

## Trabajo de Fin de Grado

Grado en Ingeniería Informática

*Ingeniería de Software*

---

# **Una herramienta para la revisión de manuscritos de investigación: un enfoque basado en Líneas de Producto**

---

*Iñigo Bereciartua Rocha*

**Dirección**

Óscar Díaz García

Junio de 2021



# Agradecimientos

En primer lugar, me gustaría agradecer a Óscar, mi tutor, la oportunidad de realizar este Trabajo de Fin de Grado para participar en el desarrollo de la Línea de Productos Software WACline.

En segundo lugar, me gustaría agradecerle a Haritz Medina, desarrollador de WACline, la paciencia y ayuda prestada para asimilar el proyecto base así como su involucración a lo largo de todo el proyecto.

También me gustaría agradecer a mis compañeros de universidad más cercanos, en concreto a David, Diego, Aingeru, Luis e Ibai por haberme acompañado estos años.

Por último me gustaría agradecer a mi familia y amigos por estar ahí y preguntarme acerca de como iba el proyecto a pesar de que cada vez que les explicaba algo no lo entendieran.



# Resumen

En este Trabajo de Fin de Grado se ha desarrollado una herramienta para la revisión de manuscritos de investigación mediante la Línea de Productos Software (LPS) WACline. Para llevar a cabo este Trabajo de Fin de Grado se han realizado modificaciones en características ya implementadas en la LPS y se han añadido algunas nuevas para que el producto que sirva para facilitar la revisión de artículos: importar listas de revisión predefinidas, evaluar los criterios de las listas importadas y editar un informe que resume la revisión realizada.



# Índice de contenidos

|   |             |
|---|-------------|
| <b>Índice de contenidos</b>                                   | <b>v</b>    |
| <b>Índice de figuras</b>                                      | <b>VIII</b> |
| <b>Índice de tablas</b>                                       | <b>x</b>    |
| <b>1 Introducción</b>   | <b>1</b>    |
| <b>2 Planificación inicial</b>                                | <b>3</b>    |
| 2.1. Antecedentes . . . . .                                   | 3           |
| 2.2. Motivación . . . . .                                     | 4           |
| 2.3. Objetivos . . . . .                                      | 4           |
| 2.4. Alcance . . . . .  | 5           |
| 2.5. Estructura de descomposición del trabajo (EDT) . . . . . | 5           |
| 2.5.1. Fase de Gestión (G) . . . . .                          | 6           |
| 2.5.2. Fase de Estudio de Competencias (EC) . . . . .         | 6           |
| 2.5.3. Fase de Desarrollo del Proyecto (DP) . . . . .         | 7           |
| 2.5.4. Fase de Documentación (D) . . . . .                    | 7           |
| 2.6. Estimación de dedicaciones . . . . .                     | 8           |
| 2.7. Diagrama de Gantt . . . . .                              | 9           |
| 2.8. Herramientas . . . . .                                   | 9           |
| 2.8.1. Trello . . . . .                                       | 9           |
| 2.8.2. Git, Github y GitKraken . . . . .                      | 10          |
| 2.8.3. Visual Studio Code . . . . .                           | 10          |
| 2.8.4. Google Chrome . . . . .                                | 10          |
| 2.8.5. pure::variants y Eclipse . . . . .                     | 10          |
| 2.8.6. Node.js y ECMAScript . . . . .                         | 10          |
| 2.8.7. LaTeX y Oveleaf . . . . .                              | 10          |
| 2.8.8. Figma . . . . .  | 11          |
| 2.8.9. Bizagi Modeler . . . . .                               | 11          |
| 2.8.10. Postman . . . . .                                     | 11          |
| 2.8.11. Google Drive . . . . .                                | 11          |
| 2.9. Interesados . . . . .                                    | 11          |
| 2.10. Gestión de riesgos . . . . .                            | 11          |
| 2.10.1. Dificultad para entender el proyecto base . . . . .   | 12          |
| 2.10.2. Problemas con las tecnologías utilizadas . . . . .    | 12          |
| 2.10.3. Planificación incorrecta . . . . .                    | 12          |

|  |           |
|--|-----------|
| 2.10.4. Problemas de índole personal . . . . .                                   | 13        |
| 2.10.5. Perdida de información . . . . .   | 13        |
| 2.11. Plan de calidad . . . . .  | 13        |
| 2.11.1. Memoria . . . . .  | 13        |
| 2.11.2. Producto . . . . .   | 14        |
| <b>3 Análisis de requisitos</b>  | <b>15</b> |
| 3.1. Requisitos funcionales . . . . .  | 15        |
| 3.2. Requisitos no funcionales . . . . .   | 17        |
| 3.3. Casos de uso . . . . .  | 17        |
| <b>4 Entorno</b>   | <b>21</b> |
| 4.1. Líneas de Producto Software (LPS) . . . . .                                 | 21        |
| 4.1.1. Diagrama de características . . . . .                                     | 23        |
| 4.2. pure::variants . . . . .  | 24        |
| <b>5 Arquitectura de la plataforma de LPS WACline</b>                            | <b>29</b> |
| 5.1. Estructura de las Anotaciones Web . . . . .                                 | 29        |
| 5.2. Características de WACline . . . . .  | 30        |
| 5.3. Estructura del proyecto . . . . .   | 32        |
| 5.4. Review&Go . . . . .   | 34        |
| <b>6 Diseño e implementación</b>   | <b>35</b> |
| 6.1. Módulo Búsqueda de Keywords (MBK) . . . . .                                 | 35        |
| 6.1.1. Objetivos . . . . .   | 35        |
| 6.1.2. Implementación . . . . .  | 35        |
| 6.1.3. Problemas encontrados . . . . .   | 36        |
| 6.2. Módulo Búsqueda de Información de Autores (MIA) . . . . .                   | 37        |
| 6.2.1. Objetivos . . . . .   | 37        |
| 6.2.2. Interfaces . . . . .  | 38        |
| 6.2.3. Implementación . . . . .  | 38        |
| 6.2.4. Problemas encontrados . . . . .   | 40        |
| 6.3. Módulo Métodos de revisión (MMR) . . . . .                                  | 41        |
| 6.3.1. Objetivos . . . . .   | 41        |
| 6.3.2. Interfaces . . . . .  | 41        |
| 6.3.3. Implementación . . . . .  | 47        |
| 6.3.4. Problemas surgidos . . . . .  | 49        |
| 6.4. Características versión final . . . . .                                     | 50        |
| <b>7 Desarrollo de iteraciones</b>   | <b>51</b> |
| 7.1. Primera iteración: Búsqueda de keywords (MBK) . . . . .                     | 51        |
| 7.2. Segunda iteración: Información de autores (MIA) . . . . .                   | 52        |
| 7.3. Tercera iteración: Métodos de revisión (MMR) . . . . .                      | 53        |
| 7.4. Cuarta iteración: Corrección del módulo Métodos de revisión (CMR) . . . . . | 56        |
| <b>8 Seguimiento y Control</b>   | <b>59</b> |
| <b>9 Conclusiones</b>  | <b>61</b> |
| 9.1. Conclusiones . . . . .  | 61        |

|  |           |
|--|-----------|
| 9.1.1. Conclusiones a nivel técnico . . . . .  | 61        |
| 9.1.2. Conclusiones a nivel personal . . . . . | 61        |
| 9.2. Posibles mejoras . . . . .                | 62        |
| <b>Anexos</b>                                  | <b>63</b> |
| <b>A. Extensiones Chrome</b>                   | <b>65</b> |
| <b>B. Demo del producto</b>                    | <b>67</b> |
| Carga del manuscrito . . . . .                 | 68        |
| Información de autores . . . . .               | 68        |
| Anotaciones y manejo del codebook . . . . .    | 69        |
| Evaluación de criterios . . . . .              | 72        |
| Informe . . . . .                              | 77        |
| <b>C. Actas de reuniones</b>                   | <b>81</b> |
| <b>Bibliografía</b>                            | <b>93</b> |

# Índice de figuras

|       |   |    |
|-------|---|----|
| 2.1.  | Diagrama E.D.T. . . . . .   | 6  |
| 2.2.  | Diagrama Gantt inicial - Primera parte . . . . .                              | 9  |
| 2.3.  | Diagrama Gantt inicial- Segunda parte . . . . .                               | 9  |
| 3.1.  | Diagrama proceso de revisión de manuscrito (Clasificación) . . . . .          | 16 |
| 3.2.  | Diagrama proceso de revisión de manuscrito (Anotación y evaluación) . . . . . | 16 |
| 3.3.  | Diagrama proceso de revisión de manuscrito (Informe) . . . . .                | 17 |
| 3.4.  | Diagrama de casos de uso Review&Go+ . . . . .                                 | 18 |
| 4.1.  | Coste desarrollo LPS[1] . . . . .   | 22 |
| 4.2.  | Ingeniería de Dominio y de Producto[2] . . . . .                              | 23 |
| 4.3.  | Ejemplo diagrama de características . . . . .                                 | 24 |
| 4.4.  | Diagrama de características de la LPS Weather Station . . . . .               | 25 |
| 4.5.  | Selección de características para variante de Weather Station . . . . .       | 25 |
| 4.6.  | <i>Family Model variante Ankara de Weather Station</i> . . . . .              | 26 |
| 4.7.  | Variante Ankara generada a partir de la plataforma Weather Station . . . . .  | 26 |
| 4.8.  | Condición de característica pure::variants . . . . .                          | 27 |
| 5.1.  | Esquema del dominio WebAnnotation . . . . .                                   | 30 |
| 5.2.  | Prefijo, sufijo y exact . . . . .   | 30 |
| 5.3.  | Diagrama de características WACline AnnotationServer y target . . . . .       | 31 |
| 5.4.  | Diagrama de características WACline operation e import/export . . . . .       | 32 |
| 5.5.  | Diagrama de características WACline Codebook . . . . .                        | 32 |
| 5.6.  | Estructura proyecto WACline . . . . .   | 32 |
| 5.7.  | Estructura scripts proyecto WACline . . . . .                                 | 33 |
| 5.8.  | Subconjunto de características de WACline, Review&Go . . . . .                | 34 |
| 6.1.  | Nueva característica MBK . . . . .  | 36 |
| 6.2.  | Interfaz selección congreso . . . . .   | 38 |
| 6.3.  | Interfaz selección congreso autocompletado . . . . .                          | 39 |
| 6.4.  | Información autores . . . . .   | 39 |
| 6.5.  | Interfaz información de autores (sin autores) . . . . .                       | 39 |
| 6.6.  | Nueva característica MIA . . . . .  | 40 |
| 6.7.  | <i>Sidebar</i> con nuevo botón de autores . . . . .                           | 40 |
| 6.8.  | Seleccionar lista de criterios . . . . .                                      | 42 |
| 6.9.  | Seleccionar lista de criterios (desplegada) . . . . .                         | 42 |
| 6.10. | Seleccionar criterios de una lista . . . . .                                  | 43 |
| 6.11. | Método importado con éxito . . . . .  | 43 |
| 6.12. | <i>Sidebar</i> con nuevos criterios . . . . .                                 | 44 |

|   |    |
|---|----|
| 6.13. Antiguo <i>Canvas</i> . . . . .   | 45 |
| 6.14. <i>Canvas</i> modificado . . . . .  | 45 |
| 6.15. <i>Canvas</i> criterios . . . . .   | 46 |
| 6.16. Vista criterio . . . . .  | 46 |
| 6.17. Like/Dislike . . . . .  | 46 |
| 6.18. Contexto de anotación . . . . .   | 47 |
| 6.19. Edición de informe . . . . .  | 47 |
| 6.20. Informe descargado . . . . .  | 48 |
| 6.21. Nueva característica MMR . . . . .  | 48 |
| 6.22. Características descartadas y modificadas para el producto Review&Go+ . . . . . | 49 |
| 6.23. Subconjunto de características de WACline, Review&Go+ . . . . .                 | 50 |
| 7.1. Prototipo elegir lista de criterios . . . . .                                    | 54 |
| 7.2. Prototipo canvas lista de criterios . . . . .                                    | 55 |
| 7.3. Prototipo evaluación de criterio . . . . .                                       | 55 |
| 7.4. Prototipo modificar informe . . . . .  | 56 |
| 1. Esquema elementos extensión [3] . . . . .  | 65 |
| 2. Proceso de revisión de la demo . . . . .   | 67 |
| 3. Icono extensión Review&Go . . . . .  | 68 |
| 4. Seleccionar congreso del documento . . . . .                                       | 69 |
| 5. Barra lateral con <i>themes</i> predefinidos . . . . .                             | 69 |
| 6. Información de autores (no authors) . . . . .                                      | 69 |
| 7. Información de autores . . . . .   | 70 |
| 8. Crear nuevo <i>theme</i> . . . . .   | 70 |
| 9. Opciones <i>theme</i> . . . . .  | 70 |
| 10. Crear nuevo <i>code</i> . . . . .   | 71 |
| 11. Nuevo <i>code</i> creado . . . . .  | 71 |
| 12. Crear anotación . . . . .   | 71 |
| 13. Opciones anotación . . . . .  | 72 |
| 14. Comentar anotación . . . . .  | 72 |
| 15. Categorías anotación . . . . .  | 72 |
| 16. Autocompletado al sugerir documentos . . . . .                                    | 73 |
| 17. <i>Canvas</i> anotaciones . . . . .   | 73 |
| 18. Seleccionar lista de criterios . . . . .  | 74 |
| 19. Seleccionar lista de criterios (desplegado) . . . . .                             | 74 |
| 20. Seleccionar lista de criterios (filtro <i>Evaluation</i> ) . . . . .              | 75 |
| 21. Seleccionar criterios de una lista . . . . .                                      | 75 |
| 22. <i>Sidebar</i> con nuevos criterios . . . . .                                     | 76 |
| 23. <i>Canvas</i> criterios (antes de evaluar) . . . . .                              | 76 |
| 24. Evaluación de criterio ‘Site description’ . . . . .                               | 77 |
| 25. Contexto de anotación . . . . .   | 77 |
| 26. <i>Canvas</i> criterios (evaluados) . . . . .                                     | 77 |
| 27. Edición de informe . . . . .  | 78 |
| 28. Borrador guardado . . . . .   | 78 |
| 29. Informe descargado (evaluación criterios) . . . . .                               | 79 |
| 30. Informe descargado (anotaciones realizadas) . . . . .                             | 79 |

# Índice de tablas

|   |    |
|---|----|
| 2.1. Estimación de dedicaciones en horas . . . . .              | 8  |
| 7.1. Estimación de dedicaciones en horas MBK . . . . .          | 52 |
| 7.2. Estimación de dedicaciones en horas MBK . . . . .          | 53 |
| 7.3. Estimación de dedicaciones en horas MMR . . . . .          | 54 |
| 7.4. Estimación de dedicaciones en horas CMR . . . . .          | 57 |
| 8.1. Desviaciones en la estimación de horas definidas . . . . . | 60 |

# Introducción

La tecnología ha causado una completa revolución mundial en muchos ámbitos. Antes se acostumbraba a tener los documentos en papel, ahora el formato estándar es el PDF. Esta revolución ha permitido una enorme serie de facilidades a la hora de leer, analizar y procesar los documentos.

En el dominio de la revisión de artículos es necesario evaluar una serie de criterios para valorar el manuscrito. Sin embargo, los revisores carecen de una herramienta que tenga el propósito de facilitar este proceso[4].

Tecnologías como la anotación web han resultado ser clave a la hora de analizar documentos digitales. Además, gracias a las Líneas de Producto Software (LPS), se ha facilitado el desarrollo de diversas herramientas personalizadas para resolver problemas específicos de un mismo dominio como el de las anotaciones web.

El objetivo de este Trabajo Fin de Grado (TFG en adelante) es desarrollar una herramienta que facilite la revisión de artículos mediante la extensión de una LPS. En este documento se realizará un análisis de la LPS a extender, WACline, y se explicarán las modificaciones realizadas y las características añadidas para generar el producto resultante. Para entender mejor el proyecto y algunos de los conceptos que van a ir apareciendo referentes a la herramienta, **se recomienda vivamente la visualización del vídeo demo**. Se encuentra disponible en YouTube en el siguiente enlace: <https://tiny.cc/demoReviewAndGoPlus>



# Planificación inicial

Este capítulo recoge la planificación inicial elaborada para un correcto desarrollo del proyecto. Se comienza dando contexto con los antecedentes. Se continúa describiendo los objetivos y el alcance del proyecto. Más tarde, se describe la estructura de descomposición del trabajo (EDT) junto a las estimaciones. Para finalizar se presentan las herramientas necesarias para la realización del proyecto, los interesados del mismo y el plan de gestión de riesgos.

## 2.1. Antecedentes

Los manuscritos siguen una serie de procesos antes de ser publicados. La revisión por pares es una de las fases fundamentales por las que pasa un manuscrito con el objetivo de mejorarlo. Durante este proceso se debe prestar atención a muchos aspectos del artículo como la originalidad o la relevancia del mismo. Además, es interesante que como resultado de este proceso el autor reciba comentarios constructivos por parte del revisor[4].

En la revisión por pares, el revisor sigue un criterio para analizar el artículo y darlo por válido. Sin embargo, uno de los problemas que se da, es que el revisor se inventa su propio criterio. La solución, es tener unos estándares que describan cuales son los criterios con los que se va a evaluar[5]. De esta manera el autor podrá conocerlos de antemano y elaborar un artículo que se ajuste mejor a ellos. Como solución el '*Empirical Standard*'<sup>1</sup> describe criterios para distintos tipos de artículos. Los métodos ahí descritos serán los que el revisor podrá elegir a la hora de realizar la revisión.

El objetivo de este Trabajo de Fin de Grado (TFG en adelante) es desarrollar una extensión que facilite este proceso de revisión de artículos. Gracias a esta extensión los revisores podrán utilizar los criterios descritos en el '*Empirical Standard*'. Además, podrán descargar un informe que resuma la revisión que han realizado, que contenga los comentarios realizados y los criterios empleados.

La extensión se realizará partiendo de WACline <sup>2</sup>, una plataforma de Línea de Produc-

---

<sup>1</sup>Empirical Standard:<https://github.com/acmsigsoft/EmpiricalStandards>

<sup>2</sup>WACline:<https://github.com/onekin/WacLine>

tos Software (LPS) especializada en anotaciones web<sup>3</sup>. Esta LPS ha sido desarrollada por Haritz Medina, parte del grupo de investigación ONEKIN<sup>4</sup> de la Universidad del País Vasco (UPV/EHU).

La LPS cuenta con un producto, *Review&Go*<sup>5</sup>, cuyo objetivo es la revisión de artículos de investigación. Esta herramienta además de realizar anotaciones permite añadirles comentarios y sugerencias de bibliografía haciendo uso de la API del repositorio bibliográfico DBLP<sup>6</sup>. Durante el proceso de anotación del documento, ofrece una vista que muestra las anotaciones realizadas hasta el momento y una opción que permite descargar un informe de la valoración realizada.

Sin embargo, esta herramienta carece de funcionalidades que den solución al problema de evaluación de criterios que se ha mencionado. Es por eso que se va a realizar una nueva versión que mejore *Review&Go*, descartando y modificando algunas de sus características y desarrollando nuevas. A esta nueva versión se le denominará *Review&Go+*.

### 2.2. Motivación

En tercero cursé la asignatura de Desarrollo Industrial de Software (DIS). En esta asignatura se estudiaban las líneas de producto software, partiendo de aprender la problemática de la reutilización en el software y la generación de código, hasta llegar a desarrollar una pequeña línea de productos software.

Gracias a esta asignatura, se llegó a comprender el potencial que tiene este tipo de desarrollo software frente a los métodos convencionales. Sin embargo, no se llegaba a poder profundizar del todo en el tema; hacía falta poner en práctica el conocimiento adquirido en un proyecto mayor.

A mediados del verano pasado Óscar contactó conmigo, organizamos una reunión y me presentó la idea de poder trabajar en el proyecto WACline para llevar a cabo este TFG. Me pareció una excelente oportunidad para poder poner en práctica lo aprendido en DIS y ampliar aún más mis conocimientos; no dude en aceptar la oferta.

Además, la idea de poder aprender acerca del desarrollo de extensiones para el navegador resultaba muy atractiva, ya que a pesar de haber utilizado alguna, nunca había investigado acerca de su estructura y desarrollo.

### 2.3. Objetivos

El proyecto tiene como fin principal el desarrollo de una herramienta que facilite la revisión de artículos mediante anotaciones web partiendo de la línea de productos *WACline*. Para completar este objetivo :

- Familiarizarse con los conceptos de Líneas de Producto

---

<sup>3</sup>Anotaciones web:<https://www.w3.org/TR/annotation-model/>

<sup>4</sup>ONEKIN:<https://onekin.org/>

<sup>5</sup>Review&Go:<https://chrome.google.com/webstore/detail/reviewgo/hgiannlbfceoomjmcgedbnkfeblbcogi>

<sup>6</sup>DBLP:<https://dblp.org/>

- Aprender JavaScript y *pure::variants*<sup>7</sup>.
- Familiarizarse con la línea de producto WACline
- Familiarizarse con el dominio: revisión de manuscritos
- Aprender acerca del desarrollo de extensiones para el navegador.
- Hacer uso de una API para obtener información de los autores de los artículos evaluados.
- Desarrollar el producto a partir de WACline

## 2.4. Alcance

En este proyecto se pretende ampliar las funcionalidades del producto Review&Go, cuyo objetivo es proporcionar una herramienta para la revisión de artículos científicos. Se ha establecido que se desarrollarán tres módulos que facilitarán la evaluación a los revisores, proporcionando listas de criterios que el artículo debe cumplir e información, que se le proporciona al revisor, acerca de los autores.

Queda excluido del alcance la evaluación del producto con usuarios ajenos al proyecto que realicen revisiones de artículos con frecuencia. Sin embargo, no se excluye que usuarios que no son revisores prueben el producto para evaluar si es intuitivo o no.

## 2.5. Estructura de descomposición del trabajo (EDT)

Para el desarrollo de este proyecto se ha decidido utilizar un ciclo de vida iterativo e incremental. En este ciclo de vida, se organizan iteraciones en las que se repiten las actividades del proyecto. Al final de cada iteración se habrá completado un entregable, o un conjunto de entregables, y el producto final será la acumulación de funcionalidades construida durante las iteraciones. Además de añadir nuevas funcionalidades, el objetivo de las iteraciones es mejorar el producto en calidad al anterior.

Gracias a este ciclo de vida, se podrán ir observando los cambios en los módulos y probar su correcto funcionamiento. Por eso, durante el proyecto se organizaran reuniones para probar el producto y comprobar que se van alcanzando los objetivos descritos. Se considerará una iteración para cada módulo a implementar con una estimación de alrededor de dos semanas para cada una de ellas. Las iteraciones concluirán con una demo del producto actual y una presentación en profundidad de los requisitos del módulo a implementar en la siguiente iteración para poder analizar los requisitos y planificarla. Otro de los motivos por los que se ha elegido este ciclo de vida, es porque a medida que se vaya trabajando en el proyecto se irán adquiriendo conocimientos que podrían ser útiles para mejorar características ya desarrolladas.

A continuación, en la figura 2.1 se describen los paquetes de trabajo que componen el proyecto y sus correspondientes tareas.

---

<sup>7</sup>pure::variants:<https://www.pure-systems.com/>

## 2. PLANIFICACIÓN INICIAL

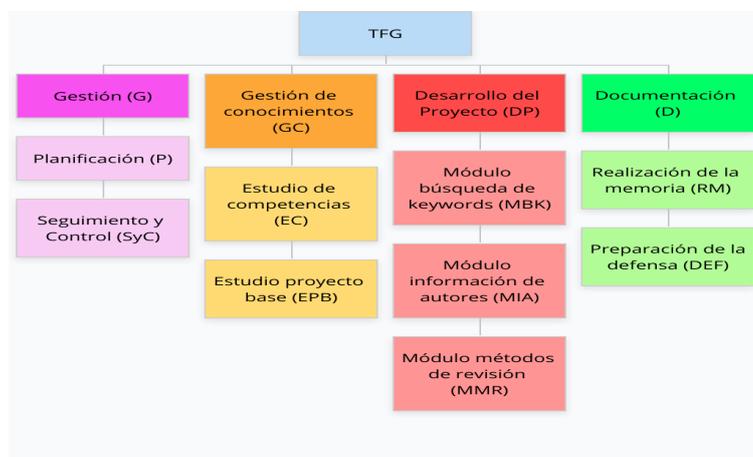


Figura 2.1: Diagrama E.D.T.

### 2.5.1. Fase de Gestión (G)

El paquete de Trabajo de **Planificación (P)** incluirá las tareas de planificación inicial así como aquellas que -en su caso- fuera necesario realizar para mantener una planificación adecuada.

- P.T1. Tareas relacionadas con la identificación de requisitos, toma de decisiones iniciales, análisis de información y resolución de dudas.
- P.T2 Planificación inicial orientada a la preparación del entorno de desarrollo del proyecto.
- P.T3 Actualización, si fuera necesaria, de la Planificación.

El paquete de Trabajo **Seguimiento y Control (SyC)** agrupará las tareas necesarias para garantizar el adecuado desarrollo del proyecto y, en particular, el seguimiento de dedicaciones, cumplimiento de plazos y especificaciones.

- SyC.T1. Reuniones de seguimiento para revisar el estado del proyecto.
- SyC.T2. Configuración de tablero Trello para recopilación de dedicaciones
- SyC.T3. Recogida de Información relevante sobre el desarrollo del proyecto.
- SyC.T4. Contraste de seguimiento con los planes, identificación de desviaciones significativas y de riesgos emergentes.

### 2.5.2. Fase de Estudio de Competencias (EC)

El paquete de Trabajo **Estudio de Competencias (EC)** agrupa las tareas de incremento de conocimiento de las tecnologías necesarias para el desarrollo de este proyecto. Dichas tareas son:

- EC.T1 Conocer y aprender a utilizar el entorno de desarrollo de la herramienta pure::variants.
- EC.T2. Conocer y adquirir conocimientos acerca de la arquitectura de las extensiones de Chrome.
- EC.T3. Refrescar conocimientos sobre el lenguaje JavaScript en su versión ES6 y aprender sobre la programación orientada a eventos.

El paquete de Trabajo ***Estudio de proyecto base (EPB)*** agrupa las tareas de análisis y estudio del proyecto sobre el que se va a trabajar, WACline. Dichas tareas son:

- EPB.T1. Análisis de la estructura del proyecto base, LPS WACline para entender su arquitectura y funcionamiento.
- EPB.T2. Resolución dudas acerca del proyecto base.
- EPB.T3. Resolución (e implementación) de una de las issues asignadas a la LPS WACline para adquirir soltura en el proyecto.

### 2.5.3. Fase de Desarrollo del Proyecto (DP)

Las tareas correspondientes a los siguientes paquetes de trabajo se definirán al comienzo de la iteración correspondiente a cada uno de ellos.

El paquete de Trabajo ***Módulo búsqueda de keywords (MBK)*** agrupa las tareas correspondientes para el desarrollo del módulo encargado de la búsqueda y anotación de palabras clave en los documentos.

El paquete de Trabajo ***Módulo información de autores (MIA)*** agrupa las tareas correspondientes para el desarrollo del módulo encargado de la búsqueda de información de autores de los documentos.

El paquete de Trabajo ***Módulo métodos de revisión (MMR)*** agrupa las tareas correspondientes para el desarrollo del módulo encargado de la selección de plantillas de revisión de artículos y la validación de dichas plantillas.

### 2.5.4. Fase de Documentación (D)

El paquete de Trabajo ***Realización de la memoria (RM)*** agrupa las tareas correspondientes para la realización de la memoria del trabajo de fin de grado. Dichas tareas son:

- RM.T1. Configuración del entorno LaTeX para el desarrollo de la memoria.
- RM.T2. Redacción de la memoria.

El paquete de Trabajo ***Preparación de la defensa (DEF)*** agrupa las tareas correspondientes para la preparación de la defensa del trabajo de fin de grado. Dichas tareas son:

- DEF.T1. Preparar el póster del proyecto para exhibirlo en la facultad.
- DEF.T2. Preparar el documento de apoyo para la presentación.
- DEF.T3. Preparar la defensa.

## 2.6. Estimación de dedicaciones

De acuerdo con los paquetes de trabajo definidos en el EDT se definen las estimaciones en la tabla 2.1. En esta tabla no constan las estimaciones realizadas para las tareas de los módulos de desarrollo, puesto que el análisis de las tareas junto a sus estimaciones y planificación se realiza al comienzo de cada iteración.

| Tareas   | Estimación en horas |
|--|---------------------|
| <b>Proyecto Fin de Grado</b>                       | <b>310</b>          |
| Fase de gestión (G)                                | 48                  |
| Planificación (P)                                  | 18                  |
| P.T1 Requisitos                                    | 3                   |
| P.T2 Entorno de desarrollo y planificación inicial | 12                  |
| P.T3 Actualización de la planificación             | 3                   |
| Seguimiento y control (SyC)                        | 30                  |
| SyC.T1 Reuniones                                   | 15                  |
| SyC.T2 Configuración Trello                        | 2                   |
| SyC.T3 Recogida de información del desarrollo      | 7                   |
| SyC.T4 Contraste de seguimiento                    | 6                   |
| Fase de gestión de conocimientos (GC)              | 37                  |
| Estudio de competencias (EC)                       | 15                  |
| EC.T1 pure::variants                               | 7                   |
| EC.T2 Extensiones Chrome                           | 5                   |
| EC.T3 JavaScript ES6                               | 3                   |
| Estudio de proyecto base (EPB)                     | 22                  |
| EPB.T1 Análisis SPL WACline                        | 15                  |
| EPB.T1 Dudas                                       | 3                   |
| EPB.T1 Solucionar issue 88                         | 4                   |
| Fase de desarrollo (DP)                            | 140                 |
| Fase de documentación (D)                          | 85                  |
| Realización de la memoria (RM)                     | 70                  |
| RM.T1 Configurar entorno LaTeX                     | 3                   |
| RM.T2 Redactar                                     | 67                  |
| Preparación de la defensa (DEF)                    | 15                  |
| DEF.T1 Póster                                      | 2                   |
| DEF.T2 Documento de apoyo                          | 5                   |
| DEF.T3 Preparar                                    | 8                   |

Tabla 2.1: Estimación de dedicaciones en horas

## 2.7. Diagrama de Gantt

En las figuras 2.2 y 2.3 se muestra el diagrama de Gantt correspondiente a los paquetes de trabajo y tareas del diagrama EDT del proyecto. Puesto que en la fase inicial del proyecto no se conocen los requisitos específicos de los módulos que se iban a implementar, no se mencionan sus tareas.

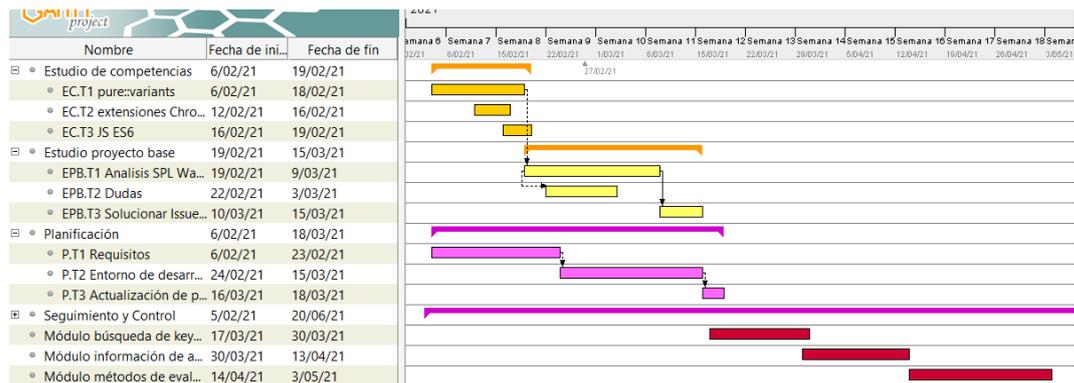


Figura 2.2: Diagrama Gantt inicial - Primera parte

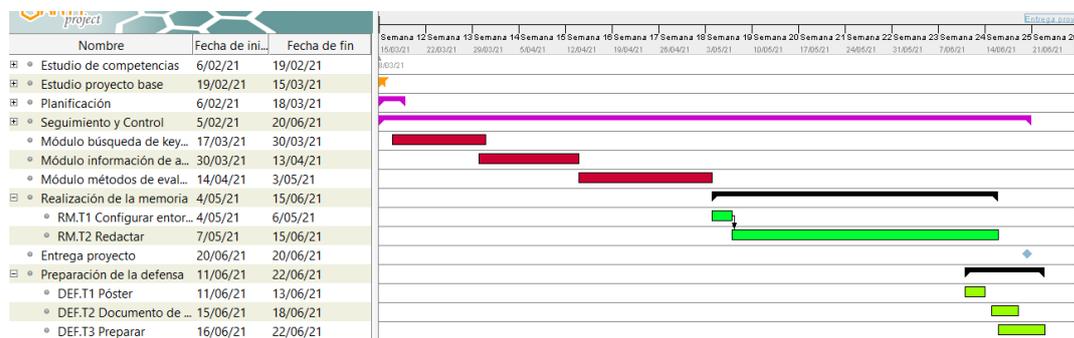


Figura 2.3: Diagrama Gantt inicial- Segunda parte

## 2.8. Herramientas

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

### 2.8.1. Trello

Trello<sup>8</sup> es un software de gestión de tareas muy útil para la gestión de proyectos. Se utilizará esta herramienta para tener un control de las tareas por hacer, las acabadas y las que están en curso además de un registro de dedicaciones.

<sup>8</sup>Trello:<https://trello.com/>

### 2.8.2. Git, Github y GitKraken

Git<sup>9</sup> es un software de control de versiones. Ayuda a gestionar los cambios en el código en el tiempo y resulta muy útil para desarrollos en equipo.

Github<sup>10</sup> es una plataforma que pertenece a Microsoft que permite alojar proyectos en la nube mediante Git.

GitKraken<sup>11</sup> es un interfaz gráfica de escritorio para git desarrollada por Electron. Se utilizará esta herramienta para tener una visión más clara del repositorio del proyecto.

### 2.8.3. Visual Studio Code

Visual Studio Code<sup>12</sup> también conocido como VS Code, es un editor de código fuente open source desarrollado por Microsoft. Este será el editor de código a utilizar para el proyecto debido a las facilidades que aporta para navegar dentro del proyecto y modificar código.

### 2.8.4. Google Chrome

Google Chrome<sup>13</sup> es el navegador web de Google.

Se utilizará para realizar las pruebas de la extensión que se va a desarrollar.

### 2.8.5. pure::variants y Eclipse

Pure::variants<sup>14</sup> es un software que provee herramientas para ayudar en cada una de las fases del desarrollo de una línea de productos. Este software es un plug-in desarrollado para el entorno de programación Eclipse<sup>15</sup>

### 2.8.6. Node.js y ECMAScript

Node.js<sup>16</sup> es un entorno para la capa del servidor basado en el lenguaje de programación JavaScript orientado a eventos.

ECMAScript es una especificación de lenguaje de programación publicada por ECMA International. Se utilizará la versión ES6 para la programación del proyecto

### 2.8.7. LaTeX y Oveleaf

LaTeX<sup>17</sup> es un sistema de composición de textos que se caracteriza por las facilidades que provee para la producción de documentos técnicos y científicos. Esta tecnología será

---

<sup>9</sup>Git:<https://git-scm.com/>

<sup>10</sup>Github:<https://github.com/>

<sup>11</sup>GitKraken:<https://www.gitkraken.com/>

<sup>12</sup>Visual Studio Code:<https://code.visualstudio.com/>

<sup>13</sup>Google 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/>

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

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

<sup>17</sup>LaTeX:<https://www.latex-project.org/>

utilizada para el desarrollo de la memoria por las facilidades de estructuración y modularización que ofrece. Además se utilizará el editor online Overleaf<sup>18</sup> que nos facilita el acceso desde cualquier dispositivo y copias de seguridad.

### 2.8.8. Figma

Figma<sup>19</sup> es un editor de gráficos vectorial y una herramienta de generación de prototipos.

### 2.8.9. Bizagi Modeler

Bizagi Modeler<sup>20</sup> es una herramienta utilizada para modelar procesos que soporta el estándar BPMN (Business Process Model and Notation). Esta herramienta será utilizada para realizar diagramas de flujo.

### 2.8.10. Postman

Postman<sup>21</sup> es un entorno de desarrollo de APIs que permite diseñar, probar y monitorizar servicios REST. Esta plataforma se utilizará para la prueba de APIs externas.

### 2.8.11. Google Drive

Google Drive<sup>22</sup> es un servicio de alojamiento de archivos de Google. Esta herramienta se utilizará para realizar copias de seguridad de la documentación en la nube.

## 2.9. Interesados

Entre los interesados se encuentra Iñigo Bereciartua, autor del proyecto. Se encargará de que el proyecto cumpla con los objetivos definidos siguiendo la planificación de la manera más fiel posible para concluir el proyecto con éxito.

El director del proyecto es Óscar Díaz, quien se encargará de guiar al alumno a llevar a cabo el proyecto correctamente.

Otro de los interesados es Haritz Medina, desarrollador de la línea de productos. El producto resultante de este proyecto extenderá su LPS. Por ello, él estará presente en las reuniones para la toma de decisiones que la involucren y ayudar al alumno en los primeros pasos de integrarse en el proyecto.

Para completar la lista de interesados se encuentra el tribunal, encargado de evaluar el trabajo y la defensa del proyecto.

## 2.10. Gestión de riesgos

Durante el proceso de desarrollo del proyecto pueden darse circunstancias que entorpezcan o impidan el que se cumpla la planificación inicial. Por ello es importante que se

---

<sup>18</sup>Overleaf:<http://overleaf.com/>

<sup>19</sup>Figma:<https://www.figma.com/>

<sup>20</sup>Bizagi Modeler:<https://www.bizagi.com/es/plataforma/modeler>

<sup>21</sup>Postman:<https://www.postman.com/>

<sup>22</sup>Google Drive:[https://www.google.com/intl/es\\_es/drive/](https://www.google.com/intl/es_es/drive/)

identifiquen dichos riesgos y halla un plan de mitigación para que el proyecto tenga éxito. A continuación se listan los posibles riesgos identificados:

### 2.10.1. Dificultad para entender el proyecto base

Debido a que se va a tener que extender la LPS WACline es importante entender bien el proyecto y adaptarse a la estructura y notación del mismo.

- **Probabilidad:** media
- **Impacto:** alto
- **Consecuencias:** retraso en la implementación de los módulos con probabilidad de incumplir con los plazos inicialmente establecidos.
- **Prevención:** dedicar tiempo al comienzo del proyecto para entender el funcionamiento y no dejar dudas sin aclarar.
- **Plan de mitigación:** continuar con el desarrollo de tareas que no dependan del problema surgido y continuar estudiando el problema para darle una solución consultando las dudas en caso de que impliquen decisiones que se puedan tomar a futuro (como futuras características o modificaciones ajenas a este proyecto). De ser necesario, también habrá que replanificar.

### 2.10.2. Problemas con las tecnologías utilizadas

Algunas de las tecnologías que se utilizarán a lo largo del proyecto son desconocidas, lo cual puede llegar a ser un problema.

- **Probabilidad:** media
- **Impacto:** medio
- **Consecuencias:** retraso con los plazos establecidos.
- **Prevención:** invertir tiempo en estudiar el uso de las tecnologías con las que se tiene menos experiencia o se desconocen.
- **Plan de mitigación:** continuar con el desarrollo de otras tareas que no sean dependientes de la tecnología y analizar las dudas de la tecnología que da problemas.

### 2.10.3. Planificación incorrecta

Es importante realizar una planificación realista y adecuada para que el proyecto tenga éxito y no resulte una pesadilla.

- **Probabilidad:** media
- **Impacto:** alto
- **Consecuencias:** retraso con los plazos establecidos.

- **Prevención:** Realizar una planificación realista al comienzo analizando las tareas a realizar durante el proyecto.
- **Plan de mitigación:** replanificar intentando realizar una planificación más precisa probablemente teniendo que invertir más horas de las estimadas en un principio.

#### 2.10.4. Problemas de índole personal

Es un proyecto de una duración larga en comparación con otros proyectos realizados hasta ahora. Durante su transcurso puede que ocurran problemas inesperados tanto al director como al alumno que pueden afectar en el desarrollo del proyecto.

- **Probabilidad:** baja
- **Impacto:** medio
- **Consecuencias:** retraso con los plazos establecidos.
- **Prevención:** pueden darse problemas que no se pueden prever.
- **Plan de mitigación:** replanificar y tratar de continuar con el desarrollo de manera normal lo antes posible.

#### 2.10.5. Pérdida de información

Puede darse el caso de que por fallos en el sistema informático se produzcan pérdidas de información durante el transcurso del proyecto.

- **Probabilidad:** baja
- **Impacto:** medio
- **Consecuencias:** retraso con los plazos establecidos.
- **Prevención:** utilizar copias de seguridad y actualizarlas diariamente para reducir la cantidad de información perdida y utilizar servicios con alojamiento en la nube
- **Plan de mitigación:** tratar de recuperar la información perdida y en caso de que no sea posible rehacer el proceso para volver a generar/obtener la información perdida.

### 2.11. Plan de calidad

Para asegurar la calidad del proyecto en su totalidad hay que asegurar la calidad tanto de la memoria como la del código del producto.

#### 2.11.1. Memoria

- **Extensión:** En la plataforma ADDI<sup>23</sup> se han observado las memorias publicadas y se ha estimado que la extensión conveniente se encuentra entre unas 80-110 páginas.

---

<sup>23</sup>ADDI:<https://addi.ehu.es/handle/10810/13140>

- **Estructura:** Para la redacción de la memoria se seguirá un modelo proporcionado en la página web de la universidad<sup>24</sup>. Además, se indican algunos consejos que se tendrán en cuenta para obtener una calidad mínima.

### 2.11.2. Producto

- **Mantenibilidad:** Dado que se parte de un proyecto base es importante seguir con la nomenclatura del mismo para asegurar la calidad del código del producto que se va a generar.
- **Funcionalidad:** Además de cumplir con su propia funcionalidad, será importante asegurar que el nuevo código no interfiere en el funcionamiento de las características que ya se encuentran en la LPS.
- **Usabilidad:** El producto debe poder ser utilizado en el navegador Google Chrome.

---

<sup>24</sup><https://www.ehu.es/es/web/informatika-fakultatea/ikasketak/gradu-amaierako-proiektua>

## Análisis de requisitos

A continuación se presentarán los requisitos funcionales y no funcionales.

### 3.1. Requisitos funcionales

Los nuevos módulos implementados en este TFG ofrecerán las siguientes principales características:

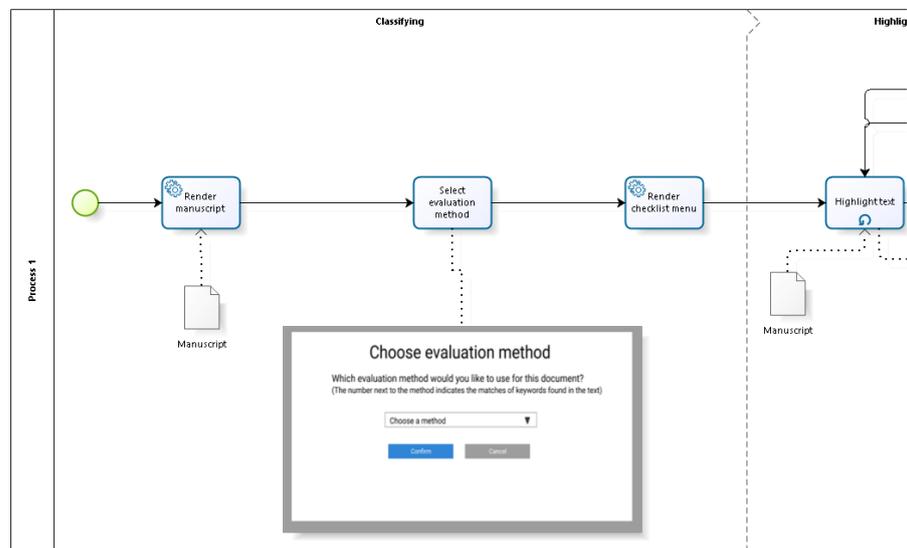
- **Información de autores:** El usuario podrá escoger el congreso al que pertenece el artículo que va a revisar. Una vez el usuario ha anotado los autores, podrá visualizar información acerca de ellos. El sistema mostrará el número de publicaciones que ha realizado el autor para ese congreso en los últimos años.
- **Métodos de revisión:** El usuario podrá elegir métodos para la revisión de artículos. El sistema buscará las palabras clave en el documento para cada método de revisión. El sistema deberá mostrar al lado de cada método de revisión el número de coincidencias de sus palabras clave a la hora de escoger un nuevo método de revisión. El sistema deberá importar los criterios que seleccione el usuario de los correspondientes al método de revisión escogido. El usuario podrá marcar como aprobado o suspendido cada criterio de los métodos de revisión.
- **Editar informe:** El sistema deberá poder mostrar en pantalla un informe de la evaluación realizada. El usuario podrá realizar cambios en el informe. El usuario podrá guardar los cambios realizados en el informe y descargar una copia. El sistema deberá mostrar la lista de críticas no válidas, correspondiente a cada lista de criterios utilizada durante la revisión, para que el usuario compruebe que no incumple ninguna.

Una vez implementados los módulos, el usuario podrá realizar el proceso de revisión. A continuación se describirá el flujo que se seguirá el proceso de revisión de los criterios, este incluye los prototipos iniciales realizados para la aplicación.

El proceso comienza en la etapa de clasificación, tal y como refleja la figura 3.1. Una vez el documento ha sido cargado, el revisor procederá a escoger métodos de revisión. El

### 3. ANÁLISIS DE REQUISITOS

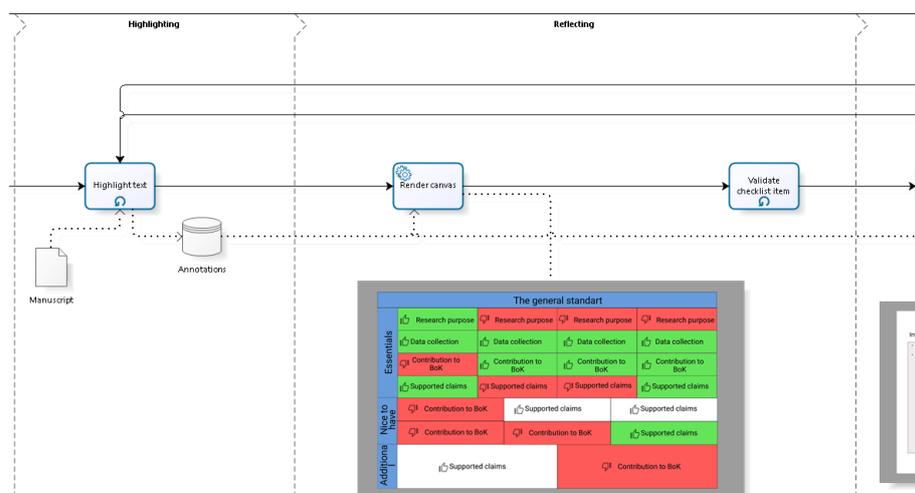
sistema procesará los nuevos criterios en el menú lateral para que el revisor pueda proceder a realizar anotaciones sobre ellos.



**Figura 3.1:** Diagrama proceso de revisión de manuscrito (Clasificación)

A continuación, se avanzará a la etapa de subrayado o anotación. Como refleja la figura 3.2, en esta etapa el revisor realizará las anotaciones y comentarios que sean necesarios. Una vez ha finalizado procederá a la evaluación de los criterios que previamente ha seleccionado.

Comienza la etapa para reflejar la valoración que se va a realizar sobre el manuscrito. Para ello se accederá al canvas que muestra los criterios y se irán evaluando como aprobados o suspendidos. Cuando el revisor ha finalizado con la evaluación se procederá al último paso.



**Figura 3.2:** Diagrama proceso de revisión de manuscrito (Anotación y evaluación)

Este paso consiste en el informe final. El revisor accederá al menú del informe, como refleja la figura 3.3, y podrá realizar modificaciones en el informe que se ha generado a

partir de las anotaciones que ha realizado. El revisor tendrá que cumplir en su valoración con las críticas inválidas correspondientes a las listas de criterios que haya utilizado. en caso de que el revisor vea necesario tanto añadir como eliminar anotaciones volverá a la segunda etapa para realizar las modificaciones y continuara con el proceso que se ha explicado. Una vez haya finalizado con la revisión del informe, procederá a descargarse el informe para poder enviarle los comentarios realizados al autor del manuscrito. Ahora el revisor podrá decidir si desea dar por finalizado el proceso al estar de acuerdo con el informe generado, o si por el contrario desea realizar modificaciones en las anotaciones y repetir estos tres últimos pasos.

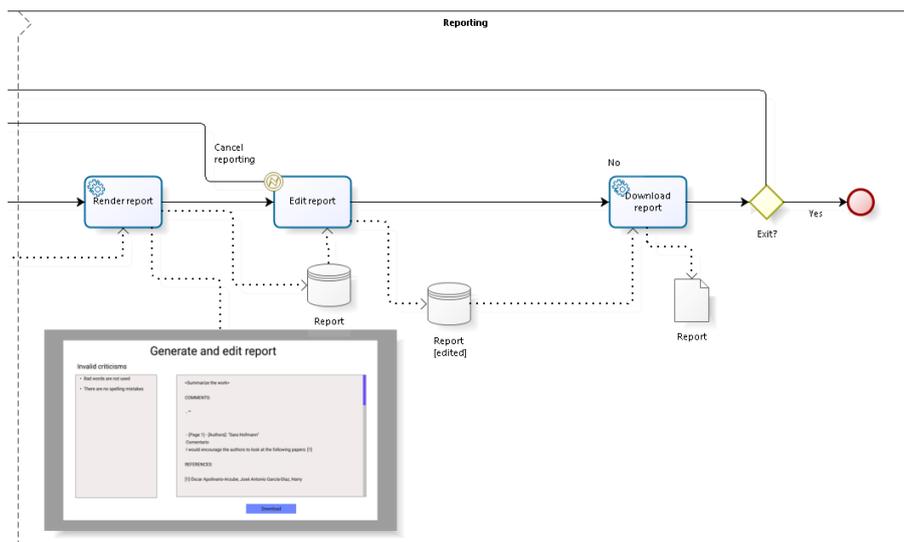


Figura 3.3: Diagrama proceso de revisión de manuscrito (Informe)

## 3.2. Requisitos no funcionales

Para cumplir con el objetivo de integrar correctamente las nuevas características a la LPS ya existente, WACline, hay que definir correctamente los puntos de variación. Además, los cambios realizados en las características ya existentes no deben interferir en el funcionamiento esperado de la LPS. Es importante respetar la modularidad y la integridad de los módulos que se van a desarrollar y asegurar la compatibilidad de los nuevos módulos con las características ya existentes.

También habrá que definir correctamente las dependencias y exclusiones de cada característica implementada para no generar conflictos.

## 3.3. Casos de uso

A continuación, en la figura 3.4, se presenta el modelo de casos de uso del producto *Review&Go+*. Se realizará una breve descripción de cada uno de los casos de uso.

Se han utilizado dos colores para realizar el diagrama. Los casos de uso en **naranja** representan los casos de uso que hay que implementar. Mientras que los que están en **negro**, representan casos de uso ya implementados en la LPS WACline.

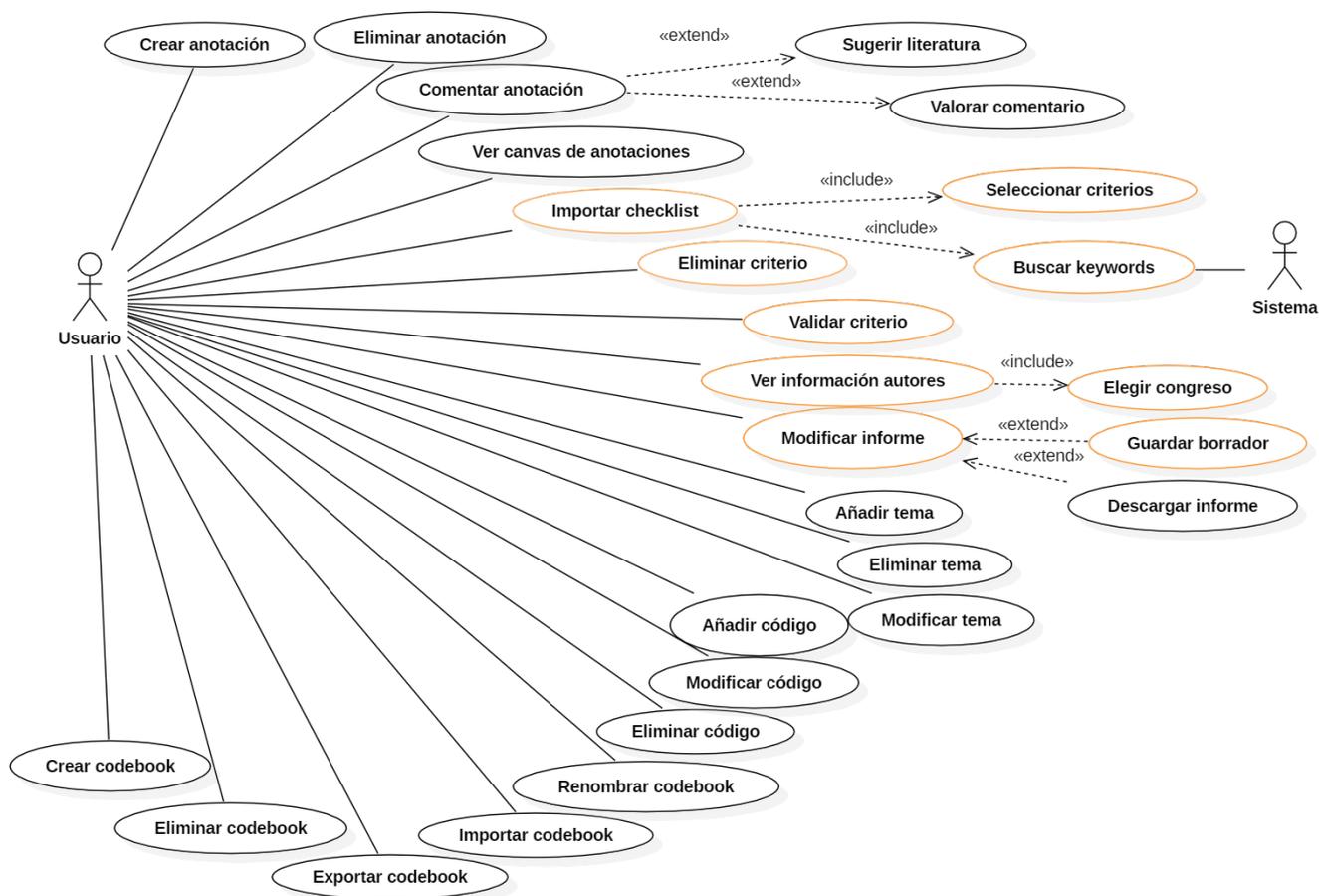


Figura 3.4: Diagrama de casos de uso Review&Go+

- **Crear anotación:** El usuario puede seleccionar texto del documento y anotar en uno de los temas o códigos
- **Comentar anotación:** El usuario puede realizar comentarios en las anotaciones realizadas.
- **Sugerir literatura:** El usuario puede añadir a la anotación un enlace bibliográfico del DBLP.
- **Valorar comentario:** El usuario puede clasificar las anotaciones en tres categorías: 'Minor weakness', 'Major weakness' y 'Strength'.
- **Eliminar anotación:** El usuario puede eliminar las anotaciones que ha realizado.
- **Importar checklist:** El usuario podrá importar listas de criterios definidas en el 'Empirical Standard'.
- **Seleccionar criterios:** Una vez el usuario ha seleccionado que lista quiere importar, deberá seleccionar qué criterios de dicha lista desea importar.

- **Buscar keywords:** Cuando el usuario vaya a seleccionar una lista de criterios, se le mostrarán el número de palabras clave encontradas por método de revisión.
- **Validar criterio:** El usuario podrá evaluar los criterios como aprobados, suspendidos o no evaluados.
- **Eliminar criterio:** El usuario podrá eliminar criterios que haya importado.
- **Ver información de autores:** El usuario podrá ver la información de los autores que haya anotado.
- **Elegir congreso:** El usuario deberá seleccionar el congreso al que pertenece el artículo para ver la información de los autores.
- **Modificar informe:** El usuario podrá modificar el informe generado por la aplicación
- **Guardar borrador:** El usuario podrá guardar un borrador del informe para continuar editándolo más tarde.
- **Descargar informe:** El usuario podrá descargar el informe que ha modificado.
- **Añadir tema:** El usuario podrá añadir un nuevo tema al codebook.
- **Eliminar tema:** El usuario podrá eliminar los temas que ha añadido
- **Modificar tema:** El usuario podrá renombrar los temas del codebook.
- **Añadir código:** El usuario podrá añadir nuevos códigos a los temas del codebook.
- **Modificar código:** El usuario podrá renombrar los códigos de los temas.
- **Eliminar código:** El usuario podrá eliminar los códigos de los temas.
- **Crear codebook:** El usuario podrá crear un nuevo codebook.
- **Eliminar codebook:** El usuario podrá eliminar los codebooks que ha creado.
- **Renombrar codebook:** El usuario podrá renombrar los codebooks.
- **Exportar codebook:** El usuario podrá exportar los codebooks en formato JSON.
- **Importar codebook:** El usuario podrá importar codebooks que hayan sido generados por la aplicación.



## Entorno

Para ubicarse en el contexto de este proyecto es necesario entender dos cosas. La primera, qué es una Línea de Producto Software y cómo funciona la herramienta `pure::variants`. La segunda, el funcionamiento de las extensiones del navegador Chrome.

### 4.1. Líneas de Producto Software (LPS)

Las líneas de producto software o LPS son un conjunto de productos producidos a partir de una misma **plataforma de la LPS** [6].

La plataforma de la LPS incluye entre otros la arquitectura, los componentes reutilizables, modelos de dominio, requerimientos y la documentación.

La base de las líneas de producto es la reutilización de características o componentes comunes que tienen los distintos productos que la conforman. Este método es comparable a los métodos que se utilizan en otras industrias. En la automovilística, por ejemplo, cuando un cliente desea comprar un coche se parte de una base común y después el cliente tendrá la opción de elegir algunas partes como el motor, la pintura, las ruedas, etc.

Este método de desarrollo software tiene muchas ventajas con respecto a los métodos tradicionales.

- **Velocidad de entrega:** Una vez establecida la plataforma de la LPS, la entrega de los sistemas se produce de una manera más rápida y económica.
- **Escalabilidad:** El desarrollo basado en líneas de producto permite una mayor escalabilidad del modelo de negocio ya que se pueden generar una gran variedad de productos que cubran distintas necesidades sobre un mismo dominio.
- **Mantenibilidad:** La entrega de productos se produce con un menor número de fallos que en el desarrollo de productos software tradicional, puesto que la solución de errores se realiza para todo el conjunto de productos de manera simultánea.
- **Modularidad:** El desarrollo de software en base a características favorecen la modularidad del mismo

- **Personalización:** Gracias a este tipo de desarrollo, se pueden generar productos específicos con un menor coste que se ajusten a las necesidades de un cliente.

Según los beneficios mencionados, parece que el desarrollo de líneas de productos es una mejor elección que el desarrollo tradicional. Sin embargo, como se puede apreciar en la figura 4.1 este desarrollo tiene un coste inicial mucho más alto que se compensará cuando se alcance un número aproximado de sistemas o productos. Al comienzo, se debe realizar un análisis exhaustivo del dominio y valorar si está justificado el uso de un desarrollo basado en líneas de producto.

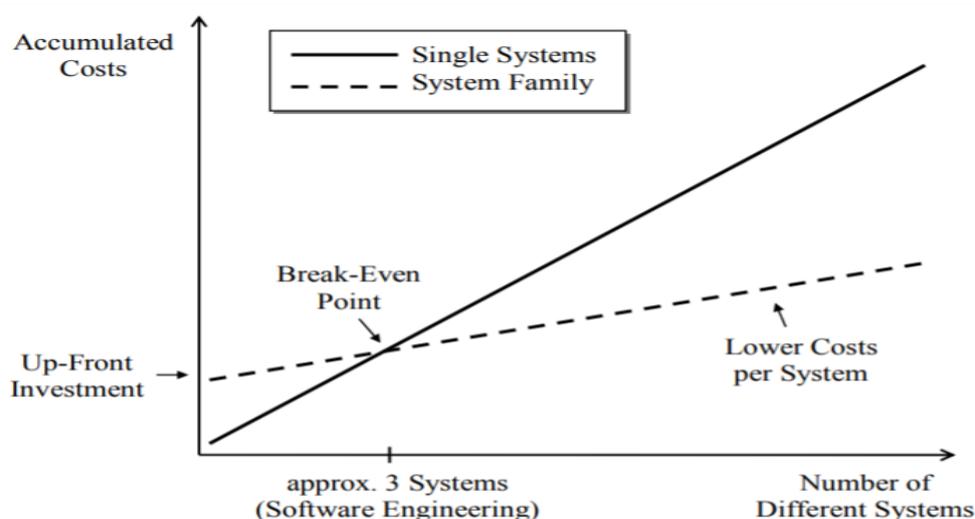


Figura 4.1: Coste desarrollo LPS[1]

La ingeniería de líneas de producto software se divide en dos procesos[1] que, como se puede observar en la figura 4.2, interactúan para generar el producto final. Estos son los dos procesos:

- **Ingeniería de dominio:** Es el proceso en el que se definen la parte común y la variabilidad de la línea de productos. Se produce el análisis de requisitos para toda la LPS.
- **Ingeniería de producto:** Es el proceso en el que los productos de la línea se construyen haciendo uso de la plataforma de la línea y explotando la variabilidad de la línea de productos.

Los características que diferencian unos productos de otros son definidas como puntos de variación. A la hora de crear un producto se podrá escoger entre las variantes que ofrecen los puntos de variación. Volviendo al ejemplo de la industria automovilística, el cliente elige el color del coche, este puede ser rojo, azul, negro, blanco, etc. Aquí el punto de variación sería el color y las variantes las distintas opciones de colores para elegir.

Para definir los puntos de variación de una línea de productos software se hace uso del **diagrama de características**. En él, aparecerán las características básicas que contienen

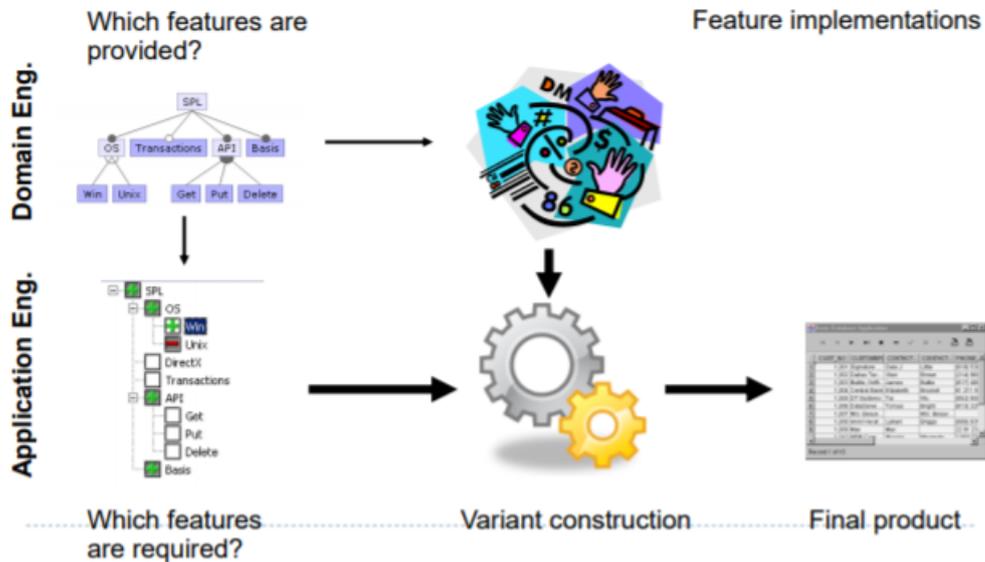


Figura 4.2: Ingeniería de Dominio y de Producto[2]

todos los productos de la línea y los distintos puntos de variación que se podrán modificar, con sus respectivas variantes.

Hay limitaciones a la hora de elegir las características para generar un producto, puede haber dependencias o restricciones entre características. Se considera que dos características son mutuamente excluyentes cuando no pueden aparecer a la vez en un producto. Al contrario decimos que dos características son simultáneas cuando deben aparecer a la vez, es decir, una de ellas depende de la otra.

#### 4.1.1. Diagrama de características

El diagrama de características es un esquema con forma de árbol en el que aparecen todas las características de la LPS. En la figura 4.3 se muestra el ejemplo de un diagrama de características para un *sistema de organización de una escuela de educación primaria*.

Procedo a explicar el uso de la notación:

- **Mandatory:** Esta anotación indica las características que son **obligatorias**
- **Optional:** Esta anotación indica las características que son **opcionales**; el cliente decide si se añade o no al producto. En caso de no elegir una característica opcional, se descartan todas las características que tienen a esta como padre.
- **Alternative:** El cliente deberá elegir obligatoriamente una de entre las características que están señaladas con esta anotación.
- **Or:** El cliente podrá elegir entre las características señaladas con esta anotación y como mínimo debe escoger una.
- **Abstract:** Son características que no se pueden instanciar.

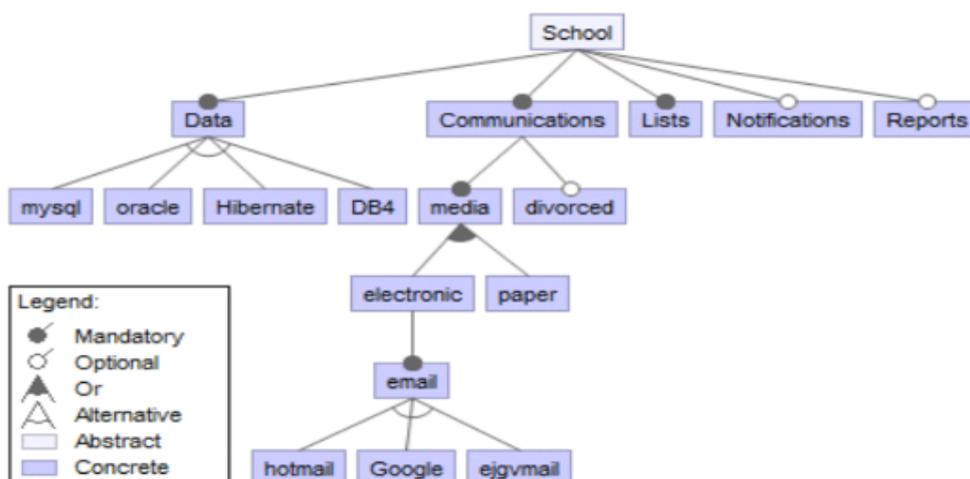


Figura 4.3: Ejemplo diagrama de características

- **Concrete:** Son características que tendrán una instancia.
- **Mencionar restricciones entre características**

Según el diagrama de la figura 4.3, la escuela debe contar con un servicio de listas y un sistema de comunicaciones. El sistema de comunicaciones tendrá la posibilidad de gestionar el envío de mensajes a ambos padres en caso de que se encuentren divorciados. La manera de hacer llegar estos mensajes podrá ser en papel, de manera electrónica o ambos a la vez. En caso de utilizar una gestión de comunicaciones electrónica, se deberá elegir el proveedor de email. Además, el colegio deberá contar con un método de almacenamiento (se podrá elegir entre mysql, oracle, Hibernate o DB4); probablemente sea elección del técnico instalador del servidor. Opcionalmente, el sistema se puede completar añadiendo la posibilidad de planificar de antemano cuándo hay que realizar alguna notificación. Es más, incluso se puede añadir la posibilidad de utilizar el sistema para comunicar a los padres los informes de los profesores sobre la evolución de los alumnos.

## 4.2. pure::variants

Pure::variants[7] es un plug-in desarrollado para el entorno de programación **eclipse**. Este plug-in nos provee las herramientas para crear y manipular una LPS.

Para la explicación de la herramienta se utilizará el proyecto “Weather Station”. Es un proyecto de ejemplo por defecto que viene con el plug-in.

Como se puede observar en la figura 4.4, con esta herramienta podremos visualizar y realizar cambios en el diagrama de características o *feature model*. La notación difiere en la explicada previamente, en este caso el símbolo de interrogación, “?”, hace referencia a las características **opcionales**, para las características **OR** se utiliza el icono ‘X’. Las características **obligatorias** se marcan con el signo de exclamación, ‘!’ y el símbolo ‘ $\Leftrightarrow$ ’ se utiliza para marcar las características **alternativas**.

## 4.2. pure::variants

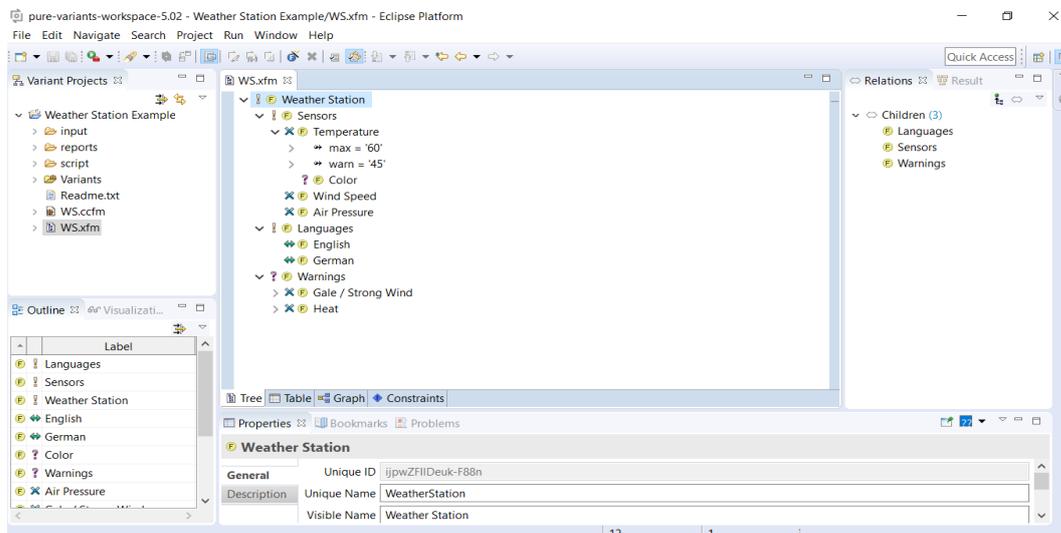


Figura 4.4: Diagrama de características de la LPS Weather Station

Pure::variants nos permite realizar el proceso de **selección de características**. Este trata de combinar un subconjunto de las características de la LPS para la generación de un producto.

Primero debemos crear un **modelo de variante**, o *Variant Model*, para después poder seleccionar las características en su propio **feature model**, como se puede ver en la figura 4.5. Estas variantes también disponen de su propio *Family Model*, el cual contiene toda la información referente al mapeo de los archivos de la LPS tal y como muestra la figura 4.6.

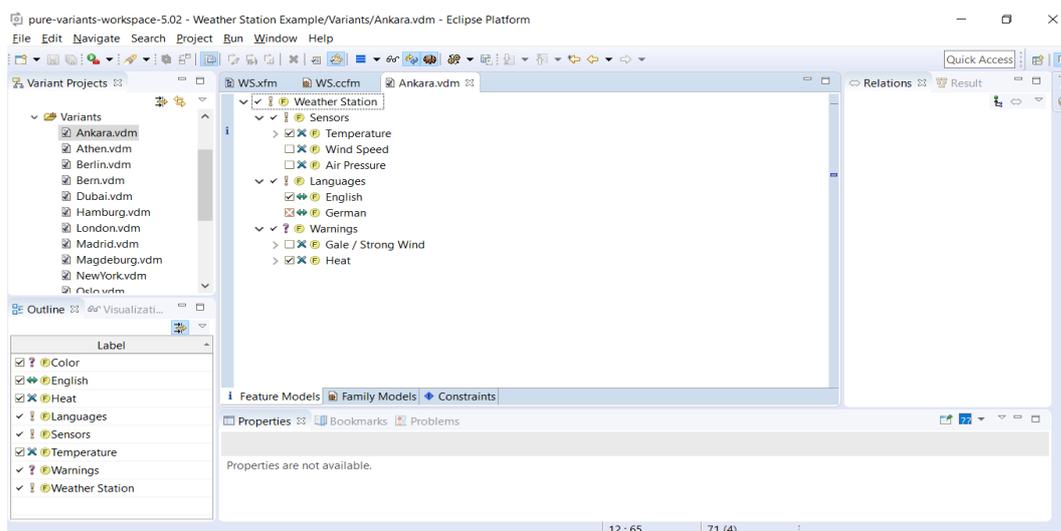


Figura 4.5: Selección de características para variante de Weather Station

Una vez definidas las características que tendrá nuestro producto, o variante, se procede a ejecutar la transformación. Esto hará que pure::variants genere una carpeta que contenga el código de nuestro producto, formado por el código correspondiente a las características seleccionadas. Esto se puede ver en la figura 4.7.

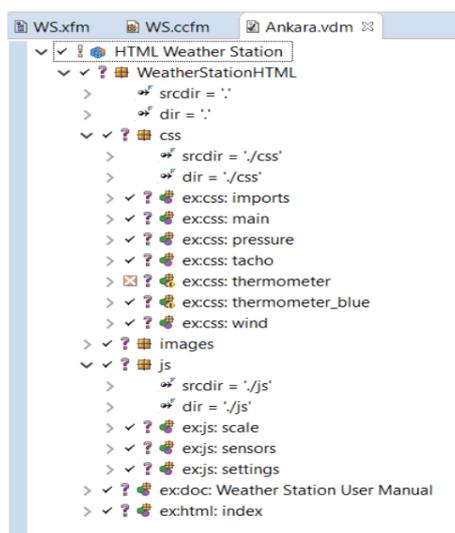


Figura 4.6: Family Model variante Ankara de Weather Station

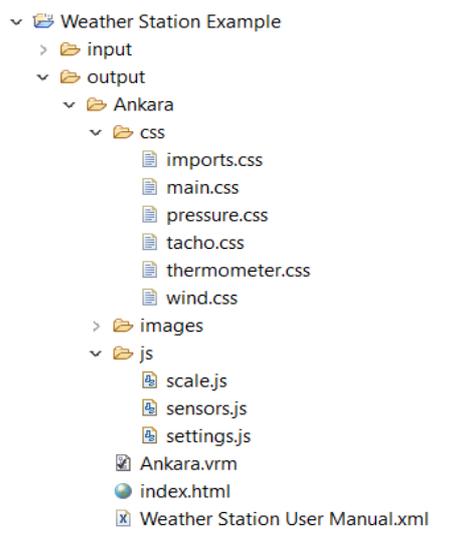


Figura 4.7: Variante Ankara generada a partir de la plataforma Weather Station

Para definir el código que corresponde a característica se utilizan anotaciones de pre-compilación. Estas anotaciones utilizan el lenguaje de expresión *pvSCL*, un lenguaje simple para expresar restricciones, constantes o cálculos. Provee operaciones lógicas para crear expresiones booleanas. Estas anotaciones se van a utilizar para definir el código correspondiente a las distintas características.

La figura 4.8 muestra un ejemplo de una anotación que utiliza *pvSCL*. En el caso de que se cumpla la condición que hay entre paréntesis, en este caso, que la característica 'WindSpeed' haya sido seleccionada en el *variant model* del producto, el código que se encuentra entre las dos anotaciones será exportado al realizar la transformación.

```
// PVSCL:IFCOND(WindSpeed)
var windMeasure = 0;
function applyWindSpeed() {
    var measureText = document.getElementById("w_measure");
    windMeasure = measureText.value;
    var pointer = document.getElementById("w_point");

    applyTachoValue(minWind, maxWind, measureText, pointer);
    setWarnings();
    return false;
}
// PVSCL:ENDCOND
```

**Figura 4.8:** Condición de característica pure::variants



# Arquitectura de la plataforma de LPS WACline

Ahora que ya se ha explicado qué es una LPS y cómo funciona la herramienta pure::variants, se procederá a explicar la estructura de las anotaciones web y las características de la LPS sobre la que se va a trabajar en este TFG, WACline.

## 5.1. Estructura de las Anotaciones Web

La LPS trabaja el dominio de las anotaciones web. Su arquitectura sigue las recomendaciones de W3C<sup>1</sup> para las anotaciones web. En la figura 5.1 se muestra un esquema del dominio de WACline.

Las anotaciones tienen dos partes fundamentales, el **Body** y el **Target**.

El Body se encarga de almacenar la información referente a la propia anotación: los comentarios añadidos, el tema en el que se ha clasificado, literatura sugerida...

Por otro lado, el Target contiene la especificación del contenido exacto anotado. Contiene información del recurso web anotado (URL, URN) y el fragmento anotado de este recurso web.

Los selectores son los que representan que fragmento del recurso se ha anotado. En la LPS se utilizan 4 de los selectores recomendados por el W3C: *FragmentSelector* que indica la página donde se encuentra la anotación, *TextPositionSelector* que contiene el número del carácter en el que comienza la anotación y el número del carácter en el que termina, *TextQuoteSelector* que contiene el texto de la anotación, su prefijo y su sufijo, ver figura 5.2 y por último *RangeSelector* que se utiliza en documentos HTML para indicar la etiqueta en la que comienza la anotación y en la que termina.

---

<sup>1</sup>W3C:<https://www.w3.org/TR/annotation-model/>

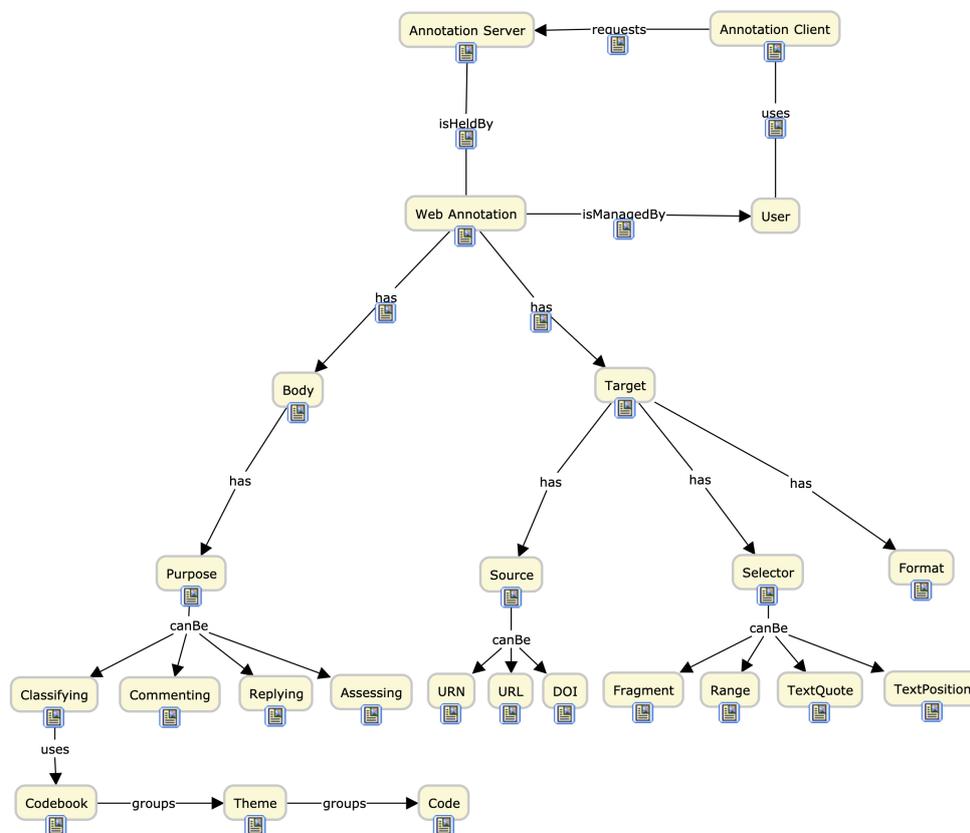


Figura 5.1: Esquema del dominio WebAnnotation

Source: <https://cmapscloud.ihmc.us/viewer/cmap/1V8RCFYS6-178WHN4-7JZNV2>

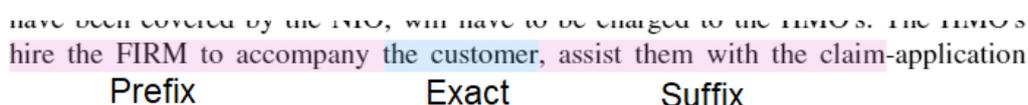


Figura 5.2: Prefijo, sufijo y exact

## 5.2. Características de WACLINe

El diagrama de características de WACLINe es bastante amplio. Por ello se dividirá el diagrama en varias partes que se irán analizando.

Los productos generados con la LPS son extensiones para realizar anotaciones. Como se puede observar en la figura 5.3, estas anotaciones pueden guardarse tanto de manera remota, en el servidor de Hypothesis<sup>2</sup> o en Neo4J<sup>3</sup>, como de manera local, haciendo uso del almacenamiento del navegador<sup>4</sup>. Como mínimo debe seleccionarse uno de los servidores de almacenamiento o el almacenamiento local.

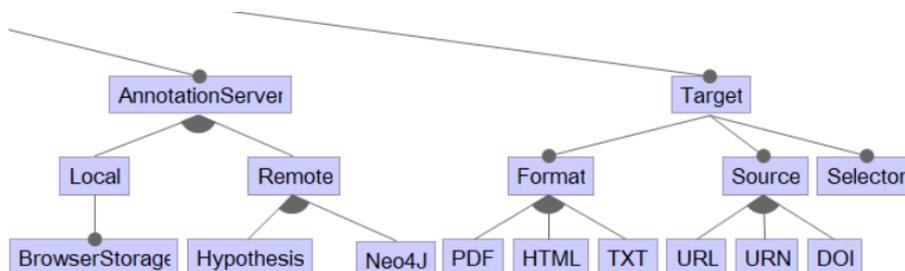
Se puede seleccionar qué tipo de contenido se va a poder anotar: paginas web (HTML), PDFs o documentos en formato de texto plano. Además, también se podrá elegir qué

<sup>2</sup>Hypothesis es otra herramienta de anotación web: <https://web.hypothes.is/>

<sup>3</sup>Neo4J es un software libre de base de datos orientado a grafos

<sup>4</sup>Almacenamiento: <https://developer.chrome.com/docs/extensions/reference/storage/>

métodos se utilizarán para identificar los recursos anotados, URL, URN y DOI.



**Figura 5.3:** Diagrama de características WACline AnnotationServer y target

La figura 5.4 muestra las operaciones que se pueden realizar con las anotaciones. El producto siempre debe incluir operaciones básicas de crear, leer y eliminar anotaciones. Opcionalmente, también se podrán actualizar anotaciones o eliminarlas todas a la vez. Se ofrece la opción de poder mostrar las anotaciones en otras aplicaciones como GoogleSheet<sup>5</sup>, MindMeister<sup>6</sup> o Moodle<sup>7</sup>. También está la opción de mostrarlas fuera de contexto, sin utilizar ninguna aplicación de terceros, a modo de lista.

Como es de esperar, los productos siempre incluirán la capacidad de subrayar texto del documento para realizar las anotaciones. Se pueden añadir opciones para ir a la última anotación, mostrar un canvas con un resumen de las anotaciones del documento o navegar a las anotaciones de cierto tema al seleccionarlo. Esto ultimo solo será posible si se ha seleccionado la opción de clasificación de anotaciones.

Las extensiones de manera opcional tendrán la capacidad de importar y exportar las anotaciones. A la hora de exportar se pueden elegir varios formatos: el documento en formato PDF subrayando las anotaciones realizadas, un informe que contenga las anotaciones con las observaciones realizadas o un documento JSON que puede ser importado de nuevo por la aplicación.

La aplicación de manera opcional podrá crear grupos de anotaciones para un mismo documento, a los que se les llama *'codebooks'*. La figura 5.5 muestra todas las características relacionadas con ellos. Si se escoge la opción manual para leer codebooks se podrá cambiar de codebook y crear nuevos, si no, será la misma aplicación o la aplicación externa Moodle la que cree un solo codebook que será el que se utilice todo el tiempo.

Los codebooks podrán ser creados de manera predeterminada con un esquema por defecto o utilizando las aplicaciones de terceros mencionadas previamente. De manera opcional estos codebooks podrán renombrarse y eliminarse. Los codebook podrán tener una jerarquía para clasificar las anotaciones. A estos subgrupos dentro de un codebook se les llama *'themes'*. También cabe la posibilidad de que se quiera añadir más de una categoría para un mismo theme, es decir, que sean multievaluados. A estas categorías se les denomina *'codes'*. Podrá elegirse el orden en el que se muestran los themes y codes: orden alfabético, por fecha de creación o por enumeración.

<sup>5</sup>GoogleSheet:<https://www.google.es/intl/es/sheets/about/>

<sup>6</sup>MindMeister:<https://www.mindmeister.com/es>

<sup>7</sup>Moodle:<https://moodle.com/>

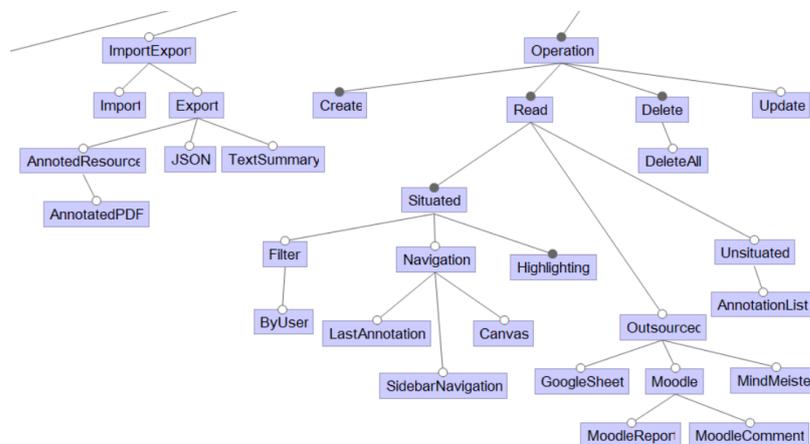


Figura 5.4: Diagrama de características WACLINÉ operation e import/export

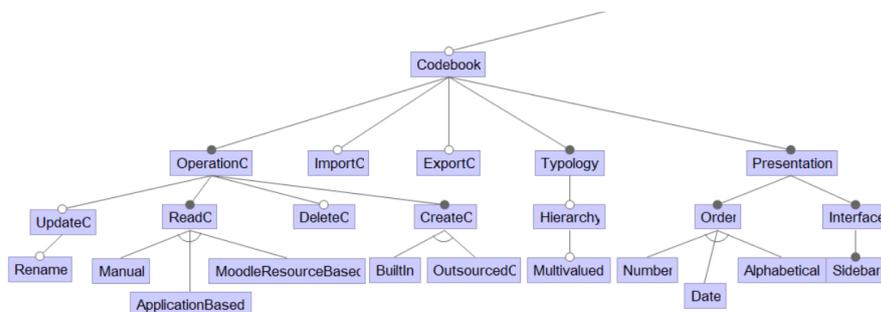


Figura 5.5: Diagrama de características WACLINÉ Codebook

### 5.3. Estructura del proyecto

Se ha comentado que WACLINÉ es una LPS de extensiones para el navegador. Su estructura, que se muestra en la figura 5.6, se puede dividir en 2 secciones:

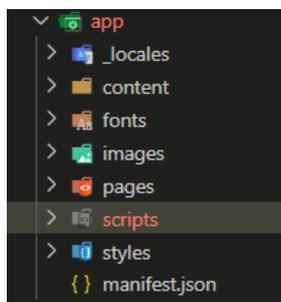


Figura 5.6: Estructura proyecto WACLINÉ

- **Apartado visual:** Corresponde a las carpetas *content*, *fonts*, *images*, *pages* y *styles*. En esta carpeta se almacenan ficheros html, css, imágenes...
- **Parte lógica:** Corresponde a la carpeta *scripts*. Contiene toda la parte lógica del proyecto. Para una mejor gestión el proyecto ha estructurado esta carpeta en base a

las características de la LPS. Esto se puede observar en la figura 5.7

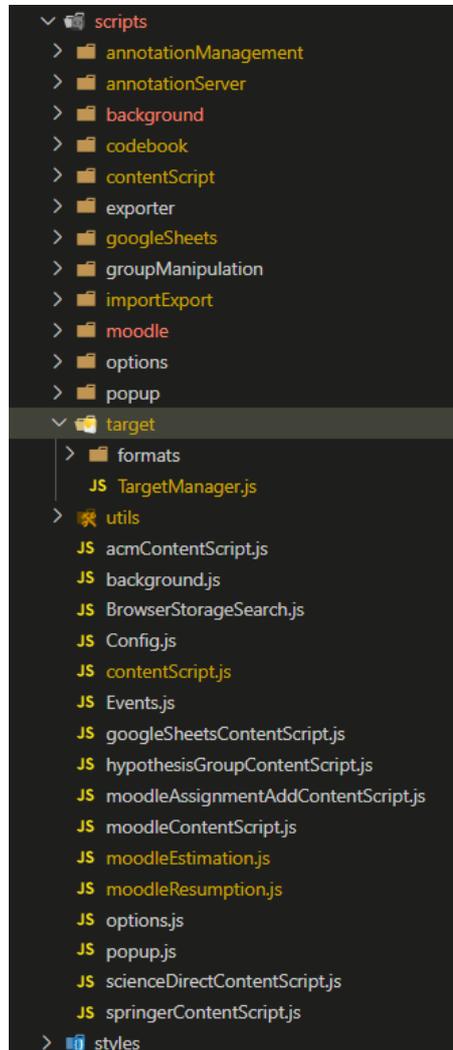


Figura 5.7: Estructura scripts proyecto WACline

Además de estas dos secciones, el proyecto cuenta con el fichero 'manifest.json' que contiene la información de la aplicación.

## 5.4. Review&Go

Para este proyecto se va a utilizar como base un producto ya existente en la LPS, *Review&Go*, cuyo objetivo también es facilitar la revisión de manuscritos. La figura 5.8 muestra el subconjunto de características de la LPS WACLIne que incluye *Review&Go*. Se puede ver como el producto ya dispone de características apropiadas para realizar revisiones, por ejemplo, la posibilidad de clasificar las anotaciones en *themes* y en sus correspondientes *codes* o una vista tipo *canvas* para tener una imagen de todas las anotaciones realizadas en el documento y su clasificación. En este proyecto se realizarán modificaciones en las características ya existentes, descartando algunas de ellas, y se añadirán otras nuevas.

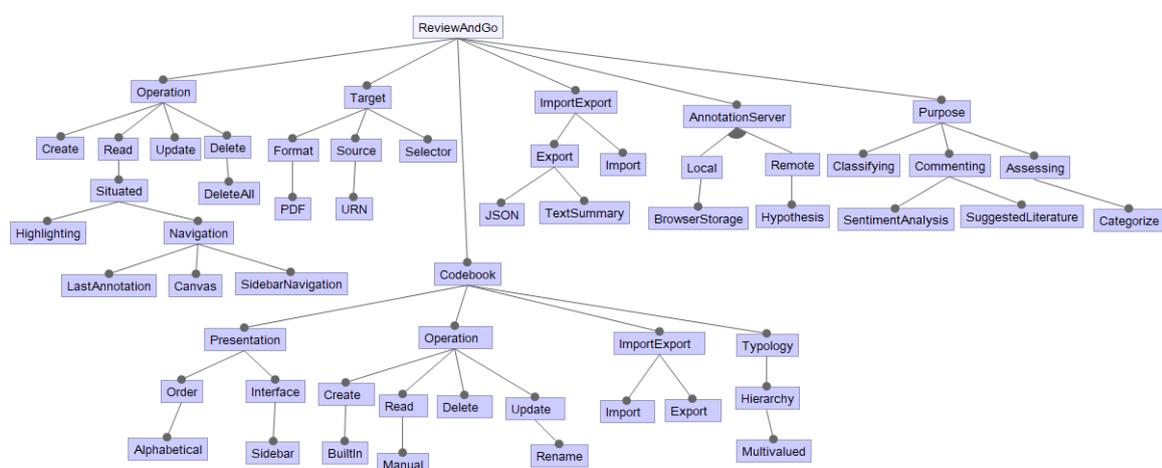


Figura 5.8: Subconjunto de características de WACLIne, *Review&Go*

# Diseño e implementación

Este capítulo contiene la descripción de objetivos y explicaciones de la implementación de cada uno de los módulos que han sido implementados durante este proyecto junto a los problemas surgidos durante el desarrollo de los mismos. Al final se mostrará un diagrama con el subconjunto de características que se han añadido o han sufrido cambios durante el proyecto. El código del proyecto se encuentra en el siguiente repositorio: <https://github.com/inigoBereciartua/WacLine/tree/product/reviewAndGo>

## 6.1. Módulo Búsqueda de Keywords (MBK)

Esta sección describe los objetivos, la implementación y los problemas surgidos correspondientes del módulo de búsqueda de palabras clave.

### 6.1.1. Objetivos

El objetivo principal de este módulo es ayudar al usuario a la hora de elegir un método de revisión. Los métodos de revisión agrupan criterios para la revisión de artículos. Cada método de revisión tiene unas palabras clave que de aparecer en un artículo, se recomienda que sea evaluado con dicho método.

Mediante este módulo el usuario podrá conocer cuantas coincidencias de palabras clave han aparecido para cada método. Gracias a esto, el usuario obtendrá bastante información del artículo para decidir que métodos seleccionar, sin tener que haber leído el artículo entero.

### 6.1.2. Implementación

La figura 6.1 muestra como para la implementación de este módulo, se ha añadido una nueva característica al diagrama de características, llamada *KeywordBasedAnnotation*.

Para realizar la búsqueda de palabras clave primero había que cargar el documento como texto. Para ello se hace uso de la librería 'PDF.js'<sup>1</sup>, una librería basada en estándares web para la conversión y renderizado de documentos PDF.

<sup>1</sup>PDF.js:<https://mozilla.github.io/pdf.js/>

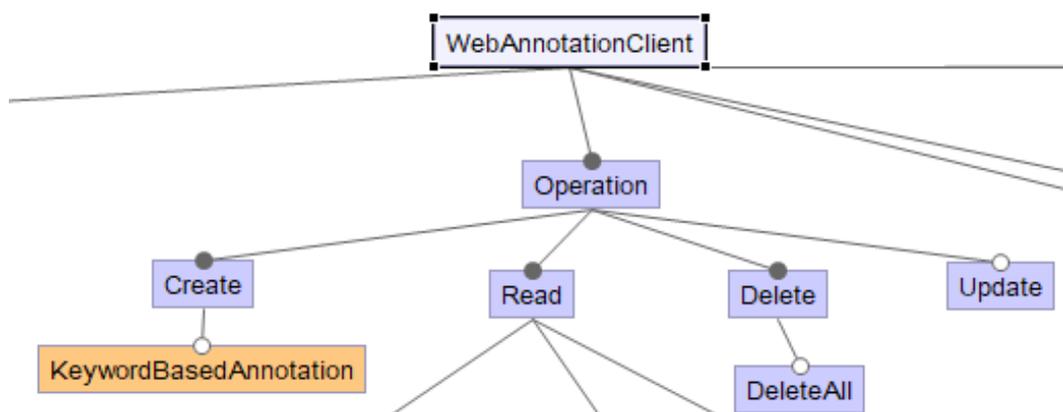


Figura 6.1: Nueva característica MBK

Gracias a esta librería se puede obtener el texto de cada página del documento. Una vez obtenido el texto, se comprueba página por página si se encuentra la palabra clave que se está buscando. Como se ha comentado en el capítulo 5, las anotaciones se ubican mediante su target, y dentro del documento a través de sus selectores. Así que una vez se encuentra una palabra clave en el documento, se deben definir sus selectores.

Gracias a que la librería permite obtener el texto por páginas, se puede definir el *FragmentSelector*. Además, como el documento se ha procesado como texto, se puede obtener el índice del carácter en el que comienza la palabra clave para determinar el *PositionSelector*, y buscar el prefijo y el sufijo para completar el *TextQuoteSelector*.

Para clasificar las palabras clave se debía crear un nuevo *theme* al que asignarlas. Por lo que al comenzar el proceso de búsqueda de palabras clave se crea el tema 'Keywords'.

La LPS solo permitía crear anotaciones de una en una. Esto era poco eficiente puesto que se lanzarían varios eventos de manera repetida por cada palabra clave que se anotaba. Para evitar esto, se añadió la capacidad de crear varias anotaciones a la vez si se indicaba el *theme* y los selectores de las anotaciones que se querían crear.

Para evitar tener que lanzar el proceso de búsqueda cada vez que se carga el documento o cada vez que se desea seleccionar un nuevo método de revisión, este proceso se ejecuta una única vez por documento. Se almacena la información recopilada (el número de coincidencias de palabras clave) por cada método de revisión.

### 6.1.3. Problemas encontrados

- **Carga del documento:** Uno de los problemas encontrados durante el desarrollo de este módulo era no saber cómo cargar el documento en formato de texto. El documento se muestra en la aplicación después de haber sido renderizado en formato HTML, pero este formato no era apropiado para encontrar los selectores.
- **Espacios entre palabras:** A veces entre palabras se detectaba más de un espacio, que no se ve reflejado al realizar anotaciones manuales, lo cual supone un problema para que la aplicación encuentre la anotación. Para solucionarlo se añadió siempre un único espacio entre palabras para minimizar el número de casos en el que falle.

- **Caracteres específicos:** Algunos caracteres como la letra ‘ñ’ y las tildes producían errores en el procesamiento, por ejemplo la palabra ‘aplicación’ se procesaba como ‘aplicaci´on’. Para solucionarlo se realiza una normalización del texto. A pesar de haber normalizado el texto, las tildes seguían dando problemas cuando las llevaba la primera letra de la palabra. Esto no pudo ser resuelto, pero no se le dio importancia puesto que los textos que serían analizados estarían escritos en inglés. Sin embargo, se abrió una ‘issue’ en el repositorio de Github de WACline.
- **Subrayado:** A pesar de haber determinado correctamente el *FragmentSelector* y el *TextQuoteSelector* (que son los principales que se utilizan) y que la aplicación hubiera encontrada la palabra clave en el texto, no se producía el subrayado. Esto se debía a que había un error en la función que realizaba el subrayado. A pesar de encontrar la palabra en el texto, si no se especificaban todos los selectores no se ejecutaría correctamente. Para solucionarlo Haritz realizó la modificación correspondiente en la función y además se añadió el último selector que faltaba a la búsqueda de las palabras clave, el *PositionSelector*.
- **Disponibilidad del codebook:** Uno de los problemas que no solo se presenta en este módulo es la no disponibilidad del objeto *codebook*, que es necesario para procesos como obtener un *theme*. Para solucionarlo, se realiza una escucha y espera de la variable del *codebook* y se inicia el proceso de búsqueda de palabras clave una vez este está definido.
- **Saturación del navegador:** A la hora de realizar pruebas, se intentaron buscar todas las palabras claves de un documento que contenía casi todas ellas. Esto causó que por cada una tuviera que crearse una anotación. El navegador se sobrecargó y llegó a bloquearse. Se valoró el no realizar la anotación de las palabras y que solo se contara el número de coincidencias. Sin embargo, era interesante que las palabras aparecieran marcadas puesto que el revisor quizá querría comprobar que las palabras clave hicieran referencia al contexto para el que estaban pensadas. Hay que mencionar que con las pruebas realizadas con artículos reales no se ha llegado a producir ninguna saturación, así que se espera que los revisores no lleguen a sufrir este problema.

## 6.2. Módulo Búsqueda de Información de Autores (MIA)

Esta sección comenzará describiendo los objetivos de este módulo de búsqueda de información de los autores. Más tarde se comentará el diseño de las interfaces que continuará presentando como ha sido implementado y por último se describirán los problemas encontrados durante su desarrollo.

### 6.2.1. Objetivos

Como ya se ha comentado, el objetivo de este módulo es la búsqueda de información de los autores. Al revisor de un artículo le puede ser de utilidad tener información acerca del número de publicaciones que tiene un autor para ajustar el nivel de exigencia con el que va a realizar la revisión. Además, es interesante que pueda tener acceso rápido a la biblioteca de artículos de cada uno de los autores para realizar consultas.

### 6.2.2. Interfaces

Este módulo se puede dividir en dos etapas. En la primera se le solicita al usuario que ingrese el congreso del que se quiere obtener la información de autores y se crea el tema para que el usuario pueda anotarlos. En la segunda se procesan los autores que ha anotado y se muestra la información que se ha obtenido de ellos.

#### 6.2.2.1. Solicitud de congreso

Este módulo añade el menú de elección de congreso y la pantalla de muestra de información. Para la implementación de la solicitud del congreso se hizo uso de las alertas que estaban implementadas en la LPS. Estas alertas son mensajes emergentes que aparecen en la pantalla haciendo uso de 'Sweetalert'<sup>2</sup>, una librería para alertas que pueden utilizarse para hacer formularios fácilmente. La figura 6.2 muestra un ejemplo de como se ven estas alertas, en este caso, para solicitar el congreso. Además, como se puede ver en la figura 6.3, para la selección del congreso se ha añadido la función de autocompletado. Este formulario para que el usuario introduzca el congreso, se mostrará nada más abrir el documento en caso de que el usuario no haya seleccionada un congreso previamente.

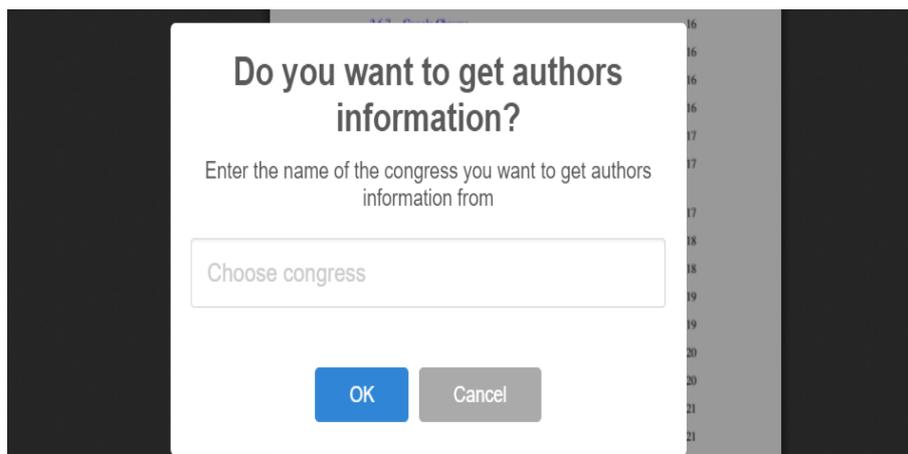


Figura 6.2: Interfaz selección congreso

#### 6.2.2.2. Visualización de la información

La pantalla que muestra la información de los autores se hizo incrustando código HTML y CSS en el DOM. La figura 6.4 muestra como se ve la pantalla cuando se han añadido autores. Si no se encuentra ningún autor con el nombre que se indica, mostrará un mensaje distinto. En caso de que no haya realizado ninguna anotación para el *theme* de los autores se mostrará una pantalla diferente, reflejada en la figura 6.5.

### 6.2.3. Implementación

La figura 6.6 muestra como para la implementación de este módulo, se ha añadido una nueva característica, al diagrama de característica, llamada *AuthorsSearch*.

---

<sup>2</sup>SweetAlert:<https://sweetalert2.github.io/>

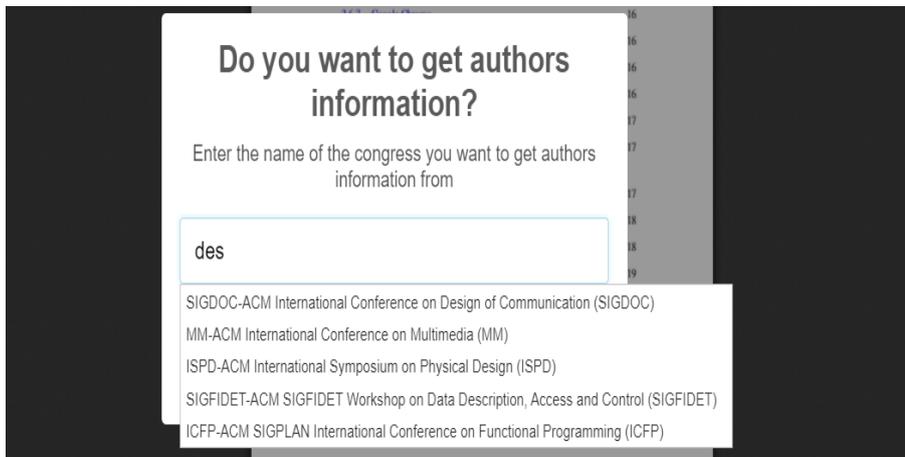


Figura 6.3: Interfaz selección congreso autocompletado

| Authors overview for DESRIST congress |                                |   |
|---------------------------------------|--------------------------------|---|
| Research Agenda                       | This author could not be found |   |
| Brian Donnellan                       | 5 publications                 | <a href="https://dblp.org/pid/57/288">https://dblp.org/pid/57/288</a>   |
| Debra E. VanderMeer                   | 6 publications                 | <a href="https://dblp.org/pid/84/5479">https://dblp.org/pid/84/5479</a> |
| Markus Helfert                        | 0 publications                 | <a href="https://dblp.org/pid/17/2368">https://dblp.org/pid/17/2368</a> |
| Jim Kenneally                         | 3 publications                 | <a href="https://dblp.org/pid/69/6642">https://dblp.org/pid/69/6642</a> |
| Marcus A. Rothenberger                | 5 publications                 | <a href="https://dblp.org/pid/12/631">https://dblp.org/pid/12/631</a>   |
| Robert Winter 0001                    | 2 publications                 | <a href="https://dblp.org/pid/84/285">https://dblp.org/pid/84/285</a>   |

Figura 6.4: Información autores



Figura 6.5: Interfaz información de autores (sin autores)

Como ya se ha comentado, para la implementación de este módulo se hace uso de la API del repositorio bibliográfico DBLP<sup>3</sup>. Se ha utilizado el endpoint<sup>4</sup> para obtener la lista de congresos que contiene y con la función 'autocomplete' de JQuery<sup>5</sup> se ha podido resolver el autocompletado para la solicitud del congreso al usuario.

Una vez el usuario ha seleccionado un congreso, se creará una anotación especial que contiene el nombre del congreso y su acrónimo y se añadirá el tema llamado 'Authors' al *codebook*. En caso de que el usuario cancele y no seleccione ningún congreso, no se añadirá el tema y tampoco se mostrará el quinto botón que se observa en la figura 6.7, que abre la pantalla de información de los autores.

Para la implementación de la pantalla que muestra la información, también se ha hecho uso de la API de los autores y de las publicaciones. Primero se hace una búsqueda del

<sup>3</sup>API DBLP faq: <https://dblp.org/faq/How+to+use+the+dblp+search+API.html>

<sup>4</sup>Endpoint: URLs de un API que responden a peticiones

<sup>5</sup>JQuery: es una biblioteca de JavaScript que permite simplifica algunas tareas de la programación

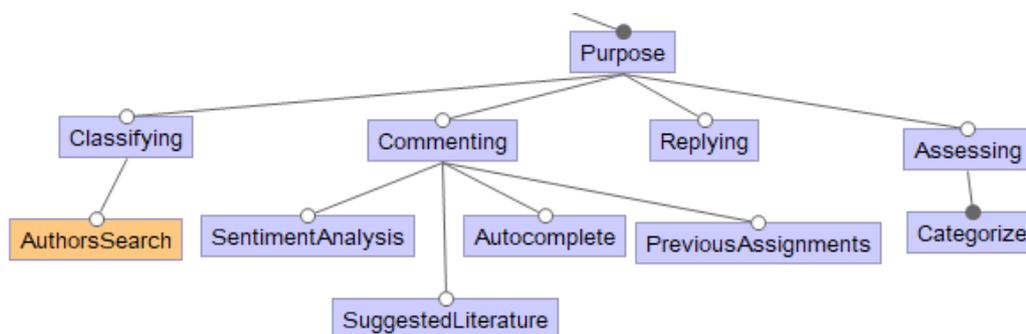


Figura 6.6: Nueva característica MIA

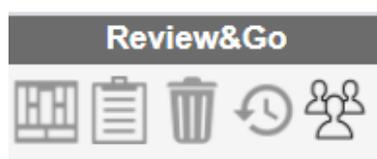


Figura 6.7: Sidebar con nuevo botón de autores

autor con la API. En caso de que no se encuentre el autor no se buscará más información sobre él, como se puede ver en la figura 6.4. Por el contrario, en caso de encontrarlo, se obtendrá la url de su perfil del DBLP para mostrarla y se utilizará la API de nuevo para buscar las publicaciones que lo incluyan como autor. Dichas publicaciones serán filtradas, para comprobar que el congreso en el que se han publicado coincide con el congreso que el usuario ha introducido previamente, y se mostrará el número de coincidencias.

#### 6.2.4. Problemas encontrados

- API request anidadas:** El mayor problema encontrado durante el desarrollo de este módulo fue el problema de la sincronía. Todas las llamadas a la API se producen de manera asíncrona y en un principio se ejecutaban dos llamadas a API seguidas, la segunda dependiente del resultado de la primera. Esto hacía que muchas veces la información para realizar la segunda llamada no estuviera disponible todavía, lo cual producía un error. Además, el programa no esperaba a que se hubieran completado todas las peticiones antes de devolver la lista de autores, por lo que no se asignaba correctamente la información. Para resolver esto, se hizo uso de las promesas. Las promesas son objetos que representan la terminación o el fracaso de una operación asíncrona. Estas permiten ser almacenadas y ejecutar una función una vez todas han sido completadas. De esta manera se podía asegurar que todas las peticiones se habían completado y que se dispone de toda la información con la lista de autores completa.
- Actualización de dependencias:** Una vez se había finalizado con la implementación, se actualizaron algunos de los paquetes de las dependencias. En concreto, se actualizó *SweetAlert*. Después de esto el autocompletado de búsqueda de congresos dejó de funcionar. El problema era que habían cambiado el nombre de uno de los parámetros que se pasaba para crear las alertas, el que hacía que se ejecutara código mientras se mostraba la alerta. Al cambiar el nombre del parámetro se solucionó el problema.

## 6.3. Módulo Métodos de revisión (MMR)

Esta sección comenzará describiendo los objetivos del módulo de métodos de revisión, después se comentarán las interfaces, se realizará una descripción de la implementación y se concluirá con los problemas surgidos en la implementación del módulo.

### 6.3.1. Objetivos

El principal objetivo de este módulo, es añadir el uso de listas de criterios del *Empirical Standard* para la revisión de los artículos. Gracias a este módulo, el usuario será capaz de importar los criterios que seleccione de estas listas a su *codebook*, para realizar las revisiones. Además, el revisor tendrá la capacidad de consultar e indicar si el artículo ha pasado o no los criterios definidos. Los criterios han sido organizados según su relevancia en tres grupos: esenciales, deseables y extraordinarios.

Una funcionalidad que ya estaba implementada en la LPS, es la posibilidad de generar y descargar un informe con las anotaciones y comentarios realizados en el documento. Esta funcionalidad ha sido ampliada. Ahora el revisor tendrá la capacidad no solo de descargar, sino que además, previamente podrá editar dicho informe y guardar un borrador. El informe descargado también contendrá información sobre los métodos de revisión utilizados en la revisión y el resultado de evaluar sus criterios.

### 6.3.2. Interfaces

La implementación de este módulo cuenta con tres interfaces: la selección de un método de revisión y sus criterios, las pantallas para valorar los criterios y la pantalla para editar el informe.

#### 6.3.2.1. Selección método de revisión

Como se puede observar en la figura 6.8, esta interfaz ha sido creada utilizando las alertas implementadas en la LPS, al igual que la de elección de congreso mencionada en la anterior sección. Consta de un desplegable que, como muestra la figura 6.9, contiene todos los métodos del *Empirical Standard* y el número de coincidencias de palabras clave correspondientes a cada método. Al usuario también se le abrirá este menú en caso de que cree un tema cuyo nombre coincida con alguno de los filtros de las listas de criterios: Evaluation, Quantitative, Visualización o IRR. En este caso únicamente se le mostrarán los métodos que hayan pasado el filtro.

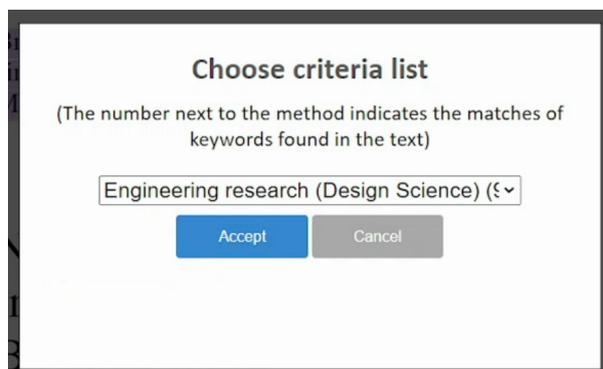


Figura 6.8: Seleccionar lista de criterios

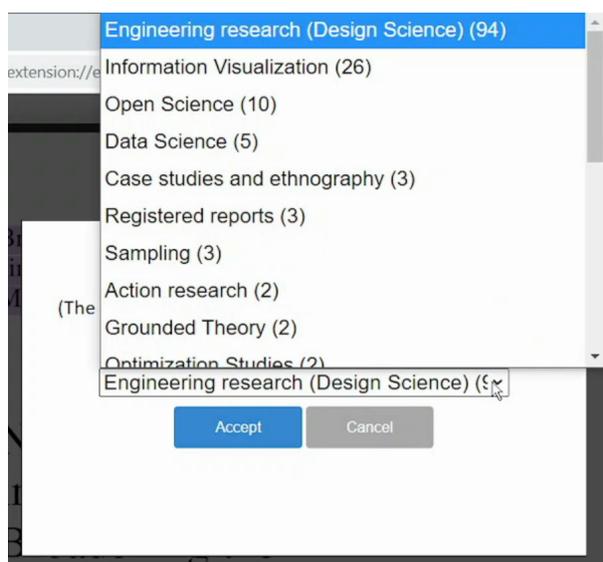


Figura 6.9: Seleccionar lista de criterios (desplegada)

Una vez el usuario ha escogido un método, se le mostrará la pantalla para seleccionar criterios. Como muestra la figura 6.10, se le mostrarán todos los criterios que incluye el método que había escogido y podrá seleccionar los que quiere añadir para revisar el documento.

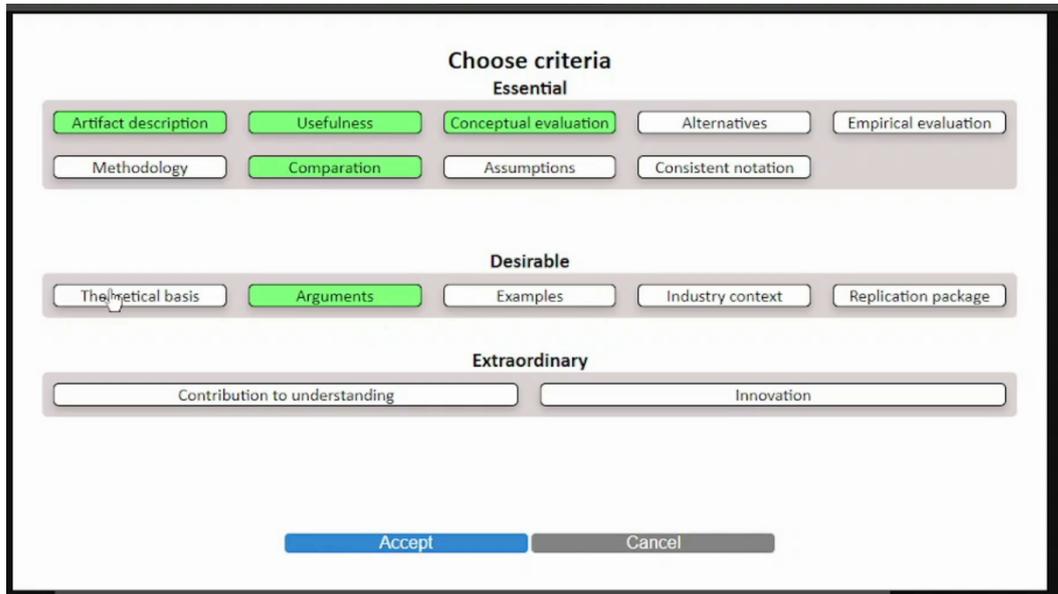


Figura 6.10: Seleccionar criterios de una lista

Una vez pulse confirmar y se hayan añadido con éxito los criterios, como se puede observar en la figura 6.11, se le mostrará un mensaje que así lo indique. La figura 6.12 muestra como cada criterio se ha añadido al tema de su nivel correspondiente.

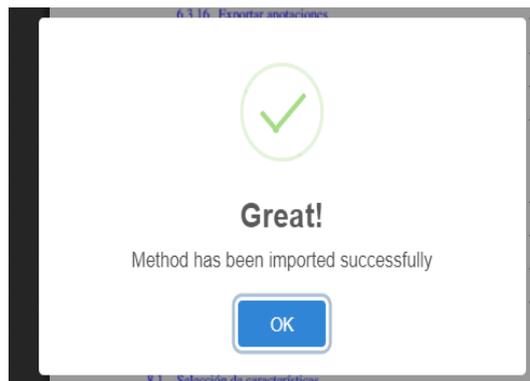


Figura 6.11: Método importado con éxito



Figura 6.12: Sidebar con nuevos criterios

### 6.3.2.2. Valorar criterios

La LPS contaba con una funcionalidad que, como se puede observar en la figura 6.13, mostraba un *Canvas* que contiene información sobre los temas del *codebook* y las anotaciones realizadas en cada uno de los temas. Este nuevo módulo, también requería un *Canvas* para visualizar toda la información de los métodos y realizar las valoraciones. Para ello, se ha aprovechado el *Canvas* ya implementado en la LPS y se ha realizado alguna modificación.

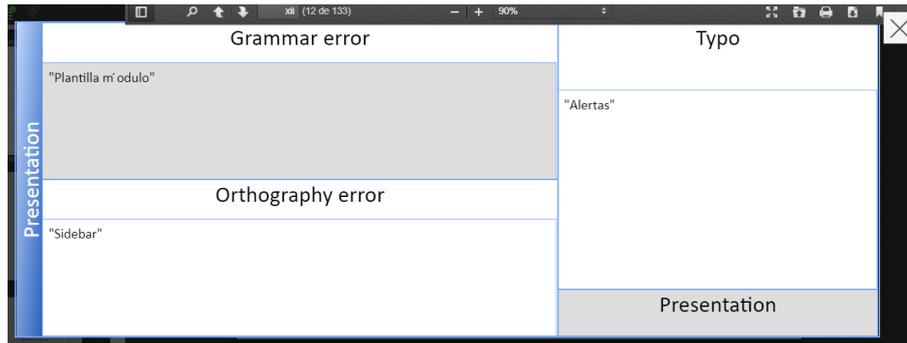


Figura 6.13: Antiguo *Canvas*

La modificación principal realizada ha sido agrupar los tres niveles que describen los criterios y añadirlos al final, tal y como muestra la figura 6.14. También se ha decidido que el tema correspondiente a las palabras clave abultaba demasiado y que no aportaba información, por eso ha sido excluido del *canvas*. Lo mismo ocurre con el tema correspondiente a los autores, 'Authors'.

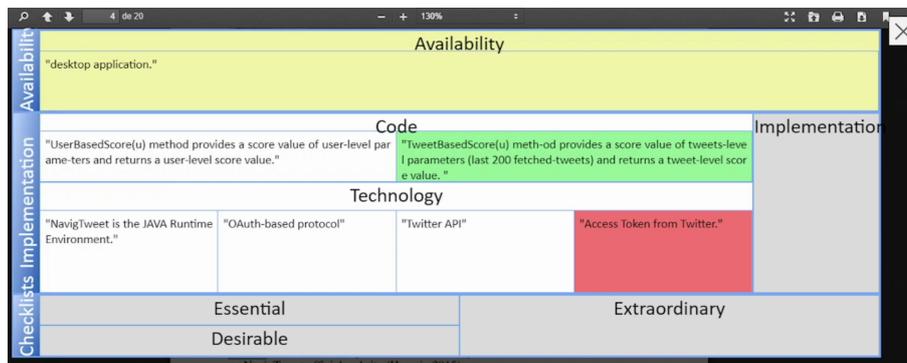


Figura 6.14: *Canvas* modificado

Si el usuario pincha en uno de los niveles de los criterios, se le abrirá un nuevo menú que como se puede ver en la figura 6.15, muestra una vista con los criterios de dicho nivel.

Los criterios serán mostrados de manera agrupada, clasificándolos por los métodos del 'Empirical Standard' del que provienen. El color de la casilla del criterio depende de su estado. Si ha sido aprobado su color será verde; si se ha suspendido, rojo; si no ha sido evaluado pero contiene anotaciones, naranja; o si todavía no se ha evaluado y no contiene anotaciones, blanco.

Si el usuario pincha en alguno de los criterios entrará al menú de evaluación de ese criterio y, como muestra la figura 6.16, podrá ver las anotaciones realizadas.



Figura 6.15: Canvas criterios

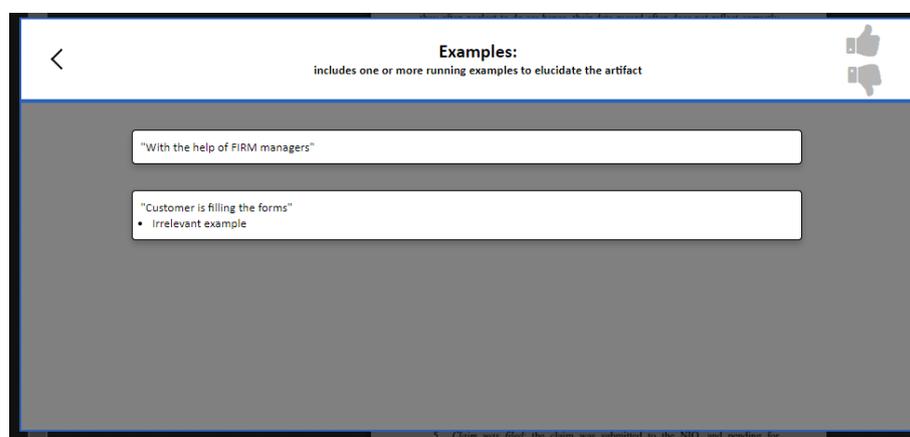


Figura 6.16: Vista criterio

El color de las manos de 'like' y 'dislike' variará según se seleccione como se puede ver en la figura 6.17.

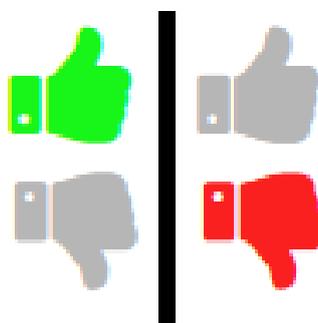


Figura 6.17: Like/Dislike

En el menú de los criterios, el usuario tendrá la posibilidad de pinchar sobre una anotación que ha realizado para ir a su localización en el texto. Como se muestra en la figura 6.18 tendrá la posibilidad de volver hacia atrás desde esta vista. Si hace doble clic, la flecha para volver atrás desaparecerá y volverá aparecer el *sidebar*.

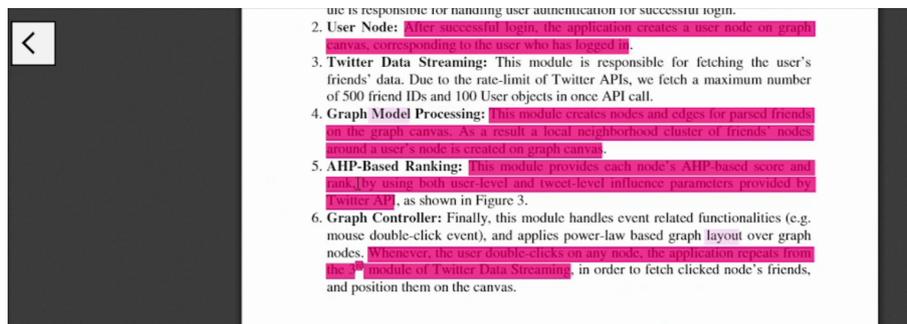


Figura 6.18: Contexto de anotación

### 6.3.2.3. Editar informe

Como se ha comentado la LPS tenía la opción de descargar el informe generado. Ahora el usuario podrá editar dicho informe como muestra la figura 6.19.

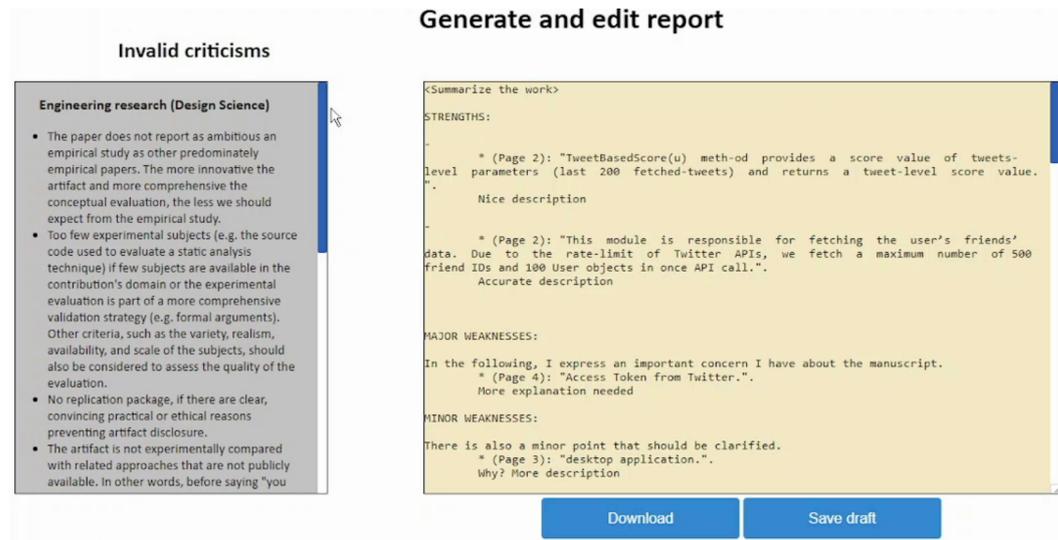


Figura 6.19: Edición de informe

Si el usuario pincha el botón 'Save draft' se guardará un borrador. Este será comparado con los próximos cambios que realice en las anotaciones para mantener la información más reciente. En caso de que presione 'Download', se descargará el informe en formato de texto plano. Como muestra la figura 6.20, el informe descargado incluye un resumen de la evaluación de criterios realizada además de la información de las anotaciones ya mencionada.

### 6.3.3. Implementación

La figura 6.21 muestra como para la implementación de este módulo, se ha añadido una nueva característica al diagrama de características llamada *ImportChecklist*.

Además, también se quitaron dos características, que se muestran en la figura 6.22 en color rojo, del producto que se iba a generar: *DeleteAll* y *LastAnnotation*. Cada una de ellas tenían un botón en el *sidebar* y no aportaban una funcionalidad relevante, por eso

## 6. DISEÑO E IMPLEMENTACIÓN

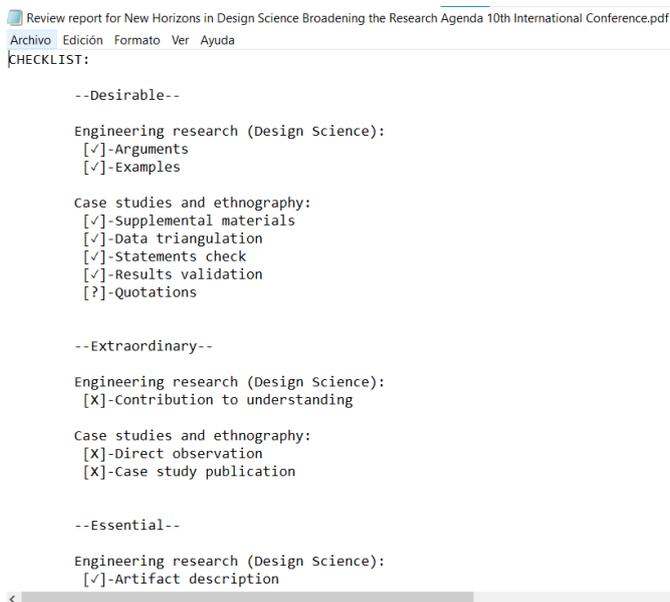


Figura 6.20: Informe descargado

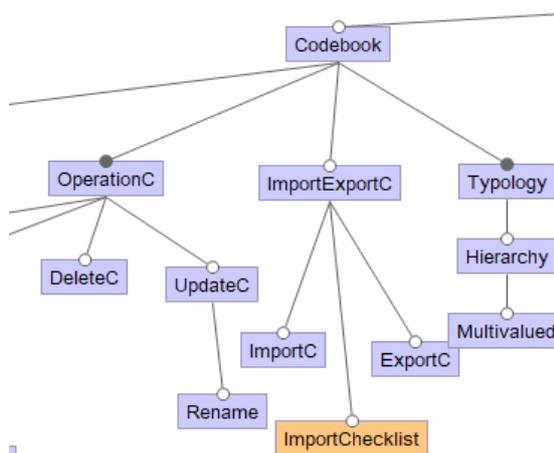
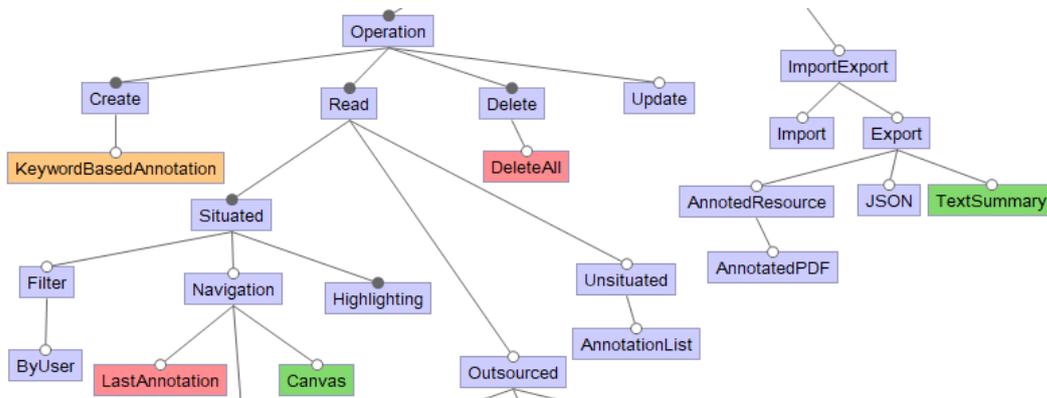


Figura 6.21: Nueva característica MMR

para mejorar la estética del menú lateral se decidió retirarlas. También hubo que realizar algunos cambios en las características mostradas en *verde*, *Canvas* y *TextSummary*. Hubo que modificar el canvas para poder unir la vista de criterios de una lista y el que canvas que ya incorporaba la LPS. *TextSummary* es la característica que se encarga de generar el informe. Necesitaba cambios para poder tratar el informe en la pantalla de edición que se ya se ha mostrado en la figura 6.19.

La información correspondiente a los métodos de revisión (sus criterios, las críticas no válidas y las palabras clave) han sido guardadas en un fichero JSON.

Como se ha comentado antes, al comentar el 'módulo de búsqueda de keywords', se almacena por cada método el número de palabras clave encontradas en el documento. Una vez el usuario selecciona uno de los métodos, se cargará una copia de la información de dicho método para mostrar los criterios y que el usuario pueda seleccionar los que desea



**Figura 6.22:** Características descartadas y modificadas para el producto Review&Go+

importar. Una vez ha completado este proceso, se cargará la información del método y los criterios seleccionados en las anotaciones que contienen la información de las listas de criterios importadas.

Esta anotación sirve para almacenar los datos de la evaluación de criterios (si han sido aprobados, suspendidos o todavía no se han definido) y modificarlos cuando el usuario interactúe con el menú de uno de los criterios.

La información del reporte también hace uso de esta información, puesto que muestra en el lateral la lista de críticas no válidas, compuesta por las listas de todos los métodos de revisión seleccionados.

El guardado de borrador ha sido implementado de forma que se comprobará que anotaciones no han cambiado desde que se hizo el último guardado del borrador, de esta forma se sabe que comentarios hay que mantener del borrador.

#### 6.3.4. Problemas surgidos

- Interpretación de las especificaciones:** Como se detallará más adelante en el capítulo 7, hubo un problema a la hora de interpretar cómo debía ser implementada esta característica. En un principio se pensó que el revisor solo iba a ser capaz de importar una lista de criterios con todos y cada uno de sus criterios; se hizo una primera implementación de esto. Antes de cargar la lista de criterios, se le cambiaría de grupo al usuario y se le preguntaba si deseaba conservar las anotaciones correspondientes a las palabras clave y a los autores del documento. Esto daba algunos problemas, puesto que al mover todas las anotaciones de un *theme* se esperaba que se eliminará dicho *theme* del antiguo grupo, pero WACline no tiene ninguna funcionalidad que facilite este proceso.
- Más de una alerta seguida:** Con la interpretación equivocada que se hizo, había que preguntar al usuario si deseaba mantener las palabras clave al cambiar de grupo y lo mismo con los autores. Para realizar esto, se utilizaban las alertas que emplea WACline. El problema, es que no había implementada ningún tipo de cola de alertas y al lanzar una alerta mientras ya se estaba mostrando otra, no ocurría nada y no se llegaría a mostrar esta segunda alerta.

- Error al anotar en un *code* nada más importar los criterios:** Al realizar una anotación para uno de los *codes* de un método de revisión recién importado, surgía un error. Esto se debía a que al no crear los *codes* de uno en uno, porque no sería eficiente, no se añadían a un objeto llamado *AnnotatedContentManager*. Este objeto contiene la información de los *theme* y los *codes* que se están empleando. Para solucionar esto, había que volver a iniciar el objeto para que así cargará correctamente el *theme* y los *codes* del método de revisión.
- Tamaño de cada criterio en canvas:** Al mostrar el canvas con los criterios de un método, como se muestra en la figura 6.15, surgió el problema de como determinar el número de filas y columnas para que quedará un canvas lo más estético posible. Si la raíz cuadrada del número de criterios era un número entero se utilizaría el mismo número de filas que de columnas. Sin embargo, si ese número era primo, esta tarea es más complicada. Para solucionar este problema, se utilizó la propiedad `grid-template-columns`<sup>6</sup> junto a `auto-fit`<sup>7</sup>, que determinan el máximo número de columnas que se pueden dado un tamaño mínimo y máximo para cada elemento.

## 6.4. Características versión final

En el capítulo 4 se ha mostrado un diagrama que contenía las características de Review&Go, el producto de partida. La figura 6.23 muestra el diagrama que contiene la nueva versión del producto. Se han marcado en rojo las características que han sido descartadas, en verde las que han sufrido modificaciones y en naranja las que han sido añadidas a la LPS y al producto.

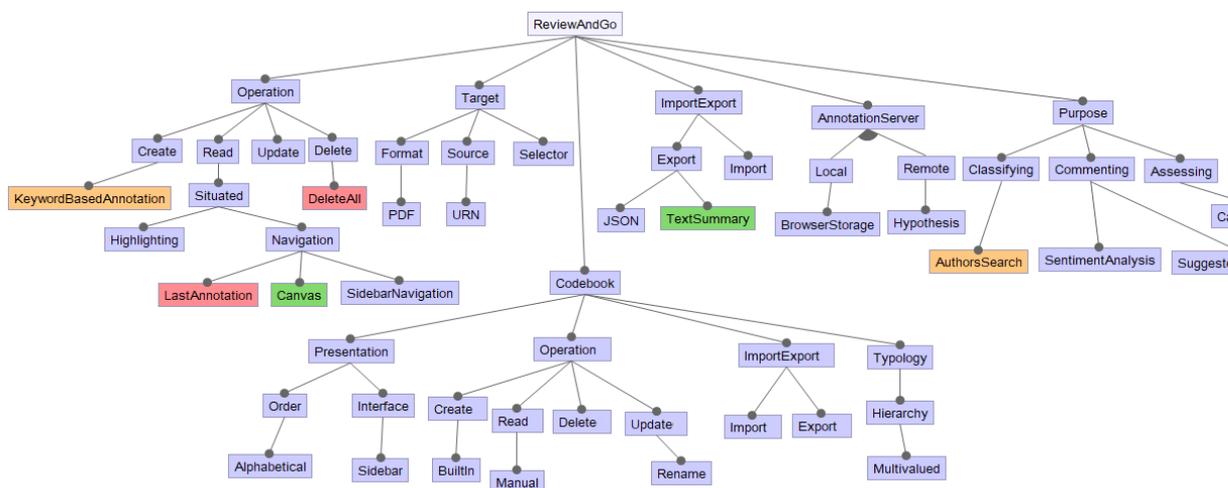


Figura 6.23: Subconjunto de características de WACline, Review&Go+

<sup>6</sup>`grid-template-columns`: <https://developer.mozilla.org/es/docs/Web/CSS/grid-template-columns>

<sup>7</sup>Tamaño automático de columnas: <https://css-tricks.com/auto-sizing-columns-css-grid-auto-fill-vs-auto-f>

## Desarrollo de iteraciones

Como se ha comentado previamente la fase de desarrollo consta de 3 iteraciones en las que se ha implementado un nuevo módulo. Estos son dichos módulos:

- Búsqueda de keywords (MBK)
- Información de autores (MIA)
- Métodos de revisión (MMR)

Sin embargo, debido a un problema de interpretación que surgió en la tercera iteración, la fase de desarrollo tuvo que ser ampliada y tuvo que realizarse otra iteración para poder finalizar el producto. A continuación se explicará como ha sido el desarrollo de las iteraciones.

### 7.1. Primera iteración: Búsqueda de keywords (MBK)

El objetivo de esta primera iteración era desarrollar una característica capaz de buscar ciertas palabras clave en el texto y marcarlas como anotaciones. Esta característica posteriormente se integraría con en el módulo de los métodos de revisión.

La iteración dio comienzo el 16 de marzo con la reunión de *Presentación de Módulo de Búsqueda de Keywords*, en la que se explicaron los requisitos que debía cumplir la característica. Tras la reunión, se realizó un análisis los requisitos del módulo y las tareas se dividieron de la siguiente forma:

- MBK.T1. Análisis del formato en el que se procesan los documentos.
- MBK.T2. Implementar búsqueda de palabras clave en el texto.
- MBK.T3. Implementar crear anotaciones para palabras clave encontradas.

| Tareas                                   | Estimación en horas | Horas reales |
|--|---------------------|--------------|
| <b>Módulo búsqueda de keywords (MBK)</b> | <b>20</b>           | <b>26</b>    |
| MBK.T1 Análisis procesamiento docs       | 2                   | 4            |
| MBK.T2 Búsqueda                          | 8                   | 7            |
| MBK.T3 Anotar                            | 10                  | 15           |

**Tabla 7.1:** Estimación de dedicaciones en horas MBK

Durante esta iteración se debía implementar el caso de uso ‘**Buscar keywords**’ descrito en el capítulo 3.

Al organizar las tareas en las que se divide el trabajo de esta iteración, se realizó una estimación de dedicaciones para cada tarea. En la tabla 7.1 se pueden observar las horas inicialmente estimadas y las horas reales que ha tomado realizar cada tarea.

La iteración concluyó con éxito. La aplicación permitía la anotación automática de las palabras. Sin embargo, todavía hubo que revisar un pequeño problema que no hacía visibles las anotaciones.

Para dar por finalizada la iteración se realizó una breve reunión de seguimiento, *Final primera iteración y establecer próximos objetivos*, en la que se revisó el estado del proyecto y los próximos pasos a tomar. Al finalizar la reunión se revisó la implementación del módulo y se solucionó el problema que impedía ver las anotaciones realizadas.

## 7.2. Segunda iteración: Información de autores (MIA)

El objetivo de esta segunda iteración era desarrollar una característica capaz de buscar información sobre los autores del documento a través de una API. Para ello, el revisor debería anotar el nombre y apellidos de los autores en un tema llamado ‘Authors’.

La iteración comenzó en la reunión de seguimiento que finalizaba la primera iteración. En ella se definieron los próximos pasos en el proyecto, que incluye la descripción de los requisitos de esta nueva característica. Tras la reunión, se organizó la planificación de la iteración definiendo las tareas para el desarrollo de este módulo:

- MIA.T1. Analizar la API del sitio web DBLP.
- MIA.T2. Diseño del prototipo de la vista de la información.
- MIA.T3. Implementar formulario para solicitar congreso al usuario.
- MIA.T4. Implementar vista de información de los autores marcados utilizando la API del DBLP.

Durante esta iteración se debían implementar los siguientes casos de uso de los descritos en el capítulo 3:

- Ver información de autores
- Elegir congreso

| Tareas                                     | Estimación en horas | Horas reales |
|--|---------------------|--------------|
| <b>Módulo información de autores (MIA)</b> | <b>35</b>           | <b>33</b>    |
| MIA.T1 API DBLP                            | 1                   | 2            |
| MIA.T2 Diseño                              | 2                   | 1            |
| MIA.T3 Formulario congreso                 | 12                  | 11           |
| MIA.T4 Vista de información                | 20                  | 19           |

**Tabla 7.2:** Estimación de dedicaciones en horas MBK

La estimación de horas realizada y las horas reales invertidas en cada una de las tareas puede verse representada en la tabla 7.2.

La iteración finalizó con la reunión *Demo del producto actual y presentación del tercer módulo*, en la que se hizo una demo del producto actual. Ahora la extensión es capaz de solicitar el congreso del artículo al usuario para que se le pueda mostrar la información extraída del DBLP, al haber anotado los nombres de los autores en el tema ‘Authors’.

### 7.3. Tercera iteración: Métodos de revisión (MMR)

El objetivo de esta iteración era añadir las funcionalidades para que el usuario pudiera utilizar las listas de criterios de revisión. El usuario podrá valorar el cumplimiento o no de dichos criterios y podrá generar y editar un informe de la revisión que ha realizado.

La iteración comienza con la reunión que dio por finalizada la segunda iteración. En ella se realizó una demo del producto actual y una presentación de este módulo. Tras la reunión se debían preparar el diseño de la solución para este módulo y comentarlo por correo. También se realizó un análisis para definir las tareas del módulo, que serían las siguientes:

- MMR.T1. Analizar métodos de revisión definidos en el Empirical Standart.
- MMR.T2. Diseño de prototipos de las vistas de validación de plantillas y revisión de reporte del artículo.
- MMR.T3. Realización del diagrama del proceso correspondiente a este módulo.
- MMR.T4. Implementar selección de método de revisión.
- MMR.T5. Implementar pantalla de validación de criterios.
- MMR.T6. Implementar vista de revisión del reporte final.

Durante esta iteración se implementarían los siguientes casos de uso de los descritos en el capítulo 3:

- Importar checklist
- Eliminar criterio
- Validar criterio

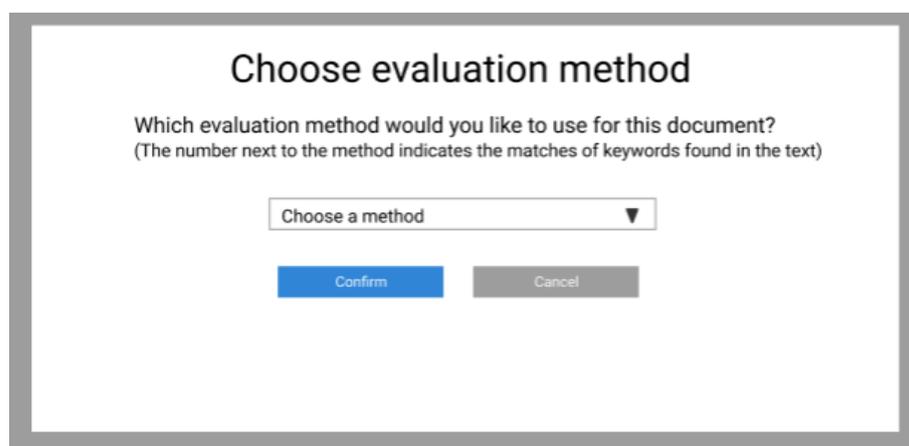
| Tareas                                       | Estimación en horas | Horas reales |
|--|---------------------|--------------|
| <b>Módulo métodos de revisión (MMR)</b>      | <b>85</b>           | <b>90</b>    |
| MMR.T1 Analizar métodos 'Empirical Standard' | 2                   | 2            |
| MMR.T2 Prototipos                            | 6                   | 7            |
| MMR.T3 Diagrama de proceso                   | 5                   | 7            |
| MMR.T4 Selección métodos de revisión         | 26                  | 28           |
| MMR.T5 Validación de criterios               | 26                  | 30           |
| MMR.T6 Reporte final                         | 20                  | 16           |

**Tabla 7.3:** Estimación de dedicaciones en horas MMR

- Modificar informe
- Guardar borrador

La estimación de horas realizada y las horas reales invertidas en cada una de las tareas puede verse representada en la tabla 7.3.

El siguiente paso fue diseñar los prototipos de las vistas que tendría este módulo, reflejados en las figuras 7.1, 7.2, 7.3, y 7.4. Se organizó una siguiente reunión, *Presentación de prototipos 3º módulo*, para presentar estos prototipos. Se comentaron cambios para mejorar los prototipos y también se realizó un esquema del diagrama que representara el proceso de evaluación de un artículo. Para la próxima reunión había que preparar el diagrama completo.



**Figura 7.1:** Prototipo elegir lista de criterios

El siguiente día se celebró la reunión para presentar el diagrama del proceso de revisión de un documento, *Presentación de diagrama de proceso de evaluación*. Este diagrama representaba los pasos que realizaría un revisor a la hora de evaluar un documento junto a los procesos que realiza el sistema y las pantallas de los prototipos que se mostrarían en los distintos estados.

Se realizó una presentación del diagrama y se comentaron algunos aspectos a mejorar para dar por finalizada la reunión. Tras realizar las modificaciones en el diagrama, que se

### 7.3. Tercera iteración: Métodos de revisión (MMR)

| The General Standard |                       |                       |                       |                       |
|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Essentials           | 👍 Research purpose    | 👎 Research purpose    | 👎 Research purpose    | 👎 Research purpose    |
|                      | 👍 Data collection     | 👍 Data collection     | 👍 Data collection     | 👍 Data collection     |
|                      | 👎 Contribution to BoK | 👍 Contribution to BoK | 👍 Contribution to BoK | 👍 Contribution to BoK |
|                      | 👍 Supported claims    | 👎 Supported claims    | 👎 Supported claims    | 👍 Supported claims    |
| Nice to have         | 👎 Contribution to BoK | 👍 Supported claims    | 👍 Supported claims    |                       |
|                      | 👎 Contribution to BoK | 👎 Contribution to BoK | 👍 Supported claims    |                       |
| Additional           | 👍 Supported claims    |                       | 👎 Contribution to BoK |                       |

Figura 7.2: Prototipo canvas lista de criterios

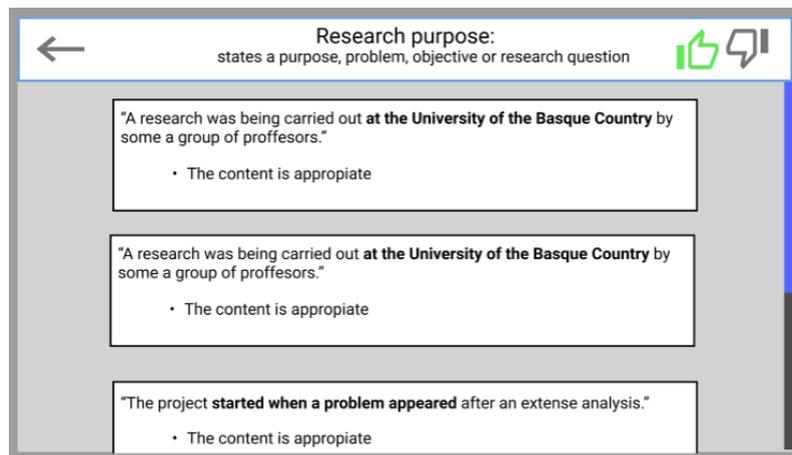


Figura 7.3: Prototipo evaluación de criterio

presenta en el capítulo 3, y enviarlo por correo comenzó la fase de implementación del módulo.

En esta fase de implementación, además de implementar los nuevos casos de uso, también se realizaron mejoras estéticas en la interfaz para visualizar la información de los autores y se mejoró y adaptó la búsqueda y anotación de palabras clave para que fuera más eficiente.

Unos días más tarde, se realizó la reunión *Demo de la tercera iteración y boceto del esquema de la memoria*, en la que se hizo una demo del producto una vez había finalizado su fase de implementación. Sin embargo, debido a un malentendido de los requisitos, que se menciona en el capítulo 6, había que realizar varias modificaciones en el producto. En esta reunión se decidió que Haritz tenía que pensar acerca de cómo se podrían fusionar el antiguo canvas de WACline con la nueva propuesta que se había realizado. Esta propuesta debía ser presentada aproximadamente dos semanas más tarde. En la reunión también se mostró un primer borrador de la memoria. Además, se decidió que ahora se iba a centrar la atención en la redacción de la memoria y que se iba a dejar de lado la implementación hasta que estuviera la nueva propuesta. Habría que realizar una nueva iteración para poder

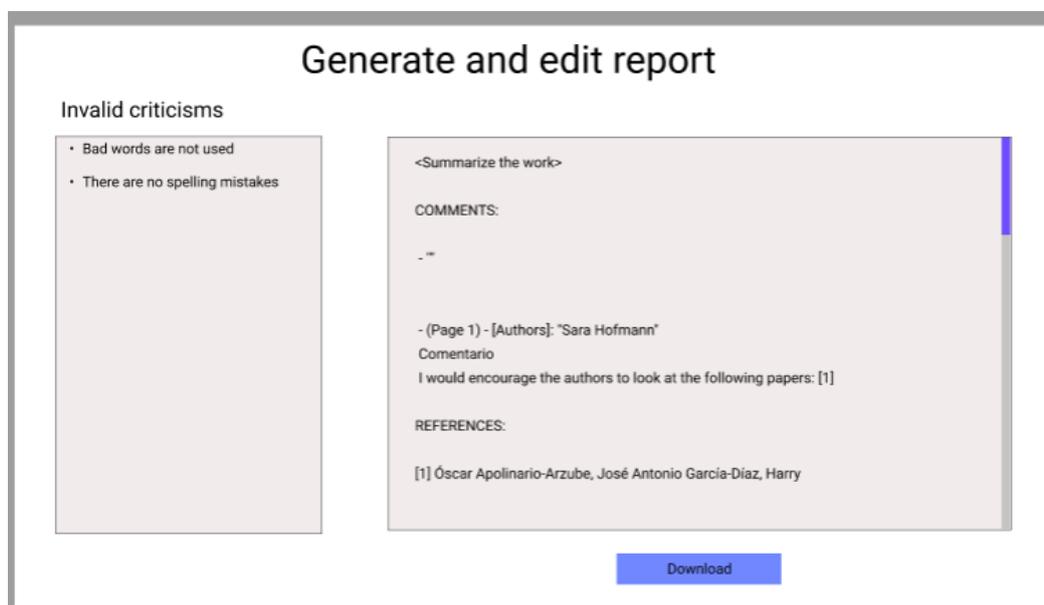


Figura 7.4: Prototipo modificar informe

realizar las modificaciones que necesitaba el producto.

#### 7.4. Cuarta iteración: Corrección del módulo Métodos de revisión (CMR)

Esta iteración da comienzo el 19 de mayo cuando, a través de correo electrónico, se plantea la nueva propuesta y las modificaciones a realizar en la implementación. Estas eran las tareas a realizar:

- **CMR.T1:** No cambiar a otro grupo cuando el usuario importa un método de revisión.
- **CMR.T2:** Permitir importar más de un método de revisión en un codebook.
- **CMR.T3:** Permitir seleccionar criterios al elegir un método de revisión.
- **CMR.T4:** Fusionar antiguo canvas con el nuevo canvas de métodos de revisión.
- **CMR.T5:** Realizar pequeñas mejoras estéticas.

Durante esta iteración se implementaría el caso de uso 'Seleccionar criterios'. También se debían realizar algunos cambios en la implementación del caso de uso 'Importar checklist', ambos descritos en el capítulo 3.

Una vez organizada la lista de tareas a realizar se hicieron las estimaciones presentadas en la tabla 7.4. También se reflejan las horas reales empleadas para cada tarea.

A pesar de tener que realizar estas tareas de implementación, se le continuó dando una gran dedicación a la memoria. Durante la primera semana de la iteración se pudieron

#### 7.4. Cuarta iteración: Corrección del módulo Métodos de revisión (CMR)

| Tareas   | Estimación | Horas reales |
|--|------------|--------------|
| <b>Corrección de Métodos de revisión (CMR)</b>       | <b>15</b>  | <b>18</b>    |
| CMR.T1 Mantener grupo al importar método de revisión | 2          | 4            |
| CMR.T2 Importar más de un método de revisión         | 4          | 3            |
| CMR.T3 Selección de criterios                        | 5          | 7            |
| CMR.T4 Fusionar antiguo canvas y nuevo               | 2          | 3            |
| CMR.T5 Mejoras estéticas                             | 2          | 1            |

**Tabla 7.4:** Estimación de dedicaciones en horas CMR

realizar las primeras tres tareas mencionadas. La segunda semana se pudo completar la cuarta tarea ultimando detalles de la interfaz.

Al haber finalizado las tareas de implementación de esta iteración, se celebró una reunión para realizar una demo del producto generado hasta el momento, *Demo del producto de la cuarta iteración*. En ella se comentaron algunos aspectos de la interfaz que necesitaban reajustarse.

El siguiente paso era añadir los cambios detallados a la aplicación y celebrar una reunión para presentar lo que sería el producto final. La fase de desarrollo concluyó en dicha reunión, *Demo del producto final*. En ella se realizó una demo del producto final resultante y se comentaron detalles a tener en cuenta para finalizar la memoria del proyecto.



## Seguimiento y Control

Este apartado presenta la tabla 8.1, en la que se muestran los paquetes de trabajo descritos en el capítulo 2. Se muestran las desviaciones respecto a las horas estimadas inicialmente.

Se puede ver reflejada una clara desviación a causa de la aparición de una cuarta iteración debido a los problemas explicados los capítulos 6 y 7. Se produjo la necesidad de realizar cambios para mejorar la usabilidad del usuario a la hora de realizar anotaciones sobre los criterios de revisión y el enfoque que se le había dado a las listas de criterios..

Otra de las desviaciones no tan significativas, es la del módulo de búsqueda keywords (MBK). Como se comenta en los problemas descritos en el capítulo 6, surgió un problema con el subrayado y no se pudo detectar el origen hasta comentar el problema con Haritz porque las anotaciones se creaban correctamente. También se le dio demasiada importancia al problema de las tildes, se emplearon demasiadas horas en tratar de resolver un problema poco relevante.

| Tareas   | Estimación | Horas reales | Desviación   |
|--|------------|--------------|--------------|
| <b>Proyecto Fin de Grado</b>                       | <b>310</b> | <b>339.5</b> | <b>9.5 %</b> |
| Fase de gestión (G)                                | 48         | 43           | -11 %        |
| Planificación (P)                                  | 18         | 17           | -5 %         |
| P.T1 Requisitos                                    | 3          | 4            | 33 %         |
| P.T2 Entorno de desarrollo y planificación inicial | 12         | 10           | -16 %        |
| P.T3 Actualización de la planificación             | 3          | 3            | 0 %          |
| Seguimiento y control (SyC)                        | 30         | 26           | -13 %        |
| SyC.T1 Reuniones                                   | 15         | 13           | -13 %        |
| SyC.T2 Configuración Trello                        | 2          | 2            | 0 %          |
| SyC.T3 Recogida de información del desarrollo      | 7          | 6            | -15 %        |
| SyC.T4 Contraste de seguimiento                    | 6          | 5            | -16 %        |
| Fase de gestión de conocimientos (GC)              | 37         | 33           | -11 %        |
| Estudio de competencias (EC)                       | 15         | 14           | -7 %         |
| EC.T1 pure::variants                               | 7          | 7            | 0 %          |
| EC.T2 Extensiones Chrome                           | 5          | 4            | -20 %        |
| EC.T3 JavaScript ES6                               | 3          | 3            | 0 %          |
| Estudio de proyecto base (EPB)                     | 22         | 19           | -13 %        |
| EPB.T1 Análisis SPL WACline                        | 15         | 13           | -13 %        |
| EPB.T2 Dudas                                       | 3          | 3            | 0 %          |
| EPB.T3 Solucionar issue 88                         | 4          | 3            | -25 %        |
| Fase de desarrollo (DP)                            | 140        | 167          | 20 %         |
| Módulo búsqueda de keywords (MBK)                  | 20         | 26           | 30 %         |
| Módulo información de autores (MIA)                | 35         | 33           | -5 %         |
| Módulo métodos de revisión (MMR)                   | 85         | 90           | 5 %          |
| Corrección del módulo Métodos de revisión (CMR)    | 0          | 18           | 100 %        |
| Fase de documentación (D)                          | 85         | 96.5         | 13 %         |
| Realización de la memoria (RM)                     | 70         | 78           | 11 %         |
| RM.T1 Configurar entorno LaTeX                     | 3          | 3            | 0 %          |
| RM.T2 Redactar                                     | 67         | 75           | 12 %         |
| Preparación de la defensa (DEF)                    | 15         | 18.5         | 23 %         |
| DEF.T1 Póster                                      | 2          | 2.5          | 25 %         |
| DEF.T2 Documento de apoyo                          | 5          | 6            | 20 %         |
| DEF.T3 Preparar                                    | 8          | 10           | 25 %         |

Tabla 8.1: Desviaciones en la estimación de horas definidas

# Conclusiones

Este apartado consta de las conclusiones obtenidas del proyecto y las posibles mejoras a implementar en un futuro.

## 9.1. Conclusiones

### 9.1.1. Conclusiones a nivel técnico

El proyecto ha finalizado de manera exitosa, cumpliendo con los objetivos establecidos en el capítulo 2. Se han desarrollado los tres módulos presentados a pesar de los problemas surgidos durante el proyecto y cambios que han podido surgir en los requisitos de alguno de los módulos. La sensación de obtenida al finalizar este proyecto es muy buena, se ha sabido adaptarse a las peticiones realizadas durante el proyecto en plazos cortos y se considera que se ha desarrollado un buen producto.

### 9.1.2. Conclusiones a nivel personal

A nivel personal el resultado del proyecto ha sido muy positivo. El poder participar en un proyecto de gran tamaño ha sido una gran oportunidad de la que se han aprendido muchas cosas. El proyecto ha resultado ser muy interesante y no se ha perdido la motivación en ningún momento. Ha sido un reto trabajar sobre un dominio del que no se sabía demasiado para intuir que le resultaría más cómodo al usuario y obtener la visión de cómo se realizan las revisiones de artículos.

Además, se ha podido aprovechar para obtener conocimientos sobre tecnologías interesantes, como las extensiones del navegador, y se ha podido profundizar en las LPS para poder ver su potencial en un proyecto de tamaño considerable.

También se ha aprendido mucho acerca de lo que sería el trato con un cliente. Las reuniones en las que se realizaba una demo eran tratadas como una presentación del producto actual al cliente, que sería el tutor, escuchando sus peticiones y haciendo los cambios correspondientes,

## 9.2. Posibles mejoras

Se han detectado posibles mejoras que podrían ser interesantes de implementar en un futuro, puesto que no ha sido posible desarrollarlas durante este proyecto.

Una de las mejoras podría ser eliminar las anotaciones de un *theme* o *code* una vez estos han sido eliminados. El usuario podría decidir si eliminar las anotaciones o mantenerlas sin que correspondan a ningún *theme* o *code*. Esta funcionalidad puede ser interesante para las *keywords*. Una vez el usuario ya las ha comprobado, puede llegar a molestarle tener tantas anotaciones. Además, en los documentos en los que se han detectado muchas *keywords* tener tantas anotaciones puede hacer que el navegador funcione más lento.

Otra funcionalidad que podría ser interesante es permitirle al usuario añadir sus propios criterios para que los pueda evaluar. Ahora el usuario puede crear nuevos *codes* en los *themes* correspondientes a los niveles de los criterios (*essential*, *desirable* y *extraordinary*). Sin embargo, aunque los añade no serán visualizados en el canvas de revisión de criterios y no podrá evaluarlos. También podría evaluarse la opción cargarlos a través de un servicio de terceros o que el propio congreso provea los criterios para la revisión.

También podría mejorarse la presentación y edición del informe para que el usuario pueda ver la información de forma más clara y precisa. Podría implementarse de manera que las anotaciones se muestran como tarjetas, tal y como se ha hecho para la interfaz de evaluar uno de los criterios, con la posibilidad de editar los comentarios.

La búsqueda de palabras clave no le da opción al usuario a elegir que listas de criterios que le interesan para que se busquen sus palabras clave. Tampoco le permite añadir palabras que le parezcan de interés localizar y anotar. Esto podría llegar a ser interesante para el usuario.

# **Anexos**



## A. Extensiones Chrome

Las extensiones de Chrome[3] son aplicaciones que se instalan en el navegador y que añaden o mejoran sus funciones. Utilizan tecnologías web como HTML, JavaScript y CSS y pueden modificar la interfaz del usuario o añadir el servicio de algún sitio al navegador. La interfaz de usuario de la aplicación debe ser minimalista para no obstaculizar la visión del usuario. El código de la extensión debe contener algunos archivos de código que interactuaran durante la ejecución del programa, esto se ve reflejado en la figura 1.

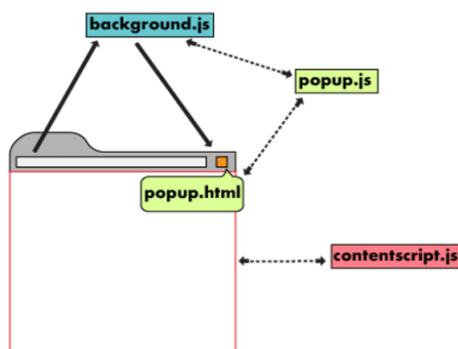


Figura 1: Esquema elementos extensión [3]

Dentro del código de una extensión de navegador nos encontraremos con un fichero llamado *Manifest.json*. Este fichero contiene la información acerca de la extensión. Como mínimo debe contener el nombre, la versión de la aplicación y la versión de dicho manifiesto. También se especifican en él los permisos, el icono de la aplicación que se mostrará en la barra de herramientas del navegador y el script que se ejecuta de fondo para gestionar la aplicación y manejar los eventos.

Cuando el usuario clic en el icono de la extensión en la barra del navegador, la aplicación cargará un fichero llamado *popup.html*. Este fichero suele tener una lógica JavaScript asignada que se encarga de activar y desactivar la aplicación.

Las extensiones utilizan una programación basada en eventos. Lo cual quiere decir que la lógica de la aplicación se estructura en eventos que se lanzan y son procesados por los elementos que estén a la escucha de dicho evento. La aplicación inicializa los eventos gracias al script *background*. Este script se mantiene inactivo hasta que se lanza uno de los eventos que está escuchando.

Las aplicaciones que realizan acciones de lectura o escritura sobre las páginas web utilizan el *contentScript*. Este fichero contiene código JavaScript que se ejecuta en el

## A. EXTENSIONES CHROME

---

contexto de la página que ha sido cargada en el navegador y hace uso del DOM<sup>1</sup> para leer o modificar la web.

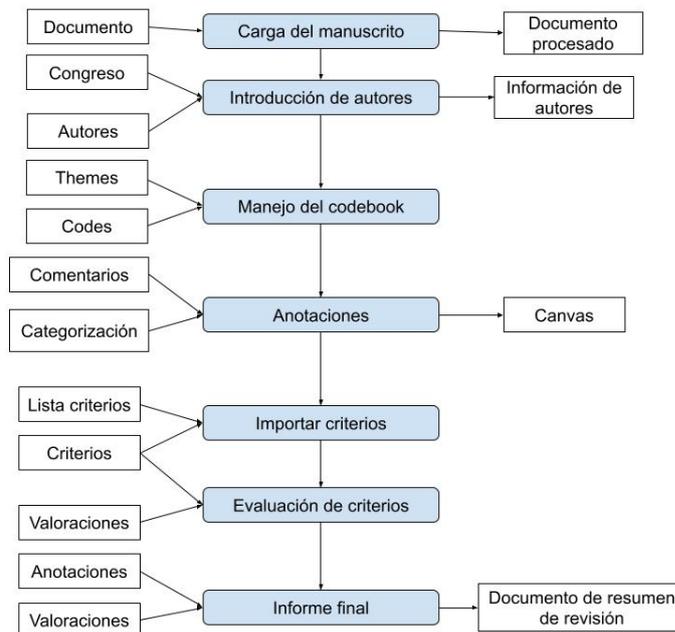
---

<sup>1</sup>DOM: El Modelo de Objetos del Documento es una interfaz de programación de aplicaciones (API) para documentos HTML y XML

## B. Demo del producto

Este capítulo describe una demo de la revisión de un artículo, realizada mediante la herramienta generada como resultado de este proyecto. Se ha decidido comenzar con este capítulo para que se puedan entender mejor los siguientes. Hay un vídeo demo disponible en YouTube mediante el siguiente enlace: <http://tiny.cc/demoReviewAndGoPlus>. **Se recomienda vivamente ver este vídeo para entender el proyecto de forma más sencilla.**

La figura 2 contiene un esquema que refleja las fases por las que pasará el manuscrito en la revisión que se realiza en la demo.



**Figura 2:** Proceso de revisión de la demo

Se comenzará indicando como se debe cargar el documento e iniciar la aplicación para poder comenzar la revisión.

El siguiente paso será seleccionar el congreso al que pertenece el artículo y localizar a los autores del manuscrito en el documento. Entonces se podrá consultar la información de los autores.

Después se procederá a explicar las acciones de manejo del *codebook* para mostrar cómo se pueden crear y eliminar tanto los *themes* como los *codes*. Una vez se han creado *themes*

y codes, se procederá a realizar anotaciones sobre ellos. También se añadirán comentarios a dichas anotaciones y serán clasificadas en alguno de los tres niveles disponibles. Cuando tengamos las anotaciones, se mostrará un *canvas* que contiene un resumen y una clasificación de las anotaciones realizadas en sus correspondientes themes y codes.

Se continuará explicando cómo importar criterios que pertenecen a una lista de las definidas en el *Empirical Standard*. Cuando se hayan importado los criterios se realizarán más anotaciones, como se ha explicado anteriormente, y se dará paso a la evaluación de los criterios importados, dándolos por aprobados o suspendidos.

Una vez se ha finalizado con la evaluación, se procederá a mostrar el informe final, que resume la revisión realizada en el manuscrito. Este informe podrá ser editado, descargado y guardado como borrador.

### Carga del manuscrito

Para comenzar con la revisión, lo primero que se va a hacer es cargar el documento PDF que se va a revisar en el navegador y clicar el icono de la extensión, como muestra la figura 3. Entonces, se comenzará a procesar el documento y la aplicación iniciará la búsqueda y anotación de palabras clave.

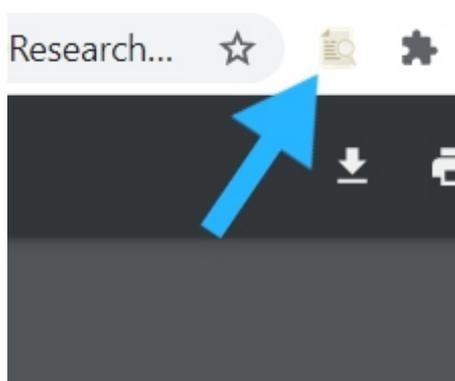


Figura 3: Icono extensión Review&Go

### Información de autores

En este apartado vamos a ver como se realiza el proceso de obtener información acerca de los autores del documento.

Lo primero que hace el sistema después de cargar el documento, es solicitar que introduzcamos el congreso al que pertenece el documento como muestra la figura 4. Una vez introducido, nos permitirá consultar la información de los autores anotados pulsando en el quinto botón de la barra lateral que se muestra en la figura 5. Como al principio no tenemos ningún autor anotado, se mostrará la pantalla que muestra la figura 6. Anotaremos los autores del artículo y al volver a entrar se nos mostrará la información, como se refleja en la figura 7.

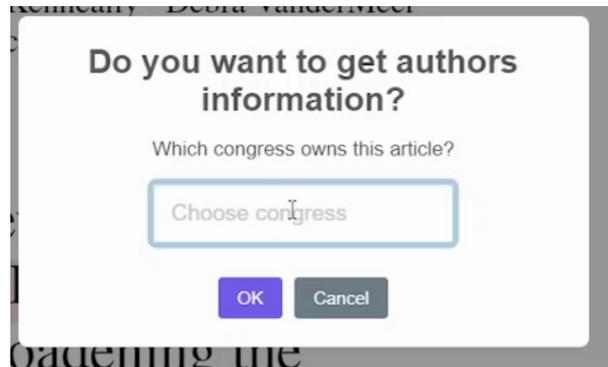


Figura 4: Seleccionar congreso del documento

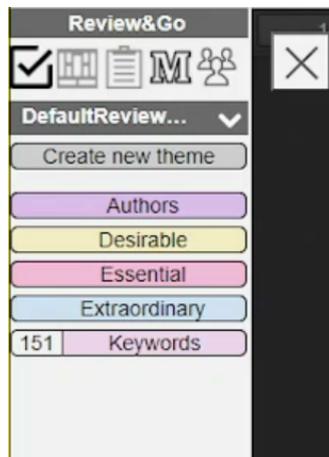


Figura 5: Barra lateral con *themes* predefinidos



Figura 6: Información de autores (no authors)

## Anotaciones y manejo del codebook

En este apartado vamos a ver como añadir y modificar nuevos *themes* y *codes* en el *codebook* para poder clasificar las anotaciones que realicemos. También se va a explicar cómo se realizan anotaciones y las opciones disponibles.

Tal y como muestra la figura 5, el codebook de momento solo contiene los themes predefinidos: *Essential*, *Desirable*, *Extraordinary*, *Keywords* y *Authors*. Crearemos un nuevo theme, pulsando el botón de 'Create new Theme'. Entonces, se nos mostrará el menú que muestra la figura 8 donde podremos especificar el nombre y una descripción que le queremos dar al theme.

Ahora podemos realizar múltiples acciones relacionadas con el theme. Si hacemos clic derecho sobre él, se nos mostrarán tres opciones, tal y como refleja la figura 9.

| Author                 | Publications                   | DBLP Profile  |
|------------------------|--------------------------------|---|
| Oth Inter              | This author could not be found |   |
| Brian Donnellan        | 5 publications                 | <a href="https://dblp.org/pid/57/288">https://dblp.org/pid/57/288</a>   |
| Markus Helfert         | 0 publications                 | <a href="https://dblp.org/pid/17/2368">https://dblp.org/pid/17/2368</a> |
| Jim Kenneally          | 3 publications                 | <a href="https://dblp.org/pid/69/6642">https://dblp.org/pid/69/6642</a> |
| Debra E. VanderMeer    | 6 publications                 | <a href="https://dblp.org/pid/84/5479">https://dblp.org/pid/84/5479</a> |
| Marcus A. Rothenberger | 5 publications                 | <a href="https://dblp.org/pid/12/631">https://dblp.org/pid/12/631</a>   |
| Robert Winter 0001     | 3 publications                 | <a href="https://dblp.org/pid/84/285">https://dblp.org/pid/84/285</a>   |

Figura 7: Información de autores

You are creating a new theme:

New theme name

Please type a description that describes this theme...

OK Cancel

Figura 8: Crear nuevo *theme*



Figura 9: Opciones *theme*

- En caso de que seleccionemos '**Create new code**', se nos abrirá un menú similar al de crear un nuevo tema, como muestra la figura 10, para especificar el nombre y la descripción del nuevo code. Una vez hemos introducido la información y hemos aceptado, nuestro code aparecerá en el menú lateral como se muestra en la figura 11.
- Si seleccionamos '**Modify theme**', volveremos a tener un menú igual que el de crear un nuevo theme con la información actual. Podremos modificar tanto el nombre como la descripción.
- Si seleccionamos '**Remove theme**', se nos mostrará un mensaje que nos solicitará confirmación para eliminar theme.

Al igual que los themes, los codes creados también podemos modificarlos y eliminarlos.

La figura 12 muestra como podemos realizar nuevas anotaciones al seleccionar un fragmento de texto y clicar en el botón de un theme o un code. El número al lado del theme

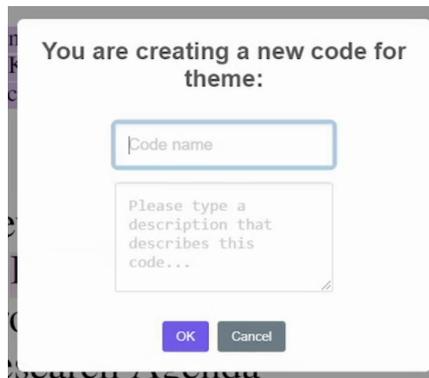


Figura 10: Crear nuevo *code*

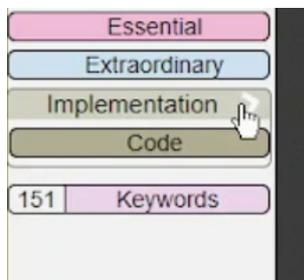


Figura 11: Nuevo *code* creado

o del *code* refleja el número de anotaciones que contiene. Si hacemos clic derecho sobre una anotación, se nos mostrarán dos opciones como en la figura 13.

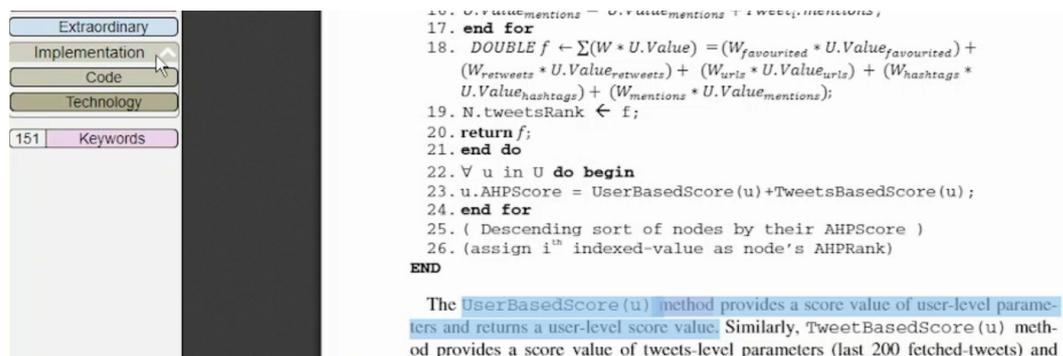


Figura 12: Crear anotación

- Si pulsamos en la opción '**Delete annotation**', se nos mostrará una alerta para confirmar si deseamos eliminar la anotación.
- Si hacemos clic en '**Comment**', se nos mostrará un nuevo menú como muestra la figura 14. La figura 15 muestra las tres categorías a elegir que tendremos en el desplegable: *Minor weakness*, *major weakness* y *Strength*. Añadiremos el comentario que queremos realizar sobre la anotación y en el último campo podemos enlazar documentos del DBLP que queramos sugerir. Tal y como muestra la figura 16, dispondrá de autocompletado.

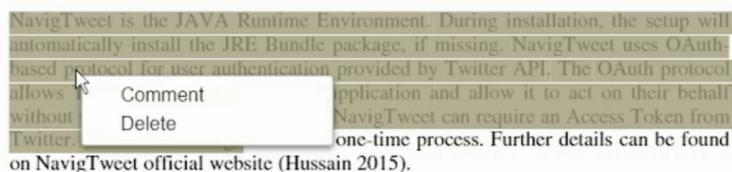


Figura 13: Opciones anotación

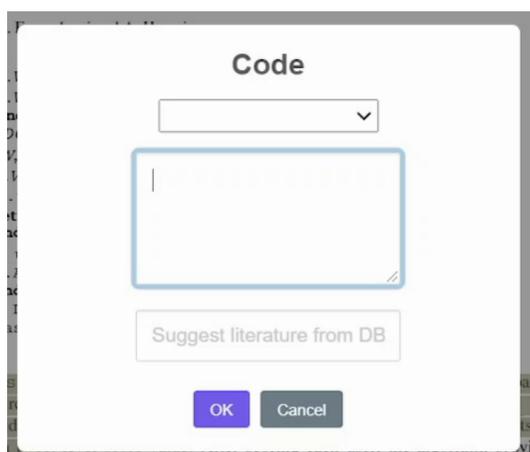


Figura 14: Comentar anotación



Figura 15: Categorías anotación

Una vez hemos realizado algunas anotaciones, accederemos al *Canvas*. Tal y como muestra la figura 17, es una pantalla que nos muestra un resumen y una clasificación por temas de las anotaciones que hemos realizado. Como se puede observar, el color de las anotaciones cambia si hemos seleccionado una de las tres categorías al realizar comentarios.

## Evaluación de criterios

En este apartado vamos a ver cómo importar criterios y realizar su evaluación.

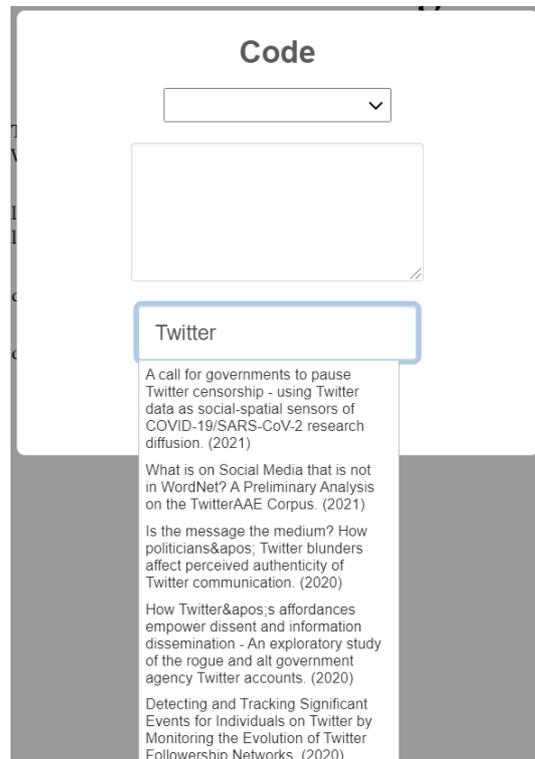


Figura 16: Autocompletado al sugerir documentos

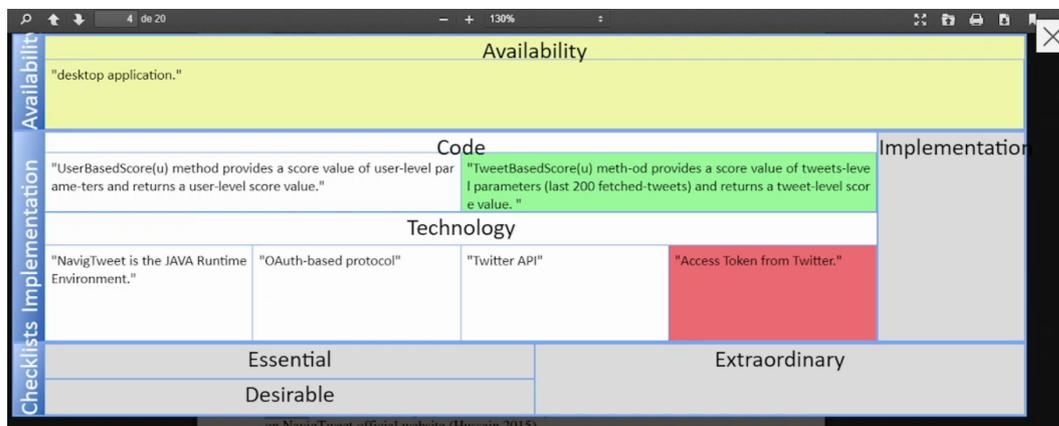


Figura 17: Canvas anotaciones

Hay que recordar que al principio se ha comentado que al abrir el documento se buscan y anotan las palabras clave. Además, como se ha podido ver en la imagen 5, el theme 'Keywords' ya contiene algunas anotaciones. Estas anotaciones nos servirán para que el sistema nos recomiende listas de criterios en función del número de palabras que ha encontrado para cada lista.

Lo primero que haremos será acceder al menú de selección de listas de métodos. Como muestra la imagen 18, se nos mostrará un desplegable que contiene las listas de criterios junto al número de coincidencias de palabras clave para esa lista. La figura 19 muestra como las listas han sido ordenadas de acuerdo a ese número.

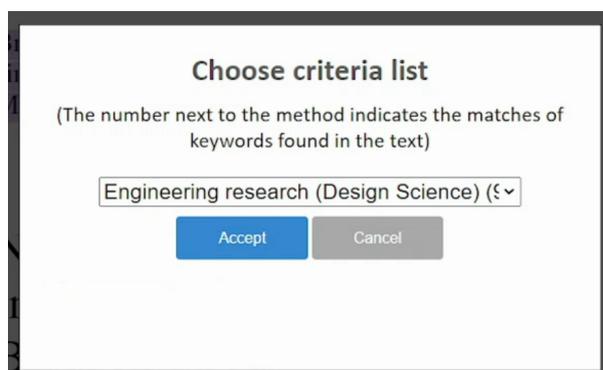


Figura 18: Seleccionar lista de criterios

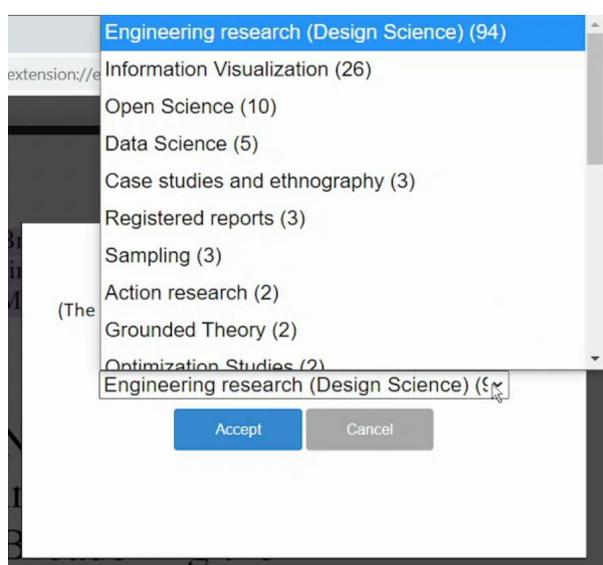


Figura 19: Seleccionar lista de criterios (desplegado)

También podemos acceder al menú de selección de listas si creamos un nuevo theme cuyo nombre coincida con uno de estos filtros: *Evaluation*, *Quantitative*, *Visualization* o *IRR*. Sin embargo, en este caso el desplegable de listas habrá hecho un filtrado y no mostrarán todas ellas tal y como muestra la figura 20.

Como muestra la figura 21, una vez hemos seleccionado una lista accederemos a la siguiente pantalla donde aparecen los criterios de la lista que hemos seleccionado clasificados en tres niveles: *Essential*, *Desirable* y *Extraordinary*. Podremos seleccionar los criterios que deseamos importar para realizar la revisión. Cuando aceptemos, los criterios se habrán añadido como nuevos *codes* al tema que corresponde su nivel, tal y como se puede observar en la figura 22.

Después de realizar las anotaciones necesarias en el documento, pasaremos a evaluar los criterios. Para ello accederemos al *Canvas* y haremos clic en una de las casillas que corresponde a uno de los tres niveles de los criterios. Ahora se nos mostrará una nueva pantalla con un resumen de los criterios añadidos para ese nivel, como refleja la figura 23, organizados por listas. En este caso podemos ver como se han añadido seis criterios

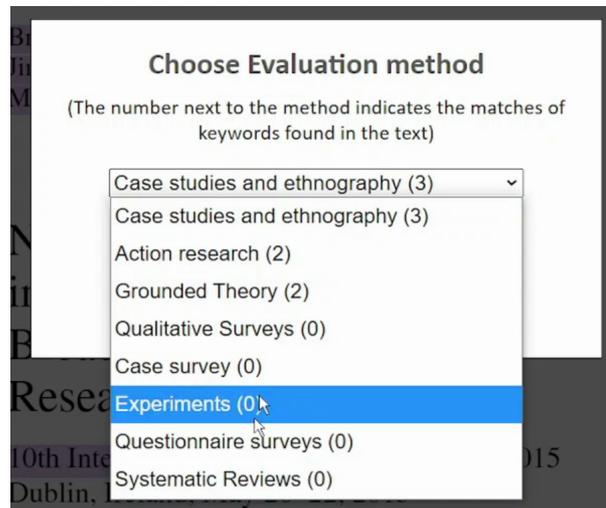


Figura 20: Seleccionar lista de criterios (filtro *Evaluation*)

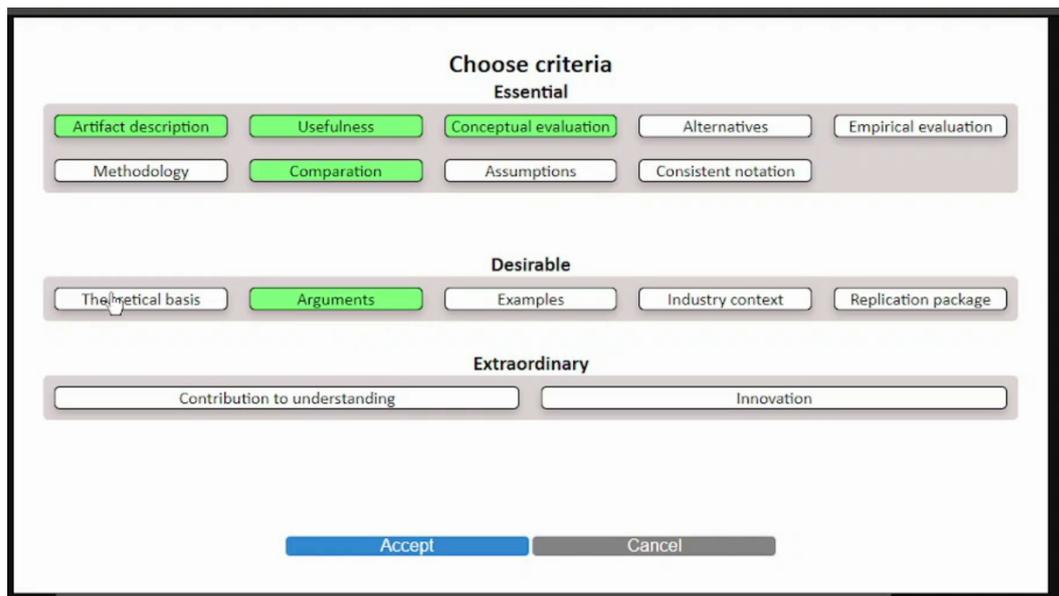


Figura 21: Seleccionar criterios de una lista

esenciales de dos de las listas: ‘Artifact description’, ‘Usefulness’ y ‘Comparison’ de la lista ‘Engineering research (Design Science)’ y ‘Case Justification’, ‘Site description’ y ‘Data source’ de la lista ‘Case studies and ethnography’. De momento se pueden observar dos colores para los criterios: ‘naranja’ para los que contienen anotaciones y ‘blanco’ para los que no. Para acceder al panel de criterios del nivel *Essential*, podemos atajar utilizar el primer botón del sidebar, que se muestra en la figura 5.

Haciendo clic en uno de los criterios, accederemos al menú donde podremos evaluarlo. La figura 24 muestra como este menú contiene las anotaciones junto a su comentario, en caso de haberlo añadido, para el *code* correspondiente a dicho criterio. Al hacer clic en una de las anotaciones, se nos muestra la anotación en el documento. Como refleja la figura 25, veremos la parte del documento donde se encuentra la anotación, subrayada de color rosa



Figura 22: Sidebar con nuevos criterios

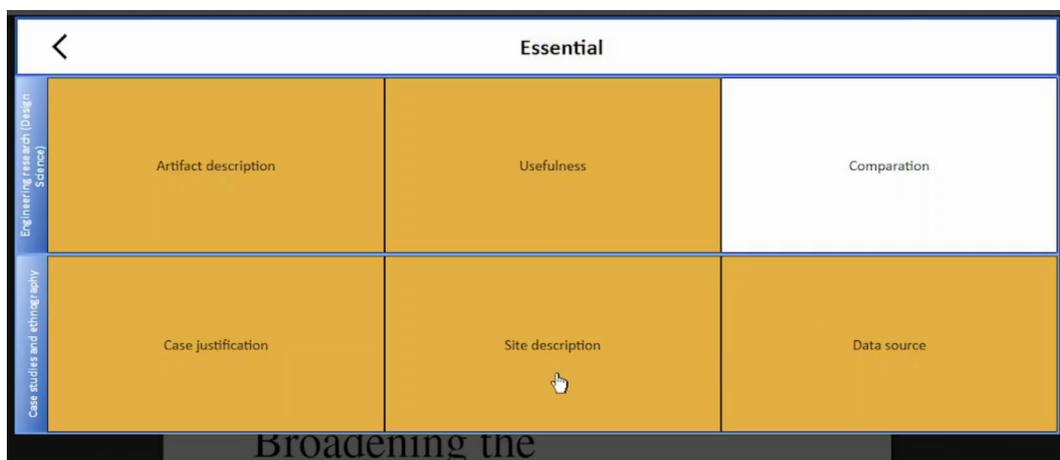


Figura 23: Canvas criterios (antes de evaluar)

en este caso, a la que hemos accedido y podremos movernos libremente por el documento. Podremos volver al menú de evaluación del criterio haciendo clic en la flecha. Con los iconos de 'like' y 'dislike' que se muestran en la figura 24, vamos a evaluar el criterio para darlo por aprobado o suspendido. Cuando modificamos la valoración que realizamos en un criterio cambiará el color con el que se muestra en la pantalla anterior.

Una vez hemos realizado la evaluación de criterios, el menú de uno de los niveles de criterios se mostrará similar a la figura 26. Podemos comparar esta imagen con la figura 23, que nos mostraba esta misma pantalla antes de haber realizado la evaluación. Los criterios que han sido aprobados ahora se muestran en verde y los que no en rojo. En caso de haber dejado algún criterio sin evaluar, habría seguido manteniendo el color blanco o naranja:

depende de si contiene anotaciones o no.

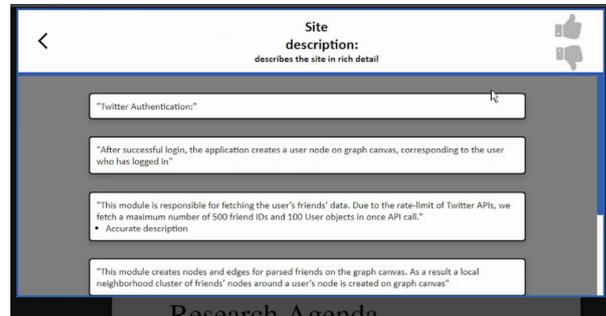


Figura 24: Evaluación de criterio 'Site description'

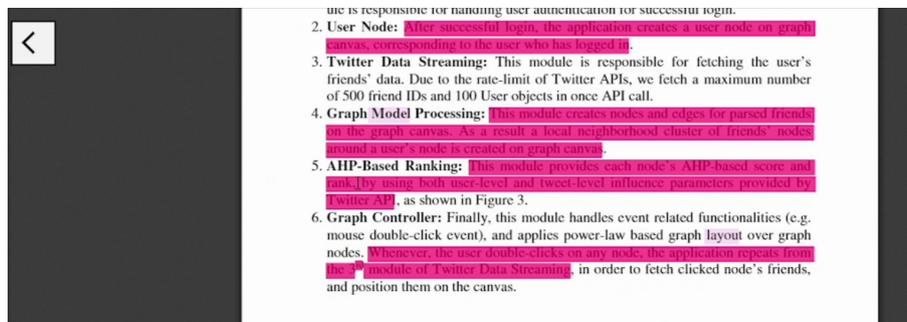


Figura 25: Contexto de anotación

|                                       | Essential        |                      |                    |
|---------------------------------------|------------------|----------------------|--------------------|
| Engineering research (Design Science) | Usefulness       | Artifact description | Comparison         |
| Clear factors and methodology         | Site description | Data source          | Case justification |

Figura 26: Canvas criterios (evaluados)

## Informe

Este informe será generado automáticamente por la aplicación y contendrá un resumen de la revisión realizada.

Para acceder al informe haremos, clic en el tercer icono del *sidebar* que ya se mostraba en la figura 5. Se nos va a mostrar la pantalla que refleja la figura 27.

La columna de la izquierda contiene las críticas invalidas correspondientes a las listas de criterios que hemos importado. Por otra parte, el apartado de la derecha muestra el informe

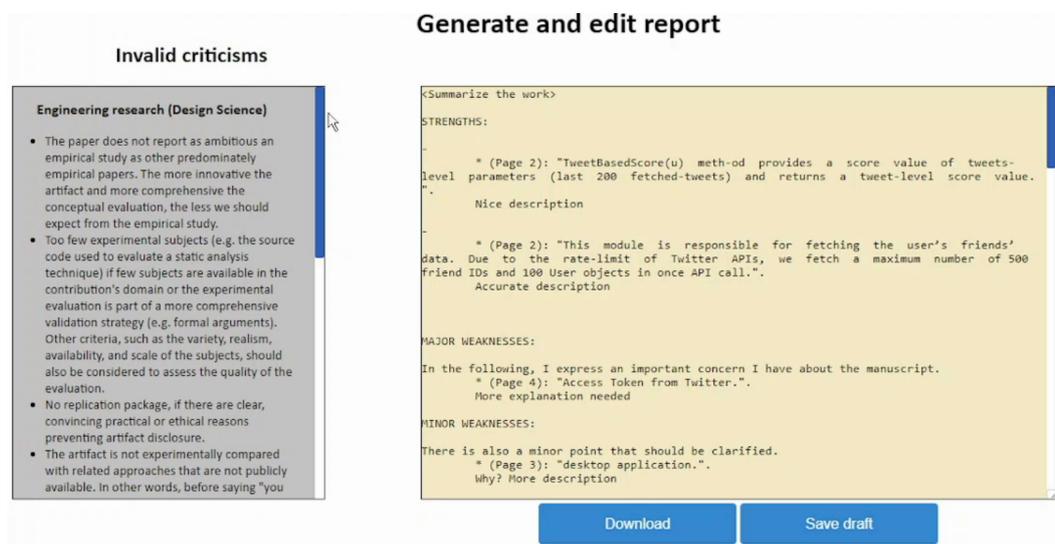


Figura 27: Edición de informe

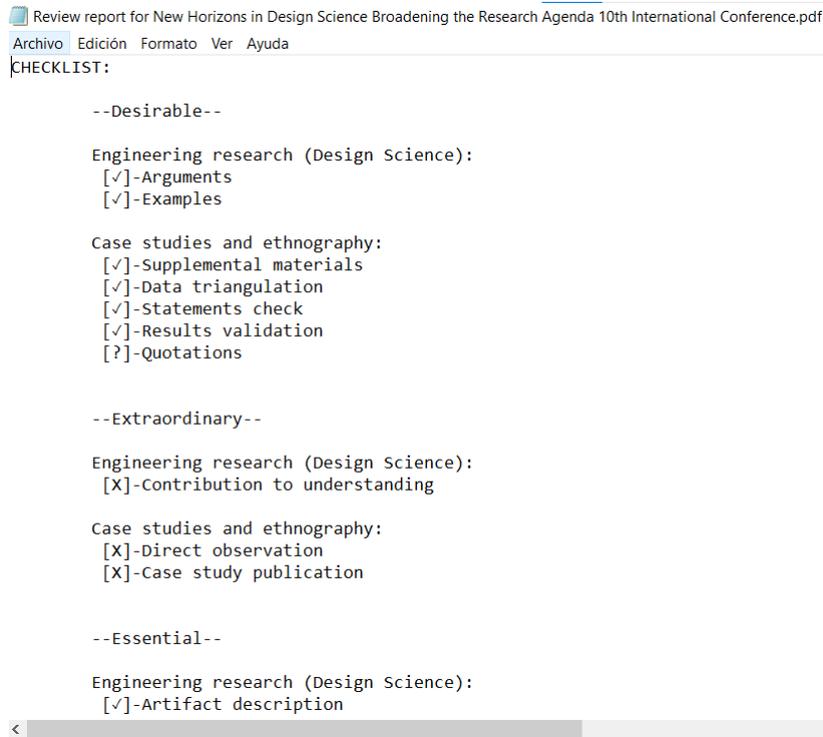
con las anotaciones realizadas y sus comentarios. Se puede ver como las anotaciones que hemos categorizado aparecen al comienzo del informe.

Cuando realizamos cambios en los comentarios del informe y pulsamos 'Save draft', se nos mostrará el mensaje de la figura 28. Ahora, a pesar de que cerremos el informe los cambios que hemos realizado se habrán guardado y se mantendrán cuando lo volvamos a abrir.

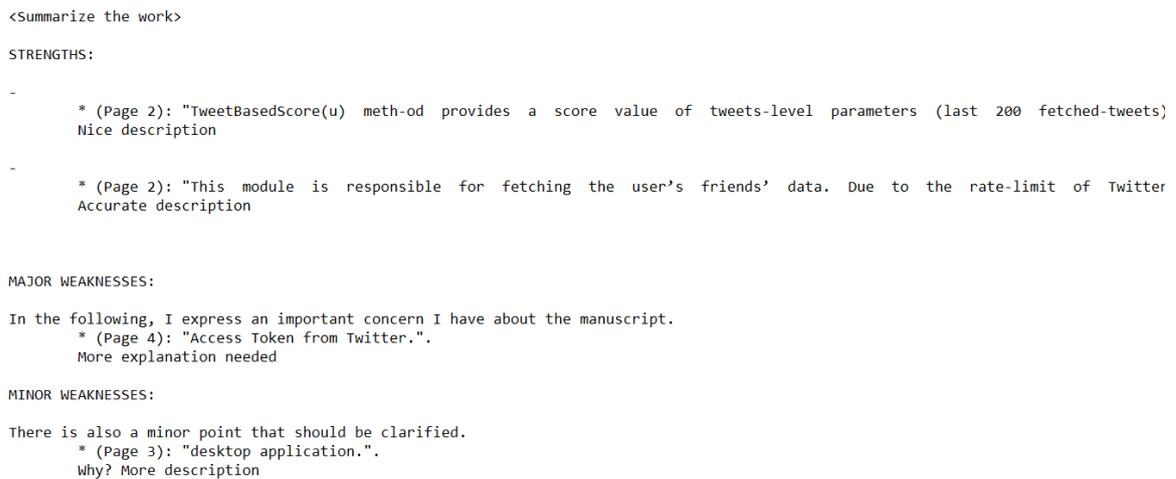


Figura 28: Borrador guardado

Una vez hemos terminado de realizar los cambios, procederemos a descargar el informe haciendo clic en 'Download'. Se nos descargará un archivo '.txt' que contiene un resumen de la evaluación de criterios que hemos realizado, tal y como muestra la imagen 29. Contiene la lista de criterios clasificados según su nivel de importancia: 'Essential', 'Desirable' y 'Extraordinary'. Además, se clasifican según la lista de criterios de la que provienen, como 'Engineering research (Design Science)'. Los que contienen el símbolo ✓, han sido aprobados; el símbolo ✗, han sido suspendidos y el de interrogación, ?, indica que no han sido evaluados. También contiene el informe que hemos editado, como se puede ver en la figura 30. Esta figura muestra las anotaciones que han sido categorizadas en: 'Strength', 'Major weakness' y 'Minor weakness'. Junto a ellas se muestra la página en la que se encuentra la anotación y el comentario realizado, en caso de haberlo.



**Figura 29:** Informe descargado (evaluación criterios)



**Figura 30:** Informe descargado (anotaciones realizadas)



## **C. Actas de reuniones**

## Acta de reunión 0

**Título de la reunión:** Toma de contacto con el proyecto

**Hora:** 10:00

**Fecha:** 5/02/2021

**Medio:** Presencial

**Duración:** 2h

### Orden del día

- Presentación del proyecto y sus objetivos
- Introducción a la herramienta pure::variants

### Resumen de la reunión

- Se ha realizado una presentación del proyecto por parte del tutor (Óscar) definiendo los objetivos y las tareas que lo van a componer. Más tarde Iñigo y Haritz han continuado con la reunión para hacer una introducción a la herramienta pure::variants.

### Estado del proyecto

- Acaba de dar comienzo el proyecto.

### Decisiones adoptadas

No se ha tomado ninguna decisión.

### Asignación de tareas a realizar

- Iñigo debe probar la herramienta pure::variants más a fondo y acostumbrarse a su uso.

**Próxima reunión:** No definida

### Asistentes:

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

# Acta de reunión 1

**Título de la reunión:** Dominio y presentación del proyecto

**Hora:** 10:00

**Fecha:** 19/02/2021

**Medio:** Presencial

**Duración:** 2h

## Orden del día

- Presentar dudas acerca de pure variants
- Presentación del dominio del proyecto
- Presentación del proyecto WACline

## Resumen de la reunión

- La reunión ha comenzado con una duda acerca de la utilidad de un componente de pure variants.
- Después ha proseguido con la presentación de Haritz acerca del dominio de las anotaciones web.
- Se ha presentado el diagrama acerca de las características del proyecto WACline y las características específicas de cada producto resultante de la línea de productos.
- Se ha descargado y probado el correcto funcionamiento del proyecto en el ordenador de Iñigo.

## Estado del proyecto

- Se ha realizado el estudio acerca del funcionamiento del programa pure variants y la programación de líneas de producto en el mismo.

## Decisiones adoptadas

No se ha tomado ninguna decisión.

## Asignación de tareas a realizar

Iñigo debe:

- investigar acerca de las extensiones de Google Chrome.
- investigar acerca del proyecto y realizar pruebas en el código y con la extensión.

**Próxima reunión:** No definida

## Asistentes:

Haritz Medina

Iñigo Bereciartua

## Acta de reunión 2

**Título de la reunión:** Presentación de Módulo de Búsqueda de Keywords

**Hora:** 11:00

**Fecha:** 16/03/2021

**Medio:** Presencial

**Duración:** 1'5h

### Orden del día

- Presentación del primer módulo a implementar dentro del proyecto WACline.

### Resumen de la reunión

- Se ha realizado una breve descripción de los 3 módulos a implementar en el proyecto WACline.
- Se ha profundizado en la primera característica a implementar en el proyecto WacLine (Módulo Búsqueda Keywords).
- Se ha revisado el código del proyecto actual para ver código reutilizable para aplicar a las características.

### Estado del proyecto

- Se ha estudiado el código del proyecto WACline.
- Se ha resuelto un error del proyecto.
- Se han presentado los requisitos del primer módulo a implementar.
- Se ha dado comienzo a la primera iteración.

### Decisiones adoptadas

### Asignación de tareas a realizar

- Iñigo debe Implementar la búsqueda y anotación de palabras clave al cargar un pdf con la extensión.

**Próxima reunión:** No definida

### Asistentes:

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

## Acta de reunión 3

**Título de la reunión:** Final primera iteración y establecer próximos objetivos

**Hora:** 12:15

**Fecha:** 30/03/2021

**Medio:** Presencial

**Duración:** 0'25h

### Orden del día

- Revisar el progreso actual del proyecto.
- Analizar la finalización del primer módulo.
- Establecer próximos pasos.

### Resumen de la reunión

- Se ha comenzado hablando acerca del avance del proyecto y de cómo ha finalizado el primer paso de implementación del primer módulo.
- Se ha planteado el siguiente paso/característica a realizar en el proyecto (Módulo de búsqueda de Información de Autores).
- Se han comentado información y fuentes para realizar la implementación del nuevo módulo.

### Estado del proyecto

- Se ha finalizado el módulo de implementación (con un problema a revisar).
- Se ha finalizado la primera iteración.
- Se ha presentado (más concretamente) el segundo módulo a implementar y ha comenzado la segunda iteración.

### Decisiones adoptadas

### Asignación de tareas a realizar

- Solucionar problema de subrayado de palabras en la primera característica (Haritz e Iñigo).

**Próxima reunión:** No definida

### Asistentes:

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

## Acta de reunión 4

**Título de la reunión:** Demo del producto actual y presentación del tercer módulo

**Hora:** 18:15

**Fecha:** 14/04/2021

**Medio:** Presencial

**Duración:** 0'5h

### Orden del día

- Realizar demo del producto actual.
- Presentación del siguiente módulo a diseñar e implementar.

### Resumen de la reunión

- Se ha comenzado con una presentación del producto actual, que tiene implementados los dos primeros módulos(1. Búsqueda de keywords y 2. información de autores).
- Se ha planteado el siguiente módulo a diseñar e implementar en el proyecto.
- Se han comentado posibles diseños de la solución.

### Estado del proyecto

- Se ha finalizado la implementación del segundo módulo y con ello la segunda iteración.
- Se ha presentado el tercer módulo a implementar y ha comenzado la tercera iteración.

### Decisiones adoptadas

### Asignación de tareas a realizar

- Diseñar una solución para el tercer módulo y comentarla a través de correo electrónico.

**Próxima reunión:** No definida

### Asistentes:

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

## **Acta de reunión 5**

**Título de la reunión:** Presentación de prototipos 3º módulo

**Hora:** 11:15

**Fecha:** 19/04/2021

**Medio:** Presencial

**Duración:** 1h

### **Orden del día**

- Presentación de los prototipos para el 3ª módulos a implementar.

### **Resumen de la reunión**

- Se ha comenzado con una presentación de los prototipos.
- Se ha realizado un esquema del diagrama de flujo del proceso de revisión de un artículo.

### **Estado del proyecto**

- Se han aprobado los prototipos.

### **Decisiones adoptadas**

### **Asignación de tareas a realizar**

- Realizar el diagrama de flujo completo.

**Próxima reunión:** No definida

### **Asistentes:**

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

## Acta de reunión 6

**Título de la reunión:** Presentación de diagrama de proceso de evaluación

**Hora:** 11:15

**Fecha:** 20/04/2021

**Medio:** Presencial

**Duración:** 1h

### Orden del día

- Presentación del diagrama del proceso de evaluación de artículos.

### Resumen de la reunión

- Se ha comenzado con una presentación del diagrama.
- Se han comentado modificaciones para mejorar el diagrama.

### Estado del proyecto

- El diagrama de proceso necesita una revisión antes de pasar a la implementación del tercer módulo.

### Decisiones adoptadas

### Asignación de tareas a realizar

- Realizar las modificaciones comentadas en la reunión en el diagrama y enviar nueva propuesta por correo.

**Próxima reunión:** No definida

### Asistentes:

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

## Acta de reunión 7

**Título de la reunión:** Demo de la tercera iteración y boceto del esquema de la memoria

**Hora:** 11:15

**Fecha:** 3/05/2021

**Medio:** Presencial

**Duración:** 1h

### Orden del día

- Demo del producto resultante de la tercera iteración.
- Presentación de boceto del esquema de la memoria.

### Resumen de la reunión

- Se ha comenzado con una demo del producto actual.
- Se han debatido los cambios acerca del enfoque que se le ha dado al menú lateral y canvas de la tercera característica.
- Se ha comentado el boceto de la memoria.

### Estado del proyecto

- El menú de la 3ª característica necesita otra vuelta para un mejor enfoque.
- Se ha finalizado la tercera iteración

### Decisiones adoptadas

- Pensar acerca del enfoque que se le va a acabar dando al producto.
- Centrarse en la redacción de la memoria.
- Necesidad de otra iteración para finalizar la implementación.

### Asignación de tareas a realizar

- Haritz va a pensar acerca del enfoque del producto.
- Refinar algunos aspectos del producto y centrarse en la redacción de la memoria.

**Próxima reunión:** No definida

### Asistentes:

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

## **Acta de reunión 8**

**Título de la reunión:** Demo del producto cuarta iteración

**Hora:** 11:30

**Fecha:** 1/06/2021

**Medio:** Presencial

**Duración:** 1h

### **Orden del día**

- Demo del producto actual.

### **Resumen de la reunión**

- Se ha comenzado con una demo del producto.
- Se han debatido los cambios de la interfaz del producto.

### **Estado del proyecto**

- El producto necesita unos cambios en la interfaz.

### **Decisiones adoptadas**

- Pensar acerca del enfoque que se le va a acabar dando al producto.

### **Asignación de tareas a realizar**

- Iñigo debe realizar los cambios de interfaz que se han comentado.

**Próxima reunión:** No definida

### **Asistentes:**

Óscar Díaz

Haritz Medina

Iñigo Bereciartua

## Acta de reunión 9

**Título de la reunión:** Demo del producto final

**Hora:** 13:30

**Fecha:** 7/06/2021

**Medio:** Presencial

**Duración:** 0'5h

### Orden del día

- Demo del producto final.

### Resumen de la reunión

- Se ha comenzado con una demo del producto
- Se han comentado mejoras y dudas para la memoria y detalles a tener en cuenta

### Estado del proyecto

- La cuarta iteración ha concluido y con ello la fase de implementación.
- El producto está finalizado.

### Decisiones adoptadas

- Realizar vídeo demo del producto.

### Asignación de tareas a realizar

Iñigo debe:

- finalizar la memoria
- realizar el vídeo demo
- realizar el póster

**Próxima reunión:** No definida

### Asistentes:

Óscar Díaz

Haritz Medina

Iñigo Bereciartua



# Bibliografía

- [1] Klaus Pohl, Günter Böckle, and Frank Linden. *Software Product Line Engineering: Foundations, Principles, and Techniques*. 01 2005. Ver páginas **viii**, **22**.
- [2] Christian Kästner, Norbert Siegmund, and Gunter Saake. Software product line engineering development process and variability modelling. Ver páginas **viii**, **23**.
- [3] Extensions - chrome developers. <https://developer.chrome.com/docs/extensions/>. Ver páginas **ix**, **65**.
- [4] Oscar Díaz, Jeremías P. Contell, and Haritz Medina. Performant peer review for design science manuscripts: A pilot study on dedicated highlighters. In Bengisu Tulu, Soussan Djamasbi, and Gondy Leroy, editors, *Extending the Boundaries of Design Science Theory and Practice*, pages 61–75, Cham, 2019. Springer International Publishing. Ver páginas **1**, **3**.
- [5] Jonathan P Tennant and Tony Ross-Hellauer. The limitations to our understanding of peer review. Ver página **3**.
- [6] Óscar Díaz. Líneas de prodcutio software 1. Ver página **21**.
- [7] pure-systems GmbH. Technical white paper variant management with pure::variants. <https://www.pure-systems.com/mediapool/pv-whitepaper-en-04.pdf>, 2006. Ver página **24**.