

ESCUELA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA INFORMÁTICA

SYSTEMATIC LITERATURE REVIEW

AUTOR: Ángel Rafael González Toro

31 de mayo de 2017

ESCUELA SUPERIOR DE INGENIERÍA

GRADO EN INGENIERÍA INFORMÁTICA

SYSTEMATIC LITERATURE REVIEW

AUTOR: ÁNGEL RAFAEL GONZÁLEZ TORO
DIRECTOR: IVAN RUÍZ RUBE

Cádiz, 31 de mayo de 2017

Agradecimientos

Introduzca aquí, si lo desea, los agradecimientos.

Resumen

Revision Sistemática de la Literatura consiste en la creación de una aplicación web para determinar el estado del arte sobre el cuál se sustenta una determinada área de investigación. La aplicación realizará una búsqueda de referencias bibliográficas en una serie de motores de búsquedas determinados y será capaz de filtrar la información obtenida acorde con los parámetros que le indique el usuario. Para este proceso, se desarrollará una web que permitirá automatizar cada una de las etapas del proceso para realizar revisiones sistemáticas de la literatura. Tras la ejecución de este proceso, se podrán exportar los metadatos de todos los estudios encontrados a través de gráficos o ser exportados a un gestor de referencias bibliográficas.

Palabras clave: revisión sistemática, literatura, referencia, systematic literature review, metadatos, investigación, mendeley.

Índice general

I	Prolegómeno	1
1.	Introducción	3
1.1.	Motivación	3
1.2.	Alcance	4
1.2.1.	Objetivos	5
1.3.	Glosario de Términos	5
1.3.1.	Acrónimos	6
1.3.2.	Términos	7
1.4.	Organización del documento	7
2.	Planificación	9
2.1.	Metodología de desarrollo	9
2.2.	Planificación del proyecto	10
2.3.	Organización	11
2.3.1.	Roles	11
2.3.2.	Recursos	17
2.4.	Costes	18
2.5.	Riesgos	19
2.6.	Aseguramiento de calidad	19
II	Desarrollo	21
3.	Requisitos del Sistema	23
3.1.	Necesidades de Negocio	23
3.2.	Objetivos del Sistema	23
3.3.	Catálogo de Requisitos	24
3.3.1.	Requisitos funcionales	24
3.3.2.	Requisitos no funcionales	25
3.3.3.	Reglas de negocio	25
3.3.4.	Requisitos de información	25
3.4.	Alternativas de Solución	26
3.5.	Solución Propuesta	28
4.	Análisis del Sistema	29
4.1.	Modelo Conceptual	29
4.2.	Modelo de Casos de Uso	30

4.2.1. Actores	30
4.2.2. Diagramas y especificación de casos de uso	30
4.3. Modelo de Interfaz de Usuario	48
5. Diseño del Sistema	53
5.1. Arquitectura del Sistema	53
5.1.1. Arquitectura Física	53
5.1.2. Arquitectura Lógica	55
5.2. Parametrización del software base	57
5.3. Diseño Físico de Datos	58
5.4. Diseño detallado de Componentes	62
5.5. Diseño detallado de la Interfaz de Usuario	64
6. Construcción del Sistema	69
6.1. Entorno de Construcción	69
6.2. Código Fuente	70
6.3. Scripts de Base de datos	73
7. Pruebas del Sistema	75
7.1. Estrategia	75
7.2. Entorno de Pruebas	75
7.3. Roles	75
7.4. Niveles de Pruebas	75
7.4.1. Pruebas Unitarias	75
7.4.2. Pruebas de Integración	76
7.4.3. Pruebas de Sistema	76
7.4.4. Pruebas de Aceptación	76
III Epílogo	77
8. Manual de implantación y explotación	81
8.1. Introducción	81
8.2. Requisitos previos	81
8.3. Inventario de componentes	81
8.4. Procedimientos de instalación	81
8.5. Pruebas de implantación	81
8.6. Procedimientos de operación y nivel de servicio	81
9. Manual de usuario	83
9.1. Introducción	83
9.2. Instalación	83
9.3. Uso del sistema	83
10. Manual del desarrollador	85
10.1. Introducción	85
10.2. Preparación del entorno de trabajo	85
10.3. Consideraciones generales sobre el desarrollo	85
10.4. Instrucciones para construcción y despliegue	85

11. Conclusiones	87
11.1. Objetivos alcanzados	87
11.2. Lecciones aprendidas	87
11.3. Trabajo futuro	87
Bibliografía	89
Información sobre Licencia	91

Índice de figuras

2.1. Diagrama del ciclo iterativo scrum	9
2.2. Diagrama de Gantt con la propuesta inicial y la adquisición de conocimientos. . .	12
2.3. Diagrama de Gantt con el Desarrollo de la Aplicación Web.	13
2.4. Diagrama de Gantt con el Desarrollo de la Aplicación Web (II).	14
2.5. Diagrama de Gantt con el Desarrollo de la Aplicación Mendeley-REST en Java. .	15
2.6. Diagrama de Gantt con la Integración de Aplicaciones, Despliegue y Redacción de la memoria.	16
4.1. Diagrama conceptual de clases UML.	29
4.2. Diagrama casos de usos del sistema.	30
4.3. Diagrama casos de usos de la gestión de usuarios.	31
4.4. Diagrama casos de usos de la gestión de revisiones sistemáticas.	31
4.5. Diagrama casos de usos de la gestión de preguntas de investigación.	32
4.6. Diagrama casos de usos de la gestión de criterios de clasificación.	32
4.7. Diagrama casos de usos de la gestión de atributos específicos.	32
4.8. Diagrama casos de usos de la gestión de búsquedas de referencias bibliográficas. .	33
4.9. Diagrama casos de usos de la sincronización con Mendeley.	33
4.10. Pantalla login.	48
4.11. Pantalla Revisiones Sistemáticas Usuario.	48
4.12. Pantalla creación revisión sistemática	49
4.13. Pantalla creación preguntas de investigación	49
4.14. Pantalla creación búsquedas de referencias bibliográficas.	50
4.15. Pantalla de creación de búsquedas en progreso.	50
4.16. Pantalla Referencias bibliográficas.	51
4.17. Pantalla exportación referencias bibliográficas y gráficos (I).	51
4.18. Pantalla exportación referencias bibliográficas y gráficos (II).	52
5.1. Arquitectura física del sistema (I).	54
5.2. Arquitectura del sistema (II).	54
5.3. Arquitectura Grails.	56
5.4. Registrar aplicación en Mendeley.	58
5.5. Registrar aplicación en Mendeley.	58
5.6. Estructura física aplicación SLR.	59
5.7. Interfaz Login Sistema	64
5.8. Interfaz Menu principal.	64
5.9. Interfaz Lista SLR.	65
5.10. Interfaz Lista Criterios.	65
5.11. Interfaz Lista Atributos específicos	65

5.12. Interfaz Crear búsqueda	66
5.13. Interfaz progreso de búsquedas	66
5.14. Interfaz Lista búsquedas	66
5.15. Interfaz notificación búsqueda	67
5.16. Interfaz Lista Referencias	67
5.17. Interfaz Referencia	67
5.18. Interfaz gráficos SLR (I)	68
5.19. Interfaz gráficos SLR (II)	68
6.1. Estructura Proyecto Grails	70
6.2. Dominio clase Search	71
6.3. Controlador clase Search	72
6.4. Vista clase Search	72

Índice de tablas

1.1. OBJ-01: Creación de una aplicación web para realizar Revisiones Sistemáticas de la Literatura.	5
1.2. OBJ-02: Realizar Búsquedas de Referencias Bibliográficas.	5
1.3. OBJ-03: Clasificar las referencias bibliográficas.	6
1.4. OBJ-04: Exportar información de los SLR y conclusiones finales.	6
2.1. Tiempo estimado frente a tiempo real de cada Sprint	11
2.2. Costes del desarrollo	19
2.3. Lista de riesgos del proyecto	20
3.1. OBJ-01: Creación de una aplicación web para realizar Revisiones Sistemáticas de la Literatura.	24
3.2. OBJ-02: Realizar Búsquedas de Referencias Bibliográficas.	24
3.3. OBJ-03: Clasificar las referencias bibliográficas.	24
3.4. OBJ-04: Exportar información de los SLR y conclusiones finales.	24
3.5. IRQ-01: Requisitos de información de los usuarios.	26
3.6. IRQ-02: Requisitos de información de las revisiones sistemáticas de la literatura. .	26
3.7. IRQ-03: Requisitos de información de las búsquedas de referencias bibliográficas de un SLR.	26
3.8. IRQ-04: Requisitos de información de las referencias bibliográficas de un SLR. . .	27
3.9. IRQ-05: Requisitos de información de los criterios de clasificación de un SLR. . .	27
3.10. IRQ-06: Requisitos de información de las preguntas de investigación de un SLR. .	27
3.11. IRQ-07: Requisitos de información de los metadatos específicos de una referencia en un SLR.	28
3.12. IRQ-08: Requisitos de información de los datos a exportar de un SLR.	28
4.1. CU-01 Registrar en sistema	34
4.2. CU-02 Realizar login	34
4.3. CU-03 Cambiar rol	35
4.4. CU-04 Listar usuarios	35
4.5. CU-05 Crear SLR	36
4.6. CU-06 Modificar SLR	37
4.7. CU-07 Borrar SLR	38
4.8. CU-08 Exportar SLR	38
4.9. CU-09 Crear Pregunta de investigación	39
4.10. CU-10 Modificar pregunta de investigación	40
4.11. CU-11 Borrar pregunta investigación	41
4.12. CU-12 Crear criterio de clasificación	41

4.13. CU-13 Modificar criterio de clasificación	42
4.14. CU-14 Borrar criterio de clasificación	42
4.15. CU-15 Crear atributo específico	43
4.16. CU-16 Modificar atributo de especificación	43
4.17. CU-17 Eliminar atributo de especificación	43
4.18. CU-18 Crear búsquedas de referencias	44
4.19. CU-19 Listar referencias	45
4.20. CU-20 Modificar referencias	45
4.21. CU-21 Eliminar referencias	45
4.22. CU-22 Listar Búsquedas	46
4.23. CU-23 Sincronizar Perfil	46
4.24. CU-24 Sincronizar SLR	47
4.25. CU-24 Sincronizar Referencias	47
5.1. Arquitectura del Servidor	55
5.2. Tabla user	58
5.3. Tabla user_mendeley	60
5.4. Tabla user_profile	60
5.5. Tabla criterion	61
5.6. Tabla specific attribute	61
5.7. Tabla slr	61
5.8. Tabla search	62
5.9. Tabla search_term_param	62
5.10. Tabla search_operator	62
5.11. Tabla search_component	63
5.12. Tabla engine	63
5.13. Tabla reference	63

Parte I

Prolegómeno

Capítulo 1

Introducción

A continuación, se describe la motivación del presente proyecto y su alcance. También se incluye un glosario de términos y la organización del resto de la presente documentación.

1.1. Motivación

Es algo muy común en el ámbito académico, que tanto estudiantes pre-doctorales como para cualquier persona ya doctorada se dedique en algún momento de su carrera a realizar una investigación sobre una determinada área. Incluso existen empresas donde existen departamentos de I+D que realizan continuamente numerosas investigaciones para sus estudios. Para la realización de la misma, estas personas necesitarán realizar una búsqueda exhaustiva de información o recursos y comprobar que existan, o no, evidencias de dicho conocimiento. Es decir, deben ser capaces de resumir la evidencia existente acerca de un tema o identificar, si se da el caso, de que haya un vacío de conocimiento sobre esa investigación.

Este estudio del arte sobre algún tema específico puede ayudarnos entre otras cosas para:

- Resumir la evidencia existente acerca de una determinada área de investigación o tema de estudio.
- Identificar los vacíos en una investigación para sugerir más áreas o recursos de investigación.
- Creación de nuevas áreas o actividades de investigación.

Una **revisión sistemática de la literatura** (*Systematic Literature Review*) es un medio para evaluar e interpretar la investigación disponible relativa a una determinada área de interés. Por tanto, una persona que desea realizar un estudio sobre un determinado tema, deberá buscar referencias bibliográficas por medio de artículos, aportes vía internet u otro tipos de documentos y saber interpretar dichos resultados para llegar a una conclusión sobre este estudio.

La mayor parte de la investigación comienza con una revisión sistemática de la literatura, por contra, éstas carecen de valor científico si no se trata de una revisión exhaustiva y justa. Éste es el motivo principal para el desarrollo de revisiones sistemáticas. Una revisión sistemática sintetiza de una manera eficaz e idónea el estudio del arte de un determinado tema. Sin embargo, hay que seguir una estrategia de búsqueda que deberá permitir la integridad de la búsqueda que se evaluará. Concretamente, los investigadores deberán hacer un esfuerzo por identificar e investigar

cada uno de los recursos que encuentre y presentar un informe que lo sustente.

Barbara Kitchenham propuso una guía [Kitchenham, 2007] o conjunto de normas para promover estos estudios de la literatura en Ingeniería del Software. Está dirigida principalmente a investigadores de ingeniería del software, incluyendo a los doctorados. Este documento no cubre ningún proceso estadístico para resumir los resultados cuantitativos en los diferentes estudios (meta-análisis), ni explicar las diferentes implicaciones que podrían obtenerse acorde a los resultados encontrados, pero sí las directrices correctas para facilitar el estudio y que están siendo tan empleadas en otras disciplinas como la medicina y las ciencias sociales.

Para estudiar la literatura de un tópico a evaluar, es necesario de una metodología confiable, rigurosa y extendida en la comunidad investigadora. Y este proyecto de fin de carrera se encargará de ayudar al investigador a encontrar las referencias bibliográficas que le ayude a facilitar el estudio y la exportación de los informes para determinar las conclusiones finales del estudio siguiendo la línea indicada por Kitchenham.

1.2. Alcance

Un **SLR** (*Systematic Literature Review*) no es más que una metodología rigurosa para identificar, analizar e interpretar de una forma no sesgada todas las evidencias referentes a una pregunta de investigación.

Este proyecto se plantea para **facilitar al investigador en el proceso de búsqueda de referencias bibliográficas en diferentes motores de búsquedas especializados** que proporciona internet **y ayudar al investigador a exportar los resultados obtenidos** con el fin de poder elaborar un informe con las conclusiones del estudio.

Para la creación de este PFC, se ha realizado una aplicación web donde el usuario puede realizar este proceso de búsqueda a través de varios motores de búsquedas específicos con el ámbito universitario e investigador de una sola vez, evitando que el usuario tenga que repetir el mismo proceso en cada uno de los sitios. Estos motores de búsquedas implican una serie de reglas o unos formatos de búsqueda concretos, y la aplicación se encargará de esta tarea tediosa para el investigador.

La literatura encontrada en cada uno de estos medios universitarios será almacenada en un gestor de referencias que los investigadores pueden manejar denominado **Mendeley**. La aplicación web conectará con este gestor y almacenará todas las referencias que han sido encontradas y creará un conjunto de carpetas y documentos donde el investigador podrá saber en todo momento en qué motor de búsqueda ha sido encontrada la referencia, la revisión sistemática a la que pertenece, así como todo el detalle disponible de cada una de las referencias.

Una vez que hemos obtenido toda la literatura a estudiar en cada uno de estos medios, el usuario tendrá la difícil tarea de indicar si las referencias bibliográficas encontradas siguen los criterios necesarios para incluirlos en el proceso de estudio. Por medio de esta aplicación, el usuario podrá diferenciar cuáles cumplen los requisitos para ser incluidos a través de unos criterios previamente definidos por el usuario, así como incluir otra información que Mendeley por defecto no ha podido

encontrar.

Por último, y una vez realizado el estudio del arte, podemos obtener un informe con las valoraciones y criterios propuestos por el investigador de un simple vistazo por medio de la exportación de documentos en diferentes formatos de salida o gráficos que ilustre el estudio realizado, así como las conclusiones finales obtenidas.

A continuación se enumeran y describen los principales objetivos que se esperan alcanzar cuando el sistema a desarrollar esté en producción.

1.2.1. Objetivos

El objetivo principal de este proyecto, será ayudar al usuario a seguir la mayor parte de las directrices de Barbara Kitchenham, ya que el estudio del arte de una área de investigación pre-determinada no puede ser automatizado. Sin embargo, podemos ayudar al usuario a quitarle tiempo de búsqueda de recursos a través de un buscador de referencias en motores de búsquedas, que en posteriores capítulos describiremos, y ayudar al usuario a determinar las conclusiones finales por medio de la exportación de la información a través de gráficos o ficheros en diferentes formatos.

Todo este proceso será desarrollado a través de una aplicación web donde el usuario podrá crear revisiones sistemáticas de la literatura, realizar búsquedas, clasificar los documentos y poder exportar la información a través de varios medios. Podemos ver más detalles de estos objetivos en tablas 3.1, 3.2, 3.3 y 3.4.

OBJ-01	Creación de una aplicación web para realizar Revisiones Sistemáticas de la Literatura.
Descripción	Se debe diseñar una aplicación web donde el usuario podrá crear revisiones sistemáticas de la literatura.

Tabla 1.1: OBJ-01: Creación de una aplicación web para realizar Revisiones Sistemáticas de la Literatura.

OBJ-02	Realizar Búsquedas de Referencias Bibliográficas.
Descripción	La aplicación web deberá poder realizar varias búsquedas de referencias bibliográficas con la información predefinida por el usuario y almacenando dichos recursos en un sistema de gestión de referencias (Mendeley) para su futuro estudio.

Tabla 1.2: OBJ-02: Realizar Búsquedas de Referencias Bibliográficas.

1.3. Glosario de Términos

Esta sección contiene una lista de los acrónimos específicos y principales términos del dominio del sistema.

OBJ-03	Clasificar las referencias bibliográficas.
Descripción	La aplicación web deberá poder ayudar al usuario a clasificar las referencias bibliográficas encontradas en las búsquedas de cada una de las revisiones sistemáticas de la literatura.

Tabla 1.3: OBJ-03: Clasificar las referencias bibliográficas.

OBJ-04	Exportar información de los SLR y conclusiones finales.
Descripción	La aplicación web deberá poder ayudar al usuario a realizar un informe con las conclusiones finales del SLR a través de gráficos que reflejen la situación del mismo o diferentes ficheros de texto.

Tabla 1.4: OBJ-04: Exportar información de los SLR y conclusiones finales.

1.3.1. Acrónimos

API Application Programming Interface

DRY Don't Repeat yourself

GORM Grails Object Relational Mapping

GSP Groovy Server Pages

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

JAR Java ARchive

LOU Ley Orgánica de Universidades

MVC Modelo Vista Controlador

OO Orientado a Objetos

PFC Proyecto Fin de Carrera

REST REpresentational State Transfer

RSL Revisiones Sistemáticas de la Literatura

SGDB Sistema Gestion de Bases de Datos

SLR Systematic Literature Review

UCA Universidad de Cádiz

UML Lenguaje Unificado de Modelado

1.3.2. Términos

Grails Framework para aplicaciones web libre desarrollado sobre el lenguaje de programación Groovy.

Groovy Lenguaje de programación orientado a objetos implementado sobre la plataforma Java.

Mendeley Aplicación web y de escritorio, propietaria y gratuita. Permite gestionar y compartir referencias bibliográficas y documentos de investigación, encontrar referencias y documentos y colaborar en línea.

Metadato Dato que describe otro dato. Por lo general, un grupo de metadatos se refiere a un grupo de datos que describen el contenido de una referencia bibliográfica.

1.4. Organización del documento

A continuación se describe de manera resumida el contenido de los capítulos en los que se encuentra dividido este PFC:

- En el [Capítulo 1](#) se ha realizado un recorrido introductorio sobre el contexto donde nos moveremos así como la motivación al respecto, haciendo hincapié en los objetivos que se pretende satisfacer.
- En el [Capítulo 2](#) se incluye la planificación del proyecto, plazos, entregables, recursos utilizados, así como la metodología de ingeniería de software empleada.
- En el [Capítulo 3](#) se detalla la situación actual de la organización y las necesidades de la misma, que originan el desarrollo de un sistema informático.
- En el [Capítulo 4](#) cubre todo el análisis del sistema de información a desarrollar, haciendo uso del lenguaje de modelado UML.
- En el [Capítulo 5](#) ...
- En el [Capítulo 6](#) ...
- En el [Capítulo 7](#) ...
- En el [Capítulo 8](#) ...
- En el [Capítulo 9](#) ...
- En el [Capítulo 10](#) ...
- En el [Capítulo 11](#) ...

Respecto al software entregado en soporte informático, se distribuye en los siguientes directorios:

codigoFuente Código fuente de todos los archivos creados para desarrollar el sistema.

instalación Instrucciones necesarias para la instalación del sistema.

memoria Archivos tex y pdf de la memoria del proyecto.

presentación Archivos tex y pdf de la presentación del proyecto.

recursos Archivos de diagramas, imágenes y otros archivos de interés.

Capítulo 2

Planificación

En esta sección se describen todos los aspectos relativos a la gestión del proyecto: metodología, organización, costes, planificación, riesgos y aseguramiento de la calidad.

2.1. Metodología de desarrollo

En este proyecto se utiliza un modelo de ciclo de vida basado en un desarrollo iterativo e incremental, empleando varias de las directrices de la metodología ágil Scrum (Ver 2.1), ya que el proyecto puede tener un cambio constante de requisitos y además se pretende mostrar un incremento ejecutable cada cierto tiempo, por lo que el trabajo se irá dividiendo en hitos o **sprints**.

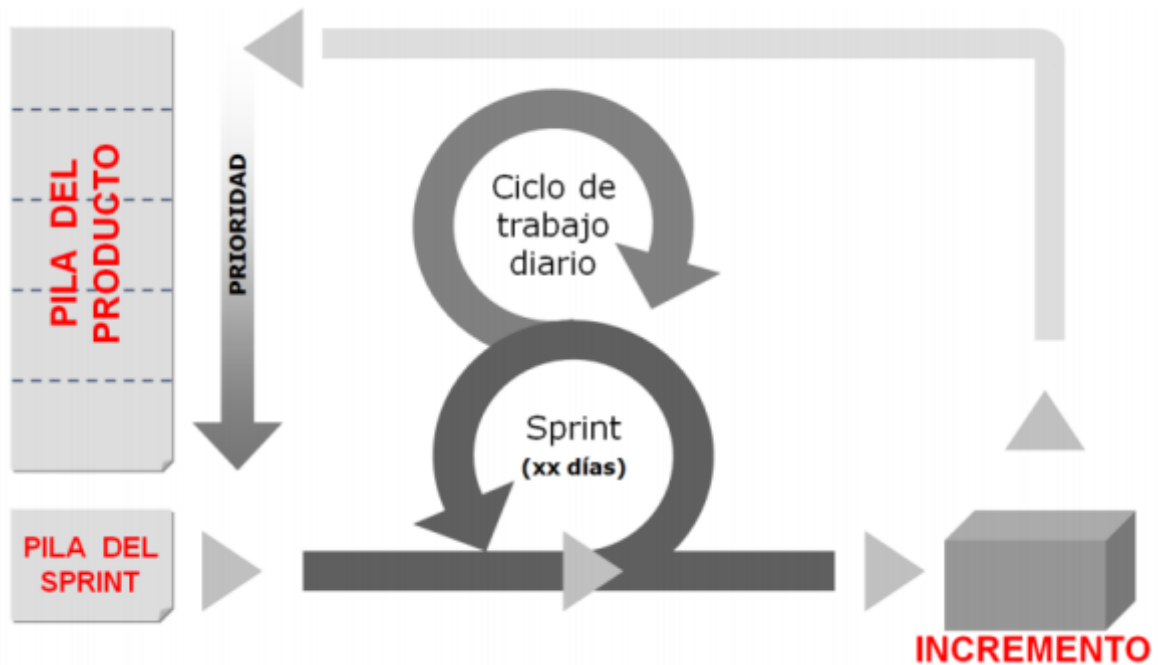


Figura 2.1: Diagrama del ciclo iterativo scrum

El marco técnico de scrum, [Bok, 2016] está formado por un conjunto de prácticas y reglas que dan respuesta a los siguientes principios de desarrollo ágil:

- Gestión evolutiva del producto, en lugar de la tradicional o predictiva.
- Calidad del resultado basado en el **conocimiento tácito de las personas**, antes que en el explícito de los procesos y la tecnología empleada.
- Estrategia de **desarrollo incremental** a través de iteraciones (sprints).

Con la visión general del proyecto, y a partir de ella se especifica y da detalle a las funcionalidades que se desean obtener en primer lugar. Cada ciclo de desarrollo o iteración (**sprint**) finaliza con la entrega de una parte operativa del producto (**incremento**). La duración de cada sprint ha sido entre una semana y dos meses.

Para diseñar el modelo de datos y la documentación se empleará UML (Lenguaje Unificado de Modelado) y un paradigma de programación OO (Orientado a Objetos). Además se buscará usar una arquitectura MVC (Modelo Vista Controlador) y tener cada parte lo más independiente posible del resto. Esta elección podrá permitir además una mejor integración entre todos los componentes y una mayor facilidad de mantenimiento y ampliación en el futuro.

2.2. Planificación del proyecto

En primer lugar, se realiza una planificación inicial y una investigación sobre las directrices que marca Barbara Kitchenham para crear una revisión sistemática de la literatura sobre un área determinada. Dentro de esta planificación inicial se comprueba que es necesario realizar un sistema de gestión para las distintas revisiones sistemáticas de un usuario, así como un sistema de búsquedas de referencias bibliográficas que pueda sincronizar toda la información con el gestor de referencias Mendeley, y, finalmente, su posterior clasificación según los criterios indicados por el usuario y la exportación de conclusiones finales.

Una vez realizada la planificación inicial y analizado todos los objetivos principales del proyecto se decide realizar el desarrollo del proyecto en dos subproyectos. El primero de ellos, será la aplicación web, donde el usuario podrá acceder a todas las opciones para las creaciones de las revisiones sistemáticas de la literatura. Por otro lado, el segundo de ellos consistirá en la creación de una librería JAR donde incluirá todo el proceso de búsquedas de referencias bibliográficas en segundo plano y de manera paralela a cualquier otra búsqueda que se realice. Esta librería, deberá ser incluida en el primer subproyecto, por lo que necesitará de un periodo de tiempo para integrar ambas aplicaciones.

Con toda esta información, se opta por dividir el proceso en ocho hitos tal y como podemos ver en la tabla 2.1. El tiempo se indica en días. Debido a que la dedicación de la elaboración de este proyecto ha sido a media jornada, se añade una columna con el porcentaje de dedicación en cada uno de los sprints. Cada uno de estos hitos, tendrá asociados varias subtarefas o subobjetivos que deberán realizarse para dar por bueno el hito principal. El hito 0 corresponderá a la adquisición de conocimientos mientras que el hito 7 es asignado a la redacción de la memoria.

Para mostrar el desarrollo detallado de toda la aplicación y desarrollo del software se muestran en las figuras 2.2, 2.3, 2.4, 2.5 y 2.6 unos diagramas de Gantt ordenados cronológicamente haciendo referencia a cada uno de los hitos que hemos explicado anteriormente. Los diagramas han sido realizados con la herramienta GanttProject [Project, 2015]. Cabe indicar, que el hito de mayor

Sprint	Descripción	Estimado	Real	% Dedicación
Hito 0	Propuesta inicial del proyecto	7	5	100 %
Hito 1	Investigación y adquisición de conocimientos	50	63	50 %
Hito 2	Desarrollo Aplicación Web	110	160	50 %
Hito 3	Desarrollo Aplicación Mendeley-REST en Java	60	100	50 %
Hito 4	Integración de aplicaciones	30	26	50 %
Hito 5	Pruebas del sistema y rendimiento	10	5	50 %
Hito 6	Despliegue de la aplicación	20	25	50 %
Hito 7	Redacción de memoria	180	280	50 %
Media		180	280	50 %
Desviación		180	280	50 %

Tabla 2.1: Tiempo estimado frente a tiempo real de cada Sprint

duración es el último, que corresponde a la redacción de la documentación de este proyecto, ya que se ha ido redactando a medida que se ha ido completando cada una de las fases del proyecto.

2.3. Organización

En este apartado recogemos las personas o roles que se encuentran involucrados en el proyecto así como una relación de los recursos inventariables utilizados en el proyecto: equipamiento informático, herramientas empleadas, etc.

2.3.1. Roles

En los proyectos que siguen una metodología ágil por SCRUM podemos encontrarnos con tres tipos de roles [Bok, 2016]:

- **Dueño del producto.** Es el representante del cliente, el cuál conoce el entorno de negocio del cliente, las necesidades y el objetivo que se persigue con el sistema que se construye. Tiene la visión del producto, así como las necesidades concretas del proyecto, para poder priorizar eficientemente el trabajo. El tutor de este proyecto es el más claro ejemplo de representante, así como la UCA ejerce de cliente.
- **Equipo de desarrollo.** Grupo o grupos de trabajo que desarrollan el producto. Todos conocen y comprenden la visión del propietario del producto. Además, aportan y colaboran con el propietario en el desarrollo de la pila del producto. En este proyecto el equipo está compuesto únicamente por el autor.
- **Scrum master.** Es el responsable del cumplimiento de las reglas de un marco de scrum técnico, asegurando que se entienden en la organización, y se trabaja conforme a ellas.

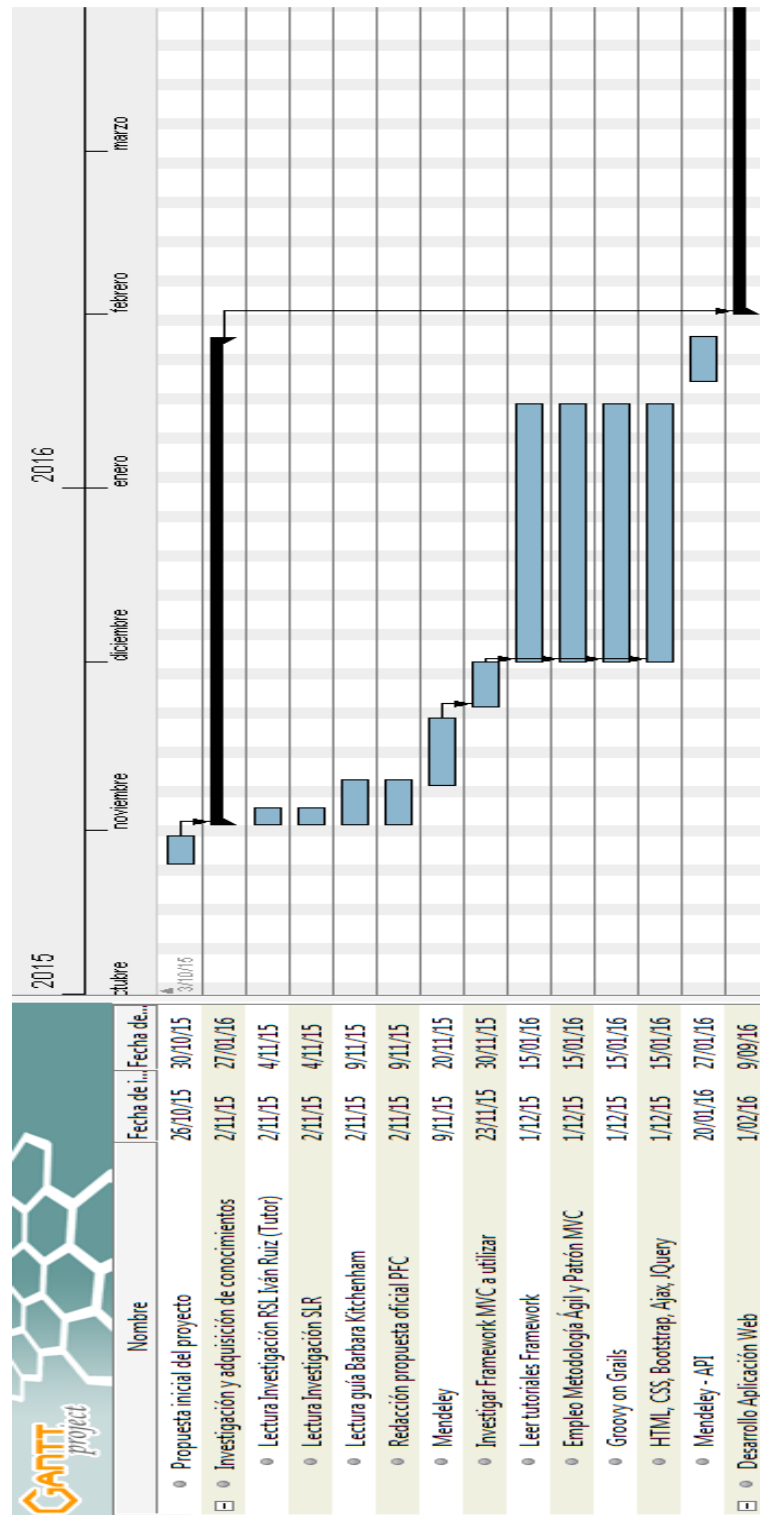


Figura 2.2: Diagrama de Gantt con la propuesta inicial y la adquisición de conocimientos.

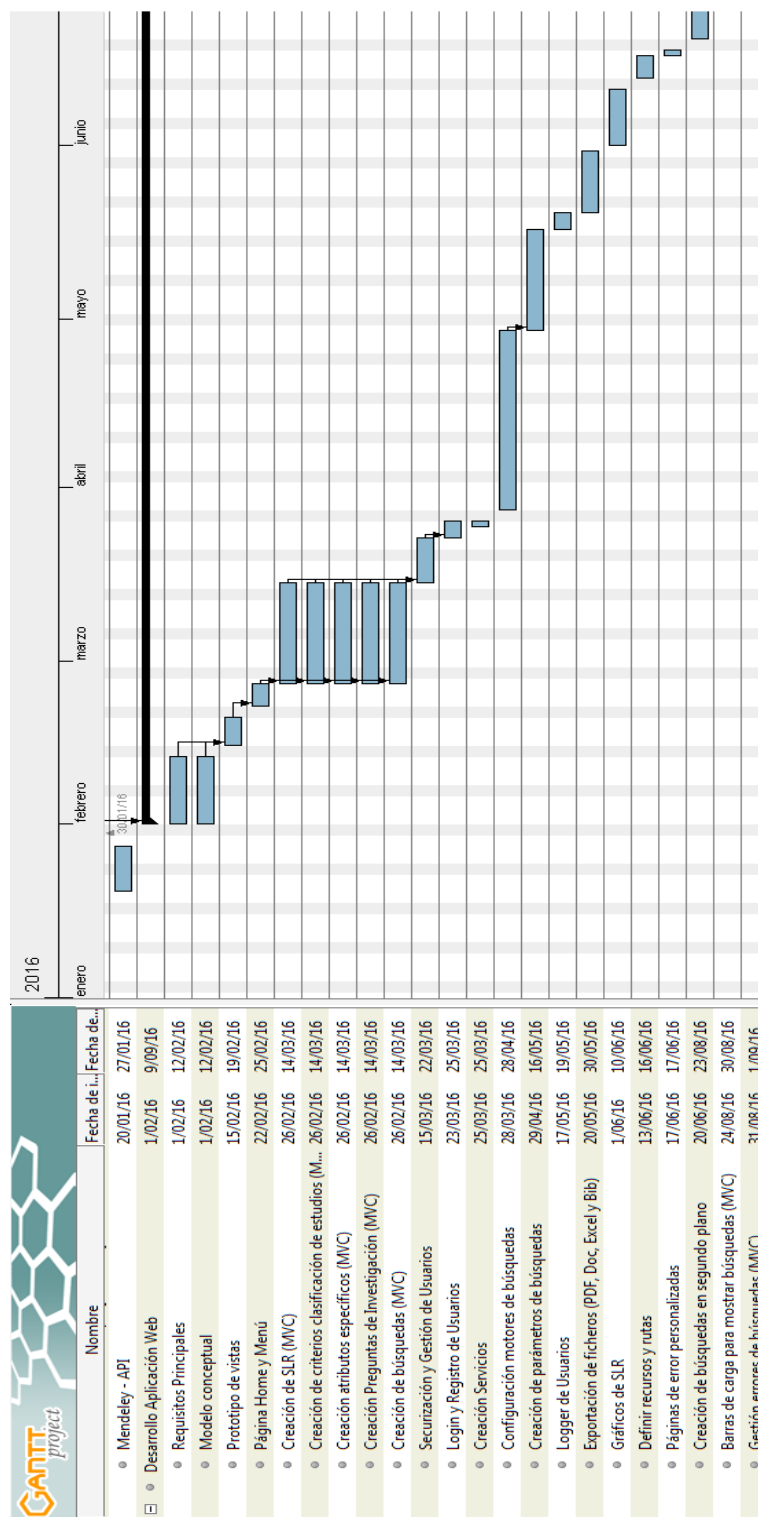


Figura 2.3: Diagrama de Gantt con el Desarrollo de la Aplicación Web.

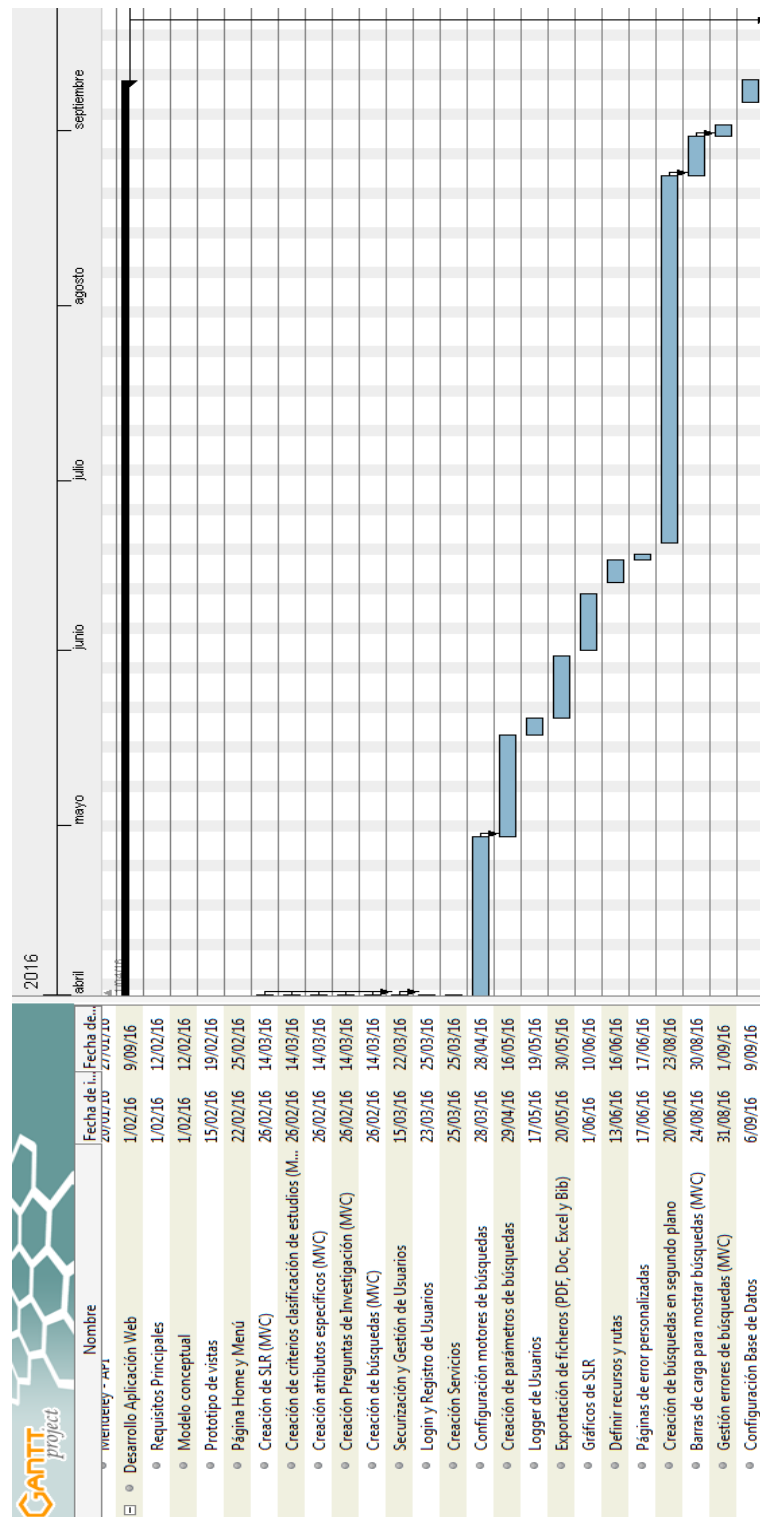


Figura 2.4: Diagrama de Gantt con el Desarrollo de la Aplicación Web (II).

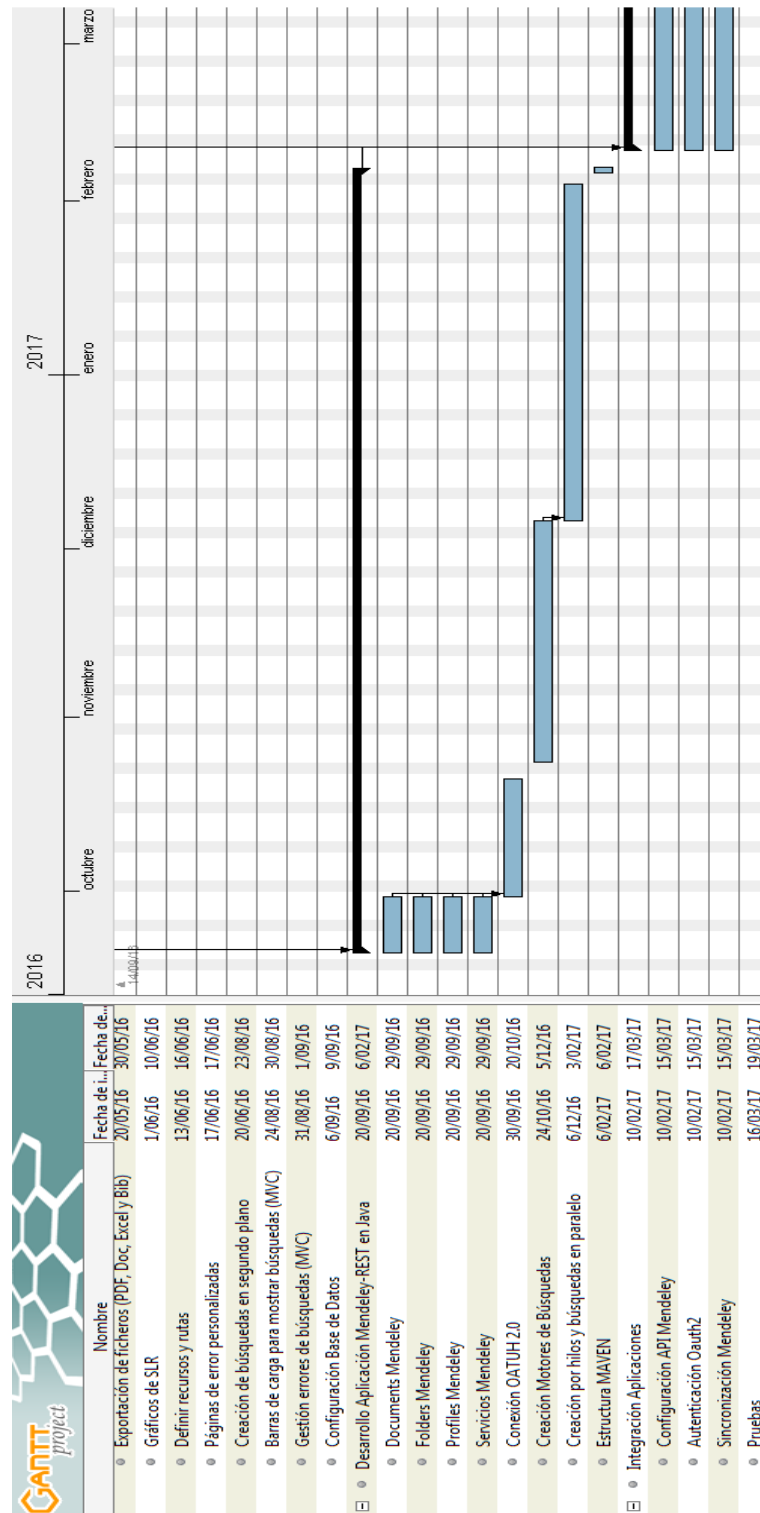


Figura 2.5: Diagrama de Gantt con el Desarrollo de la Aplicación Mendeley-REST en Java.

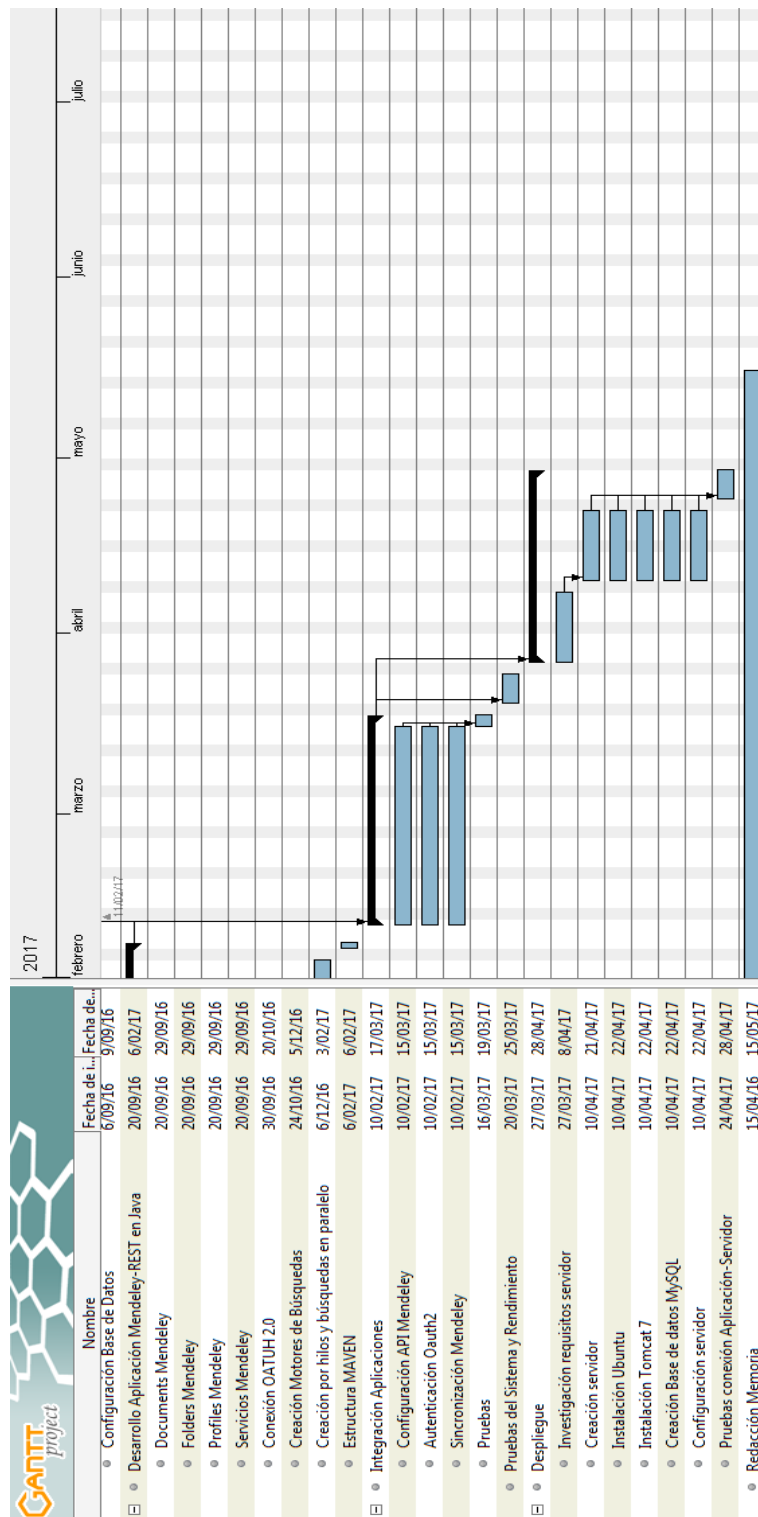


Figura 2.6: Diagrama de Gantt con la Integración de Aplicaciones, Despliegue y Redacción de la memoria.

Además, proporciona la asesoría y formación necesaria al propietario del producto y al equipo. En el proyecto estará representado por el autor, por lo que éste representará tanto al equipo de desarrollo como al scrum master.

2.3.2. Recursos

En este apartado se van a listar todos los recursos inventariables de hardware y software, así como las herramientas utilizadas y los lenguajes de programación.

En primer lugar se listan los recursos de hardware.

- Equipo donde se ha realizado el proyecto: Toshiba Satellite L750/L755, Intel ®Core™i5-2410M CPU @ 2.30GHz x 2, 4GB RAM
- Servidor donde se ha desplegado el proyecto: 1 vCore CPU, 2048MB RAM, 40 GB SSD Storage, 2000 GB Bandwidth

A continuación se listan los recursos de software

- Equipo donde se ha realizado el proyecto:
 - SO: Windows 7 64 bits
 - API Externa: Google Charts y Mendeley
- Servidor donde se ha desplegado el proyecto:
 - SO: Ubuntu 16.04 7 64 bits
 - Contenedor Web: Tomcat 7 64 bits
 - Base de Datos: MySql

A continuación se listan las herramientas empleadas.

- IDE: Grails Tool Suite
- Control de versiones: Subversion
- Forja: Assembla
- SGBD: Mysql
- Diseño de diagramas: DIA
- Diseño de Mockups: Balsamiq Mockups
- Navegadores empleados: Firefox y Google Chrome
- Memoria y presentación: L^AT_EX
- Procesador de texto: Microsoft Word 2010

- Gestión Hojas de cálculo: Microsoft Excel 2010
- Vision de documentos: Adobe Acrobat Reader

Para finalizar, se listan los lenguajes de programación utilizados:

- Groovy
- Java
- JavaScript
- Ajax
- JQuery
- CSS3
- Bootstrap
- HTML

2.4. Costes

Para poder realizar una estimación de los costes del proyecto debemos tener en cuenta tanto los recursos humanos como los recursos en material empleados. Los costes indirectos como electricidad o conexión a Internet son los que denominaremos “Costes indirectos”. El porcentaje de su valor podemos optar por un 10 % del gasto del personal tal como se indica en el artículo 83 de la LOU y que la Universidad de Cádiz emplea para la realización de trabajos externos. Podemos ver dicha información en: <http://www.uca.es/vrteit/portal.do?IDM=55&NM=2>.

De los recursos hardware debemos tener en cuenta tanto el equipo empleado para el desarrollo como el servidor donde se desplegará la aplicación web. Normalmente, los ordenadores suelen tener un periodo de amortización entre los 2 y 4 años. Al igual que antes, tomaremos la media de este periodo, 3 años. Si el equipo costó 750 euros, se amortizan unos 250 euros al año y como el tiempo de desarrollo del proyecto es de entre 2 y 3 años a tiempo parcial, tomaremos 2,5 años como periodo medio y el coste final será de 625 euros. El coste del servidor son de unos 10 euros al mes, sólomente ha estado desplegado en el servidor alrededor de 3 meses, por lo que su gasto alcanza los 30 euros.

Con respecto a los recursos del software, no tendrá ningún gasto, puesto que el sistema operativo Windows 7 viene incluido en los gastos del equipo con el que se ha desarrollado el proyecto, así como los paquetes informáticos (hojas de cálculo y editor de texto). Además, el servidor tiene instalado Ubuntu y tampoco supone ningún gasto económico puesto que viene en el gasto mensual del servidor.

Para el cálculo de costes de personal se ha consultado las tablas salariales de la UCA para el personal técnico de apoyo contratado laboral [CCOO, 2010]. El coste es de unos 20.298,30 euros anuales, lo equivale a 1.353,22 euros mensuales. Como el tiempo dedicado en este tiempo ha sido lo correspondiente a media jornada, tomaremos como un gasto mensual de 676.61 euros. De esta

manera, el coste anual será de unos 8119.32 euros y alcanzaremos unos gastos anuales de 20298 euros.

En la tabla 2.2 desglosamos los costes mensuales y totales del proyecto:

Unidades	Descripción	Coste Unitario	Coste total
1	Ingeniero Técnico Informático	8119.32 €/año	20298 €
1	Ordenador personal	286.67 €/año	625 €
1	Alquiler Host	10 €/mes	30 €
1	Costes indirectos	1217 €/año	3044 €
Total			23967 €

Tabla 2.2: Costes del desarrollo

2.5. Riesgos

En esta sección vamos a describir los posibles riesgos del proyecto ordenados de mayor a menor prioridad, indicando su posible impacto (efecto que la ocurrencia del citado riesgo tendría en el desarrollo del proyecto) y la probabilidad de ocurrencia. Además, indicamos el plan necesario para reducir los efectos del riesgo una vez se haya materializado o disminuir que este ocurra. Esta información viene recogida en la tabla 2.3.

2.6. Aseguramiento de calidad

Para asegurar el cumplimiento de la calidad de este se contempla los siguientes aspectos:

- Realización de controles o pruebas para asegurar su calidad.
- Tomar medidas de actuación relacionadas con el control de calidad.

Se realizará un análisis para cada uno de los hitos o tareas, con lo que es necesario un plan de verificación y validación dentro de los mismos para poder asegurar la calidad del producto y el correcto funcionamiento de los mismos. Estas comprobaciones se realizarán a lo largo del proyecto y así comprobar que todo está correcto y se cumple los requisitos establecidos.

Riesgo	Prob.	Magnitud	Plan a realizar
Tiempo infraestimado para el desarrollo de las tareas	30 %	5-8 semanas	Se comunica al tutor del proyecto del posible retraso, modificando la planificación temporal del proyecto o bien el plazo de entrega para poder finalizarlo.
Desarrollo de métodos y funciones software incorrectos	30 %	3-4 semanas	Se comprueba la calidad del código y su validez, para así evitar el denominado DRY (<i>Don't Repeat yourself</i>) y estudiar su re-utilización una vez corregido.
Cambios en API de Mendeley	30 %	3 semanas	Comunicar al tutor del proyecto del posible retraso y realizar una rápida investigación de los cambios producidos en la API de Mendeley, así como el desarrollo de la nueva implementación de la misma.
Cambios en API de motores de búsquedas de referencias bibliográficas	30 %	3 semanas	Comunicar al tutor del proyecto del posible retraso y realizar una rápida investigación de los cambios producidos en los motores de búsquedas, como el desarrollo de la nueva implementación de los mismos.
Elección incorrecta en las herramientas empleadas de desarrollo	15 %	2-3 semanas	Comunicar al tutor del proyecto del posible retraso y realizar una rápida investigación de las nuevas herramientas a emplear, así como de la búsqueda y lectura de documentación.
Paro del proyecto	20 %	1-2 semanas	Revisar la planificación y ajustarla, así como añadir horas extras una vez que se retome el desarrollo.
Problemas en los tests	30 %	1-2 semanas	Se analizan los tests y se planteará la nueva revisión de éstos y, en el caso correspondiente, crear unos nuevos.

Tabla 2.3: Lista de riesgos del proyecto

Parte II

Desarrollo

Capítulo 3

Requisitos del Sistema

En esta sección se detalla la situación actual de la organización y las necesidades de la misma, que originan el desarrollo o mejora de un sistema informático. Luego se presentan los objetivos y el catálogo de requisitos del nuevo sistema. Finalmente se describen las diferentes alternativas tecnológicas y el análisis de la brecha entre los requisitos planteados y la solución base seleccionada, si aplica.

3.1. Necesidades de Negocio

Los miembros del grupo de investigación SPI-FM (<http://tic195.uca.es/>) desean agilizar la realización de las revisiones sistemáticas de literatura. Por tanto, el objetivo principal es la realización de una aplicación web con la que un usuario pueda automatizar, en la mayor parte posible, todas las etapas de creación de revisiones sistemáticas de la literatura.

Una vez que el cliente tome la decisión de realizar una determinada investigación, deberá de realizar la búsqueda de la literatura que precisa para extraer conclusiones al final del mismo. Por ello, necesita un sistema con el que realizar varias búsquedas sobre varios motores de referencias específicos a la vez con las que posteriormente, el cliente estudiará, clasificará y exportará dicha información para obtener las conclusiones finales.

3.2. Objetivos del Sistema

El objetivo principal de este proyecto, será ayudar al usuario a seguir la mayor parte de las directrices de Barbara Kitchenham, ya que el estudio del arte de una área de investigación pre-determinada no puede ser automatizado. Sin embargo, podemos ayudar al usuario a quitarle tiempo de búsqueda de recursos a través de un buscador de referencias en motores de búsquedas, que en posteriores capítulos describiremos, y ayudar al usuario a determinar las conclusiones finales por medio de la exportación de la información a través de gráficos o ficheros en diferentes formatos.

Todo este proceso será desarrollado a través de una aplicación web donde el usuario podrá crear revisiones sistemáticas de la literatura, realizar búsquedas, clasificar los documentos y poder exportar la información a través de varios medios. Podemos ver más detalles de estos objetivos en tablas 3.1, 3.2, 3.3 y 3.4.

OBJ-01	Creación de una aplicación web para realizar Revisiones Sistemáticas de la Literatura.
Descripción	Se debe diseñar una aplicación web donde el usuario podrá crear revisiones sistemáticas de la literatura.

Tabla 3.1: OBJ-01: Creación de una aplicación web para realizar Revisiones Sistemáticas de la Literatura.

OBJ-02	Realizar Búsquedas de Referencias Bibliográficas.
Descripción	La aplicación web deberá poder realizar varias búsquedas de referencias bibliográficas con la información predefinida por el usuario y almacenando dichos recursos en un sistema de gestión de referencias (Mendeley) para su futuro estudio.

Tabla 3.2: OBJ-02: Realizar Búsquedas de Referencias Bibliográficas.

OBJ-03	Clasificar las referencias bibliográficas.
Descripción	La aplicación web deberá poder ayudar al usuario a clasificar las referencias bibliográficas encontradas en las búsquedas de cada una de las revisiones sistemáticas de la literatura.

Tabla 3.3: OBJ-03: Clasificar las referencias bibliográficas.

OBJ-04	Exportar información de los SLR y conclusiones finales.
Descripción	La aplicación web deberá poder ayudar al usuario a realizar un informe con las conclusiones finales del SLR a través de gráficos que reflejen la situación del mismo o diferentes ficheros de texto.

Tabla 3.4: OBJ-04: Exportar información de los SLR y conclusiones finales.

3.3. Catálogo de Requisitos

A continuación vamos a detallar el conjunto de requisitos específicos del sistema a desarrollar y con el que se pretende satisfacer las necesidades del cliente.

3.3.1. Requisitos funcionales

Los requisitos funcionales que ofrece el sistema estarán separados en dos grupos, dependiendo de si estamos hablando de un cliente que además, tenga permisos de administrador (Rol Administrador) o no (Rol Usuario).

■ Rol Usuario

- Creación de revisiones sistemáticas de la literatura
- Realizar búsquedas de referencias bibliográficas para los SLR.
- Clasificar las referencias bibliográficas por medio de criterios de clasificación aportados por el usuario.

- Obtener información de las referencias bibliográficas, así como la posibilidad de insertar nuevos campos o metadatos sobre los mismo.
- Exportar la información obtenida del estudio de la investigación a través de varios formatos de salida.
- Visualizar gráficos con la información obtenida de todas las búsquedas.

■ **Rol Administrador**

- Todas las tareas del Rol Usuario.
- Gestión de usuarios del sistema.
- Gestión de errores en procesos de búsquedas.
- Gestión de motores de búsquedas de referencias bibliográficas.
- Gestión API REST de Mendeley con la aplicación.

3.3.2. Requisitos no funcionales

Descripción de otros requisitos que el sistema deberá satisfacer:

- Todos los módulos de la aplicación web deberán estar disponibles para su instalación por parte de la UCA. Para ello, se ha decidido subir toda la aplicación en un repositorio al que puedan acceder.
- La aplicación deberá tener un alto grado de accesibilidad y usabilidad, ya que se espera que en el futuro haya personas que puedan acceder y modificar o mejorar cualquier componente del sistema.
- El sistema deberá ser seguro, las revisiones sistemáticas de cada usuario tienen acceso restringido a cualquier otro a no ser que tenga permisos para hacerlo.
- El sistema deberá ser portable en cualquiera de los navegadores web.
- El proceso de búsqueda de referencias deberá ser lo más transparente posible al usuario.

3.3.3. Reglas de negocio

Para el desarrollo del sistema, se ha indicado las siguientes reglas de negocio:

- El usuario debe estar registrado en Mendeley. De esta manera, el usuario dispondrá de un gestor de referencias donde la aplicación web insertará toda la documentación que la aplicación web encuentre.
- La aplicación debe permitir la posibilidad de insertar un nuevo motor de búsqueda, para ello se explicará más adelante como un usuario administrador del sistema puede descargar el código de la aplicación e insertarlo.

3.3.4. Requisitos de información

Describimos a los requisitos de gestión de información (datos) que el sistema debe gestionar en las tablas 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11 y 3.12.

IRQ-01	Información de usuarios.
Objetivos asociados	OBJ-01 3.1
Requisitos asociados	-
Datos específicos	Email Mendeley Password Mendeley Rol
Estabilidad	Alta
Comentarios	Información de los usuarios para hacer login con Mendeley.

Tabla 3.5: IRQ-01: Requisitos de información de los usuarios.

IRQ-02	Información de revisiones sistemáticas de la literatura.
Objetivos asociados	OBJ-01 3.1
Requisitos asociados	-
Datos específicos	Identificador Título Justificación
Estabilidad	Alta
Comentarios	Información de las revisiones sistemáticas de la literatura.

Tabla 3.6: IRQ-02: Requisitos de información de las revisiones sistemáticas de la literatura.

IRQ-03	Información de las búsquedas de referencias bibliográficas de un SLR.
Objetivos asociados	OBJ-02 3.2
Requisitos asociados	-
Datos específicos	Identificador Términos búsqueda Motores Búsquedas Total referencias a buscar Rango años
Estabilidad	Alta
Comentarios	Información de las búsquedas de referencias bibliográficas de un SLR.

Tabla 3.7: IRQ-03: Requisitos de información de las búsquedas de referencias bibliográficas de un SLR.

3.4. Alternativas de Solución

Para el desarrollo de esta aplicación, se pensó en la posibilidad de crear un **DSL** textual (**Domain Specific Language**) donde el usuario por línea de comandos escribiera las instrucciones para conectar con el motor de búsqueda de referencias y descargar las referencias encontradas. Se comenzó a trabajar con Eclipse como IDE de desarrollo, pero era un entorno complejo para desarrollar DSL debido a la poca documentación que existe en la actualidad. Además, la creación de un SLR sería llevada mejor a cabo si el usuario puede ver en todo momento la realización del mismo así como todos los pasos y directrices necesarios para el proceso.

IRQ-04	Información de las referencias bibliográficas de un SLR.
Objetivos asociados	OBJ-01 3.1 y OBJ-02 3.2
Requisitos asociados	IRQ-03 3.7
Datos específicos	Identificador Título Autores Publicación Año publicación Idioma Abstract Palabras clave enlaces de referencia Páginas Volumen Criterio
Estabilidad	Alta
Comentarios	Información de las referencias bibliográficas de un SLR.

Tabla 3.8: IRQ-04: Requisitos de información de las referencias bibliográficas de un SLR.

IRQ-05	Información de criterios de clasificación de un SLR.
Objetivos asociados	OBJ-01 3.1, OBJ-02 3.2 y OBJ-03 3.3
Requisitos asociados	IRQ-03 3.7 y IRQ-04 3.8
Datos específicos	Identificador Nombre Total referencias
Estabilidad	Alta
Comentarios	Información de los criterios de clasificación de una revisión sistemática de la literatura.

Tabla 3.9: IRQ-05: Requisitos de información de los criterios de clasificación de un SLR.

IRQ-06	Información de preguntas de investigación de un SLR.
Objetivos asociados	OBJ-01 3.1
Requisitos asociados	IRQ-02 3.6
Datos específicos	Identificador Enunciado
Estabilidad	Alta
Comentarios	Información de las preguntas de investigación asociadas a una revisión sistemática de la literatura.

Tabla 3.10: IRQ-06: Requisitos de información de las preguntas de investigación de un SLR.

Finalmente, se tomó la decisión de realizar una aplicación web con el que los investigadores tendrán un fácil acceso y de manera visual obtener las conclusiones finales y, además, de un fácil desarrollo y mantenimiento gracias al framework *Groovy on Rails* que nos permite aplicar el patrón *MVC* y realizar un mejor mantenimiento de la aplicación.

IRQ-07	Información de metadatos específicos de una referencia en un SLR
Objetivos asociados	OBJ-02 3.2 y OBJ-03 3.3
Requisitos asociados	IRQ-03 3.7 y IRQ-04 3.8
Datos específicos	Identificador Nombre Tipo Campo Valor defecto
Estabilidad	Alta
Comentarios	Información de los metadatos específicos de una referencia en un SLR.

Tabla 3.11: IRQ-07: Requisitos de información de los metadatos específicos de una referencia en un SLR.

IRQ-08	Información de datos a exportar de un SLR
Objetivos asociados	OBJ-02 3.2 y OBJ-04 3.4
Requisitos asociados	IRQ-02 3.6, IRQ-03 3.7, IRQ-04 3.8, IRQ-05 3.9, IRQ-06 3.9 y IRQ-07 3.11
Datos específicos	Datos SLR Datos Búsquedas Datos referencias
Estabilidad	Alta
Comentarios	Información a exportar de un SLR.

Tabla 3.12: IRQ-08: Requisitos de información de los datos a exportar de un SLR.

3.5. Solución Propuesta

El desarrollo propuesto será la realización de un sistema web, donde el usuario podrá conectar a través de su cuenta de Mendeley y crear las revisiones sistemáticas de la literatura que considere necesario.

La aplicación al ser en un entorno web, se desarrolla con un IDE denominado *Grails Tool Suite* donde por medio de los lenguajes *Groovy* y *Java* podemos aplicar el patrón MVC y desarrollar una aplicación fácil de mantener y modularizada.

Mendeley ofrece un gestor de referencias al usuario por medio de una API REST con la que desde una aplicación, como este proyecto de fin de carrera, pueda conectarse y obtener la información necesaria para realizar los estudios de la literatura y permitir la sincronización de datos entre ambas.

Capítulo 4

Análisis del Sistema

Esta sección cubre el análisis del sistema de información a desarrollar, haciendo uso del lenguaje de modelado UML.

4.1. Modelo Conceptual

A partir de los requisitos de información, se desarrollará un diagrama conceptual de clases UML. En él identificaremos las clases, atributos y relaciones entre ellas más significativas tal y como podemos ver en la figura 4.1.

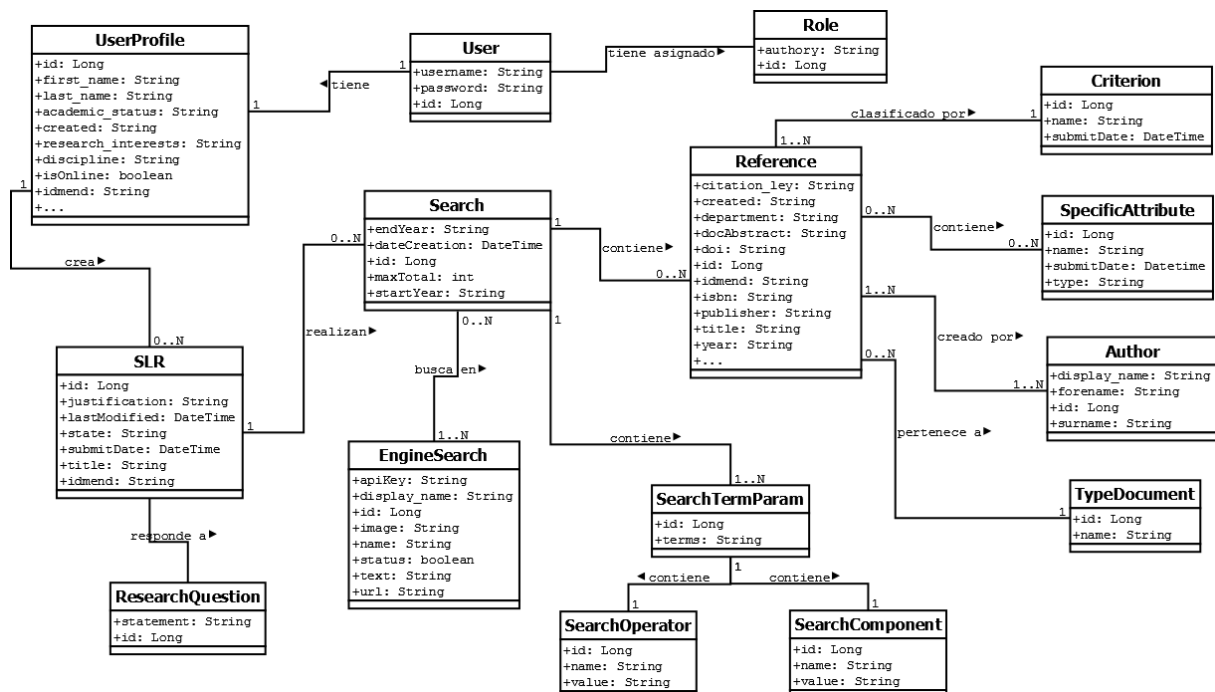


Figura 4.1: Diagrama conceptual de clases UML.

4.2. Modelo de Casos de Uso

A partir de los requisitos funcionales descritos en apartados anteriores, se emplearan los casos de uso como mecanismo para representar las interacciones entre los actores y el sistema a desarrollar.

4.2.1. Actores

Los diferentes roles que con el sistema pueden interactuar son los siguientes:

- Investigador. Esta figura representa el usuario principal del sistema web. Una persona con este rol podrá realizar revisiones sistemáticas de la literatura y efectuar todas las tareas relacionadas con ellas.
- Administrador. Esta figura podrá realizar todas las funciones del rol investigador, junto con el mantenimiento de gestión de los usuarios y los errores de búsquedas de referencias bibliográficas producidas en el sistema.

4.2.2. Diagramas y especificación de casos de uso

En esta sección se mostrarán los diagramas de casos de uso de la aplicación web (4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9), así como la especificación de los mismos por medio de tablas descriptivas.

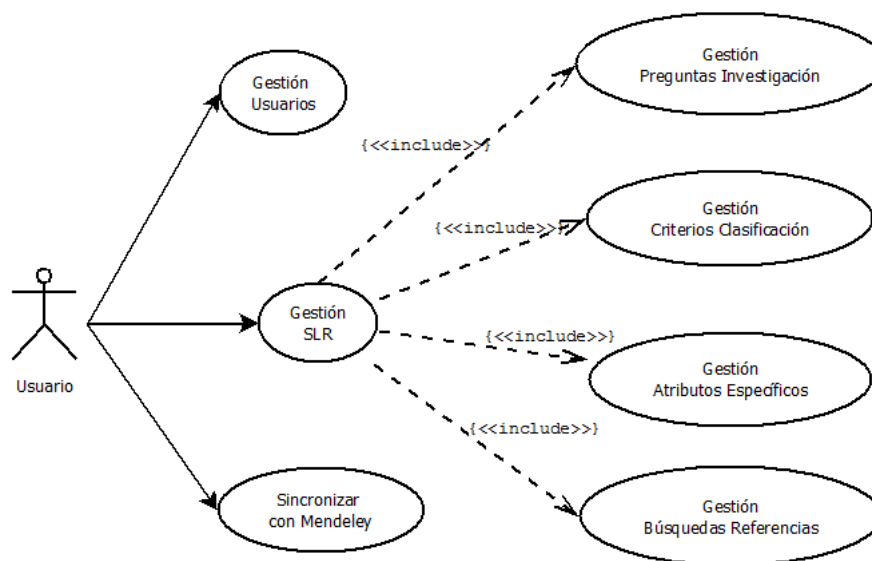


Figura 4.2: Diagrama casos de usos del sistema.

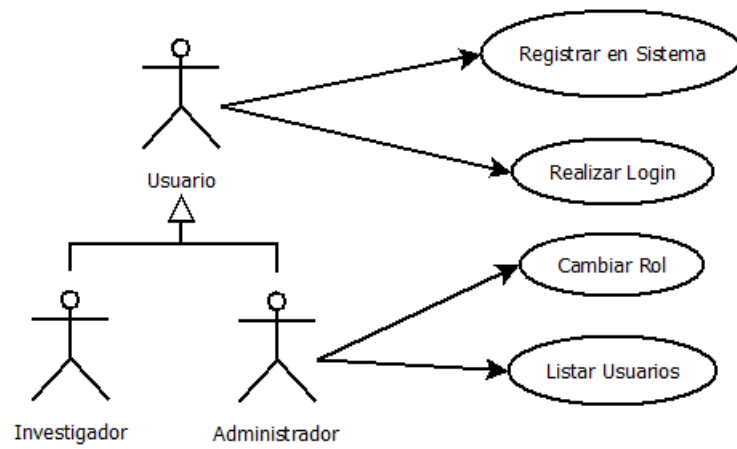


Figura 4.3: Diagrama casos de usos de la gestión de usuarios.

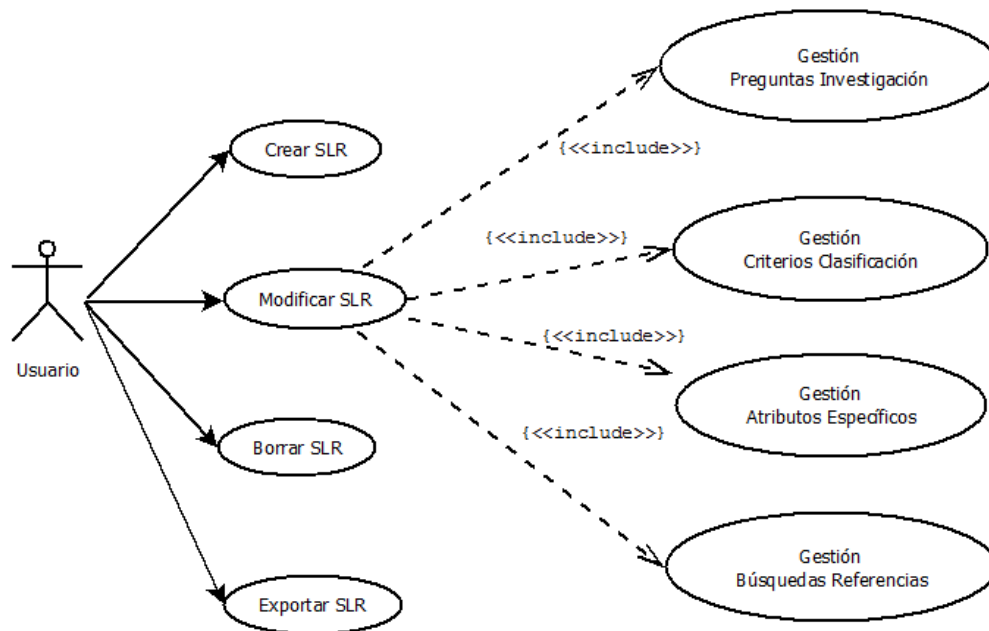


Figura 4.4: Diagrama casos de usos de la gestión de revisiones sistemáticas.

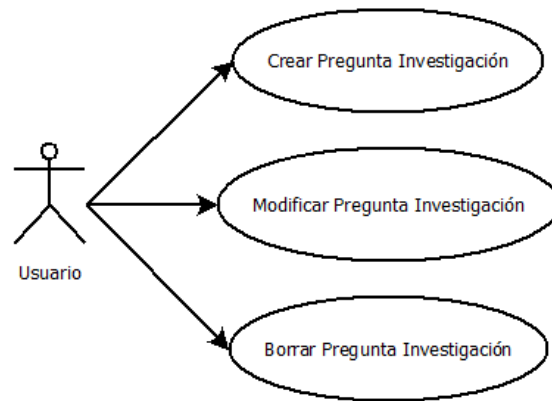


Figura 4.5: Diagrama casos de usos de la gestión de preguntas de investigación.

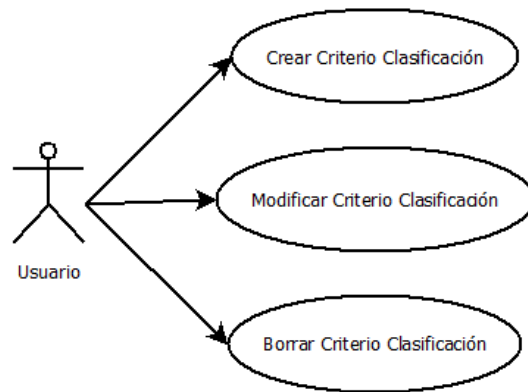


Figura 4.6: Diagrama casos de usos de la gestión de criterios de clasificación.

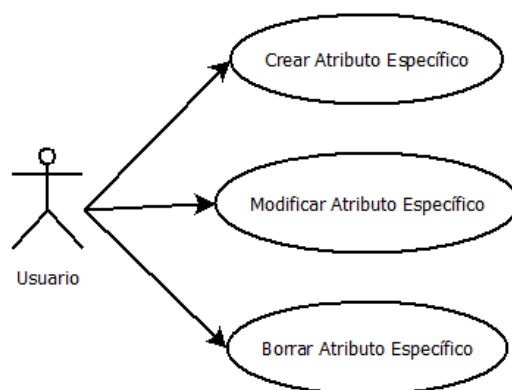


Figura 4.7: Diagrama casos de usos de la gestión de atributos específicos.

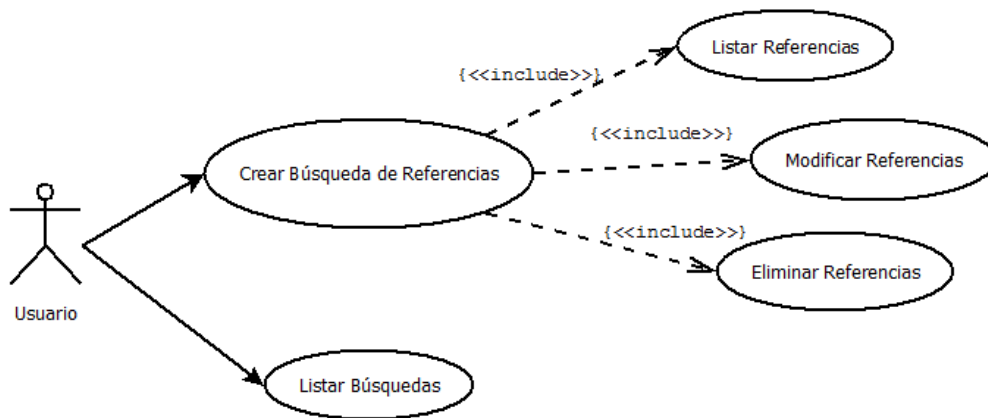


Figura 4.8: Diagrama casos de usos de la gestión de búsquedas de referencias bibliográficas.

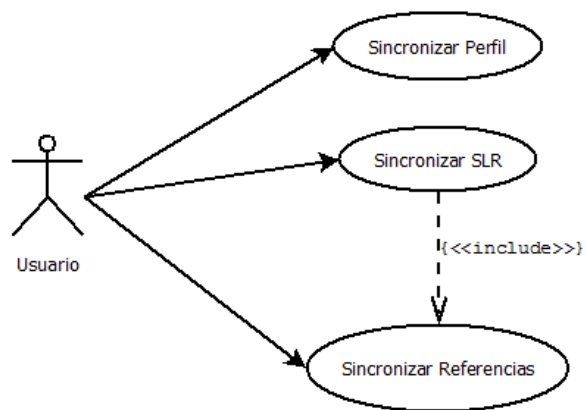


Figura 4.9: Diagrama casos de usos de la sincronización con Mendeley.

Nombre	CU-01 Registrar en sistema
Descripción	El usuario se registra en el sistema por primera vez para poder crear revisiones sistemáticas de la literatura.
Precondición	El usuario tiene una cuenta de Mendeley y no se ha logado antes en el sistema.
Postcondición	El usuario tiene acceso a la aplicación web.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario introduce el email y contraseña de la cuenta Mendeley. 2. El sistema valida que el email y la contraseña sean correctas redirigiendo al usuario a la pantalla principal de la aplicación web.
Escenarios alternativos	<p>2 a El sistema verifica que el email y la contraseña no son correctas. Volvemos al paso 1.</p>

Tabla 4.1: CU-01 Registrar en sistema

Nombre	CU-02 Realizar login
Descripción	El usuario se identifica en el sistema web.
Precondición	El usuario debe estar registrado en el sistema web.
Postcondición	El usuario tiene acceso a la aplicación web.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario introduce el email y contraseña de la cuenta Mendeley. 2. El sistema valida que el email y la contraseña sean correctas redirigiendo al usuario a la pantalla principal de la aplicación web.
Escenarios alternativos	<p>2 a El sistema verifica que el email y la contraseña no son correctas. Volvemos al paso 1.</p>

Tabla 4.2: CU-02 Realizar login

Nombre	CU-03 Cambiar rol
Descripción	El administrador del sistema puede cambiar de rol a otro usuario.
Precondición	El usuario debe estar logado en el sistema y poseer rol administrador.
Postcondición	El administrador cambiar de rol a otro usuario.
Actores	Administrador
Escenario principal	<ol style="list-style-type: none"> 1. El administrador del sistema busca al usuario a través de su email. 2. El sistema ofrece toda la información del usuario. 3. El administrador selecciona la opción de cambiar de rol al usuario. 4. El sistema realiza el cambio y muestra un mensaje de que el proceso se ha realizado correctamente.
Escenarios alternativos	–

Tabla 4.3: CU-03 Cambiar rol

Nombre	CU-04 Listar Usuarios
Descripción	El administrador puede obtener una lista de todos los usuarios registrados en la aplicación web.
Precondición	El administrador está logado en el sistema y tiene rol administrador.
Postcondición	El administrador obtiene un listado completo de los usuarios registrados en el sistema.
Actores	Administrador
Escenario principal	<ol style="list-style-type: none"> 1. El administrador elige la opción de listado de usuarios. 2. El sistema muestra el listado de todos los usuarios registrados en la aplicación.
Escenarios alternativos	–

Tabla 4.4: CU-04 Listar usuarios

Nombre	CU-05 Crear SLR
Descripción	El usuario elige la opción de crear una revisión sistemática de la literatura.
Precondición	El usuario debe estar registrado en el sistema web.
Postcondición	El usuario crea una revisión sistemática de la literatura.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción Crear SLR. 2. El sistema muestra una ventana donde se debe insertar el título y la justificación del mismo. 3. El usuario introduce el título y justificación del SLR a crear. 4. El sistema comprueba que no haya un SLR con el mismo título y que el formato del mismo sea correcto mostrando un mensaje de confirmación de la creación del SLR.
Escenarios alternativos	<p>4 a El sistema verifica que el título no cumple el formato correcto o el título está siendo usado por otro SLR. Volvemos al paso 2.</p>

Tabla 4.5: CU-05 Crear SLR

Nombre	CU-06 Modificar SLR
Descripción	El usuario puede realizar modificaciones de una revisión sistemática de la literatura realizando operaciones sobre las preguntas de investigación, criterios de clasificación, atributos específicos o búsquedas de referencias.
Precondición	El usuario debe estar registrado en el sistema web y el SLR a modificar existe.
Postcondición	El usuario modifica una revisión sistemática de la literatura.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario realiza operaciones sobre las preguntas de investigación. Ver tablas 4.9, 4.10, 4.11. 2. El usuario realiza operaciones sobre los criterios de clasificación. Ver tablas 4.12, 4.13, 4.14, . 3. El usuario realiza operaciones sobre los atributos específicos. Ver tablas 4.15, 4.16, 4.17. 4. El usuario realiza operaciones sobre las búsquedas de referencias 4.18
Escenarios alternativos	Ver escenarios alternativos de casos de uso..

Tabla 4.6: CU-06 Modificar SLR

Nombre	CU-07 Borrar SLR
Descripción	El usuario elige la opción de borrar una revisión sistemática de la literatura.
Precondición	El usuario debe estar registrado en el sistema web y el SLR a borrar existe.
Postcondición	El usuario borra una revisión sistemática de la literatura del sistema.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción Borrar SLR. 2. El sistema muestra una ventana donde pregunta al usuario si confirma el borrado de la revisión sistemática de la literatura. 3. El usuario confirma el borrado del SLR. 4. El sistema elimina el SLR del sistema mostrando un mensaje de confirmación del borrado.
Escenarios alternativos	<p>3 a El usuario elige la opción Cancelar. El sistema mostrará un listado de las revisiones sistemáticas del usuario.</p>

Tabla 4.7: CU-07 Borrar SLR

Nombre	CU-08 Exportar SLR
Descripción	El usuario elige la opción de exportar datos de una revisión sistemática de la literatura.
Precondición	El usuario debe estar registrado en el sistema web y la revisión sistemática de la literatura debe ser propia del mismo.
Postcondición	El usuario obtiene los datos de una revisión sistemática de la literatura a través de un fichero de salida.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción de exportar una revisión sistemática de la literatura a través de un fichero o a través de gráficos. 2. El sistema exporta los datos en el formato elegido por el usuario.
Escenarios alternativos	—

Tabla 4.8: CU-08 Exportar SLR

Nombre	CU-09 Crear Pregunta de investigación
Descripción	El usuario puede introducir preguntas de investigación en una revisión sistemática de la literatura.
Precondición	El usuario debe estar registrado en el sistema web y la revisión sistemática debe ser propia del usuario.
Postcondición	El usuario introduce una pregunta de investigación en una revisión sistemática de la literatura.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción Insertar Pregunta de Investigación. 2. El sistema pide por pantalla el enunciado de la pregunta. 3. El usuario introduce el enunciado de la pregunta. 4. El sistema comprueba que el enunciado cumpla un formato y muestra un mensaje satisfactorio de la creación.
Escenarios alternativos	<p>4 a El sistema verifica que el enunciado no cumple el formato. El sistema muestra un mensaje de error y se vuelve al paso 2.</p>

Tabla 4.9: CU-09 Crear Pregunta de investigación

Nombre	CU-10 Modificar Pregunta de investigación
Descripción	El usuario elige la opción de modificar una pregunta de investigación de una revisión sistemática de la literatura elegida.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario modifica la pregunta de investigación de un determinado SLR.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción modificar pregunta de investigación. 2. El sistema muestra una pantalla con el enunciado de la pregunta a modificar. 3. El usuario introduce el enunciado de la pregunta a modificar. 4. El sistema comprueba que el enunciado de la pregunta cumple un formato y muestra un mensaje de confirmación al usuario.
Escenarios alternativos	<p>4 a El sistema verifica que el enunciado no cumple un formato correcto. Volvemos al paso 2 junto con un mensaje de alerta al usuario.</p>

Tabla 4.10: CU-10 Modificar pregunta de investigación

Nombre	CU-11 Borrar pregunta de investigación
Descripción	El usuario elige la opción de borrar una pregunta de investigación de una revisión sistemática.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática.
Postcondición	El usuario realiza el borrado de una pregunta de investigación.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción Borrar pregunta de investigación. 2. El sistema muestra una ventana donde se pide la confirmación de la pregunta de investigación. 3. El usuario realiza la confirmación del borrado. 4. El sistema borra la pregunta del sistema y muestra un mensaje de confirmación del borrado al usuario.
Escenarios alternativos	–

Tabla 4.11: CU-11 Borrar pregunta investigación

Nombre	CU-12 Crear Criterio de clasificación
Descripción	El usuario elige la opción de crear un criterio de clasificación de referencias.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario crea un criterio de clasificación de referencias.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el 4.9 pero adaptado a los criterios de clasificación de referencias.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.9 pero adaptados a los criterios de clasificación de referencias.

Tabla 4.12: CU-12 Crear criterio de clasificación

Nombre	CU-13 Modificar Criterio de clasificación
Descripción	El usuario elige la opción de modificar un criterio de clasificación de referencias.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario modifica un criterio de clasificación de referencias.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el 4.10 pero adaptado a los criterios de clasificación de referencias.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.10 pero adaptados a los criterios de clasificación de referencias.

Tabla 4.13: CU-13 Modificar criterio de clasificación

Nombre	CU-14 Borrar Criterio de clasificación
Descripción	El usuario elige la opción de borrar un criterio de clasificación de referencias.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario borra un criterio de clasificación de referencias.
Actores	Usuario
Escenario principal	<ul style="list-style-type: none"> ■ El escenario principal es el mismo que el 4.11 pero adaptado a los criterios de clasificación de referencias. ■ Todas las referencias bibliográficas que contengan dicha referencia, tendrán un nuevo criterio 'incluido'.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.11 pero adaptados a los criterios de clasificación de referencias.

Tabla 4.14: CU-14 Borrar criterio de clasificación

Nombre	CU-15 Crear atributo específico
Descripción	El usuario elige la opción de crear un atributo específico.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario crea un atributo específico para las referencias de un slr.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el 4.9 pero adaptado a los atributos específicos.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.9 pero adaptados a los atributos específicos.

Tabla 4.15: CU-15 Crear atributo específico

Nombre	CU-16 Modificar atributo específico
Descripción	El usuario elige la opción de modificar atributo específico.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario modifica un atributo específico.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el 4.10 pero adaptado a los atributos específicos.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.9 pero adaptados a los criterios de clasificación de referencias.

Tabla 4.16: CU-16 Modificar atributo de especificación

Nombre	CU-17 Borrar atributo específico
Descripción	El usuario elige la opción de eliminar atributo específico.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario elimina un atributo específico.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el 4.11 pero adaptado a los atributos específicos.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.11 pero adaptados a los criterios de clasificación de referencias.

Tabla 4.17: CU-17 Eliminar atributo de especificación

Nombre	CU-18 Crear búsquedas de referencias
Descripción	El usuario elige la opción de crear una búsqueda de referencias.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario realiza una búsqueda de referencias.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción de creación de una búsqueda de referencias 2. El sistema muestra todos los campos que debe de rellenar. 3. El usuario introduce los términos de búsquedas, elige los motores de búsquedas, el total de referencias a buscar y en qué intervalo de años deben pertenecer. 4. El sistema realiza la búsqueda, indicando el estado de la misma y confirmando que se ha realizado correctamente cuando la búsqueda ha sido finalizada.
Escenarios alternativos	<ol style="list-style-type: none"> 4 a El sistema ha detectado algún error en el formato de los datos de entrada. El sistema alerta al usuario y se vuelve al paso 3. 4 b El sistema no puede realizar una búsqueda debido a problemas de conexión con los motores de búsquedas. El sistema guarda el error, advierte al usuario y el administrador del sistema puede gestionarlo.

Tabla 4.18: CU-18 Crear búsquedas de referencias

Nombre	CU-19 Listar Referencias
Descripción	El usuario elige la opción de listar las referencias bibliográficas.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario lista todas las referencias bibliográficas de una revisión sistemática de la literatura.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción de listar las referencias. 2. El sistema muestra todas las referencias bibliográficas que pertenecen a una revisión sistemática.
Escenarios alternativos	–

Tabla 4.19: CU-19 Listar referencias

Nombre	CU-20 Modificar referencias
Descripción	El usuario elige la opción de modificar referencias bibliográficas.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario modifica una referencia bibliográfica.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el 4.10 pero adaptado a las referencias bibliográficas.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.10 pero adaptados a las referencias bibliográficas.

Tabla 4.20: CU-20 Modificar referencias

Nombre	CU-21 Eliminar referencias
Descripción	El usuario elige la opción de eliminar referencias bibliográficas.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario elimina una referencia bibliográfica.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el 4.11 pero adaptado a las referencias bibliográficas.
Escenarios alternativos	Los escenarios alternativos son los mismos que el 4.11 pero adaptados a las referencias bibliográficas.

Tabla 4.21: CU-21 Eliminar referencias

Nombre	CU-22 Listar Búsquedas
Descripción	El usuario elige la opción de listar las búsquedas realizadas en una revisión sistemática de la literatura.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario lista todas las búsquedas de una revisión sistemática de la literatura.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que el de 4.19 pero adaptado a las búsquedas de una revisión sistemática de la literatura.
Escenarios alternativos	Los escenarios alternativos son los mismos que los de 4.19 pero adaptados a las búsquedas de una revisión sistemática de la literatura.

Tabla 4.22: CU-22 Listar Búsquedas

Nombre	CU-23 Sincronizar perfil
Descripción	El usuario elige la opción de sincronizar los datos de su perfil con los que tiene almacenados en su cuenta de Mendeley.
Precondición	El usuario debe estar registrado en el sistema web.
Postcondición	El usuario sincroniza los datos de su perfil.
Actores	Usuario
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige la opción Sincronizar Perfil. 2. El sistema conecta vía API REST con Mendeley y actualiza la información con los datos que proporciona Mendeley. El sistema además, mostrará un mensaje de confirmación.
Escenarios alternativos	<ol style="list-style-type: none"> 2 a El sistema detecta un error de sincronización. El sistema avisa al usuario y le indica que debe intentarlo más tarde.

Tabla 4.23: CU-23 Sincronizar Perfil

Nombre	CU-24 Sincronizar SLR
Descripción	El usuario elige la opción de sincronizar los datos de un SLR con los proporcionados por Mendeley.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario sincroniza los datos del SLR.
Actores	Usuario
Escenario principal	El escenario principal es el mismo que 4.23 pero adaptado a las revisiones sistemáticas de la literatura.
Escenarios alternativos	Los escenarios alternativos son los mismos que 4.23 pero adaptados a las revisiones sistemáticas de la literatura.

Tabla 4.24: CU-24 Sincronizar SLR

Nombre	CU-25 Sincronizar Referencias
Descripción	El usuario elige la opción de sincronizar los datos de una referencia con los proporcionados por Mendeley.
Precondición	El usuario debe estar registrado en el sistema web y ser propietario de la revisión sistemática de la literatura.
Postcondición	El usuario sincroniza los datos de la referencia
Actores	Usuario
Escenario principal	El escenario principal es el mismo que 4.23 pero adaptado a las referencias bibliográficas.
Escenarios alternativos	Los escenarios alternativos son los mismos que 4.23 pero adaptados a las referencias bibliográficas.

Tabla 4.25: CU-24 Sincronizar Referencias

4.3. Modelo de Interfaz de Usuario

En esta sección se incluye un prototipo de baja fidelidad (*mockup*) de la interfaz de usuario del sistema. Se han realizado *mockups* por medio de *Balsamiq* [myBalsamiq,]. En las figuras 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17 y 4.18 podemos ver algunas de estas interfaces del usuario.

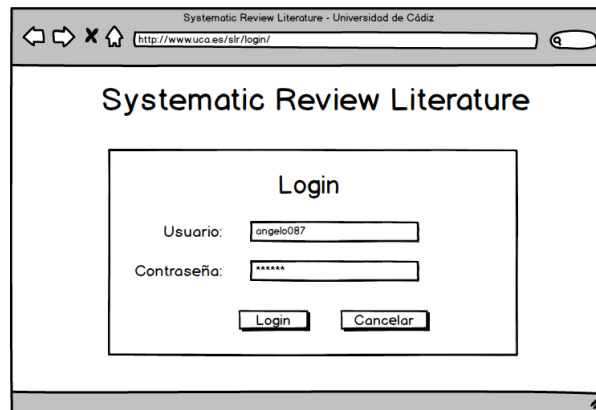


Figura 4.10: Pantalla login.

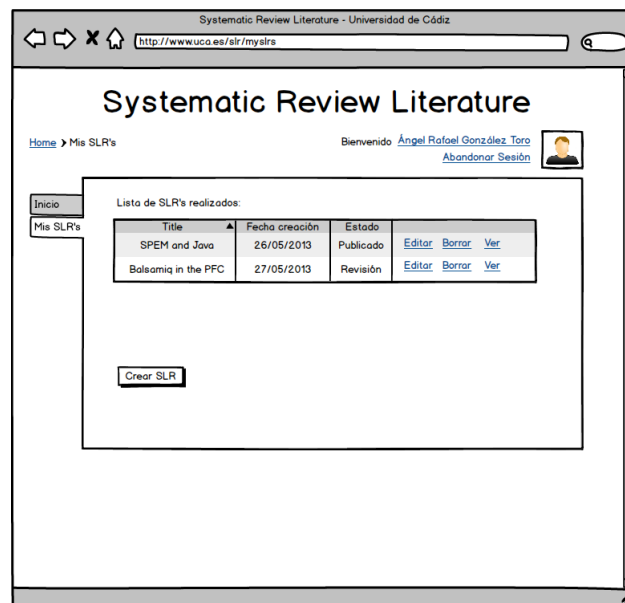


Figura 4.11: Pantalla Revisiones Sistemáticas Usuario.

Systematic Review Literature - Universidad de Cádiz

Home > Mis SLR's > Crear SLR

Bienvenido [Ángel Rafael González Toro](#)
[Abandonar Sesión](#)

[Inicio](#)
[Mis SLR's](#)

Crear Systematic Review Literature

PLANIFICACIÓN DE LA REVISIÓN

1. Identificación de Necesidad de la Revisión

Título:

Justificación:

En el año 2002, la OMG propuso un lenguaje para el modelado de procesos software, en línea con el enfoque de desarrollo MDA. Se trata de SPEM (Software & Systems Process Engineering Meta-Model).

Este lenguaje dispone de los elementos básicos para la definición de metodologías y procesos software: roles, actividades, tareas, herramientas, productos de trabajo, iteraciones, hitos, etc.

[Siguiente >](#)

Figura 4.12: Pantalla creación revisión sistemática

Systematic Review Literature - Universidad de Cádiz

Home > Mis SLR's > Crear SLR

Bienvenido [Ángel Rafael González Toro](#)
[Abandonar Sesión](#)

[Inicio](#)
[Mis SLR's](#)

Crear Systematic Review Literature

PLANIFICACIÓN DE LA REVISIÓN

2. Definición de preguntas de Investigación

Pregunta:

Pregunta:

Pregunta:

Pregunta:

[Añadir otra pregunta](#)

[< Atrás](#) [Siguiente >](#)

Figura 4.13: Pantalla creación preguntas de investigación

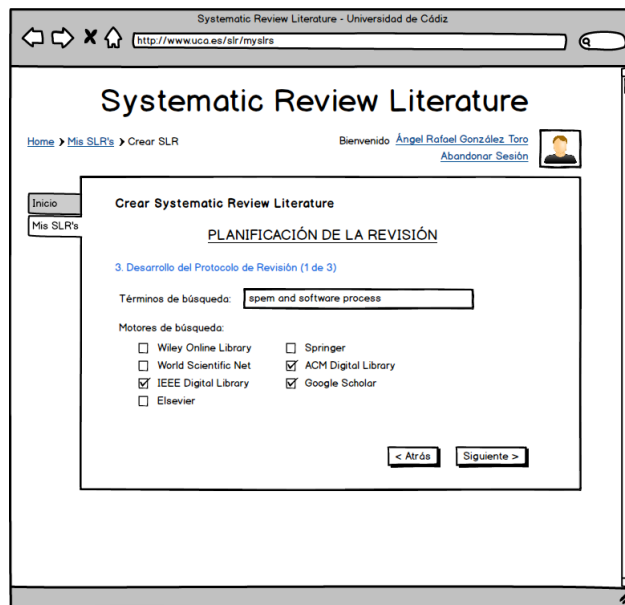


Figura 4.14: Pantalla creación búsquedas de referencias bibliográficas.

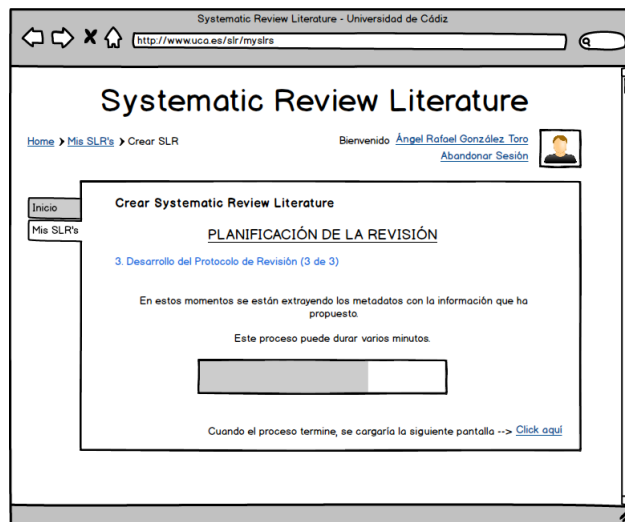


Figura 4.15: Pantalla de creación de búsquedas en progreso.

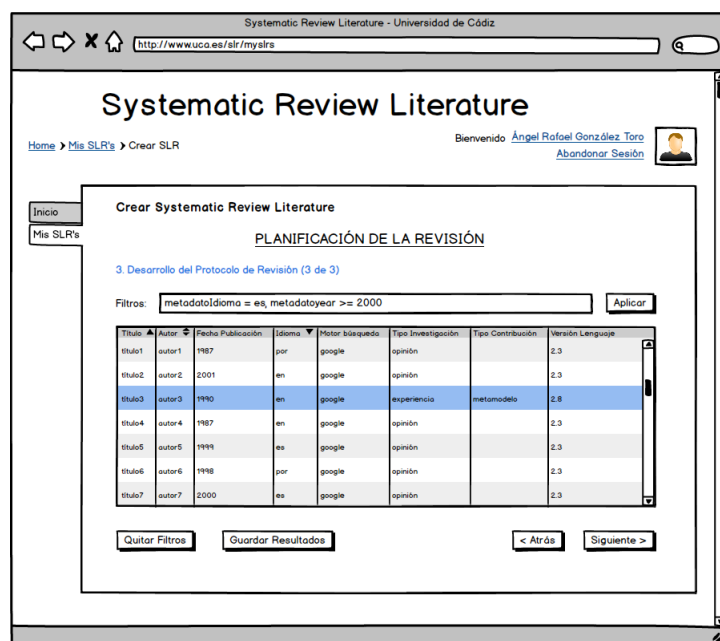


Figura 4.16: Pantalla Referencias bibliográficas.

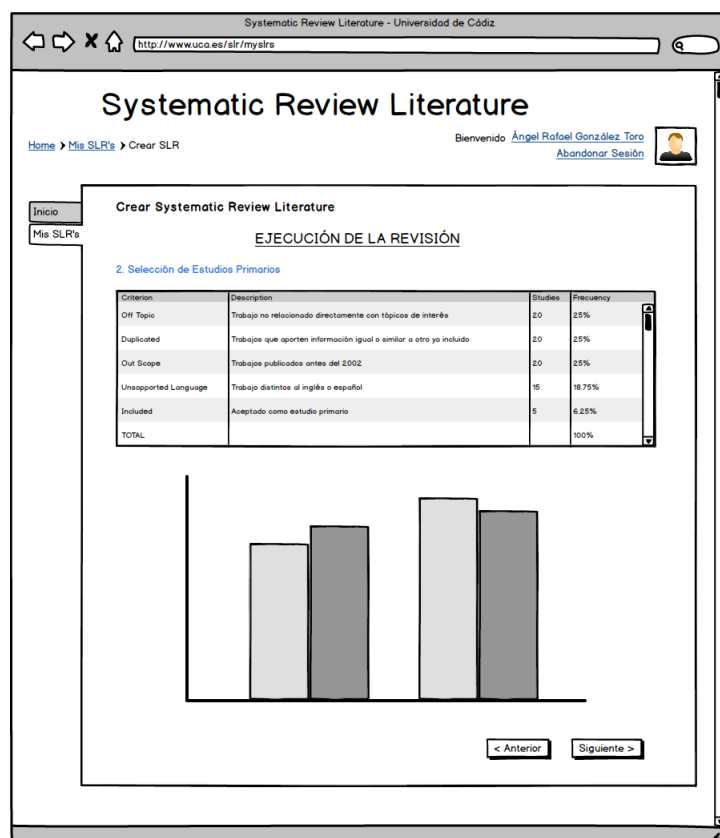


Figura 4.17: Pantalla exportación referencias bibliográficas y gráficos (I).

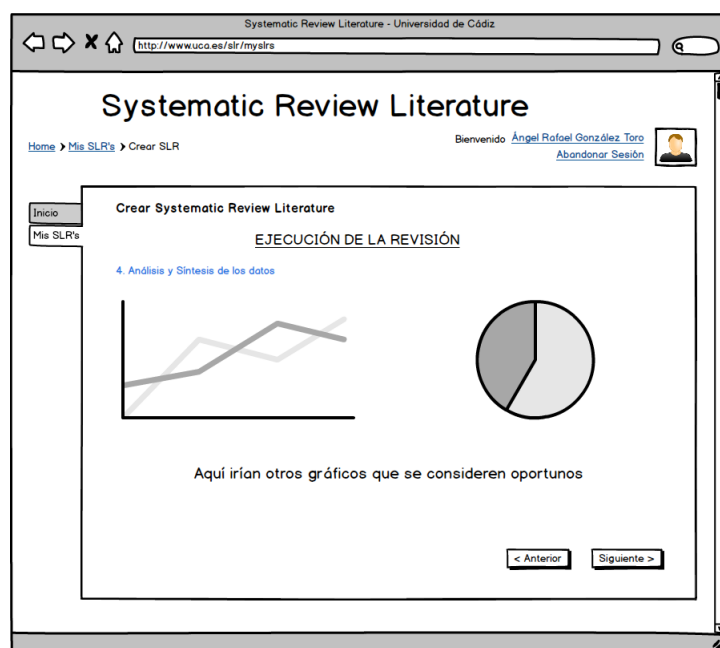


Figura 4.18: Pantalla exportación referencias bibliográficas y gráficos (II).

Capítulo 5

Diseño del Sistema

En esta sección se recoge la arquitectura general del sistema de información, la parametrización del software base, el diseño físico de datos, el diseño detallado de componentes software y el diseño detallado de la interfaz de usuario.

5.1. Arquitectura del Sistema

En esta sección definiremos la arquitectura general del sistema de información, especificando la infraestructura tecnológica necesaria para dar soporte al software y la estructura de los componentes que lo forman.

5.1.1. Arquitectura Física

Los componentes que compondrán la arquitectura física de esta aplicación se puede ver reflejado en las figura 5.1 y 5.2.

- **Navegador del cliente.** Un navegador estándar HTML capaz de soportar CSS, Javascript + Document Object Model, XML y XSLT. Este servirá como dispositivo de interfaz de usuario. Toda la interacción entre los usuarios y el sistema se realiza a través del navegador.
- **Servidor web.** El navegador del cliente accederá al sistema a través del servidor Web (Tomcat), el cual acepta las peticiones del cliente y ejecuta, si es necesario, los scripts del lado del servidor necesarios. El resultado, una página HTML formateada, será enviada al cliente.
- **Conexión HTTP.** Es el protocolo más común actualmente entre el cliente y el servidor.
- **Servidor de Aplicaciones.** Es el principal motor para ejecutar la lógica del negocio del lado del servidor.
- **Servidor de Base de datos.** Es la parte del sistema que mantiene el estado actual del negocio.
- **Servicio Web.** En este caso, Mendeley actuará de gestor de referencias del cuál podremos obtener la información de las revisiones sistemáticas (carpetas), ficheros (referencias) e información del perfil del usuario.

El servidor tiene las características descritas en la tabla 5.1. En él, se ha instalado Tomcat, Apache y MySQL para la base de datos.

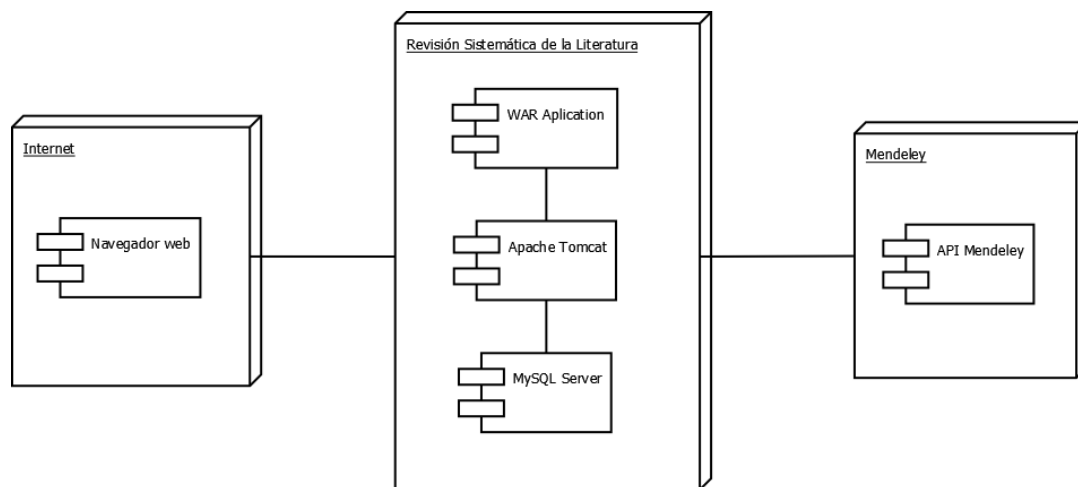


Figura 5.1: Arquitectura física del sistema (I).

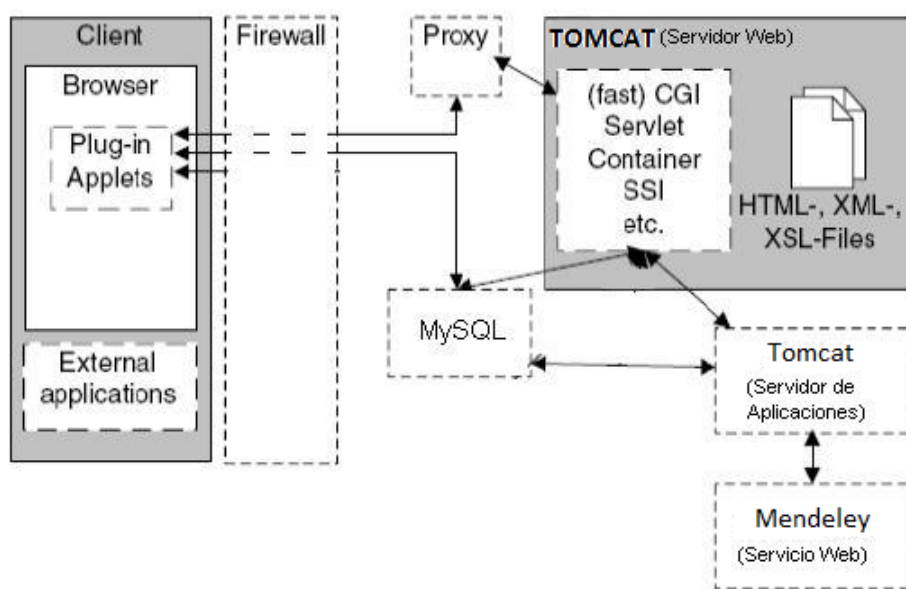


Figura 5.2: Arquitectura del sistema (II).

CPU	1vCore
RAM	2048 MB.
Storage	40 GB SSD.
Bandwidth	2000 GB

Tabla 5.1: Arquitectura del Servidor

5.1.2. Arquitectura Lógica

La arquitectura lógica de la aplicación web se va a regir en todo momento según el patrón **Modelo Vista Controlador (MVC)**. Éste separa los datos de la planificación, la interfaz del usuario y la lógica de control o negocio en tres modelos distintos.

- **Capas de datos.** Contiene los componentes que representan y gestionan los datos manejados por la aplicación. En el caso más típico, los objetos encargados de leer y escribir en base de datos.
- **Capa de presentación.** Los componentes de esta capa son responsables de mostrar al usuario el estado actual del modelo de datos, y presentarle las distintas acciones disponibles.
- **Capa de control.** Contendrá todos los componentes que reciban las órdenes del usuario, gestionan la aplicación de la lógica de negocio sobre el modelo de datos, y determinan qué vista debe mostrarse a continuación.
- **Capa de servicios.** Se trata de una cuarta capa que contiene los elementos encargados de implementar la lógica de negocio de nuestra aplicación.

Para el desarrollo de la aplicación, se ha decidido seguir la arquitectura del framework **Grails** (ver figura 5.3) que acopla correctamente el patrón MVC [Brito, 2009].

- **Modelo.** Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de éstos. El SGBD para gestionar los datos corresponde a este componente. En este sistema el mapeo se realiza a través de **GORM** (*Grails Object-Relational Mapping*) que a través de clases escritas en **Groovy** describe todo el modelo de la aplicación.
- **Vista.** Presenta el modelo en un formato adecuado para interactuar, normalmente la interfaz de usuario. La vista estará formada por un conjunto de páginas webs que facilitaran al usuario la interacción con el sistema. Las páginas se renderizan a partir de ficheros fuentes GSP (Groovy Server Pages) que mezcla etiquetas HTML con otras propias de Grails.
- **Controlador.** Responde a los eventos, generalmente provocados por el usuario a través de las vistas, ajustando los modelos. Estos controladores serán clases escritas en Groovy con cada una de las acciones posibles.

Grails es un framework para desarrollo de aplicaciones web construido sobre cinco fuertes pilares:

- Groovy. Para la creación de propiedades y métodos dinámicos en los objetos de la aplicación.

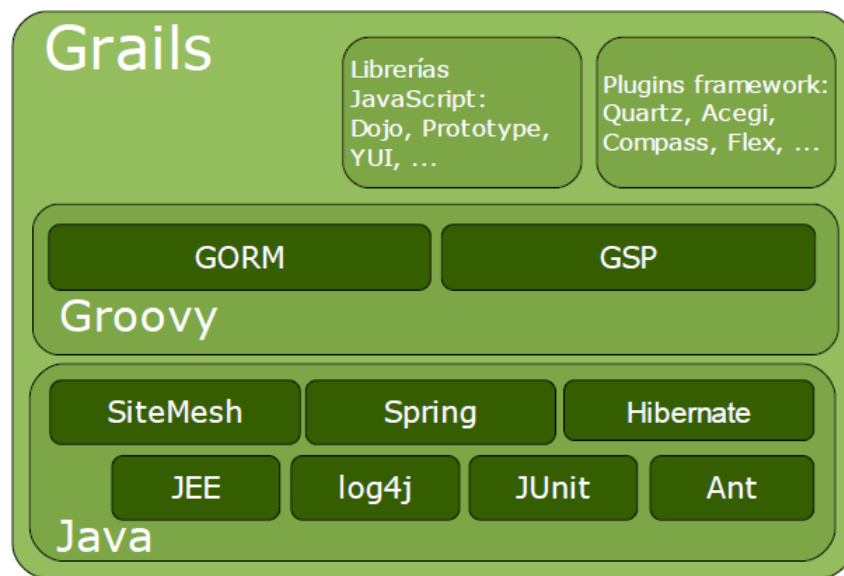


Figura 5.3: Arquitectura Grails.

- Spring. Para los flujos de trabajo e inyección de dependencias.
- Hibernate. Para la persistencia de los datos de la aplicación.
- SiteMesh. Para la composición de la vista.
- Ant. Para la gestión del proceso de desarrollo.

La estructura de esta aplicación será la siguiente:

- grails-app
 - conf: Archivos de configuración.
 - hibernate: Configuración de Hibernate.
 - spring: Configuración de Spring.
 - controllers: Controladores.
 - domain: Entidades (Clases de Dominio).
 - i18n: Message Bundles.
 - services: Servicios.
 - taglib: Librerías de etiquetas.
 - util: Clases de utilidad.
 - views: Vistas.
 - layouts: Layouts SiteMesh.
 - lib
 - scripts

- src
 - groovy: Clases Groovy.
 - java: Clases Java.
- test: Casos de prueba.
- web-app: Raíz de la aplicación web.

5.2. Parametrización del software base

Esta aplicación hace conexión a través de una API al gestor de referencias Mendeley. Para realizar este proceso, se necesita un usuario que actúe como administrador del sistema y que se conecte con sus datos al portal de Mendeley para indicar que esta aplicación pueda tener conexión con Mendeley.

Mendeley, dispone de una sección donde todos los usuarios pueden insertar la configuración necesaria para que Mendeley conecte con sus aplicaciones. Para ello, es necesario dirigirse a <http://dev.mendeley.com/> y a continuación elegir la opción *My Apps*.

Una vez que se ha logado en el sistema, podemos obtener un listado de aplicaciones (ver figura 5.4) con todas las aplicaciones que un usuario disponga. Mendeley pide por pantalla varios datos de configuración, pero es necesario recordar tres campos que serán necesarios para la configuración en nuestra aplicación como describiremos más adelante (marcados en negrita). Como podemos comprobar, se puede tener tantas configuraciones como se desee. Ésto podrá ser válido si, por ejemplo, tuviésemos varios entornos de desarrollo.

- Application name. Nombre de la aplicación.
- Description. Descripción de lo que realiza esta aplicación.
- **ID Cliente**. Identificador de esta aplicación en Mendeley. Es generado automáticamente por Mendeley.
- **Redirect URL**. Esta aplicación hará conexión con Mendeley a través de OAuth2, donde se nos pedirá unos datos que enviarán a una página de nuestra aplicación.
- **Client Secret**. Código privado del cliente que debe ser generado a través del botón *Generate secret*.

Una vez que hemos realizado la configuración en Mendeley, debemos insertar la información obtenida en nuestra aplicación. Para ello, Grails dispone de un fichero situado en *conf/Bootstrap.groovy* donde por medio de una clase de dominio realizada denominada *MendeleyApi* podemos parametrizar esta configuración tal y como podemos ver reflejado en la figura 5.5.

Si la aplicación se encuentra desplegada sobre el servidor, y siempre siendo administrador, podemos modificar esta configuración a través de un menú que dispone un usuario con este rol.

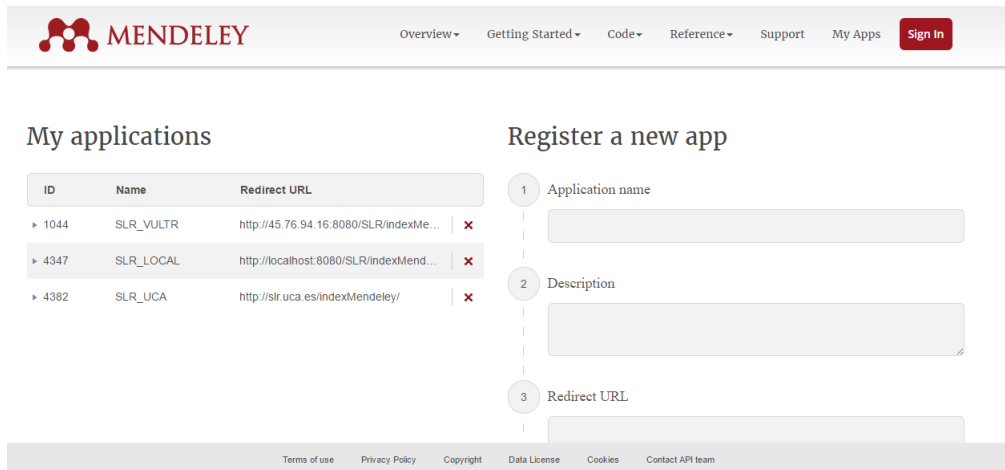


Figura 5.4: Registrar aplicación en Mendeley.

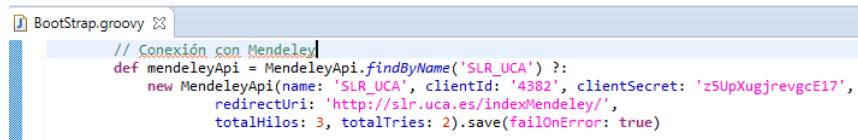


Figura 5.5: Registrar aplicación en Mendeley.

5.3. Diseño Físico de Datos

El almacenamiento de los datos físicos se va a realizar sobre el SGBD MySQL. La estructura física de datos que utilizará el sistema se define a partir del modelo conceptual de clases, de manera que teniendo presente los requisitos establecidos para el sistema de información y las particularidades del entorno tecnológico, se consiga un acceso eficiente de los datos. Podemos ver en la figura 5.6 todas las tablas empleadas en la base de datos, así como una descripción de las más importantes (Tablas 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13).

user: tabla de usuarios registrados en la aplicación.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
user_mendeley_id	Identificador de la tabla user_mendeley	BIGINT	No	No
user_profile_id	Identificador de la tabla user_profile	BIGINT	No	No
username	Nombre usuario.	VARCHAR	No	No

Tabla 5.2: Tabla user

user_mendeley: tabla de información de acceso a mendeley de los usuarios.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
access_token	Token acceso de mendeley vía OAuth2	VARCHAR	Si	No
token_type	Tipo de token vía OAuth2	VARCHAR	Si	No
email_mend	Email de acceso a mendeley	VARCHAR	No	No
pass_mend	Password de acceso a mendeley cifrado.	VARCHAR	No	No
refresh_token	token actualizado vía oauth2	VARCHAR	No	No
expires_in	Tiempo máximo de duración del token vía OAuth2	VARCHAR	No	No

Tabla 5.3: Tabla user_mendeley

user_profile: tabla de información de los usuarios registrados en mendeley.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
academic_status	Estado académico	VARCHAR	Si	No
biography	Biografía	VARCHAR	Si	No
created	Fecha creación en mendeley	VARCHAR	No	No
discipline	Disciplina a la que pertenece.	VARCHAR	No	No
display_name	Nombre completo	VARCHAR	No	No
fecha_registro	Fecha registro en la aplicación web	VARCHAR	No	No
first_name	Nombre	VARCHAR	No	No
guid	Identificador único global	VARCHAR	No	No
idmend	Identificador de mendeley	VARCHAR	No	No
is_online	Estado de conexión en la aplicación	BIT	No	No
last_name	Apellidos	VARCHAR	No	No
link	Link perfil Mendeley	VARCHAR	No	No
url_photo	Imagen perfil Mendeley	VARCHAR	No	No

Tabla 5.4: Tabla user_profile

criterion: criterios de clasificación de las referencias bibliográficas.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
description	Descripción	LONGTEXT	No	No
name	Nombre	VARCHAR	No	No
submit_date	Fecha creación	DATETIME	No	No
slr_id	Identificador de la tabla slr.	BIGINT	No	No

Tabla 5.5: Tabla criterion

specific_attribute: atributos específicos de las referencias bibliográficas de un SLR.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
name	Nombre	VARCHAR	No	No
submit_date	Fecha creación	DATETIME	No	No
slr_id	Identificador de la tabla slr.	BIGINT	No	No
tipo	Tipo de atributo.	VARCHAR	No	No

Tabla 5.6: Tabla specific attribute

slr: tabla de revisiones sistemáticas de la literatura.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
guid	Identificador único global	VARCHAR	Si	No
idmend	Identificador de la carpeta de Mendeley	VARCHAR	Si	No
justification	Justificación de un SLR	LONGTEXT	No	No
last_modified	Última fecha de modificación.	VARCHAR	No	No
num_visits	Número visitas	INT	No	No
state	Fase de la revisión en la que se encuentra	VARCHAR	No	No
submit_date	Fecha creación	DATETIME	No	No
title	Título	VARCHAR	No	No
total_references	Total referencias	INT	No	No
user_profile_id	Identificador de la tabla user_profile	BIT	No	No

Tabla 5.7: Tabla slr

search: tabla de búsquedas de las revisiones sistemáticas de la literatura.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
end_year	Año máximo	VARCHAR	No	No
fecha	Fecha creación	DATETIME	No	No
guid	Identificador único global	VARCHAR	No	No
max_total	Número máximo de referencias a encontrar.	INT	No	No
slr_id	Identificador de la tabla SLR	INT	No	No
start_year	Año mínimo	VARCHAR	No	No

Tabla 5.8: Tabla search

search_term_param: tabla de términos de búsquedas.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
component_id	Identificador de la tabla componente	BIGINT	No	No
operator_id	Identificador de la tabla operator	BIGINT	No	No
search_id	Identificador de la tabla búsqueda	BIGINT	No	No
terminos	Términos de búsqueda.	VARCHAR	No	No

Tabla 5.9: Tabla search_term_param

search_operator: tabla de operadores en una búsqueda.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
name	Nombre	VARCHAR	No	No
value	Valor	VARCHAR	No	No

Tabla 5.10: Tabla search_operator

5.4. Diseño detallado de Componentes

Para cada uno de los módulos funcionales del sistema debemos realizar un diagrama de secuencia, para definir la interacción existente entre las clases de objetos que permitan responder a eventos externos.

search_component: tabla de los componentes en una búsqueda.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
name	Nombre	VARCHAR	No	No
value	Valor	VARCHAR	No	No

Tabla 5.11: Tabla search_component

engine_search: tabla de los motores de búsquedas para las referencias.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
api_key	Api Key	VARCHAR	No	No
display_name	Nombre completo	VARCHAR	No	No
image	Imagen logo	VARCHAR	No	No
name	Nombre	VARCHAR	No	No
status	Estado	BIT	No	No
url	Link página búsqueda	VARCHAR	No	No

Tabla 5.12: Tabla engine

reference: tabla de las referencias bibliográficas.				
Atributo	Descripción	Tipo dato	Nulo	Clave
id	Identificador	BIGINT	No	Si
citation_key	Citation Key	VARCHAR	No	No
bibtex	Bibtex	LONGTEXT	No	No
created	Fecha creación	DATETIME	No	No
criterion_id	Identificador de la tabla criterio	BIGINT	No	No
doc_abstract	Abstract	LONGTEXT	No	No
doi	Identificador DOI	VARCHAR	No	No
engine_id	Identificador de la tabla engine.	BIGINT	No	No
idmend	Identificador en Mendeley	VARCHAR	No	No
isbn	ISBN	VARCHAR	No	No
language_id	Identificador de la tabla Language	BIGINT	No	No
title	Título referencia	LONGTEXT	No	No
type_document_id	Identificador de la tabla type_document	BIGINT	No	No

Tabla 5.13: Tabla reference

5.5. Diseño detallado de la Interfaz de Usuario

Para el desarrollo de las interfaces se ha empleado *Bootstrap*[[Bootstrap, 2016](#)]. Este framework dispone de un conjunto de plantillas con HTML, CSS y JS que permite el fácil desarrollo de cada una de las vistas. En las figuras 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17, 5.18, 5.19 podemos ver algunas de las interfaces que un usuario puede ver en el sistema.

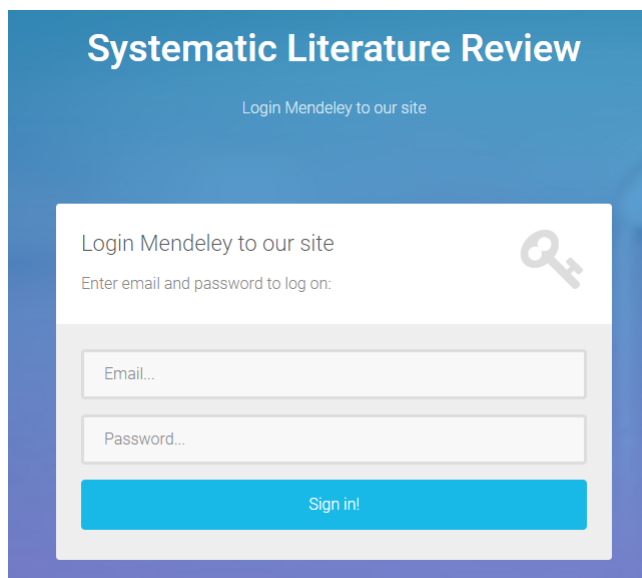


Figura 5.7: Interfaz Login Sistema

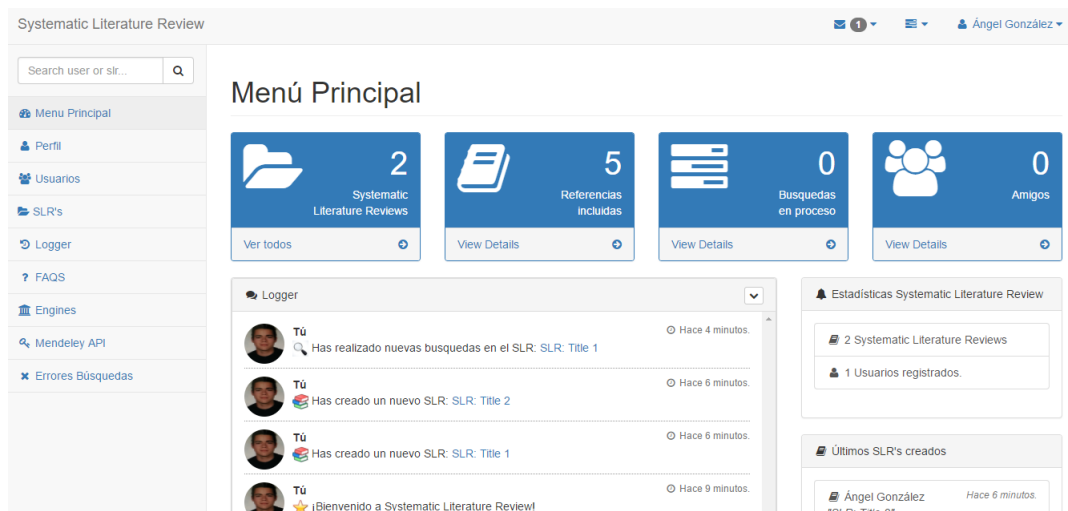


Figura 5.8: Interfaz Menu principal.

Systematic Literature Review

Search user or slr...

Menu Principal

Perfil

Usuarios

SLR's

Logger

FAQS

Engines

Mendeley API

Errores Búsquedas

Mis SLR's

Home / Mis SLR's

Crear SLR Sincronizar (Mendeley) Información

Show 10 entries Search:

Titulo	Fecha creación	Nº Búsquedas	Nº Referencias	Acciones
SLR: Title 1	28/may/2017	1 Búsquedas	5 Referencias	? + - x
SLR: Title 2	28/may/2017	0 Búsquedas	0 Referencias	? + - x

Showing 1 to 2 of 2 entries

Previous 1 Next

Figura 5.9: Interfaz Lista SLR.

Systematic Literature Review

Search user or slr...

Menu Principal

Perfil

Usuarios

SLR's

Logger

FAQS

Engines

Mendeley API

Errores Búsquedas

Criterios en SLR: Title 1

Home / Mis SLR's / SLR: Title 1 / Criterios

Crear Criterio

Show 10 entries Search:

Nombre	Fecha creación:	Última modificación:	Referencias asignadas:	Acciones
included	28/may/2017 16:23	28/may/2017 16:23	5	x
language	28/may/2017 16:39	28/may/2017 16:39	0	x
version	28/may/2017 16:39	28/may/2017 16:39	0	x

Showing 1 to 3 of 3 entries

Previous 1 Next

Figura 5.10: Interfaz Lista Criterios.

Systematic Literature Review

Search user or slr...

Menu Principal

Perfil

Usuarios

SLR's

Logger

FAQS

Engines

Mendeley API

Errores Búsquedas

Atrib. Especificos en SLR: Title 1

Home / Mis SLR's / SLR: Title 1 / Atributos Especificos

Crear Atributo

Show 10 entries Search:

Nombre	Tipo:	Fecha creación:	Última modificación:	Opciones	Acciones
version	string	16:40 - 28/may/2017	16:40 - 28/may/2017	--	x

Showing 1 to 1 of 1 entries

Previous 1 Next

Figura 5.11: Interfaz Lista Atributos específicos

Systematic Literature Review

Search user or slr...

Q

Menu Principal

Perfil

Usuarios

SLR's

Logger

FAQS

Engines

Mendeley API

Errores Búsquedas

Crear búsqueda SLR: Title 1

Engine:

☐
☒
☐
☒
☐

Total Máximo:

5

Total Máximo:

1980199120092017

Insertar más términos

Buscar

Abstract

ALL

estos términos:

spem

Buscar

Author

ANY

estos términos:

smith

Buscar

Title

NONE

estos términos:

sppem

Crear Búsqueda

Figura 5.12: Interfaz Crear búsqueda

Systematic Literature Review

Search user or slr...

Q

Menu Principal

Perfil

Usuarios

SLR's

Logger

FAQS

Engines

Mendeley API

Errores Búsquedas

TASKS SEARCH

Inicio / Task Searches

SLR: Title 1

Estado: Realizando búsqueda en los engines...

Comienzo: 28/may/2017 16:24

25%

Figura 5.13: Interfaz progreso de búsquedas

Systematic Literature Review

Search user or slr...

Q

Menu Principal

Perfil

Usuarios

SLR's

Logger

FAQS

Engines

Mendeley API

Errores Búsquedas

Búsquedas en SLR: Title 1

Inicio / Mis SLR's / SLR: Title 1 / Búsquedas

Crear búsqueda

Show 10 entries

Search:

Realizada en:	Año Comienzo	Año finalización	Total Máximo	Encontrados:	Engines	Términos
28/may/2017 16:25:28	1980	2017	5	5 Referencias	• IEEE	• ALL "spem" en Abstract

Showing 1 to 1 of 1 entries

Previous

1

Next

Figura 5.14: Interfaz Lista búsquedas

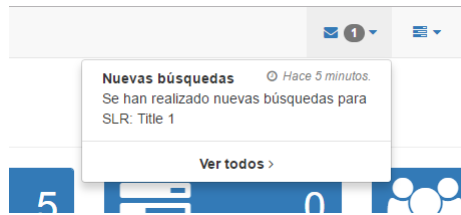


Figura 5.15: Interfaz notificación búsqueda

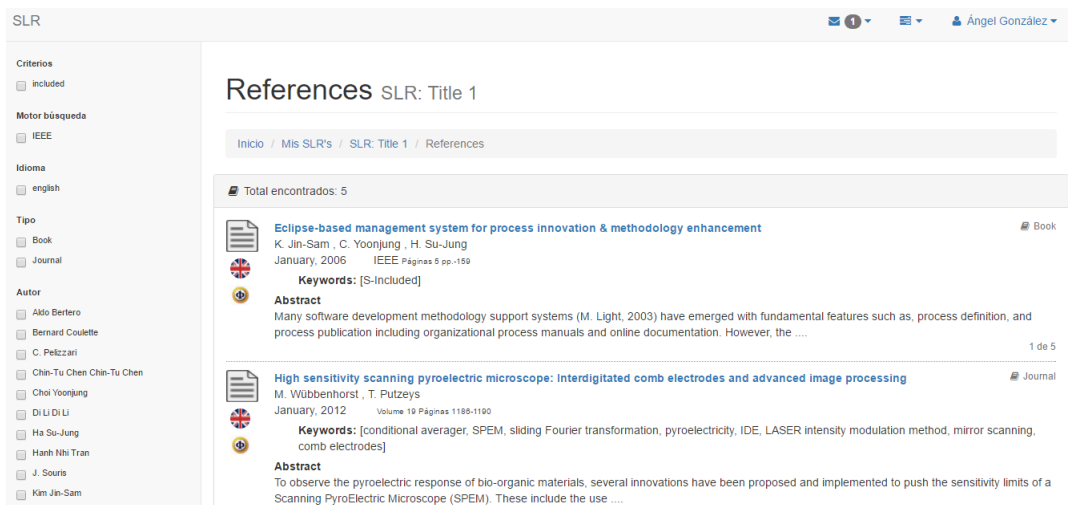


Figura 5.16: Interfaz Lista Referencias

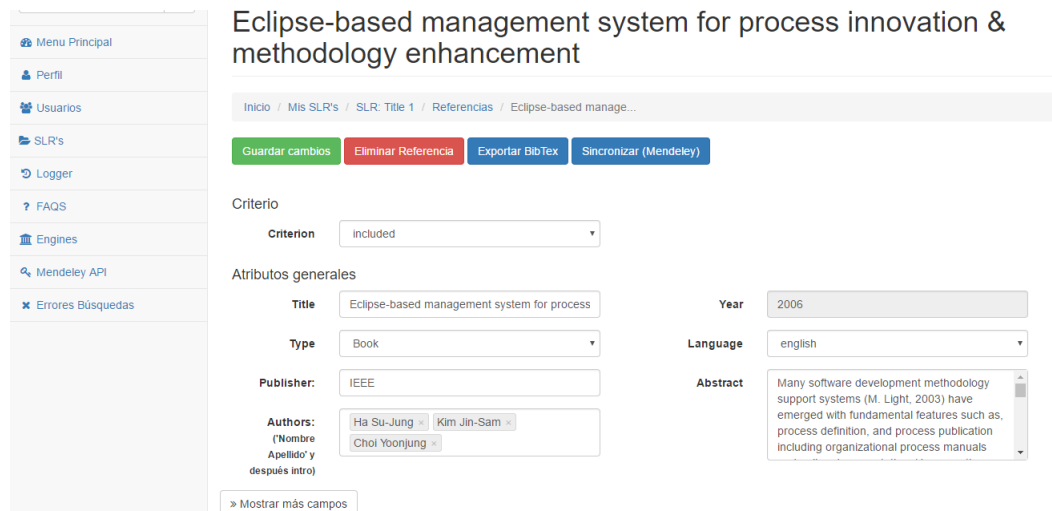


Figura 5.17: Interfaz Referencia

Gráficos SLR: Title 1

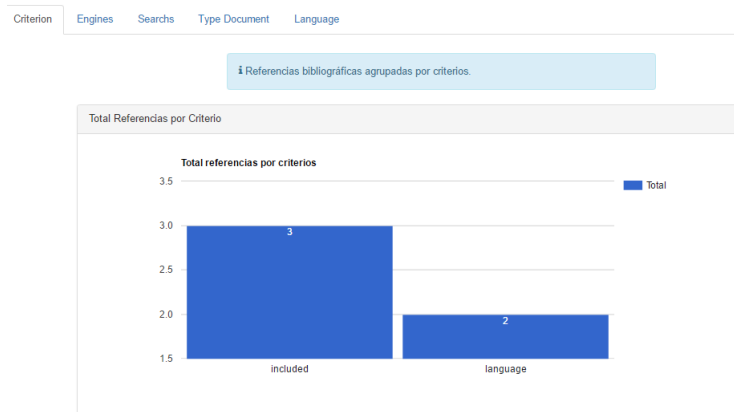


Figura 5.18: Interfaz gráficos SLR (I)

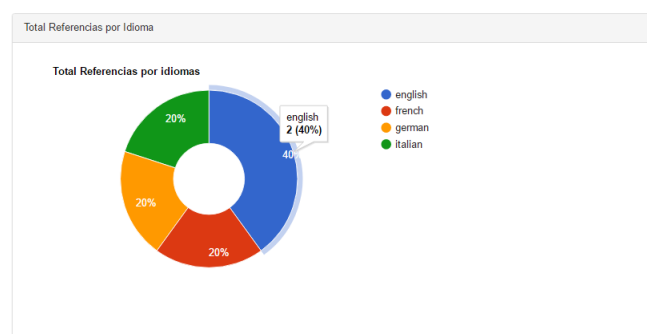


Figura 5.19: Interfaz gráficos SLR (II)

Capítulo 6

Construcción del Sistema

Este capítulo trata sobre todos los aspectos relacionados con la implementación del sistema en código, haciendo uso de un determinado entorno tecnológico.

6.1. Entorno de Construcción

El framework empleado en la creación de esta página web es *Grails*, un framework para aplicaciones web libre desarrollado sobre el lenguaje de programación *Groovy* (el cual a su vez se basa en *Java*). Groovy posee una sintaxis muy parecida a *Java*, comparte el mismo modelo de objetos, de hilos y de seguridad, además, puede acceder directamente a todas las API existentes en *Java*. Esta relación entre ambos lenguajes, ha permitido que algunas partes del desarrollo se haya empleado tanto *Java* como *Groovy*.

Para emplear estos lenguajes en el desarrollo, se emplea el IDE *Grails Tool Suite*, un entorno de trabajo donde podemos aplicar el paradigma **Modelo Vista Controlador** (MVC).

Grails Tool Suite es un IDE altamente productivo que sigue paradigmas como convención sobre configuración o no te repitas (DRY) proporcionando un entorno de desarrollo estandarizado y ocultando gran parte de los detalles de configuración al programador.

Una de las ventajas que proporciona este tipo de proyectos, es que podemos instalar dependencias y *plugins* que nos permitan instalar utilidades a nuestro proyecto Grails para ser empleadas directamente sobre el código y aplicar el paradigma DRY que se ha explicado anteriormente. Algunos de ellos son:

- *tomcat* Para instalar el contenedor de Java Tomcat.
- *jquery* Para instalar JQuery en el proyecto.
- *spring-security-core* Para realizar la securización de la aplicación web.
- *executor* Para realizar tareas en segundo plano.
- *apache.poi* Para la creación de documentos word, excel o pdf.
- *mysql-connector-java* Para permitir la conexión de la aplicación con MySQL.

Todo el código de la aplicación se encuentra ubicado en un espacio de trabajo de Assembla cuya dirección es <https://app.assembla.com/spaces/systematic-literature-review/> que puede ser descargado en un equipo por Subversion (SVN). En este espacio de trabajo se ha insertado además todo el contenido de la memoria así como recursos necesarios que puedan ser de utilidad para este trabajo.

6.2. Código Fuente

En la figura 6.1 podemos ver la estructura de directorios que tiene este proyecto Grails.

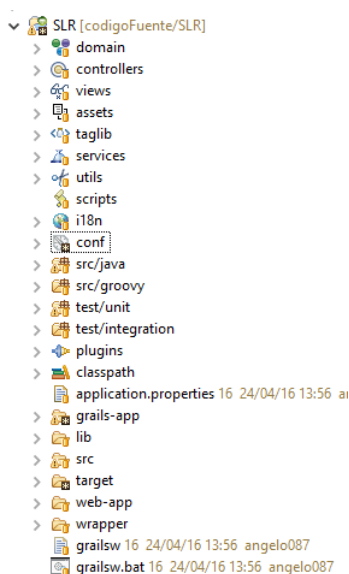


Figura 6.1: Estructura Proyecto Grails

- **conf** Archivos de configuracion
- **controllers** Controladores
- **domain** Entidades
- **i18n** messages bundles
- **services** Servicios
- **taglib** Libreria de etiquetas
- **util** Clases de utilidad
- **views** Vistas
- **layouts** Layouts SiteMesh
- **lib**

- **scripts**
- **src**
 - **groovy** Otras clases Groovy
 - **java** Otras clases Java
- **test** Casos de prueba
- **web-app** Raíz de la aplicación web

Para almacenar todo el código del proyecto en el repositorio assembla, se ha instalado un plugin para el IDE Grails Tool Suite denominado *Subclipse* con el que rápidamente podemos subir todas las modificaciones y ver el histórico de cambios con respecto a otras versiones del código.

Como ejemplo de código se va a describir la funcionalidad de crear una búsqueda en una revisión sistemática de la literatura. Primeramente, debemos tener una clase que representa la entidad Search dentro de la carpeta *domain* a la que hemos denominado *Search.groovy* tal y como podemos ver en la 6.2. En ella, podemos destacar la relación con otras clases y los atributos de la clase como una clase de Java.

```
package es.uca.pfc

import java.util.Date;

class Search {

    static belongsTo = [slr: Slr]
    static hasMany = [references: Reference, engines: EngineSearch, termParams: SearchTermParam]

    Date fecha = new Date()
    String startYear = ""
    String endYear = ""
    int maxTotal = 5
    String guid = UUID.randomUUID().toString();

    static constraints = {
    }

    String toString()
    {
        return "${terminos}"
    }
}
```

Figura 6.2: Dominio clase Search

Cuando el usuario desee crear una búsqueda, elegirá la acción de crear una búsqueda a través de un botón. El controlador será el encargado de recoger esta acción propuesta por el usuario, tratará la información correspondiente y redirigirá a la vista correspondiente. Este controlador se encuentra dentro de la carpeta *controller*. La clase resultante es *SearchController.groovy* tal y como podemos ver en la figura 6.3. En el controlador se recoge todas las acciones que el usuario desee realizar con respecto a una búsqueda. En nuestro caso, comprobará que la búsqueda tenga una revisión sistemática asignada o cargar la lista de motores de búsquedas que el usuario puede emplear entre otras tareas.

```

class SearchController {

    static allowedMethods = [save: "POST", update: "PUT", delete: "DELETE"]

    def springSecurityService
    def toolService
    def mendeleyToolService

    def index()
    {
        redirect(controller: 'index', action: 'index')
    }

    def create()
    {
        def isLogin = springSecurityService.loggedIn

        String guidSlr = (null == params.guidSlr ? "" : params.guidSlr.toString())

        if(!isLogin || guidSlr.equals(""))
        {
            redirect(controller: 'index', action: 'index')
        }
        else
        {
            def slrInstance = Slr.findByGuid(guidSlr)
            def error = (null == params.error ? "" : params.error.toString())

            if (null == slrInstance)

```

Figura 6.3: Controlador clase Search

Una vez que el controlador acaba sus tareas y comprobaciones, redirigirá la acción a una vista donde mostraremos el formulario con el que el usuario podrá ingresar los parámetros de búsqueda que estime necesario. Grails posee una carpeta denominada *view* donde se encuentran organizadas todas las vistas por controladores y acciones. En nuestro caso, habrá una vista denominada *create.gsp* dentro de la carpeta *search*, que a su vez estará dentro de la carpeta *view*. Esta vista (ver figura 6.4) posee todo el contenido HTML y etiquetas de grails que muestra el formulario que permite al usuario crear una búsqueda dentro de la aplicación web.

Posteriormente, el usuario rellena el formulario y elegirá la opción de realizar la búsqueda. Por tanto, volvemos al mismo mecanismo que hemos explicado anteriormente pero con diferentes acciones. En este caso, volvemos al controlador de la búsqueda donde tendrá la acción *save* y ésta se encarga de comprobar que los parámetros introducidos son correctos. En caso de ser así, redigirá la acción a la lista de búsquedas ya realizadas notificando con un mensaje que la búsqueda está en proceso. En caso contrario, redigirá la salida a la pantalla de creación de búsquedas con un mensaje de error.

Por último, dentro de cualquier controlador, podremos implementar unas clases o servicios que se encargan de la implementación de algunas de las funcionalidades. Ya en 6.3 pudimos ver que hay un servicio definido denominado *springSecurityService*, el cuál, se encarga de la gestión de usuarios. Este servicio es creado gracias a un plugin que se ha instalado, pero podemos crear nuestros propios servicios. En nuestro caso, se ha decidido crear un servicio que realice las búsquedas para la conexión con Mendeley y realice las búsquedas en segundo plano al que hemos determinado *mendeleyToolService* y que también podemos ver definido en 6.3. La implementación de este servicio se encuentra dentro de la carpeta *service* bajo el nombre *MendeleyToolService.groovy* tal y como podemos ver en la figura ?? y se realizaría bajo el método *insertSearchsBackground* como una función típica de Java.

```

<div id="wrapper">

  <%-- Head --%>
  <g:render template="head" contextPath="/" />

  <div id="page-wrapper">
    <div class="row">
      <div class="col-lg-12">
        <h1 class="page-header">Crear búsqueda <small>${slrInstance.title}</small></h1>
      </div>
      <!-- /.col-lg-12 -->
    </div>

    <div class="row">
      <div id="divError" class="alert alert-danger" role="alert" style="display: none;"><i clas
    </div>

    <g:form class="form-horizontal" controller="search" action="save" method="POST" name="myForm"

      <g:hiddenField name="guidSlr" value="${slrInstance.guid}" />

      <%-- Engines --%>
      <div class="row">
        <div class="col-lg-12" style="margin-bottom: 20px;">
          <div class="form-group">
            <label class="col-sm-2 control-label" style="margin-right: 15px;">Engine:</la
            <g:each in="${engineListInstance}" var="engineInstance">
              <label class="checkbox-inline">
                <g:if test="${engineInstance.name.toLowerCase().equals("acm")} && opAC

```

Figura 6.4: Vista clase Search

```

package es.uca.pfc

import java.lang.annotation.Documented;

@Transactional
class MendeleyToolService {

  def springSecurityService
  def toolService

  String encodePasswordMendeley(String password)
  {
    return EncodeDecodeMendeley.encodePasswordMendeley(password);
  }

  String decodePasswordMendeley(String password)
  {
    return EncodeDecodeMendeley.decodePasswordMendeley(password);
  }

  boolean insertSearchsBackground(String guidTaskSearch, String nameSlr, String guidSlr, List<String> 1
  List<SearchComponent> components, String minYear, String maxYear, String maxTotal, Map<String, Bk
  {
    println "insertSearchsBackground"

    increaseProgressBar(0, guidTaskSearch, null, null)
    boolean isSuccess = true
    def userInstance = User.get(springSecurityService.currentUser.id)
    String emailMend = userInstance.userMendeley.email_mend
    String passMend = decodePasswordMendeley(userInstance.userMendeley.pass_mend)

```

Figura 6.5: Vista clase Search

6.3. Scripts de Base de datos

Organización del código fuente, describiendo la utilidad de los diferentes ficheros y su distribución en paquetes o directorios. Asimismo, se incluirá el script de algún disparador o un procedimiento almacenado, que sea de interés para ilustrar algún aspecto concreto de la gestión de la base de datos.

Capítulo 7

Pruebas del Sistema

En este capítulo se presenta el plan de pruebas del sistema de información, incluyendo los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales (mediante listas de comprobación) o automatizadas mediante algún software específico de pruebas.

7.1. Estrategia

En esta sección se debe incluir el alcance de las pruebas, hasta donde se pretende llegar con ellas, si se registrarán todas o sólo aquellas de un cierto tipo y cómo se interpretarán y evaluarán los resultados. También, se incluirá el procedimiento a seguir para las pruebas de regresión, esto es, la repetición de ciertas pruebas para comprobar que nuevos cambios que se vayan introduciendo no originen errores en el software ya probado.

7.2. Entorno de Pruebas

Incluir en este apartado los requisitos de los entornos hardware/software donde se ejecutarán las pruebas.

7.3. Roles

Describir en esa sección cuáles serán los perfiles y participantes necesarios para la ejecución de cada uno de los niveles de prueba.

7.4. Niveles de Pruebas

En esta sección se documentan los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales o automatizadas mediante algún software específico de pruebas.

7.4.1. Pruebas Unitarias

Las pruebas unitarias tienen por objetivo localizar errores en cada nuevo artefacto software desarrollado, antes que se produzca la integración con el resto de artefactos del sistema.

7.4.2. Pruebas de Integración

Este tipo de pruebas tienen por objetivo localizar errores en módulos o subsistemas completos, analizando la interacción entre varios artefactos software.

7.4.3. Pruebas de Sistema

En esta actividad se realizan las pruebas de sistema de modo que se asegure que el sistema cumple con todos los requisitos establecidos: funcionales, de almacenamiento, reglas de negocio y no funcionales. Se suelen desarrollar en un entorno específico para pruebas.

Pruebas Funcionales

Con estas pruebas se analiza el buen funcionamiento de la implementación de los flujos normales y alternativos de los distintos casos de uso del sistema.

Pruebas No Funcionales

Estas pruebas pretenden comprobar el funcionamiento del sistema, con respecto a los requisitos no funcionales identificados: eficiencia, seguridad, etc.

7.4.4. Pruebas de Aceptación

El objetivo de estas pruebas es demostrar que el producto está listo para el paso a producción. Suelen ser las mismas pruebas que se realizaron anteriormente pero en el entorno de producción. En estas pruebas, es importante la participación del cliente final.

Parte III

Epílogo

En esta última parte quedarán recogidas las conclusiones y los manuales necesarios para el manejo de la aplicación resultado del desarrollo. Si se ha realizado algún tipo de evaluación de la solución proporcionada, más allá de las pruebas del sistema, también deberá venir recogida en un capítulo separado dentro de esta parte. Pueden consultarse diversos tipos de evaluaciones sobre sistemas de información en [\[Hevner et al., 2004\]](#): casos de estudio, análisis estático, análisis dinámico, simulación, experimento controlado, etc.

Capítulo 8

Manual de implantación y explotación

Las instrucciones de instalación y explotación del sistema se detallan a continuación. Este manual resulta de aplicación en aquellos casos en que se requiere realizar la instalación del software objetivo de este proyecto sobre algún entorno de servidor o cuando su instalación no sea trivial.

8.1. Introducción

Resumen de los principales objetivos, ámbito, características y alcance del software desarrollado.

8.2. Requisitos previos

Requisitos hardware y software para la correcta instalación del sistema.

8.3. Inventario de componentes

Lista de los componentes hardware y software que se incluyen en la versión del producto.

8.4. Procedimientos de instalación

Procedimientos de instalación y configuración de cada componente hardware y software (base y desarrollado) para asegurar la correcta instalación y explotación del sistema, así como aquellos procedimientos necesarios de migración/carga de datos.

8.5. Pruebas de implantación

Descripción de las pruebas a realizar después de la instalación del sistema.

8.6. Procedimientos de operación y nivel de servicio

Procedimientos necesarios para asegurar el correcto funcionamiento, rendimiento, disponibilidad y seguridad del sistema: back-ups, chequeo de logs, etc. También, es preciso indicar claramente aquellas actuaciones precisas necesarias para el mantenimiento preventivo del sistema y así prevenir posibles fallos en el mismo.

Capítulo 9

Manual de usuario

Las instrucciones de uso del software se detallan a continuación. Este manual se dirige al usuario final del software objetivo de este proyecto.

9.1. Introducción

Resumen de los principales objetivos, ámbito, características y alcance del software desarrollado.

9.2. Instalación

Se detallarán los pasos necesarios para la obtención e instalación del software, así como los requisitos previos de hardware y software.

9.3. Uso del sistema

Describir todos los aspectos necesarios para una utilización efectiva y eficiente de las características del sistema por parte de los usuarios.

Capítulo 10

Manual del desarrollador

A continuación se recogen las instrucciones necesarias para evolucionar el software. Este manual está dirigido a los desarrolladores que pretenden extender o modificar el código fuente, con el fin de incorporar nuevas funcionalidades o modificar las ya existentes. A lo largo de este capítulo se deberán hacer referencias explícitas a aquellos epígrafes de los capítulos de Diseño, Construcción y Pruebas del Sistema que resulten de interés.

10.1. Introducción

Resumen de los principales objetivos, ámbito, características y alcance del software desarrollado.

10.2. Preparación del entorno de trabajo

Descripción de los requisitos (hardware y software) previos. Datos de interés relativos al control de versiones del software. Detalles sobre la instalación en local del entorno de desarrollo y, si fuesen necesarios, de otros componentes como bases de datos, servidores de aplicaciones, etc.

10.3. Consideraciones generales sobre el desarrollo

Aspectos importantes a tener en cuenta a la hora de modificar y extender el código fuente, guías de estilo, etc. Asimismo, se detallarán las directrices que sean de aplicación a la hora de realizar pruebas sobre las nuevas mejoras introducidas.

10.4. Instrucciones para construcción y despliegue

Secuencia de pasos requeridos para llevar a cabo la compilación del código fuente y así poder construir y depurar el software sobre una máquina de desarrollo.

Capítulo 11

Conclusiones

En este último capítulo se detallan las lecciones aprendidas tras el desarrollo del presente proyecto y se identifican las posibles oportunidades de mejora sobre el software desarrollado.

11.1. Objetivos alcanzados

Este apartado debe resumir los objetivos generales y específicos alcanzados, relacionándolos con todo lo descrito en el capítulo de introducción.

11.2. Lecciones aprendidas

A continuación, se detallan las buenas prácticas adquiridas, tanto tecnológicas como procedimentales, así como cualquier otro aspecto de interés.

Resumir cuantitativamente el tiempo y esfuerzo dedicados al proyecto a lo largo de su desarrollo que escribir un sencillo 'he trabajado mucho en este proyecto'.

11.3. Trabajo futuro

En esta sección, se presentan las diversas áreas u oportunidades de mejora detectadas durante el desarrollo del proyecto y que podrán ser abarcadas en futuras versiones del software.

Los elementos aquí descritos deben estar en relación con lo relatado en el apartado de objetivos y alcance del proyecto descritos en la introducción.

Bibliografía

- [Bok, 2016] Bok, S. M. (2016). Scrum Manager - Guía de formación. http://www.scrummanager.net/bok/index.php?title=Scrum_Manager_BoK.
- [Bootstrap, 2016] Bootstrap (2016). Bootstrap Framework. <http://getbootstrap.com/>.
- [Brito, 2009] Brito, N. (2009). Manual de desarrollo web con GRAILS. <https://es.scribd.com/doc/189709106/Manual-Grails-pdf>.
- [CCOO, 2010] CCOO (2010). Tablas salariales 2010 IV Convenio Colectivo. <http://www.uca.es/sindicato/ccoo/documentos/tabla-salarial-pas-laboral-2010.pdf>.
- [Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1):75–105.
- [Kitchenham, 2007] Kitchenham, B. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf.
- [myBalsamiq,] myBalsamiq. Balsamiq - Universidad de Cádiz. <https://www.mybalsamiq.com/login>.
- [Project, 2015] Project, G. (2015). Gantt Project: free desktop project management app. <http://www.ganttproject.biz/>.

Información sobre Licencia

Incluir aquí la información relativa a la licencia seleccionada para la documentación y software del presente proyecto.