



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



TFG del Grado en Ingeniería Informática
**Integración de los datos del
CENIEH en *ARIADNEplus***



Presentado por Gonzalo Cuesta Marín
en Universidad de Burgos — 19 de julio de 2020
Tutor: Dr. Carlos López Nozal
y D. Mario Juez Gil

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	V
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	6
Apéndice B Especificación de Requisitos	11
B.1. Introducción	11
B.2. Objetivos generales	11
B.3. Catálogo de requisitos	12
B.4. Especificación de requisitos	17
Apéndice C Especificación de diseño	73
C.1. Introducción	73
C.2. Diseño de datos	73
C.3. Diseño arquitectónico	75
C.4. Diseño procedimental	87
C.5. Diseño de interfaces	88
Apéndice D Documentación técnica de programación	93
D.1. Introducción	93
D.2. Estructura de directorios	93
D.3. Manual de programador	94
D.4. Pruebas del sistema	109

ÍNDICE GENERAL

II

Apéndice E Documentación de usuario	111
E.1. Introducción	111
E.2. Requisitos	111
E.3. Instalación	112
E.4. Manual del usuario	119
Bibliografía	120

Índice de figuras

A.1. Gráfico <i>Burndown</i> del Sprint 2	3
A.2. Gráfico <i>Burndown</i> del Sprint 3	4
A.3. Gráfico <i>Burndown</i> del Sprint 4	4
A.4. Gráfico <i>Burndown</i> del Sprint 5	5
A.5. Gráfico <i>Burndown</i> del Sprint 6	5
 B.1. Diagrama de casos de la aplicación completa.	18
 C.1. Diagrama E/R de datos relacionados con la gestión de datos.	74
C.2. Diagrama que muestra la relación entre el Modelo, Vista y Controlador del patrón MVC	76
C.3. Diagrama que muestra el comportamiento de la aplicación ante una petición HTTP.	77
C.4. Diagrama de paquetes de la aplicación.	78
C.5. Diagrama de paquetes del complemento ficticio <i>FuncExtra</i>	79
C.6. Paquetes de los complementos instalados en la aplicación.	80
C.7. Paquete tipo del complemento ficticio <i>FuncExtra</i>	82
C.8. Diagrama de clases del complemento ficticio <i>FuncExtra</i>	83
C.9. Diagrama de clases del complemento <i>AdminMenuDesign</i>	84
C.10. Diagrama de clases del complemento <i>AriadnePlus Tracking</i>	84
C.11. Diagrama de clases del complemento <i>CENIEHExport</i>	85
C.12. Diagrama de clases del complemento <i>CollectionFiles</i>	86
C.13. Diagrama de clases del complemento <i>AutoDublinCore</i>	87
C.14. Diagrama de clases del complemento <i>TagsManager</i>	87
C.15. Diagrama de secuencia para un caso general.	87
C.16. Prototipos: página principal (<i>ARIADNEplus Tracking</i>)	88
C.17. Prototipos: creación de un ticket (<i>ARIADNEplus Tracking</i>)	89
C.18. Prototipos: primera y segunda fase de un ticket (<i>ARIADNEplus Tracking</i>)	90

C.19. Prototipos: tercera y cuarta fase de un ticket (<i>ARIADNEplus Tracking</i>)	91
C.20. Prototipos: quinta y sexta fase de un ticket (<i>ARIADNEplus Tracking</i>)	92
D.1. Vista del panel de administración de Google Cloud tras finalizar los pasos marcados.	103
D.2. Ejecución del workflow.	106
D.3. Panel principal de la plataforma Codacy.	107
D.4. Página principal de la documentación del proyecto hospedada en Read The Docs.	108
D.5. Resultados de la ejecución de las pruebas unitarias para cada uno de los complementos.	110
E.1. Página principal del manual de usuario.	119

Índice de tablas

A.1. Relación entre esfuerzo (<i>story point</i>) y tiempo	2
A.2. Contrato prácticas curriculares	6
A.3. Contrato prácticas extracurriculares	7
A.4. Contrato estándar a media jornada de un desarrollador profesional.	7
A.5. Costes de hardware	7
A.6. Otros costes	8
A.7. Costes totales	8
A.8. Licencias de las dependencias utilizadas	9
B.1. CU-01 Buscar ítems públicos.	19
B.2. CU-02 Listar ítems públicos.	20
B.3. CU-03 Listar colecciones públicas.	21
B.4. CU-04 Listar etiquetas públicas.	21
B.5. CU-05 Ver ítem público.	22
B.6. CU-06 Ver colección pública.	22
B.7. CU-07 Ver fichero público.	23
B.8. CU-08 Buscar ítem público en mapa.	24
B.9. CU-09 Información del proyecto.	24
B.10. CU-10 Acceso al área de administración.	25
B.11. CU-11 Gestión de ítems.	26
B.12. CU-11 Añadir ítem.	27
B.13. CU-13 Editar ítem.	28
B.14. CU-14 Eliminar ítem.	29
B.15. CU-15: Listar ítems.	30
B.16. CU-16 Ver ítem.	31
B.17. CU-17 Exportar ítem.	32
B.18. CU-18 Buscar ítems.	32
B.19. CU-19 Exportar ítems.	33
B.20. CU-20 Gestión de colecciones.	33
B.21. CU-21 Añadir colección.	34

B.22. CU-22: Editar colección.	35
B.23. CU-23 Eliminar colección.	36
B.24. CU-23 Listar colecciones.	37
B.25. CU-25 Ver colección.	38
B.26. CU-26 Exportar colección.	38
B.27. CU-27 Gestión de tipos de ítem.	39
B.28. CU-28 Añadir tipo de ítem.	39
B.29. CU-29 Editar tipo de ítem.	40
B.30. CU-30 Eliminar tipo de ítem.	41
B.31. CU-31 Listar tipos de ítem.	41
B.32. CU-32 Ver tipo de ítem.	42
B.33. CU-33 Gestión de etiquetas.	43
B.34. CU-34 Editar etiqueta.	44
B.35. CU-35 Eliminar etiqueta.	44
B.36. CU-36 Listar etiquetas.	45
B.37. CU-37 Eliminar etiquetas.	46
B.38. CU-38 Importación de metadatos CSV.	47
B.39. CU-39 Ejecutar importación CSV.	47
B.40. CU-40 Listar importaciones CSV.	48
B.41. CU-41 Deshacer importación CSV.	49
B.42. CU-42 Recolección de metadatos (OAI-PMH).	50
B.43. CU-43 Ejecutar recolección.	51
B.44. CU-44 Actualizar recolección.	51
B.45. CU-45 Listar recolecciones.	52
B.46. CU-46 Ver recolecciones.	53
B.47. CU-47 Deshacer recolección.	54
B.48. CU-48 Seguimiento <i>ARIADNEplus</i> .	55
B.49. CU-49 Crear ticket de seguimiento.	56
B.50. CU-50 Eliminar ticket de seguimiento.	56
B.51. CU-51 Administrar ticket de seguimiento.	57
B.52. CU-52 Cambiar de fase.	58
B.53. CU-53 Edición masiva de metadatos.	60
B.54. CU-54 Gestión de complementos.	60
B.55. CU-55 Instalar complemento.	61
B.56. CU-56 Configurar complemento.	62
B.57. CU-57 Desinstalar complemento.	63
B.58. CU-58 Configuración de la apariencia.	64
B.59. CU-59 Usar plantilla.	65
B.60. CU-60 Configurar plantilla.	66
B.61. CU-61 Configurar otros aspectos de apariencia.	66
B.62. CU-62 Configuración de la aplicación.	67
B.63. CU-63 Gestión de usuarios.	68
B.64. CU-64 Añadir usuario.	69
B.65. CU-65 Editar usuario.	70

Índice de tablas

VII

B.66.CU-66 Eliminar usuario.	71
B.67.CU-67 Listar usuarios.	72

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La finalidad de este apartado es definir el conjunto de actividades que se han llevado a cabo para gestionar la evolución del proyecto. Estas actividades se pueden dividir en dos fases: planificación temporal y estudio de viabilidad.

La fase de planificación temporal agrupa todas aquellas actividades que tienen como objetivo organizar, planificar y desarrollar una serie de tareas dentro de un marco temporal. Todas ellas definen el intervalo de tiempo en el que se va a desarrollar el proyecto. Existen multitud de estrategias para determinar cada una de estas actividades, en este caso, se ha optado por emplear metodologías ágiles.

En la segunda fase, se reúnen todas y cada una de las actividades que estudian y analizan la viabilidad económica y legal del proyecto.

A.2. Planificación temporal

Para llevar a cabo la planificación temporal del proyecto se ha optado por utilizar metodologías ágiles, concretamente *Scrum* [16].

El desarrollo se ha dividido en iteraciones (*sprints*), cada una de las cuales ha tenido una duración aproximada de 2 semanas. Cada iteración se compone de una serie de tareas cuyo principal cometido es incluir nuevas funcionalidades a la infraestructura.

Las tareas de cada *sprint* son elaboradas en la reunión que se establece al terminar la iteración anterior. Además de planificar las tareas de la siguiente

iteración, se supervisan las del anterior. En nuestro caso, las reuniones se hacían tanto con la UBU [18] como con el CENIEH [2].

Dado que el desarrollo se ha llevado a cabo en la plataforma *Github* [7], es importante aclarar que el término *sprint* comentado anteriormente se ve reflejado en esta plataforma como *milestone* y, cada una de sus tareas, se corresponden con las denominadas *issues*.

Además, se ha utilizado la herramienta *ZenHub* [20] para llevar a cabo la planificación de los *sprints*. Esta herramienta permite llevar un seguimiento de cada *sprint* mediante gráficos estadísticos como *burndown*. También incluye en *GitHub* un nuevo atributo denominado *story points*, el cual permite asociar, a cada tarea, una estimación del esfuerzo que se va a realizar sobre ella.

Para poder trazar un seguimiento óptimo en cada *sprint*, se ha asociado un valor temporal a cada posible valor de *story points*.

Story points	Estimación temporal
1	< 1h
2	1h - 2h
3	2h - 4h
5	4h - 8h
8	8h - 16h
13	16h - 24h
21	1d - 3d
40	3d - 7d

Tabla A.1: Relación entre esfuerzo (*story point*) y tiempo.

A continuación, se describen brevemente cada uno de los *sprints* ejecutados durante el desarrollo del proyecto.

Sprint 0 (28/02/2020 - 13/03/2020)

Cuando se llevó a cabo la reunión de planificación de este primer *sprint*, llevaba trabajando ya con el CENIEH más de tres semanas. Durante este periodo de tiempo, llevé a cabo multitud de tareas relacionadas con el proyecto. Por ello, durante esta primera reunión, mostré todos los avances que había realizado hasta la fecha para conocer la opinión de mis dos tutores, Carlos López y Mario Juez. Tras hacer una valoración de dichos avances, se plantearon nuevos objetivos: migrar todo el trabajo almacenado en el repositorio local del CENIEH al repositorio de Github, realizar un estudio detallado de las diferentes ontologías involucradas en el proyecto (*CIDOC-CRM*, *ACDM* y *AO*-

CAT) y mejorar mi formación en los lenguajes de programación utilizados por la infraestructura escogida (*PHP*, *HTML* y *JavaScript*).

[Ver listado de tareas](#) asociadas al *Sprint 0*.

Sprint 1 (13/03/2020 - 27/03/2020)

Los objetivos planteados para este *sprint* fueron: cubrir las nuevas necesidades del proyecto mediante el desarrollo de nuevos plugins para *Omeka*, estudiar la opción de implementar la tecnología *Docker*, avanzar con la documentación en *LaTeX* y aprender a utilizar el servicio de documentación *Read the docs*.

[Ver listado de tareas](#) asociadas al *Sprint 1*.

Sprint 2 (27/03/2020 - 17/04/2020)

Los objetivos planteados para este *sprint* fueron: finalizar la formación en *Docker*, adaptar el diseño de la aplicación web a las necesidades del usuario objetivo, desarrollar un nuevo plugin que incorpore un sistema de estados para las colecciones/items, gestionar el proyecto a través de ZenHub y, por último, estudiar e implementar las prácticas software de la CI (Integración continua).

[Ver listado de tareas](#) asociadas al *Sprint 2*.



Figura A.1: Gráfico *Burndown* del Sprint 2.

Sprint 3 (17/04/2020 - 08/05/2020)

Los objetivos planteados para este *sprint* fueron: seguir con el análisis e implementación de la “Integración Continua” en el proyecto, finalizar el desarrollo del plugin “ARIADNEplus Monitor”, desplegar la infraestructura sobre

el servidor del CENIEH, actualizar el fichero README y avanzar con la documentación.

[Ver listado de tareas](#) asociadas al *Sprint 3*.

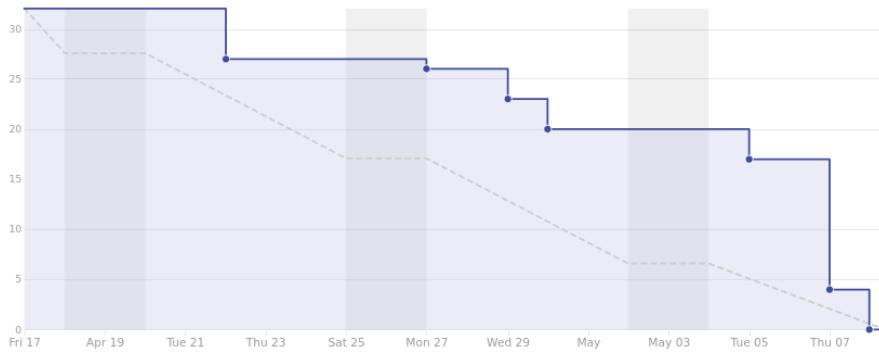


Figura A.2: Gráfico *Burndown* del Sprint 3.

Sprint 4 (08/05/2020 - 22/05/2020)

Los objetivos planteados para este *sprint* fueron: emplear alguna de las herramientas propuestas para evaluar la calidad del código. Añadir mejoras de usabilidad. Solucionar bugs de la interfaz. Desarrollar un plugin para la gestión de tags. Mejorar el README y seguir avanzando con la documentación.

[Ver listado de tareas](#) asociadas al *Sprint 4*.

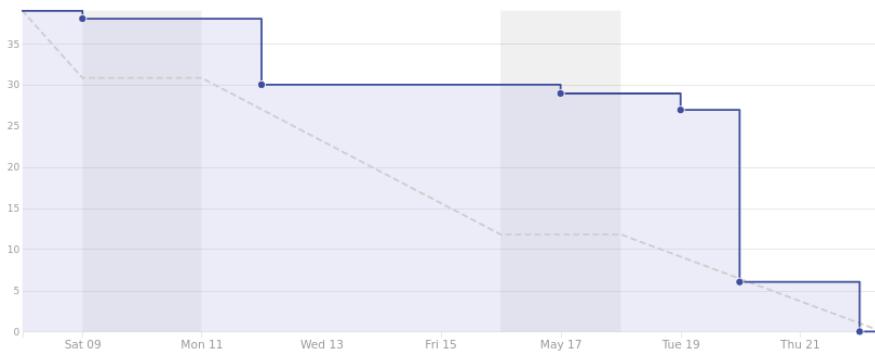


Figura A.3: Gráfico *Burndown* del Sprint 4.

Sprint 5 (22/05/2020 - 12/06/2020)

Los objetivos planteados para este *sprint* fueron: introducir ciertas mejoras sobre el plugin ARIADNEplus Tracking. Crear la colección del CENIEH en pe-

riodO. Completar la colección “CIR” con ayuda del CENIEH. Añadir un estilo al repositorio OAI-PMH. Preparar la importación del CIR en ARIADNEplus. Actualizar el README.md y avanzar con la documentación.

[Ver listado de tareas](#) asociadas al *Sprint 5*.

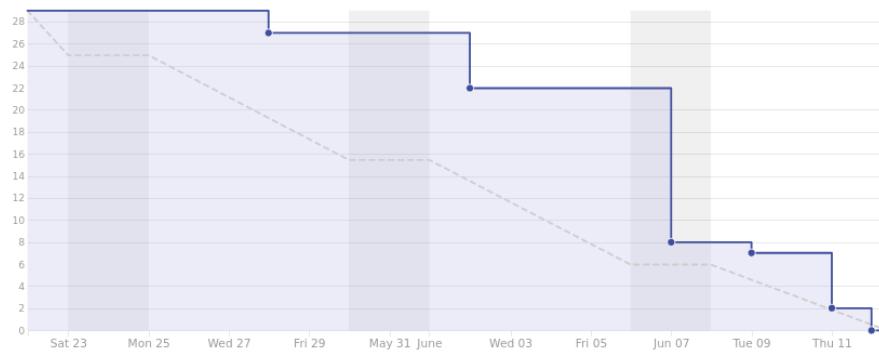


Figura A.4: Gráfico *Burndown* del Sprint 5.

Sprint 6 (12/06/2020 - 26/06/2020)

Los objetivos planteados para este *sprint* fueron: finalizar el desarrollo de la aplicación, publicar la colección de periodos del CENIEH en periodO, publicar la colección del CIR en ARIADNEplus y continuar con la documentación (Memoria).

[Ver listado de tareas](#) asociadas al *Sprint 6*.



Figura A.5: Gráfico *Burndown* del Sprint 6.

Sprint 7 (26/06/2020 - 17/07/2020)

Los objetivos planteados para este *sprint* fueron: continuar con el proceso de integración del CIR en ARIADNEplus. Seguir desarrollando labores de documentación (memoria y anexos).

[Ver listado de tareas](#) asociadas al *Sprint 7*.

A.3. Estudio de viabilidad

Viabilidad económica

Este apartado recoge los análisis de costes y beneficios que se han llevado a cabo para determinar la viabilidad económica del proyecto.

Costes

Los costes del proyecto se pueden clasificar en función de su naturaleza.

Costes de personal

El proyecto se ha llevado a cabo por un desarrollador empleado a tiempo parcial durante 6 meses. La contratación se llevó a cabo mediante el programa de cooperación educativa entre la Universidad de Burgos y el consorcio CE-NIEH. De este acuerdo salieron dos contratos, uno de cuatro meses de duración y otro de dos.

Concepto	Coste
Salario mensual neto	400,00€
Retención IRPF	0,00€
Seguridad social	0,00€
Salario mensual bruto	400,00€
Total (4 meses)	1600,00€

Tabla A.2: Contrato prácticas curriculares

Concepto	Coste
Salario mensual neto	391,51€
Retención IRPF	0,00€
Seguridad social	56,90€

Concepto	Coste
Salario mensual bruto	448,91€
Total (2 meses)	897,82€

Tabla A.3: Contrato prácticas extracurriculares

El primer contrato, al tratarse de prácticas curriculares, queda exento de pagar cualquier tipo de cotización. Sin embargo, en el segundo contrato, al tratarse de prácticas extracurriculares se debe cotizar a la seguridad social.

En el caso de que el sueldo hubiera sido estándar, es decir, sin una beca de por medio, los costes de personal se corresponderían con la siguiente tabla.

Concepto	Coste
Salario mensual neto	521,08€
Retención IRPF (24 %)	271,28€
Seguridad social (29,9 %)	337,97€
Salario mensual bruto	1130,33€
Total (6 meses)	6782,00€

Tabla A.4: Contrato estándar a media jornada de un desarrollador profesional.

Para el cálculo del IRPF se ha consultado el cuadro informativo tipos de retención aplicables (2020) ofrecido por la Agencia Estatal de Administración Tributaria [4]. En cuanto a la seguridad social, se han consultado las bases y tipos de cotización (2019) desde la página oficial del Ministerio de Trabajo, Migraciones y Seguridad Social [17].

Costes de *hardware*

En este apartado se describen los costes relacionados con el equipamiento *hardware* que se ha utilizado para el desarrollo del proyecto. Para calcular el coste amortizado, se ha tenido en cuenta que el tiempo de uso coincide con la duración del proyecto (6 meses) y que su vida útil gira en torno a los 5 años.

Concepto	Coste	Coste amortizado
Ordenador portátil	1050€	105,00€
Monitor auxiliar	249€	24,90€
Servidor rack	1400€	160,00€
Total	2499€	269,90€

Tabla A.5: Costes de hardware

Costes de *software*

Todo el *software* utilizado para el desarrollo del proyecto era totalmente gratuito o contaba con planes libres de pago.

Otros costes

En la siguiente tabla se recogen los demás costes incluídos en el proyecto.

Concepto	Coste
Dominio ubucenh.es	1,00€
Internet	136,00€
Total	137,00€

Tabla A.6: Otros costes

Costes totales

Sumando el total de cada uno de los costes anteriores mostrados con anterioridad, se obtiene el coste total.

Concepto	Coste
Personal	2497,82€
<i>Hardware</i>	269,90€
<i>Software</i>	0,00€
Otros	137,00€
Total	2904,72€

Tabla A.7: Costes totales

Si en el coste de personal tenemos en cuenta un salario estándar, el coste total subiría hasta los 7188,90€.

Además, se podría añadir a esta cantidad el importe de 91.935€ que la Comisión Europea concedió al CENIEH para su integración en *ARIADNEplus* [1].

Beneficios

Con la realización de este proyecto no se pretende obtener ningún beneficio económico o material. Cualquier persona puede hacer uso del material desarrollado de forma totalmente gratuita.

Viabilidad legal

Uno de los factores más importante a tener en cuenta en el desarrollo de un proyecto es escoger el tipo de licencia con el que se distribuirá cada una de sus partes. De esta forma, se define el marco legal en el que se puede utilizar cada parte, es decir, lo que se autoriza a hacer y lo que no.

A continuación, se asignará a cada una de las partes del proyecto, *Software* y documentación, la licencia que más se adapte a nuestros objetivos.

Software

A lo largo del proyecto se han desarrollado varios complementos *software* para una misma aplicación. En este apartado se buscará qué licencia es la más adecuada para todos ellos.

Al hacer uso de recursos de terceros durante el desarrollo de los complementos, la elección de la licencia se ve condicionada por las licencias a las que estos están sujetos.

Dependencia	Licencia
Zend Framework	BSD
Omeka Classic	GPLv3
ZipStream	MIT
Leaflet Draw	MIT
Sweetalert 2	MIT
Notify.js	MIT

Tabla A.8: Licencias de las dependencias utilizadas

Se ha considerado escoger la licencia que *GNU General Public License v3.0* [8] (GPLv3) para cada uno de los complementos *software* desarrollados por varios motivos:

- Es compatible con todas las licencias mostradas en la [licencias](#).
- Al ser una licencia *copyleft*, garantiza que el *software* mantenga su carácter "libre", es decir, que pueda ser siempre utilizado, modificado y redistribuido por cualquier usuario.

Documentación

En el caso de la documentación (memoria y anexos), se ha optado por utilizar la versión más sencilla de las licencias *Creative Commons* [3], que es

conocida como *Creative Commons Attribution 4.0 International* (CC BY 4.0).

Esta licencia es bastante permisiva ya que autoriza a realizar cualquier tipo de operación sobre el documento involucrado, esté o no modificado, sea o no comercial, siempre y cuando se cite al autor.

Existen otras versiones más restrictivas, sin embargo, nuestra intención es dar la mayor libertad posible sobre el material ofrecido en este proyecto.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este anexo se especifican todos los requisitos funcionales y no funcionales de la aplicación *software* propuesta, incluyendo todos aquellos que han sido aportados por complementos (*plugins*).

La aplicación se divide en dos áreas bien diferenciadas, una pública (*frontend*) y otra de administración (*backend*). Por este motivo, se llevará a cabo una especificación de requisitos para cada área.

B.2. Objetivos generales

Los objetivos generales que se desean alcanzar con el desarrollo del proyecto son los siguientes:

- Proporcionar una infraestructura *software* que permita:
 - Gestionar los metadatos del CENIEH en la integración con *ARIADNEplus*.
 - Transformar el esquema de metadatos de origen (CENIEH) a un esquema estandarizado compatible con *ARIADNEplus*.
 - Compartir los datos de forma que estos sean accesibles desde el exterior.
 - Facilitar la integración de los metadatos en *ARIADNEplus*.

B.3. Catálogo de requisitos

En este apartado se muestran todos los requisitos considerados para alcanzar todos y cada uno de los objetivos generales del proyecto.

Requisitos funcionales del área pública

- **RFAP-1 Búsqueda de ítems públicos:** el usuario debe ser capaz de hacer una búsqueda sobre los ítems públicos almacenados en la plataforma.
- **RFAP-2 Visualización de un ítem público:** el usuario debe poder visualizar el contenido (metadatos, tags, localización y/o ficheros) de un determinado ítem público.
 - **RFAP-2.1 Ver fichero:** el usuario debe poder visualizar la información del fichero asociado a un determinado ítem público.
- **RFAP-3 Listado de ítems públicos:** el usuario debe poder listar todos los ítems públicos de la plataforma.
- **RFAP-4 Listado de colecciones públicas:** el usuario debe poder listar todas las colecciones públicas almacenadas en la plataforma.
- **RFAP-5 Visualización de una colección pública:** el usuario debe poder visualizar una colección pública.
- **RFAP-6 Listado de etiquetas públicas:** el usuario debe poder listar todas las etiquetas existentes en la plataforma.
- **RFAP-7 Búsqueda de ítems por localización:** el usuario debe poder buscar un ítem tomando como referencia su ubicación.
- **RFAP-8 Información del proyecto:** el usuario debe poder conocer más información acerca del proyecto.

Requisitos no funcionales del área pública

- **RNFAP-1 Usabilidad:** la aplicación debe presentar los datos de la forma más sencilla posible, evitando así que el usuario se pierda en el proceso de búsqueda o visualización.
- **RNFAP-2 Internacionalización:** la aplicación debe contar con un sistema que permita mostrar el contenido de la interfaz en múltiples idiomas.

- **RNFAP-3 Integridad:** la aplicación debe mostrar los datos tal y como se publicaron desde el área de administración, sin ningún tipo de alteración.

Requisitos funcionales del área de administración

- **RFAA-1 Acceso al área de administración:** la aplicación debe controlar el acceso al área de administración.
- **RFAA-2 Gestión de ítems:** la aplicación tiene que ser capaz de gestionar ítems.
 - **RFAA-2.1 Añadir ítem:** el operario debe poder añadir un nuevo ítem compuesto por metadatos, ficheros, etiquetas y una localización. Además, podrá asociarse a una colección, ser público o privado y ser destacado o normal.
 - **RFAA-2.2 Editar ítem:** el operario debe poder editar el contenido de un ítem ya existente.
 - **RFAA-2.3 Eliminar ítem:** el operario debe poder eliminar un ítem ya existente.
 - **RFAA-2.4 Listar ítems:** el operario debe poder listar todos los ítems existentes.
 - **RFAA-2.5 Ver ítem:** el operario debe poder visualizar todo el contenido asociado a un ítem.
 - **RFAA-2.5.1 Exportar ítem:** el operario debe poder exportar un ítem ya existente.
 - **RFAA-2.6 Buscar ítems:** el operario debe poder buscar un ítem de entre todos los existentes.
 - **RFAA-2.7 Exportar ítems:** el operario debe poder exportar varios ítems existentes a la vez.
- **RFAA-3 Gestión de colecciones:** la aplicación tiene que ser capaz de gestionar colecciones de ítems.
 - **RFAA-3.1 Añadir colección:** el operario debe poder añadir una colección nueva compuesta por metadatos o ficheros. Además, podrá ser pública o privada y ser destacada o normal.
 - **RFAA-3.2 Editar colección:** el operario debe poder editar el contenido de una colección ya existente.
 - **RFAA-3.2.1 Eliminar colección:** el operario debe poder eliminar una colección ya existente.

- **RFAA-3.3 Listar colecciones:** el operario debe poder listar todas las colecciones existentes.
- **RFAA-3.4 Ver colección:** el operario debe poder visualizar toda la información relativa a una colección.
 - **RFAA-3.4.1 Exportar colección:** el operario debe poder exportar una colección ya existente.
- **RFAA-4 Gestión de tipos de ítem:** la aplicación tiene que ser capaz de gestionar tipos de ítem.
 - **RFAA-4.1 Añadir tipo de ítem:** el operario debe poder añadir un tipo de ítem con los elementos apropiados.
 - **RFAA-4.2 Editar tipo de ítem:** el operario debe poder editar un tipo de ítem ya existente.
 - **RFAA-4.2.1 Eliminar tipo de ítem:** el operario debe poder eliminar un tipo de ítem ya existente.
 - **RFAA-4.3 Listar tipos de ítem:** el operario debe poder listar todos los tipos de ítem existentes.
 - **RFAA-4.4 Ver tipo de ítem:** el operario debe poder visualizar toda la información relativa a un tipo de ítem.
- **RFAA-5 Gestión de etiquetas:** la aplicación tiene que ser capaz de gestionar etiquetas.
 - **RFAA-5.1 Editar etiqueta:** el operario debe poder editar una etiqueta ya existente.
 - **RFAA-5.2 Eliminar etiqueta:** el operario debe poder eliminar una etiqueta ya existente.
 - **RFAA-5.3 Listar etiquetas:** el operario debe poder eliminar una etiqueta ya existente.
 - **RFAA-5.4 Eliminar etiquetas:** el operario debe poder eliminar varias etiquetas ya existentes.
- **RFAA-6 Importación de metadatos CSV:** la aplicación tiene que ser capaz de importar metadatos en formato CSV.
 - **RFAA-6.1 Ejecutar importación CSV:** el operario debe poder ejecutar una importación de metadatos en formato CSV.
 - **RFAA-6.2 Listar importaciones CSV:** el operario debe poder listar todas las importaciones ejecutadas.

- **RFAA-6.2.1 Deshacer importación CSV:** el operario debe poder deshacer una importación ya ejecutada.
- **RFAA-7 Recolección de metadatos (OAI-PMH):** la aplicación tiene que ser capaz de recolectar metadatos a través del protocolo OAI-PMH.
 - **RFAA-7.1 Ejecutar recolección:** el operario debe poder ejecutar una recolección a través del protocolo OAI-PMH.
 - **RFAA-7.2 Actualizar recolección:** el operario debe poder actualizar una recolección ya ejecutada.
 - **RFAA-7.3 Listar recolecciones:** el operario debe poder listar todas las recolecciones ejecutadas.
 - **RFAA-7.4 Ver recolección:** el operario debe poder visualizar toda la información relativa a una recolección.
 - **RFAA-7.4.1 Deshacer recolección:** el operario debe poder deshacer una recolección ya ejecutada.
- **RFAA-8 Seguimiento *ARIADNEplus*:** el operario tiene que ser capaz de realizar un seguimiento del proceso de integración de datos en *ARIADNEplus*.
 - **RFAA-8.1 Crear ticket de seguimiento:** el operario debe poder crear un ticket de seguimiento.
 - **RFAA-8.2 Eliminar ticket de seguimiento:** el operario debe poder eliminar un ticket de seguimiento ya existente.
 - **RFAA-8.3 Administrar ticket de seguimiento:** el operario debe poder administrar un ticket de seguimiento ya existente.
 - **RFAA-8.3.1 Cambiar de fase:** el operario debe poder cambiar de fase de un ticket existente.
- **RFAA-9 Edición masiva de metadatos:** el operario debe ser capaz de editar una gran cantidad de metadatos a la vez.
- **RFAA-10 Gestión de complementos:** el operario debe ser capaz de gestionar los complementos existentes en la aplicación.
 - **RFAA-10.1 Instalar complemento:** el operario debe poder instalar un complemento existente.

- **RFAA-10.2 Configurar complemento:** el operario debe poder configurar un complemento ya instalado.
- **RFAA-10.3 Desinstalar complemento:** el operario debe poder desinstalar un complemento ya instalado.
- **RFAA-11 Configuración de la apariencia:** el operario debe ser capaz de configurar la apariencia de la aplicación.
 - **RFAA-11.1 Usar plantilla:** el operario debe poder instalar (usar) una plantilla/tema existente.
 - **RFAA-11.2 Configurar plantilla:** el operario debe poder configurar la plantilla instalada (usada).
 - **RFAA-11.3 Configurar parámetros:** el operario debe poder configurar parámetros relacionados con la apariencia (número de ítems por página, entradas de navegación, etc.).
- **RFAA-12 Configuración de la aplicación:** el operario debe ser capaz de configurar aspectos de la aplicación.
- **RFAA-13 Gestión de usuarios:** la aplicación tiene que ser capaz de gestionar usuarios.
 - **RFAA-13.1 Añadir usuario:** el operario debe ser capaz de crear un nuevo usuario.
 - **RFAA-13.2 Editar usuario:** el operario debe ser capaz de editar un usuario existente.
 - **RFAA-13.3 Eliminar usuario:** el operario debe ser capaz de eliminar un usuario existente.
 - **RFAA-13.4 Listar usuarios:** el operario debe poder listar todas los usuarios existentes.

Requisitos no funcionales del área de administración

- **RNFAA-1 Usabilidad:** la aplicación debe contar con una estructura clara y sencilla, que permita al operario desplazarse por todas las funcionalidades disponibles a través de pestañas o apartados bien posicionados.
- **RNFAA-2 Seguridad:** los datos de la aplicación solo podrán ser administrados o visualizados por usuarios autorizados.
- **RNFAA-3 Extensibilidad:** la aplicación debe contar con un sistema que permita agregar características y/o funcionalidades adicionales exigidas por los operarios.
- **RNFAA-4 Disponibilidad:** la aplicación debe ser accesible desde cualquier navegador común.

- **RNFAA-5 Robustez:** la aplicación debe estar preparada para un funcionamiento correcto ante excepciones que pudieran surgir.
- **RNFAA-6 Rendimiento y Escalabilidad:** la aplicación debe estar a la altura en cuanto a tiempos de respuesta y debe ser capaz de prestar servicio de acuerdo al tipo y tamaño de datos para el que ha sido concebido.
- **RNFAA-7 Mantenibilidad:** la aplicación debe ser de fácil instalación y mantenimiento, es decir, que cuente con código y diseño documentado y además pueda ser actualizada a versiones más modernas.
- **RNFAA-8 Estandarización:** la aplicación debe hacer uso de estándares internacionalmente aceptados para el almacenamiento de metadatos.
- **RNFAA-9 Interoperabilidad:** la aplicación debe permitir intercambiar información con otros sistemas de la misma índole.

B.4. Especificación de requisitos

A continuación se mostrará el diagrama de casos de usos que agrupa cada uno de los requisitos funcionales expuestos en el catálogo de requisitos. Además, se tratará cada uno de ellos por separado.

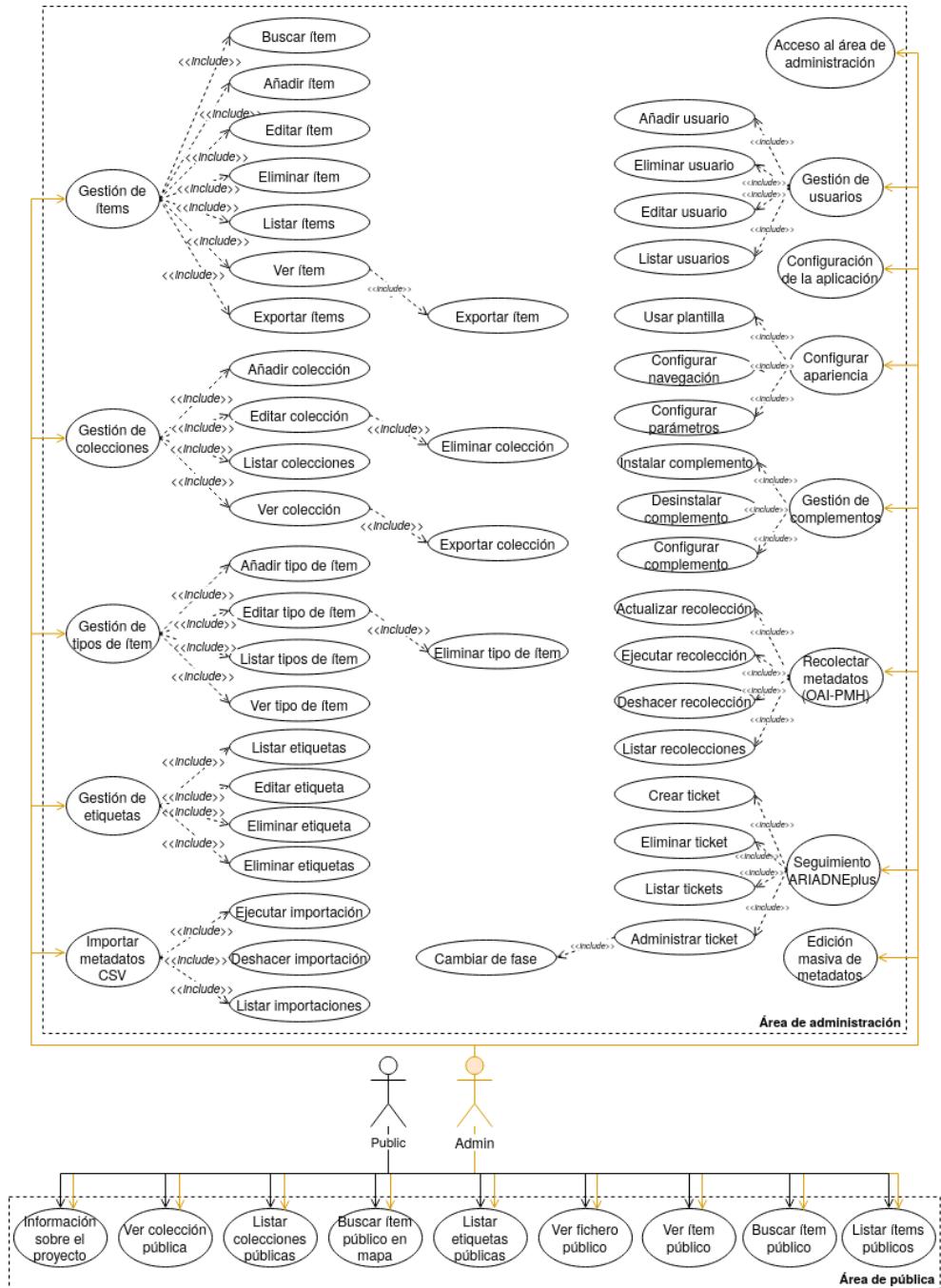


Figura B.1: Diagrama de casos de la aplicación completa.

Actores

Se han considerado dos actores principales:

- **Public:** usuario no registrado en la aplicación. Solo tiene acceso a los datos públicos de la aplicación.
- **Admin:** usuario registrado en la aplicación. Puede adquirir diversos roles que determinan las zonas a las que este puede acceder dentro del área de administración. Actualmente hay 4 roles:
 - *Super:* gestiona todos los datos existentes en la aplicación y lleva a cabo las tareas de mantenimiento y configuración.
 - *Admin:* gestiona todos los datos existentes en la aplicación.
 - *Contributor:* solo puede gestionar sus propios datos y visualizar todos los datos de la aplicación.
 - *Researcher:* únicamente puede visualizar los datos de la aplicación.

Casos de uso

CU-01	Buscar ítems públicos
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAP-1
Descripción	Permite al usuario buscar objetos digitales públicos en la plataforma.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario introduce texto sobre el campo “Search”. 2. El usuario presiona la tecla “Enter”. 3. Se muestran todos los resultados (ítems).
Postcondición	Todos los ítems resultantes presentan coincidencias con el campo de texto introducido.
Excepciones	<ul style="list-style-type: none"> ■ Ningún ítem encontrado (mensaje).
Importancia	Alta

Tabla B.1: CU-01 Buscar ítems públicos.

CU-02	Listar ítems públicos
Versión	1.0
Autor	Gonzalo Cuesta Marín

CU-02	Listar ítems públicos
Requisitos asociados	RFAP-3
Descripción	Permite al usuario ver una lista de todos los ítems públicos en la plataforma.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Menu”. 2. Se muestran todas las entradas del menú de navegación público. 3. El usuario pulsa sobre la entrada “Browse Items”. 4. Se muestra una lista con todos los ítems públicos de la plataforma.
Postcondición	El número de ítems mostrado es igual al número de ítems públicos almacenado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ No existe ningún ítem público (vista especial).
Importancia	Alta

Tabla B.2: CU-02 Listar ítems públicos.

CU-03	Listar colecciones públicas
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAP-4
Descripción	Permite al usuario ver una lista de todas las colecciones públicas en la plataforma.
Precondición	Base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Menu”. 2. Se muestran todas las entradas del menú de navegación público. 3. El usuario pulsa sobre la entrada “Browse Collections”. 4. Se muestra una lista con todas las colecciones públicas de la plataforma.

CU-03	Listar colecciones públicas
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ No existe ninguna colección pública (vista especial).
Importancia	Alta

Tabla B.3: CU-03 Listar colecciones públicas.

CU-04	Listar etiquetas públicas
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAP-6
Descripción	Permite al usuario ver una lista de todas las etiquetas almacenadas en la plataforma.
Precondición	Base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Browse all tags”. 2. Se muestra una lista con todas las etiquetas almacenadas en la plataforma.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ No existe ninguna etiqueta (vista especial).
Importancia	Alta

Tabla B.4: CU-04 Listar etiquetas públicas.

CU-05	Ver ítem público
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAP-2, RFAP-2.1
Descripción	Permite al usuario visualizar un ítem público existente.
Precondición	Servidor y base de datos disponible.
	El ítem público a visualizar existe.

CU-05	Ver ítem público
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el título del ítem. 2. Se muestra información general del ítem (metadatos, tags, localización y cita bibliográfica).
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ El ítem no existe (mensaje).
Importancia	Alta

Tabla B.5: CU-05 Ver ítem público.

CU-06	Ver colección pública
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos	RFAP-5
asociados	
Descripción	Permite al usuario visualizar un ítem público existente.
Precondición	Servidor y base de datos disponible.
	La colección pública a visualizar existe.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el título de la colección. 2. Se muestra el título de la colección. 3. Se muestra un botón para acceder a la lista de ítems de la colección.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ La colección no existe (mensaje).
Importancia	Alta

Tabla B.6: CU-06 Ver colección pública.

CU-07	Ver fichero público
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAP-2.1
Descripción	Permite al usuario visualizar el fichero de un ítem público existente.
Precondición	Servidor y base de datos disponible. El ítem público asociado al fichero existe.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el nombre del fichero. 2. Se muestra información general del fichero (nombre, metadatos e ítem asociado).
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ El fichero no existe (mensaje).
Importancia	Alta

Tabla B.7: CU-07 Ver fichero público.

CU-08	Buscar ítem público en mapa
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAP-7
Descripción	Permite al usuario buscar objetos digitales públicos en la plataforma.
Precondición	Servidor y base de datos disponible.

CU-08	Buscar ítem público en mapa
Acciones	<ol style="list-style-type: none"> 1. El usuario se desplaza a la localización deseada arrastrando el cursor por el mapa. 2. El usuario ajusta el zoom para una búsqueda más precisa (opcional). 3. El usuario cambia la plantilla del mapa (opcional). 4. Se muestran los ítems, en forma de marcador, que estén ubicados en la zona geográfica fijada. 5. El usuario pulsa sobre un marcador (ítem). 6. Se muestra un bocadillo con información mínima del ítem (título).
Postcondición	
Excepciones	
Importancia	Alta

Tabla B.8: CU-08 Buscar ítem público en mapa.

CU-09	Información del proyecto
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAP-8
Descripción	Permite al usuario conocer más información acerca del proyecto.
Precondición	Base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Menu”. 2. Se muestran todas las entradas del menú de navegación público. 3. El usuario pulsa sobre la entrada “About”. 4. Se muestra la información general del proyecto
Postcondición	
Excepciones	
Importancia	Media

Tabla B.9: CU-09 Información del proyecto.

CU-10	Acceso al área de administración
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-1
Descripción	Permite al usuario acceder al área de administración.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. Se muestra al usuario el formulario de inicio de sesión. 2. El usuario introduce su nombre de usuario. 3. El usuario introduce su contraseña de usuario. 4. El usuario pulsa sobre el botón “Login”. 5. Si los datos son correctos, se traslada al usuario al interior del área de administración.
Postcondición	El usuario accede a la pantalla principal del área de administración
Excepciones	<ul style="list-style-type: none"> ▪ Nombre de usuario o contraseña incorrectos (mensaje). ▪ Usuario inactivo (mensaje).
Importancia	Alta

Tabla B.10: CU-10 Acceso al área de administración.

CU-11	Gestión de ítems
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2, RFAA-2.1, RFAA-2.2, RFAA-2.3, RFAA-2.4, RFAA-2.5, RFAA-2.5.1, RFAA-2.6, RFAA-2.7
Descripción	Permite al usuario gestionar ítems.
Precondición	Servidor y base de datos disponible.
	Sesión iniciada en área de administración.

CU-11	Gestión de ítems
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Items” del submenú “Data Manager” del menú principal. 2. Se listan los ítems. 3. Por cada ítem se da la opción de ver en detalle, editar o eliminar. 4. Se muestra un botón para añadir un ítem. 5. Se muestra un botón y un desplegable para buscar o filtrar ítems. 6. Se muestran múltiples formatos de exportación disponibles.
Postcondición	El número de ítems listado es igual al número de ítems almacenado en la base de datos
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar ítems (mensaje). ■ No existe ningún mensaje (vista especial)
Importancia	Alta

Tabla B.11: CU-11 Gestión de ítems.

CU-12	Añadir ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2.1
Descripción	Permite al usuario añadir un nuevo ítem.
Precondición	Servidor y base de datos disponible.

CU-12 Añadir ítem

Acciones

1. El usuario pulsa sobre el botón “Add an Item”.
2. Se muestra al usuario una ventana con varios formularios divididos en pestañas y varios botones.
3. El usuario pulsa sobre la pestaña “Dublin Core” para añadir metadatos estandarizados (opcional).
4. El usuario pulsa sobre la pestaña “Item Type Metadata” para añadir metadatos específicos (opcional).
5. El usuario pulsa sobre la pestaña “Files” para añadir ficheros (opcional).
6. El usuario pulsa sobre la pestaña “Tags” para añadir etiquetas (opcional).
7. El usuario pulsa sobre la pestaña “Map” para añadir una localización (opcional).
8. El usuario marca la casilla “Public” para hacerlo público (opcional).
9. El usuario marca la casilla “Feature” para hacerlo destacado (opcional).
10. El usuario selecciona la colección a la que pertenece (opcional).
11. El usuario pulsa el botón “Add Item”.
12. Si no hay ningún error, se guarda un nuevo ítem junto a los demás objetos asociados.

Postcondición Existe un ítem más en la base de datos.

- Excepciones**
- Error al crear ítem (mensaje).
 - Error con el tipo de fichero introducido (mensaje).

Importancia Alta

Tabla B.12: CU-11 Añadir ítem.

CU-13 Editar ítem

Versión 1.0

Autor Gonzalo Cuesta Marín

Requisitos asociados RFAA-2.2

Descripción Permite al usuario editar un ítem existente.

CU-13 Editar ítem

PrecondiciónServidor y base de datos disponible.

El ítem a editar existe.

Acciones

1. El usuario pulsa sobre el elemento “Edit” del ítem que va a editar.
2. Se muestra al usuario una ventana con varios formularios divididos en pestañas y tres botones (guardar, ver en área pública y eliminar).
3. El usuario pulsa sobre la pestaña “Dublin Core” para editar metadatos estandarizados (opcional).
4. El usuario pulsa sobre la pestaña “Item Type Metadata” para editar metadatos específicos (opcional).
5. El usuario pulsa sobre la pestaña “Files” para editar ficheros (opcional).
6. El usuario pulsa sobre la pestaña “Map” para editar una localización (opcional).
7. El usuario modifica el estado de la casilla “Public” para hacerlo público/privado (opcional).
8. El usuario modifica el estado de la casilla “Feature” para hacerlo o no destacado (opcional).
9. El usuario modifica la colección a la que pertenece (opcional).
10. El usuario pulsa el botón “Save Changes”.
11. Si no hay ningún error, se actualiza el ítem junto a todos sus objetos asociados (metadatos/ficheros/etiquetas/localización).

PostcondiciónLa información del ítem y de todos sus objetos asociados ha sido actualizada.

Excepciones

- Error al guardar ítem (mensaje).
- Error con el tipo de fichero introducido (mensaje).

Importancia Alta

Tabla B.13: CU-13 Editar ítem.

CU-14 Eliminar ítem

Versión 1.0

CU-14	Eliminar ítem
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2.3
Descripción	Permite al usuario eliminar un ítem existente.
Precondición	Servidor y base de datos disponible.
	El ítem a eliminar existe.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre “Delete”. 2. Se muestra al usuario un mensaje de confirmación. 3. El usuario pulsa sobre el botón “Delete” para confirmar la eliminación. 4. Se elimina el ítem de la vista. 5. Se informa al usuario que el ítem ha sido eliminado correctamente.
Postcondición	El ítem y sus objetos asociados (metadatos, ficheros y localización) no existen en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al eliminar ítem (mensaje).
Importancia	Alta

Tabla B.14: CU-14 Eliminar ítem.

CU-15	Listar ítems
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2.4
Descripción	Permite al usuario ver una lista de todos los ítems almacenados en la aplicación.
Precondición	Servidor y base de datos disponible.

CU-15	Listar ítems
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Items” del submenú “Data Manager” del menú principal. 2. Se muestran los ítems en una tabla indicando su título, creador, tipo y fecha de creación. 3. Opcionalmente se pueden ordenar los ítems pulsando sobre alguna de las cabeceras de la tabla.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ No existen ítems (vista especial).
Importancia	Alta

Tabla B.15: CU-15: Listar ítems.

CU-16	Ver ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2.5, RFAA-2.5.1
Descripción	Permite al usuario visualizar un ítem
Precondición	Servidor y base de datos disponible. El ítem a visualizar existe.

CU-16	Ver ítem
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el título del ítem a visualizar. 2. Se obtiene la información del ítem (incluída la de sus objetos asociados). 3. Se muestra información general del ítem. 4. Se muestra información de cada uno de los objetos asociados al ítem. <ul style="list-style-type: none"> a. Metadatos (Elementos de texto) b. Ficheros (Miniatura y nombre) c. Etiquetas (Nombres) d. Localización (Minimap) 5. Se muestra información adicional. <ul style="list-style-type: none"> a. Estado ARIADNE+. b. Cita bibliográfica. c. Formatos de exportación.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar el ítem (mensaje).

Importancia

Alta

Tabla B.16: CU-16 Ver ítem.

CU-17	Exportar ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2.5.1
Descripción	Permite al usuario exportar un ítem
Precondición	Base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona el formato de exportación. 2. Se exporta el ítem en el formato seleccionado.
Postcondición	El formato del fichero exportado es el seleccionado.
Excepciones	

CU-17	Exportar ítem
Importancia	Alta

Tabla B.17: CU-17 Exportar ítem.

CU-18	Buscar ítems
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2.6
Descripción	Permite al usuario buscar uno o varios ítems
Precondición	Servidor y base de datos disponible.
Acciones	<p>Situado en la vista de gestión de ítems.</p> <ol style="list-style-type: none"> 1. El usuario pulsa el botón “Search Items”. 2. Se muestra el formulario de búsqueda avanzada. 3. El usuario introduce sus criterios de búsqueda. 4. El usuario pulsa el botón “Search for Items”. 5. Se muestran los resultados (ítems) de la búsqueda.
Postcondición	Los ítems mostrados cumplen con los criterios establecidos en la búsqueda.
Excepciones	<ul style="list-style-type: none"> ■ La búsqueda no ha obtenido resultados (vista especial).
Importancia	Alta

Tabla B.18: CU-18 Buscar ítems.

CU-19	Exportar ítems
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-2.7
Descripción	Permite al usuario exportar varios ítems.
Precondición	Servidor y base de datos disponible.

CU-19	Exportar ítems
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona el formato de exportación. 2. Se exportan todos los ítems de la página actual en el formato seleccionado.
Postcondición	El formato del fichero exportado es el seleccionado.
Excepciones	
Importancia	Alta

Tabla B.19: CU-19 Exportar ítems.

CU-20	Gestión de colecciones
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos	RFAA-3, RFAA-3.1, RFAA-3.2, RFAA-3.2.1, RFAA-3.3,
asociados	RFAA-3.4, RFAA-3.4.1
Descripción	Permite al usuario gestionar colecciones.
Precondición	Servidor y base de datos disponible.
	Sesión iniciada en área de administración.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Collections” del submenú “Data Manager” del menú principal. 2. Se listan las colecciones. 3. Por cada colección se da la opción de ver en detalle, editar o eliminar. 4. Se muestra un botón para añadir colección.
Postcondición	El número de colecciones listado debe coincidir con el número de colecciones almacenado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ No existe ninguna colección (vista especial).
Importancia	Alta

Tabla B.20: CU-20 Gestión de colecciones.

CU-21	Añadir colección
Versión	1.0

CU-21	Añadir colección
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-3.1
Descripción	Permite al usuario añadir una nueva colección.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Add a Collection”. 2. Se muestra al usuario una ventana con varias pestañas para desplazarse durante el proceso de creación. 3. El usuario pulsa sobre la pestaña “Dublin Core” para añadir metadatos estandarizados (opcional). 4. El usuario pulsa sobre la pestaña “Files” para añadir ficheros (opcional). 5. El usuario marca la casilla “Public” para hacer pública la colección (opcional). 6. El usuario marca la casilla “Feature” para hacer destacada la colección (opcional). 7. El usuario pulsa sobre el botón “Add Collection”.
Postcondición	Existe una colección más en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Error al crear la colección (mensaje). ▪ Error con el tipo de fichero introducido (mensaje)

Importancia Alta

Tabla B.21: CU-21 Añadir colección.

CU-22	Editar colección
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-3.2
Descripción	Permite al usuario editar una colección existente.
Precondición	Servidor y base de datos disponible.

CU-22	Editar colección
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el texto “Edit” situado en la colección a editar. 2. Se muestra al usuario una ventana con varios formularios divididos en pestañas y varios botones. 3. El usuario pulsa sobre la pestaña “Dublin Core” para editar metadatos estandarizados (opcional). 4. El usuario pulsa sobre la pestaña “Files” para editar ficheros (opcional). 5. El usuario modifica el estado de la casilla “Public” para hacer pública la colección (opcional). 6. El usuario modifica el estado de la casilla “Feature” para hacer destacada la colección (opcional). 7. El usuario pulsa sobre el botón “Save Changes”.
Postcondición	La información de la colección y de todos sus objetos asociados ha sido actualizada.
Excepciones	<ul style="list-style-type: none"> ■ Error al guardar el ítem (mensaje). ■ Error con el tipo de fichero introducido (mensaje).
Importancia	Alta

Tabla B.22: CU-22: Editar colección.

CU-23	Eliminar colección
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-3.2.1
Descripción	Permite al usuario eliminar una colección existente.
Precondición	Servidor y base de datos disponible.

CU-23 Eliminar colección

Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el elemento “Delete”. 2. Se muestra al usuario un mensaje de confirmación para ejecutar la eliminación. 3. El usuario pulsa de nuevo sobre el botón “Delete”. 4. Se elimina la colección de la vista. 5. Se informa al usuario que la colección ha sido eliminada satisfactoriamente.
-----------------	---

Postcondición La colección eliminada y sus objetos asociados (metadatos y ficheros) no existen en la base de datos.

Excepciones ■ Error al eliminar la colección (mensaje).

Importancia Alta

Tabla B.23: CU-23 Eliminar colección.

CU-24 Listar colecciones

Versión 1.0

Autor Gonzalo Cuesta Marín

Requisitos asociados RFAA-3.3

Descripción Permite al usuario ver una lista de todas las colecciones almacenadas en la aplicación.

Precondición Servidor y base de datos disponible.

Acciones

1. El usuario pulsa sobre la entrada “Collections” del submenú “Data Manager” del menú principal.
2. Se muestran las colecciones en una tabla indicando su título, contribuidores, fecha de creación y número de ítems asociados.
3. Opcionalmente se pueden ordenar las colecciones pulsando sobre alguna de las cabeceras de la tabla.

Postcondición

Excepciones ■ No existen colecciones (vista especial).

CU-24 Listar colecciones

Importancia Alta

Tabla B.24: CU-23 Listar colecciones.

CU-25	Ver colección
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-3.4, RFAA-3.4.1
Descripción	Permite al usuario visualizar una colección.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el título de la colección a visualizar. 2. Se obtiene la información de la colección (incluida la de sus objetos asociados). 3. Se muestra información general de la colección. 4. Se muestra información de cada uno de los objetos asociados a la colección. <ul style="list-style-type: none"> a. Metadatos (Elementos de texto). b. Ficheros (Nombre). 5. Se muestra información adicional. <ul style="list-style-type: none"> a. Estado ARIADNE+. b. Lista de ítems asociados. c. Formatos de exportación. d. Número total de ítems. e. Contribuidores. 6. Se muestran múltiples botones. <ul style="list-style-type: none"> a. Editar. b. Ver en área pública. c. Eliminar.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar la colección (mensaje).
Importancia	Alta

CU-25	Ver colección
--------------	----------------------

Tabla B.25: CU-25 Ver colección.

CU-26	Exportar colección
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-3.4.1
Descripción	Permite al usuario exportar una colección.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona el formato de exportación de la colección. 2. Se exporta la colección en el formato seleccionado.
Postcondición	El formato del fichero exportado es el seleccionado.
Excepciones	
Importancia	Alta

Tabla B.26: CU-26 Exportar colección.

CU-27	Gestión de tipos de ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-4, RFAA-4.1, RFAA-4.2, RFAA-4.2.1, RFAA-4.3, RFAA-4.4
Descripción	Permite al usuario gestionar tipos de ítem.
Precondición	Servidor y base de datos disponible.
Acciones	<p>Sesión iniciada en el área de administración.</p> <ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Item Types” del submenú “Data Manager” del menú principal. 2. Se listan los tipos de ítem. 3. Por cada tipo de ítem se da la opción de ver en detalle o editar. 4. Se muestra un botón para añadir un tipo de ítem.

CU-27	Gestión de tipos de ítem
Postcondición	El número de tipos de ítem listado es igual al número de tipos de ítem almacenado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar tipos de ítem (mensaje). ■ No existe ningún tipo de ítem (vista especial).

Importancia Alta

Tabla B.27: CU-27 Gestión de tipos de ítem.

CU-28	Añadir tipo de ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-4.1
Descripción	Permite al usuario añadir un nuevo tipo de ítem.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Add an Item Type”. 2. Se muestra al usuario el formulario para introducir datos del tipo de ítem. 3. El usuario introduce el nombre. 4. El usuario introduce una descripción (opcional). 5. El usuario añade elementos existentes o nuevos (opcional). 6. El usuario pulsa sobre el botón “Add Item Type”. 7. Si no hay ningún error, se guarda un nuevo tipo de ítem con los datos introducidos.
Postcondición	Existe un tipo de ítem más en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al crear el tipo de ítem (mensaje). ■ No se ha introducido nombre del tipo de ítem (mensaje).

Importancia Alta

Tabla B.28: CU-28 Añadir tipo de ítem.

CU-29	Editar tipo de ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-4.2, RFAA-4.2.1
Descripción	Permite al usuario editar un tipo de ítem existente.
Precondición	Servidor y base de datos disponible.
	Tipo de ítem a editar existente.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el elemento “Edit”. 2. Se muestra al usuario el formulario con los datos actuales del tipo de ítem. 3. El usuario modifica el nombre. 4. El usuario modifica la descripción (opcional). 5. El usuario modifica los elementos (opcional). 6. El usuario pulsa sobre el botón “Save Changes”. 7. Si no hay ningún error, se actualiza el tipo de ítem en la base de datos.
Postcondición	La información del tipo de ítem y de sus elementos ha sido actualizada en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al editar el tipo de ítem (mensaje). ■ No se ha introducido nombre del tipo de ítem (mensaje).
Importancia	Alta

Tabla B.29: CU-29 Editar tipo de ítem.

CU-30	Eliminar tipo de ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-4.2.1
Descripción	Permite al usuario eliminar un tipo de ítem existente.
Precondición	Servidor y base de datos disponible.
	Tipo de ítem a eliminar existente.

CU-30	Eliminar tipo de ítem
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Delete”. 2. Se muestra al usuario un mensaje de confirmación antes de ejecutar la acción. 3. El usuario pulsa de nuevo sobre el botón “Delete”. 4. Se elimina el tipo de ítem de la vista. 5. Se informa al usuario que el tipo de ítem ha sido eliminado satisfactoriamente.
Postcondición	Existe un tipo de ítem menos en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al eliminar el tipo de ítem (mensaje).
Importancia	Alta

Tabla B.30: CU-30 Eliminar tipo de ítem.

CU-31	Listar tipos de ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-4.3
Descripción	Permite al usuario ver una lista de todos los tipos de ítem almacenados en la aplicación.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Item Types” del submenú “Data Manager” del menú principal. 2. Se muestran los tipos de ítem en una tabla indicando su nombre, descripción y número de ítems asignados.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ No existen tipos de ítem (vista especial).
Importancia	Alta

Tabla B.31: CU-31 Listar tipos de ítem.

CU-32	Ver tipo de ítem
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-4.4
Descripción	Permite al usuario visualizar un tipo de ítem.
Precondición	Base de datos disponible.
	El tipo de ítem a visualizar existe.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el título del tipo de ítem a visualizar. 2. Se obtienen los datos del tipo de ítem de la base de datos (incluidos sus elementos e ítems asignados). 3. Se muestra su información general (nombre, descripción y número de ítems asignados). 4. Se muestran todos sus elementos (metadatos). 5. Se muestra una pequeña lista con los últimos ítems asignados. 6. Se muestra un botón de editar y otro de eliminar.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar el tipo de ítem (mensaje).
Importancia	Alta

Tabla B.32: CU-32 Ver tipo de ítem.

CU-33	Gestión de etiquetas
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-5, RFAA-5.1, RFAA-5.2, RFAA-5.3, RFAA-5.4
Descripción	Permite al usuario gestionar etiquetas.
Precondición	Servidor y base de datos disponible.
	Sesión iniciada en el área de administración.

CU-33	Gestión de etiquetas
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Tags” del submenú “Data Manager” del menú principal. 2. Se listan las etiquetas. 3. Por cada etiqueta se da la opción de editar o eliminar. 4. Se muestra un botón de eliminar. 5. Se muestra un formulario de búsqueda.
Postcondición	El número de etiquetas listado es igual al número de etiquetas almacenado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Error al cargar etiquetas (mensaje). ▪ No existe ninguna etiqueta (vista especial).
Importancia	Alta

Tabla B.33: CU-33 Gestión de etiquetas.

CU-34	Editar etiqueta
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RRFAA-5.1
Descripción	Permite al usuario editar una etiqueta existente.
Precondición	Servidor y base de datos disponible.
Acciones	<p>La etiqueta a editar existe.</p> <ol style="list-style-type: none"> 1. El usuario pulsa sobre el nombre de la etiqueta a editar. 2. El nombre de la etiqueta es ahora editable. 3. El usuario introduce el nuevo valor para el nombre de la etiqueta 4. El usuario presiona la tecla enter o pulsa fuera de la etiqueta. 5. Si no hay ningún error, se actualiza la etiqueta en la base de datos.

CU-34	Editar etiqueta
Postcondición	La información de la etiqueta ha sido actualizada en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Error al cargar etiquetas (mensaje). ▪ Campo vacío (ignora la acción).
Importancia	Alta

Tabla B.34: CU-34 Editar etiqueta.

CU-35	Eliminar etiqueta
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RRFAA-5.2
Descripción	Permite al usuario eliminar una etiqueta existente.
Precondición	Servidor y base de datos disponible.
Acciones	<p>La etiqueta a eliminar existe.</p> <ol style="list-style-type: none"> 1. El usuario pulsa sobre el aspa de la etiqueta a eliminar. 2. Se muestra al usuario un mensaje de confirmación antes de ejecutar la acción. 3. El usuario pulsa sobre el botón “Delete”. 4. Se elimina la etiqueta de la vista.
Postcondición	Existe una etiqueta menos en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Error al eliminar la etiqueta (mensaje).
Importancia	Alta

Tabla B.35: CU-35 Eliminar etiqueta.

CU-36	Listar etiquetas
Versión	1.0
Autor	Gonzalo Cuesta Marín

CU-36 Listar etiquetas

Requisitos asociados RRFAA-5.3

Descripción Permite al usuario ver una lista de todas las etiquetas almacenadas en la aplicación.

Precondición Servidor y base de datos disponible.

Acciones

1. El usuario pulsa sobre la entrada “Tags” del submenú “Data Manager” del menú principal.
2. Se muestran todas las etiquetas en forma de flecha con su nombre en el interior.
3. Opcionalmente el usuario puede buscar etiquetas por su nombre.
4. Opcionalmente el usuario puede ordenar etiquetas por su nombre, número de ítems asignados y fecha de creación.
5. Opcionalmente el usuario puede filtrar etiquetas por el tipo de objeto asignado.

Postcondición Las etiquetas listadas contienen en su nombre el texto buscado.

Excepciones

- Error al eliminar las etiquetas (mensaje).
- No se poseen los permisos necesarios (mensaje).

Importancia Alta

Tabla B.36: CU-36 Listar etiquetas.

CU-37 Eliminar etiquetas

Versión 1.0

Autor Gonzalo Cuesta Marín

Requisitos asociados RRFAA-5.4

Descripción Permite al usuario eliminar varias etiquetas existentes.

Precondición Servidor y base de datos disponible.

Las etiquetas a eliminar existen.

CU-37	Eliminar etiquetas
Acciones	<ol style="list-style-type: none"> 1. El usuario busca las etiquetas a eliminar por su nombre (opcional) 2. El usuario pulsa sobre el botón rojo “Delete Results”. 3. Se muestra al usuario un mensaje de confirmación antes de ejecutar la acción. 4. El usuario pulsa sobre el botón “Delete”. 5. Se eliminan las etiquetas de la vista.
Postcondición	Existen N etiquetas menos en la base de datos, siendo N el número de etiquetas eliminadas.
Excepciones	<ul style="list-style-type: none"> ▪ Error al eliminar las etiquetas (mensaje).
Importancia	Alta

Tabla B.37: CU-37 Eliminar etiquetas.

CU-38	Importación de metadatos CSV
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RRFAA-6, RFAA-6.1, RFAA-6.2, RFAA-6.2.1
Descripción	Permite al usuario importar metadatos en formato CSV en la aplicación.
Precondición	Servidor y base de datos disponible.
	Sesión iniciada en el área de administración.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “CSV Import+” del submenú “Import Tools” del menú principal. 2. Se muestra una ventana con dos pestañas, “Import” (actual) y “Status”.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ▪ Error al cargar el complemento (mensaje).

CU-38 Importación de metadatos CSV

Importancia Alta

Tabla B.38: CU-38 Importación de metadatos CSV.

CU-39 Ejecutar importación CSV

Versión 1.0

Autor Gonzalo Cuesta Marín

Requisitos RRFAA-6.1

asociados

Descripción Permite al usuario importar metadatos en formato CSV en la aplicación.

Precondición Servidor y base de datos disponible.

Acciones

1. El usuario accede a la pestaña “Import” del complemento “CSV Import+”.
2. Se muestra el formulario de importación.
3. El usuario carga el fichero CSV con los metadatos.
4. El usuario selecciona los delimitadores y cierres del fichero CSV cargado.
5. El usuario establece los valores por defecto.
6. El usuario configura las características para el proceso de importación.
7. El usuario pulsa sobre el botón “Next”.
8. El usuario establece las relaciones entre el esquema de datos original y el esquema de datos de la aplicación (opcional).
9. El usuario pulsa sobre el botón “Next” (opcional).
10. Si no hay ningún error, se ejecuta el proceso de importación.

Postcondición Existen nuevos objetos digitales (ítem, colecciones, ficheros, etc.) en la base de datos.

Excepciones

- No se ha introducido el campo identificador (mensaje).
- El formato del fichero no es compatible (mensaje).
- El tamaño del fichero excede el límite impuesto por el servidor (mensaje).

Importancia Alta

Tabla B.39: CU-39 Ejecutar importación CSV.

CU-40	Listar importaciones CSV
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RRFAA-6.2, RFAAA-6.2.1
Descripción	Permite al usuario ver una lista de todas las importaciones CSV ejecutadas.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pestaña “Status” del complemento “CSV Import+”. 2. Se listan en una tabla las importaciones CSV ejecutadas indicando su: <ol style="list-style-type: none"> a. Fecha de importación. b. Nombre del fichero. c. Número de filas contadas e ignoradas. d. Número de objetos importados/actualizados/ignorados. e. Estado. 3. Por cada importación se da la opción de deshacer dicha importación.

Postcondición El número de importaciones listado es igual al número de importaciones almacenado en la base de datos.

Excepciones ■ Error al cargar las importaciones (mensaje).

Importancia Alta

Tabla B.40: CU-40 Listar importaciones CSV.

CU-41	Deshacer importación CSV
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RRFAA-6.2.1
Descripción	Permite al usuario deshacer una importación CSV ya ejecutada.
Precondición	Servidor y base de datos disponible.
	La importación CSV a deshacer existe.

CU-41 Deshacer importación CSV

Acciones

1. El usuario pulsa sobre el botón “Undo Import” de la importación CSV que pretende deshacer.
2. Se muestra un mensaje de confirmación antes de ejecutar la acción.
3. El usuario pulsa sobre el botón “Ok”.
4. Se eliminan todos los objetos asociados a esa importación.
5. Opcionalmente el usuario puede eliminar el registro de la importación deshecha pulsando sobre el botón “Clear History”.

Postcondición No existe ningún objeto en la base de datos asociado a la importación.

Excepciones ■ Error al deshacer la importación (mensaje).

Importancia Alta

Tabla B.41: CU-41 Deshacer importación CSV.

CU-42 Recolección de metadatos (OAI-PMH)

Versión

1.0

Autor

Gonzalo Cuesta Marín

Requisitos

RRFAA-7, RRFAA-7.1, RFAA-7.2, RFAA-7.3, RFAA-7.4, RFAA-7.4.1

asociados**Descripción**

Permite al usuario recolectar metadatos de otros repositorios de datos.

Precondición

Servidor y base de datos disponible.

Sesión iniciada en el área de administración.

Acciones

1. El usuario pulsa sobre la entrada “OAI-PMH Harvester” del submenú “Import Tools” del menú principal.
2. Se muestra el formulario del proveedor de datos.
3. Se listan las importaciones ejecutadas.
4. Por cada importación se da la opción de ver en detalle o actualizar.

CU-42	Recolección de metadatos (OAI-PMH)
Postcondición	El número de recolecciones listado es igual al número de recolecciones almacenado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Error al cargar recolecciones (mensaje).
Importancia	Alta

Tabla B.42: CU-42 Recolección de metadatos (OAI-PMH).

CU-43	Ejecutar recolección
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RRFAA-7.1
Descripción	Permite al usuario recolectar metadatos de otro repositorio de datos.
Precondición	Servidor y base de datos disponible. Disponer de un enlace al servicio OAI-PMH de un repositorio externo de datos.
Acciones	<ol style="list-style-type: none"> 1. El usuario introduce en enlace en el campo “Base URL”. 2. El usuario pulsa sobre el botón “View sets”. 3. Se listan todos los conjuntos de datos del repositorio que se pueden recolectar. 4. El usuario selecciona el formato de los datos del conjunto que desea recolectar. 5. El usuario pulsa sobre el botón “Go” del conjunto que desea recolectar. 6. Si no hay ningún error, se comunica al usuario que la recolección se ha ejecutado satisfactoriamente.

Postcondición Existe una recolección más en la base de datos.

Excepciones	<ul style="list-style-type: none"> ▪ Error de comunicación con el repositorio externo (mensaje). ▪ URL inválida (mensaje). ▪ Repositorio externo vacío (vista especial).
--------------------	---

CU-43 Ejecutar recolección

Importancia Alta

 Tabla B.43: CU-43 Ejecutar recolección.

CU-44 Actualizar recolección

Versión 1.0

Autor Gonzalo Cuesta Marín

Requisitos asociados RRFAA-7.2

Descripción Permite al usuario actualizar una recolección existente

Precondición Servidor y base de datos disponible.

Acciones La recolección a actualizar existe.

1. El usuario pulsa sobre el botón “Re-Harvest” de la recolección a actualizar.
2. Se vuelve a ejecutar esa misma recolección.
3. Se comunica al usuario que la recolección ha sido ejecutada satisfactoriamente.

Postcondición La información de los objetos recolectados ha sido actualizada en la base de datos.

Excepciones

- Error de comunicación con el repositorio externo (mensaje).

Importancia Alta

 Tabla B.44: CU-44 Actualizar recolección.

CU-45 Listar recolecciones

Versión 1.0

Autor Gonzalo Cuesta Marín

Requisitos asociados RRFAA-7.3

Descripción Permite al usuario ver una lista de todas las recolecciones ejecutadas.

Precondición Servidor y base de datos disponible.

CU-45	Listar recolecciones
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “OAI-PMH Harvester” del submenú “Import Tools” del menú principal. 2. Se muestran las recolecciones en una tabla indicando su: <ol style="list-style-type: none"> a. Enlace del servicio. b. Prefijo del esquema de metadatos. c. Nombre del conjunto de datos recolectado. d. Fecha de recolección.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ No existen recolecciones (vista especial).
Importancia	Alta
Tabla B.45: CU-45 Listar recolecciones.	
CU-46	Ver recolecciones
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RRFAA-7.4, RFAA-7.4.1
Descripción	Permite al usuario visualizar una recolección ya ejecutada.
Precondición	<p>Servidor y base de datos disponible.</p> <p>La recolección a visualizar existe.</p>

CU-46	Ver recolecciones
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el estado de la recolección a visualizar. 2. Se obtienen los datos de la recolección de la base de datos. 3. Se muestra su información general. <ol style="list-style-type: none"> a. Especificación y nombre del conjunto de datos recolectado. b. Prefijo de metadatos escogido. c. Enlace del servicio. d. Estado actual. e. Fecha de inici y fin. f. Registros del proceso.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar la recolección (mensaje).
Importancia	Alta

Tabla B.46: CU-46 Ver recolecciones.

CU-47	Deshacer recolección
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-7.4.1
Descripción	Permite al usuario deshacer una recolección ya ejecutada.
Precondición	<p>Servidor y base de datos disponible.</p> <p>La recolección a deshacer existe.</p>

CU-47	Deshacer recolección
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el elemento “Delete Items”. 2. Se incia el proceso para eliminar los ítems recolectados. 3. Se informa al usuario que la recolección está siendo eliminada. 4. Se cambia el estado de la recolección a eliminada.
Postcondición	Los ítems asociados a la recolección no existen en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al eliminar la recolección (mensaje).
Importancia	Alta

Tabla B.47: CU-47 Deshacer recolección.

CU-48	Seguimiento <i>ARIADNEplus</i>
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-8, RFAA-8.1, RFAA-8.2, RFAA-8.3, RFAA-8.3.1
Descripción	Permite al usuario llevar un seguimiento de los procesos de integración de datos en <i>ARIADNEplus</i> .
Precondición	Servidor y base de datos disponible.
	Sesión iniciada en el área de administración.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “<i>ARIADNEplus Tracking</i>” del menú principal. 2. Se listan los tickets. 3. Por cada ticket se da la opción de administrar o eliminar. 4. Se muestra el elemento “Create a new one” para añadir un nuevo ticket.
Postcondición	El número de tickets listado es igual al número de tickets almacenado en la base de datos.

CU-48	Seguimiento <i>ARIADNEplus</i>
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar tickets (mensaje). ■ No existe ningún ticket (vista especial).
Importancia	Alta

Tabla B.48: CU-48 Seguimiento *ARIADNEplus*.

CU-49	Crear ticket de seguimiento
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-8.1
Descripción	Permite al usuario eliminar un ticket de seguimiento para el proceso de integración de una colección/ítem en <i>ARIADNEplus</i> .
Precondición	Servidor y base de datos disponible.
	El objeto (ítem/colección) sobre el que se va a llevar el seguimiento existe.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el elemento “create a new one”. 2. Se muestra al usuario un formulario guiado. 3. El usuario selecciona el tipo de objeto que se va a integrar. 4. El usuario pulsa sobre el botón “Continue”. 5. El usuario selecciona el objeto que quiere a integrar. 6. El usuario pulsa sobre el botón “Continue”. 7. Se muestra al usuario el objeto que ha seleccionado. 8. El usuario selecciona la categoría ARIADNE a la que pertenece el objeto. 9. El usuario pulsa sobre el botón “Create”. 10. Se requiere la confirmación del usuario antes de ejecutar la acción. 11. El usuario pulsa sobre el botón “Yes, create it!”. 12. Si no hay ningún error se le informa al usuario que el ticket a sido creado satisfactoriamente.
Postcondición	Existe un ticket más en la base de datos.

CU-49 Crear ticket de seguimiento

Excepciones	<ul style="list-style-type: none"> ■ Alguno de los campos requeridos están vacíos (alerta). ■ El objeto que se desea integrar ya está en otro ticket de seguimiento (mensaje). ■ Error al crear el ticket (alerta).
--------------------	--

Importancia Alta

Tabla B.49: CU-49 Crear ticket de seguimiento.

CU-50 Eliminar ticket de seguimiento

Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-8.2
Descripción	Permite al usuario eliminar un ticket de seguimiento existente.
Precondición	Servidor y base de datos disponible.
	El ticket de seguimiento a eliminar existe.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón × del ticket a eliminar. 2. Se requiere la confirmación del usuario antes de ejecutar la acción. 3. El usuario pulsa sobre el botón “Yes, delete it!”. 4. Se elimina el ticket de la vista. 5. Se informa al usuario de que el ticket a sido eliminado correctamente.
Postcondición	Existe un ticket menos en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al eliminar el ticket (alerta).

Importancia Alta

Tabla B.50: CU-50 Eliminar ticket de seguimiento.

CU-51	Administrar ticket de seguimiento
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-8.3, RFAA-8.3.1
Descripción	Permite al usuario administrar un ticket de seguimiento existente.
Precondición	Servidor y base de datos disponible.
Acciones	<p>El ticket de seguimiento a administrar existe.</p> <ol style="list-style-type: none"> 1. El usuario pulsa sobre la fila del ticket a administrar. 2. Se muestra en una barra la fase actual del ticket. 3. Se muestran las tareas de la fase actual del ticket. 4. Se muestra un botón para cambiar de fase.
Postcondición	
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar el ticket (mensaje). ■ El ticket se encuentra en progreso (vista especial). ■ El ticket es inválido (alerta).
Importancia	Alta

Tabla B.51: CU-51 Administrar ticket de seguimiento.

CU-52	Cambiar de fase
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-8.3.1
Descripción	Permite al usuario cambiar la fase de un ticket de seguimiento existente.
Precondición	Servidor y base de datos disponible.
	El ticket de seguimiento a cambiar de fase existe.

CU-52	Cambiar de fase
Acciones	<ol style="list-style-type: none"> 1. Se muestra al usuario las operaciones que debe realizar. 2. El usuario sigue las indicaciones de la fase actual. 3. Se completan todas las indicaciones de la fase actual. 4. El usuario pulsa sobre el botón “Next Phase”. 5. Si no hay ningún error, se informa al usuario que el cambio de fase ha sido un éxito.
Postcondición	La fase del ticket ha sido modificada en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Error al cambiar de fase (alerta). ▪ Algun paso interno de la fase es incorrecto (notificación).
Importancia	Alta

Tabla B.52: CU-52 Cambiar de fase.

CU-53	Edición masiva de metadatos
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos	RFAA-9
asociados	
Descripción	Permite al usuario editar metadatos de múltiples ítems a la vez.
Precondición	Servidor y base de datos disponible.
	Sesión iniciada en el área de administración.

CU-53 Edición masiva de metadatos

Acciones

1. El usuario pulsa sobre la entrada “Bulk Editor” del submenú “Edit Tools” del menú principal.
2. Se muestra el formulario para configurar el proceso de edición.
3. El usuario selecciona los ítems a editar en función de la colección a la que pertenecen.
4. El usuario selecciona los ítems a editar en función del contenido de sus metadatos (opcional).
5. El usuario pulsa sobre el botón “Preview selected items” para previsualizar los ítems seleccionados (opcional).
 - a. Se muestra una tabla con todos los ítems seleccionados indicando su título, descripción y tipo de ítem.
6. El usuario selecciona los metadatos que va a editar en los ítems seleccionados.
7. El usuario pulsa sobre el botón “Preview selected fields” para previsualizar el contenido actual de los metadatos seleccionados (opcional).
 - a. Se muestra una tabla con todos los metadatos seleccionados indicando su ítem asociado, nombre y contenido.
8. El usuario define los cambios a realizar sobre los metadatos seleccionados.
9. El usuario marca la casilla “Background Job” para ejecutar los cambios en segundo plano (opcional).
10. El usuario pulsa sobre el botón “Apply Edits Now”.
11. Se ejecutan todos los cambios definidos sobre los metadatos seleccionados de los ítems seleccionados.
12. Si no hay ningún error se informa al usuario de que la operación ha sido ejecutada.

Postcondiciones Los metadatos seleccionados de los ítems seleccionados han sido editados en la base de datos.

Excepciones

- Error al modificar un ítem (mensaje).
- Error al ejecutar el proceso de edición (mensaje).

CU-53 Edición masiva de metadatos

Importancia Alta

Tabla B.53: CU-53 Edición masiva de metadatos.

<hr/> CU-54 Gestión de complementos <hr/>	
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-10, RFAA-10.1, RFAA-10.2, RFAA-10.3
Descripción	Permite al usuario gestionar complementos que están en el interior de la aplicación.
Precondición	Servidor y base de datos disponible. Sesión iniciada en el área de administración
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Plugins” del menú global. 2. Se muestran todos los complementos existentes en la aplicación. 3. Por cada complemento no instalado se puede Instalar. 4. Por cada complemento instalado se puede Desinstalar y Configurar.
Postcondición	El número de complementos es igual al número de carpetas existentes en la carpeta <i>/plugins/</i> .
Excepciones	<ul style="list-style-type: none"> ■ Complemento inválido (mensaje). ■ Complemento no compatible (mensaje) ■ Complemento desactualizado (mensaje). ■ No existen complementos (vista especial).

Importancia Alta

Tabla B.54: CU-54 Gestión de complementos.

CU-55 Instalar complemento

Versión	1.0
Autor	Gonzalo Cuesta Marín

CU-55	Instalar complemento
Requisitos asociados	RFAA-10.1
Descripción	Permite al usuario instalar un complemento existente.
Precondición	Servidor y base de datos disponible.
Acciones	<p>El complemento a instalar existe en la aplicación.</p> <ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Install” del complemento a instalar. 2. Se muestra al usuario la página de configuración del complemento (opcional). <p>2.1. El usuario rellena el formulario de configuración. 2.2. El usuario pulsa sobre el botón “Save changes”.</p> <ol style="list-style-type: none"> 3. Se notifica al usuario que el complemento se ha instalado con éxito.
Postcondición	Existe un complemento más en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Tablas duplicadas en la base de datos (mensaje). ■ Error en la instalación (mensaje).
Importancia	Alta

Tabla B.55: CU-55 Instalar complemento.

CU-56	Configurar complemento
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-10.2
Descripción	Permite al usuario configurar un complemento instalado.
Precondición	Servidor y base de datos disponible.
	El complemento a configurar está instalado en la aplicación y es configurable.

CU-56	Configurar complemento
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Configure” del complemento a configurar. 2. Se muestra al usuario la página de configuración del complemento. 3. El usuario rellena el formulario de configuración. 4. El usuario pulsa sobre el botón “Save changes”. 5. Se notifica al usuario que el complemento ha sido actualizado con éxito.
Postcondición	Las opciones de configuración del complemento han sido actualizadas en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Campos obligatorios vacíos (mensaje). ▪ Campos inválidos (mensaje). ▪ Error al guardar la configuración (mensaje).
Importancia	Alta

Tabla B.56: CU-56 Configurar complemento.

CU-57	Desinstalar complemento
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-10.3
Descripción	Permite al usuario desinstalar un complemento instalado.
Precondición	Servidor y base de datos disponible. El complemento a desinstalar está instalado en la aplicación.

CU-57	Desinstalar complemento
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Uninstall” del complemento a desinstalar. 2. Antes de ejecutar la acción, se muestra al usuario la pantalla de confirmación. 3. El usuario marca la casilla “Yes, I want to uninstall this plugin.”. 4. El usuario pulsa sobre el botón “Uninstall”. 5. Se actualiza la vista del complemento desinstalado. 6. Se notifica al usuario que el complemento ha sido desinstalado con éxito.
Postcondición	Existe un complemento menos en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al desinstalar el complemento (mensaje). ■ Marca la casilla de confirmación (mensaje).
Importancia	Alta

Tabla B.57: CU-57 Desinstalar complemento.

CU-58	Configuración de la apariencia
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-11, RFAA-11.1, RFAA-11.2, RFAA-11.3
Descripción	Permite al usuario configurar la apariencia de la aplicación.
Precondición	<p>Servidor y base de datos disponible.</p> <p>Sesión iniciada en el área de administración.</p>

CU-58	Configuración de la apariencia
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Appearance” del menú global 2. Se muestra una pantalla con tres pestañas de configuración. <ol style="list-style-type: none"> a. <i>Themes</i> (actual): configurar temas/plantillas. b. <i>Navigation</i>: configurar la navegación. c. <i>Settings</i>: configurar parámetros de apariencia. 3. Se listan todos los temas existentes en la aplicación.
Postcondición	El número de temas es igual al número de carpetas existentes en la carpeta <i>/themes/</i> .
Excepciones	<ul style="list-style-type: none"> ■ Tema inválido (mensaje). ■ Tema no compatible (mensaje).
Importancia	Alta

Tabla B.58: CU-58 Configuración de la apariencia.

CU-59	Usar plantilla
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-11.1
Descripción	Permite al usuario instalar (usar) una plantilla/tema existente.
Precondición	<p>Servidor y base de datos disponible.</p> <p>El tema a usar existe en la aplicación.</p>

CU-59	Usar plantilla
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la pestaña “Themes”. 2. Se muestran todos los temas existentes en la aplicación. 3. El usuario pulsa sobre el botón “Use this theme” del tema a usar. 4. Se actualiza la vista del tema actual. 5. Se informa al usuario que el tema actual se ha cambiado satisfactoriamente.
Postcondición	Las vistas del área pública han sido actualizadas.
Excepciones	<ul style="list-style-type: none"> ▪ Error al usar el tema (mensaje).
Importancia	Alta

Tabla B.59: CU-59 Usar plantilla.

CU-60	Configurar plantilla
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-11.2
Descripción	Permite al usuario configurar una plantilla/tema en uso.
Precondición	Servidor y base de datos disponible. El tema a configurar está siendo usado por la aplicación y es configurable.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la pestaña “Themes”. 2. Se muestran todos los temas existentes en la aplicación. 3. El usuario pulsa sobre el botón “Configure theme” del tema a configurar. 4. El usuario rellena el formulario de configuración. 5. El usuario pulsa sobre el botón “Save changes”. 6. Se notifica al usuario que el tema ha sido actualizado con éxito.

CU-60	Configurar plantilla
Postcondición	Las opciones de configuración del tema se han actualizado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Campos obligatorios vacíos (mensaje). ▪ Campos inválidos (mensaje). ▪ Error al guardar el tema (mensaje).
Importancia	Alta

Tabla B.60: CU-60 Configurar plantilla.

CU-61	Configurar otros aspectos de apariencia
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-11.3
Descripción	Permite al usuario configurar parámetros relacionados con la apariencia.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la pestaña “Navigation” o “Settings”. 2. Se muestran todas las opciones de configuración. 3. El usuario rellena el formulario de configuración. 4. El usuario pulsa sobre el botón “Save changes”. 5. Se notifica al usuario que la configuración ha sido guardada con éxito.
Postcondición	Las opciones de configuración de apariencia se han actualizado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Campos obligatorios vacíos (mensaje). ▪ Campos inválidos (mensaje). ▪ Error al guardar la configuración (mensaje).
Importancia	Alta

Tabla B.61: CU-61 Configurar otros aspectos de apariencia.

CU-62	Configuración de la aplicación
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-12
Descripción	Permite al usuario configurar parámetros de la aplicación.
Precondición	Servidor y base de datos disponible. Sesión iniciada en el área de administración.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Settings” del menú global. 2. Se muestra una pantalla con tres pestañas de configuración. <ol style="list-style-type: none"> a. <i>General</i> (actual): parámetros generales b. <i>Security</i>: parámetros de seguridad. c. <i>Search</i>: parámetros de búsqueda. d. <i>Element Set</i>: esquemas de metadatos. e. <i>Item Type Elements</i>: elementos de los tipos de ítem. f. <i>API</i>: parámetros de la API. 3. El usuario pulsa sobre una de las pestañas de configuración. 4. Se actualiza el formulario de configuración. 5. El usuario rellena el formulario de configuración. 6. El usuario pulsa sobre el botón “Save changes”. 7. Se notifica al usuario que la configuración ha sido actualizada con éxito.
Postcondición	Las opciones de configuración de la aplicación se han actualizado en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Tema inválido (mensaje). ■ Tema no compatible (mensaje).
Importancia	Alta

Tabla B.62: CU-62 Configuración de la aplicación.

CU-63	Gestión de usuarios
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-13, RFAA-13.1, RFAA-13.2, RFAA-13.3, RFAA-13.4
Descripción	Permite al usuario gestionar complementos que están en el interior de la aplicación.
Precondición	Servidor y base de datos disponible.
	Sesión iniciada en el área de administración
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Users” del menú global. 2. Se listan todos los usuarios existentes en la aplicación. 3. Por cada usuario se puede editar o eliminar. 4. Se muestra un botón para añadir usuarios. 5. Se muestra un buscador de usuarios.
Postcondición	El número de usuarios mostrado es igual al número de usuarios almacenados en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Complemento inválido (mensaje). ■ Complemento no compatible (mensaje) ■ Complemento desactualizado (mensaje). ■ No existen complementos (vista especial).
Importancia	Alta

Tabla B.63: CU-63 Gestión de usuarios.

CU-64	Añadir usuario
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-13.1
Descripción	Permite al usuario registrar un nuevo usuario.
Precondición	Servidor y base de datos disponible.

CU-64	Añadir usuario
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “Add a User”. 2. Se muestra al usuario el formulario de registro. 3. El usuario rellena el formulario con un nombre de usuario, nombre público, <i>email</i> y <i>role</i>. 4. El usuario pulsa sobre el botón “Add User”. 5. Se notifica al usuario que el usuario ha sido añadido con éxito.
Postcondición	Existe un usuario más en la base de datos.
Excepciones	<ul style="list-style-type: none"> ▪ Campos inválidos (mensaje). ▪ Campos obligatorios vacíos (mensaje). ▪ Error al crear usuario (mensaje).
Importancia	Alta

Tabla B.64: CU-64 Añadir usuario.

CU-65	Editar usuario
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-13.2
Descripción	Permite al usuario editar un usuario existente.
Precondición	<p>Servidor y base de datos disponible.</p> <p>El usuario a editar existe.</p>

CU-65	Editar usuario
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el texto “Edit” del usuario a editar. 2. Se muestra una pantalla con tres pestañas de configuración: <ol style="list-style-type: none"> a. General: campos generales del usuario. b. Change password: campos para cambiar la contraseña. c. API Keys: campo para establecer una clave para la API. 3. El usuario rellena los campos oportunos. 4. El usuario pulsa sobre el botón “Save changes”. 5. Se notifica al usuario que la información del usuario ha sido actualizada con éxito.
Postcondición	La información del usuario ha sido actualizada en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Campos obligatorios vacíos (mensaje). ■ Campos inválidos (mensaje). ■ Error al guardar el usuario (mensaje).
Importancia	Alta

Tabla B.65: CU-65 Editar usuario.

CU-66	Eliminar usuario
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-13.3
Descripción	Permite al usuario eliminar un usuario existente.
Precondición	Servidor y base de datos disponible.
	El usuario a eliminar existe.

CU-66	Eliminar usuario
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el texto “Delete” del usuario a eliminar. 2. Antes de ejecutar la acción, se muestra una alerta de confirmación. 3. El usuario pulsa sobre el botón “Delete”. 4. Se elimina al usuario de la vista. 5. Se informa al usuario de que la acción se ha completado con éxito.
Postcondición	Existe un usuario menos en la base de datos.
Excepciones	<ul style="list-style-type: none"> ■ Error al eliminar el usuario (mensaje).
Importancia	Alta

Tabla B.66: CU-66 Eliminar usuario.

CU-67	Listar usuarios
Versión	1.0
Autor	Gonzalo Cuesta Marín
Requisitos asociados	RFAA-13.4
Descripción	Permite al usuario ver una lista de todos los usuarios existentes.
Precondición	Servidor y base de datos disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa sobre la entrada “Users” del menú global. 2. Se muestran todos los usuarios dentro de una tabla. 3. Por cada usuario se muestra su nombre de usuario, nombre público, <i>email</i> y <i>role</i>. 4. Opcionalmente se pueden ordenar en función de alguno de sus campos. 5. Opcionalmente se pueden buscar por su nombre de usuario, nombre público o <i>email</i>.
Postcondición	El número de usuarios listado es igual al número de usuarios almacenados en la base de datos.

CU-67	Listar usuarios
Excepciones	
Importancia	Alta

Tabla B.67: CU-67 Listar usuarios.

Apéndice C

Especificación de diseño

C.1. Introducción

Este anexo recoge todos los aspectos relacionados con el diseño de la aplicación, incluyendo aquellos que forman parte de los complementos desarrollados. Con ello se pretende conocer en detalle tanto su funcionamiento como su estructura.

C.2. Diseño de datos

Para la gestión de datos, la aplicación hace uso de las siguientes entidades:

- **Ítem (Item)**: es la materia prima de la aplicación. Agrupa toda la información de cada uno de los datos existentes en la plataforma. La mayor parte de entidades están relacionadas con esta entidad.
- **Conjunto de elementos (ElementSet)**: representa un esquema de metadatos.
- **Elemento (Element)**: representa un elemento (metadato) de un esquema de metadatos. Está relacionado con un conjunto de elementos (*ElementSet*).
- **Texto del elemento (ElementText)**: representa el contenido de un elemento para una determinada entidad (e.g. ítem).
- **Colección (Collection)**: representa una agrupación de ítems.
- **Etiqueta (Tag)**: representa una etiqueta con texto simple.
- **Registros etiqueta (RecordsTag)**: relaciona una etiqueta con uno o varios ítems.
- **Tipo de ítem (ItemType)**: permite asignar un tipo a un ítem y añadirle a este elementos (metadatos) específicos del tipo.

- **Elementos de tipos de ítem** (*ItemTypesElements*): representa la relación entre un elemento (metadato) y un tipo de ítem.
 - **Fichero** (*File*): representa un fichero asignado a un ítem.
 - **Fichero de colección** (*CollectionFile*): representa un fichero asignado a una colección.
 - **Localización** (*Location*): representa la localización de un ítem.
 - **Localización rectangular** (*BoxLocation*): representa la localización rectangular de un ítem.
 - **Ticket de seguimiento ARIADNEplus** (*AriadnePlusTrackingTicket*): representa el *ticket* de seguimiento utilizado para el proceso de integración de un ítem o colección.
 - **Entrada de registro ARIADNEplus** (*AriadnePlusLogEntry*): recoge todos los registros acontecidos durante un cambio de fase.
 - **Mensaje de registros ARIADNEplus** (*AriadnePlusLogMsg*): recoge el contenido de un registro.
 - **Usuario** (*User*): representa un usuario. registro.

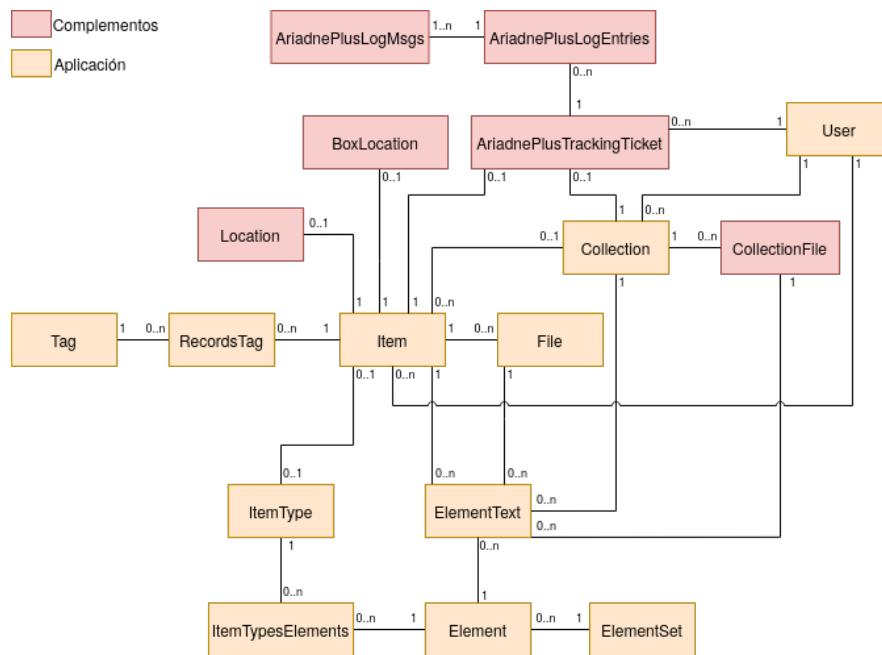


Figura C.1: Diagrama E/R de datos relacionados con la gestión de datos.

Se han excluido aquellas entidades que forman parte de complementos utilizados pero no desarrollados en este trabajo.

C.3. Diseño arquitectónico

Como es lógico, el diseño de los complementos (*plugins*) desarrollados se ha visto condicionado por el diseño de la aplicación para la que iban dirigidos.

Modelo-Vista-Controlador (MVC)

La aplicación propuesta usa el patrón de diseño *modelo-vista-controlador (MVC)*, el cual ofrece una serie de consejos para organizar correctamente el código desarrollado y facilitar así su mantenimiento.

Las aplicaciones que siguen este patrón contienen clases que implementan la lógica de negocio (*modelos*), ficheros con código HTML y PHP (*vistas*), y clases que interactúan con los usuarios (*controladores*). En las vistas no se ha utilizado el término *clase* para referirnos a estas ya que, por lo general, son documentos simples que contienen fragmentos de código HTML y PHP.

Por tanto, podemos deducir que el diseño se divide en tres capas:

- **Modelo:** modifica, gestiona y actualiza los datos de la aplicación. En nuestro caso, al contar con una única base de datos, es la capa donde se encuentra el código relacionado con las consultas, búsquedas, filtros y actualizaciones.
- **Vista:** muestra al usuario final la interfaz gráfica de la aplicación, es decir, las páginas, ventanas, formularios, etc. En términos de programación se correspondería con el *frontend*. En la aplicación se correspondería con la interfaz pública y de administración.
- **Controlador:** gestiona, atiende y procesa las peticiones realizadas por parte de los usuarios. A través de esta capa se comunican el *modelo* y la *vista*. El *controlador* solicita los datos necesarios al *modelo*, se manipulan acorde a la petición del usuario, y se entregan a la *vista* de forma que el usuario pueda visualizar los resultados esperados.

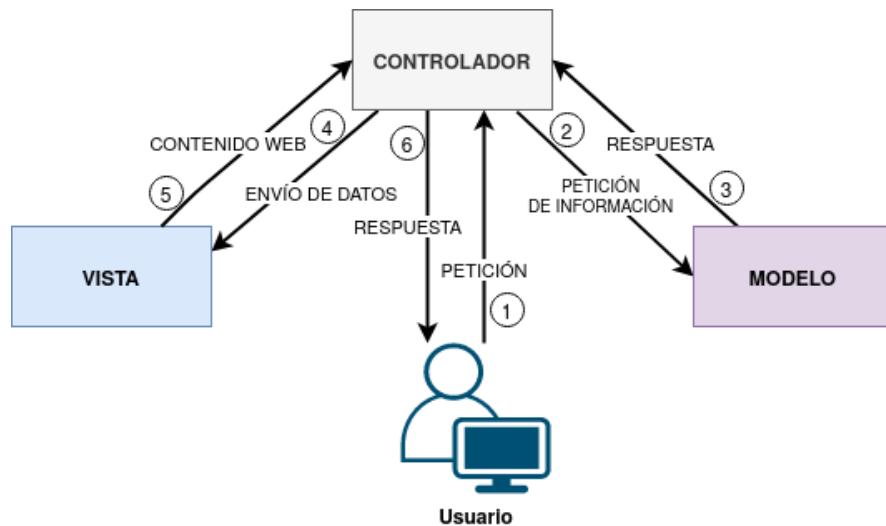


Figura C.2: Diagrama que muestra la relación entre el Modelo, Vista y Controlador del patrón MVC

En el siguiente diagrama se muestra el comportamiento de la aplicación ante una *petición HTTP*.

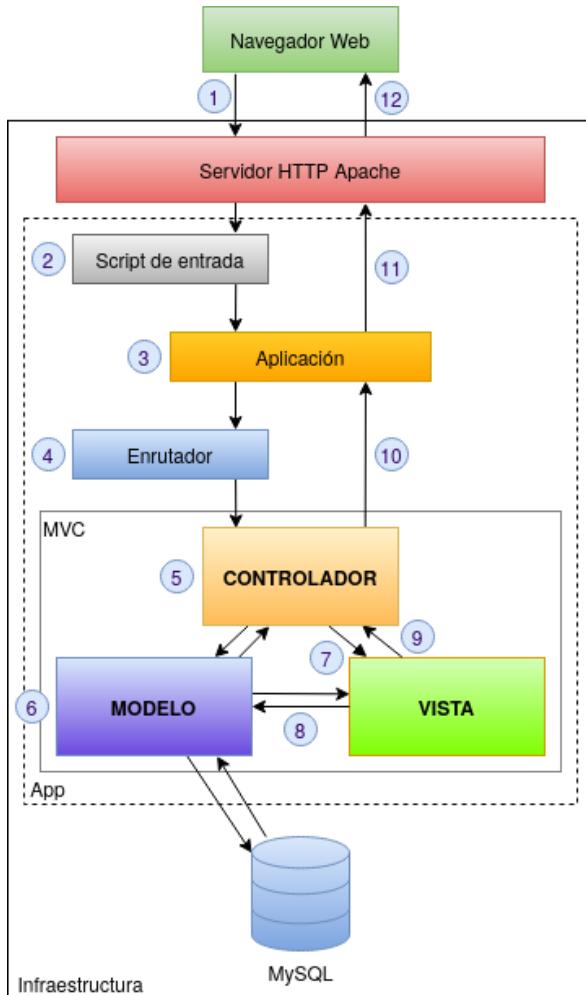


Figura C.3: Diagrama que muestra el comportamiento de la aplicación ante una petición HTTP.

1. El usuario entra a la aplicación a través de su navegador web con la dirección de la aplicación (e.g. `http://miaplicación.es`).
2. El servidor web con ayuda de la extensión PHP ejecuta el script de entrada (`index.php`).
3. Se crea una instancia de la aplicación (*Application*).
4. La aplicación usa el componente enrutador (*router*) para analizar la URL con la que se ha accedido y determinar qué controlador procesará la petición. Si la ruta existe, se instancia al *controlador* y se llama a la acción involucrada.
5. El método de la *acción* recupera los parámetros de las variables globales (e.g. *GET*, *POST*, *FILES*, etc.) y los procesa haciendo uso de los métodos

de las clases *modelo*.

6. Las clases *modelo* recogen los datos provistos por el controlador y llevan a cabo las tareas oportunas (e.g recuperar, añadir, eliminar o modificar datos de la base de datos).
7. Después de llamar a los *modelos*, se pasa a la *vista* correspondiente para renderizar la página HTML.
8. La *vista* podría, en caso de necesitarlo, consultar datos del *modelo* para la renderización.
9. La *vista* produce la salida HTML.
10. El *controlador* envía los datos a la instancia de la *aplicación*.
11. Se envía la respuesta HTTP al *servidor web*.
12. La respuesta HTTP es enviada al navegador del *cliente* (usuario).

Diseño de paquetes

Antes de mostrar cómo se encuentran organizados los complementos (*plugins*) que se han desarrollado, se va a realizar un estudio de cómo lo están los paquetes principales de la aplicación.

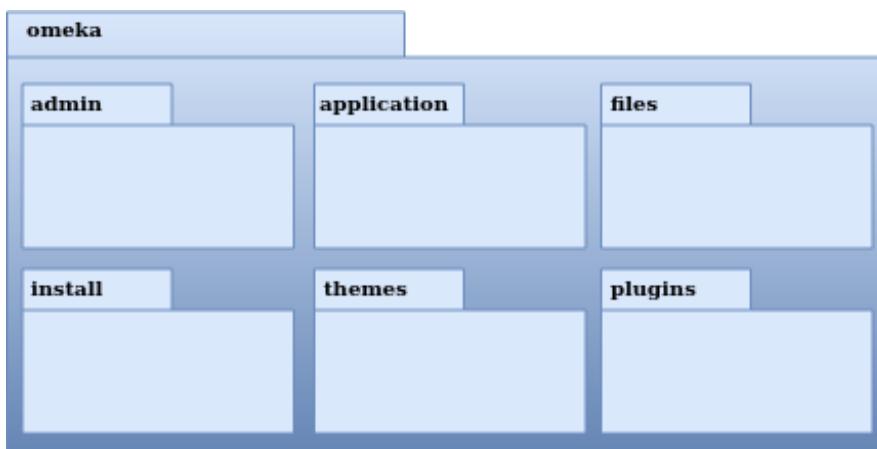


Figura C.4: Diagrama de paquetes de la aplicación.

- *omeka.admin* : contiene todos los ficheros de cada una de las *vistas* del área de administración.
- *omeka.application*: contiene la aplicación. Alberga todo el sistema *MVC*, así como las configuraciones y servicios utilizados.
- *omeka.files*: recoge todos los ficheros almacenados en la plataforma.
- *omeka.install*: contiene los ficheros de instalación inicial, necesarios para inicializar los parámetros principales de la aplicación.

- *omeka.themes*: recoge las plantillas de diseño (*themes*) utilizadas para personalizar el área pública (*frontend*) de la aplicación.
- *omeka.plugins*: contiene todos los complementos (*plugins*) utilizados para añadir nuevas funcionalidades a la aplicación.

De todos estos paquetes únicamente se especificará en detalle el paquete *plugins* por el hecho de que sólo se ha trabajado en la creación, modificación e instalación de complementos (*plugins*).

Complementos (*plugins*)

Para obtener una visión más clara de cómo están organizados los complementos (*plugins*) se mostrará su estructura de directorios general.

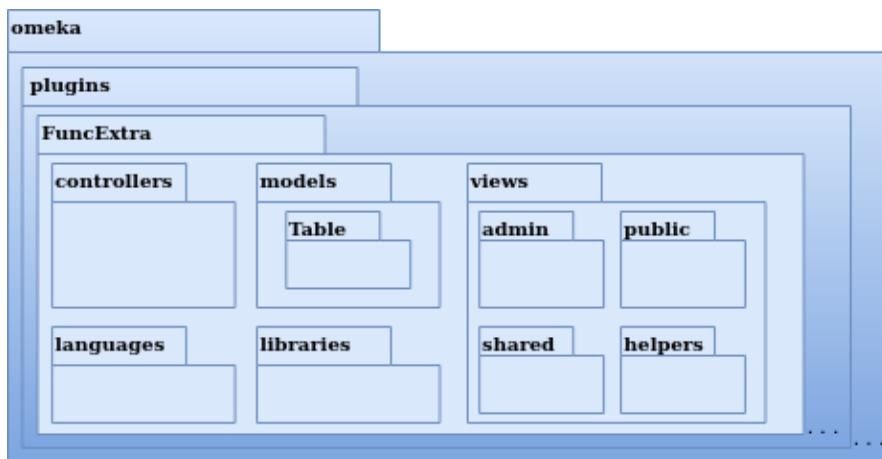


Figura C.5: Diagrama de paquetes del complemento ficticio *FuncExtra*.

- *omeka.plugins.FuncExtra*: representa el nivel superior del complemento. Alberga todo el sistema *MVC* del complemento.
- *omeka.plugins.FuncExtra.controllers*: contiene todas las clases de la capa *controlador*.
- *omeka.plugins.FuncExtra.libraries*: contiene clases externas utilizadas por el complemento.
- *omeka.plugins.FuncExtra.languages*: contiene las traducciones del texto existente en el complemento.
- *omeka.plugins.FuncExtra.models*: contiene las clases de la capa *modelo*. Permite al complemento crear y gestionar sus propias tablas en la base de datos.

- *omeka.plugins.FuncExtra.Table*: contiene parte de las clases de la capa *modelo*.
- *omeka.plugins.FuncExtra.views*: contiene los archivos (que no clases) de la capa *vista*.
 - *omeka.plugins.FuncExtra.views.admin*: contiene las *vistas* que solo son visibles en el área de administración.
 - *omeka.plugins.FuncExtra.views.public*: contiene las *vistas* que solo son visibles en el área pública.
 - *omeka.plugins.FuncExtra.views.shared*: contiene las *vistas* que son visibles en ambas áreas.

A continuación, se muestran los paquetes de todos los complementos instalados en la aplicación.

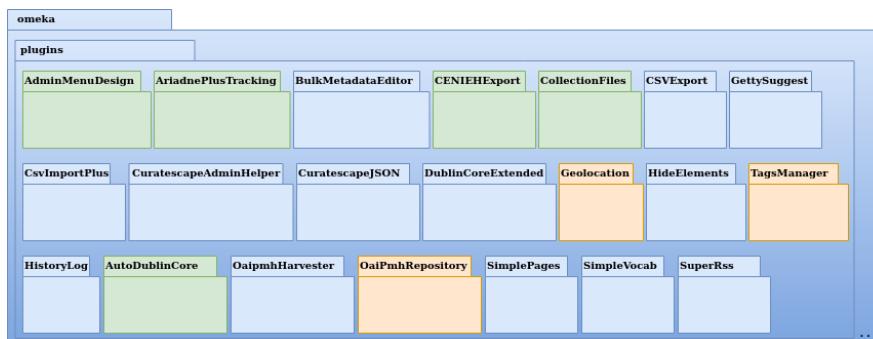


Figura C.6: Paquetes de los complementos instalados en la aplicación.

Son un total de 21 complementos, de los cuales 6 han sido creados específicamente para el proyecto (verdes) y el resto (azules) han sido recogidos de la página oficial de Omeka /citeomeka:plugins o de repositorios externos /cigit-hub:omekaplugins. De estos últimos se han modificado 3 para añadir nuevas funcionalidades (naranjas).

- *omeka.plugins.AdminMenuDesign*: permite ordenar las entradas del menú principal de navegación del área de administración en secciones (submenús).
- *omeka.plugins.AriadnePlus Tracking*: implementa todas las funcionalidades relacionadas con los tickets de seguimiento para los procesos de integración en *ARIADNEplus*.
- *omeka.plugins.BulkMetadataEditor*: permite añadir, editar o eliminar metadatos de ítems de forma masiva.
- *omeka.plugins.CENIEHExport*: permite exportar ítems y colecciones en un formato compatible con *ARIADNEplus*.

- *omeka.plugins.CollectionFiles*: permite asociar ficheros a colecciones.
- *omeka.plugins.GettySuggest*: permite sugerir términos de los vocabularios *Getty* durante el relleno de un metadato.
- *omeka.plugins.CsvImportPlus*: permite importar elementos (metadatos, localizaciones, etc.) en formato CSV y gestionar las importaciones.
- *omeka.plugins.CuratescapeAdminHelper*: implementa funcionalidades que brindan ayuda a los administradores de la aplicación.
- *omeka.plugins.CuratescapeJSON*: implementa funcionalidades para la plantilla de diseño (*theme*).
- *omeka.plugins.DublinCoreExtended*: implementa nuevos elementos en el esquema de metadatos (*ElementSet*) *Dublin Core*.
- *omeka.plugins.Geolocation*: implementa diversas funcionalidades relacionadas con la geolocalización de los ítems.
- *omeka.plugins.HideElements*: permite ocultar elementos de los esquemas de metadatos (*ElementSet*) existentes en la plataforma.
- *omeka.plugins.TagsManager*: añade funcionalidades relacionadas con las etiquetas (*tags*).
- *omeka.plugins.HistoryLog*: permite llevar un registro detallado de todas las acciones (eliminar, editar, crear, etc.) ejecutadas en la plataforma.
- *omeka.plugins.AutoDublinCore*: permite automatizar el relleno de algunos elementos del esquema *Dublin Core*.
- *omeka.plugins.OaipmhHarvester*: permite recolectar metadatos de otros repositorios web y gestionar las recolecciones ejecutadas.
- *omeka.plugins.OaiPmhRepository*: permite que otros repositorios web recolecten metadatos de nuestra aplicación.
- *omeka.plugins.SimplePages*: permite añadir páginas simples como la de “About” al área pública.
- *omeka.plugins.SimpleVocab*: permite crear y gestionar vocabularios simples para elementos de un determinado esquema.
- *omeka.plugins.SuperRss*: muestra enlaces para compartir publicaciones (área pública) en redes sociales.

Diseño de clases

Cada complemento puede contar con las siguientes clases, de las cuales sólo la primera es de uso obligatorio.

- *FuncExtraPlugin*: representa la clase principal del complemento *FuncExtra*. Permite definir las llamadas a “hooks” y “filters” y establecer las opciones de configuración del complemento.
- *FuncExtraRecord*: implementa la capa *modelo* del complemento *FuncExtra*. Cada complemento puede implementar varios *modelos* o ninguno.

- *Table_FuncExtraRecord*: es parte de la implementación de la capa *modelo*. Sobre ella se implementan métodos para hacer búsquedas sobre la base de datos y obtener como resultado objetos de la clase *FuncExtraRecord*.
- *FuncExtra_IndexController*: implementa la capa *controlador* del complemento *FuncExtra*. En este caso, implementaría el *controlador index*. Cada complemento puede implementar varios *controladores* o ninguno.
- *FuncExtraHelper_View_Helper_Extra*: implementa el ayudante *Extra*. Este provee a las *vistas* del complemento *FuncExtra* métodos para llevar a cabo funciones complejas como, por ejemplo, añadir elementos a un formulario. Es una clase auxiliar.

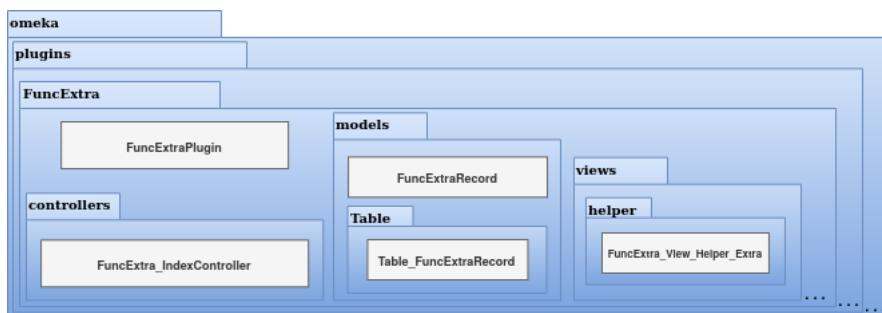


Figura C.7: Paquete tipo del complemento ficticio *FuncExtra*.

Como se puede apreciar, el nombre de cada clase varía en función del complemento al que pertenece y, en el caso de los *modelos* y *controladores*, hay que considerar además el nombre del *modelo* o *controlador* que se está implementando. Adoptando estas medidas, se evitan posibles conflictos en la nomenclatura de las clases.

En el siguiente diagrama se muestra la interacción entre los componentes del complemento ficticio *FuncExtra* y la aplicación principal.

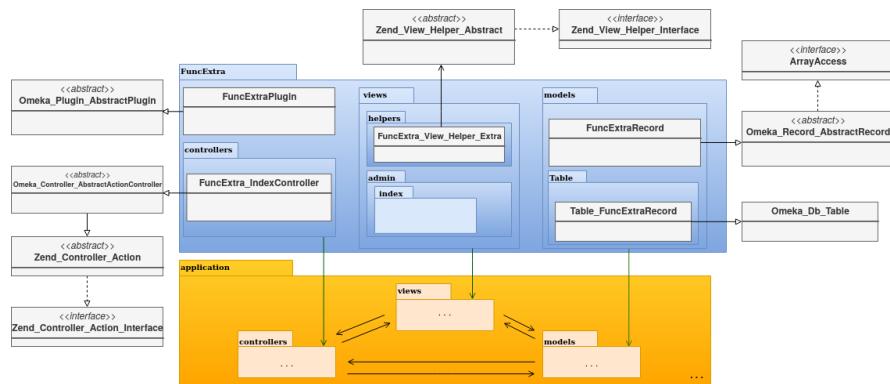


Figura C.8: Diagrama de clases del complemento ficticio *FuncExtra*.

Vemos como las implementaciones de las tres capas del complemento *FuncExtra* (*models*, *views* y *controllers*) se acoplan a las capas de la aplicación principal para después interactuar entre ellas junto a todas las demás implementaciones de la aplicación, incluyendo las de los otros complementos instalados. Este acoplamiento hace posible que desde nuestro complemento se puedan reutilizar implementaciones tanto de la propia aplicación como de los otros complementos.

Además de estas clases, se pueden añadir clases externas dentro del paquete *libraries*.

El paquete *views* no tiene clases por el hecho de que las *vistas* no son consideradas como clases en el patrón *MVC*, sino una mezcla de código HTML y PHP.

Todos los complementos que se han instalado en la plataforma siguen esta estructura, sin embargo, al ser todos los componentes opcionales (salvo la clase principal), existen ciertas diferencias entre ellos.

A continuación, por motivos de brevedad, se mostrarán únicamente los diagramas de clase de los seis complementos que se han desarrollado de forma exclusiva para el proyecto. Aquellos que contengan paquetes nuevos se explicará su significado.

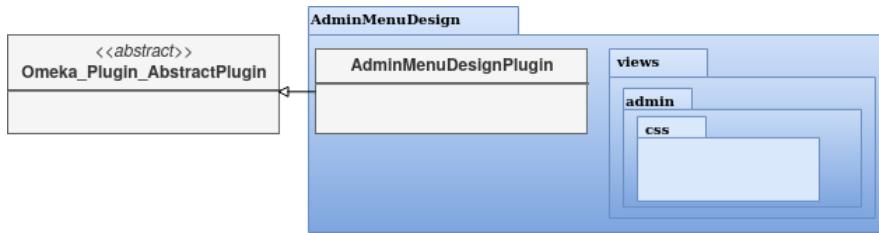


Figura C.9: Diagrama de clases del complemento *AdminMenuDesign*.

En el complemento *AdminMenuDesign* se hace uso de un paquete nuevo:

- *omeka.plugins.AriadnePlusTracking.views.css*: almacena las hojas de estilo *CSS* utilizadas por las *vistas* del complemento.

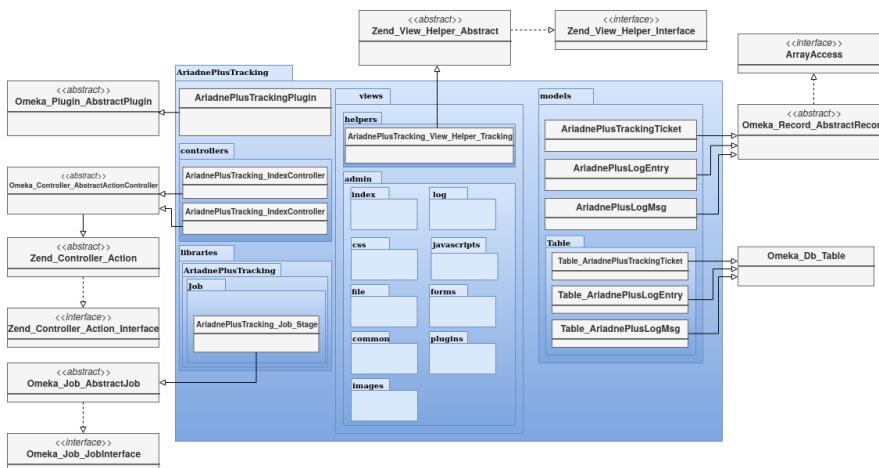


Figura C.10: Diagrama de clases del complemento *AriadnePlus Tracking*.

En el complemento *AriadnePlus Tracking* se utilizan varios paquetes nuevos:

- *omeka.plugins.AriadnePlusTracking.libraries.AriadnePlusTracking*: librería que implementa una nueva funcionalidad que permite ejecutar en segundo plano el proceso de cambio de fase del ticket.
- *omeka.plugins.AriadnePlusTracking.views.javascripts*: facilita el uso de *JavaScipt* dentro de las vistas del complemento.
- *omeka.plugins.AriadnePlusTracking.views.file*: implementa la carga de ficheros. En este caso se utiliza para el campo “JSON file of your matchings to Getty AAT” del esquema Monitor.

- *omeka.plugins.AriadnePlusTracking.views.forms*: implementa los formularios de las *vistas*.
- *omeka.plugins.AriadnePlusTracking.views.common*: implementa funcionalidades que se usan en varias *vistas*.
- *omeka.plugins.AriadnePlusTracking.views.plugins*: implementa la página de configuración del complemento.
- *omeka.plugins.AriadnePlusTracking.views.images*: facilita el uso de imágenes dentro de las *vistas* del complemento.

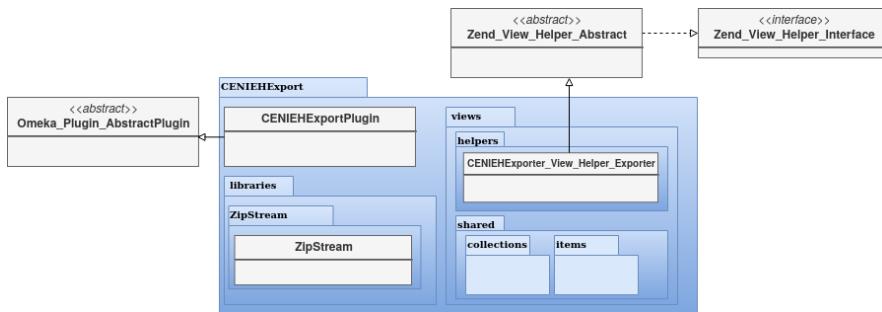


Figura C.11: Diagrama de clases del complemento *CENIEHExport*.

En el complemento ***CENIEHExport*** se hace uso de una nueva librería:

- *ZipStream*: librería que permite comprimir varios ficheros (.xml) en formato .zip de forma dinámica, sin tener que almacenar ningún fichero en el servidor.

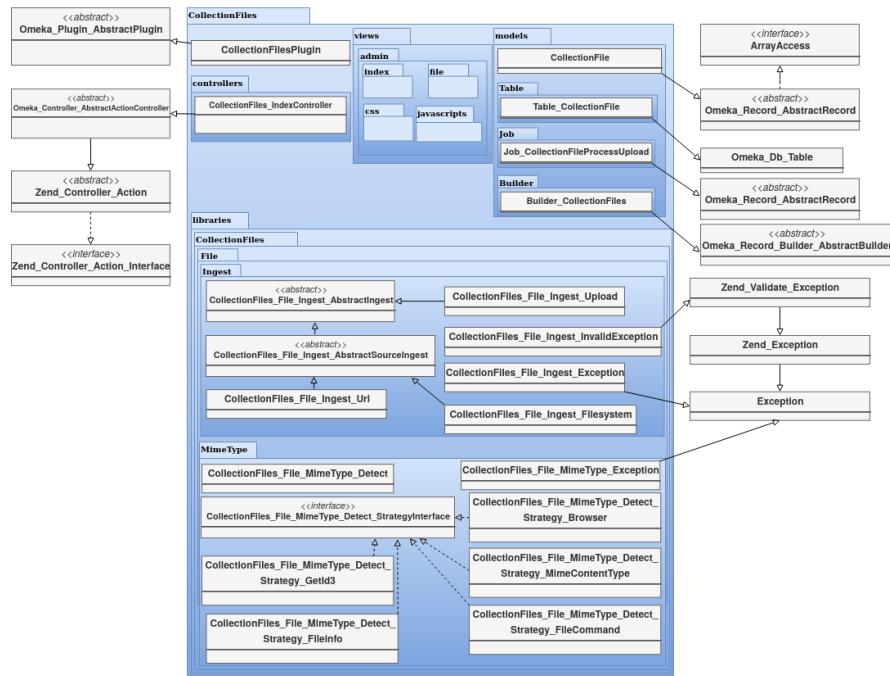


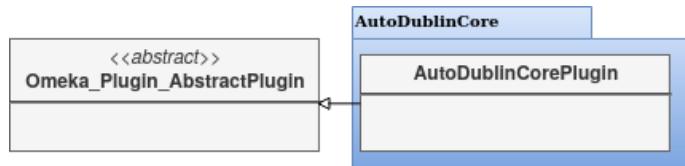
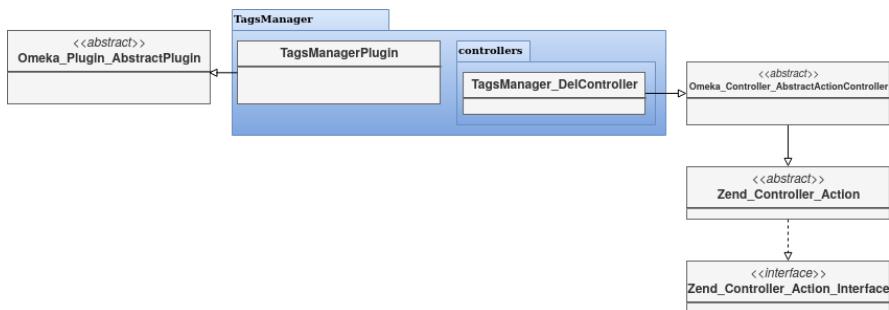
Figura C.12: Diagrama de clases del complemento *CollectionFiles*.

En el complemento ***CollectionFiles*** se utiliza una nueva librería:

- *CollectionFiles*: librería que implementa todas las funcionalidades que permiten asociar ficheros a colecciones.

Además, se utilizan dos paquetes nuevos:

- *omeka.plugins.CollectionFiles.models.Builder*: paquete utilizado para implementar *builders*. En este caso, implementa el *builder* para el objeto *CollectionFile*.
- *omeka.plugins.CollectionFiles.models.Job*: paquete utilizado para implementar *jobs*. En este caso, el *job* implementado procesa la carga de ficheros.

Figura C.13: Diagrama de clases del complemento *AutoDublinCore*.Figura C.14: Diagrama de clases del complemento *TagsManager*.

C.4. Diseño procedimental

En este apartado se muestra cómo interactúan los principales componentes de la aplicación ante un determinado evento.

En el diagrama de secuencia que se expone a continuación, se describe el funcionamiento interno de la aplicación ante una situación general donde el usuario accede a la aplicación para llevar a cabo una determinada acción.

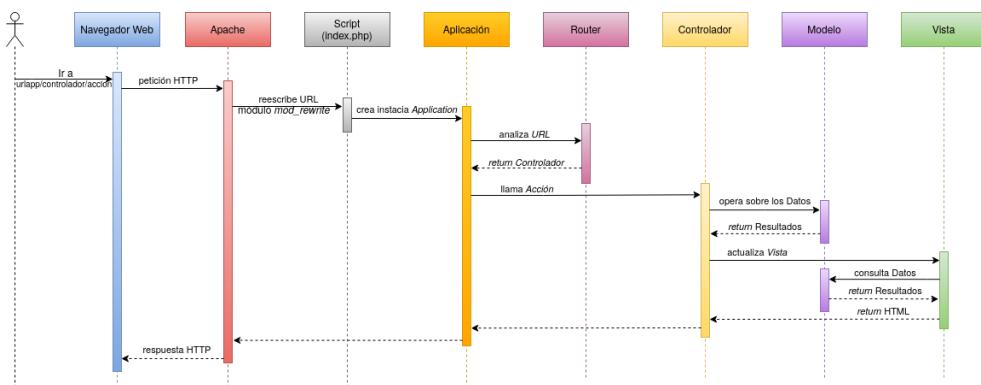


Figura C.15: Diagrama de secuencia para un caso general.

En este caso se presupone que tanto el *controlador* como la *acción* indicada por el usuario son válidas. En caso contrario, se enviarían las excepciones correspondientes.

C.5. Diseño de interfaces

Para la creación del complemento *AriadnePlus Tracking* se llevaron a cabo una serie de prototipos que sirvieron de ayuda visual en las fases posteriores de desarrollo.

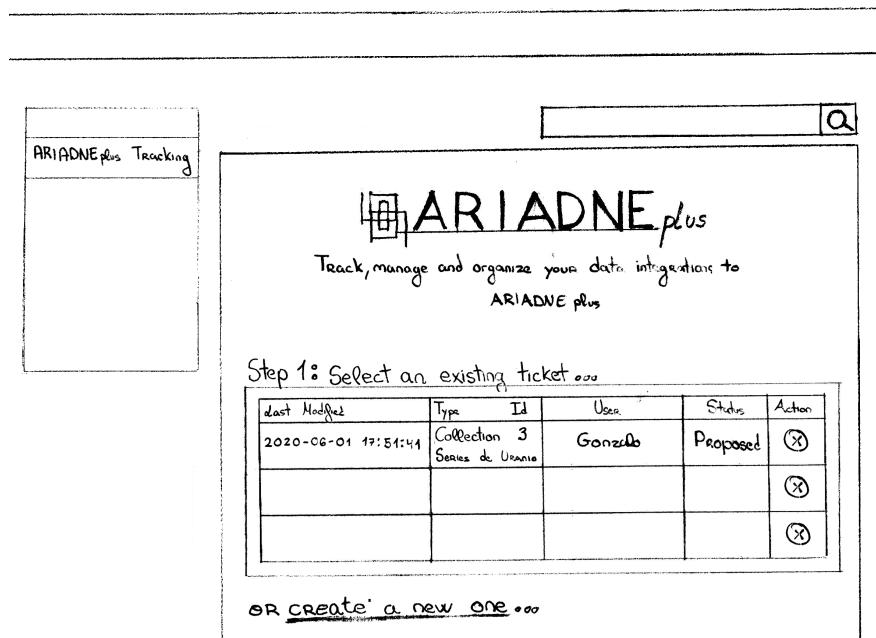
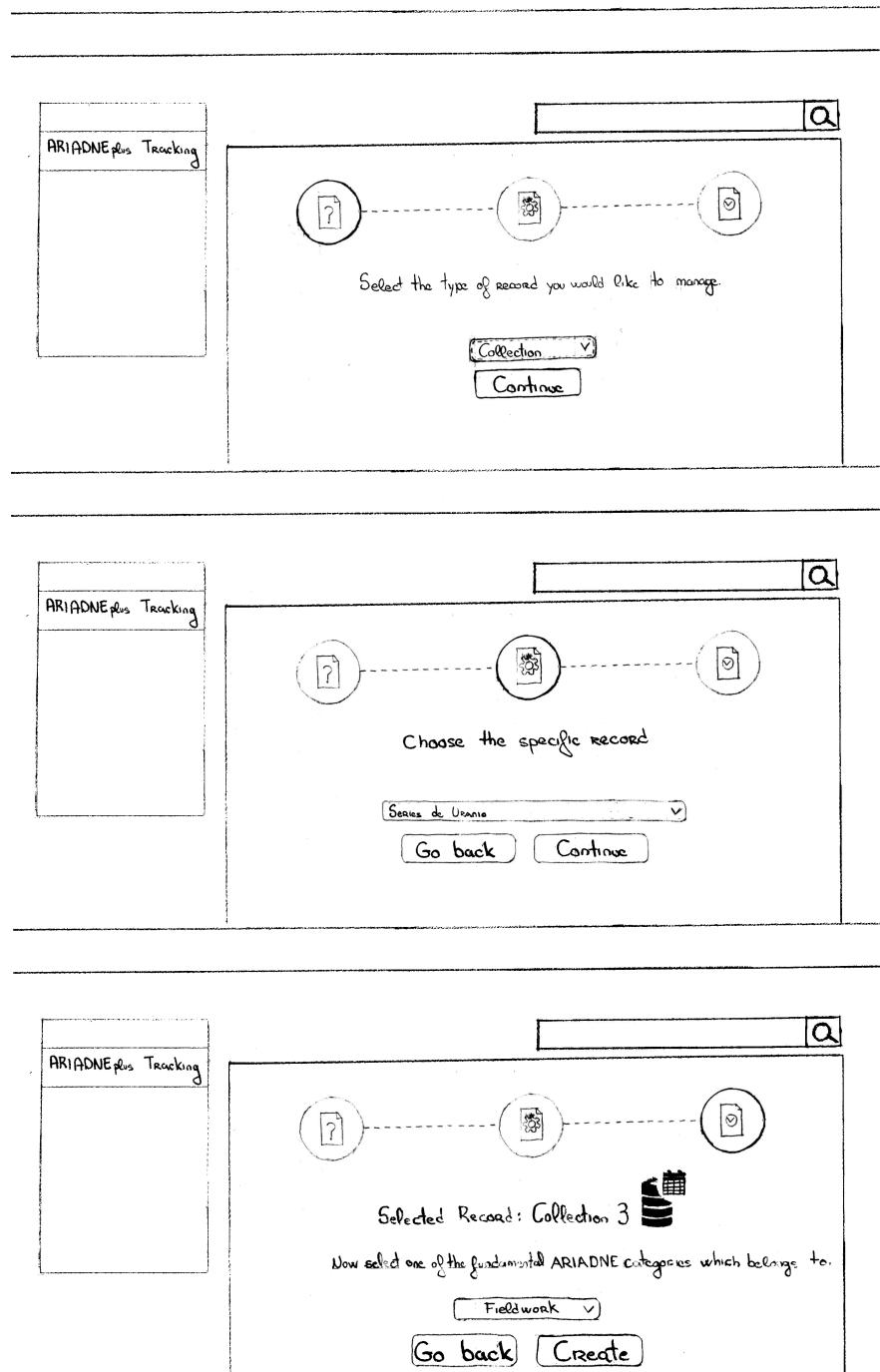


Figura C.16: Prototipos: página principal (*ARIADNEplus Tracking*)

Figura C.17: Prototipos: creación de un ticket (*ARIADNEplus Tracking*)

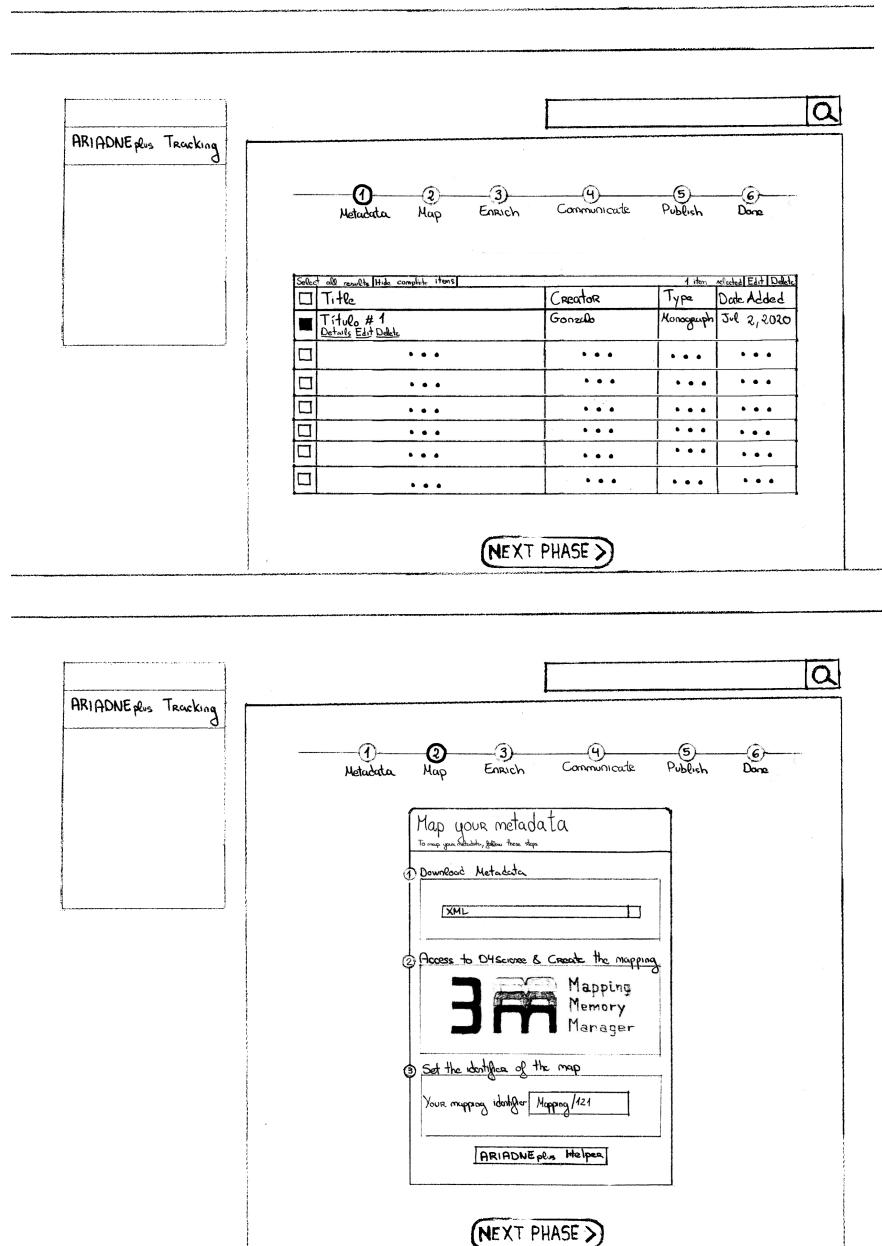


Figura C.18: Prototipos: primera y segunda fase de un ticket (*ARIADNEplus Tracking*)

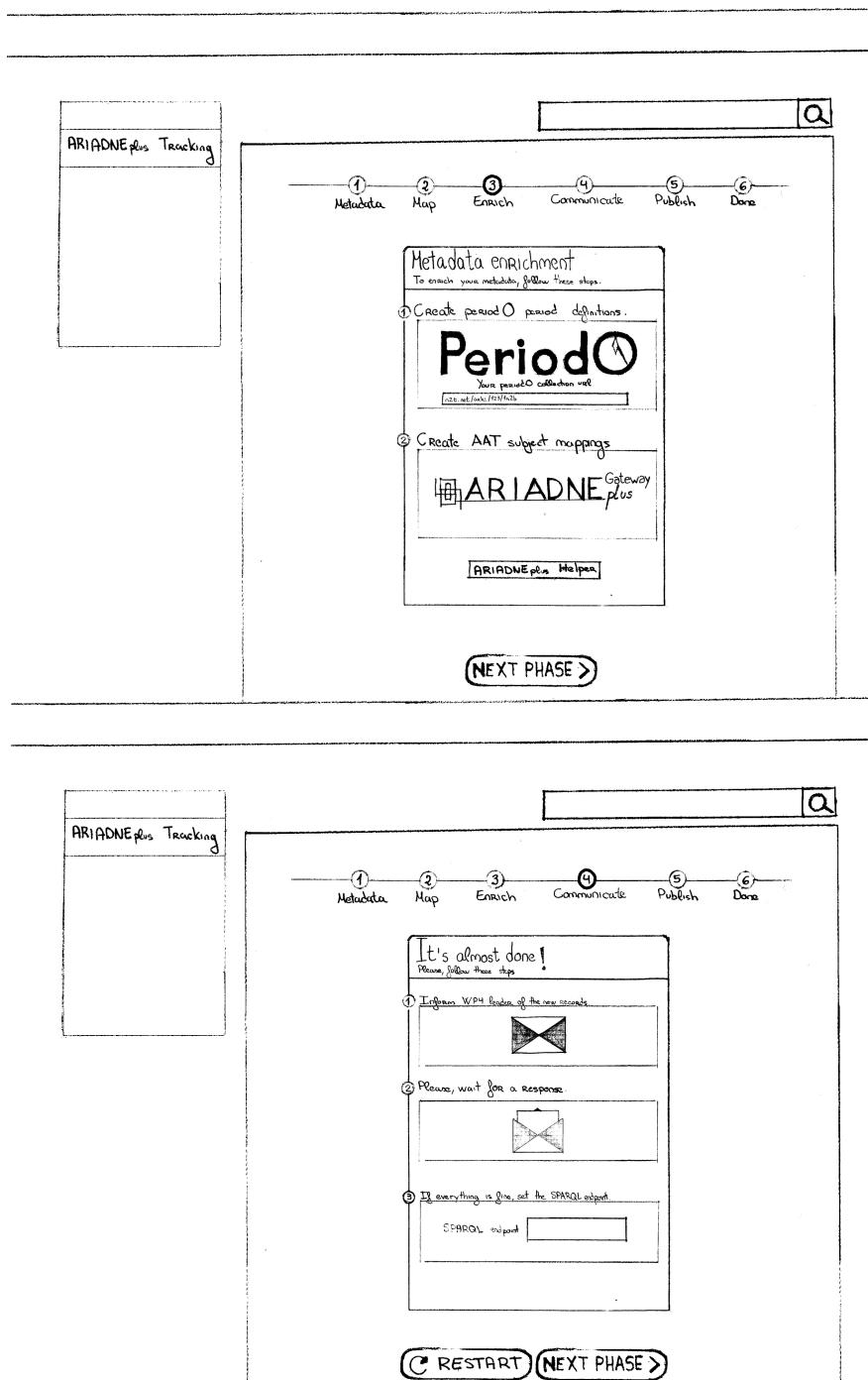


Figura C.19: Prototipos: tercera y cuarta fase de un ticket (*ARIADNEplus Tracking*)

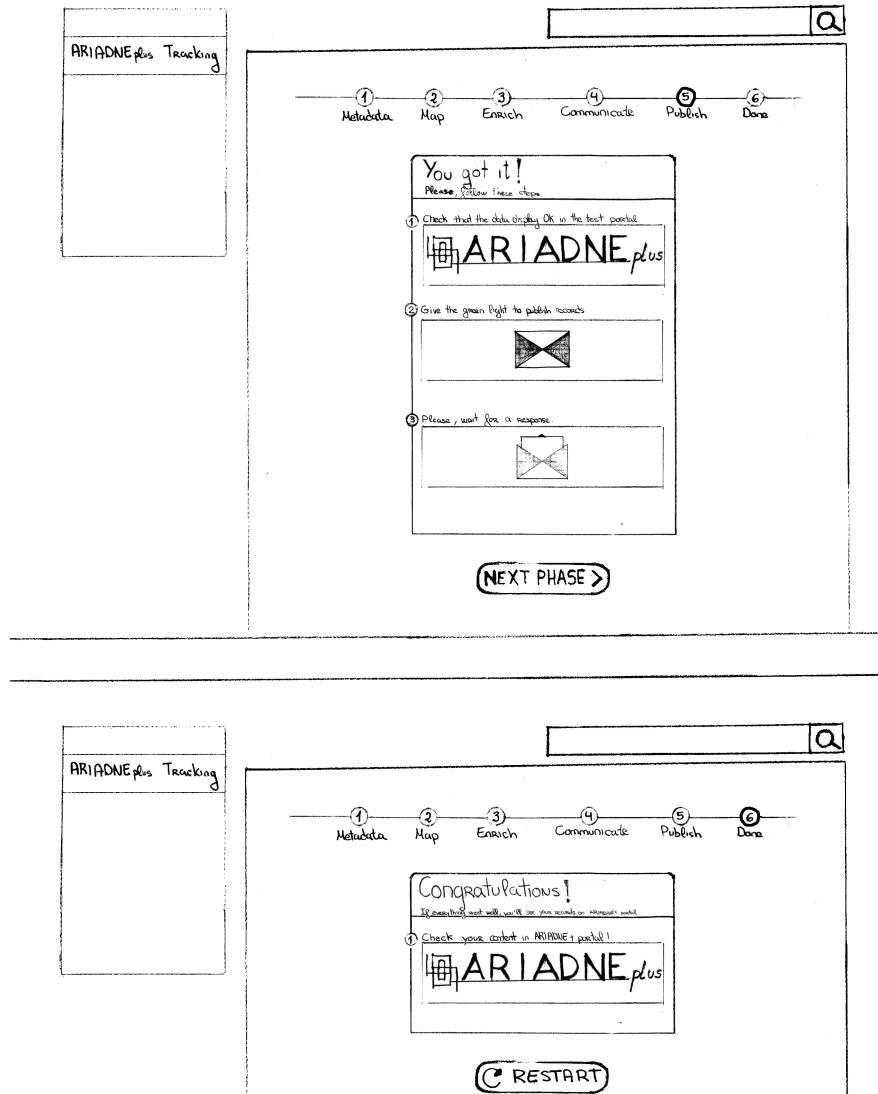


Figura C.20: Prototipos: quinta y sexta fase de un ticket (*ARIADNEplus Tracking*)

Apéndice D

Documentación técnica de programación

D.1. Introducción

Este anexo recoge la información necesaria para que cualquier usuario sea capaz de instalar, configurar y mantener la aplicación junto a todos los complementos (*plugins*) y temas (*themes*) propuestos en este proyecto.

D.2. Estructura de directorios

A continuación se listan los directorios que forman parte del repositorio del proyecto:

- /: raíz del proyecto. Esta contiene ficheros de configuración de *Docker*, *Kustomize* (*Kubernetes*) y *GitHub*, el fichero *README.md*, y una copia de la licencia del repositorio.
- *./github/workflows/*: contiene los flujos de trabajo (*workflows*) utilizados por el servicio *GitHub Actions*.
- *./configFiles/*: recoge los ficheros de configuración de la aplicación utilizados para crear las imágenes de *Docker* que contienen a la aplicación.
- *./docs/*: contiene toda la documentación del proyecto.
- *./docs/LaTeX/*: documentación en formato *LaTeX*.
- *./docs/readme/*: recoge material auxiliar (imágenes y gifs) utilizados por el *README.md*.
- *./docs/sphinx/*: documentación en formato *reStructure Text*.
- *./gke-mysql/*: contiene los ficheros de configuración *Kustomize* utilizados para desplegar en *Kubernetes* el contenedor de la base de datos.

- */gke-omeka/*: contiene los ficheros de configuración *Kustomize* utilizados para desplegar en *Kubernetes* el contenedor de la aplicación.
- */omeka/*: recoge los archivos de la aplicación.
- */omeka/plugins/*: complementos propuestos para la aplicación.
- */omeka/themes/*: tema propuesto para la aplicación.
- */omeka/themes/curatescape/*: nivel superior del tema. Contiene todas las vistas públicas, ficheros de configuración, información y personalización del tema, y la imagen de la portada del tema.

El directorio */omeka/plugins/* cuenta además con un total de 21 subdirectorios, cada uno de los cuales se corresponde con un complemento. A continuación se expone la estructura de directorios que puede adquirir cada complemento:

- */omeka/plugins/<NombreDelComplemento>/*: nivel superior del complemento. Contiene el fichero principal y el fichero de información del complemento. En caso de ser configurable, puede contener además el fichero de la página de configuración.
- */omeka/plugins/<NombreDelComplemento>/controllers/*: controladores del complemento.
- */omeka/plugins/<NombreDelComplemento>/views/*: vistas.
- */omeka/plugins/<NombreDelComplemento>/views/public/*: vistas del área pública.
- */omeka/plugins/<NombreDelComplemento>/views/admin/*: vistas del área de administración.
- */omeka/plugins/<NombreDelComplemento>/views/shared/*: vistas comunes.
- */omeka/plugins/<NombreDelComplemento>/models/*: modelos.
- */omeka/plugins/<NombreDelComplemento>/libraries/*: funcionalidades externas a la aplicación.
- */omeka/plugins/<NombreDelComplemento>/languages/*: traducciones.
- */omeka/plugins/<NombreDelComplemento>/forms/*: formularios.
- */omeka/plugins/<NombreDelComplemento>/tests/*: test unitarios.

D.3. Manual de programador

A lo largo de este apartado se van a explicar los procedimientos necesarios para acondicionar el entorno de desarrollo, instalar la aplicación junto a todos los complementos y temas propuestos, adoptar diversos servicios de integración continua y, por último, ejecutar sobre la aplicación/complementos pruebas.

Entorno de desarrollo

Antes de poder realizar cualquier tipo de operación sobre la aplicación o sobre cualquiera de los complementos propuestos, es necesario adaptar el entorno de desarrollo.

El sistema requerido por la aplicación es conocido como *LAMP* [19], el cual debe contar con las siguientes herramientas:

- **Linux** como sistema operativo.
- **Apache** como servidor web.
- **MySQL/MariaDB** como gestor de base de datos.
- **PHP** como lenguaje de programación.

Además, la aplicación necesita tener a su disposición el conjunto de herramientas *software* provisto por **ImageMagick**.

Aparte de los requerimientos propios de la aplicación, también es necesario un **editor de código** desde donde llevar a cabo las labores propias de desarrollo.

A continuación, se muestra información adicional de cada herramienta.

Linux

Cualquier sistema operativo que sea de tipo *Unix* es válido para la aplicación. En este proyecto se ha utilizado la distribución *Ubuntu 19.10*.

Servidor HTTP Apache

El servidor HTTP Apache es un servidor web de código abierto que aporta al entorno las condiciones necesarias para mostrar la aplicación sobre el navegador. En el siguiente [enlace](#) se puede encontrar una guía detallada para su instalación.

Si nuestro entorno ya cuenta con esta herramienta, para que la aplicación funcione correctamente, **se debe activar el módulo *mod_rewrite***. Este permite que la aplicación pueda procesar *URLs* mucho menos complejas para los usuarios (*clientes*).

El proceso de activación no tiene mucha complicación. Desde cualquier terminal de Linux:

1. Activar el módulo *rewrite* y aplicar cambios reiniciando el demonio de Apache.

```
sudo a2enmod rewrite  
sudo /etc/init.d/apache2 restart
```

2. Editar el archivo de configuración del sitio asignado para la aplicación (el sitio por defecto es *000-default*).

```
sudo nano /etc/apache2/sites-enabled/000-default
```

3. Dentro de la etiqueta *Directory*, asignar el valor *All* a la directiva *AllowOverride*.

```
<VirtualHost>  
...  
    <Directory>  
        AllowOverride All  
    ...  
    </Directory>  
</VirtualHost>
```

MySQL/MariaDB

MySQL es un *software* bastante conocido para la gestión de bases de datos relacionales. La aplicación propuesta **es compatible con cualquier versión de MySQL que sea superior a la 5.0**. A través de este [enlace](#) se accede a la guía de instalación oficial.

PHP

Es fundamental que el entorno tenga instalada una **versión de PHP superior a la 5.7**. En este [enlace](#) se explica cómo hacerlo.

Además, para poder hacer uso tanto de la aplicación como de todos los complementos propuestos, **es necesario instalar y activar los siguientes módulos/extensiones:**

- *mysqli*: permite acceder a la funcionalidad proporcionada por *MySQL 4.1* y posterior.
- *exif*: permite trabajar con metadatos de imágenes.
- *curl*: permite conectarse y comunicarse con diferentes tipos de servidores y diferentes tipos de protocolos.

- *mbstring*: permite manejar codificaciones basadas en *Unicode*, tales como *UTF-8* y *UCS-2*.

Una vez instalados, se deben realizar los siguientes cambios en el fichero de configuración PHP del servidor Apache (se suele encontrar en la ruta */etc/php/<version>/apache2/*):

1. Comenzar la edición del fichero.

```
sudo nano /etc/php/7.2/apache2/php.ini
```

2. Activar las extensiones instaladas descomentando (quitar el ';') las siguientes líneas.

```
extension=curl
extension=mbstring
extension=exif
extension=mysqli
```

Recuerda que los cambios cometidos en este fichero no se aplican hasta reiniciar el servidor Apache.

ImageMagick

ImageMagick es un producto *software* que provee al entorno un conjunto de herramientas que permiten visualizar, modificar y transformar todo tipo de formatos de imagen. La aplicación propuesta requiere contar con esta *suite* instalada ya que la utiliza para procesar las imágenes internas. Los detalles de la instalación se encuentran en este [enlace](#).

Editor de código

En el proyecto se ha utilizado como editor de código **NetBeans**. Se eligió principalmente porque, además de ser uno de los editores más populares para PHP, da soporte al *framework* que utiliza la aplicación, *Zend Framework*. También ofrece funcionalidades a otros lenguajes utilizados en la aplicación como *JavaScript*, *HTML* y *CSS*. Se puede obtener de forma gratuita a través de este [enlace](#).

En su página oficial se puede encontrar un [video-tutorial](#) que explica cómo desarrollar desde *NetBeans* aplicaciones PHP que utilizan como marco de trabajo *Zend Framework*.

Instalación de la aplicación

Con el entorno de desarrollo ya preparado, podemos proceder con la instalación de la aplicación.

El primer paso consiste en **configurar el servidor**:

1. **Crear la base de datos (DB) MySQL** desde un usuario con permisos suficientes como para poder realizar operaciones sobre ella.

- Durante el proceso, conviene apuntar los siguientes datos:
 - *Hostname* donde se encuentra alojada la DB.
 - Nombre de la DB.
 - Nombre del usuario de la DB.
 - Contraseña de usuario de la DB.
- La base de datos ha de estar codificada en utf8.

```
sudo mysql -u root -
CREATE DATABASE omekadb CHARACTER SET utf8mb4 \
COLLATE utf8mb4_unicode_ci;
CREATE USER 'usuario'@'localhost' IDENTIFIED BY 'contraseña';
GRANT ALL ON omekadb.* TO 'usuario'@'localhost' \
IDENTIFIED BY 'contraseña' WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

2. **Descargar** la versión 2.7.1 de **Omeka**, desde su [web oficial](#) o desde su [repositorio oficial](#) en GitHub.

```
cd /tmp && wget \
https://github.com/omeka/Omeka/releases/download/v2.7.1/omeka-2.7.1.zip
```

3. **Descomprimir** el fichero .zip recién descargado sobre un directorio desde donde podamos trabajar.

```
unzip omeka-2.7.1.zip -d <directorio_de_trabajo>
```

4. Desde el directorio escogido, buscar el fichero db.ini y **sustituir los valores 'XXXXX' por los datos de la base de datos** (anotados en el paso 1).

```
cd <directorio_de_trabajo>
nano db.ini
```

No es necesario modificar los parámetros ‘prefix’ o ‘port’.

```
[database]
host      = "localhost"
username  = "usuario"
password  = "contraseña"
dbname    = "omekadb"
prefix    = "omeka_"
charset   = "utf8"
;port     = ""
```

5. Descargar el contenido del [repositorio del proyecto](#).

```
cd /tmp && wget \
https://github.com/gcm1001/TFG-CeniehAriadne/archive/master.zip
```

6. Descomprimir las carpetas `/omeka/plugins` y `/omeka/themes` del fichero `.zip` recién descargado.

```
unzip master.zip 'TFG-CeniehAriadne-master/omeka/plugins/*' \
'TFG-CeniehAriadne-master/omeka/themes/*' -d <directorio_de_trabajo>
```

7. Desde el directorio de trabajo, reemplazar las carpetas originales `plugins` y `themes` por las previamente descargadas.

```
cd <directorio_de_trabajo>
rm -rf ./plugins ./themes
sudo cp -r ./TFG-CeniehAriadne-master/omeka/* .
rm -rf ./TFG-CeniehAriadne-master
```

8. Mover todo el contenido del directorio de trabajo a la carpeta del servidor Apache.

```
mv -r <directorio_de_trabajo>/* <directorio_del_servidor>
```

9. Dar permisos de lectura y escritura sobre todo el contenido de la aplicación.

```
cd <directorio_del_servidor>
sudo chown -R www-data:www-data <directorio_de_trabajo>
sudo chmod -R 755 <directorio_de_trabajo>
```

Desde este instante, la aplicación será accesible desde el navegador (puerto 80).

Para finalizar con la instalación, se debe **completar el formulario de instalación** disponible en el directorio `/install` de la aplicación (e.g. `miaplicacion.es/install`).

Cuando se haya completado, la aplicación únicamente contará con la funcionalidad básica, es decir, no se verán los cambios introducidos por los complementos/temas. Para ello, es necesario instalarlos desde la interfaz. En los siguientes apartados se explicará como hacerlo.

Añadir funcionalidades a la aplicación

Una de las características que hacen de la aplicación una magnífica plataforma para el proyecto es su **escalabilidad**. Gracias a su sistema de **complementos** o *plugins*, cualquier programador tiene la posibilidad de adaptarla a sus necesidades individuales sin necesidad de modificar el código base de la aplicación.

Además, cuenta con una fabulosa comunidad de desarrolladores que hacen públicas sus implementaciones. Por tanto, antes de comenzar con el desarrollo de un nuevo *plugin*, es recomendable comprobar que la funcionalidad que se desea implementar no está ya desarrollada (ver [Repositorio de complementos oficial](#) o [Repositorio de Github](#)).

Cómo instalar complementos en la aplicación

En este apartado se muestra el procedimiento a seguir para instalar complementos en la aplicación.

Si se ha instalado la aplicación siguiendo los pasos incluídos en este manual (ver Instalación de la aplicación), los complementos que incluyen cada una de las funcionalidades desarrolladas en este proyecto se encuentran ya ubicados en el interior de la aplicación.

En el caso de que se quiera añadir algún complemento adicional a los propuestos en este proyecto, se deben trasladar antes sus ficheros al directorio */plugins/* de la aplicación.

Con los complementos ya ubicados en el interior de la aplicación, hay que hacer uso de la interfaz para completar su instalación. Los pasos a seguir son:

1. Acceder al área de administración (*/admin/*).
2. Desde el gestor de complementos (*/admin/plugins*).
3. Localizar el nombre del complemento que se desea instalar.
4. Hacer clic sobre el botón “*Install*” situado en la parte derecha del complemento.
5. En caso de que el *plugin* sea configurable, rellenar el formulario de configuración y hacer clic sobre el botón “*Save Changes*”.

Para obtener información más detallada acerca de la gestión de complementos, ver el [manual de usuario](#).

Personalizar el diseño de la aplicación

Si nuestra intención es modificar la estética de la aplicación, esta cuenta con un sistema de **temas** o plantillas que permite personalizar el área pública (*frontend*) del sitio.

Existe también la posibilidad de reutilizar temas de otros desarrolladores (ver [Repositorio de temas oficial](#) o [Repositorio de Github](#)).

Cómo instalar temas en la aplicación

Su proceso de instalación es muy similar al de los complementos. Al igual que pasaba con estos, si se han seguido los pasos de instalación (ver Instalación de la aplicación), el tema propuesto se encuentra ya almacenado en el interior de la aplicación.

En el caso de que se quiera añadir algún otro tema, se deben trasladar antes sus ficheros al directorio `/themes/` de la aplicación.

Con el tema ya almacenado en la aplicación, se puede llevar a cabo el proceso de instalación desde la interfaz.

Para instalar un tema (*theme*):

1. Acceder al área de administración (`/admin/`).
2. Desde la página de configuración de diseño (`/admin/appearance/`).
3. Hacer clic sobre la entrada “*Themes*” de la barra de navegación existente.
4. Localizar el nombre del tema que se desea instalar.
5. Hacer clic sobre el botón “*Use this theme*”.

Para obtener información más detallada acerca de la gestión de temas, ver el [manual de usuario](#).

Integración continua

El repositorio del proyecto dispone de varios mecanismos de integración continua que facilitan la ejecución de alguna de las tareas típicas de desarrollo como, por ejemplo, el despliegue de la aplicación. En los siguientes apartados se explicará como se implementan cada uno de ellos.

GitHub Actions

GitHub Actions es uno de los servicios ofrecidos por *Github* que permite crear, compartir y ejecutar código desde la misma plataforma, sin necesidad de utilizar servicios externos.

En este proyecto se ha utilizado esta herramienta para automatizar dos flujos de trabajo:

- *Workflow 1*: Despliegue de la aplicación sobre el entorno de desarrollo, incluyendo en ella los complementos/temas almacenados en el repositorio del proyecto.
- *Workflow 2*: Compilación y publicación de la imagen *Docker* utilizada por el entorno de producción para el despliegue de la aplicación y de los *plugins*/temas almacenados en el repositorio del proyecto.

A continuación se muestran las etapas por las que se ha pasado para desarrollar el primer flujo de trabajo.

Etapa 01: Montar el servidor en la nube

Para montar el servidor en la nube se ha utilizado la plataforma *Google Kubernetes Engine* (GKE) de *Google Cloud* [9].

El procedimiento a seguir es el siguiente:

1. Crear un **nuevo proyecto** en *Google Cloud*.
2. Habilitar los siguientes **servicios**: *Container Registry* y *Kubernetes Engine API* (acceder a este [enlace](#) para activarlos automáticamente).
3. Crear un **nuevo clúster** en *Google Cloud* (ver [tutorial](#)).
4. Crear una **nueva cuenta de servicio** (ver [tutorial](#)).
5. Añadir, a la cuenta recién creada, los siguientes **roles** (ver [tutorial](#)).
 - *Kubernetes Engine Developer*: nos permitirá desplegar aplicaciones en la plataforma GKE.
 - *Storage Admin*: nos permitirá publicar contenedores Docker en la plataforma Container Registry.
6. Crear una **clave** para la cuenta creada en el paso 4 (ver [tutorial](#)).

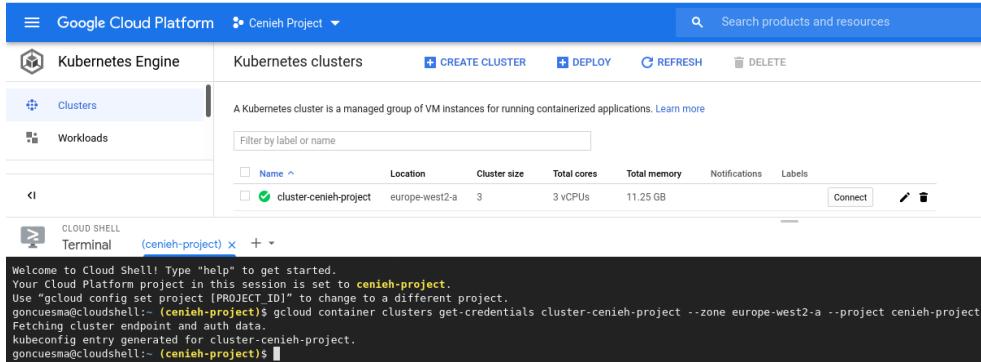


Figura D.1: Vista del panel de administración de Google Cloud tras finalizar los pasos marcados.

Etapa 02: Configuración del *workflow*

Para implementar las técnicas de integración continua a través de *Github Actions*, es necesario crear un flujo de trabajo (*workflow*) donde definir los procesos que se pretenden automatizar.

Github Actions permite definir más de un flujo de trabajo por repositorio. Estos deben ser almacenados dentro del repositorio sobre el directorio `.github/workflows`. La sintaxis que siguen estos ficheros es *YAML*, por lo que la extensión ha de ser `.yaml`.

En este proyecto, el fichero de configuración utilizado para definir el *workflow* que automatiza el despliegue de la aplicación se llama `gke.yaml`.

A continuación se explica brevemente en qué consiste cada una de las etiquetas utilizadas en este fichero:

- *name*: nombre del *workflow*.
- *on*: propiedades de activación del *workflow*.
 - *push*: se activa al realizar una operación de *push*.
 - *branches*: ramas sobre las que se activa.
 - *paths-ignore*: directorios que se ignoran.
 - *env*: variables de entorno.
 - *runs-on*: SO donde queremos ejecutar cada una de las acciones.
 - *steps*: agrupa el conjunto de acciones a ejecutar.
 - *uses*: selecciona una acción externa para ser ejecutada.
 - *with*: indica parámetros de entrada para la acción externa.

- *name*: nombra un paso/acción.
- *run*: indica los comandos a ejecutar .

Los procesos que se han definido son los siguientes:

1. *Checkout*: recoge el contenido del repositorio.
2. *Setup gcloud CLI*: prepara el entorno para tener acceso a todas las herramientas existentes en la plataforma *Google Cloud*.
3. *Gcloud: Configure Docker*: prepara la configuración para *Docker*.
4. *Gcloud: GKE credentials*: obtiene las credenciales necesarias para publicar la imagen *Docker* en nuestro repositorio privado de *Google Cloud*.
5. *Build the Docker image*: compila la imagen *Docker* que contiene la aplicación y los complementos/temas.
6. *Push the Docker image to Google Container Registry*: publica la imagen *Docker* recién compilada en nuestro repositorio privado de *Google Cloud*.
7. *Set up kustomize*: instala la herramienta *Kustomize*, necesaria para administrar los ficheros de configuración *.yaml*.
8. *Deploy the Docker images to the GKE cluster*: compila los ficheros *.yaml*, actualiza el servidor, y comprueba que se han creado todos los servicios correspondientes.

Además, se utilizan los *secrets* de GitHub para ocultar información sensible en alguno de los procesos previamente definidos.

Etapa 03: Configurar ficheros *.yaml* para *Kustomize*

Kustomize será la aplicación que nos permitirá instalar la infraestructura completa sobre el sistema *Kubernetes* del servidor de *Google Cloud*.

El primer paso consiste en **configurar los recursos base** de nuestra plataforma, que son la aplicación (*Omeka Classic*) y el gestor de la base de datos (*MySQL*).

Para configurar ambos recursos hay que crear los siguientes ficheros:

- *service.yaml*: configura el servicio del recurso.
- *deployment.yaml*: configura despliegue del recurso.
- *kustomization.yaml*: recoge los componentes (servicio y despliegue) del recurso. Es utilizado por *Kustomize* para construir el entorno.

En el repositorio del proyecto, estos ficheros se encuentran ubicados en las carpetas */gke-omeka/* y */gke-mysql/*.

A continuación, se modifica la plantilla base del recurso *gke-omeka* a través del fichero de configuración */patch.yaml*. En él se definen las variables de entorno que recogerán la información sensible de la aplicación (todas asociadas con un valor *secreto*).

Para finalizar, sobre el directorio raíz del repositorio, se crea el fichero de configuración principal */kustomization.yaml*. Este indicará a *Kustomize* qué recursos pretendemos instalar (*gke-mysql* y *gke-mysql*) y las modificaciones a realizar sobre la plantilla de la aplicación (*patch.yaml*).

Etapa 04: Crear los secretos y mapas de configuración en el servidor

Los *secrets* y *configMaps* utilizados por los ficheros *.yaml* de la etapa anterior tienen que estar definidos en el servidor de *Google Cloud*.

Para ello se ejecutan los siguientes comandos (sustituir los *<valores>* por los datos apropiados):

- *omeka-db*: *secretos* relacionados con la base de datos.

```
kubectl create secret generic omeka-db \
--from-literal=user-password=<contraseña_db_usuario> \
--from-literal=root-password=<contraseña_db_root> \
--from-literal=username=<nombre_usuario> \
--from-literal=database=<nombre_bd>
```

- *omeka-snmp*: *secretos* relacionados con el protocolo SNMP.

```
kubectl create secret generic omeka-snmp \
--from-literal=host=<host_snmp> \
--from-literal=username=<correo_electronico> \
--from-literal=password=<contraseña_correo> \
--from-literal=port=<puerto_snmp> \
--from-literal=ssl=<protocolo_seguridad_snmp>
```

- *omeka-imap*: *secretos* relacionados con el protocolo IMAP.

```
kubectl create secret generic omeka-imap \
--from-literal=host=<host_imap> \
--from-literal=username=<correo_electronico> \
--from-literal=password=<contraseña_correo> \
--from-literal=port=<puerto_imap> \
--from-literal=ssl=<protocolo_seguridad_imap>
```

- *db-config*: mapa de configuración para la base de datos.

```
kubectl create configmap db-config \
--from-file=./configFiles/db.ini.gke
```

- *snmp-config*: mapa de configuración para el protocolo SNMP.

```
kubectl create configmap snmp-config \
--from-file=./configFiles/config.ini.gke
```

- *imap-config*: mapa de configuración para el protocolo IMAP.

```
kubectl create configmap imap-config \
--from-file=./configFiles/mail.ini.gke
```

Etapa final: comprobar el funcionamiento del *workflow*

La última etapa consiste en ejecutar un *commit* sobre la rama *master* (siempre que el directorio afectado no sea */docs*). De esta manera, se comprueba que la acción recién creada se activa y finaliza correctamente.

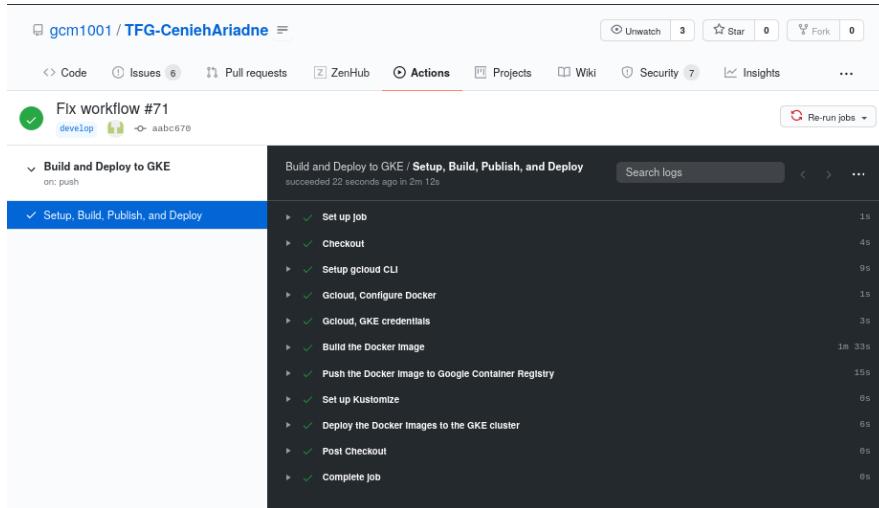


Figura D.2: Ejecución del workflow.

Codacy

Codacy proporciona una plataforma de revisión de código automatizada capaz de integrarse con múltiples repositorios, entre los que se encuentra *GitHub*.

Para poder utilizar esta plataforma con *GitHub* hay que seguir los siguientes pasos:

1. Instalar el complemento desde la [tienda oficial de Github](#).
2. Acceder a la plataforma [Codacy](#).
3. Ingresar con la cuenta de *GitHub* y, en la pantalla emergente, seleccionar el repositorio que deseamos integrar.
4. En la siguiente pantalla se da la posibilidad de añadir otras integraciones (como *Slack* o *JIRA*).
5. Esperar a que finalice la revisión de código.

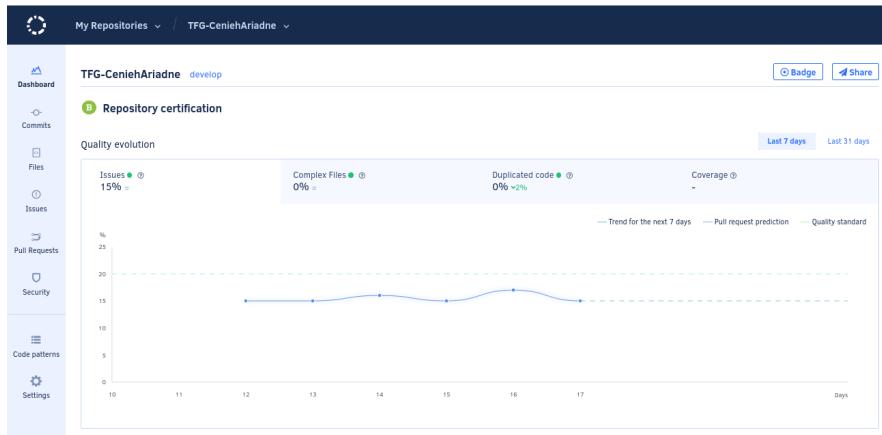


Figura D.3: Panel principal de la plataforma Codacy.

Tras esta primera revisión, cada vez que se ejecute un *commit* sobre la rama *main* del repositorio, *Codacy* evaluará la calidad de los cambios cometidos de forma automática.

Una ventaja de esta herramienta es que no necesita que el repositorio sobre el que está trabajando cuente con un fichero de configuración. Desde su plataforma, es posible realizar todas las tareas propias de configuración:

- Ignorar directorios.
- Activar / Desactivar patrones de código.
- Seleccionar las ramas a analizar.

- Gestionar las integraciones.
- Establecer las condiciones en las que los *commits* o *pulls request* son exitosos/fallidos.
- Indicar el umbral a partir del cual el repositorio es catalogado como “saludable”.

Read the Docs

Read the Docs es una plataforma web que facilita la tarea de documentar productos *software* automatizando la compilación, versionado y hospedaje de los ficheros generados por la herramienta de documentación *Sphinx*. En el repositorio del proyecto, estos ficheros se encuentran dentro del directorio */docs/sphinx/*.

Para utilizar este servicio, basta con iniciar sesión en su página web a través de *GitHub*, otorgar los permisos necesarios, e importar el repositorio (proyecto) sobre el que se integrará el servicio.

Además, se pueden configurar otros aspectos de la documentación. Para ello, es necesario indicar a la herramienta donde se encuentra el fichero de configuración *conf.py*, que en este proyecto se ubica también en */docs/sphinx/*.



Figura D.4: Página principal de la documentación del proyecto hospedada en Read The Docs.

D.4. Pruebas del sistema

Durante el desarrollo de los complementos (*plugins*), se han ido elaborado un conjunto de pruebas unitarias para comprobar el correcto funcionamiento de las distintas partes en las que estos se componen.

Para realizar esta tarea, se ha utilizado el *framework* de pruebas *PHPUnit*, el cual cuenta con una implementación adaptada a la estructura de la aplicación.

Antes de poder utilizar esta implementación, se debe configurar la sección de pruebas de la aplicación mediante el fichero de configuración *config.ini*. Este se encuentra localizado en el directorio `/application/tests/`.

Se deben indicar, al menos, los datos requeridos para la base de datos de prueba. **Es muy importante** que esta no sea la misma que la base de datos de la aplicación ya que, en cada ejecución de las pruebas, se ejecuta un *reset*.

A continuación se describen las propiedades de configuración del fichero *config.ini*:

- *db.host*: *hostname* donde se aloja la DB.
- *db.username*: nombre de usuario que tiene permisos en la DB.
- *db.password*: contraseña de usuario para acceder a la DB.
- *db dbname*: nombre de la DB.
- *paths.tempDir*: directorio temporal (se resetea por cada ejecución).

Configurada la base de datos de prueba, se pueden empezar a desarrollar las pruebas unitarias.

Para el desarrollo de pruebas unitarias existen dos clases fundamentales:

- *Omeka_Test_AppTestCase*: extiende a la clase *TestCase* de *PHPUnit*. La función más importante que ofrece esta implementación se llama *dispatch*. Esta permite poner a prueba las tres capas en las que está diseñada la aplicación: *modelo*, *vista* y *controlador*.
- *Omeka_Test_Helper_Plugin*: permite instalar e inicializar complementos durante la ejecución de las pruebas.

Existe un inconveniente en este sistema y es que si un complemento (*plugin*) depende de otro/s complemento/s, no es posible ponerlo a prueba. Por este motivo, solo se han desarrollado pruebas para aquellos complementos que no dependían de otro/s.

Todas las pruebas desarrolladas se encuentran dentro del directorio `/tests/` de cada complemento. A continuación, se exponen los resultados obtenidos en la ejecución de las pruebas.

```
gon@pop-os:/var/www/TFG-CeniehAriadne/omeka/plugins/AdminMenuDesign/tests$ phpunit .
PHPUnit 7.5.6 by Sebastian Bergmann and contributors.

..
2 / 2 (100%)

Time: 1.29 seconds, Memory: 20.00 MB

OK (2 tests, 15 assertions)
gon@pop-os:/var/www/TFG-CeniehAriadne/omeka/plugins/CollectionFiles/tests$ phpunit .
PHPUnit 7.5.6 by Sebastian Bergmann and contributors.

.....
6 / 6 (100%)

Time: 3.21 seconds, Memory: 24.00 MB

OK (6 tests, 40 assertions)
gon@pop-os:/var/www/TFG-CeniehAriadne/omeka/plugins/TagsManager/tests$ phpunit .
PHPUnit 7.5.6 by Sebastian Bergmann and contributors.

.....
13 / 13 (100%)

Time: 5.55 seconds, Memory: 20.00 MB

OK (13 tests, 81 assertions)
gon@pop-os:/var/www/TFG-CeniehAriadne/omeka/plugins/CENIEHExport/tests$ phpunit .
PHPUnit 7.5.6 by Sebastian Bergmann and contributors.

....
4 / 4 (100%)

Time: 2.14 seconds, Memory: 18.00 MB

OK (4 tests, 17 assertions)
```

Figura D.5: Resultados de la ejecución de las pruebas unitarias para cada uno de los complementos.

Apéndice E

Documentación de usuario

E.1. Introducción

En este apartado se pretende dotar al usuario de la información necesaria para la correcta utilización de la aplicación.

En primer lugar, se especificarán los requisitos *software* con los que el usuario debe contar para instalar y acceder a la aplicación. Posteriormente, se explicará paso a paso el proceso de instalación y, finalmente, se mostrará el manual de usuario.

E.2. Requisitos

Los requisitos de usuario varían en función del modo de instalación que vaya a llevar a cabo el usuario. Existen tres modos de instalación: Manual, *Docker* y *Kubernetes*.

Como es lógico, además de los requisitos que se van a mostrar a continuación, se debe contar con un **navegador web** desde el que se pueda acceder a la aplicación web. Este debe estar actualizado y tener activado el uso de *JavaScript*. A continuación se listan los navegadores sobre los que se han realizado pruebas:

- Firefox Browser (versión 78.0.2).
- Google Chrome (versión 84.0.4147.89).
- Opera (versión 69.0.3686.77).

Manual

Si se escoge la opción manual hay que estar seguro de que el servidor cumple con todos y cada uno de los siguientes **requisitos**:

- Sistema Operativo Linux [12]
- Apache HTTP Server [15] (con el módulo *rewrite* activado).
- MySQL / MariaDB v5.0 [13] o superior.
- PHP v5.4 [14] o superior con las siguientes extensiones instaladas:
 - mysqli
 - exif
 - curl
 - mbstring
- ImageMagick [10] (Tratamiento de imágenes)

Docker

En este caso, solo es necesario un único **requisito**:

- *Docker* [5] (Probado con la versión 19.03.6).

Kubernetes

Si se pretende utilizar *Kubernetes* para el despliegue de la infraestructura se requiere:

- *Docker* [5] (Probado con la versión 19.03.6).
- La herramienta de línea de comandos de *Kubernetes*, *kubectl* [11] (Probado en v1.18.2).
- *Kustomize* [6] (probado en v3.1.0)

E.3. Instalación

Como se ha comentado en el apartado anterior, existen tres posibilidades distintas para instalar la aplicación en un servidor: *Manual*, *Docker* o *Kubernetes*.

Antes de proceder con la instalación, es posible probar la aplicación desde ubucenh.es. Además, para poder acceder al área de administración (ubucenh.es/admin/), se ha creado un usuario de prueba cuyas credenciales son las siguientes:

- Nombre de usuario: *testUBU*
- Contraseña: *Usu4ri0dePrueb4*

Manual

El primer paso consiste en **configurar el servidor**. Para ello, hay que seguir una serie de indicaciones:

1. **Crear la base de datos (DB) MySQL** desde un usuario con permisos suficientes como para poder realizar operaciones sobre ella.

- Durante el proceso, conviene apuntar los siguientes datos:
 - *Hostname* donde se encuentra alojada la DB.
 - Nombre de la DB.
 - Nombre del usuario de la DB.
 - Contraseña de usuario de la DB.
- La base de datos ha de estar codificada en utf8.

```
sudo mysql -u root -
CREATE DATABASE omekadb CHARACTER SET utf8mb4 \
  COLLATE utf8mb4_unicode_ci;
CREATE USER 'usuario'@'localhost' IDENTIFIED BY 'contraseña';
GRANT ALL ON omekadb.* TO 'usuario'@'localhost' \
  IDENTIFIED BY 'contraseña' WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

2. **Descargar** la versión 2.7.1 de **Omeka**, desde su [web oficial](#) o desde su [repositorio oficial](#) en GitHub.

```
cd /tmp && wget \
  https://github.com/omeka/Omeka/releases/download/v2.7.1/omeka-2.7.1.zip
```

3. **Descomprimir** el fichero *.zip* recién descargado sobre un directorio desde donde podamos trabajar.

```
unzip omeka-2.7.1.zip -d <directorio_de_trabajo>
```

4. Desde el directorio escogido, buscar el fichero db.ini y **sustituir los valores 'XXXXX' por los datos de la base de datos** (anotados en el paso 1).

```
cd <directorio_de_trabajo>
nano db.ini
```

No es necesario modificar los parámetros ‘prefix’ o ‘port’.

```
[database]
host      = "localhost"
username  = "usuario"
password  = "contraseña"
dbname   = "omekadb"
prefix    = "omeka_"
charset   = "utf8"
;port     = ""
```

5. Descargar el contenido del repositorio del proyecto.

```
cd /tmp && wget \
https://github.com/gcm1001/TFG-CeniehAriadne/archive/master.zip
```

6. Descomprimir las carpetas /omeka/plugins y /omeka/themes del fichero .zip recién descargado.

```
unzip master.zip 'TFG-CeniehAriadne-master/omeka/plugins/*' \
'TFG-CeniehAriadne-master/omeka/themes/*' \
-d <*directorío_de_trabajo*>
```

7. Desde el directorio de trabajo, reemplazar las carpetas originales *plugins* y *themes* por las previamente descargadas.

```
cd <*directorío_de_trabajo*>
rm -rf ./plugins ./themes
sudo cp -r ./TFG-CeniehAriadne-master/omeka/* .
rm -rf ./TFG-CeniehAriadne-master
```

8. Mover todo el contenido del directorio de trabajo a la carpeta del servidor Apache.

```
mv -r <*directorío_de_trabajo*>/* <*directorío_del_servidor*>
```

9. Dar permisos de lectura y escritura sobre todo el contenido de la aplicación.

```
cd <*directorío_del_servidor*>
sudo chown -R www-data:www-data <*directorío_de_trabajo*>
sudo chmod -R 755 <*directorío_de_trabajo*>
```

10. Configurar el servidor Apache:

- 10.1. Crear el archivo de configuración correspondiente a la aplicación.

```
nano /etc/apache2/sites-available/omeka.conf
```

Cambiar los valores “*DocumentRoot*” y “*ServerName*”. . . code-block:

```

<VirtualHost *:80>
    ServerAdmin [email protected]
    DocumentRoot <directorio_del_servidor>
    ServerName <nombre_del_servidor>

    <Directory /var/www/html/omeka/>
        Options FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

```

10.2 Activar el sitio y el módulo rewrite y reiniciar el servidor

para aplicar los cambios.

```

a2ensite omeka.conf
a2enmod rewrite
systemctl restart apache2.service

```

Desde este instante, la aplicación será accesible desde el navegador (puerto 80). El último paso consiste en **completar la instalación guiada desde el navegador**, disponible en el directorio /install (e.g. <http://aplicacion.es/install>).

Una vez instalada la aplicación, para poder disfrutar de todas las mejoras propuestas en este proyecto, se deben instalar tanto los *plugins* como el tema propuesto (ver Instalar complementos (plugins) e Instalar temas (themes))

Por temas de seguridad, conviene eliminar la carpeta /install/ del directorio raíz una vez terminada la instalación de la aplicación.

Docker

Para proceder al despliegue **se deben descargar**, del [repositorio del proyecto](#), los siguientes ficheros:

- /Dockerfile
- /docker-compose.yml
- /ConfigFiles/*.modificar
- /omeka/plugins/*

Mantén los subdirectorios intactos.

Para poder enviar mensajes desde la aplicación deberás configurar el protocolo SNMP desde el fichero de configuración config.ini.modificar ubicado en la carpeta /ConfigFiles/.

Adicionalmente puedes configurar otras opciones desde ese mismo fichero o desde php.ini.modificar, también ubicado en la carpeta /ConfigFiles/.

A continuación debes **compilar la imagen**. Para ello, desde el directorio donde hayas almacenado la descarga anterior, ejecuta el siguiente comando:

```
docker build -t nombre_imagen:tag .
```

Recuerda muy bien el nombre de la imagen y el tag que pongas porque será necesario para el siguiente paso, que consiste en configurar el archivo docker-compose.yml.

En él, solo tenemos que cambiar la etiqueta image del servicio omeka_app con el nombre y el tag de la imagen recién compilada:

```
...
omeka_app:
    image: nombre_imagen:tag
```

Si se ha publicado la imagen en *DockerHub*, se puede hacer referencia a esta indicando el nombre de usuario seguido de la imagen (e.g. userame/nombre_de_mi_imagen:tag).

Por último, crear los *secrets* correspondientes a las contraseñas de la base de datos:

```
echo 'contraseña_usuario_db' | docker secret create omeka_db_password -
echo 'contraseña_root_db' | docker secret create omeka_db_root_password -
cp configFiles/db.ini.modificar configFiles/db.ini
cp configFiles/mail.ini.modificar configFiles/mail.ini
```

Debes modificar los ficheros recién creados (db.ini y mail.ini) con los datos relacionados con la base de datos y el protocolo IMAP. Ten en cuenta que la contraseña que introduzcas en el fichero db.ini tiene que coincidir con la del *secret omeka_db_password*.

Ahora ya se puede desplegar la infraestructura ejecutando el siguiente comando desde el directorio de trabajo (donde se encuentra la descarga del primer paso).

```
docker stack deploy -c docker-compose.yml nombre_del_entorno
```

Desde este instante la aplicación es accesible desde el navegador (puerto 80). Los siguientes pasos son los mismos que en la instalación manual.

Kubernetes

El primer paso para desplegar la aplicación mediante *Kubernetes* es montar nuestra imagen *Docker* (Sigue los primeros pasos del punto anterior, **hasta la compilación de la imagen**).

El siguiente paso consiste en desplegar la aplicación. Para esta tarea se utiliza el gestor de objetos *Kustomize*. Por ello, deberás contar con dicha herramienta. Además debes estar en posesión de los siguientes ficheros alojados en este repositorio:

- /kustomization.yaml
- /patch.yaml
- /gke-mysql/*
- /gke-omeka/*
- /configFiles/*.gke

Se deben definir en el servidor los *secrets* y *configMaps* utilizados por los ficheros de configuración *.yaml*.

Para ello se ejecutan los siguientes comandos (sustituir los *<valores>* por los datos apropiados):

- *omeka-db*: secretos relacionados con la base de datos.

```
kubectl create secret generic omeka-db \
--from-literal=user-password=<contraseña_db_usuario> \
--from-literal=root-password=<contraseña_db_root> \
--from-literal=username=<nombre_usuario> \
--from-literal=database=<nombre_bd>
```

- *omeka-imap*: secretos relacionados con el protocolo IMAP.

```
kubectl create secret generic omeka-imap \
--from-literal=host=<host_imap> \
--from-literal=username=<correo_electronico> \
--from-literal=password=<contraseña_correo> \
--from-literal=port=<puerto_imap> \
```

```
--from-literal=ssl=<protocolo_seguridad_imap>
```

- *omeka-snmp*: secretos relacionados con el protocolo SNMP.

```
kubectl create secret generic omeka-snmp \
--from-literal=host=<host_snmp> \
--from-literal=username=<correo_electronico> \
--from-literal=password=<contraseña_correo> \
--from-literal=port=<puerto_snmp> \
--from-literal=ssl=<protocolo_seguridad_snmp>
```

- *db-config*: mapa de configuración para la base de datos.

```
kubectl create configmap db-config \
--from-file=./configFiles/db.ini.gke
```

- *snmp-config*: mapa de configuración para el protocolo SNMP.

```
kubectl create configmap snmp-config \
--from-file=./configFiles/config.ini.gke
```

- *imap-config*: mapa de configuración para el protocolo IMAP.

```
kubectl create configmap imap-config \
--from-file=./configFiles/mail.ini.gke
```

Por último, debemos indicar el identificador de nuestra imagen *Docker* en el fichero `/gke-omeka/deployment.yaml`.

```
...
spec:
  containers:
    - image: nombre_imagen:tag
...

```

A continuación, desde el directorio raíz, ejecutar el siguiente comando:

```
kustomize build . | kubectl apply -f -
```

Desde este instante, la aplicación es accesible desde el navegador (puerto 80). Los siguientes pasos son los mismos que en la instalación manual.

E.4. Manual del usuario

Gracias a *Read the Docs* es posible acceder al manual de usuario desde el siguiente enlace:

https://tfg-ceniehariadne.rtfd.io/es/latest/anexos/E_Manual_usuario.html#manual-de-usuario

The screenshot shows the homepage of the TFG-CeniehAriadnePlus documentation. On the left, there's a sidebar with navigation links for 'MEMORIA' (Introducción, Objetivos del proyecto, Conceptos teóricos, Técnicas y herramientas, Aspectos relevantes del desarrollo del proyecto, Trabajos relacionados, Conclusiones y Líneas de trabajo futuras) and 'ANEXOS' (Plan de proyecto software, Especificación de requisitos, Especificación de diseño, Documentación técnica de programación). Below these are sections for 'Documentación de usuario' (Introducción, Requisitos de usuario, Instalación, Manual de usuario). The main content area has a title 'Manual de usuario'. It includes a yellow 'Advertencia' box stating that the manual is not valid for the original Omeka Classic version and a blue 'Nota' box linking to the original manual. A section titled 'Área de administración' describes the administration area as a place to manage data sets and configure other application aspects. At the bottom right is a screenshot of the Omeka login interface.

Figura E.1: Página principal del manual de usuario.

Bibliografía

- [1] CENIEH. Ariadneplus — proyectos, 2020. [Internet; Accedido 13-junio-2020].
- [2] CENIEH. Centro nacional de investigación sobre la evolución humana, 2020. [Internet; Accedido 13-junio-2020].
- [3] Creative Commons, 2020. [Internet; Accedido 19-junio-2020].
- [4] Agencia Estatal de Administración Tributaria. Cuadro informativo tipos de retención aplicables (2020), 2020. [Internet; Accedido 13-junio-2020].
- [5] Docker. Documentation — install docker engine, 2020. [Internet; Accedido 21-junio-2020].
- [6] Github. Kustomize, 2020. [Internet; Accedido 21-junio-2020].
- [7] GitHub. The world's leading software development platform, 2020. [Internet; Accedido 13-junio-2020].
- [8] GNU. General public license v3.0, 2020. [Internet; Accedido 13-junio-2020].
- [9] Google. Google cloud, 2020. [Internet; Accedido 21-junio-2020].
- [10] ImageMagick. Install for unix and windows source, 2020. [Internet; Accedido 21-junio-2020].
- [11] Kubernetes. Documentación — instalar y configurar kubectl, 2020. [Internet; Accedido 21-junio-2020].
- [12] Linux. What is linux, 2020. [Internet; Accedido 21-junio-2020].
- [13] MySQL. Documentation — installation guide, 2020. [Internet; Accedido 21-junio-2020].

- [14] PHP. Manual — instalación y configuración, 2020. [Internet; Accedido 21-junio-2020].
- [15] Apache HTTP Server Project. Documentation — install, 2020. [Internet; Accedido 21-junio-2020].
- [16] Scrum. The home of scrum, 2020. [Internet; Accedido 13-junio-2020].
- [17] Seguridad Social. Bases y tipos de cotización 2019, 2020. [Internet; Accedido 13-junio-2020].
- [18] UBU. Universidad de burgos, 2020. [Internet; Accedido 13-junio-2020].
- [19] Wikipedia. Lamp — wikipedia, the free encyclopedia, 2020. [Internet; Accedido 21-junio-2020].
- [20] ZenHub. Agile project management for github, 2020. [Internet; Accedido 13-junio-2020].

This work is licensed under a [Creative Commons Attribution 4.0 International License](#).

