del sistema de control de la web ya que debemos realizar la petición con web scraping, este problema hace que el modulo sea difícilmente escalable y generalizable.

---

<sup>4</sup>Web Scraping: [https://en.wikipedia.org/wiki/Web\\_scraping](https://en.wikipedia.org/wiki/Web_scraping)

La finalidad del módulo es que sea generalizado para el uso en otras variantes de la SPL WacLine con lo que el uso de web scraping hace que eso sea inviable.

Por último, como se ha comentado antes, cada una de las bases de datos tienen su forma particular de implementar la descarga de los documentos, incluso en caso de Cendoj no existe esta opción. Además, este módulo debería estar integrado en una extensión para el navegador Chrome. Esto implica nos limitaría el uso de los documentos descargados al navegador. Esto último hace que la descarga de documentos sea un característica meramente cuantitativa y no cualitativa debido a que al tener que analizar los documentos en el navegador, y al poder analizar documentos online, el usuario no verá necesaria la funcionalidad de descargar los documentos.

Finalmente, debido a los problemas identificados y analizados en este apartado, se ha decidido descartar el desarrollo del *módulo gestor de documentos*. En su defecto, se ha implementado un nuevo módulo llamado: *módulo de sesiones* descrito en la sección [7.2](#).

## 6. CAPÍTULO

---

### Análisis de Requisitos

---

Teniendo en cuenta que este proyecto se desarrolla en el marco de un proyecto ya existente, es necesario analizar dos principales fuentes de requisitos:

- Los requisitos propios del producto a desarrollar, teniendo en cuenta las funcionalidades y objetivos fijados para el mismo.
- Los requisitos derivados de la integración con las funcionalidades ya desarrolladas dentro del proyecto WacLine por Haritz Medina en el grupo de investigación ONEKIN.

A continuación se describirán los requisitos funcionales y no funcionales del proyecto.

#### 6.1. Requisitos funcionales

En primer lugar, teniendo en cuenta que se va a utilizar la SPL WacLine como base para desarrollar nuestro producto final, los módulos encargados de gestionar anotaciones, servidores de anotaciones y los libros de códigos serán los implementados por WacLine. No obstante, muchos de estos módulos se adaptarán o modificarán para que sean compatibles con las nuevas funcionalidades.

Dicho esto, al iniciarse la aplicación, ésta le dará la opción de generar un *Codebook* con los códigos de anotación por defecto si es la primera vez que se abre la aplicación. A

continuación, el usuario deberá crear una *Sesión* que describa la actividad que va a realizar. Una vez iniciada, la aplicación ofrecerá la posibilidad de anotar y guardar dichas anotaciones.

La aplicación desarrollada en este TFG ofrecerá las siguientes funcionalidades principales:

- **Sesiones.** El usuario podrá crear sesiones las cuales diferenciarán las anotaciones realizadas para que no interfieran con las anotaciones recopiladas de otras sesiones. Las sesiones ofrecerán además la funcionalidad de registrar los documentos anotados. El usuario controlará en todo momento los documentos que engloban la sesión actual pudiendo navegar entre documentos directamente desde la interfaz de usuario. Por último, la aplicación ofrecerá la opción de transferir todos los datos recopilados de la sesión actual a otra. Esta opción estará diseñada para el caso de que el usuario quiera guardar el progreso de la sesión a la hora de eliminarla.
- **Enlaces ente anotaciones.** El usuario podrá enlazar una anotación con otra realizada anteriormente independientemente de si está en el mismo documento o en otro documento de la sesión. Además, el usuario tendrá la opción de visualizar las anotaciones enlazadas de una anotación en concreto, pudiendo navegar entre anotaciones fácilmente.
- **Exportar.** La aplicación tendrá la opción de exportar todas las anotaciones realizadas en un documento Word utilizando una plantilla definida específicamente para este proyecto (Ver anexo [A](#)). Además, este documento generado, incluirá información específica de la sesión como los enlaces de las anotaciones, los comentarios de las anotaciones y demás información de interés.

## 6.2. Requisitos no funcionales

Uno de los objetivos de este TFG es que las funcionalidades implementadas específicamente para este proyecto, sean integradas en la SPL WacLine. Para ello, los módulos desarrollados en este TFG deberán incluir los puntos de variación correspondientes (ver capítulo 7), para la correcta integración en la SPL. Esto permitirá añadir las partes del código fuente siempre que el módulo esté seleccionado en la variante generada desde la SPL.

Además los módulos deberán ser generalizables para poder usarse en todos los ámbitos relacionados con las anotaciones web.

Y por último, tanto los cambios realizados en los módulos ya implementados en WacLine, como los nuevos módulos tienen que ser compatibles entre ellos. Por lo que en la implementación de los nuevos módulos, será necesario controlar la interacción y dependencia con los módulos ya existentes para no generar conflictos.

## 6.3. Casos de uso

A continuación, en la figura 6.1, se presenta el modelo de casos de uso donde se pueden diferenciar dos tipos de actores, el usuario de la aplicación y el sistema.

Se han utilizado dos colores para diferenciar la distribución de los casos de uso. En **rojo** se definen los casos de uso ya implementados en la versión base de WacLine. Mientras que en **negro**, se definen los casos de uso que se han desarrollado en este TFG. Estas funcionalidades están asociadas a los requisitos funcionales descritos en la sección 6.1.

A continuación se desarrollarán los casos de uso haciendo una breve descripción de los mismos así como explicando los flujos de eventos asociados a cada uno.

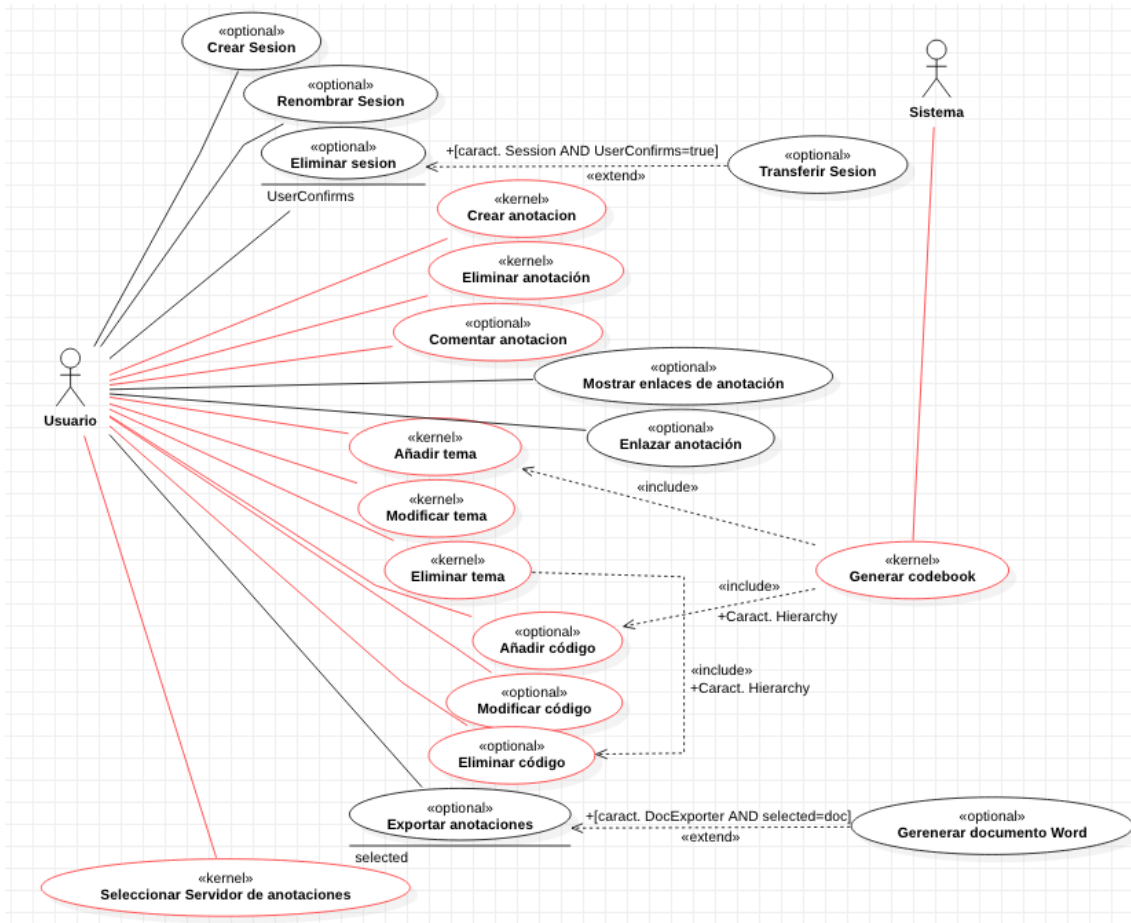


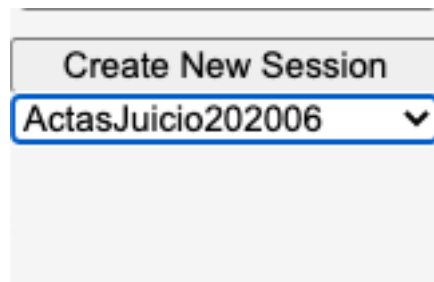
Figura 6.1: Modelo de Casos de Uso

### 6.3.1. Crear Sesión

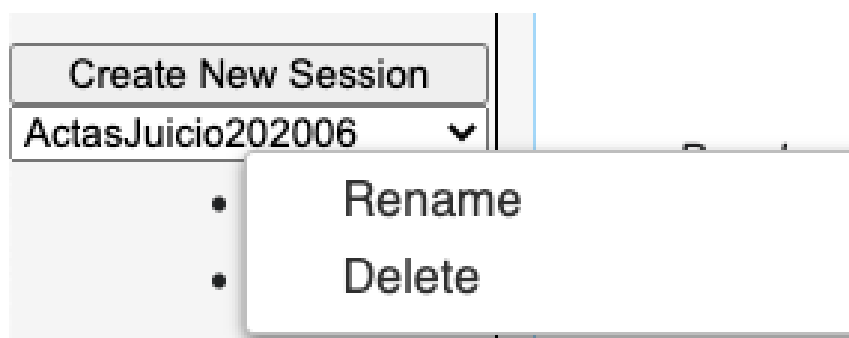
El usuario puede crear sesiones para realizar actividades específicas diferenciando así, las acciones realizadas con las diferentes sesiones.

*Flujo de eventos:*

1. El usuario selecciona la opción de *Create new session* (Ver figura 6.2).
2. El sistema muestra un formulario para que el usuario indique el nombre que tendrá la nueva sesión (Ver figura 6.3).



**Figura 6.2:** Opción *Create new Session*



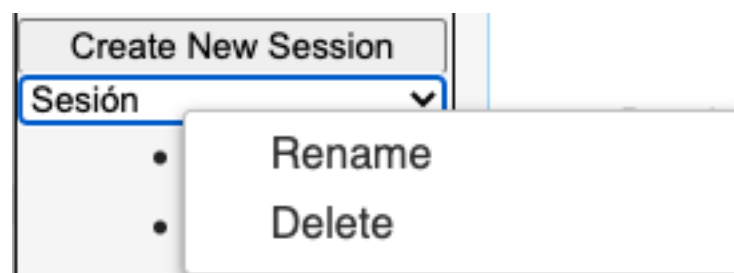
**Figura 6.3:** Formulario de creación de sesiones

### 6.3.2. Renombrar Sesión

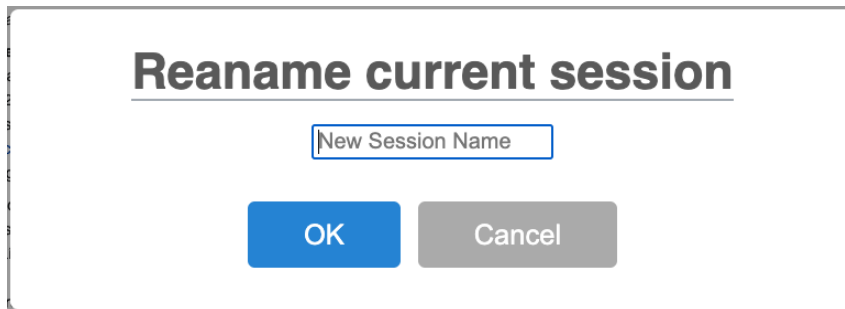
El usuario tiene la opción de renombrar la sesión actual.

*Flujo de eventos:*

1. El usuario selecciona la opción *Rename* desde el menú contextual de la sesión (Ver figura 6.4).
2. El usuario inserta en nuevo nombre de la sesión en el formulario (Ver figura 6.5).



**Figura 6.4:** Menú contextual de las sesiones



Reaname current session

New Session Name

OK Cancel

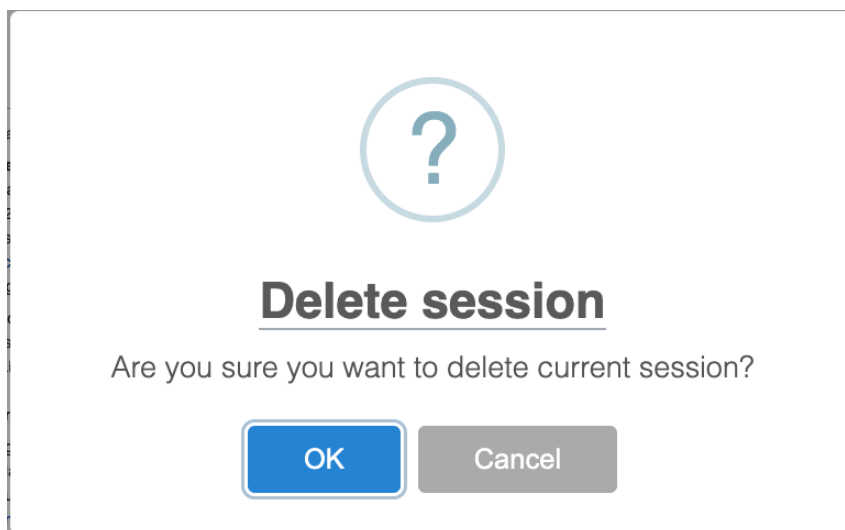
**Figura 6.5:** Formulario para renombrar la sesión actual

### 6.3.3. Eliminar sesión

El usuario puede eliminar la sesión actual. Al hacerlo el sistema le dará la opción de transferir la información recopilada en la sesión a otra.

*Flujo de eventos:*

1. El usuario selecciona la opción *Delete* en el menú contextual de la sesión (Ver figura 6.4).
2. El sistema le pide que confirme la acción por seguridad (Ver figura 6.6).
3. Si el usuario acepta se ejecuta el caso de uso 6.3.4.
4. Si el usuario cancela, no se elimina la sesión.



?

**Delete session**

Are you sure you want to delete current session?

OK Cancel

**Figura 6.6:** Formulario de confirmación para eliminar la sesión

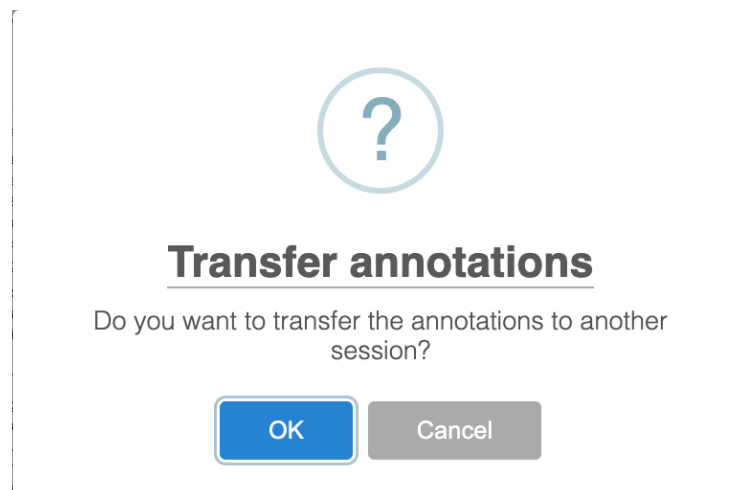


### 6.3.4. Transferir sesión

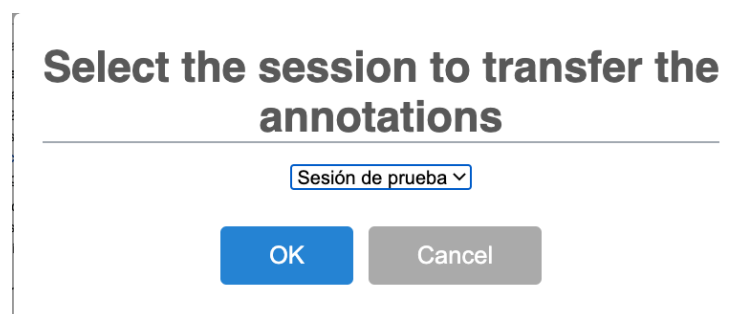
Cuando el usuario confirme que quiere eliminar la sesión, podrá transferir los datos de la sesión actual a otra sesión creada.

*Flujo de eventos:*

1. El sistema le pregunta al usuario si desea transferir la información de la sesión a otra (ver figura 6.7).
2. En caso de que el usuario acepte, el sistema le mostrará las sesiones disponibles (Ver figura 6.8).
3. Al elegir una sesión, el sistema transferirá toda la información de una sesión a otra.



**Figura 6.7:** Formulario de confirmación para transferir la sesión



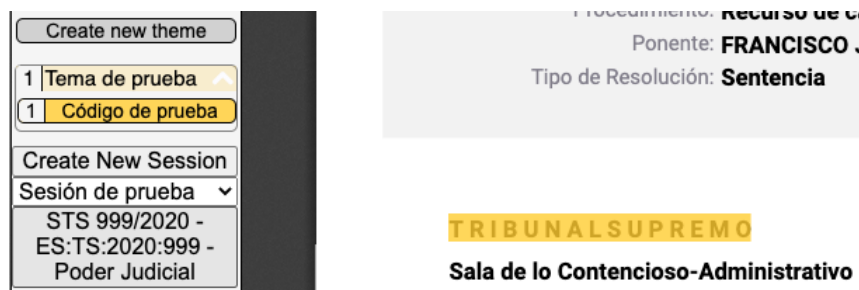
**Figura 6.8:** Formulario para seleccionar a qué sesión transferir la información

### 6.3.5. Crear anotación

El usuario podrá crear anotaciones sobre el documento que esté analizando.

*Flujo de eventos:*

1. El usuario selecciona el texto que quiere anotar.
2. El usuario selecciona el código o tema con el que quiere anotar el texto seleccionado (Ver figura 6.9).



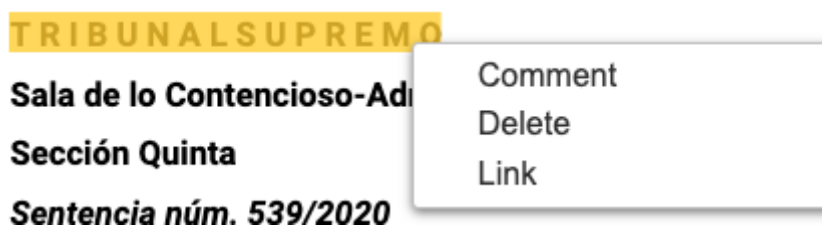
**Figura 6.9:** Proceso de creación de una anotación

### 6.3.6. Eliminar anotación

El usuario podrá eliminar una anotación realizada.

*Flujo de eventos:*

1. El usuario abre el menú contextual de la anotación (Ver figura 6.10).
2. El usuario selecciona la opción *Delete*.



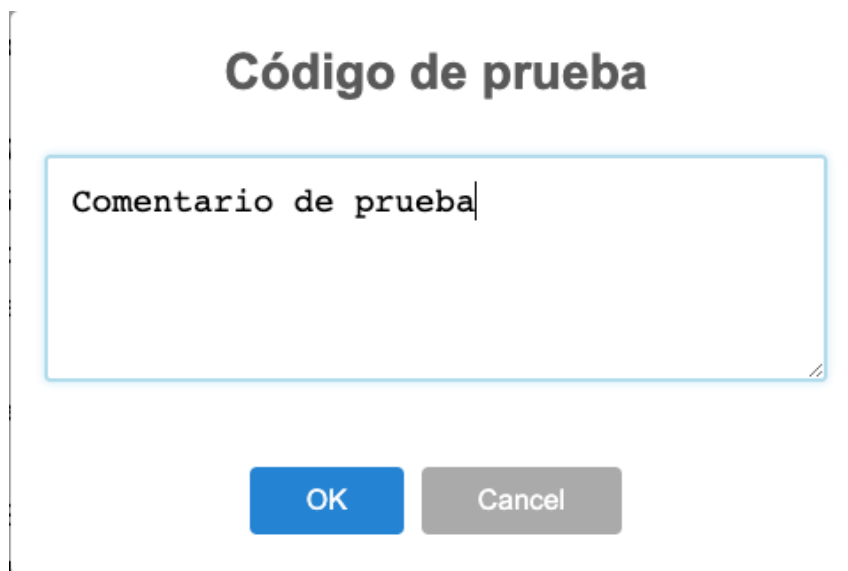
**Figura 6.10:** Menú contextual de las anotaciones

### 6.3.7. Comentar anotación

El usuario podrá realizar comentarios en las anotaciones realizadas.

*Flujo de eventos:*

1. El usuario seleccionará la opción *Comment* en el menú contextual de la anotación (Ver figura 6.10).
2. Escribirá el comentario que quiere realizar en el formulario (Ver figura 6.11).

El formulario tiene un título "Código de prueba" en la parte superior. Debajo del título hay un campo de texto con el texto "Comentario de prueba" y un cursor al final. En la parte inferior del formulario hay dos botones: "OK" (azul) y "Cancel" (gris).

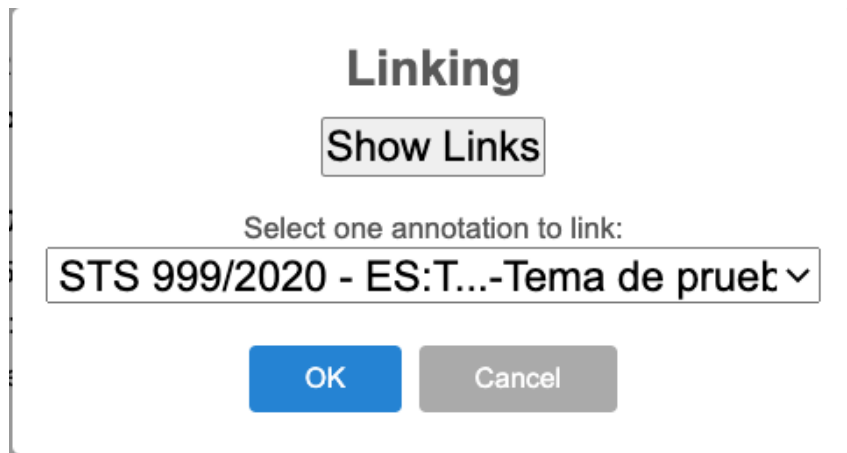
**Figura 6.11:** Formulario de comentarios

### 6.3.8. Enlazar anotación

El usuario podrá enlazar la anotación con las anotaciones ya realizadas en la sesión, independientemente del documento en el que se encuentren.

*Flujo de eventos:*

1. El usuario selecciona la opción *Link* en el menú contextual de anotaciones (Ver figura 6.10).
2. El usuario selecciona entre las anotaciones realizadas en la sesión qué anotación desea enlazar y confirma la operación (Ver figura 6.12)



**Figura 6.12:** Menú de enlaces de una anotación

### 6.3.9. Mostrar enlaces de la anotación

El usuario podrá ver las anotaciones a las que está enlazada una anotación en concreto.

*Flujo de eventos:*

1. El usuario elige la opción *Link* en el menú contextual de la anotación (Ver figura 6.10).
2. El usuario selecciona la opción *Show Links* del menú de enlaces (Ver figura 6.12) .
3. El usuario podrá visualizar las anotaciones enlazadas agrupadas por documentos. Además podrá navegar hasta la anotación seleccionada (Ver figura 6.13).



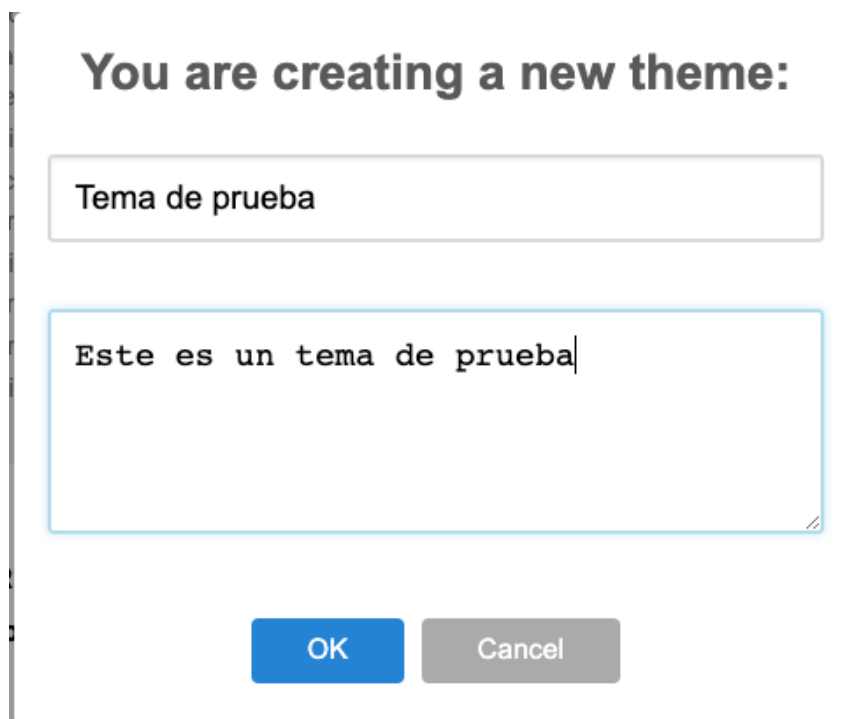
**Figura 6.13:** Visualización de anotaciones enlazadas

### 6.3.10. Añadir tema

El usuario podrá crear temas para anotar y agrupar códigos de anotación.

*Flujo de eventos:*

- El usuario selecciona la opción *Create new theme* en la interfaz de usuario.
- El sistema muestra un formulario para que el usuario cree un título y una descripción para el nuevo tema (Ver figura 6.14).



The image shows a dialog box titled "You are creating a new theme:". It features a text input field containing "Tema de prueba" and a larger text area containing "Este es un tema de prueba". At the bottom, there are two buttons: "OK" (highlighted in blue) and "Cancel" (greyed out).

**Figura 6.14:** Formulario de creación de temas

### 6.3.11. Modificar tema

El usuario podrá modificar la información relacionada con un tema.

*Flujo de eventos:*

1. El usuario selecciona la opción *Modify theme* en el menú contextual del tema (Ver figura 6.15).
2. El usuario modifica la información del tema y confirma los cambios (Ver figura 6.16).

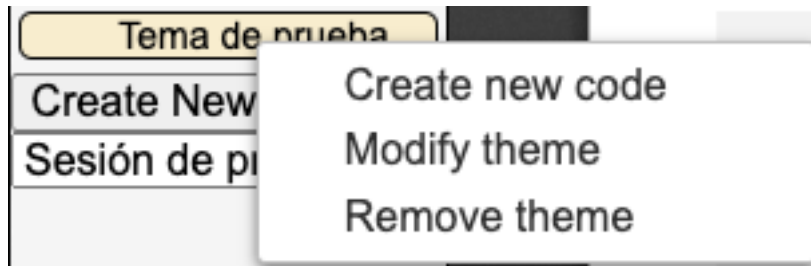


Figura 6.15: Menú contextual del tema

A screenshot of a form for updating a theme. The title is 'You are updating the theme Tema de prueba'. Below the title is a text input field containing 'Tema de prueba'. Below that is a larger text area containing the text 'Este es un tema de prueba. Prueba de modificación de tema'. At the bottom are two buttons: 'OK' (blue) and 'Cancel' (grey).

Figura 6.16: Formulario de modificación de temas

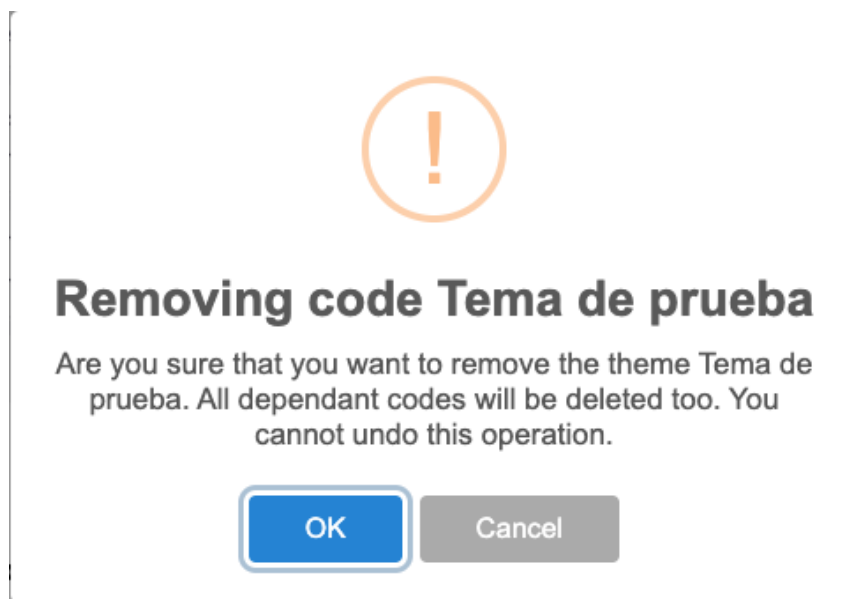
### 6.3.12. Eliminar tema

El usuario tendrá la opción de eliminar un tema. Al eliminar el tema, se eliminarán los códigos asociados a dicho tema.

*Flujo de eventos:*

1. El usuario selecciona la opción *Remove theme* en el menú contextual del tema (Ver figura 6.15).

2. El sistema pide una confirmación de la acción al usuario que debe aceptar para eliminar el tema (Ver figura 6.17).
3. El sistema elimina el tema seleccionado y todos los códigos asociados a ese tema.



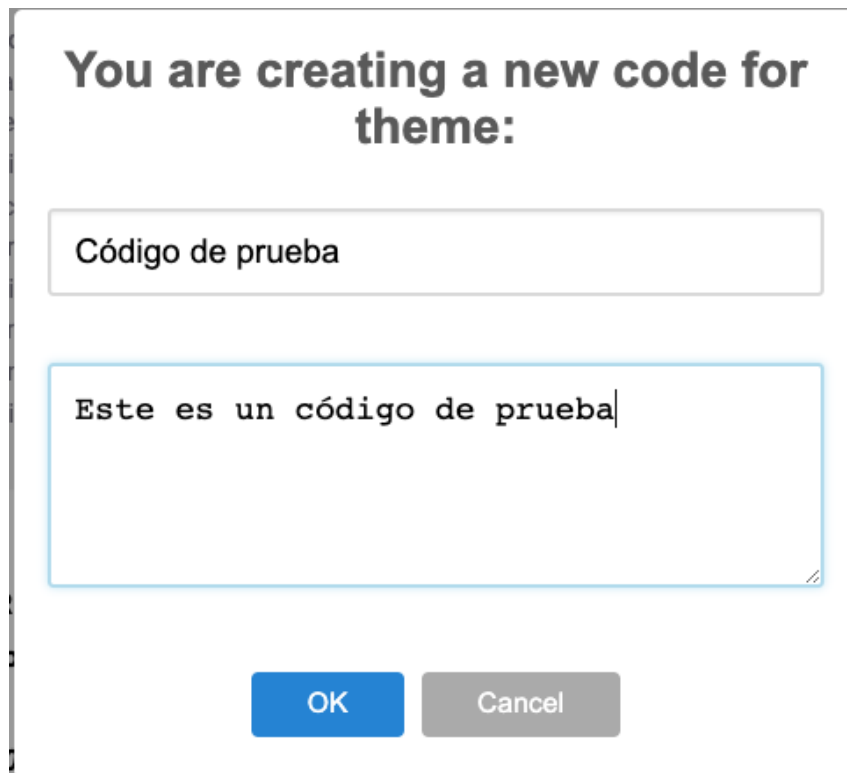
**Figura 6.17:** Confirmación para eliminar un tema

### 6.3.13. Añadir código

El usuario podrá crear códigos que pertenezcan a un tema para utilizarlo como código de anotación.

*Flujo de eventos:*

- El usuario seleccionará la opción *Create new code* en el menú contextual del tema (Ver figura 6.15).
- El sistema mostrará un formulario para que el usuario cree un título y una descripción para el nuevo código (Ver figura 6.18).



The image shows a dialog box titled "You are creating a new code for theme:". It contains two input fields. The first is a text box with the placeholder text "Código de prueba". The second is a larger text area containing the text "Este es un código de prueba" with a cursor at the end. At the bottom, there are two buttons: "OK" (blue) and "Cancel" (grey).

**Figura 6.18:** Formulario de creación de códigos

#### 6.3.14. Modificar código

El usuario podrá modificar la información relacionada con un código.

*Flujo de eventos:*

1. El usuario selecciona la opción *Modify code* en el menú contextual del código (Ver figura 6.19).
2. El usuario modifica la información del código y confirma los cambios (Ver figura 6.20).



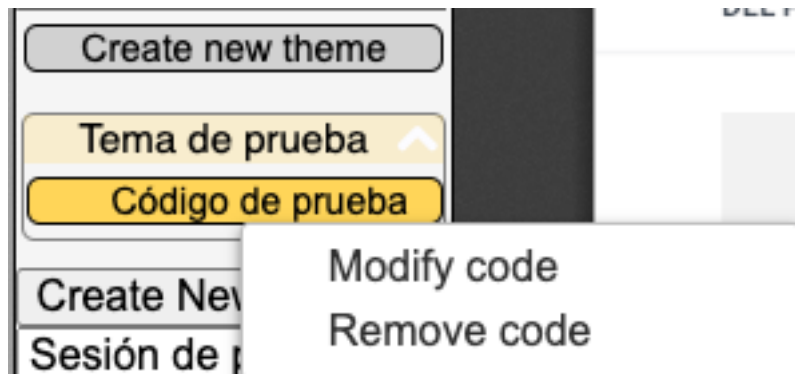


Figura 6.19: Menú contextual de los códigos

A screenshot of a mobile application dialog box. The title is 'You are updating the code Código de prueba pertaining to theme Tema de prueba'. Below the title is a text input field containing 'Código de prueba'. Below the input field is a text area containing the text 'Este es un código de prueba. Prueba de modificación'. At the bottom of the dialog are two buttons: 'OK' (blue) and 'Cancel' (grey).

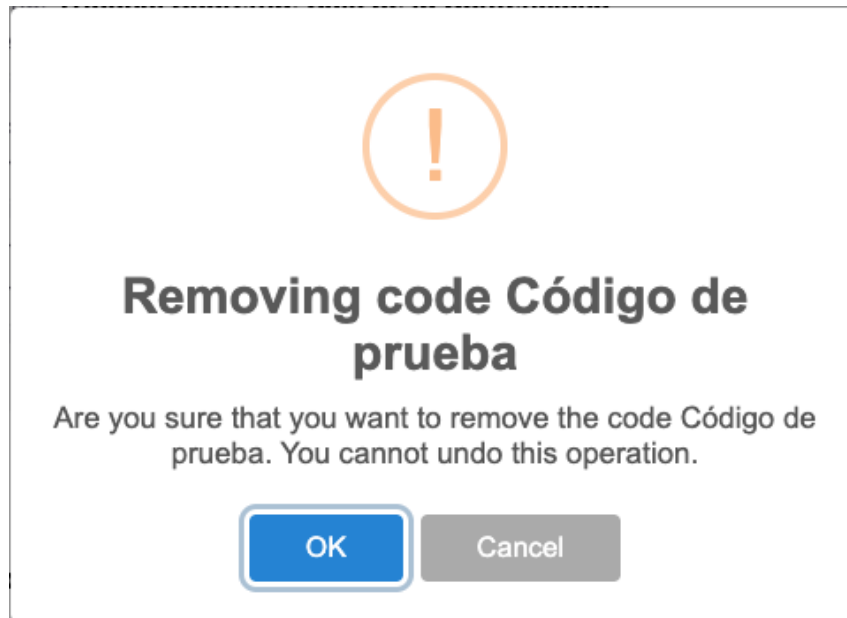
Figura 6.20: Formulario modificación de códigos

### 6.3.15. Eliminar código

El usuario tendrá la opción de eliminar un código. *Flujo de eventos:*

1. El usuario selecciona la opción *Remove code* en el menú contextual del código (Ver figura 6.19).

2. El sistema pide una confirmación de la acción, que debe aceptar para eliminar el código (Ver figura 6.21).



**Figura 6.21:** Confirmación para eliminar un tema

### 6.3.16. Exportar anotaciones

El usuario tendrá la opción de exportar las anotaciones realizadas en la sesión. *Flujo de eventos:*

1. El usuario selecciona la opción de exportar a Word desde el apartado de herramientas, en la interfaz de usuario (Ver figura 6.22).
2. El sistema recupera todas las anotaciones y genera un documento Word con toda la información.



**Figura 6.22:** Opción de exportar a documento Word

### 6.3.17. Generar documento Word

A la hora de exportar el sistema debe generar el documento Word con las anotaciones realizadas.

*Flujo de eventos:*

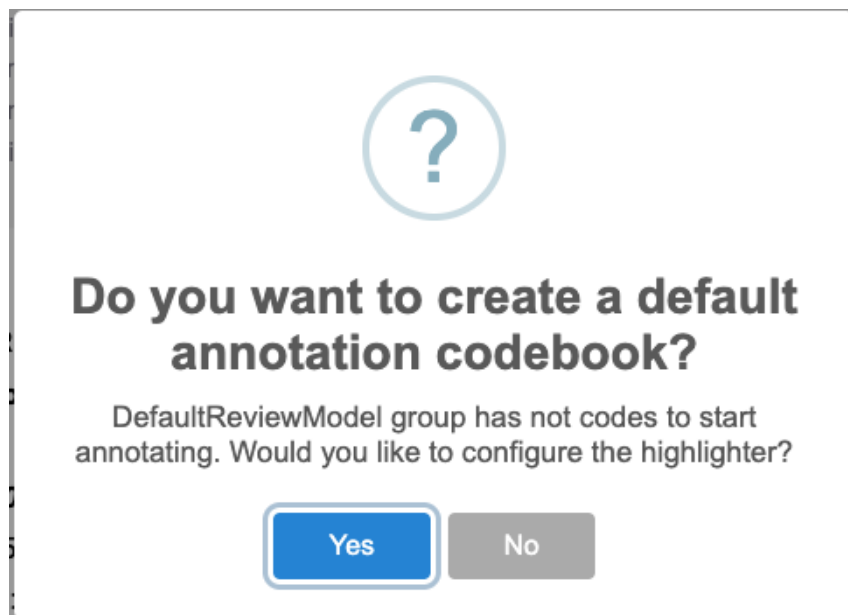
1. El sistema recopila y procesa todas las anotaciones realizadas en la sesión.
2. El sistema carga la plantilla en formato .docx para rellenarla.
3. El sistema genera y guarda el documento final en el sistema de ficheros del usuario.

### 6.3.18. Generar Codebook

Al iniciar por primera vez la aplicación el sistema crea un *codebook* por defecto.

*Flujo de eventos:*

1. El sistema comprueba que no hay ningún *codebook*.
2. El sistema pide confirmación al usuario para crear el *codebook* (Ver figura 6.23).
3. El sistema crea los temas y los códigos por defecto definidos por el sistema.



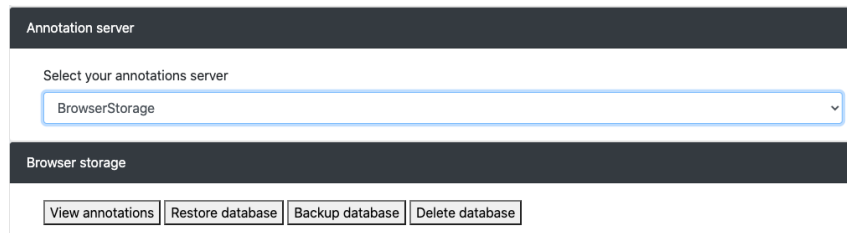
**Figura 6.23:** Confirmación para generar el *codebook* por defecto

### 6.3.19. Seleccionar Servidor de anotaciones

El usuario podrá elegir si desea usar Hypothesis como servidor de anotaciones o un servidor local.

*Flujo de eventos:*

1. El usuario elige el servidor que desea utilizar en el apartado *Annotation Server* en la configuración de la aplicación (Ver figura 6.24).



The screenshot displays a configuration panel for the 'Annotation server'. At the top, the title 'Annotation server' is shown in a dark header. Below this, the instruction 'Select your annotations server' is followed by a dropdown menu currently set to 'BrowserStorage'. Underneath the dropdown is another dark header labeled 'Browser storage'. At the bottom of the panel, there are four buttons: 'View annotations', 'Restore database', 'Backup database', and 'Delete database'.

**Figura 6.24:** Selección del servidor de anotaciones

## 7. CAPÍTULO

---

### Diseño e implementación de la nueva arquitectura de WacLine

---

En este capítulo se explican las modificaciones implementadas en la arquitectura de WacLine como resultado de este TFG. Se presenta el nuevo diagrama de características y una tabla en la que se pueden ver las relaciones entre las nuevas características y casos de uso. Además, en este capítulo se muestra la implementación de las nuevas características.

#### 7.1. Nuevo diagrama de características

Como se ve en la figura 7.1 se han añadido tres nuevas características, resaltadas en color rojo, *Session*, *Linking* y *DocExporter*. Estas características aportan las nuevas funcionalidades definidas anteriormente (ver capítulo 6.1) y engloban los nuevos casos de uso definidos en la sección 6.3.

La característica *Session* está asignada a la raíz de la SPL debido a que se trata de una característica nueva que añade nuevas funcionalidades a la plataforma de la línea de producto, permitiendo diferenciar las anotaciones entre actividades.

La característica *Linking* que proporciona una nueva funcionalidad, en este caso asociado a la característica de *Purpose*, engloba todas las acciones que se pueden realizar con las anotaciones. Esta nueva funcionalidad permite enlazar diferentes anotaciones

Y por ultimo, la característica *DocExporter*, dota a la SPL WacLine de exportar las anotaciones recopiladas en un documento Word en formato docx.

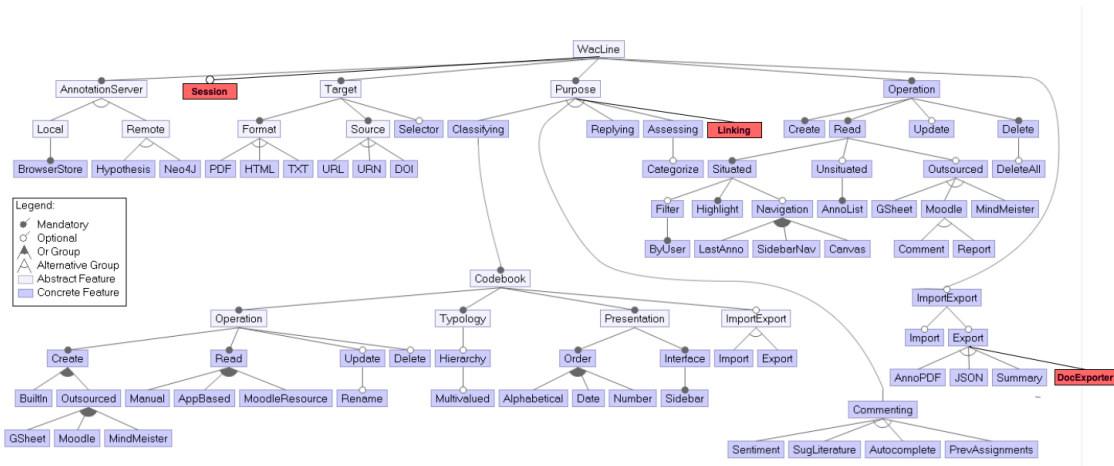


Figura 7.1: Nuevo diagrama de características de WacLine

En la tabla 7.1 podemos ver la relación entre las nuevas características añadidas en WacLine y los nuevos casos de uso detallados en la sección 6.3.

A continuación se explicará la implementación de cada característica explicando su objetivo y cómo se ha conseguido implementar. Además se incluirá una tabla donde se especificará las funcionalidades implementadas que incluirán dos apartados. El primer apartado mostrará las clases, ficheros y líneas de código nuevas que se han creado, mientras que en el segundo apartado aparecerán qué clases y ficheros han sido modificados. Además se incluirá cuántas líneas incluye el punto de variación y en qué línea comienza.

Característica	Casos de uso asociados
<b>Linking</b>	<b>Mostrar Enlaces de Anotación</b> <b>Enlazar Anotación</b>
<b>Session</b>	<b>Crear Sesión</b> <b>Eliminar Sesión</b> <b>Renombrar Sesión</b> <b>Transferir Sesión</b>
<b>DocExporter</b>	<b>Exportar Anotaciones</b> <b>Generar Documento Word</b>

Tabla 7.1: Tabla de relaciones entre las nuevas características y casos de uso

## 7.2. Característica *Session*

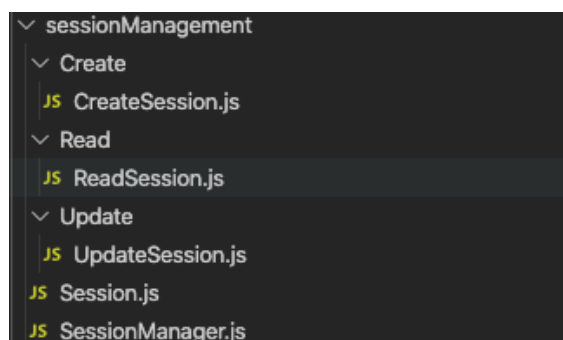
La característica *Session* otorga a WacLine la posibilidad de diferenciar todas las anotaciones que se realizan entre actividades. El usuario puede crear sesiones en las cuales centrarse en realizar un determinado análisis, por ejemplo al análisis de un caso judicial, sin que interfieran las anotaciones realizadas en el análisis de otros casos. Además permitirá al usuario, como medida de seguridad, transferir todas las anotaciones a otra sesión. En la tabla 7.2 se muestra un resumen de las clases implementadas como se ha explicado al final del apartado anterior.

Función	Clase	Fichero	Número de líneas de código	PV introducidos
Gestión de sesiones	Session Manager	SessionManager.js	42	-
	Content Script Manger	ContentScriptManager.js	453	5 (83) 3 (319) 15 (339)
Gestión de anotaciones realizadas	Session Reader	ReadSession.js	113	-
	Annotation Creator	CreateAnnotation.js	201	4 (104)
Gestión de documentos anotados	Session Reader	ReadSession.js	242	-
	Annotation Creator	CreateAnnotation.js	201	11 (67)
Renderizar la interfaz de usuario	Session Reader	ReadSession.js	113	-
	-	-	-	-

**Tabla 7.2:** Tabla de funcionalidades implementadas en la característica *Session*

Las de la característica *Session* son la diferenciación de anotaciones utilizando etiquetas propias de las sesiones, la visualización de los documentos anotados en la interfaz, el proceso de conversión de las sesiones para su correcto almacenamiento en el servidor y la transferencia de anotaciones entre sesiones.

La estructura de los archivos de la característica *Session* (ver figura 7.2) se asemeja a la estructura del código. Las sesiones se controlan a través del *Session Manager* el cual es el encargado de gestionar las operaciones de *Create*, *Update* y *Read* que están implementadas en los respectivos archivos.



**Figura 7.2:** Sistema de ficheros de la característica *Session*

A continuación, en las siguientes subsecciones, se explican los detalles de la implementación junto con ejemplos del código.

### Definición de la estructura de una sesión

Las sesiones están compuestas por cinco campos. Estos campos son los encargados de guardar el identificador de la sesión (*id*), el nombre que el usuario ha definido para esa sesión (*sessionName*), los documentos anotados en la sesión (*sessionURIs*), el grupo al que pertenece el usuario (*group*) y la última fecha en la que se ha modificado la sesión (*updated*).

### Session Manager

Como hemos comentado al principio de esta sección, las sesiones son controladas por el *Session Manager* el cual se instancia al iniciarse la aplicación, en el *Content Script*. Para ello debe añadirse un punto de variación en dicho archivo para así que se instancie únicamente cuando se haya seleccionado la característica de *Session* (ver figura 7.1).

---

#### Listing 7.1 Instancia del *Session Manager* en el *Content Script* (ContentScriptManager.js)

```
1 // PVSCL:IFCOND(Session, LINE)
2   .then(() => {
3     return this.loadSessionManager()
4   })
5 // PVSCL:ENDCOND
6
7 *****
8 *****
9
10 // PVSCL:IFCOND(Session, LINE)
11
12 loadSessionManager () {
13   return new Promise((resolve, reject) => {
14     window.abwa.sessionManagement = new SessionManager()
15     window.abwa.sessionManagement.init(() => {
16       resolve()
17     })
18   })
19 }
20
21 destroySessionManager () {
22   if (window.abwa.sessionManagement) { window.abwa.sessionManagement.destroy() }
23 }
```



```
24 // PVSCL:ENDCOND
```

El *Session Manager* es el encargado de controlar los submódulos encargados de la manipulación de las sesiones (ver figura C.1): el controlador *SessionReader*, encargado del redimensionado de la interfaz de usuario y el control de las sesiones existentes, el controlador *SessionCreator* el cual se encarga de la operación de crear nuevas sesiones y el controlador *sessionUpdater* que se encarga de las operaciones de actualización y borrado de sesiones.

### Control y filtrado de anotaciones en una sesión

Como se ha indicado al principio, esta característica incluye la funcionalidad de distinguir las anotaciones que se realizan en la sesión. Para ello se modifica incluyendo un punto de variación el archivo *CreateAnnotation.js* para que al crear una anotación se asigne en el campo *tags* una etiqueta identificando la sesión en la que se ha realizado (ver código 7.2).

---

#### Listing 7.2 Asociación de la anotación creada con la sesión actual (*CreateAnnotation.js*)

```
1 // PVSCL:IFCOND(Session, LINE)
2     const sessionTag = Config.namespace + ':session:' +
      window.abwa.sessionManagement.sessionReader.currentSession.id
3     tags.push(sessionTag)
4 // PVSCL:ENDCOND
```

En el archivo *ReadAnnotation.js*, el encargado de recuperar las anotaciones desde el servidor y mostrárselas al usuario, se ha añadido un punto de variación donde se filtran las anotaciones que tengan únicamente el *tag* correspondiente a la sesión actual (ver figura 7.3). De esta forma se consigue que el usuario solo trabaje con las anotaciones realizadas en esa sesión.

---

#### Listing 7.3 Filtrado de las anotaciones de la sesión actual (*ReadAnnotation.js*)

```
1     // PVSCL:IFCOND(Session, LINE)
2     this.allAnnotations = _.filter(this.allAnnotations, (a) => {
3         return a.tags.includes(
4             Config.namespace + ':session:' +
      window.abwa.sessionManagement.sessionReader.currentSession.id
5         )
6     })
7 // PVSCL:ENDCOND
```

## Control de documentos anotados

Al crear la anotación se comprueba, en el fichero *CreateAnnotation.js*, si el documento al que pertenece esta anotación está asignado a la sesión actual, si no lo está se añade y se actualiza la sesión (ver figura 7.4).

---

### Listing 7.4 Asignación del documento anotado a la lista de documentos de la sesión (CreateAnnotation.js)

```
1 // PVSCL:IFCOND(Session, LINE)
2     if (window.abwa.sessionManagement.sessionReader.currentSession.sessionURIs.length === 0
3         ||
4         window.abwa.sessionManagement.sessionReader.currentSession.sessionURIs.filter((url)
5     => {
6         return url.source === deserializedAnnotation.target[0].source.url
7         }).length < 1) {
8         window.abwa.sessionManagement.sessionReader.currentSession.sessionURIs.push({
9         source: deserializedAnnotation.target[0].source.url, title:
10        deserializedAnnotation.target[0].source.title
11        })
12        LanguageUtils.dispatchCustomEvent(Events.updateSession, { session:
13        window.abwa.sessionManagement.sessionReader.currentSession })
14    }
15 // PVSCL:ENDCOND
```

Estos documentos se muestran en la interfaz para que el usuario pueda llevar el control en todo momento. Además, si el usuario lo desea, puede abrir el documento en una pestaña nueva realizando click sobre el documento deseado. En caso de querer eliminar el documento del índice, se ha creado un menú contextual con la opción de *Delete* para que el usuario pueda eliminar fácilmente el documento (ver código C.2 del fichero *ReadSession.js*).

## Renderizado de la interfaz de usuario

Como hemos comentado anteriormente, el encargado de gestionar las sesiones es el submódulo *SessionReader*, con lo que al iniciarse se recupera desde el servidor de anotaciones las sesiones del usuario (ver figura C.3). Después recupera el apartado del *sidebar* donde se almacenan las sesiones y renderiza la interfaz de usuario como podemos ver en la figura C.4.

## Almacenamiento de las sesiones

En el caso de WacLine, los servidores de anotaciones únicamente guardan anotaciones, por lo que para almacenar las sesiones hay que realizar un proceso de conversión entre sesiones y anotaciones para poder guardarlas. Para diferenciar las sesiones de las anotaciones normales se almacena un *tag* el cual define que esa anotación es una sesión.

Para realizar la operación inversa, se ha implementado una función que dada una anotación la reconstruye en una sesión. De esta forma se pueden almacenar todas las sesiones del usuario en el servidor de anotaciones fácilmente. En la figura C.5 se pueden observar las funciones encargadas de realizar estas conversiones.

## Operaciones disponibles con las sesiones

Para implementar las acciones disponibles para las sesiones, se ha optado por definir un menú contextual donde aparecerán las opciones y se ejecutará la acción seleccionada (ver figura C.6). La opción de *Rename* desplegará un alerta con un formulario para que el usuario indique el nuevo nombre de la sesión y después se actualiza dicha sesión (ver figura C.7).

En el caso de la opción *Delete* la aplicación pedirá confirmación ya que es una acción irreversible, además en caso de que el usuario confirme, se le da la opción de transferir el contenido de la sesión a otra que ya esté creada. Para ello el usuario selecciona la sesión a la que se van a transferir los datos y se le añaden los documentos que habían sido anotados en la sesión que se va a eliminar. A continuación se modifican todos los *tag* de las anotaciones para que incluyan el de la sesión nueva (ver figuras C.8 y 7.5)

---

### Listing 7.5 Código de transferencia de anotaciones entre sesiones (*ReadAnnotation.js*)

```
1 // PVSC:IFCOND(Session, LINE)
2 transferCurrentSessionAnnotationsHandler () {
3   return (event) => {
4     let newSessionTag = event.detail.newTag
5     let fromSession = event.detail.fromSession
6     this.retrieveAllAnnotationsFromSession(fromSession, (err, annotations) => {
7       if (err) {
8         console.log(err)
9       } else {
10        annotations.forEach((ann) => {
11          ann.tags.push(newSessionTag)
12          LanguageUtils.dispatchCustomEvent(Events.updateAnnotation, { annotation: ann })
13        })
14      }
15    })
16   }
17 }
```

```

14     }
15     })  }}
16 // PVSCL:ENDCOND

```

### 7.3. Característica *Linking*

La característica *Linking* otorga la funcionalidad de enlazar anotaciones con otras anotaciones realizadas, independientemente del documento en el que se hayan realizado. Además da la posibilidad al usuario de visualizar todas las anotaciones enlazadas a una anotación y poder navegar a ella directamente si necesidad de abrir el documento manualmente y localizarla. En la tabla 7.3 se muestra un resumen de las clases implementadas y los ficheros que se han creado y modificado junto las líneas de código y los puntos de variación.

Función	Clase	Fichero	Número de líneas de código	PV introducidos
<b>Menú de enlaces de anotaciones</b>	Linking Form	LinkingForm.js	224	-
	Annotation Reader	ReadAnnotation.js	829	3 (556) 17 (636)
<b>Gestión de enlaces entre anotaciones</b>	LinkingForm	LinkingForm.js	224	-
	-	-	-	-
<b>Visualización y navegación entre enlaces</b>	ShowLinks	LinkingForm.js	224	-
	-	-	-	-
<b>Eliminación de enlaces entre anotaciones</b>	-	-	-	-
	Annotation Reader	ReadAnnotation.js	12	5 (212) 8 (227)

**Tabla 7.3:** Tabla de funcionalidades implementadas en la característica *Linking*

En la implementación de la característica *Linking* se puede destacar la modificación de la clase *Annotation* además de la funcionalidad de visualizar el menú de enlaces, el proceso de enlazado de anotaciones y el apartado de visualización de enlaces con posibilidad de navegar a dichas anotaciones enlazadas.

Su implementación se ha realizado en los ficheros que gestionan dichas anotaciones.

#### Modificación de las anotaciones

En primer lugar, se ha modificado la estructura de las anotaciones para añadir un campo para guardar los identificadores de las anotaciones enlazadas (ver figura 7.6). Además de modificar las funciones correspondientes de la clase *Annotation* para poder guardar dicho nuevo campo.

**Listing 7.6** Modificación en la estructura *Annotation* (*Annotation.js*)

```
1 // PVSCL:IFCOND(Linking,LINE)
2   annotationlinks = [],
3 // PVSCL:ENDCOND
4
5 *****
6 *****
7
8 // PVSCL:IFCOND(Linking,LINE)
9   this.annotationlinks = annotationlinks
10 // PVSCL:ENDCOND
```

## Menú de enlazar anotaciones

Para crear la interfaz se ha modificado el menú contextual de las anotaciones para añadir la opción *Link* la cual desplegará el menú de enlazar anotaciones. Para ello se ha generado un punto de variación en el documento *ReadAnnotation.js* donde se añade la opción anteriormente comentada la cual ejecuta la función *showForm* del *LinkingForm* (ver figura 7.7). Dicha función muestra al usuario la interfaz donde tendrá la opción de visualizar las anotaciones enlazadas con la anotación seleccionada o por el contrario, seleccionar una anotación para enlazarla (ver figura 6.12).

**Listing 7.7** Código opción *Link* de las anotaciones (*ReadAnnotation.js*)

```
1 // PVSCL:IFCOND(Linking,LINE)
2   items.link = { name: 'Link' }
3 // PVSCL:ENDCOND
4   }
5   return {
6     callback: (key, opt) => {
7       *****
8       // PVSCL:IFCOND(Linking,LINE)
9       else if (key === 'link') {
10        LinkingForm.showLinkingForm(annotation, (err, aAnnotation, bAnnotation) => {
11          if (err) {
12            Alerts.errorAlert({ text: 'Unexpected error when linking. Please reload webpage
and try again. Error: ' + err.message })
13          } else {
14            LanguageUtils.dispatchCustomEvent(Events.updateAnnotation, {
15              annotation: aAnnotation
16            })
17            LanguageUtils.dispatchCustomEvent(Events.updateAnnotation, {
18              annotation: bAnnotation
19            })

```

```
21         }           })
22     }
23     /* PVSCL:ENDCOND */
```

### Filtrado y realización del enlace entre anotaciones

Las anotaciones disponibles para enlazarse son las que todavía no se han enlazado. Por ese motivo, se filtran todas las anotaciones entre las ya enlazadas y se le muestra al usuario la lista completa. Cuando el usuario selecciona una anotación se guarda el identificador de la anotación correspondiente, es decir, si se quiere enlazar una anotación **A** con una anotación **B**, se guarda el identificador de **A** en el apartado de enlaces de **B** y viceversa (ver figura C.9). Una vez enlazadas las anotaciones se devuelven y se actualizan ambas anotaciones, como se puede ver en la figura 7.7).

### Funcionalidad *Show Links*

La opción de *Show Links* sin embargo, permite al usuario visualizar las anotaciones enlazadas a la anotación seleccionada. Para ello, se recuperan todas las anotaciones y se filtran las anotaciones enlazadas correspondientes (ver figura C.10). Después se genera un objeto donde se agrupan las anotaciones por documento y se genera una alerta donde el usuario puede visualizar las anotaciones enlazadas (ver figura 6.13). Además estos enlaces tiene la funcionalidad de que cuando el usuario hace click en ellos se abre en una pestaña nueva el documento al que pertenecen y se coloca dicho documento en la posición de la anotación (ver imagen C.11). Esto se consigue debido a que el enlace esta generado de esta forma: <URL>#<paramName>:<ID> donde la **URL** es la dirección del documento donde está la anotación, el **paramName** es un parámetro definido en el archivo *Config* el cual identifica la variante de WacLine y por último el campo **ID** que hace referencia al identificador de la anotación. De esta forma, al abrir el documento, se inicia la aplicación y navega hasta la anotación deseada.

### Eliminación del enlace al eliminar la anotación

Por último, al eliminar una anotación, se elimina el enlace en todas las anotaciones enlazadas. Para ello se elimina el identificador de la anotación eliminada en todas las anotaciones

con las que estaba enlazada (ver imagen 7.8). Una vez eliminado el enlace, se actualizan las anotaciones para que se hagan efectivos los cambios.

#### Listing 7.8 Actualización de las anotaciones al eliminar un enlace (*ReadAnnotation.js*)

```

1 // PVSCL:IFCOND(Linking,LINE)
2   this.allServerAnnotations.forEach((a) => {
3     a.annotationlinks = a.annotationlinks.filter((link) => link !== annotation.id)
4     LanguageUtils.dispatchCustomEvent(Events.updateAnnotation, { annotation: a })
5   })
6
7   this.allAnnotations.forEach((a) => _.remove(a.annotationlinks, (link) => { return link ===
  annotation.id })))
8 // PVSCL:ENDCOND
9
10 // Dispatch annotations updated event
11 LanguageUtils.dispatchCustomEvent(Events.updatedAllAnnotations, { annotations:
  this.allAnnotations })

```

## 7.4. Característica *DocExporter*

La característica *DocExporter* es la encargada de recopilar los datos generados por el usuario (anotaciones, enlaces, comentarios...) y generar un documento Word estructurado. Con esta característica, el usuario podrá visualizar de una forma fácil y ordenada todas las anotaciones realizadas como los enlaces y comentarios de las anotaciones si se han seleccionado dichas características. En la tabla 7.4 se muestra un resumen de las clases implementadas y los ficheros que se han creado y modificado junto las líneas de código y los puntos de variación.

Función	Clase	Fichero	Número de líneas de código	PV introducidos
<b>Renderizado del icono de la funcionalidad</b>	- ToolSet	- Toolset.js	- 287	- 9 (70)
<b>Procesamiento de los datos</b>	- DocExporter	- DocExporter.js	- 114	- -
<b>Generación del documento</b>	- DocExporter	- DocExporter.js	- 114	- -

**Tabla 7.4:** Tabla de funcionalidades implementadas en la característica *DocExporter*

Para implementar esta característica se ha utilizado una librería externa llamada *Docx-Templater* [DocxTemplater, 2019]. Esta librería nos permite introducir etiquetas en un

documento docx para generar bucles, condiciones e indicar dónde queremos que se inserten los datos. Utilizando esta librería se ha diseñado un documento docx el cual usaremos como plantilla para generar los documentos resultantes, incluida en el anexo [A](#).

Como aspecto remarcable en la implementación de esta característica tendríamos el procesamiento de todas las anotaciones para generar un objeto JSON con el que rellenar la plantilla.

### Inclusión de la librería *DocxTemplater* en WacLine

Para utilizar la librería de *DocxTemplater* es necesario incluirla en el sistema de librerías de WacLine. Para ello, como WacLine hace uso de la tecnología *Node.js*, deberemos incluir la librería y sus dependencias en el archivo *package.json*, en el apartado *dependencies*.

### Generación de la interfaz

En el caso de la característica *DocExporter* carece de interfaz de usuario debido a que es un proceso interno, exceptuando la opción para ejecutar la funcionalidad. Este icono de funcionalidad está alojado en el apartado *ToolBox* el cual está diseñado para albergar las herramientas de WacLine. Para incluir la nueva funcionalidad se ha generado un punto de variación nuevo en el archivo *Toolset.js* el cual incluye el icono de la nueva funcionalidad en caso de que esté seleccionada la característica (Ver figura [7.9](#)).

---

#### Listing 7.9 Icono de la funcionalidad de exportar a documento Word (*ToolSet.js*)

```
1 // PVSCL:IFCOND(DocExporter, LINE)
2     const docxExporterImageUrl = chrome.extension.getURL('/images/docexport.png')
3     this.docxExporterImageUrl = toolsetButtonTemplate.content.firstChild.clone().get(0)
4     this.docxExporterImageUrl.src = docxExporterImageUrl
5     this.docxExporterImageUrl.title = 'Export to Word document'
6     this.toolsetBody.appendChild(this.docxExporterImageUrl)
7     this.docxExporterImageUrl.addEventListener('click', () => {
8         this.docxExportButtonHandler()
9     })
10 // PVSCL:ENDCOND
```



### Recopilación y procesamiento de los datos

Para generar el documento resultante debemos generar un objeto JSON con todos los datos deseados para rellenar la plantilla. Para ello se recuperan las anotaciones y se agrupan por documento además de extraer los comentarios y los enlaces para mayor comodidad a la hora de procesar la plantilla. Al final se recopila el nombre de la aplicación, el *codebook*, la fecha en la que se realiza el documento, el nombre de la sesión, los enlaces de las anotaciones (si están las características correspondientes seleccionadas) y las anotaciones procesadas anteriormente (ver figura C.12). Por último se llama a la función *export* pasándole el objeto que hemos creado con toda la información para que genere el documento.

### Generación del documento

Una vez que tenemos nuestro objeto con toda la información necesaria, en la figura C.13 podemos ver el proceso de generación. Primero se carga la plantilla en formato binario para que la librería *DocxTemplater* pueda procesarla. A continuación, se le asigna el objeto con los datos recopilados a la plantilla utilizando la función *SetData(object)* y generamos el documento resultante con la función *render()*. Por último almacenamos el documento resultante en el sistema de ficheros del usuario.



## 8. CAPÍTULO

---

### Descripción de la aplicación DOCAL

---

Como hemos comentado en capítulos anteriores, DOCAL es una extensión para el navegador Chrome generada utilizando la SPL WacLine extendida con las tres nuevas características desarrolladas en este TFG (*Session*, *Linking* y *DocExporter*).

En este capítulo se mostrará la selección de características realizada para la generación de la aplicación DOCAL y los aspectos de usabilidad que incluye dicha aplicación.

#### 8.1. Selección de características

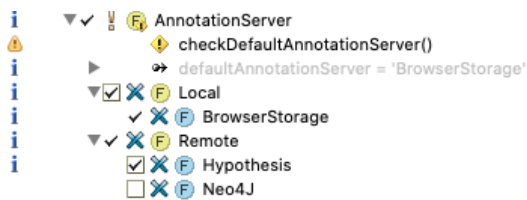
En la figura 8.1 se pueden ver la selección de las características para la aplicación DOCAL. A continuación se detallarán brevemente los motivos por los que se han seleccionado dichas características.



**Figura 8.1:** Selección de características

La aplicación DOCAL, al estar enfocada en el análisis de documentos, además de en el análisis de jurisprudencia, en necesaria la inclusión de un servidor de anotaciones público como Hypothesis, haciendo que nuestras anotaciones sean accesibles desde cualquier dispositivo. Además DOCAL contará con un servidor de anotaciones local, el cual nos permitirá la creación de copias de seguridad las cuales se podrán usar para recuperar toda

la información en caso de pérdida o cambio de dispositivo (Ver figura 8.2).



**Figura 8.2:** Selección del servidor de anotaciones

Respecto a los formatos aceptados por DOCAL, la aplicación será capaz de analizar documentos PDF además de, por los problemas analizados en el capítulo de *Módulo gestor de documentos* (ver capítulo 5), formatos html. Esto permite al usuario no solo anotar un pdf, sino que además tendrá la capacidad de realizar anotaciones en cualquier página web.

Analizando los propósitos por los que se podrían realizar las anotaciones, se ha llegado a la conclusión de que DOCAL debe disponer de la característica de comentar las anotaciones (*Commenting*), clasificarlas según su código de marcado (*Classifying*) y enlazarlas a otras anotaciones (*Linking*), las cuales pueden estar en el mismo documento o en documentos diferentes.

En el caso de las operaciones relacionadas con las anotaciones, DOCAL permitirá las opciones básicas como crear una anotación, eliminar una anotación y por supuesto leer una anotación. Además se incluirá la nueva funcionalidad de exportar las anotaciones (*DocExporter*) ya que mejora las opciones de exportación ya implementadas en WacLine, proporcionando un documento estructurado con todas las anotaciones en formato Word editable .

Por último, además de la característica de disponer de un libro de códigos *Codebook*, el cual nos permitirá crear código para realizar nuestras anotaciones, la aplicación DOCAL dispondrá de la características de (*Session*). Esta característica, es una de las nuevas características desarrolladas en este TFG la cual nos permitirá crear sesiones para diferenciar distintas actividades, es decir, diferentes grupos de documentos, que podríamos realizar en paralelo (Ver seccion 7.2).

## 8.2. Aspectos de usabilidad de la aplicación DOCAL

La aplicación DOCAL, por medio de la SPL WacLine, nos proporciona una interfaz simple y usable la cual permitirá al usuario concentrarse en el análisis de los documentos.

Las interfaces de la aplicación están detalladas en el anexo B.

Además, como objeto de este TFG y para la aplicación DOCAL, se han desarrollado varios módulos que aportan las funcionalidades necesarias para el análisis de documentos:

- Las sesiones (*Session*) las cuales proporciona al usuario la capacidad de distinguir entre anotaciones de diferentes análisis o sesiones. Con esta funcionalidad el usuario podrá anotar los mismos documentos sin que interfieran las anotaciones de los diferentes análisis.
- Enlazar anotaciones (*Linking*). En el análisis de anotaciones es imprescindible la funcionalidad de enlazar anotaciones, el usuario dispone de la capacidad de enlazar anotaciones entre documentos de la misma sesión creando así un ecosistema centrado en el análisis de documentos.
- Navegación entre enlaces. De una forma sencilla el usuario tendrá la opción de visualizar los enlaces realizados en una anotación y desplazarse a esa anotación de forma sencilla. Esta funcionalidad aporta fluidez a la hora de navegar entre las anotaciones y/o documentos.
- Comentarios en las anotaciones (*Commenting*). El usuario podrá añadir comentarios a las anotaciones que realice.
- Transferencia de anotaciones entre sesiones. Al eliminar una sesión, por seguridad, la aplicación le proporcionará al usuario la opción de transferir todas las anotaciones realizadas en la sesión a otra. Esto evitará la pérdida de la información recopilada durante la sesión eliminada.
- Libros de código (*Codebook*). El usuario tendrá la capacidad de definir sus propios códigos de marcado donde les podrá añadir una descripción que detalle la finalidad de dicho código.
- Generar informes (*DocExporter*) detallando las anotaciones realizadas durante la sesión en un documento estructurado. De esta forma el usuario tendrá la opción de generar el informe del análisis, el cual incluye los códigos definidos con sus explicaciones, las anotaciones realizadas, con los comentarios realizados por el usuario, las cuales estarán agrupadas por documentos, y los enlaces realizados entre las anotaciones. Este documento se generará automáticamente en formato docx. El formato docx de Word es un formato de documentos muy extendido y permite al usuario poder editar dicho documento para añadir o modificar la información recopilada.

## 9. CAPÍTULO

---

### Seguimiento y Control

---

En este capítulo se indican los diferentes problemas surgidos durante el desarrollo del TFG además de una tabla con las desviaciones respecto a las horas inicialmente planificadas.

#### 9.1. Problemas encontrados durante el desarrollo del TFG

A continuación se comentan los problemas más importantes encontrados durante el desarrollo del TFG.

- **Reestructuración de la SPL WacLine.** Antes del inicio del TFG se realizó una reunión de presentación con el desarrollador de WacLine, Haritz Medina. En dicha reunión se nos explicó la estructura de la SPL pero meses después, se realizó una reestructuración de la SPL para mejorar su escalabilidad y para que se adaptase al diagrama de características presentado. Por lo que hubo que realizar un aprendizaje de la nueva estructura de WacLine por cuenta propia.
- **Recuperar las anotaciones del usuario.** En la característica *Linking* es necesario trabajar con las anotaciones realizadas por el usuario, independientemente en qué documento se hayan realizado. Esta funcionalidad no estaba prevista en WacLine ya que únicamente se podía trabajar con las anotaciones de un documento en concreto. Debido a este problema, hubo que implementar esta funcionalidad en el controlador de anotaciones *annotationReader*.

- **Actualización de anotaciones.** Realizando la característica *Linking*, surgió la necesidad de actualizar anotaciones de varios documentos al mismo tiempo. Esta necesidad apareció a la hora de realizar los enlaces ya que las anotaciones que participaban en el enlace debían actualizarse. Hasta ese momento, esta opción no estaba contemplada ya que al actualizar cualquier anotación, el algoritmo modificaba el documento al que pertenecía la anotación asignándole siempre el documento que el usuario tenía abierto en ese momento. Esto produjo desviaciones importantes en las horas asignadas para la realización de la característica *Linking* pero se consiguió solventar el problema. Una vez arreglado se proporcionó un parche con la solución al desarrollador de la SPL, Haritz Medina, para que lo integrara en la versión actual de WacLine.
- **Servidor de anotaciones local.** Desde el inicio del proyecto se utilizó como servidor de anotaciones el servidor Hypothesis debido a que al utilizar el servidor de anotaciones local surgía un error que rompía la inicialización de la aplicación. Después de investigar el problema utilizando la herramienta de depuración de Google Chrome y analizando la pila de eventos llegué a la conclusión de que el problema era la función encargada del renderizado de un apartado del *Sidebar*. El problema lo solucioné modificando levemente dicha función para que se renderizara el elemento antes de manipularlo en el código.
- **Compilador de LaTeX.** Durante el transcurso de la elaboración de la memoria surgieron problemas con el compilador de LaTeX que se utilizaba hasta el momento, *pdfLaTeX*. Después de varios días de retraso se optó por cambiar el compilador de LaTeX a *XeLaTeX*, que proporciona el editor de LaTeX *Overleaf*.
- **Error en el análisis de factibilidad del TFG.** Debido a un error en el análisis de factibilidad hubo que desechar el *módulo gestor de documentos*, se han explicado los motivos en el capítulo 5. En su lugar se ha implementado el *módulo gestor de sesiones* explicado en la sección 7.2.



## 9.2. Desviaciones

En la tabla 9.1 se muestran las horas estimadas para cada paquete de trabajo definidos en la sección 2.4, el número de horas reales invertidas en cada paquete de trabajo y la desviación. Se han marcado en rojo las desviaciones superiores al 50%

Podemos observar que en el paquete del *módulo gestor de documentos* (MGA) no se ha llegado a completar debido a los problemas explicados en el capítulo 5. En cambio aparece un nuevo paquete de trabajo que engloba el desarrollo del *módulo gestor de sesiones* (MGS).

Se puede observar una desviación importante en el *módulo export a google Docs* (MEG) debido a que por complicaciones con el sistema de verificación de Google se reorientó el módulo a la exportación el formato docx.

Por último se aprecia una desviación del 57.5% en el paquete de trabajo referente a la realización de la memoria. Este desvío es causado por problemas surgidos con el compilador de LaTeX y por una mala estimación en la envergadura de la memoria.

<b>Tareas</b>	<b>Estimación horas</b>	<b>Horas realizadas</b>	<b>Desviación</b>
Proyecto Fin de Grado	319	339	6,27 %
Fase gestión de Conocimiento (GC)	65	61	-6,15 %
Adquisición de Competencias (AC)	25	17	-32 %
AC.1: pure::variants	5	3	-40 %
AC.2: Extensión Chrome	3	1	-66,66 %
AC.3: JS ES6	5	5	0 %
AC.4: Plantillas GoogleDocs	5	2	-60 %
AC.5: Node.js	2	3	-50 %
AC.6: Herramientas secundarias	5	3	-40 %
Análisis del proyecto base (APB)	40	44	10 %
APB.1: Arquitectura base	30	36	20 %
APB.2: Dudas	10	8	-20 %
Fase de Gestión (G)	47	40	-14,9 %
Planificación (P)	19	20	5,27 %
P.1: Requisitos	2	2	0 %
P.2: Entorno de desarrollo y planificación	14	16	14,3 %
P.3: Actualización de planificación	3	2	-33,33 %
Seguimiento y Control (SyC)	28	20	-28,6 %
SyC.1: Reuniones	10	4	-60 %
SyC.2: Dedicaciones	5	4	20 %
SyC.3: Entorno de control de dedicaciones	2	2	0 %
SyC.4: Recopilación de información	5	5	0 %
SyC.5: Contraste de información	6	5	17 %
Fase Desarrollo del proyecto (DP)	155	163	5,2 %
Modelo de datos (MD)	20	7	-65 %
MD.1: Modelo Anotaciones Web	10	5	-50 %
MD.2: Estructura jurisprudencia	10	2	-80 %
Módulo gestor de archivos (MGA)	55	10	-82 %
MGA.1: Análisis	15	10	-33,3 %
MGA.2: Búsqueda	20	0	- %
MGA.3: Descarga	20	0	- %
Módulo Gestor de Sesiones (MGS)	0	40	100 %
MAL.1: Análisis	0	12	100 %
MAL.2: Implementación	0	28	100 %
Módulo de anotaciones linking (MAL)	50	58	16 %
MAL.1: Análisis	20	18	-10 %
MAL.2: Implementación	30	40	33,33 %
Módulo export a GoogleDocs (MEG)	30	48	60 %
MEG.1: Plantilla	10	14	40 %
MEG.2: Procesamiento de datos	20	34	70 %
Fase Documentación (D)	52	75	44,23 %
Realización del informe (INF)	42	65	54,77 %
INF.1: Entorno LaTeX	2	2	0 %
INF.2: Desarrollo memoria	40	63	57,5 %
Preparación de la defensa (DEF)	10	10	0 %
DEF.1: Presentación	4	4	0 %
DEF.2: Revisión	6	6	0 %

**Tabla 9.1:** Desviaciones en las horas definidas para los paquetes de trabajo

## 10. CAPÍTULO

---

### Conclusiones

---

Partiendo de WacLine, una SPL para generar extensiones para el navegador Chrome y definida en el dominio de las anotaciones web, la idea original del proyecto era la de generar un producto de la SPL WacLine añadiendo las características necesarias para dicho producto. Para ello se ha cumplido el objetivo de desarrollar tres características nuevas las cuales aportan las funcionalidades de enlazar anotaciones, visualizar los enlaces de las anotaciones, crear sesiones las cuales diferencian los grupos de documentos analizados y las acciones realizadas por el usuario, transferir la información entre sesiones además de la funcionalidad de exportar todas las anotaciones a un documento Word. Con estas nuevas características y las ya implementadas en WacLine se ha podido crear automáticamente (seleccionando las características apropiadas) el producto DOCAL, una extensión Chrome para el análisis de documentos de jurisprudencia, pero además dicha extensión ha resultado útil para el análisis de todo tipo de documentos. La selección de características realizadas, incluyendo las tres nuevas características, aportan las herramientas necesarias para el análisis tanto de documentos pdf como páginas web. Además incluye la característica *Session*, la cual ha sido implementada como objetivo de este TFG, que permite analizar los mismos documentos o distintos grupos de documentos diferenciando las anotaciones realizadas en cada análisis. Además se ha incluido la característica de comentar las anotaciones, ya implementada en WacLine, y la característica de enlazar anotaciones (*Linking*) implementada en este TFG, que aportan dos herramientas importantísimas en el ámbito del análisis de documentos. En este sentido se puede concluir que los objetivos han sido llevados a cabo con éxito, mostrando también el potencial que tiene las Líneas de Producto del Software independientemente de los problemas surgidos en el transcurso

de este TFG.

A nivel técnico, los resultados obtenidos son muy positivos, se han implementado características de uso general que pasarán a formar parte de la arquitectura de WacLine. Además se ha podido trabajar con una SPL de tamaño considerable desarrollada por otra persona. Esto ha implicado demostrar, no solo la capacidad individual técnica para analizar y adaptarse a la arquitectura de WacLine, sino tener que estructurar el código propio y trabajar en contacto con el desarrollador original para amoldar las nuevas características a la arquitectura de la SPL. Además se ha trabajado con una herramienta totalmente desconocida, *Pure::Variants*, que incorpora anotaciones dentro del código para definir los puntos de variación de la SPL a diferencia de las herramientas vistas anteriormente.

A nivel personal, las conclusiones respecto a los objetivos planteados son positivas. El conocimiento que se tenía sobre las SPL era limitado ya que su utilización no está del todo extendida de momento. Además se han obtenido conocimientos no solo del dominio de las anotaciones web, sino de las extensiones para los navegadores y de la tecnología *Node.js*. Además de tener que enfrentarse por primera vez al desarrollo de un proyecto de tal envergadura el cual se asemeja a proyectos que podemos encontrar en el mundo laboral.

Como mejoras futuras para seguir extendiendo la SPL WacLine estaría la implementación de la posibilidad de incluir comentarios en los enlaces realizados por la funcionalidad *Linking*, la asignación de varios libros de códigos *Codebook* a las sesiones o el diseño de varias plantillas para la característica *DocExporter* para que el usuario pueda seleccionar el formato de documento que más le interese. Además de realizar pruebas con usuarios reales los cuales proporcionarían cierto *feedback* que ayudase a mejorar la usabilidad de las aplicaciones además de implementar mejoras a las características ya existentes.

# **Anexos**



### Plantilla módulo export

---

La plantilla contenida en este anexo se trata de una plantilla Word para crear documentos docx. El objetivo de la plantilla es describir la estructura general de los informes que recopilarán las anotaciones realizadas sobre varios documentos web. Además, al final de este anexo, se incluye un ejemplo del resultado de la exportación utilizando la esta plantilla.

Esta plantilla es la utilizada por el módulo *DocExport* para generar los documentos finales. Dicha plantilla está compuesta por una parte fija y elementos variables, que se indican entre llaves {}. Para los elementos variables hay dos tipos de etiquetas, las etiquetas de tipo a, las cuales indican la posición donde asignar el valor de la variable “a” y las etiquetas de tipo {#a}{a}, las cuales sirven para delimitar un apartado del documento que se debe repetir tantas veces como valores tenga “a”.

La plantilla está compuesta por cuatro apartados principales. El primer apartado muestra el nombre de la aplicación, ya que al integrarse en la SPL WacLine pueden generarse distintas aplicaciones, el nombre de la sesión, si existe, y la fecha en la que se ha generado el documento.

El segundo hace referencia al *Codebook*, en este apartado se nos mostrará el nombre y la descripción de cada código de marcado que hemos usado para realizar las anotaciones. Esto nos permitirá consultar en todo momento el significado del código de cada anotación.

El tercer apartado es la recopilación de las anotaciones. En él se mostrarán todas las anotaciones, con sus respectivos comentarios, agrupadas por documentos. Como se puede ver en la plantilla este apartado se va a repetir tantas veces como *documents* tengamos, el

título del documento se asignará en el *placeholder* {title}, mientras que las anotaciones se asignarán en el apartado inferior con sus comentarios, si existen.

Por último, tenemos la sección de enlaces la cual aparecerá si está la característica de *Linking* seleccionada. En este apartado aparecerán los enlaces de las anotaciones referenciados usando su identificador. Estos identificadores hacen referencia al documento y a la posición de la anotación. Es decir, la referencia A02 haría referencia a la segunda anotación del primer documento. De todas formas, todas las anotaciones incluyen su referencia al lado para facilitar su búsqueda.



# Annotation summary

**AppName:** {appName}

{#session}**Session name:** {session}/{/session}

**Date:** {date}

{#codebook}

---

## Codebook:

---

{#themes}

**{name}:** {description}

{#codes}

- **{name}:** {description}

{/codes}

{/themes}

{/codebook}

---

## Annotated Documents

---

{#documents}

{#sourceAnnotation}

---

### {title}

---

#### **{/sourceAnnotation}**

{#annotations}{#codeortheme}

- **{name}{/codeortheme}:** {#target}{#selector}{#exact} “{exact}” \*{id} {/exact}  
 {/selector}{/target}{#comments}
  - o “{value}”{/comments}

{/annotations}

{/documents}

---

## \*LINKS\*

---

{#links}

\*{a} <—> \*{b}

{/links}

# Annotation summary

**AppName:** DOCAL

**Session name:** Análisis de prueba

**Date:** 6-15-2020 10:43

---

## Codebook

---

**Cronología:** Anotaciones de fechas

**Entidades:** Entidades que participan en la resolución

**Hechos:**

- **Hechos no probados:** Hechos que no quedan totalmente probados. Sin validez
- **Hechos probados:** Los hechos que queden correctamente probados

**Leyes:** Leyes que se nombran o participan en la resolución del caso

---

## Annotated Documents

---

---

### [STS 999/2020 - ES:TS:2020:999 - Poder Judicial](#)

---

- **Cronología:** “*Fecha de Votación y Fallo: 04/03/2020*” \*0.0
- **Cronología:** “*En Madrid, a 25 de mayo de 2020*” \*0.1
- **Cronología:** “*la sentencia de 29 de mayo de 2017*” \*0.2

- **Entidades:** “Asociación Soriana para la Defensa y Estudio de la Naturaleza (en adelante, ASDEN) ” \*0.3
- **Entidades:** “ 67/2015por la Sala de lo Contencioso Administrativo del TSJ de Castilla y León (Burgos) ” \*0.4
- **Entidades:** “ La Comunidad de Castilla y León ” \*0.5
- **Entidades:** “I Ayuntamiento de Garray ” \*0.6
- **Cronología:** “La sentencia de 29 de mayo de 2017 co” \*0.7
  - o “Esto es un comentario en la anotación”
- **Hechos:** “tiene parte dispositiva del siguiente tenor: «1°).- Serechaza de conformidad con lo razonado en el Fundamento de Derecho Quinto la causa de inadmisibilidadesgrimida tanto por la Administración Autonómica demandada como por el Ayuntamiento codemandado” \*0.8
  - o “Se trata de un hecho que ha demostrado ser un error”

---

## STSJ PV 30/2020 - ES:TSJPV:2020:30 - Poder Judicial

---

- **Hechos no probados:** “Magistrado don Juan Carlos Iturri Gárate ni la Ilma. Sra. Magistrada doña MaiteAlejandro Aranzamendi, siendo sustituidos por los Ilmos. Sres. don José Luis Asenjo Pinilla y don JoséFélix Lajo González, siendo asignada la ponencia al Ilmo. Sr. Magist” \*1.0
- **Hechos probados:** “ne en los territorios de Bizkaia, 32 bases entre transporte sanitario denominado "concertado" y " RTSU" y de Gipuzkoa, 15 bases entre RTSU, Servicio Vital Básico 14 y RTSU, SVA enfermerizada, 1, afectando, aproximadamente, a 825 traba” \*1.1
- **Cronología:** “ realizan tareas de abordaje de emergencias a exposición a agentes biológicos con apercibimientopara el caso de que se incumpla el requerimiento 0/002408/20; en contestación al escrito de 26 de marzode 2020 se señala que el empresario tiene obligación de informar a los trabajadores y sus representantessobre las situaciones laborales en las que se puede producir un contacto estrecho con casos probables oconfirmados SARS-CoV-2, y adoptar las medidas para garantizar la salud y seguridad de los trabajadores, sin que dicha obligación suponga la identificación de las personas afectadas; informe de fecha de salidade 22-4-2020, por el que se requie” \*1.2

- **Entidades:** “*tiones anteriores. Nada se acuerda -indica el informe- en cuanto que se entiende que son materias relativas a las autoridades sanitarias; informe con fecha de salida de 22-4-2020, sobre lavado de ropa, nollevándose -señala- a cab*”  
\*1.3
  - o “Esto es un comentario es una anotación para comprobar el export”
- **Leyes:** “*tado por la Sala de lo Social del TSJPV en medidas cautelarísimas en este procedimiento, pieza separada 4/20, se ha procedido por la demandada a suministrar al personal afectado gel hidroalcohólico, buzos, guantes, mascarillas FFP2, y*” \*1.4

---

## \*LINKS\*

---

\*0.1 <—> \*1.2

\*1.1 <—> \*0.2

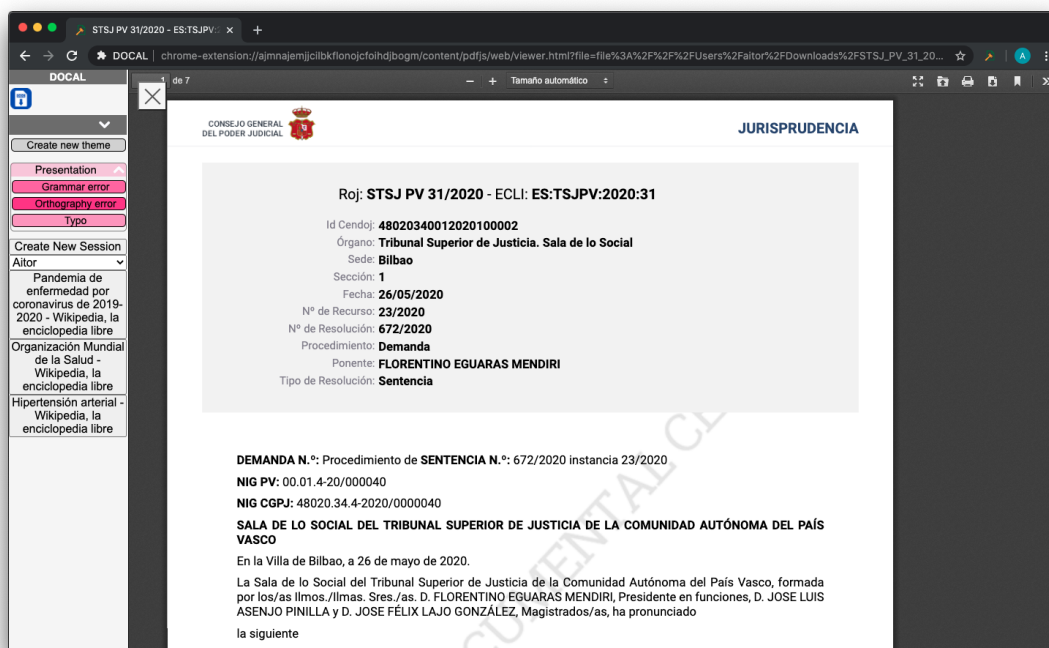
## B. ANEXO

---

### Diseño de las interfaces

---

Las aplicaciones generadas por WacLine son extensiones. Esto quiere decir que toda la interfaz de usuario debe ser simple y discreta para no obstaculizar la visión del documento que se esté analizando. Por ello WacLine cuenta con una interfaz a modo de *Sidebar* como podemos observar en la figura B.1. Esta interfaz, la mayoría del tiempo, permanece oculta para no obstaculizar la visión del documento. El *Sidebar* se despliega automáticamente al detectar que se ha seleccionado alguna parte del documento para anotarlo, además de poder desplegarse en cualquier momento pulsando el botón que esta siempre visible en la parte superior izquierda. A continuación se explican los apartados del *sidebar* y las alertas utilizadas por la aplicación.

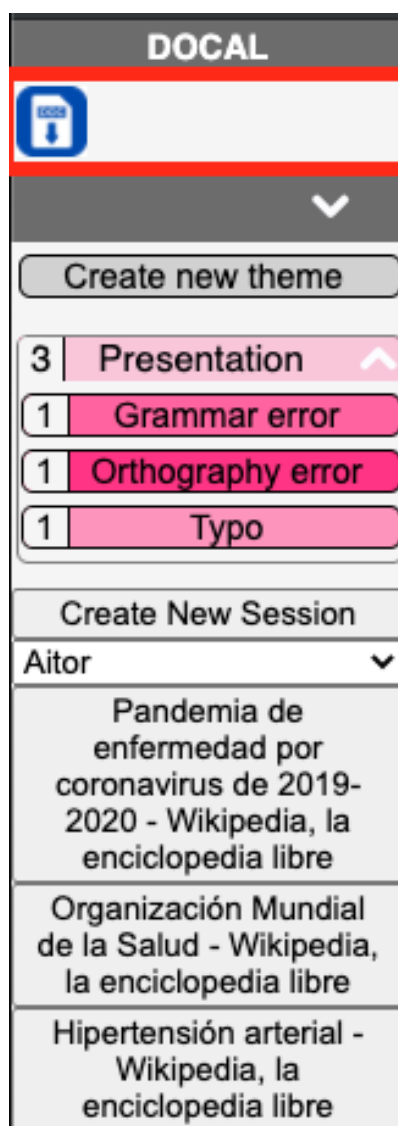


**Figura B.1:** Captura de pantalla de la interfaz de usuario

## B.1. Sidebar

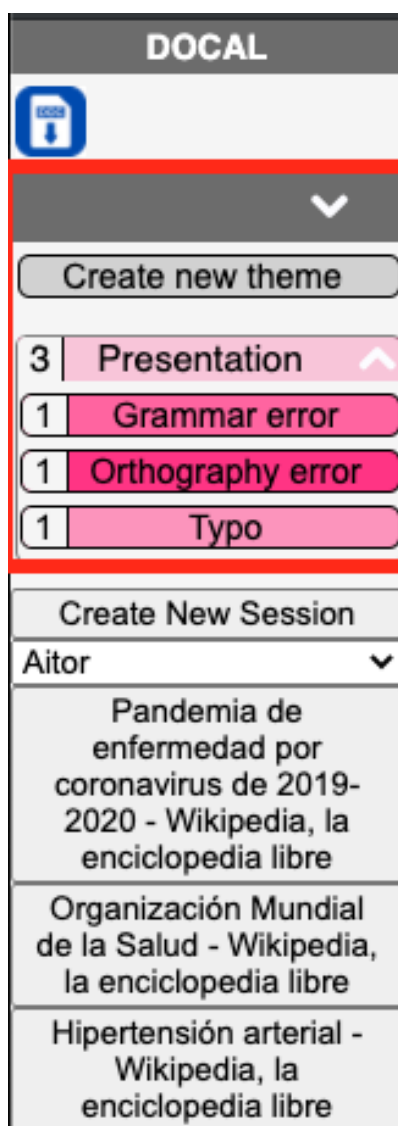
El *Sidebar* es la interfaz de usuario principal de la aplicación. En ella podemos ver que se distribuye toda la información referente a las anotaciones y acciones que puede realizar el usuario. En este apartado analizaremos las partes que componen el *Sidebar*.

En primer lugar, como podemos observar en la imagen B.2, se puede ver el nombre de la aplicación y resaltado en rojo, podemos ver el apartado *ToolBox*. En este apartado se encuentran las herramientas disponibles, en este caso encontramos solamente la herramienta de exportar anotaciones a un documento Word.



**Figura B.2:** Sección ToolBox del Sidebar

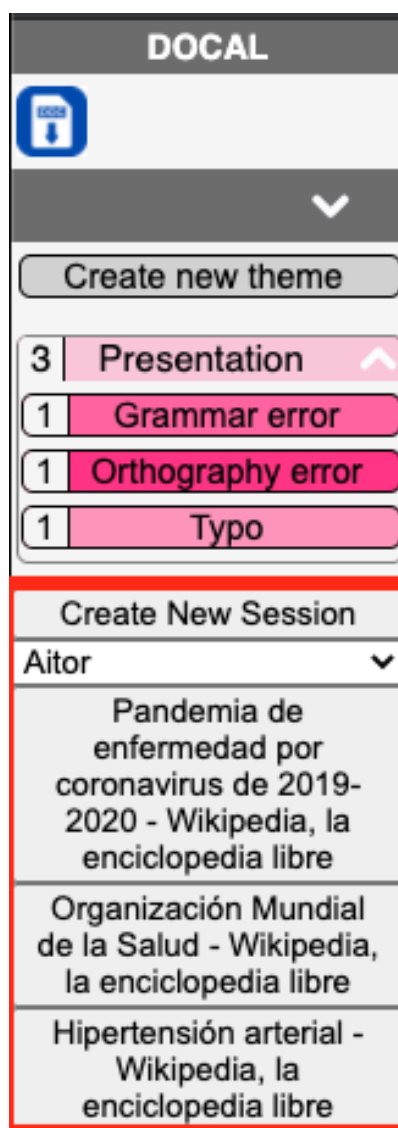
A continuación podemos observar el apartado *Codebook*, donde se encuentran todos los temas y códigos asociados al libro de códigos definido por el usuario. Estos códigos nos permiten agrupar las anotaciones y clasificarlas dándoles un contexto. Además, podemos observar un contador al lado de cada código el cual nos indica el número de anotaciones del mismo código que hay en el documento actual. En la figura B.3 podemos observar, resaltado en rojo, este apartado y cómo se ha realizado una anotación por código.



**Figura B.3:** Sección Codebook del Sidebar

Por último, podemos observar el apartado de las sesiones (Ver figura B.4). En este apartado podemos observar un botón para crear sesiones, un desplegable con la sesión actual y sesiones disponibles, y el índice de documentos anotados. En esta última parte se muestran los documentos anotados en la sesión pudiendo abrirlos en cualquier momento pulsando sobre ellos.





**Figura B.4:** Sección de sesiones del Sidebar

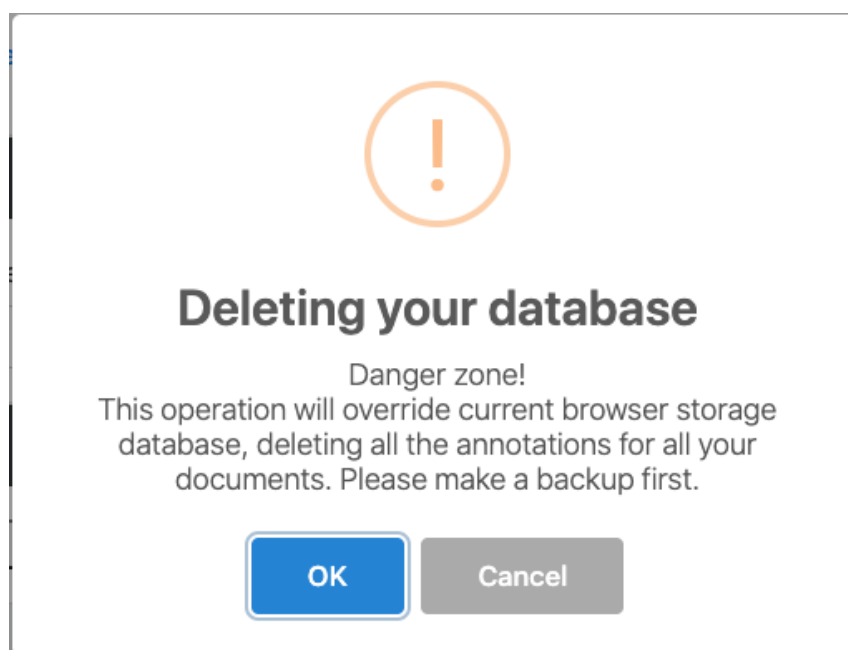
## B.2. Alertas

Debido a lo comentado anteriormente, la aplicación hace uso de *Alerts*, que son en esencia interfaces emergentes, para interactuar y mostrar información concreta.

Para ello la aplicación dispone de distintos tipos de *Alerts*:

### B.2.1. Alerta de aviso

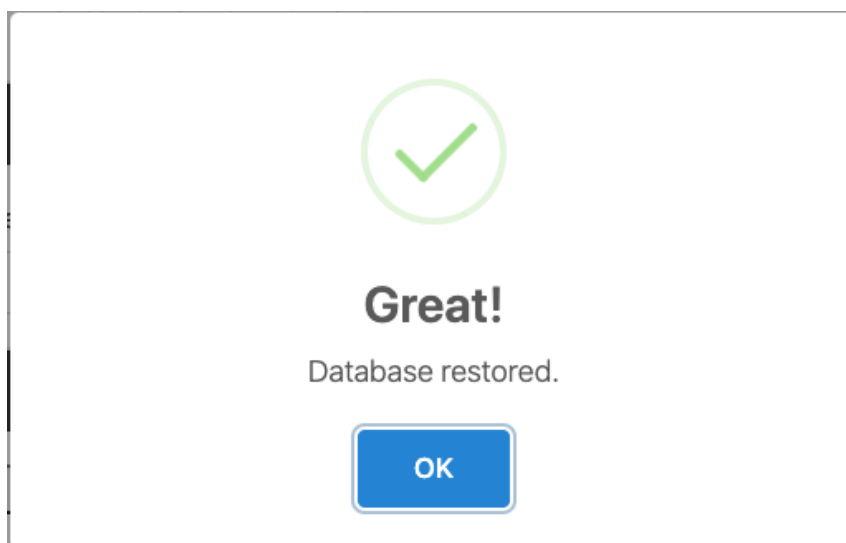
Estos *Alerts* se pueden distinguir por el icono de exclamación. Estas ventanas se utilizan para informar de una acción potencialmente peligrosa la cual no es habitual, como por ejemplo eliminar la base de datos local (ver figura B.5).



**Figura B.5:** Ejemplo de alerta de aviso

### B.2.2. Alerta de confirmación

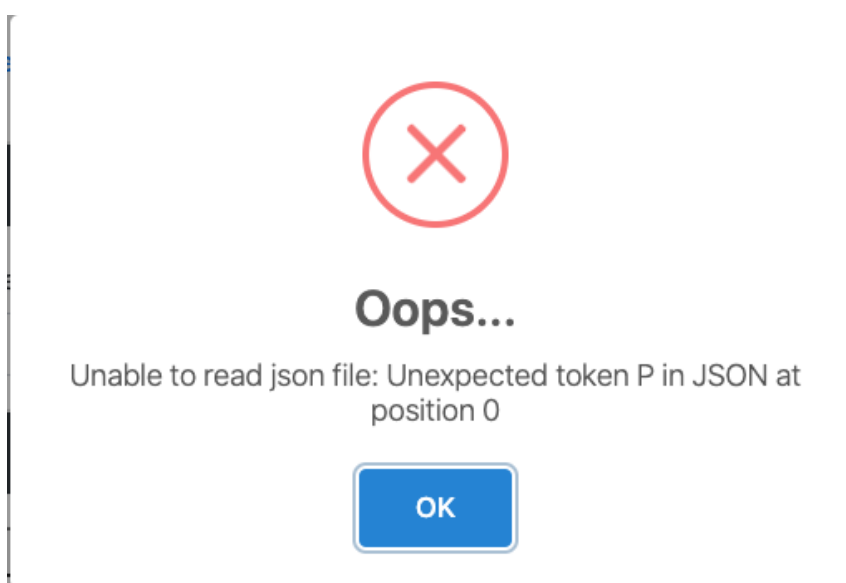
Las alertas de confirmación, generalmente son usadas para proporcionar *feedback* positivo al usuario. Estas suelen aparecer al completarse exitosamente acciones importantes en la aplicación. Por ejemplo cargar una copia de la base de datos (Ver imagen B.6).



**Figura B.6:** Ejemplo de alerta de confirmación

#### Alerta de error

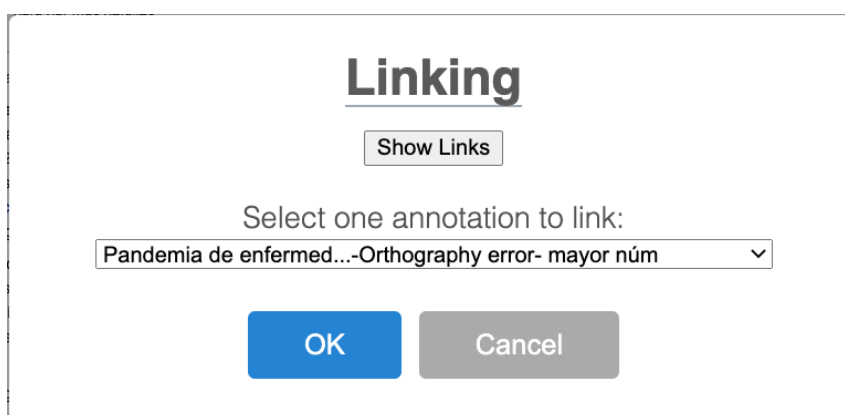
Este tipo de alertas informan al usuario de que ha ocurrido un error realizando la acción solicitada. Estas alertas suelen ir acompañadas de un mensaje de error explicando el motivo por el que ha fallado la acción (Ver figura B.7).



**Figura B.7:** Ejemplo de alerta de error

### B.2.3. Alerta de propósito general

Este último tipo de alerta se utiliza para todo tipo de acciones. El resto de alertas mostradas eran informativas ya que informaban de lo peligrosa que era la acción o si la acción se había completado con éxito o no. En este caso las alertas de propósito general se utilizan a modo de interfaces interactivas para permitir al usuario que realice la acción deseada. Por ejemplo en la figura B.8 se utiliza este tipo de alerta para mostrar al usuario las opciones disponibles a la hora de enlazar una anotación.



**Figura B.8:** Ejemplo de alerta de propósito general

### Implementación de las características

---

En este anexo se encuentra el código de los nuevos archivos incluidos en WacLine los cuales no contienen puntos de variación y están comentados en el capítulo 7.

---

#### Listing C.1 Creación del *Session Manager*

```
1  constructor () {
2      this.events = {}
3      this.sessionCreator = new CreateSession()
4      this.sessionReader = new ReadSession()
5      this.sessionUpdater = new UpdateSession()
6  }
7
8  init (callback) {
9      this.sessionReader.init(() => {
10         this.sessionUpdater.init(() => {
11             this.sessionCreator.init(() => {
12                 if (_.isFunction(callback)) {
13                     callback()
14                 }
15             })
16         })
17     })
18 }
19 }
```

#### Listing C.2 Creación del menú contextual de los documentos anotados

```
1  createContextMenuForSessionURI (uri) {
2      $.contextMenu({
```

```

3     selector: '[data-sessionURI-source="' + uri.source + '"',
4     build: () => {
5         const items = {}
6         // Delete uri
7         items.delete = { name: 'Delete' }
8         return {
9             callback: (key, opt) => {
10                if (key === 'delete') {
11                    this.currentSession.sessionURIs = this.currentSession.sessionURIs.filter((curi) =>
12                { return uri.source !== curi.source })
13                    LanguageUtils.dispatchCustomEvent(Events.updateSession, { session:
14                this.currentSession })
15                }
16            },
17            items: items
18        }
19    })
20 }

```

### Listing C.3 Recuperación de las sesiones del usuario

```

1 retrieveSessions (callback) {
2     let promise = new Promise((resolve, reject) => {
3         let call = {}
4         call.group = window.abwa.groupSelector.currentGroup.id
5         call.order = 'asc'
6         call.sort = 'updated'
7         call.tag = Config.namespace + ':isSession'
8         window.abwa.annotationServerManager.client.searchAnnotations(call, (err, sessionObjects) =>
9         {
10            if (err) {
11                reject(err)
12            } else {
13                // console.log(sessionObjects)
14                let sessions = sessionObjects.map((session) => Session.deserialize(session))
15                sessions = _.orderBy(sessions, 'updated', 'desc')
16                console.log(sessions)
17                resolve(sessions)
18            }
19        })
20    })
21    promise.catch((err) => {
22        callback(err)
23    }).then((annotations) => {
24        callback(null, annotations)
25    })
26 }

```

### Listing C.4 Renderización de la interfaz de usuario

```
1 reloadSessionContainer () {
2     this.sidebarContainer = document.querySelector('#abwaSidebarContainer')
3     this.uriContainer = this.sidebarContainer.querySelector('#documentsContainer')
4     this.sessionSelector = this.sidebarContainer.querySelector('#sessionSelector')
5     this.sessionSelector.setAttribute('data-session-selector', 'sessionSelector')
6     this.sessionSelector.innerHTML = ''
7     this.uriContainer.innerHTML = ''
8     this.createContextMenuForSession()
9     this.sessions.forEach((session) => {
10        let option = document.createElement('option')
11        option.text = session.sessionName
12        option.value = session.id
13        if (!_isUndefined(this.currentSession) && session.id === this.currentSession.id) {
14            option.setAttribute('selected', '')
15        }
16        this.sessionSelector.appendChild(option)
17    })
18 }
```

### Listing C.5 Funciones encargadas de la conversión de las sesiones

```
1 toAnnotation () {
2     const sessionTag = Config.namespace + ':session:' + this.group
3     const tags = [sessionTag, Config.namespace + ':isSession']
4     // Construct text attribute of the annotation
5     return {
6         name: this.sessionName,
7         group: this.group,
8         permissions: {
9             read: ['group:' + this.group]
10        },
11        references: [],
12        motivation: 'defining',
13        tags: tags,
14        target: [],
15        sessionURIs: this.sessionURIs || []
16    }
17 }
19 static deserialize (annotation) {
20     const sessionOptions = {
21         id: annotation.id,
22         sessionName: annotation.name,
23         sessionURIs: annotation.sessionURIs,
24         group: annotation.group,
25         updated: annotation.updated
26     }
27     let session
```

```

28     session = new Session(sessionOptions)
29     console.log(session)     return session
30 }

```

### Listing C.6 Creación del menú contextual de las sesiones

```

1  createContextMenuForSession () {
2      $.contextMenu({
3          selector: '[data-session-selector="sessionSelector"]',
4          build: () => {
5              let session = this.sessions.filter(s => s.id ===
6                  document.querySelector('[data-session-selector="sessionSelector"]').value)[0]
7              const items = {}
8              // Rename session
9              items.rename = { name: 'Rename' }
10             // Delete session
11             items.delete = { name: 'Delete' }
12             return {
13                 callback: (key, opt) => {
14                     if (key === 'delete') {
15                         LanguageUtils.dispatchCustomEvent(Events.deleteSession, { session: session })
16                     } else if (key === 'rename') {
17                         this.renameSessionForm(session)
18                     }
19                 },
20                 items: items
21             }
22         }
23     })
24 }

```

### Listing C.7 Código de la opción *Rename*

```

1  renameSessionForm (session) {
2      Alerts.multipleInputAlert({
3          title: 'Reaname current session',
4          html: '<input type="text" id="newSessionName" placeholder="New Session Name">',
5          preConfirm: () => {
6              let newName = document.querySelector('#newSessionName').value
7              session.sessionName = newName
8              LanguageUtils.dispatchCustomEvent(Events.updateSession, { session: session })
9          }
10     })
11 }

```

### Listing C.8 Código de la opción de transferir sesión





**Listing C.10** Filtrado y generación de documento con las anotaciones enlazadas

```

1  static generateShowLinksHtml ({ annotation }) {
2      let links = {}
3      let html = '<div id="table-div" style="overflow:auto;" >'
4      annotation.annotationlinks.forEach((link) => {
5          let targetAnnotation = _.find(
6              window.abwa.annotationManagement.annotationReader.allServerAnnotations, (a) => {
7                  return a.id === link })
8          let text = ''
9          targetAnnotation.target[0].selector.forEach((select) => {
10             if (select.type === 'TextQuoteSelector') {
11                 text = select.exact
12             }})
13             if (links[targetAnnotation.target[0].source.title]) {
14                 links[targetAnnotation.target[0].source.title].push(targetAnnotation)
15             } else {
16                 links[targetAnnotation.target[0].source.title] = [targetAnnotation]
17             }})
18             for (const [key, values] of Object.entries(links)) {
19                 const title = document.createElement('h6')
20                 title.innerText = key
21                 const ul = document.createElement('ul')
22                 values.forEach((an) => {
23                     const li = document.createElement('li')
24                     li.id = an.id
25                     li.innerText = _.find(an.target[0].selector, (s) => { return s.type === '
TextQuoteSelector' }).exact
26                     li.style = 'cursor: pointer; font-size:10px'
27                     ul.appendChild(li)}})
28                 html += title.outerHTML + ul.outerHTML}
29             html += '</div>'
30             return html
31         }

```

**Listing C.11** Funcionalidad de navegar entre anotaciones de distintos documentos

```

1  onBeforeOpen: () => {
2      document.getElementById('table-div').getElementsByTagName('ul').forEach((ul) => {
3          ul.onclick = (e) => {
4              Alerts.closeAlert()
5              let targetAnnotation =
6              window.abwa.annotationManagement.annotationReader.allServerAnnotations.find((annot) => {
7                  return annot.id === e.target.id
8              })
9              window.open((targetAnnotation.target[0].source.url ||
10                 targetAnnotation.target[0].source.uri) + '#' + Config.urlParamName + ':' +
11                 targetAnnotation.id)
12          }

```

**Listing C.12** Generación del objeto con las anotaciones procesadas

```
1  static exportAllAnnotationsToDocx () {
2      let exporterObject = {}
3      let annotationExport = []
4      let annotations = []
5      let linksExport = []
6      // PVSCL:IFCOND(Linking,LINE)
7      annotations = window.abwa.annotationManagement.annotationReader.allServerAnnotations
8      // PVSCL:ELSECOND
9      annotations = window.abwa.annotationManagement.annotationReader.allAnnotations
10     // PVSCL:ENDCOND
11     annotations.forEach((a) => {
12         a.group = ''
13         a.permissions = {}
14         a.body.forEach((b) => {
15             if (b.purpose && b.purpose === 'commenting') {
16                 a.comments = { value: b.value }
17             }
18             if (b.purpose && b.purpose === 'classifying') {
19                 a.codeortheme = b.value
20             }
21         })
22         let sourceAnnotation = a.target[0].source
23         if (annotationExport.some((an) => { return an.sourceAnnotation.url === a.target[0]
24             .source.url }))) {
25             annotationExport.forEach((an) => { if (an.sourceAnnotation.url === a.target[0].source.url
26             ) { an.annotations.push(a) } })
27         } else {
28             annotationExport.push({ sourceAnnotation, annotations: [a] })
29         })
30     })
31     *****
32     let date = new Date()
33     exporterObject = {
34         codebook: window.abwa.codebookManager.codebookReader.codebook,
35         appName: chrome.runtime.getManifest().name,
36         date: date.getMonth() + 1 + '-' + date.getDate() + '-' + date.getFullYear() + ' ' +
37         date.getHours() + ':' + date.getMinutes(),
38         // PVSCL:IFCOND(Session,LINE)
39         session: window.abwa.sessionManagement.sessionReader.currentSession.sessionName,
40         // PVSCL:ENDCOND
41         // PVSCL:IFCOND(Linking,LINE)
42         links: linksExportsUnique,
43         // PVSCL:ENDCOND
44         documents: annotationExport
45     }
46     DocxExporter.export(exporterObject)}
```

**Listing C.13** Generación del documento final con las anotaciones

```
1  static export (object) {
```

```
3   var PizZip = require('pizzip')    var DocxTemplater = require('docxtemplater')
4   const path = chrome.extension.getURL('content/docxExporter/DocxTemplate.docx')
5   /* Recuperamos la plantilla y la cargamos en formato binario */
6   axios.get(path, { responseType: 'arraybuffer' }).then((response) => {
7     let fileTemplate = new Blob([response.data])
8     FileUtils.readBinaryFile(fileTemplate, (_err, content) => {
9       /* Añadimos la plantilla al objeto DocxTemplater */
10      let zip = new PizZip(content)
11      let doc = new DocxTemplater()
12      doc.loadZip(zip)
13      /* Asignamos el objeto creado previamente con los datos a la plantilla y lo renderizamos
14      */
14      doc.setData(object)
15      doc.render()
16
17      /* Guardamos el documento resultante */
18      let buf = doc.getZip().generate({
19        type: 'blob',
20        mimeType: 'application/vnd.openxmlformats-officedocument.wordprocessingml.document'
21      })
22
23      FileSaver.saveAs(buf, 'output.docx')
24
25    })
26  })
27 }
```

---

## Bibliografía

---

- [DocxTemplater, 2019] DocxTemplater (2019). Docxtemplater library. <https://docxtemplater.com>.
- [Eichberg, 2017] Eichberg, D. M. (2016-2017). Software engineering design & construction. <http://stg-tud.github.io/sedc/Lecture/ws16-17/6-SPL.pdf>.
- [pure-systems GmbH, 2006] pure-systems GmbH (2006). Technical white paper variant management with pure::variants. <https://www.pure-systems.com/mediapool/pv-whitepaper-en-04.pdf>.